



Representing hydrodynamically important blocking features in coastal or riverine lidar topography

B. R. Hodges

Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, Austin, USA

Correspondence to: B. R. Hodges (hodges@utexas.edu)

Received: 1 December 2014 – Published in Nat. Hazards Earth Syst. Sci. Discuss.: 13 February 2015

Revised: 27 April 2015 – Accepted: 28 April 2015 – Published: 21 May 2015

Abstract. New automated methods are developed for identifying narrow landscape features that cause hydrodynamic blocking and might have critical impacts for management models of river flooding, coastal inundation, climate change, or extreme event analysis. Lidar data processed into a fine-resolution raster ($1\text{ m} \times 1\text{ m}$) can resolve narrow blocking features in topography but typically cannot be directly used for hydrodynamic modeling. For practical applications such data are abstracted to larger scales, which can result in a loss of hydrodynamic blocking effects. The traditional approach to resolving hydrodynamic blocking features is to represent them as cell boundaries within a customized unstructured grid that is tuned to the spatial features. A new automated edge-blocking approach is developed, which allows application of an arbitrarily structured (Cartesian) mesh at coarser scales and provides contiguous representation of blocking features along Cartesian cell boundaries. This approach distorts the shape of a blocking feature (i.e., making it rectilinear along grid cell faces) but retains its critical hydrodynamic blocking height characteristics and spatial continuity within the topographic model.

mate patterns (Wen et al., 2013; Vastila et al., 2010). Unfortunately, modeling annual to decadal timescales for management and climate change analyses typically requires hydrodynamic model grid scales that might not adequately represent narrow blocking features. Herein we develop new methods for upscaling a digital elevation model of topography to ensure hydrodynamic blocking features are retained.

The working hypothesis of this paper is that at any sufficiently coarse-grid scale (ΔC) there might be topographic features of width scale $\ell_W < \Delta C$ and length scale $\ell_L \geq \Delta C$ that can be represented as “edge” or “face” features of the grid cell. These features, if given the correct continuity across multiple grid cells, can represent hydrodynamic blocking that is lost when subgrid features are represented as topographic roughness. We will call this an *edge-blocking* technique.

By way of background, the present state of the art for processed lidar data can readily provide a $\sim 1\text{ m} \times 1\text{ m}$ digital terrain model (DTM) for use in high-resolution hydrodynamic modeling (Schubert et al., 2008; Sampson et al., 2012). Unfortunately, modeling at such fine spatial resolution invariably drives the model time step down to 1 s or lower, depending on the numerical model scheme¹. At such scales, even a small river delta of 10^4 ha will require 10^8 grid cells and approximately 10^7 time steps per year of simulation – pushing even a small system into supercomputer territory for multi-decadal and/or Monte Carlo simulations for sensitivity analyses, which are both desirable for adaptive coastal management. However, by coarsening to a $20\text{ m} \times 20\text{ m}$ grid resolution, 10^4 ha requires only 2.5×10^5 grid cells and can typically be run at time steps of 10 to 30 s in a semi-implicit

1 Introduction

Hydrodynamic models are useful tools for exploring how climate change, rising sea levels, and hydrological regime alterations might affect the interaction between tides, rivers, and coastlines (Purvis et al., 2008; Bhuiyan and Dutta, 2012; Nardin and Edmonds, 2014), as well as urban coastal flooding (Gallien et al., 2013; Webster et al., 2014). Similarly, such models are vital in analysis of river hydrodynamics and floodplain inundation that might be affected by changing cli-

¹The BreZo code of Sanders et al. (2010) required time steps of 0.05 s for 0.8 m triangular mesh in Schubert et al. (2008)

model², i.e., $\sim 10^6$ time steps per year of simulation. The reduced memory requirement allows multiple model instances to be simultaneously run on a standard multi-core desktop workstation. The larger allowable time step allows faster simulations over longer timescales without requiring extensive high-end computational resources. Indeed, it is likely that, as computers get more powerful, our desire to decrease grid resolution will be countered by our desire to run larger areas and longer timescale simulations. Thus, the need for grid coarsening of lidar data for hydrodynamics is unlikely to disappear, and the challenge we face is in upscaling the topography while capturing the hydrodynamic effects of unresolved features.

As an alternative to grid coarsening, grid nesting (from fine to coarse grids) can be applied to somewhat reduce computational costs (e.g., Webster et al., 2014), but such methods inherently require an expert modeler to make a judgement as to where model resolution might be coarsened. Artificial porosity approaches seem appropriate for urban areas with multiple pathways around unresolved objects (Sanders et al., 2008), but it is not clear that they are necessarily useful in broader natural settings or where narrow objects are blocking over multiple cells. Quadtree subgrid nesting for hydrodynamic models appears to be one approach for upscaling effects of fine-resolution topography without resorting to edge features (Stelling, 2012; Volp et al., 2013), but such methods are still in the early stages of development and cannot be considered a definitive solution. Indeed, it is not clear that simple application of quadtree methods would necessarily preserve contiguous blocking edges, so combining quadtree with the present edge-blocking technique might be necessary.

Upscaling of lidar data to a coarser grid presents hydrodynamic modeling challenges (Fewtrell et al., 2008). Prior to lidar, our topographic data were generally coarser than model grid scales (Bates, 2004), so translating topography to a model grid was a matter of simple interpolation from a sparse data set and calibrating roughness for effects of unknown features. With the advent of lidar, as noted by Bates et al. (2003), “we need methods to identify and connect linear topographic features . . . given their significant hydraulic impact.” Indeed, in associated work of Cobby et al. (2003), Bates’ research team pioneered the use of image-processing techniques (skeletonization) for linear features that were either incorporated in a highly resolved mesh or represented as roughness. Nevertheless, despite more than 150 citations of these pioneering works, the general challenge of automatically *identifying and connecting linear topographic features* has not been previously addressed, and thus provides the fundamental motivation for the present work. Herein we address this challenge with a new automated method for identifying, extracting, and representing topographic features that extend over multiple coarse-grid cells but are so narrow that their

hydrodynamic blocking effects would be lost in common upscaling techniques.

As a simple example of the challenge, consider the satellite photograph and corresponding lidar data in Fig. 1a and b, which show a portion of a railroad dike that cuts across the Nueces River delta in Texas, USA, just outside the City of Corpus Christi. The dike is approximately 5 m across at the top and 15 m across at the base. If the lidar data are rasterized to a 20 m \times 20 m grid using a simple arithmetic mean of the finer-scale data (Fig. 1c), the dike loses both its overall blocking height and continuity. A hydrodynamic model using this coarser grid would have flow paths through the dike at 1.5 to 2 m a.s.l. (above sea level) rather than being contiguously blocked at a 3 m elevation.

Where dikes, embankments, and complex topography are narrower than a practical hydrodynamic grid, the traditional solution has been use of unstructured grids designed with cell boundaries coincident to narrow blocking features (e.g., Cobby et al., 2003; Tsubaki and Fujita, 2010). Unfortunately, unstructured grids usually require significant expertise and “hands-on” artistry to develop an acceptable balance between hydrodynamic modeling practicality and fidelity to the physical topography (Schubert et al., 2008; Mandlburger et al., 2009). As another approach, Ryan and Hodges (2011) modeled the 7500 ha of the Nueces River delta with a structured Cartesian grid, where the narrow railroad dike across the delta (Fig. 1) was represented as an elevated edge feature in a 2-D hydrodynamic model. Identifying this contiguous edge feature on the raster grid was a labor-intensive manual task, but it was critical to obtaining the correct hydraulic blocking effects.

It can be stated without reservation that hydrodynamic modeling is simpler if a single size and shape of grid element is used across an entire domain. Such a grid might be called an “arbitrary” grid as, by definition, it is not tuned to match any particular topographic feature. The problem with arbitrary grids is that at coarse resolution they will not have the proper connectivity of either narrow flow paths or narrow blocking structures. Thus, the simplicity of arbitrary gridding does not necessarily imply a better model. However, if narrow blocking features can be upscaled to the edges of the arbitrary grid cells and the correct continuity preserved over multiple cells, then the feature’s effect on flow blocking can be retained, albeit at some distortion of the shape and location of the object. However, such distortions are inherently at the model grid scale and hence should be acceptable for a coarse-grid model. That is, both the landscape and an associated hydrodynamic model are subject to distortions scaling on the grid size (e.g., a single grid cell has a uniform elevation and roughness and a single velocity representing the flow); thus, the distortion caused by moving a subgrid blocking feature from the grid cell center to its edge is consistent with approximations associated with the selection of the coarse-grid scale.

²Depending on the hydrodynamic discretization and the typical flow velocities.

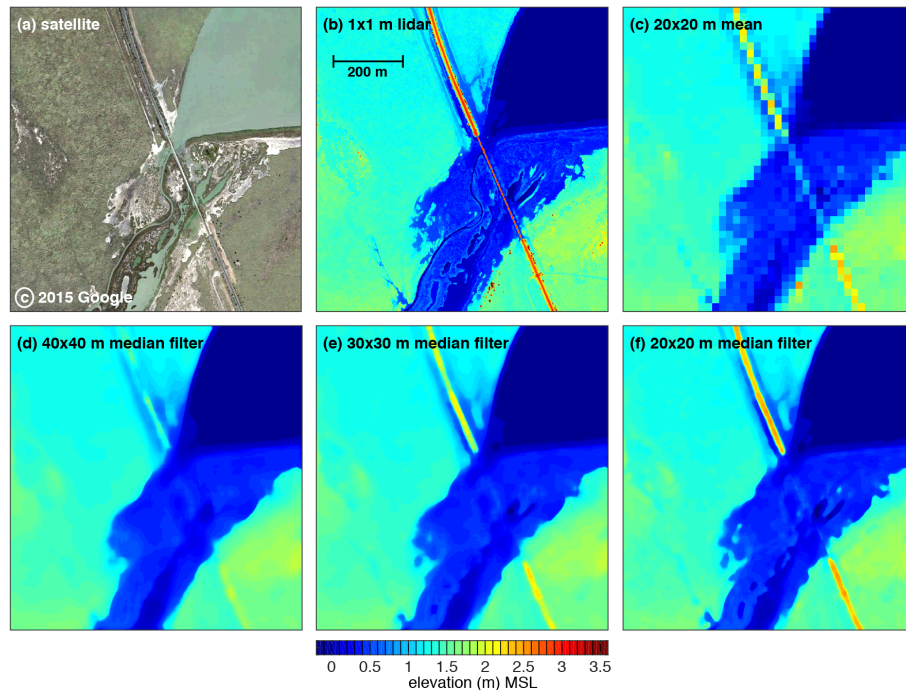


Figure 1. A 75 ha section of the Rincon Bayou in the Nueces River delta shown in a Google Earth satellite photo (a), which is centered at $27^{\circ}53'20''$ N, $97^{\circ}34'11''$ W. Comparing the $1\text{ m} \times 1\text{ m}$ resolution lidar DTM (b) courtesy of J. Gibeaut, Texas A & M Corpus Christi, and the arithmetic mean of the data computed on a $20\text{ m} \times 20\text{ m}$ grid (c) illustrates the loss of hydrodynamic blocking height and continuity when a simple mean is used for coarse-grid elevations (results for a $20\text{ m} \times 20\text{ m}$ median, not shown, are almost indistinguishable). Frames (d) through (f) show applications of median filtering (see Methods) with varying filter scales, retaining the original $1\text{ m} \times 1\text{ m}$ resolution but eliminating unresolvable features from the data set. All elevations are relative to mean sea level (m.s.l.).

For an arbitrary gridding approach to actually simplify modeling, the definition of the coarse-grid edge blocking must be automatic – otherwise we are back to treating grid definition as a labor-intensive art form. The “hands-off” approach introduced herein captures both the obvious large-scale features (such as the railroad dike) and smaller features that are more difficult to identify. The new approach uses Cartesian grids that are relatively easy to create, modify, and hydrodynamically model. Although unstructured grids have been popular over the past two decades as a way to direct computational power at specifically desired scales, one can make the argument that continually increasing computer power will eventually lead to a return to structured grid modeling that provides simpler automation, requires less expertise in model and mesh development, and allows for easier communication between models.

This paper provides a set of methods to represent fine-scale topographic data to allow hydrodynamic modeling of blocking features at coarser scales on a Cartesian grid. Full implementation and testing of these ideas require a hydrodynamic model, whose characteristics will influence the resulting behaviors and would require detailed description and investigation. To keep the focus of this paper on the topographic techniques, analysis of the hydrodynamic solution is

reserved for future investigations and will not be addressed herein.

2 Methods

The goal of our lidar processing is to produce a coarse Cartesian grid at some scale, ΔC , that retains valuable hydrodynamic characteristics associated with contiguous blockage features visible at a finer resolution, ΔF (e.g., the railroad dike in Fig. 1). In a conventional Cartesian raster grid, each cell has a single piece of data: the landscape elevation. However, for the coarse grid we will store a representative value for the elevation over the bulk of the grid cell and separate blocking elevation values on each cell face, which is typically allowed in Cartesian grid models (e.g., Casulli and Cheng, 1992) and in some unstructured grid models (Ge et al., 2012). For convenience in data processing, we will limit our focus to systems with integer values of the coarse-to-fine raster ratio, defined as

$$R_{\Delta} = \frac{\Delta C}{\Delta F}. \quad (1)$$

For sufficiently fine resolution lidar data, requiring integer values for R_{Δ} is not a significant limitation. It follows that the coarse-grid raster is of size $n_{cx} \times n_{cy}$ such that

$$n_{cx} = \frac{n_{fx}}{R_{\Delta}}, \quad (2)$$

$$n_{cy} = \frac{n_{fy}}{R_{\Delta}}, \quad (3)$$

where n_{fx} and n_{fy} are the number of fine-grid cells in the data set (this approach generally requires truncating some fine-grid data at edges to ensure integer values for n_{cx} and n_{cy}). For simplicity in exposition, we will confine ourselves to the case where the x and y directions are resolved with the same R_{Δ} , although the method is readily extensible to rectangular vice square grid cells. Further extension to unstructured grids is theoretically possible, but the methods developed herein would require significant modification for a non-Cartesian mesh.

The general procedure for data processing is as follows:

1. create a fine-grid background topography (Fig. 1d),
2. create a coarse-grid representation of background topography (Fig. 2a),
3. compute the difference between fine and coarse topography (Fig. 2b),
4. identify contiguous objects that occur in the difference set (Fig. 3),
5. identify blocking objects and assign elevations to grid cell faces (Figs. 4 and 5).

Fine-scale background topography – the first step is to separate unresolvable topographic features (at coarse-scale ΔC) from a background topography, i.e., estimating what the fine-scale landscape would look like with coarse-scale unresolvable features removed. Herein we apply a *median filter*, which was originally designed for image noise removal but has proven more widely useful, e.g., removing the signature of large woody debris from bathymetric data (White and Hodges, 2005). A median filter replaces the value at position (x, y) with the median of the values in some neighborhood around the point; the neighborhood size is defined as the filter size, ΔM . The filter operation is accomplished on a moving window over the fine-scale grid to produce a smooth rendition of the background elevations that are resolvable at a coarser resolution; that is, the original resolution of the data set is maintained (e.g., unlike the averaging in Fig. 1c), while the unresolvable features are removed. For example, the $1 \text{ m} \times 1 \text{ m}$ lidar data are processed with different-size median filters as shown in Fig. 1d–f, providing smooth, high-resolution background elevations.

Clearly, the filter scale for defining background topography should be equal to or greater than the desired coarse-grid scale, i.e., $\Delta M \geq \Delta C$. If the median filter is smaller than the coarse grid, then objects that cannot be resolved will remain in the filtered data set and hence not in the difference data set (discussed below). Indeed, it seems prudent to generally apply a median filter of $\Delta M \approx 2\Delta C$ to ensure that objects only

slightly larger than a grid cell are readily identified. That is, if a $20 \text{ m} \times 20 \text{ m}$ coarse grid is desired and a $20 \text{ m} \times 20 \text{ m}$ median filter is applied, there can be features slightly larger than 20 m that will appear across two coarse-grid cells and hence will not really be hydrodynamically resolved. This effect is clear in Fig. 1f, where the $20 \text{ m} \times 20 \text{ m}$ filter shows a partial signature of the railroad dike that would be lost in upscaling to the coarse grid as in Fig. 1c. Sensitivity of blocking identification to the choice of the ΔM parameter is provided in the Results section, below.

Coarse background and difference data set – the median filtered fine-scale data, Fig. 1d, are used to produce a coarse-grid approximation of the landscape elevation, Fig. 2a. This step can be accomplished using either the simple arithmetic mean or median of elevations inside the coarse-grid cell (herein the median is used). This coarse-grid representation is pushed back to the fine grid (i.e., using identical values for all the fine cells within a single coarse-grid cell) so that a fine-scale difference map can be created (Fig. 2b). Note that it is also possible to directly use the median filtered fine-scale data for the difference map (i.e., without pushing back to the coarse grid), but this approach affects the interpretation of the difference value relative to the coarse-grid cell elevation.

Identification of objects – the difference map (Fig. 2b) contains both negative objects (unresolved depressions) and positive objects (unresolved blockages). The present work focusses on the positive objects³ that are relatively easy to handle in a hydrodynamic model that includes cell edge elevations. To identify blocking positive objects, a cutoff height (Δh) is specified above which an object is deemed a hydrodynamic blockage rather than topographic roughness. The number and size of objects will be a function of this cutoff. The object size, i.e., the number of fine grid cells in a contiguous blocking object, is denoted as N with subscripts to distinguish the x and y faces of a coarse-grid cell. For the present demonstration, the cutoff height is $\Delta h = 0.2 \text{ m}$. Sensitivity of blocking results to the choice of Δh is provided in the Results section, below.

A binary data set can be defined as $\{0, 1\}$ based on whether fine-grid cells are respectively below or above Δh , as shown in Fig. 3a. Our goal is to identify discrete objects, where an *object* is defined as a contiguous set of cells with the value of 1. Algorithms to identify connected cells are relatively easy to write but are difficult to make efficient for large data sets. Fortunately, binary object identification is a standard image-processing task and efficient algorithms are available (Gonzalez et al., 2004).

However, even efficient algorithms have computational costs scaling on the number of objects, so it is useful to first remove single cells, i.e., where $N = 1$, which cannot cause hydraulic blocking. In image processing, single pixels in a

³Note that negative objects are fine-scale flow paths that are not represented within the coarse-grid topography and are an important subject for further research.

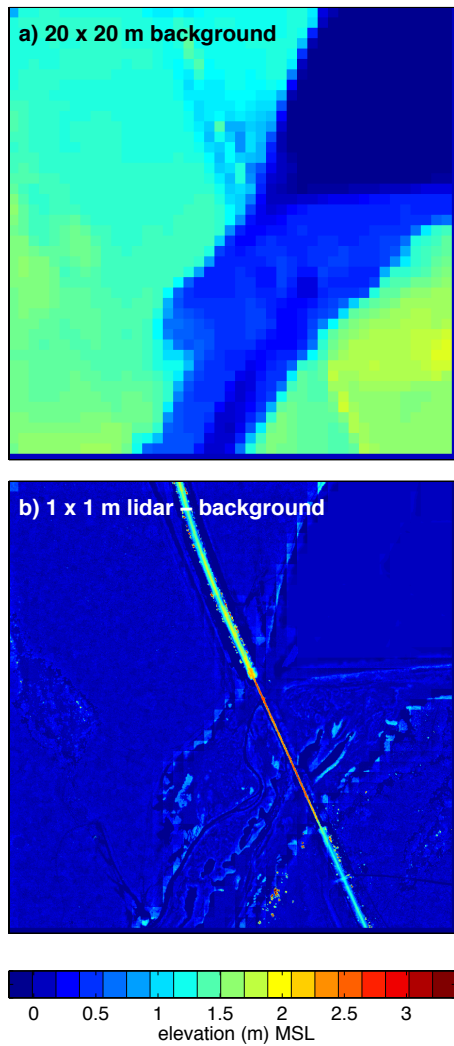


Figure 2. (a) The 20 m × 20 m coarse-grid background topography based on the 40 m × 40 m filter of Fig. 1d; (b) the difference between the 1 m × 1 m lidar data of Fig. 1b and the coarse-grid background topography.

binary data set are considered “noise” and addressed with noise-removal techniques. Again, one can readily write an algorithm to check the neighbors for each cell with a value of 1 and eliminate all those whose neighbors are all 0, but it is simpler to rely on the expertise of the image-processing community for this task. In the present work the Matlab Image Processing Toolbox *bwmorph* function with the “clean” option was used to remove single-cell objects. Identification of multi-cell objects was accomplished with the Moore neighbor tracing algorithm (Gonzalez et al., 2004) as implemented in the *bwboundaries* function in Matlab. The resulting objects are shown in Fig. 3b.

It is also useful to remove objects that are too small to block a coarse-grid cell; i.e., for a 20 m × 20 m coarse grid based on a 1 m × 1 m data set ($R_{\Delta} = 20$), any object in Fig. 3a

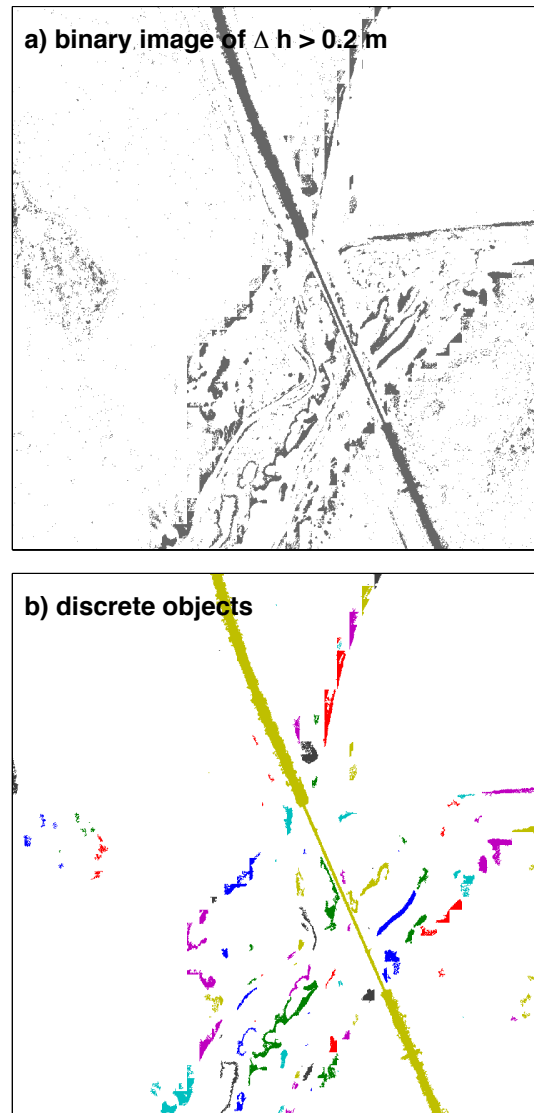


Figure 3. Binary image (a) of difference data set from Fig. 2b, which can be used to identify separate objects, shown in colors in (b). Note that objects smaller than 20 fine-grid cells in (a) have been eliminated in (b).

that consists of fewer than 20 fine-grid cells cannot hydraulically block a coarse-grid cell and can be excised from the object data set. To allow some flexibility, it is useful to define a small object removal criterion based on N blocking cells as

$$N \leq R_{\Delta} - \delta, \tag{4}$$

where $\delta \in \{0, 1, 2, \dots\}$ is a user-defined parameter that allows nearly blocking objects ($\delta > 0$) to be retained in the object data set. Herein $\delta = 1$ is used.

Snap-grid object blocking – each object can be processed separately to provide hydraulic blocking conditions on both row and column faces of a coarse-grid cell. A typical object and its elevations (Fig. 4a) has a binary representation

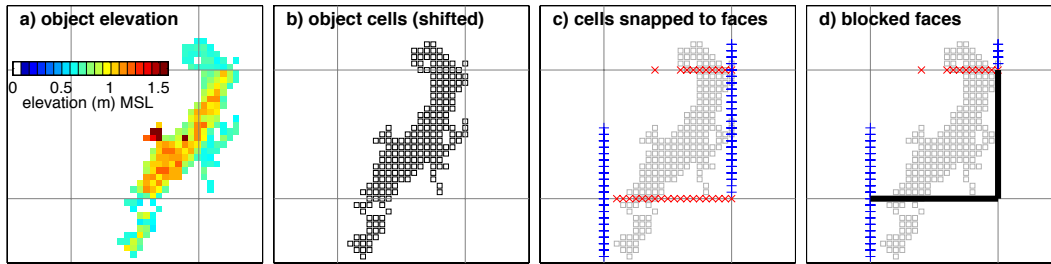


Figure 4. Blocking caused by a small object. The red \times and blue $+$ represent fine-grid blocking cells (gray \square) snapped to the coarse-grid faces G_y and G_x ; black lines are the resulting blocked coarse-grid faces. Here and throughout this paper $\delta = 1$ is used for defining blocking. The object cells in (b) are slightly shifted from (a) as discussed in the text to better align with coarse grid.

at the fine-grid level, as shown in Fig. 4b. Let $f_o(x_c, y_c)$ be the set of fine-grid cells of the object, where x_c and y_c are coordinates measured relative to the coarse grid. The coarse-grid demarcation lines are, by definition, integer values in our coarse-grid numbering scheme. This is perhaps slightly unconventional for hydrodynamic modelers as the coarse-grid cell centers are therefore at non-integer values; i.e., the location

$$(x_c, y_c) = (i - 1/2, j - 1/2) \quad (5)$$

defines a coarse-grid cell in the raster set for

$$i = \{1, 2, 3, \dots, n_{cx}\} \quad (6)$$

$$j = \{1, 2, 3, \dots, n_{cy}\} \quad (7)$$

with faces at coarse-grid indexes

$$(x_c, y_c) \in \{(i, j - 1/2), (i - 1, j - 1/2), \dots, (i - 1/2, j), (i - 1/2, j - 1)\}. \quad (8)$$

Using this indexing, we can define

$$G_y(i - 1/2, j) = f_o(x_c, \text{round}(y_c)) \quad (9)$$

as a set of fine-grid cells of varying x_c that are “snapped” to an integer y_c face (the red \times markers in Fig. 4c). Similarly

$$G_x(i, j - 1/2) = f_o(\text{round}(x_c), y_c) \quad (10)$$

is the set of blocking cells of varying y_c snapped to an integer x_c face (the blue $+$ markers in Fig. 4c). For the purposes of determining whether or not blocking occurs, the cells sets G_x and G_y are true mathematical sets that do not include duplicate values. However, data sets with duplicates (denoted as G_{yy} and G_{xx}) are retained for computation of blocking height (discussed below).

To determine snap-grid blocking of coarse-grid faces (Fig. 4d), the sizes of set $G_y(i - 1/2, j)$ and $G_x(i, j - 1/2)$ are defined as $N_y(i - 1/2, j)$ and $N_x(i, j - 1/2)$, respectively. These are the number of unique values of x_c on a round (y_c) face (red \times markers on a column face) and the number of

unique y_c values on a round(x_c) face (blue $+$ markers on a row face). Snap-grid blocking occurs along coarse-grid column faces that have a number of blocking fine-grid cells exceeding the removal criterion of Eq. (4), that is,

$$N_y(i - 1/2, j) \geq R_\Delta - \delta, \quad (11)$$

and coarse-grid row faces that satisfy

$$N_x(i, j - 1/2) \geq R_\Delta - \delta. \quad (12)$$

Note that this approach allows the fine cells to serve as blocking in the both x and y directions simultaneously, which is necessary to represent the hydraulic blocking of objects at an angle to the coarse grid. After a face is blocked, e.g., as shown in Fig. 4d, the corresponding $G_y(i - 1/2, j)$ and $G_x(i, j - 1/2)$ are set to 0 so that the fine-grid cells used to define a snap-grid block are not used in computing *cross-cell blocking* (discussed below).

Small object shift – the snap-grid blocking approach will necessarily depend on the spatial relationship between the objects and the coarse grid. For small objects, a slight shift of the object position can change whether or not the object is judged to be blocking. For example, the lower column face blocking in Fig. 4d would not have been identified as blocking in the original object position, shown in Fig. 4a, because some of the fine-grid cells would have shown up in an adjacent column such that Eq. (11) would not have been satisfied for either face. For small objects (overall length less than $1.5\Delta C$), it is convenient to simply shift the object (as in Fig. 4b) to maximize the number of fine-grid blocking cells within a single coarse-grid cell. As long as the shift is fewer than $\Delta C/4$ cells, it does not significantly affect the coarse-grid physical relationships. Object shifting can be accomplished with an automated algorithm that is based on the total extent of a small object and the overhang of the object into adjacent cells.

Cross-cell object blocking – depending on an object’s topology, the blocked faces determined by the snap-grid approach (described above) might not provide a contiguous blocked path. Consider the larger object in Fig. 5a, where snap-grid blocking provides the results in Fig. 5b and c. It

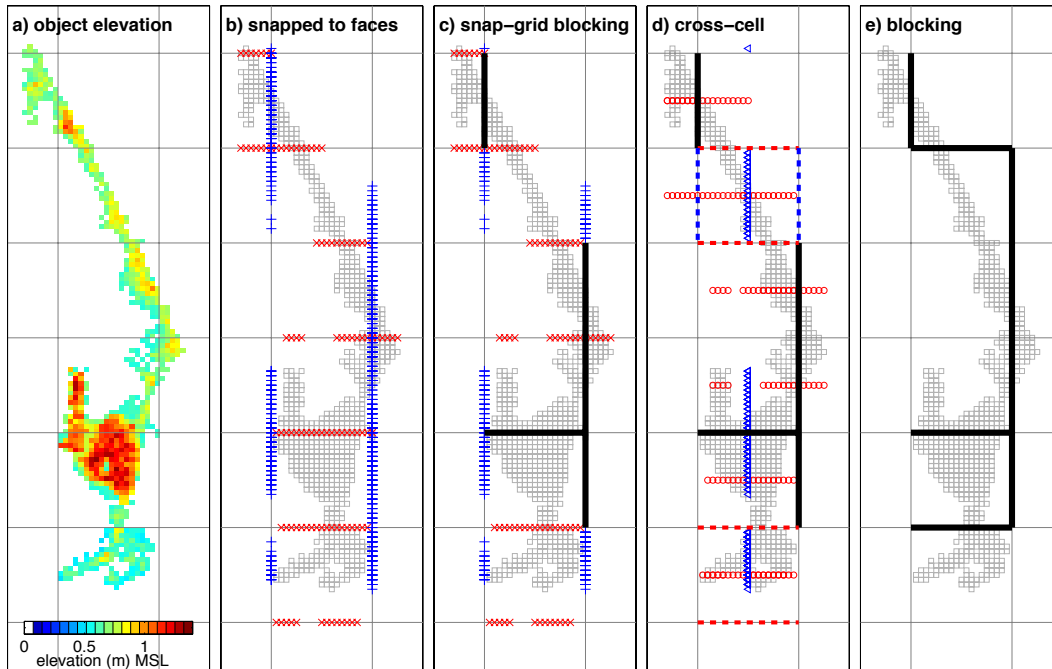


Figure 5. Blocking caused by a large object similar to Fig. 4 for frames (a) through (c). In (d) the remaining G_y and G_x are transformed to cell-center H_y and H_x blocking shown with red \circ and blue Δ , and different possible blocking paths are illustrated with dashed lines; (e) shows final blocking paths that are contiguous.

is clear that some hydraulic blocking has not been captured: the G_x (the blue + markers) and G_y (red x markers) in the second coarse-grid cell from the top do not satisfy Eqs. (11) or (12) as they are split between different faces. Furthermore in the lowermost coarse-grid cell, the red x markers on either face are insufficient for blocking. These effects arise because the snap-grid approach uses rounding, which can split the blocking fine-grid cells to the upper and lower faces of the coarse-grid cell. To address this issue, we define a cross-cell blocking approach for column and row faces. Cross-cell blocking is conducted after snap-grid blocking and only uses the fine-grid cells that *were not* applied in snap-grid blocking, e.g., the remaining red x and blue + in Fig. 5c.

For the coarse-grid cell centered at $(i - 1/2, j - 1/2)$, we define sets of unique fine-grid blocking cells across the cell center as

$$H_y(i - 1/2, j - 1/2) = G_y(i - 1/2, j) \cup G_y(i - 1/2, j - 1), \quad (13)$$

$$H_x(i - 1/2, j - 1/2) = G_x(i, j - 1/2) \cup G_x(i - 1, j - 1/2), \quad (14)$$

which are illustrated in Fig. 5d. Blocking conditions are defined similar to Eqs. (11) and (12), using N_y and N_x as the size of the unique H_y and H_x cell sets. For determining object blocking height (discussed below), we also define sets that retain non-unique elements:

$$H_{yy}(i - 1/2, j - 1/2) = G_{yy}(i - 1/2, j) \cup G_{yy}(i - 1/2, j - 1), \quad (15)$$

$$H_{xx}(i - 1/2, j - 1/2) = G_{xx}(i, j - 1/2) \cup G_{xx}(i - 1, j - 1/2). \quad (16)$$

As the H_y and H_x fine-cell sets cross the coarse-cell center, face blocking could be at either of the grid cell faces, as shown by dashed blocking lines in Fig. 5d. Indeed, there can be more than one set of blocking faces that provides a reasonable representation of cross-cell blocking. The critical issue is ensuring that cross-cell blocking is contiguous; i.e., there are choices for cross-cell blocking faces in Fig. 5d that would *not* provide contiguous blocking. The simplest algorithm for selecting blocking is to process column faces (red \circ) and row faces (blue Δ) sequentially. If a coarse cell contains only a single blocked end point (e.g., the lowermost complete cell in Fig. 5d), then the cross-cell blocked face (either row or column) must connect to that blocked end. If a coarse cell contains two blocked end points, then the algorithm must distinguish between diagonally blocked points (e.g., the second coarse cell from the top in Fig. 5d) and co-linear blocked end points along a face (either row or column). Where two blocked end points are along a single column face and cell-center blocking for a column face exists (red \circ), the cell-center blocking logically must be along the face that connects the blocked end points. Similarly, where two blocked

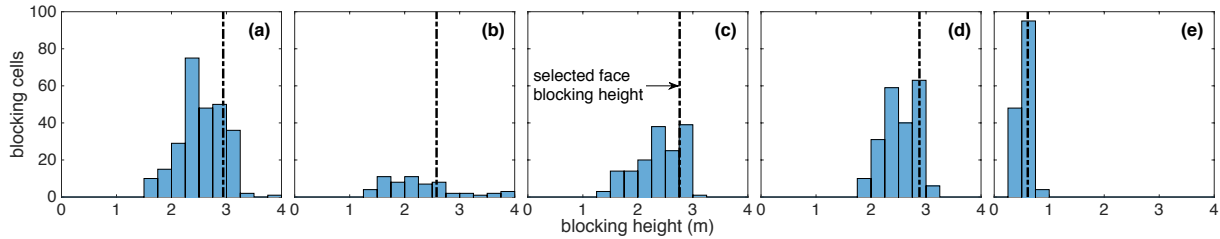


Figure 6. Histograms of the number of fine-grid blocking cells at different heights for five randomly selected coarse-grid cells in the data set. Dashed line represents the blocking height for the coarse-grid face selected as the 75th-percentile blocking height. Note that the maximum number of fine-grid cells in a single coarse-grid cell is 400.

end points are along a row face and cell-center blocking for a row face is indicated (blue \triangle), the blocking is necessarily along the row face connecting the blocked end points. However, where two blocked end points are co-linear along a column face and the cell-centered blocking is indicated for a row face (or vice versa), the choice of which face to block may be taken arbitrarily. Similarly, when two blocked end points are diagonally opposed, the selection of the blocking face is arbitrary. Note that these arbitrary choices will necessarily set up a condition where three end points are blocked. Because rows and columns are processed sequentially, diagonal blocking of two end points solved using columns (first cycle) sets up a cell with three blocked end points for solving using rows (second cycle). If three end points are blocked, then the cross-cell blocking must connect the two blocked end points that are co-linear along the column or row face (as appropriate), which ensures continuity of the feature.

In the present data set, all the objects achieved contiguous edge-blocking representations using the snap-grid and cross-cell object blocking algorithms outlined above. However, one can imagine a feature that is longer and narrower than shown in Fig. 5 for which the procedure might fail. For a long narrow feature, it is possible that multiple iterations of the cross-cell algorithm would be required to define a blocking condition. That is, the cross-cell algorithm described above combines (for example) the G_x blocking cells of two opposite column faces into a single cell-centered H_x that is tested for blocking. If $H_x < R_\Delta - \delta$, there is no blocking, and one can loop the algorithm to look for larger-scale blocking by combining the H_x of two adjacent cells into an I_x that is evaluated on the cell face for blocking. This iterative approach can be continued for J_x , K_x , etc., until blocking is achieved or the coarse-grid length of the object is reached.

Object blocking height – the snap-grid and cross-cell blocking methods determine which faces are blocked by an object. Unfortunately, establishing an exact fine-grid blocking height for a face requires analysis of the lowest contiguous path through the blocking cells. Although such an algorithm is theoretically possible, it has not been attempted. Given the uncertainties at the fine-grid level within typical lidar data, there are questions as to whether such a compli-

cated analysis would be worthwhile; thus, a simple statistical approach is adopted herein.

A blocking face is a single height value that represents a distribution of the fine-grid cell heights, which are retained in the G_{xx} , G_{yy} , H_{xx} , and H_{yy} sets described above. Typical distributions are shown in Fig. 6. A reasonable estimation of the blocking height for a G_{xx} set must be bounded by the maximum and minimum values in the set. It is convenient to denote the number of cells in the G_{xx} as N_{xx} , with similar definitions for other cell sets. We then define the relative “thickness” of the blocking cell set for the xx face as

$$T_{xx} \equiv \frac{N_{xx}}{R_\Delta}. \quad (17)$$

This thickness can be thought of as the number of rows of fine-grid cells that would be blocked if the fine grid cells were rearranged against the coarse-grid face. It can be argued that larger a T_{xx} should lead to higher blocking heights, i.e., making it less likely that a low-level path is available through the object. In the present work, we take an ad hoc approach: if $T_{xx} < 2$, we use the median of G_{xx} as the blocking height. Where $T_{xx} \geq 2$, we take the median of the largest half of the data set, i.e., the height of the 75th percentile. Similar arguments are used for, e.g., G_{yy} .

3 Results

Application to complex topography – applying these edge-blocking methods to the lidar data of Fig. 1b results in coarse-grid elevations and edge blocking as shown in Fig. 7. Note that many of the blocked faces are relatively close to the heights of adjacent cells and so are not obvious in Fig. 7b. The method is clearly successful in capturing the blocking of the railroad dike across the landscape, which was the primary motivation for this work.

However, it should be noted that the narrow section of the railroad dike across the open-water sections in Fig. 1a and b is actually a bridge, which is not hydraulically blocking and should be removed from the coarse data. Removing the bridge from the lidar data at the $1\text{ m} \times 1\text{ m}$ scale before edge blocking is applied is perhaps possible, but it is difficult

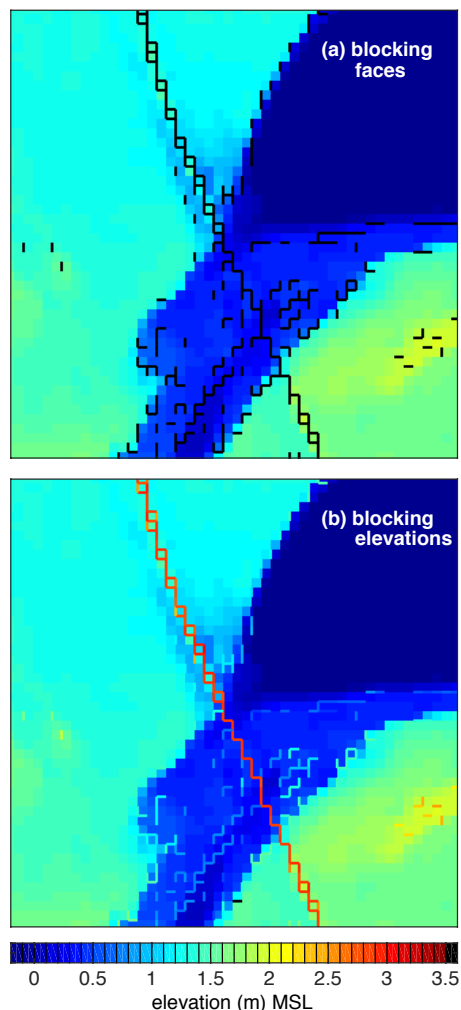


Figure 7. The snap-grid and cell-center blocking methods applied to the median-filtered background topography from Fig. 2. In (a) the locations of all blocked faces are shown; in (b), the blocked faces are given colors corresponding to their blocking height.

to automate as one must (i) identify bridge pixels and how they are different from dike pixels and (ii) decide what fill values to use for the bridge pixels. An advantage of the edge-blocking approach is that it simplifies removing the bridge from the coarse-grid data set, although we have not yet developed an automated approach for this task. Once the edge-blocking data set is defined, as in Fig. 7b, the start and the end of the bridge section (visually apparent from the change in the width of the dike in Fig. 1b) can be used to identify the edge-blocking cells in between. These data are easily removed from the edge cell set – i.e., their values are simply set to “not a number”. That is, where edge blocking is not identifiable, there is no need to store any edge-blocking heights. No fill values for bridge pixels are necessary because the coarse-grid cell elevations on either side of the bridge edge feature have already been determined by the median filter. In theory,

this technique might have wider applicability in removing the bridge from the original $1\text{ m} \times 1\text{ m}$ lidar. It might be possible to trace back the relationship between the bridge edges removed, the pixels that represented this blocking (i.e., G_{xx} , G_{yy} , H_{xx} , and H_{yy}), and the difference data set of Fig. 2b so the corresponding bridge pixels in the $1\text{ m} \times 1\text{ m}$ lidar Fig. 1b can be replaced by their median-filtered values from Fig. 1d.

Parameter sensitivity – there are three user-defined parameters in this method: the median grid filter (ΔM) whose effect is shown in Fig. 1d–f; the object cutoff height (Δh) that determines discrete objects as in Fig. 3a; and the δ of Eq. (4) that has a minor effect on small object removal. The δ has no impact on the larger features, and analyses (not shown) indicate that the set of blocking faces is relatively insensitive to the choice of $\delta \in \{0, 1, 2\}$ for the present work. The set of blocking faces is also relatively insensitive to the choice of ΔM , as shown in Fig. 8. A slight trend of increasing size and number of blocking features can be seen as the size of the median filter is increased. The choice of $\Delta M = 2\Delta C$, i.e., the 40 m filter in the present work, seems to be reasonable. Unsurprisingly, the third parameter, Δh , has a significant impact on both the number and size and blocking features, with higher values of Δh resulting in fewer and smaller blocking features (Fig. 9). Naturally, these results are highly dependent on the particular topography being studied. The selected value for Δh depends on the particular goal of the analysis. Indeed, identification of the 3 m elevation Nueces Delta railroad dike could be easily accomplished using $\Delta h = 1\text{ m}$ (not shown).

Computational considerations – although the median filter for a large data set can be computationally expensive (tens of hours on a non-GPU desktop computer for 10^8 grid cells of a 10^4 ha data set at $1\text{ m} \times 1\text{ m}$ resolution), the edge-blocking algorithm is otherwise relatively simple and inexpensive. Raw object identification from the binary image (Fig. 3a) provides 3754 individual objects from the 7.5×10^5 pixels; however 2075 of these are single-pixel objects that are easily eliminated. Further removing objects where $N_c \leq R_\Delta - \delta$ leaves only 129 objects for processing (Fig. 3b). During the processing steps, only 56 objects were found to be large enough to block any cell faces, and 21 of these blocked only a single cell face. The importance of such single-face blocking should be considered with analysis in hydrodynamic simulations. It may be that such blocking is better represented by roughness coefficients. For insight on how these results might translate to a larger scale, analysis of the 10^4 ha of the full Nueces River delta (not shown) resulted in 54 889 objects requiring processing, with 1880 identified as blocking multiple cell faces and 16 476 single-face blocking features. This processing required several hours on a non-GPU desktop workstation.

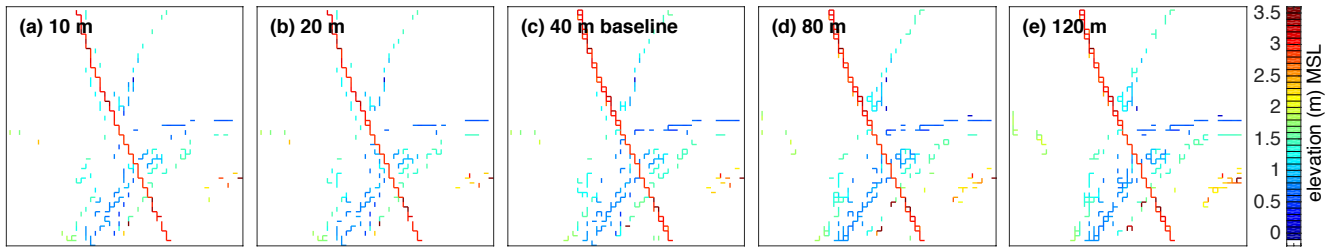


Figure 8. Sensitivity of edge blocking to choice of median filter scale, ΔM .

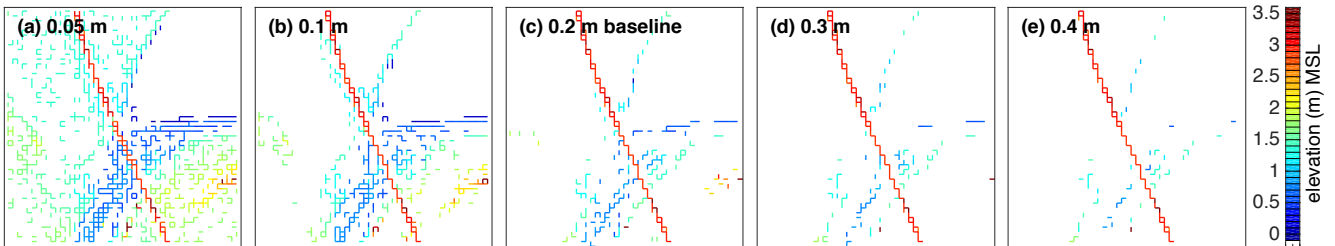


Figure 9. Sensitivity of edge blocking to choice of object height cutoff Δh .

4 Discussion

The present work is a first attempt at automating the identification of linear features for hydrodynamic blocking; as such there remain a number of areas where further analysis and improvement are needed. One of the less-satisfying aspects in the present work is the ad hoc selection of the 75th-percentile height as the blocking height for a face. There are a wide variety of statistical approaches that could be used for estimating the blocking height, and the present approach merely tries to account for the greater likelihood of higher blocking from a denser set of blocking fine-grid cells. The effect of this estimation method remains an open question, as analysis would be best accomplished in conjunction with hydrodynamic modeling.

As an exercise in practical application to a larger system, the edge-blocking methods have been used for the full 10^4 ha of the Nueces River delta, with an extract shown in Fig. 10 for 1100 ha of marshes that are southeast of the railroad bridge of Fig. 1. The blockages were computed using the automated techniques with manual removal of blocking faces for known bridges and culverts. This figure illustrates a key future challenge: identifying preferential flow paths that are unresolved at the coarse-grid scale (i.e., the *negative objects* resulting from unresolved depressions). Within the lidar data are several important narrow channels that are not in the $20\text{ m} \times 20\text{ m}$ grid. This effect can be seen in greater detail in Fig. 1a, where we can clearly see a narrow stream channel on one side of the railroad dike. This channel is entirely absent in the final topography of Fig. 7. Such channels could be readily identified by using the snap-grid and cell-center blocking techniques as “path” techniques for negative

objects (i.e., objects determined similar to Fig. 3a, but using a negative Δh for discrimination). However, it is not clear how such objects could be used in most hydrodynamic models. Definition of preferential narrow flow paths that are unresolved within coarse-grid topography remains an area with no clear solution (D’Alpaos and Defina, 2007). However, there are interesting possibilities in 1-D–2-D models, such as that of Viero et al. (2013), which might provide a good starting point. In any case, inversion of the techniques developed above provides a basis for defining preferential flow paths along coarse-grid cell edges, which can be seen as a precursor for new hydrodynamic modeling techniques.

A potential area where the edge-blocking method might be expanded is in the estimation of topographic roughness, which has been a subject of extensive prior research (e.g., Abu-Aly et al., 2014; Casas et al., 2010; Dorn et al., 2014; Forzieri et al., 2011; Straatsma and Baptist, 2008). By defining edge features, a portion of the difference between the grid cell elevation and subgrid features can be removed from the roughness estimation; i.e., we could use, for example, G_{xx} , G_{yy} , H_{xx} , and H_{yy} to remove pixels that have been resolved in to edge features and only consider the remaining pixels in a coarse-grid cell as contributing to roughness.

The methods developed above presume the lidar DTM is a reasonable representation of the underlying topography. Naturally, any DTM has noise, elevation inaccuracies, and the possible inclusion of permeable features (e.g., vegetation) as impermeable landscape. For noise and permeable features smaller than the coarse-grid scale, the present approach maintains the typical advantages of a median filter in removing smaller objects while retaining sharp edges. However, in larger forested areas or farmlands with thick hedgerows, sim-

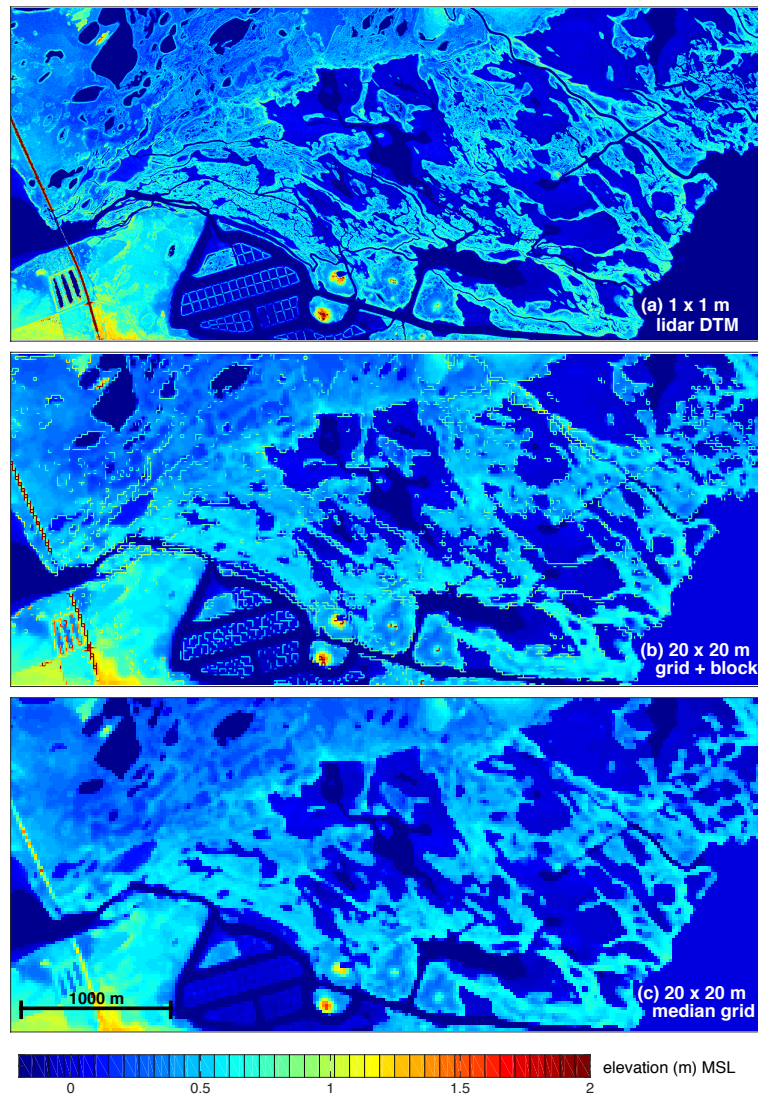


Figure 10. A section of the Nueces River delta ($2.2 \text{ km} \times 5 \text{ km}$ of a larger data set) centered at $27^{\circ}52'25'' \text{ N } 97^{\circ}32'21'' \text{ W}$. Blocking faces at bridges and culverts have been manually removed. For clarity in (a), where data are missing from the original lidar DTM (principally at water surfaces), the data have been replaced by -0.2 m . For the $20 \text{ m} \times 20 \text{ m}$ grid, these data have been replaced by survey data.

ple automated application of the present techniques is likely to create undesirable blocking features. There is an open issue as to whether the signature of such features could be a priori identified and used to create permeable cell faces with increased roughness in a hydrodynamic model.

Finally, as pointed out by P. Bates (personal communication, 2015), we lack a quantitative metric for evaluating how well the blocking faces model the underlying topography. Unfortunately, there is no obvious approach for comparing the results (Fig. 7b) to the original data (Fig. 1b) without resorting to image-processing techniques, which would simply provide a circular argument. It seems likely that a quantitative metric will require using a hydrodynamic model at both the original DTM and with the new coarse-grid topography to evaluate blockage effectiveness at different water elevations.

5 Conclusions

This paper provides automated identification of hydrodynamic blocking features in fine-scale rasterized lidar topography, along with upscaling of the blockage to a coarser raster grid. These techniques could be used for modeling coastal flood inundation at the practical coarse-grid scales necessary for addressing large-scale adaptive management questions, while retaining the blocking effects of fine-scale features that cannot otherwise be captured.

A coarse grid developed with the new edge-blocking technique could be immediately applied in any number of 2-D and 3-D hydrodynamic models that permit grid cells to use different topographic elevations on different flow faces. Note that, because raster topographic data sets with separate face

elevations have not been generally available, many hydrodynamic models do not provide for separate cell-face elevation data. Nevertheless, some models could be readily adapted to using such data with minor modifications and new tools for input data manipulation.

Acknowledgements. This work was originally developed during US National Science Foundation grant no. 0710901 and has continued under NSF grant no. CCF-1331610. Support of the Texas Water Development Board, the Coastal Bend Bays and Estuaries Program, the City of Corpus Christi, and the US Army Corps of Engineers was instrumental in the development of this project. The lidar data for the Nueces River delta were processed to the 1×1 m scale by J. Gibeaut, Texas A & M University – Corpus Christi.

Edited by: J. Brown

Reviewed by: M. Horritt and another anonymous referee

References

- Abu-Aly, T. R., Pasternack, G. B., Wyrick, J. R., Barker, R., Massa, D., and Johnson, T.: Effects of LiDAR-derived, spatially distributed vegetation roughness on two-dimensional hydraulics in a gravel-cobble river at flows of 0.2 to 20 times bankfull, *Geomorphology*, 206, 468–482, doi:10.1016/j.geomorph.2013.10.017, 2014.
- Bates, P.: Remote sensing and flood inundation modelling, *Hydrol. Process.*, 18, 2593–2597, doi:10.1002/hyp.5649, 2004.
- Bates, P., Marks, K., and Horritt, M.: Optimal use of high-resolution topographic data in flood inundation models, *Hydrol. Process.*, 17, 537–557, doi:10.1002/hyp.1113, 2003.
- Bhuiyan, M. J. A. N. and Dutta, D.: Assessing impacts of sea level rise on river salinity in the Gorai river network, Bangladesh, *Estuar. Coast. Shelf Sci.*, 96, 219–227, doi:10.1016/j.ecss.2011.11.005, 2012.
- Casas, A., Lane, S. N., Yu, D., and Benito, G.: A method for parameterising roughness and topographic sub-grid scale effects in hydraulic modelling from LiDAR data, *Hydrol. Earth Syst. Sci.*, 14, 1567–1579, doi:10.5194/hess-14-1567-2010, 2010.
- Casulli, V. and Cheng, R. T.: Semiimplicit finite-difference methods for 3-dimensional shallow-water flow, *Int. J. Numer. Meth. Fluids*, 15, 629–648, 1992.
- Cobby, D., Mason, D., Horritt, M., and Bates, P.: Two-dimensional hydraulic flood modelling using a finite-element mesh decomposed according to vegetation and topographic features derived from airborne scanning laser altimetry, *Hydrol. Process.*, 17, 1979–2000, doi:10.1002/hyp.1201, 2003.
- D’Alpaos, L. and Defina, A.: Mathematical modeling of tidal hydrodynamics in shallow lagoons: A review of open issues and applications to the Venice lagoon, *Comput. Geosci.*, 33, 476–496, doi:10.1016/j.cageo.2006.07.009, 2007.
- Dorn, H., Vetter, M., and Hoefle, B.: GIS-Based Roughness Derivation for Flood Simulations: A Comparison of Orthophotos, LiDAR and Crowdsourced Geodata, *Remote Sensing*, 6, 1739–1759, doi:10.3390/rs6021739, 2014.
- Fewtrell, T. J., Bates, P. D., Horritt, M., and Hunter, N. M.: Evaluating the effect of scale in flood inundation modelling in urban environments, *Hydrol. Process.*, 22, 5107–5118, doi:10.1002/hyp.7148, 2008.
- Forzieri, G., Degetto, M., Righetti, M., Castelli, F., and Preti, F.: Satellite multispectral data for improved floodplain roughness modelling, *J. Hydrol.*, 407, 41–57, doi:10.1016/j.jhydrol.2011.07.009, 2011.
- Gallien, T. W., Barnard, P. L., van Ormondt, M., Foxgrover, A. C., and Sanders, B. F.: A Parcel-Scale Coastal Flood Forecasting Prototype for a Southern California Urbanized Embayment, *J. Coast. Res.*, 29, 642–656, doi:10.2112/JCOASTRES-D-12-00114.1, 2013.
- Ge, J., Chen, C., Qi, J., Ding, P., and Beardsley, R. C.: A dike-groyne algorithm in a terrain-following coordinate ocean model (FVCOM): Development, validation and application, *Ocean Modell.*, 47, 26–40, 2012.
- Gonzalez, R. C., Woods, R. E., and Eddins, S. L.: *Digital Image Processing Using MATLAB*, Pearson, Prentice Hall, 2004.
- Mandlburger, G., Hauer, C., Hoefle, B., Habersack, H., and Pfeifer, N.: Optimisation of LiDAR derived terrain models for river flow modelling, *Hydrol. Earth Syst. Sci.*, 13, 1453–1466, 2009, <http://www.hydrol-earth-syst-sci.net/13/1453/2009/>.
- Nardin, W. and Edmonds, D. A.: Optimum vegetation height and density for inorganic sedimentation in deltaic marshes, *Nat. Geosci.*, 7, 722–726, doi:10.1038/NGEO2233, 2014.
- Purvis, M. J., Bates, P. D., and Hayes, C. M.: A probabilistic methodology to estimate future coastal flood risk due to sea level rise, *Coast. Eng.*, 55, 1062–1073, doi:10.1016/j.coastaleng.2008.04.008, 2008.
- Ryan, A. and Hodges, B. R.: Modeling Hydrodynamic Fluxes in the Nueces River Delta, Tech. Rep. CRWR Online Report 11-07, Center for Research in Water Resources, University of Texas at Austin, <http://www.cwrw.utexas.edu/online.shtml> (last access: May 2015), 2011.
- Sampson, C. C., Fewtrell, T. J., Duncan, A., Shaad, K., Horritt, M. S., and Bates, P. D.: Use of terrestrial laser scanning data to drive decimetric resolution urban inundation models, *Adv. Water Resour.*, 41, 1–17, doi:10.1016/j.advwatres.2012.02.010, 2012.
- Sanders, B. F., Schubert, J. E., and Gallegos, H. A.: Integral formulation of shallow-water equations with anisotropic porosity for urban flood modeling, *J. Hydrol.*, 362, 19–38, doi:10.1016/j.jhydrol.2008.08.009, 2008.
- Sanders, B. F., Schubert, J. E., and Detwiler, R. L.: ParBreZo: A parallel, unstructured grid, Godunov-type, shallow-water code for high-resolution flood inundation modeling at the regional scale, *Adv. Water Resour.*, 33, 1456–1467, doi:10.1016/j.advwatres.2010.07.007, 2010.
- Schubert, J. E., Sanders, B. F., Smith, M. J., and Wright, N. G.: Unstructured mesh generation and landcover-based resistance for hydrodynamic modeling of urban flooding, *Adv. Water Resour.*, 31, 1603–1621, doi:10.1016/j.advwatres.2008.07.012, 2008.
- Stelling, G. S.: Quadtree flood simulations with sub-grid digital elevations models, *Water Manage.*, 165, 567–580, 2012.
- Straatsma, M. W. and Baptist, M. J.: Floodplain roughness parameterization using airborne laser scanning and spectral remote sensing, *Remote Sens. Environ.*, 112, 1062–1080, doi:10.1016/j.rse.2007.07.012, 2008.
- Tsubaki, R. and Fujita, I.: Unstructured grid generation using LiDAR data for urban flood inundation modelling, *Hydrol. Process.*, 24, 1404–1420, doi:10.1002/hyp.7608, 2010.

- Vastila, K., Kumm, M., Sangmanee, C., and Chinvarno, S.: Modelling climate change impacts on the flood pulse in the Lower Mekong floodplains, *J. Water Clim. Change*, 1, 67–86, doi:10.2166/wcc.2010.008, 2010.
- Viero, D. P., D'Alpaos, A., Carniello, L., and Defina, A.: Mathematical modeling of flooding due to river bank failure, *Adv. Water Resour.*, 59, 82–94, doi:10.1016/j.advwatres.2013.05.011, 2013.
- Volp, N. D., van Prooijen, B. C., and Stelling, G. S.: A finite volume approach for shallow water flow accounting for high-resolution bathymetry and roughness data, *Water Resour. Res.*, 49, 4126–4135, 2013.
- Webster, T., McGuigan, K., Collins, K., and MacDonald, C.: Integrated River and Coastal Hydrodynamic Flood Risk Mapping of the LaHave River Estuary and Town of Bridgewater, Nova Scotia, Canada, *Water*, 6, 517–546, doi:10.3390/w6030517, 2014.
- Wen, L., Macdonald, R., Morrison, T., Hameed, T., Saintilan, N., and Ling, J.: From hydrodynamic to hydrological modelling: Investigating long-term hydrological regimes of key wetlands in the Macquarie Marshes, a semi-arid lowland floodplain in Australia, *J. Hydrol.*, 500, 45–61, doi:10.1016/j.jhydrol.2013.07.015, 2013.
- White, L. and Hodges, B. R.: Filtering the signature of submerged large woody debris from bathymetry data, *J. Hydrol.*, 309, 53–65, 2005.