Arun River Flood Analysis

CEGEG034: Mapping Science

Word Count :

Method: 473 (method),

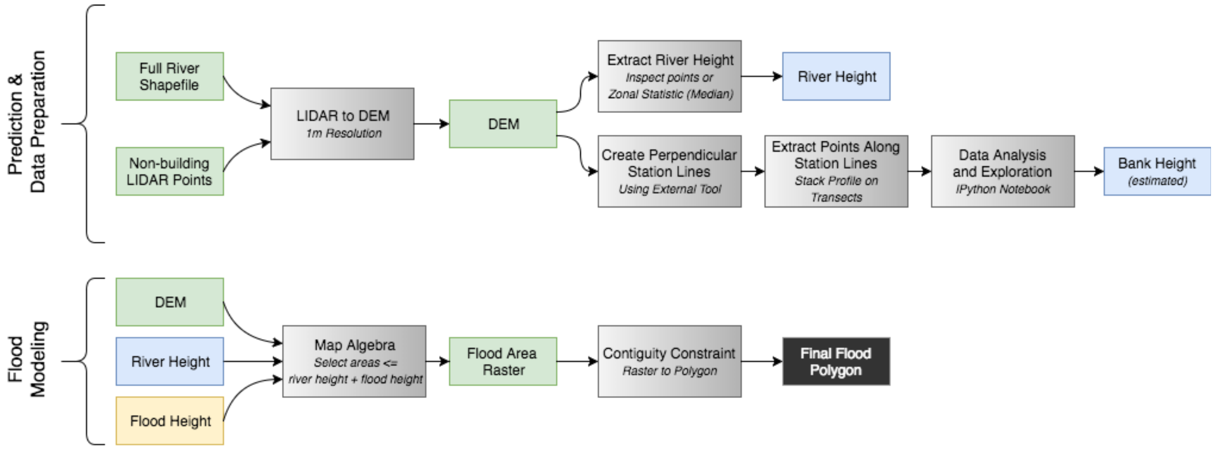
Body, total excluding maps: 1422

Table of Contents

Method

As shown in Figure 1 below, the analysis for this project is broken into two primary phases: prediction/data preparation and flood modeling.

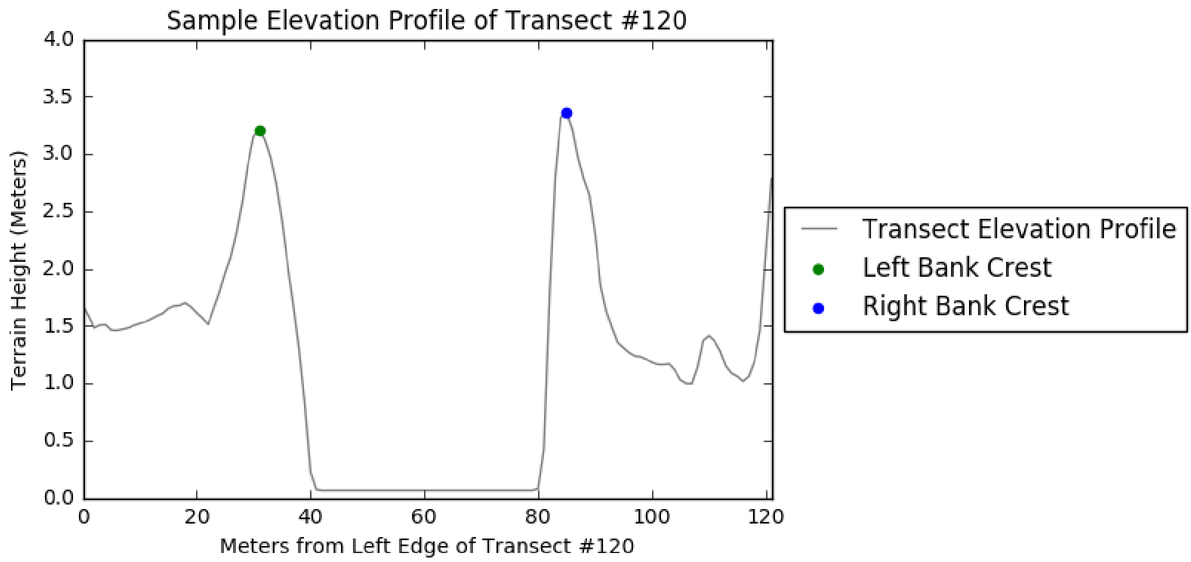
Figure 1: Method Flowchart



In the prediction and data preparation phase, a shapefile is generated for the full river extent based on MasterMap data augmented to cover the A27 bridge. This shapefile is used as a "lake" feature in generating a DEM from filtered non-building LIDAR points, which forces all parts of the river to have the same height. The resultant DEM is produced at 1m resolution, consistent with the LIDAR point spacing. River height is extracted by calculating zonal statistics. For more information on this process see Model Builder Appendix, Model 1: DEM Generation and Height Extraction.

In addition to river height estimation, this methodology also includes bank height estimation. Using a tool developed by Vini Indriasari (2015), this method creates transect lines perpendicular to the centerline of the river polygon at 1m intervals (see Map 1). Heights along these lines are extracted using the Stack Profile tool and exported for further analysis in Python. Bank heights are estimated by determining the peak or crest located on either side of the river (see Figure 2 below and Code Appendix for more details).

Figure 2: Sample Elevation Profile of a Transect with Left and Right bank crests identified.



These crests are then extracted for every transect to approximate the profile of the left and right bank along the length of the study area. The river might be expected to flood at the minimum bank height value, as the water will no longer be contained by the bank at that particular location. However, this method returned a few values as low as 0.16m, which based on visual data inspection seemed erroneous. To account for the ambiguity in this method, the dataset was culled of outliers using the method proposed by Iglewicz and Hoaglin (1993) with a threshold of 3.5. For results of this method, including a visualization of the raw data vs the data with outliers removed, see Figure 3.

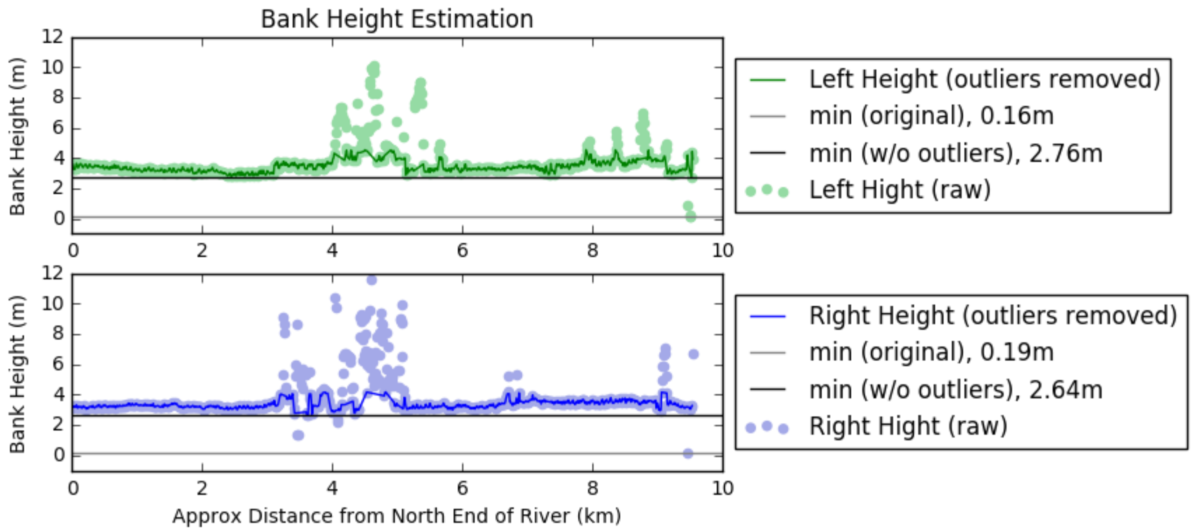
In the second phase of this project, the DEM and estimated river height are used to simulate the impact for a given floodwater height scenario. Using map algebra, all areas of the DEM situated at or below the flooded river height are selected and marked as "potentially flooded". To enforce continuity with the river's path, this potential flood raster is converted to a polygon and all shapes overlapping or intersecting the river boundary are selected. These polygons comprise the final flood extent and are used to map the flood in both two and three dimensions. For more information, see Model Builder Appendix, Model 2: Flood Polygon Creation. Finally, buildings were classified as untouched, affected, impacted, or inundated based on spatial queries and the definitions presented in Figure 4.

Results

According to the method described above, the river lies at an average height of 0.06m.

Height estimates for the left and right banks range between 0.16m - 10.14m and 0.19m - 11.58m respectively as shown in Figure 3, with median values of 3.38m and 3.31m. Due to ambiguities in this method of bank height determination and a lack of supporting literature it is impossible to say directly from the bank height when the river might flood. Nevertheless, the minimum of the dataset with outliers removed (Iglewicz and Hoaglin 1993) provide a reasonable proxy. These data indicate that a flood height of approximately 2.69m and 2.58m would cause the river Arun to burst its left and right banks respectively. (See Certainty for a discussion about confidence in bank height and river height values.)

Figure 3: Left and right bank height estimates, including min values and no outlier datasets.



Analysis of flood extents in ArcGIS indicates that a flood height around 2.75m and 2.90m would cause the river Arun to burst its left and right banks respectively, resulting in significant flooding. The discrepancy between this value and the bank height derived from the Python analysis (a difference of 0.06m (2.1%) on the left and 0.32m (12.5%) on the right) underscores the need for further study, including spatial methods for determining outliers and rigorous bank definition.

Figure 4: Comparison of scenarios for various flood heights.

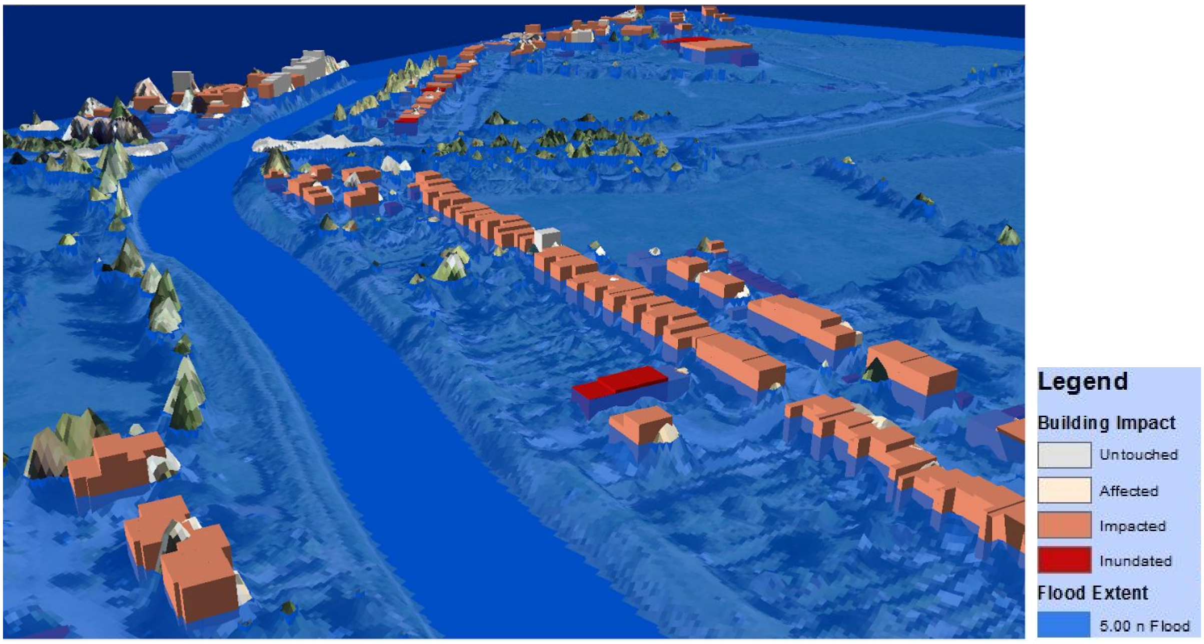
Flood *Height*	Flood *Extent*	Flood *Increase*	Buildings *Affected*	Buildings *Impacted*	Buildings *Inundated*
2.75m Map 2	60% 578883 m^2	1333%	31/657 (5%) 1247 m^2 (4%)	77/657 (12%) 5533 m^2 (19%)	8/657 (1%) 149 m^2 (0%)
2.90m Map 3	62% 599487 m^2	1384%	39/657 (6%) 1837 m^2 (6%)	95/657 (14%) 6353 m^2 (21%)	8/657 (1%) 149 m^2 (0%)
5.00m Map 4	83% 789998 m^2	1856%	14/657 (2%) 404 m^2 (1%)	293/657 (45%) 14858 m^2 (50%)	126/657 (19%) 4653 m^2 (16%)
Desc.	*Areal extent of the floodwater (vs study area minus normal river area)*	*Increase in river surface area between flood and non-flood*	*Buildings inside study area that intersect the flooding polygon only*	*Buildings that overlap the flooding polygon by 0.5m or more only (selection: within -0.5m)*	*Buildings completely within the flooding polygon only*

Maps of three flooding scenarios (2.75m, 2.90m, and 5.00m) are shown in 3D below and presented in the Map Appendix. These flood heights correspond to left bank breach, right bank breach, and an extreme flooding event (98% of buildings impacted).

Figure 5: 3D Visualization - 2.75m Flood

Figure 6: 3D Visualization - 2.90m Flood

Figure 7: 3D Visualization - 5.00m Flood



Discussion

Accuracy

Data used in this analysis comes from UK Ordnance Survey MasterMap and LIDAR. MasterMap data is accurate to 0.9m (OS Mastermap Topography Layer, nd). The accuracy of LIDAR measurements varies based on the equipment and flying height, but for a RIEGL LMS-Q680i air-based scanner is approximately 0.2m vertically (Fowler and Kadatisky, 2011). Accordingly, measurements and estimates in this analysis have been reported to the 1cm level, indicating one decimal place of uncertainty beyond the original data per scientific convention.

Certainty

There are a number of factors that influence confidence in the estimated river height (6cm) derived in this analysis. First, the LIDAR data used has intrinsic uncertainty that may result in errors as large as 20cm vertically for a given point (Fowler and Kadatisky, 2011). In addition to this intrinsic uncertainty, filtering methods for LIDAR data (e.g. removing points within a building) may result in noisy elevation surfaces, as discussed in the first Mapping Science coursework. Uncertainty in location of the MasterMap dataset may impact the quality of the resultant DEM, particularly as MasterMap building outlines were used to filter the LIDAR data for DTM generation. Given both the accuracy and uncertainty in the initial data sources, the estimated river height must be taken with caution.

In addition, there are a number of other factors impacting river height that cannot be derived from LIDAR alone. Most importantly, the river Arun is tidal and over the course of a week high tide may vary over a meter (Arundel Tidal Prediction, nd). Data recorded during the week of

November 25 at the Queen Street Bridge tide gauge, located at the north end of the study area, shows water levels as different as nearly 2m within 24 hours as well as a typical high/low range of nearly 3m (Arun at Arundel Queen St Bridge, nd). Any flood model for this area would need to account for the tidal nature of the river Arun.

In addition to uncertainty in river height, there is significant uncertainty in the bank height estimated using this method. First, a firm definition of bank height or a way of determining bank height using GIS was not clearly present within the literature. While the author has attempted to create a rigorous method for bank height determination, no independent verification has been conducted and this method cannot be assumed to be scientifically rigorous.

From a visual inspection of the derived bank heights and the DEM, it seemed that outliers were present that skewed the minimum bank height and thus the estimated flood height. However, any method to determine outliers runs the risk of excluding legitimate data points. This particular application requires identification of the lowest (reasonable) bank height values; any process that disregards any low bank height values must be viewed with suspicion. Despite these limitations, the method presented here provides a lower bound for the minimum bank height that sits comfortably below the flood height predicted using GIS.

References

Arun at Arundel Queen St Bridge. *River Levels*. Available at: https://www.riverlevels.uk/arun-arundel-arundel-queen-st-bridge#.Wh7yG7SFgWo [Accessed November 29, 2017].

Arundel Tidal Prediction. *UK Hydrographic Office*. Available at: http://www.ukho.gov.uk/easytide/easytide/ShowPrediction.aspx?PortID=0074B&PredictionLength=7 [Accessed November 29, 2017].

Fowler, A. and Kadatskiy, V., 2011, May. Accuracy and error assessment of terrestrial, mobile and airborne lidar. In *Proceedings of American Society of Photogrammetry and Remote Sensing Conference (ASPRP 2011), 1–5 May 2011, Milwaukee, Wisconsin*.

Iglewicz, B. & Hoaglin, D.C., 1993. *How to detect and handle outliers*, Milwaukee, WI: ASQC Quality Press.

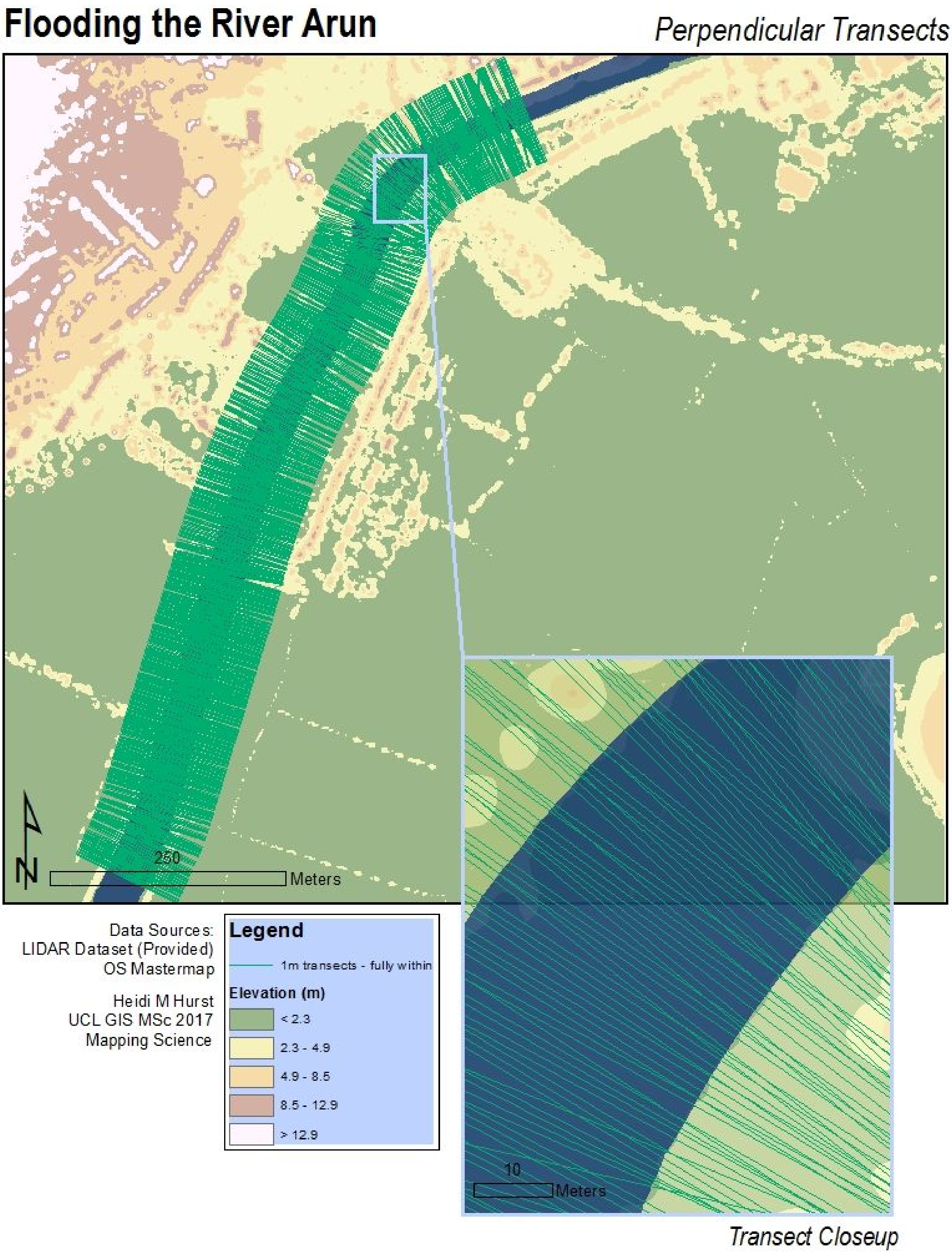
Indriasari, V., 2015. Linear Sampling Toolbox. *ArcGIS*. Available at: http://www.arcgis.com/home/item.html?id=d16ec9f1a16b4e0caca7b2398bff941b [Accessed November 29, 2017].

Kington, J., Pythonic way of detecting outliers in one dimensional observation data. *Stack Overflow*. Available at: https://stackoverflow.com/questions/22354094/pythonic-way-of-detecting-outliers-in-one-dimensional-observation-data [Accessed November 29, 2017].

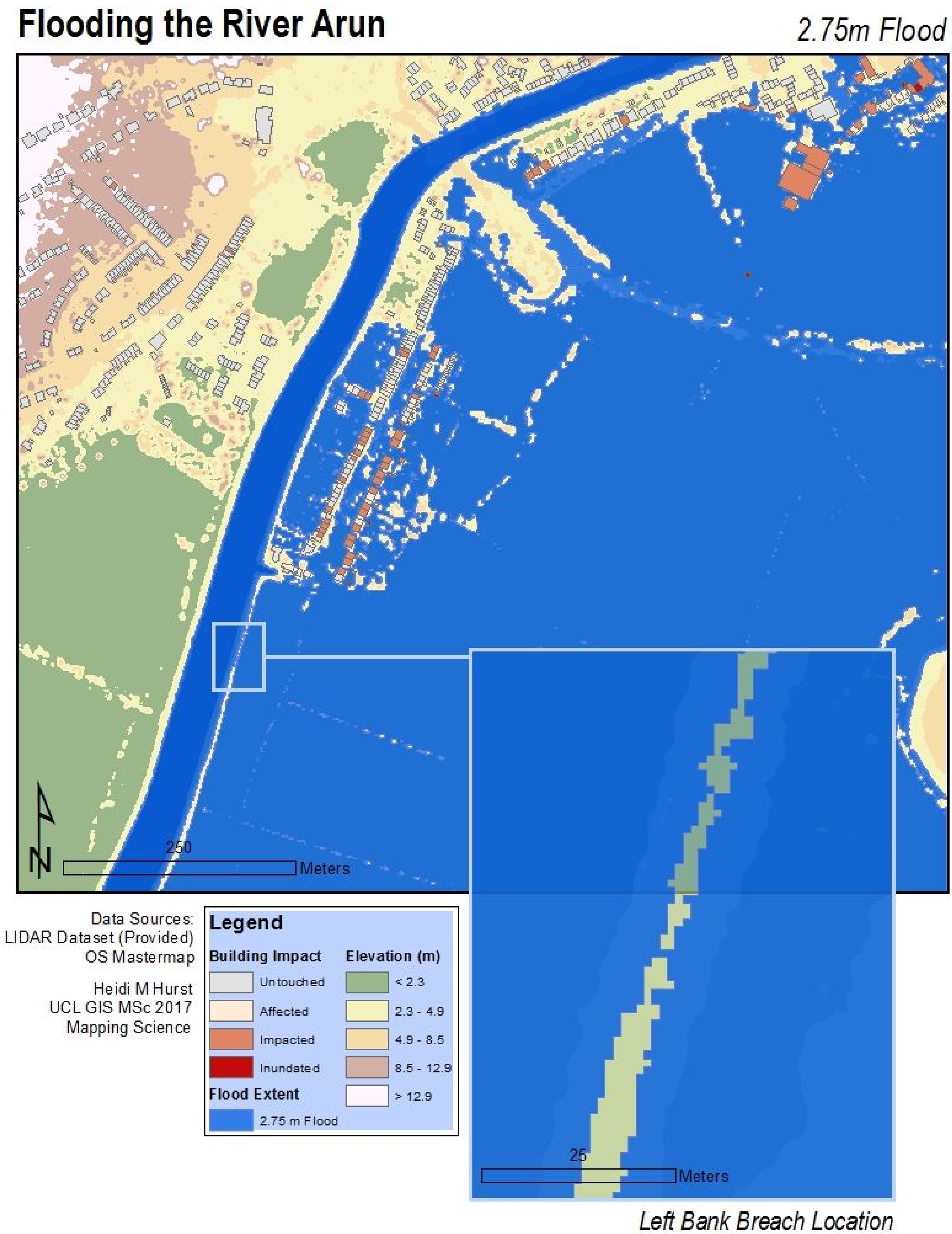
OS MasterMap Topography Layer. *Ordnance Survey*. Available at: https://www.ordnancesurvey.co.uk/business-and-government/products/topography-layer.html [Accessed November 29, 2017].

Map Appendix

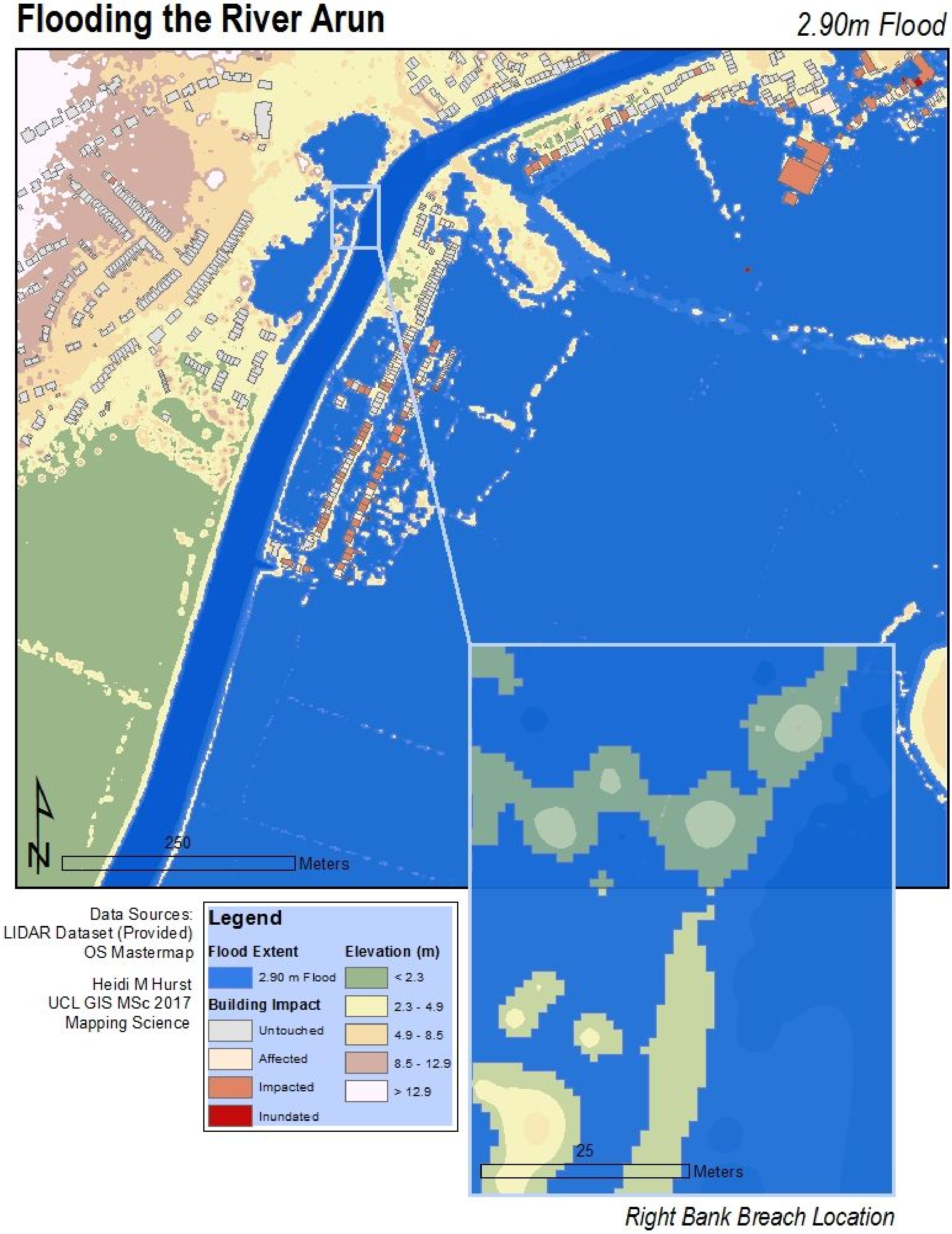
Map 1: Perpendicular Transects



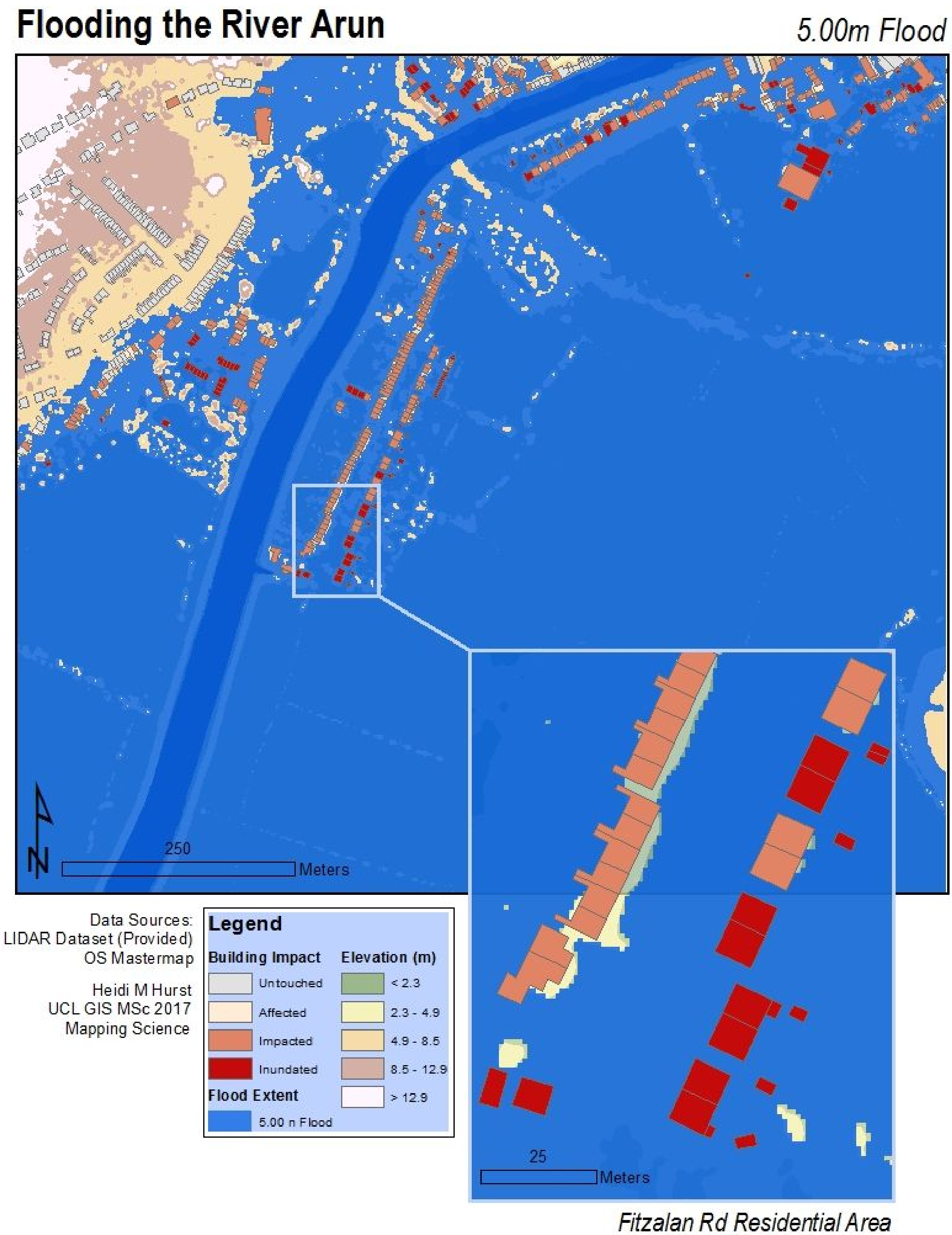
Map 2: Left Bank Breach - *Flooding the River Arun - 2.75m Flood*



Map 3: Right Bank Breach - *Flooding the River Arun - 2.90m Flood*

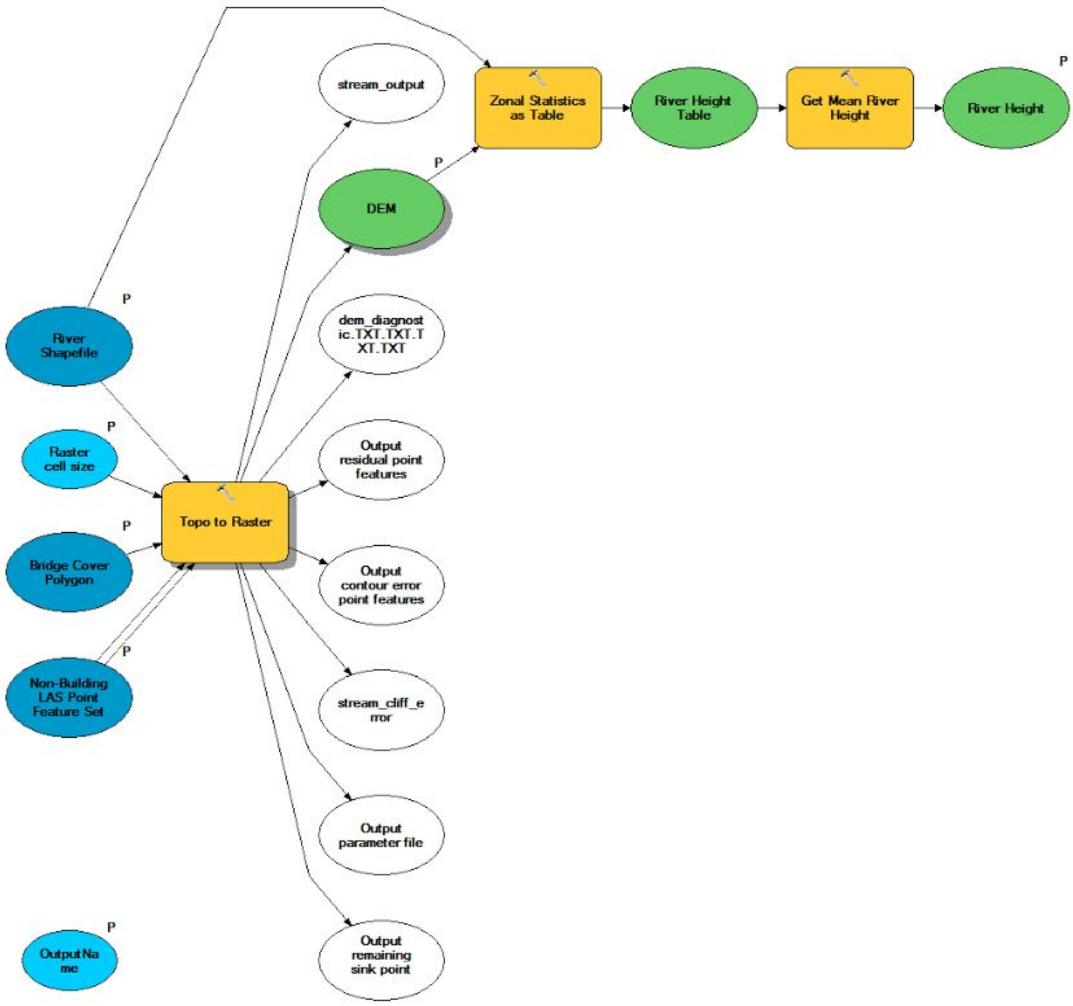


Map 4: Cataclysmic Flooding - *Flooding the River Arun - 5.00m Flood*

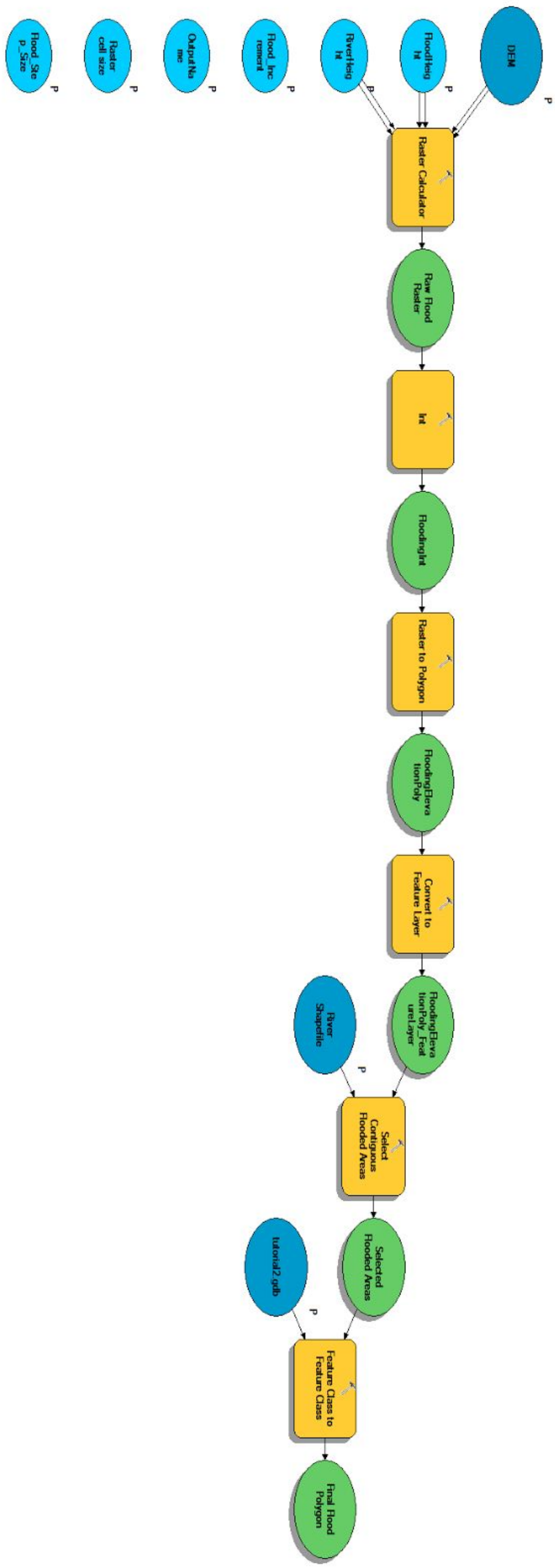


Model Builder Appendix

Model 1: DEM Generation and Height Extraction



Model 2: Flood Polygon Creation



Code Appendix

Available online at https://github.com/heidimhurst/ucl_msc/tree/master/mapping_science/tutorial2.

Imports and Function Definitions

In [362]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sys
import scipy
```

In [363]:

```
data = pd.read_csv("1mtransect_clipped_1mres.csv")
```

In [364]:

```
# river height, from DEM
river_height = 0.0640

# flood height estimates, from simulation
flood_left = 2.75
flood_right = 2.90
```

In [365]:

```
def extractBanks(line, river_height = river_height):
   values = line['FIRST_Z']

   # determine where the first value equal to the river height is located
   start = values[values - river_height < 0.00001].index[0] - values.index[0]
   max_index = values.index[-1] - values.index[0]

   #first bank
   if start < max_index:
      a = values.iat[start]
      b = a
      k = start
      # iteratively compare points to determine where the bank crests
      while(a <= b and k < (max_index - 1)):
         a = b
         b = values.iat[k]
         k += 1
      right = a

   #second bank
   if start > 0:
      a = values.iat[start]
      b = values.iat[start - 1]
      k = start
      while(a <= b and k > 0):
```

```
        k -= 1
        a = b
        b = values.iat[k]
    left = a

  return(left,right)
```

In [366]:

```
def is_outlier(points, thresh=3.5):
  """
  Returns a boolean array with True if points are outliers and False
  otherwise.

  Parameters:
  -----------
    points : An numobservations by numdimensions array of observations
    thresh : The modified z-score to use as a threshold. Observations with
        a modified z-score (based on the median absolute deviation) greater
        than this value will be classified as outliers.

  Returns:
  --------
    mask : A numobservations-length boolean array.

  References:
  ----------
    Boris Iglewicz and David Hoaglin (1993), "Volume 16: How to Detect and
    Handle Outliers", The ASQC Basic References in Quality Control:
    Statistical Techniques, Edward F. Mykytka, Ph.D., Editor.
  """
  if len(points.shape) == 1:
    points = points[:,None]
  median = np.median(points, axis=0)
  diff = np.sum((points - median)**2, axis=-1)
  diff = np.sqrt(diff)
  med_abs_deviation = np.median(diff)

  modified_z_score = 0.6745 * diff / med_abs_deviation

  return modified_z_score > thresh
  # from
https://stackoverflow.com/questions/22354094/pythonic-way-of-detecting-outliers-in-on
e-dimensional-observation-data
```

Data Processing

In this section, raw data on transect elevation profiles is used to derive left and bank heights. Outliers are determined perusant to Iglewicz and Hoaglin (1993).

In [367]:

```python
left = []
right = []
# iterate through all lines entirely within the study area
for i in range(28, 1104):
   line = data.loc[data['LINE_ID'] == i]
   # ensure that the line isn't empty
   if len(line) != 0:
       l, r = extractBanks(line, river_height = river_height)
       left.append(l)
       right.append(r)
```

In [368]:

```python
# create a version without outliers, with specific threshold
thresh = 3.5

# left bank
outlier_l = is_outlier(np.asarray(left), thresh = thresh)
left_no = []
left_no_index = []
for i in range(0,len(outlier_l)):
   if outlier_l[i] == False:
       left_no.append(left[i])
       left_no_index.append(i)

# right bank
outlier_r = is_outlier(np.asarray(right), thresh = thresh)
right_no = []
right_no_index = []
for i in range(0,len(outlier_r)):
   if outlier_r[i] == False:
       right_no.append(right[i])
       right_no_index.append(i)
```

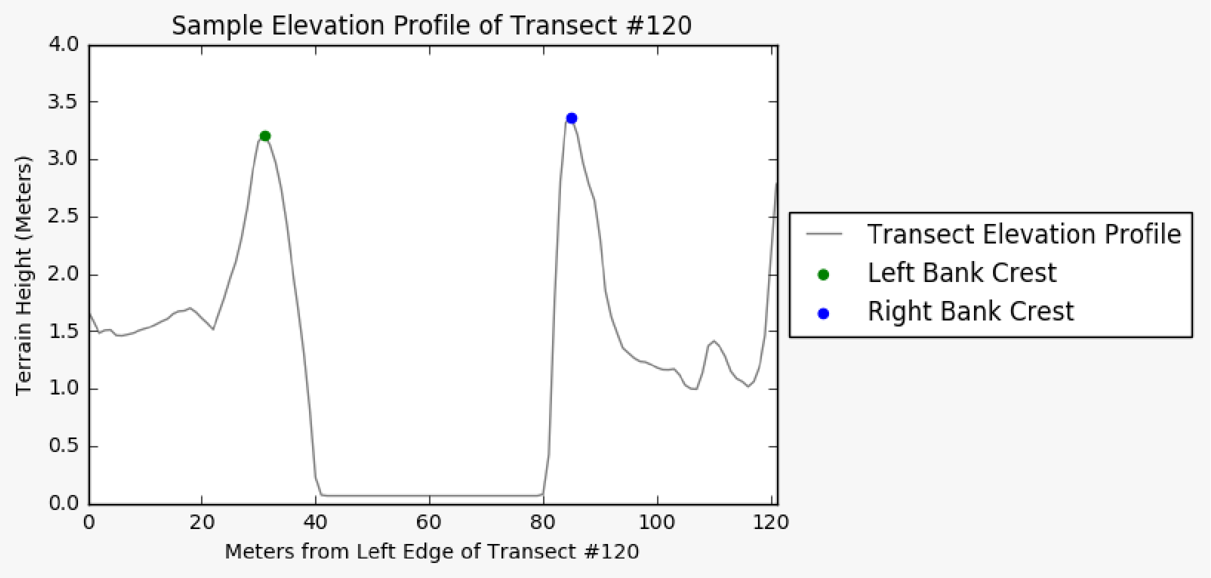
Transect Examination

To confirm that the method used was appropriate, we plot the elevation profile of one transect and the calculated left and right bank locations.

In [369]:

```python
# extraction of a single transect to confirm methodology
line_numb = 120
line = data.loc[data['LINE_ID'] == line_numb]
values = line['FIRST_Z']

# find locations, height of left and right banks
l, r = extractBanks(line)
li = values[values == l].index[0]
ri = values[values == r].index[0]
```

```
plt.plot(values, label = "Transect Elevation Profile", zorder = 1, color = "grey")
plt.scatter(x = li, y = l, label = "Left Bank Crest", color = "green", zorder = 2)
plt.scatter(x = ri, y = r, color = "blue", label = "Right Bank Crest", zorder = 2)

x = np.arange(values.index[0],values.index[-1],20)
xtics = np.arange(0,121,20)
plt.xticks(x,xtics)

plt.ylim([0,4])
plt.xlim([values.index[0], values.index[-1]])
plt.ylabel("Terrain Height (Meters)")
plt.xlabel("Meters from Left Edge of Transect #%i" %line_numb)
plt.title("Sample Elevation Profile of Transect #%i" %line_numb)
plt.legend(loc = 'center left', bbox_to_anchor=(1, 0.5), scatterpoints = 1)
plt.show()
```



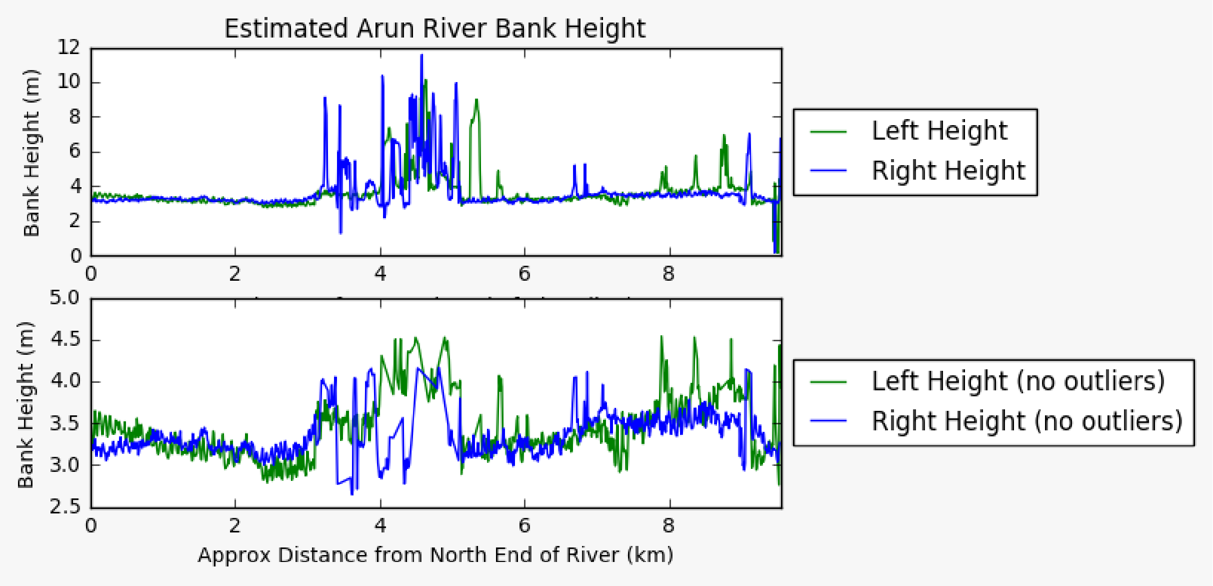
Visualising Raw Estimated Bank Height for River Arun

These charts allow us to see the bank height estimated using this method along the length of River Arun.

In [370]:

```
# adjust scales to kilometers
x = np.arange(0, 10001, 200)
xtics = np.arange(0, 11, 2)

plt.subplot(2,1,1)
plt.xticks(x,xtics)
plt.plot(left, label = "Left Height", color = "green")
plt.plot(right, label = "Right Height", color = "blue")
plt.legend(loc = 'center left', bbox_to_anchor=(1, 0.5))
plt.xlabel("Distance from North End of River (km)")
```

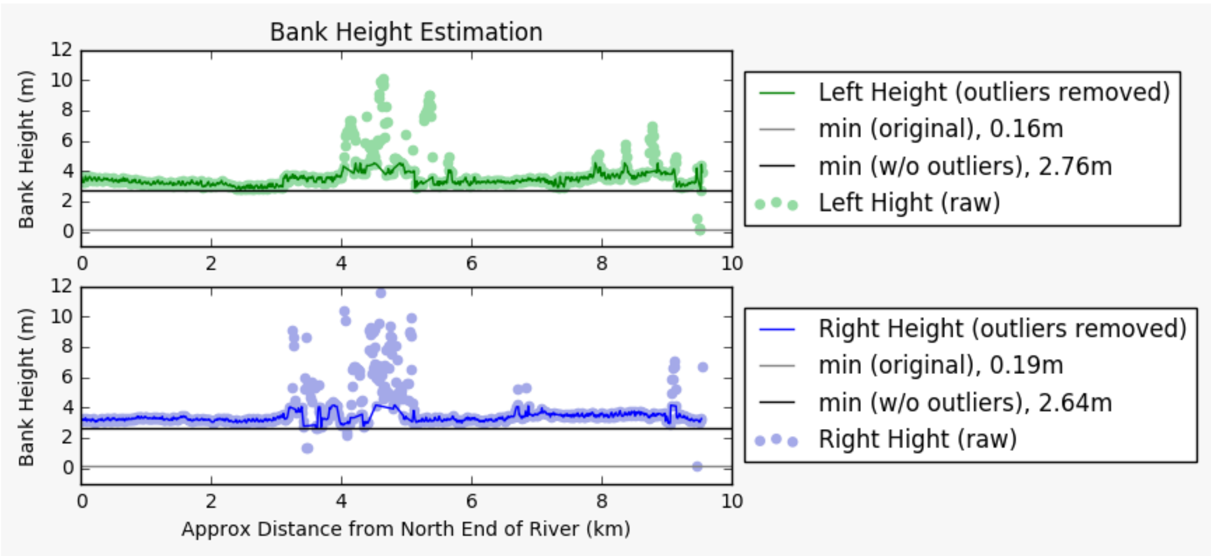
```
plt.ylabel("Bank Height (m)")
plt.title("Estimated Arun River Bank Height")

plt.subplot(2,1,2)
plt.xticks(x,xtics)
# plot without outliers
plt.plot(left_no_index, left_no, label = "Left Height (no outliers)", color = "green")
plt.plot(right_no_index, right_no, label = "Right Height (no outliers)", color = "blue")
plt.legend(loc = 'center left', bbox_to_anchor=(1, 0.5))
plt.xlabel("Approx Distance from North End of River (km)")
plt.ylabel("Bank Height (m)")
plt.show()
```



In [373]:

```
# plot outliers overlaid with minimums
x = np.arange(0, 10001, 200)
xtics = np.arange(0, 11, 2)
xindex = np.arange(0,len(left),1)

# left bank
plt.subplot(2,1,1)
plt.scatter(x = xindex, y = left, color = "#95dba4", label = "Left Hight (raw)")
plt.plot(left_no_index, left_no, label = "Left Height (outliers removed)", color = "green")
plt.axhline(np.min(left), label = "min (original), %1.2fm" %np.min(left), color = "gray")
plt.axhline(np.min(left_no), label = "min (w/o outliers), %1.2fm" %np.min(left_no),
color = "k")

plt.legend(loc = 'center left', bbox_to_anchor=(1, 0.5))
plt.ylabel("Bank Height (m)")
plt.title("Bank Height Estimation")
plt.xticks(x,xtics)
```

```
plt.xlim([0, 1000])
plt.ylim([-1, 12])

# right bank
plt.subplot(2,1,2)
plt.scatter(x = xindex, y = right, color = "#a4a9e8", label = "Right Hight (raw)")
plt.plot(right_no_index, right_no, label = "Right Height (outliers removed)", color = "blue")
plt.axhline(np.min(right), label = "min (original), %1.2fm" %np.min(right), color = "gray")
plt.axhline(np.min(right_no), label = "min (w/o outliers), %1.2fm" %np.min(right_no), color = "k")

plt.legend(loc = 'center left', bbox_to_anchor=(1, 0.5))
plt.ylabel("Bank Height (m)")
plt.xlabel("Approx Distance from North End of River (km)")
plt.xticks(x,xtics)
plt.xlim([0, 1000])
plt.ylim([-1, 12])
plt.show()
```



Comparison of Results

How close were the estimates of flood height we derived from bank height using Python to the estimates obtained fro simulating flooding in ArcPy?

In [375]:

```
# estimates for flooding
# left bank:
lflood = np.min(left) - river_height
lflood_no = np.min(left_no) - river_height
ldiff = np.abs(flood_left-lflood_no)
print("=== Left bank ===")
```

```python
print("Estimated raw flood height: %1.2fm" %lflood)
print("Estimated flood height (outliers removed): %1.2fm" %lflood_no)
print("Simulated flood height: %1.2fm" %flood_left)
print("Difference between estimations: %1.2fm, %2.1f%%" %(ldiff,
100*np.divide(ldiff,lflood_no)))


# right bank
rflood = np.min(right) - river_height
rflood_no = np.min(right_no) - river_height
rdiff = np.abs(flood_right-rflood_no)
print("\n=== Right bank ===")
print("Estimated raw flood height: %1.2fm" %rflood)
print("Estimated flood height (outliers removed): %1.2fm" %rflood_no)
print("Simulated flood height: %1.2fm" %flood_right)
print("Difference between estimations: %1.2fm, %2.1f%%" %(rdiff,
100*np.divide(rdiff,rflood_no)))
```

```
=== Left bank ===
Estimated raw flood height: 0.10m
Estimated flood height (outliers removed): 2.69m
Simulated flood height: 2.75m
Difference between estimations: 0.06m, 2.1%

=== Right bank ===
Estimated raw flood height: 0.12m
Estimated flood height (outliers removed): 2.58m
Simulated flood height: 2.90m
Difference between estimations: 0.32m, 12.5%
```