Application of a new method to improve river cross sections derived from satellite images

ELIN ANDERSSON

SOFIA HIETALA
Abstract

In hydrological and hydraulic modelling, river geometry is a crucial input data. Recent investigations have been looking at methods to improve the description of cross sections extracted by DEM derived by satellite images. SRTM derived DEM are often lacking precise information as the sensors cannot detect the submerged river parts, but, on the other hand, it is available on a global scale which makes it very attractive and useful, especially in data scarce regions. This study aims at applying the so called “slope break” method to improve river cross section geometry extracted from SRTM DEM. The report is divided into three parts: a) The making of a Matlab-code to improve cross sections geometry extracted by satellite derived DEM; b) an application of the code to real cross-sections from the river Po in Italy and c) hydraulic simulations with and without SRTM modified cross sections to test the performance of the method, in collaboration with senior colleagues. The Matlab successfully performs the slope break point and finds, when appropriate, the approximated lowest point $z_{min}$ of the cross section below the water surface. The comparison of the river geometry of the modified SRTM cross sections versus LiDAR available cross sections show the good performance of the method in improving the river geometry description. This code can simplify the work and improve many SRTM river cross sections in an effective way. The hydraulic simulations performed with and without the modified cross sections show how the modified SRTM model improves when compared to LiDAR results.

Keywords

Cross Section, Slope Break, Po River, Hydraulic modelling, Hydrological modelling, Matlab
Acknowledgments

We would like to thank our supervisor Luigia Brandimarte for her guidance and great inspiration throughout this difficult project, it would not have been possible without all our discussions and meetings. Also, we will thank Milos Dinic and Victor Alejandro Ortiz Peña for the collaboration during this work. We would also like to thank Michael Sederlin for the discussions regarding the Matlab code, the smart inputs made it able for us to proceed in our work on times we got stuck and those were many.
Table of contents
1. Introduction .......................................................................................................................... 1
   Background ............................................................................................................................. 1
   Purpose and aim ...................................................................................................................... 3
   Research activities .................................................................................................................. 3
2. Methods .................................................................................................................................. 4
   Computation of the height-width pairs .................................................................................. 4
   Determination of the Slope Break point ................................................................................. 5
   Interpolate to approximate the lowest part of the cross section ............................................ 5
   ArcGIS and Excel .................................................................................................................... 5
3. Results ...................................................................................................................................... 7
4. Discussions ............................................................................................................................. 12
5. Conclusions ............................................................................................................................ 12
6. References ............................................................................................................................... 13
Appendix A .................................................................................................................................. 14
Appendix B .................................................................................................................................. 21
Abbreviations
CS – Cross Section
DEM – Digital Elevation Models
h-w – height- width
SB – Slope Break
SRTM–Shuttle Radar Topography Mission
1. Introduction

Background
Recently, the interest for hydraulic and hydrological models has increased, carrying the need for improving model results (Domeneghetti, 2016). Mersel et al. (2013) explains the reason for the increased interest with the following: fresh water is a necessity for everyday life, still the knowledge of spatial and temporal dynamics of rivers in remote areas is poor. Also, the rising stress on the rivers worldwide as a cause of growing population, industrialisation and climate change has contributed to the increasing interest.

In hydrological and hydraulic modelling, river geometry is a crucial input data. In flood modelling, information of the river geometrical characteristics is essential for reproducing past events, calibrate the model and make predictions that are used for flood management purposes. Whenever possible, river geometry is described by means of in situ surveys from which river bathymetry can be depicted. Geometry such as cross sections, floodplain and longitudinal profile acts as input in hydraulic models. Nevertheless, detailed topography, either in situ or Light Detection and Ranging (LiDAR) derived, are not always affordable: both from an economic and practical point of view.

Nowadays, hydraulic and hydrological modelling are highly benefiting from the proliferation of satellite based information (e.g., water levels, topography and soil moisture) which is a great resource to relate to when there is a lack of data, especially in remote areas.

The influence of different resolutions of Digital Elevation Models (DEM) derived by satellite images in hydraulic modelling has been widely studied (e.g. Yan et. al., 2013; Ali et.al., 2015). The study made by Ali et. al. (2015) concluded that there are mainly two ways to collect topographical data. Firstly, remote techniques in form of satellite-based Digital Elevation Models (DEM) such as SRTM, and secondly, ground-based DEM such as Light detection and ranging (LiDAR). These two types of DEM are differing in price, remote techniques are low-cost and LiDAR are high-cost. In the study both types of DEM information was used as input in a modelling program, Hec-Ras. This showed that there is a big difference between the results when using SRTM or LiDAR while modelling.

Another study made by Yan et. al. (2013) did show that there is a significant difference between SRTM and LiDAR-based models, however that difference was not that extensive when including other affecting parameters and when looking at medium to large scale rivers. Still the difference is too affecting if looking at more precise modelling.

Topographic data generated from NASA’s Shuttle Radar Topography Mission (SRTM) is the most widely used satellite derived DEM. SRTM has a resolution of either 90 m or 30 m, while LiDAR is much more precise with a higher resolution, about 2m. Figure 1 shows the difference between these resolutions in form of a river stretch of the Lower Limpopo, in Mozambique. Firstly DEM derived SRTM 90 m is shown, secondly SRTM 30 m and lastly the LiDAR with much higher resolution.

Figure 1: Lower Limpopo, Mozambique—Comparison of DEM derived by a) SRTM 90m, b) SRTM 30m and c) LiDAR 2m
In figure 2 a comparison of data on a cross section at the Po river in Italy, evaluated with the two different measurements systems is visualized. Although LiDAR gives a clear picture of the river cross section, it is not available all around the globe, especially in remote areas. SRTM (90m) is freely available on a global scale; however the resolution is too poor to give a reliable picture of the river cross section.

![Figure 2: Comparison of SRTM and LiDAR data for a river cross section.](image)

Furthermore, DEM-information derived by satellite images are often lacking precise information as the sensors cannot detect the submerged river parts. Therefore, while DEM derived by sensors mounted on satellites or aircrafts are widely used, the description of the extrapolated river cross sections from these DEMs is still uncertain and rough when it comes to the portion of the section below the water level captured by the satellite or aircraft.

Mersel et al. (2013) and Domeneghetti (2016) have recently investigated methods to improve DEM derived by satellite images, for making models of river bathymetry. As Mersel et al. (2013) mentions, remote sensing technology is not able to observe geometry below the water surface of the river. Further, what they have studied is how this part of the river could be modelled only with the knowledge of the correlation between the river height and river width, i.e. with information provided for example by the SWOT (The Surface Water and Ocean Topography) program. SWOT’s mission is to map and understand Earth’s surface water with satellites and is developed by hydrologists and oceanographers from the U.S and France and also the space agencies in the U.S, United Kingdom, Canada and France. (SWOT.jpl.nasa.gov 2018-01-31).

The aim of the work made by these authors has been to try out and compare existing methods of reducing the uncertainties of the extrapolated cross-section from satellite derived DEM. Mentioned methods are Channel Bankfull Depth approach, Linear Method for Depth Estimation and The Slope-Break Method, which is the focus of this thesis.

The Slope-Break Method considers for each cross section a linear relationship between the section width and water height of the exposed cross section. Domeneghetti (2016) begins with extracting the h-w pairs with the stride of 0.5 meters, figure 3 shows the cross section and the extracted h-w pairs. From the four initial h-w pairs the finite forward difference method is used to calculate the mean derivative. Thereafter the derivatives for the following h-w pairs are computed and compared to the mean derivative, the first derivative that differs from the mean derivative with more than 40 % determines the slope break point. From the slope break point and all the h-w pairs below an
interpolation is made, with a minimum of 4 points. Interpolating a line based on these points, and setting w=0 gives the desired lowest point of the cross section.

![Graph of a river cross section and extracted h-w pairs](image)

**Figure 3:** The left plot shows a river cross section where the blue part has been determined with the SB method. The right plot shows the extracted h-w pairs and where the slope break point is found (after Domeneghetti, 2016).

Domeneghetti (2016) and Mersel et al. (2013) apply the slope break method to approximate river cross sections. However, they have used different approaches regarding for example the deviation, i.e. how much the calculated derivative is allowed to differ from the mean derivative. Domeneghetti uses a deviation of 40% while Mersel et.al uses 30%. The approximation is afterwards compared to already known cross sections. Domeneghetti applies the method on the Po river in Italy, while Mersel et. al. has made a broader study on several large rivers in the world, e.g. the Mississippi river and Ganges-Brahmaputra.

**Purpose and aim**

The general goal of this study is to apply a new methodology to refine the river cross section geometry derived by satellite images. In particular, the final product of this work is a Matlab code as an instruction for future use of the slope break method for applications considering river cross sections.

**Research activities**

The research activities carried out in this study are divided into three parts:

a) This study investigated the slope break point method to modify river cross sections derived by satellite images and, more precisely, produced a code to apply the method to any river cross section.

b) The slope break point method was applied and the Matlab code tested on 8 cross sections along the middle portion of the Po river in Italy. The cross sections extracted from a 90m SRTM DEM were processed by the code and the calculated river thalweg lowering was applied to each cross section. The original and modified SRTM cross sections were then compared to high quality cross sections extracted, in the same locations, from LiDAR 2m. Figure 4 shows the SRTM 90m DEM and the LiDAR DEM of a stretch of the Po river from where 8 cross sections were selected to test our code (see also figure 7).

c) Our Matlab code was adopted by colleagues working on their MSc research focusing on the use of satellite derived information to support hydraulic modelling. The Matlab code was used to modify all the cross sections extracted from the 90 m SRTM of the Po river from Cremona to Borgoforte (about 100 km). Simulations were run using the original SRTM extracted cross sections, modified cross sections using our code and LiDAR derived cross sections.
2. Methods

Computation of the height-width pairs

The purpose of this part is to derive the calculation of the h-w pairs. SRTM based satellite data provide points of the cross section, the blue points in figure 5. However, the values in between these points are not given. Since the h-w pairs with an incremental step of 0.5 meters between the heights are desired, there is a need to find a way to compute the h-w pairs for every 0.5-meter step.

The starting point of the iteration is the value \( h_{\text{min}} + 0.5 \), where \( h_{\text{min}} \) is the lowest point in the cross section. The given values of the cross section, set as vectors, will be compared to the given height, starting from the left of the cross section, to see if a point is above or under the desired height \( h_{\text{min}} + 0.5 \). If the point is greater than the height the code will continue to the following point until it finds a point below the height. If this happens, the code will take the previous point, just above the specified height, and the point below the height, and approximate a station-value for the desired height (elevation-value). This is possible for the left side from the CS-bottom. For the right side, the code has to look for the first point above the height, then go back to the previous point and thereafter compute the second station-value that associated with the desired height (the elevation-value). The approximation is made by the following equation:

\[
\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}
\]  

(1)
which describes linear relations between points, where \( y \) is the chosen height. \((x_1, y_1)\) and \((x_2, y_2)\) are the given points in the station and elevation vectors, lying above and under the specified height \( y \). The only unknown variable will be the x-value, in between the two given points.

**Determination of the Slope Break point**

From the calculated h-w relationship, the slope break point can be determined. In the study made by Domeneghetti, the finite forward differences method is used to calculate the derivatives.

\[
f'(a) \approx \frac{f(a + h) - f(a)}{h}
\]  

(2)

From the four initial points the mean derivative is calculated. Thereafter the following derivatives is compared with the mean derivative: if the slope gets too flat the computation will stop, this indicates the slope break point. The following expression shows the requirement, as used in Domeneghetti’s study (2016):

\[
\frac{dh}{dw} > 0.4 \cdot \frac{\overline{dh}}{dw}
\]  

(3)

**Interpolate to approximate the lowest part of the cross section**

With the points from the previous step (at least 4 points, as suggested by Domeneghetti, 2014), an interpolation will be made with the least square method. From this interpolation, an approximation of the lowest height of the cross section can be calculated by setting the width equal to zero. This point is referred to as \( z_{\text{min}} \). Once the lowering of the cross section is estimated, it will be subtracted to the lowest point or points of the section and the new section compared to the original one. How great the lowering are is given by the difference between \( z_{\text{min}} \) and \( h_{\text{min}} \), and is referred to as \( h_{\text{low}} \).

**ArcGIS and Excel**

The code is tested on a part of the river Po in Italy. The location of the river is shown in figure 6. From the river a few cross sections were chosen for the application, they are shown in figure 7.

![Figure 6: The stretch of the river Po in northern Italy with the studied part of the river at the tip of the arrow.](image-url)
Data of the chosen cross sections is extracted from available 90m SRTM DEM (Jarvis et al., 2008, http://srtm.csi.cgiar.org/SELECTION/inputCoord.asp) and LiDAR 2m DEM (Interregional Agency for the Po River, AIPO 2005) by means of ArcMap. To make the work more clear the ID:s used in ArcMap is changed into new numbers as seen in table 1.

Table 1: The cross sections chosen from ArcMap are given new ID:s.

<table>
<thead>
<tr>
<th>New ID</th>
<th>GIS ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S40</td>
</tr>
<tr>
<td>2</td>
<td>S39F</td>
</tr>
<tr>
<td>3</td>
<td>S39E</td>
</tr>
<tr>
<td>4</td>
<td>S39D</td>
</tr>
<tr>
<td>5</td>
<td>S39C</td>
</tr>
<tr>
<td>6</td>
<td>S39B</td>
</tr>
<tr>
<td>7</td>
<td>S38E</td>
</tr>
<tr>
<td>8</td>
<td>S37C</td>
</tr>
</tbody>
</table>

Two Excel-files are created, the first containing SRTM data and the second containing LiDAR based data. The files are structured based on cross section ID, station and elevation values, as seen in table 2 below. All chosen cross sections will therefore be listed in sequence.
Table 2: Example of the structure of the Excel-file with cross section ID (new ID) and station/elevation-values.

<table>
<thead>
<tr>
<th>ID</th>
<th>Station</th>
<th>Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>136,8966</td>
<td>14,2865</td>
</tr>
<tr>
<td>1</td>
<td>205,3445</td>
<td>14</td>
</tr>
<tr>
<td>1</td>
<td>273,7921</td>
<td>14</td>
</tr>
<tr>
<td>1</td>
<td>342,2394</td>
<td>14,54</td>
</tr>
<tr>
<td>1</td>
<td>410,6865</td>
<td>17,9076</td>
</tr>
<tr>
<td>2</td>
<td>129,6121</td>
<td>17,6146</td>
</tr>
<tr>
<td>2</td>
<td>194,4179</td>
<td>15,2659</td>
</tr>
<tr>
<td>2</td>
<td>259,2236</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>583,2497</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>648,0544</td>
<td>14,0852</td>
</tr>
<tr>
<td>2</td>
<td>712,859</td>
<td>15,1534</td>
</tr>
<tr>
<td>2</td>
<td>907,2719</td>
<td>23,834</td>
</tr>
</tbody>
</table>

With this structure, the Matlab code can read the Excel-files to compute and to plot the information.

3. Results

a) Matlab code for finding lowering of the cross section through the slope break method

A Matlab code has been conducted for evaluation of the cross section, for the whole code see Appendix A. Firstly the cross section points must be extracted from an excel file into Matlab in order to run the code. In figure 8, a part of the code is shown. This part calls for an excel file with name ExCS. How the excel file is constructed is explained above. The code finds how many unique ID values there are, i.e. how many cross sections there are. For every unique ID two vectors are formed, one with the elevation values and one with the station values.

```matlab
v=xlsread('ExCS'); %name of file
f=unique(v(:,1));

----------- %some code...

extraction of CS vectors

for b=1:length(v) %for 2
    if v(b,1)==q %if 1
        e=[e v(b,2)];
        e=[e v(b,3)];
    end %if 1
end %for 2
```

Figure 8: Extraction of station/elevation vector form excel

Moreover, the code continues with computation of the h-w-pairs, shown in figure 9. As the method says, h should be given a specified value and the code will search for the first element in the elevation vector which lies below the specified h. When this point is found a calculation will be made from equation 1. This calculation will give a corresponding x-value to the height h. When this point is found the code continues to locate the first point that is above the given h, notice that it starts from the point that was found previously and not from the beginning of the elevation vector. The structure of this procedure is shown in figure 10.
The code begins with finding the point \( x_i \) and then moves back to \( x_{i-1} \) for calculations with the linear interpolation equation. From the point \( x_i \) the code continues to search for the first point above the given \( h \) which is \( x_j \). Likewise, the code then picks \( x_{j-1} \) and do the linear interpolation to find the \( x \)-value. Therefore it has conducted two \( x \)-values for the given height. For more complex cross sections there can be more \( x \)-values, depending on the shape. From these two \( x \)-values the width of the cross section is calculated.

As the code computes \( h \)-\( w \)-pairs it also starts computing the slope between these pairs. Firstly the slope is calculated between the first two \( h \)-\( w \) pairs, and then it continues on with the next coming pairs. When three slopes have been calculated, the mean derivative of these are determined. This mean slope will then work as a reference, when the following calculated slope is getting to flat the code will stop. That is to say, the slope break point has been found according to equation 3. These parts of the code are shown in figure 11 and figure 12.
while n>=0.4

Figure 11: The code will stop when the SB point is found, i.e. when the slope gets to flat and therefore n is greater than or equal to 0.4.

```matlab
% calculates the slope between two h-w-pairs
if m>2
    slope=(hnew(m-1)-hnew(m))/(wnew(m-1)-wnew(m));
    slopevector=[slopevector slope];
    u=length(slopevector);
end

% meanvalue calculation
if u==3
    meanslope=mean(slopevector);
end

% quotient that determines the SB
if u>=3
    n=slopevector(u)/meanslope;
end

end %if 7
```

Figure 12: The part of the code where the condition of the SB is computed.

**b) Application of the code to eight selected cross section along the Po river, Italy**

With the finalized Matlab code the application to the Po river was made. The code presents the following plots shown in figure 13 and also gives information as shown in table 3. In Figure 13 the results and plots for cross section S39D is shown, the results and plots for the other cross sections are presented in Appendix B. Subplot a) displays the cross sections plotted from SRTM and the calculated h-w pairs, the filled dot represents the slope break point. Subplot b) shows the linear interpolation of the h-w-pairs and the yellow box represents the lowest part of the cross section, \( z_{\text{min}} \). Subplot c) compares the LiDAR extracted cross section to the original SRTM and the two modified SRTM cross sections: one applying the calculated lowering assuming a rectangular cross section below the water level and one assuming a triangular cross section. Subplot d) compares the LiDAR to the original SRTM and the two modified SRTM cross sections with rectangular and triangular lowering, but the lowering is not \( z_{\text{min}} \) but the average \( z_{\text{min}} \) for all eight cross sections. The average lowering for the eight cross sections is 4.5 meters.
Table 3: Information regarding the cross sections given by the code.

<table>
<thead>
<tr>
<th>Cross section 1</th>
<th>Cross section 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slope Break is found</td>
<td>Slope Break is found</td>
</tr>
<tr>
<td>$z_{min}$, 9.9842</td>
<td>$z_{min}$, 8.3202</td>
</tr>
<tr>
<td>$h_{low}$, 4.0158</td>
<td>$h_{low}$, 6.6798</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cross section 2</th>
<th>Cross section 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slope Break is found</td>
<td>Slope Break is found</td>
</tr>
<tr>
<td>$z_{min}$, 6.9083</td>
<td>$z_{min}$, 14.0032</td>
</tr>
<tr>
<td>$h_{low}$, 7.0917</td>
<td>$h_{low}$, 1.0767</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cross section 3</th>
<th>Cross section 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slope Break is found</td>
<td>Slope Break is found</td>
</tr>
<tr>
<td>$z_{min}$, 9.3233</td>
<td>$z_{min}$, 9.3438</td>
</tr>
<tr>
<td>$h_{low}$, 4.6767</td>
<td>$h_{low}$, 7.6562</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cross section 4</th>
<th>Cross section 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slope Break is found</td>
<td>Slope Break is found</td>
</tr>
<tr>
<td>$z_{min}$, 9.8983</td>
<td>$z_{min}$, 18.3871</td>
</tr>
<tr>
<td>$h_{low}$, 4.1017</td>
<td>$h_{low}$, 0.61288</td>
</tr>
</tbody>
</table>

Figure 13: a) CS 3 and h-w pairs b) Linear interpolation c) Approximated CS d) Approximated CS with meanvalue lowering.
c) **Hydraulic simulations with and without SRTM modified cross sections**

By the collaboration with the colleagues working on their research of hydraulic modelling (Milos Dinic and Victor Alejandro Ortiz Peña), they shared the results displayed in figure 14 and figure 15. In their model they used a hydrograph upstream in Cremona as a boundary condition, and then the bed slope in Borgoforte as a downstream condition. The model was calibrated based on the water levels observed in Boretto, a cross section in the center of the river stretch, where a gauge station is available. This shows the difference if using the mean value (average) lowering or not. In figure 14 the difference between the observed water level and the simulated water level from satellite based DEM is shown, note that it is before the code has been applied. Figure 15 shows the difference between the observed water level and the simulated water level based on the code modified SRTM data with mean value lowering, i.e. the code has now been applied. The difference has decreased. The mean value lowering for all cross sections is approximately 3.7 m.

**Figure 14:** Water level at Boretto station showing the observed and the simulated from SRTM without the lowering.

**Figure 15:** Water level at Boretto station showing the observed and the simulated from modified SRTM with average lowering.
4. Discussions

The Matlab code presents the results requested, i.e. the slope break point, approximated lowest point $z_{min}$ and a comparison of the LiDAR and modified SRTM-data. While conducting the code there was a discussion regarding the range for the slope break point. Firstly, a different approach in relation to Domeneghetti (2016) was tried. This was about checking each slope between the computed h-w-pairs within a range of 0.5 to 1.5 multiplied the previous slope. This turned out to not be successful. Therefore, we did eventually go back to the way Domeneghetti had shown in his work On the use of SRTM and altimetry data for flood modelling in data-sparse regions (2016), to use the range as seen in equation 3. This way did stress a more uniform and clear slope to use further, since it was based on the mean value of the three first h-w pairs.

Depending of the shape of the cross section the comparison between LiDAR and the modified SRTM results will vary in reliability. When the lowering were made with the mean value of $z_{min}$ for all the eight cross sections, an improvement could be seen for several of the cross section for example S39B and S37C. The code will only be an approximation of the reality and there are possibilities to improve the code to make it more precise.

Of course there is also a big difference between the rectangular and triangular lowering. The rectangular lowering can carry a greater volume than the triangular. Which one who makes the best approximation depends on the shape of the cross section. Perhaps a good approximation will be to simulate the water depth for both of these options in Hec-Ras and with this knowledge calculate the mean depth of these two lowerings.

The significance of this study is broad, i.e. for future applications regarding river cross sections, for example flood prediction or other hydrological or hydraulic modelling. In the current situation, it is hard to get high quality data of river cross sections in respect to price, quality and quantity. This code can simplify the work of improve a big amount of SRTM river data in an effective way to a reasonable price. For instance, more precise flood modelling could be made possible in remote areas.

5. Conclusions

a) A code was created that computes slope break point, approximates the lowest point $z_{min}$ and shows a comparison of the LiDAR and modified SRTM-data.

b) The code was applied on multiple cross sections, and illustrates the three cases: SRTM data and modified SRTM data compared with LiDAR data.

c) Hydraulic simulations demonstrated that the code improved the bathymetry of the river cross section, i.e. the water level approximation coincides better with the observed water levels.

Overall, the modified SRTM agree to a greater extent with the LiDAR than the original SRTM. The mean value lowering seems to give the modified SRTM- cross section a more similar lowering in relation to the LiDAR and therefore gives improved water levels as shown by the hydraulic model.
6. References


Appendix A

%Following code will compute an approximated lowering of a river thalweg, with input based on SRTM data. The input file should be build up as follows: In a Excel file, three columns should be created where the first one should contain the ID numbers, the second should include station values and the third includes elevation values. The same procedure should be applied when creating an input file of LiDAR data.

clear, clc, clf
tic;
v=xlsread('SRTM'); %name of file
v2=xlsread('Lidar'); %name of file
f=unique(v(:,1));

iter=0; iter2=0; iter3=0; hlowvec=[];

for q=1:1:length(f)
s=[]; e=[]; slidar=[]; elidar=[];

%extracting CS vectors
for b=1:1:length(v)
    if v(b,1)==q %if 1
        s=[s v(b,2)];
        e=[e v(b,3)];
    end %if 1
end %for 2

%identify some variables
hmin=min(e);
h=hmin+0.5;
wnew=[]'; hnew=[]'; slopevector=[]'; n=1;
eends=[e(1) e(end)];
eendmin=find(eends==min(eends));

%condition for SB
while n>=0.4

wx=[]'; i=1;

    if h>eends(eendmin)
        break
    end

%evaluates all the points in the vector and calculates the width
while i<length(e)
    i=i+1;

    if e(i)<h
        if iter<1
            % code
        end
    else
        % code
    end
end

end

end
iter=1;
xi=((s(i-1)-s(i))*(h-e(i)))/((e(i-1)-e(i)))+s(i);

j=i;
end

while e(j) <= h
  j=j+1;
  if e(j)> h
    if iter2<1
      iter2=1;
      xj=(((h-e(j-1))*(s(j)-s(j-1)))/(e(j)-e(j-1)))+s(j-1);

      wij=xj-xi;
      wx=[wx wij];
    
    i=j;
    end

  end

  %resets, makes it able to move past the peaks
  if i==j
    iter=0; iter2=0;
  end

end
end

%sums up all the widths found for a specific height
wny=sum(wx);

hnew=[hnew h];
wnew=[wnew wny];
m=length(wnew);

%s calculates the slope between two h-w-pairs
if m>=2
  slope=(hnew(m-1)-hnew(m))/(wnew(m-1)-wnew(m));
slopevector=[slopevector slope];
u=length(slopevector);

%meanvalue calculation
if u>=3
  meanslope=mean(slopevector);

%quotient that determines the SB
end

if u>=3
n = slopevector(u) / meanslope;
end

h = h + 0.5;
end

%%%linear interpolation least square method

w = []; hw = [];
for o = 1:m-1
    w = [w wnew(o)];
    hw = [hw hnew(o)];
end

if length(w) >= 4
    wn = [ones(size(w))
          w]';
    wny = wn';
    c = (wny*wn) \ (wny*hw');
    x = linspace(0, wnew(m));
    y = c(2)*x + c(1);
    zmin = c(1);
    xmin = s(1) - 2;
    xmax = s(end) + 2;
    ymin = zmin - 2;
    ymax = max(e) + 2;
figure(q)
subplot(2, 2, 1)
plot(s, e, 'b')
hold on
plot(s, e, '*b')
xlabel('Station [m]')
ylabel('Elevation [m]')
title(['CS ' num2str(q) ' and h-w-pairs'])
axis square
axis([xmin xmax ymin ymax])
hold on
plot(wnew, hnew, 'o', 'markerEdgeColor', [1 0.4 0.6])
hold on
plot(w(end), hw(end), 'o', 'MarkerFaceColor', [1 0.4 0.6])
subplot(2, 2, 3)
plot(w, hw, 'o', 'markerEdgeColor', [1 0.4 0.6])
hold on
plot(x, y, 'k')
axis square
plot(0, zmin, 'square', 'markerFaceColor', [1 1 0], 'markersize', 10)
xlabel('Cross section width [m]')
ylabel('Cross section height [m]')
title(['Linear interpolation'])

hlow=hmin-zmin;
hlowvec=[hlowvec hlow];

disp(['Cross section ' num2str(q)]

disp('Slope Break is found')
disp(['zmin, ' num2str(zmin)])
disp(['hlow, ' num2str(hlow)])

r=find(e==min(e));
ii=1;
jj=1;

if e(r(end)+1)-e(r(end))<1.5
    while e(r(end)+ii)-e(r(end))<1.5
        if e(r(end)+ii)-e(r(end))<1.5
            iter3=1;
            qt=[r find(e==e(r(end)+ii))];
        end
    end
    ii=ii+1;
end

r=sort(qt);
end

if e(r(1)-1)-e(r(1))<1.5
    while e(r(1)-jj)-e(r(1))<1.5
        if e(r(1)-jj)-e(r(1))<1.5
            iter3=1;
            qt=[find(e==e(r(1)-jj)) r];
        end
    end
    jj=jj+1;
end

end

r=sort(qt);
end

if length(r)<2
    g1=find(e==e(r(1)-1));
    g2=find(e==e(r(end)+1));
    qt=[g1 r g2];
    r=sort(qt);
end

%extracting CS lidar vectors
for b2=1:1:length(v2)
    if v2(b2,1)==q
        slidar=[slidar v2(b2,2)];
        elidar=[elidar v2(b2,3)];
    end
end

subplot(2,2,2)
plot(slidar,elidar,'k')
hold on
plot(s,e,'-b')
esq=[e(r(1)) zmin zmin e(r(end))];
ssq=[s(r(1)) s(r(1)) s(r(end)) s(r(end))];
etri=[e(r(1)) zmin e(r(end))];
stri=[s(r(1)) (s(r(end))+s(r(1)))/2 s(r(end))];

plot(stri, etri,'--',ssq, esq,'--')
title(['Approximated CS'])
xlabel('Station [m]')
ylabel('Elevation [m]')
legend('Lidar','Original SRTM','Modified SRTM triangular', 'Modified SRTM rectangular')
else
    disp(['Cross section ' num2str(q) ''])
    disp('not enough points')
xmin=s(1)-2;
xmax=s(end)+2;
ymin=min(e)-2;
ymax=max(e)+2;

    figure(q)
    subplot(2,2,2)
    plot(s,e,'b')
    hold on
    plot(s,e,'*b')
    xlabel('Station [m]')
    ylabel('Elevation [m]')
    title(['CS ' num2str(q) ' SB not found'])
    hold on
    axis square
    axis([xmin xmax ymin ymax])
    plot(wnew, hnew, 'o')
end

end
meanhlow=sum(hlowvec)/q;
for q=1:1:length(f)
s=[]; e=[]; slidar=[]; elidar=[];

for b=1:1:length(v)
    if v(b,1)==q
        s=[s v(b,2)];
        e=[e v(b,3)];
    end
end

r=find(e==min(e));
ii=1;
jj=1;

if e(r(end)+1)-e(r(end))<1.5
    while e(r(end)+ii)-e(r(end))<1.5
        if e(r(end)+ii)-e(r(end))<1.5
            iter3=1;
            qt=[r find(e==e(r(end)+ii))];
            end
        ii=ii+1;
        end
    r=sort(qt);
end

if e(r(1)-1)-e(r(1))<1.5
    while e(r(1)-jj)-e(r(1))<1.5
        if e(r(1)-jj)-e(r(1))<1.5
            iter3=1;
            qt=[find(e==e(r(1)-jj)) r];
            end
        jj=jj+1;
        end
    r=sort(qt);
end

if length(r)<2
    g1=find(e==e(r(1)-1));
    g2=find(e==e(r(end)+1));
    qt=[g1 r g2];
    r=sort(qt);
end
for b2=1:length(v2)
    if v2(b2,1)==q
        slidar=[slidar v2(b2,2)];
        elidar=[elidar v2(b2,3)];
    end
end

hmin=min(e);
zmin=hmin-meanhlow;

figure(q)
subplot(2,2,4)
plot(slidar,elidar,'k')
hold on
plot(s,e,'+-b')
esq=[e(r(1)) zmin zmin e(r(end))];
ssq=[s(r(1)) s(r(1)) s(r(end)) s(r(end))];
etri=[e(r(1)) zmin e(r(end))];
stri=[s(r(1)) (s(r(end))+s(r(1)))/2 s(r(end))];
plot(stri, etri,'--',ssq, esq,'-')

title(['Approximated CS with meanvalue lowering'])
xlabel('Station [m]')
ylabel('Elevation [m]')

end

toc;
Appendix B

Figure 16: S40

Figure 17: S39F
Figure 22: S38E

Figure 23: S37C