



TECHNISCHE  
UNIVERSITÄT  
WIEN  
Vienna University of Technology

MASTERARBEIT

# **Towards an Automated SAR-Based Classification of Water Bodies**

Ausgeführt am Department für  
Geodäsie und Geoinformation  
der Technischen Universität Wien

unter der Anleitung von Prof. Dr. Wolfgang Wagner

durch

Felix Greifeneder  
Piaristengasse 18/8, 1080 Wien

## Contents

1	List of Acronyms .....	4
2	Abstract.....	6
3	Zusammenfassung .....	6
4	Acknowledgments.....	8
5	Introduction .....	9
6	Methods.....	12
6.1	Synthetic Aperture Radar.....	12
6.2	Image segmentation methods .....	16
6.2.1	Otsu's Algorithm .....	19
6.3	Temporal Stability Approach.....	20
6.3.1	Regression Analysis.....	23
6.3.2	Averaging Methods .....	30
7	Data.....	33
7.1	Datasets.....	33
7.2	Advanced Synthetic Aperture Radar .....	33
7.3	Study Area .....	36
8	Processing .....	40
8.1	Pre-Processing.....	40
8.1.1	Geo-Rectification.....	41
8.1.2	Resampling.....	43
8.1.3	Normalisation .....	43
8.2	Otsu Method .....	44
8.2.1	Image pre-processing for the Otsu Method.....	44
8.2.2	Otsu Algorithm.....	45
8.3	Temporal stability approach.....	47
8.4	Post classification filtering.....	48

9	Results .....	50
9.1	R-Layer .....	51
9.2	Averaging results.....	56
9.3	Otsu's Method .....	59
9.4	Temporal Stability Approach.....	62
9.4.1	Comparison of different averaging areas .....	62
9.4.2	Comparison between mean and median .....	71
9.4.3	Simple least squares or total least squares (orthogonal) regression .....	73
9.5	Post classification filtering.....	76
10	Accuracy Assessment.....	77
10.1	Flood extent marks.....	77
10.2	Geocoding.....	80
11	Conclusion.....	82
12	Bibliography .....	85
13	Appendix 1 .....	90
14	Appendix 2 .....	94
15	Appendix 3 .....	104

## 1 List of Acronyms

CRP	Centre de Recherche Public
ISDR	International Strategy for Disaster Reduction
UNISDR	United Nations International Strategy for Disaster Reduction
ASAR	Advanced Synthetic Aperture Radar
UK	United Kingdom
SAR	Synthetic Aperture Radar
ESA	European Space Agency
IEEE	Institute of Electrical and Electronics Engineers
ASCAT	Advanced Scattermoeter
HH	Horizontal – Horizontal
VV	Vertical – Vertical
HV	Horizontal – Vertical
VH	Vertical – Horizontal
WS	Wide Swath
PCA	Principal Component Analysis
ENVISAT	Environmental Satellite
WSM	Wide Swath Mode
SRTM	Shuttle Radar Topography Mission
DEM	Digital Elevation Model

DORIS	Doppler Orbitography and Radio-Positioning Integrated by Satellite
ERS	European Remote Sensing Satellite
MWR	Microwave Radiometer
RA-2	Radar Altimeter 2
SSO	Sun-Synchronous Polar Orbit
MLST	Mean Local Solar Time
SSR	Solid State Recorder
MERIS	Medium Resolution Imaging Spectrometer
BBC	British Broadcasting Corporation
IDL	Interactive Data Language
NEST	Next ESA SAR Toolbox
CORINE	Coordination of Information on the Environment
RMS	Root Mean Square Error

## 2 Abstract

Spaceborne microwave remote sensing in general and Synthetic Aperture Radar particularly is a valuable data source for water body classification. Due to its insensitivity to atmospheric conditions it can be reliably used for flood delineation purposes, where – due to cloud cover – it often is impossible to acquire optical data. The objective of this thesis is to study the possibilities and limitations of automatically derived flooding maps, based on SAR-imagery. For that purpose two approaches for water body classification are compared: The first is based on Otsu's Algorithm (*N. Otsu, 1979*) which performs a histogram thresholding on the grey-level histogram from a single radar image. The second approach exploits the concept of temporal stability (*Wagner et al., 2008*), here the spatiotemporal relation between regional and local backscatter is exploited to detect flooding areas.

Both techniques are tested and optimised using different processing parameters. As a study area a major flooding of the river Severn is used which occurred in July 2007. The temporal stability approach exceeds Otsu's algorithm in terms of classification quality and robustness. Albeit, if no historic data archive is available (the temporal stability approach is based on a time series analysis), Otsu's method can serve as an acceptable alternative.

## 3 Zusammenfassung

Satellitengestützte Mikrowellen-Fernerkundung generell, und Synthetic Aperture Radar (SAR) im Besonderen, ist eine wichtige Datenquelle für die Klassifizierung von Wasserflächen. Auf Grund der mit Überschwemmungen oft einhergehenden Wolkendecke, sind Daten aus optischen Sensoren für die Kartierung von Überschwemmungsflächen oft unbrauchbar. Die Erfassung von SAR-Daten hingegen wird durch atmosphärische Bedingungen kaum beeinflusst. Das Ziel dieser Arbeit ist eine Untersuchung der Möglichkeiten und Grenzen einer automatisierten Kartierung von Überflutungen mit Hilfe von SAR. Zwei verschiedene Klassifizierungsansätze werden getestet: Die erste Methode basiert

auf Otsu's Algorithmus (*N. Otsu*, 1979). Wasserflächen werden durch die automatische Berechnung eines Histogramm-Grenzwertes erkannt. Die zweite Methode basiert auf dem Konzept der „zeitlichen Stabilität“ (*Wagner et al.*, 2008), dabei wird der zeitliche und räumliche Zusammenhang von Radar-Rückstreuung verwendet um überflutete Flächen zu erkennen.

Beide Ansätze werden, unter Verwendung verschiedener Parameter, getestet und verglichen. Als Testgebiet dient der „River Severn“, im Westen Groß Britanniens, und schwere Überschwemmungen die dort im Juli 2007 stattfanden. Die Ergebnisse, basierend auf der „zeitlichen Stabilität“ von Radar Rückstreuung, übertreffen jene der Otsu Methode im Bezug auf Klassifizierungsgenauigkeit und Stabilität, ist allerdings kein historisches SAR-Datenarchiv verfügbar, bietet Otsu's Algorithmus eine gute Alternative zur Kartierung von Überschwemmungsflächen.

## 4 Acknowledgments

I would like to express my greatest gratitude to the people who have helped and supported me throughout my project. I am grateful to my Professor Dr. Wolfgang Wagner for his continuous support of the project, from initial advice and contact in the early stages of conceptual inception to constructive advice and encouragement up to this day.

A special thank of mine goes to my colleagues who helped me in completing the project. Particularly I want to mention Daniel Sabel and Stefan Schlaffer. Daniel was of great help during the early stages of this work, supporting me in the phase of conceptualisation and finding relevant data in the university's vast data archives, Stefan – with his expertise on the topic – aided the project through the exchange of interesting ideas and thoughts.

Furthermore I would like to thank the Cambridge University Unit for Landscape Modelling for enabling the use of aerial photography of the river Severn region. The data was the basis for the derivation of flood extent marks, which were used for quality assessment in this work. The flood extent marks were digitised by my colleague Stefan Schlaffer in cooperation with Patrick Matgen, Laura Giustarini and Renaud Hostache during a recent stay at CRP Lippmann in Belgium.

At last but not least I want to thank my family and friends who supported me throughout my studies and helped to reach the point at which I am now, at the brink of concluding my master degree.



## 5 Introduction

Even though the numbers on occurrence and impact of natural disasters remain patchy, there has been an increase of reported events in recent years. Due to the increasing global population and the increase of densely populated areas the humanitarian and economical impact of natural disasters has been rising steadily. According to a report by the Feinstein International Centre, on average, more than 250 million people per year have been affected by natural disasters. The number has been increasing by 50,000 to 60,000 people per decade since the early 1970s.

Compared to other natural disasters flooding has a very high relevancy. According to statistics from the International Strategy for Disaster Reduction (ISDR) flooding is the most frequent disaster in all parts of the world except Africa and Oceania. Between the years 2001 and 2010, based on a yearly average, 45.6 % of all natural disasters were floods [UNISDR, 2011]. It is the number one cause for losses from natural hazards; it causes losses of both life and property and about one-third of all geophysical hazards are floods.

There are some signals indicating that the occurrence rate of extreme weather events will further increase in the future, due to climate change [Allan, 2008]. Karl and Knight (1998) used the data from approximately 200 long-term precipitation monitoring stations to show that the sums of the highest monthly daily precipitation events had been significantly increasing (between the years 1910 and 1995) for four of the nine regions of interest. According to R. P. Allan (2008), projections of tropical precipitation through this century show an increase in moist equatorial regions. Some climate models suggest that extreme precipitation will become more common in an anthropogenically warmed climate. The effects and severity of climate change will show differently for different global regions, observational evidence supports the thesis that already moist regions are becoming wetter while dry regions are getting dryer [Allan, 2008]. However, the prediction of climate change effects is not as straight forward as one might think; complex atmospheric processes and a large number of influences have to be considered. Reynard et al. (2004) investigated the

influence of climate change on flood frequencies in the United Kingdom, based on three different climate change scenarios. For example, the predicted change in peak flow, for a 50-year return period flood in the 2080s, for the Thames at Kingston catchment, ranges between 0.6% and 17.7% [Reynard *et al.*, 2004]. The outlook for the River Severn area, which has been prone to intense flooding throughout history, indicates a decrease of peak flows. Depending on the applied climate change scenario, the changes in the 50-year return period peak flows range between -6.7% and -23.8%. This shows that due to the complex combination of many variables, predictions of the effects of climate change are afflicted with a high level of uncertainty and can only be made for specific regions, under special consideration of the local situation.

Remote sensing data can be a tool to increase the understanding of flooding and to enhance hydraulic models to improve predictions about the likely occurrence and severity of such events in the future. It can be carried out from terrestrial, airborne or from space-borne platforms. This work will be based on data from space borne platforms, primarily. One advantage of data acquired from space is that there are systems with a high temporal resolution (e.g. on average three days for the ASAR instrument in Global Monitoring Mode). With airborne systems, it is difficult to achieve a temporal resolution as high. Usually, the acquisition is very costly. Therefore, satellite remote sensing is highly suitable for the monitoring of long term environmental changes but also, for the near real time response to flooding events including mapping and damage assessment. Data can be made accessible very quickly. Presently, maps derived from remotely sensed data already are an important factor for the coordination and the execution of rescue missions and instant help for natural hazard incidents around the globe.

Remote sensing data is widely used for applications in the field of natural hazards control and has been subject of research many times. Although optical data has a benefit in terms of spatial resolution, its use during flooding events is often problematic due to persistent cloud cover during such events. Thus, the application of such data is feasible only in a limited number of situations. The clear advantages of radar are the low sensitivity to atmospheric conditions (i.e.

clouds, rain) and the independence of the sun as a natural source of illumination. This means that there are few factors hindering a successful data acquisition.

There is a range of different radar measurement techniques. The most appropriate for the purposes in this thesis is the so-called Synthetic Aperture Radar (SAR). SAR utilises the Doppler-Shift to synthesise a large antenna to overcome the necessity of a physically large antenna, which would be needed to achieve spatial resolutions, high enough for the application of water body classification. In 2013 a new SAR instrument will be launched on board the new ESA Sentinel-1 mission. Contrariwise to the common practice on present SAR sensors, where it is switched between imaging modes as they are needed (different spatial resolutions and polarisations), the new Sentinel satellites will have a fixed schedule to follow. This further improves the availability and the usability of the data.

For a near real time classification of water bodies, there should be an automated data processing routine. Extensive research has been done on the topic of automatic image thresholding [e.g. *Matgen et al*, 2011; *Di Baldassarre et al.*, 2009; *Martinis et al.*, 2008]. This means, water and non-water surfaces are distinguished by defining a threshold value which separates the two classes. Most of the studies available either use a fixed grey-level threshold or a histogram based auto-thresholding approach. However, many of these approaches require at least some kind of manual interaction.

The objective of this thesis is, to study the possibilities and limitations of automatically derived flooding maps, based on SAR-imagery. The approach should be applicable globally. For that purpose we want to investigate two different techniques. One will be a histogram based thresholding approach based on the so called Otsu – Algorithm (*N. Otsu*, 1979). The other technique features a more complex approach; it exploits the so called temporal stability of radar backscatter time-series, as it was described by *Wagner et al.* (2008).

## 6 Methods

### 6.1 Synthetic Aperture Radar

Synthetic Aperture Radar (SAR) or Radar (short for Radio Detection and Ranging) in general utilises electromagnetic radiation within the microwave frequency range. Although there isn't a firm definition of this range, it is a common convention that it extends from 0.3 to 300 GHz. Usually the microwave spectrum is subdivided into a number of classes, usually named by letters. There are different classification systems, the position and size of the classes' origin from the use for different applications and are more or less arbitrarily set. Table 6.1.1: Band designations according to the IEEE standard 521-1984 shows the classification according to the IEEE standard number 521-1984.

Band Designator	Frequency [GHz]	Wavelength [cm]
<b>L</b>	1 to 2	30.0 to 15.0
<b>S</b>	2 to 4	15.0 to 7.5
<b>C</b>	4 to 8	7.5 to 3.8
<b>X</b>	8 to 12	3.8 to 2.5
<b>Ku</b>	12 to 18	2.5 to 1.7
<b>K</b>	18 to 27	1.7 to 1.1
<b>Ka</b>	27 to 40	1.1 to 0.75
<b>V</b>	40 to 75	0.75 to 0.40
<b>W</b>	75 to 110	0.40 to 0.27

*Table 6.1.1: Band designations according to the IEEE standard 521-1984*

The one major advantage of remote sensing systems using microwaves, rather than electromagnetic radiation in the region of visible light; especially for lower

frequencies (up to 20 GHz) measurements are hardly affected by clouds or rain. This property makes radar an extremely powerful system for applications where one has to be able to acquire data in regular intervals and it often compensates the drawback of a, compared to optical systems, poor spatial resolution. SAR is an active radar configuration. Thus, Synthetic Aperture Radar is independent of any natural source of illumination upon the target area and therefore independent of the day – night circle, which constitutes another advantage of SAR in particular.

In radar imaging, a distinction has to be made between the resolution in the range direction (cross-track resolution) and the resolution in the azimuth direction (along-track resolution). Classical radar systems can achieve only a rather poor spatial resolution. As an example, the Advanced Scatterometer (ASCAT) which is flown on a series of METOP satellites since 2006 can provide products with a resolution of 25 and 50 km. For many applications, including water body classification) this is too coarse. To achieve a good azimuthal resolution, in a classical radar imaging process, a physically very large antenna would be necessary to focus the transmitted and received energy into a sharp beam, due to the radar beam widening, caused by diffraction. This is a problem for space borne radar systems, especially.

The SAR technology uses a small antenna, moving along a flight path, to synthesize a much larger antenna. The synthetic aperture technique exploits the fact that a target stays within the beam for a significant amount of time and is observed by the radar from numerous locations along the satellite path. As long as the signals are coherently recorded, a single antenna moving along a flight path is equivalent to an array of antennas. In a SAR measurement both the amplitude and the phase of the returned signal has to be measured. A phenomenon called the “Doppler Shift” is utilized to determine from which part of the beam the echo returns. The “Doppler Shift” describes frequency changes of a signal received or transmitted by a moving source or receiver, respectively. If a satellite with a receiving antenna is moving towards the source of radiation, the signal is compressed, which means a decrease of wavelength and thus, an increase of frequency. On the other hand, if the receiver is moving away from the

source, the opposite effect applies. The wave gets stretched, which results in a longer wavelength and thus, lower frequency.

This phenomenon can be easily observed in everyday situations, like the passing of an ambulance car. If the car is approaching the sound of the siren is higher pitched, whereas when the car moves away the pitch is lower. The frequency shift is expressed by Equation 6.1.1, with  $V_{rel}$  being the relative motion between detector and source and  $\lambda$  being the wavelength of the signal.

$$f_D = \frac{V_{rel}}{\lambda} \quad 6.1.1$$

Modern SAR systems will be able to acquire imagery with a spatial resolution down to 1 m.

For synthetic aperture radar systems, there is no need for a physically large antenna. Unlike in classical radar imaging systems, a larger antenna doesn't increase the spatial resolution. On the contrary, theoretically a small antenna will increase the quality of the measurement. A small antenna causes a strong widening of the radar beam. According to Equation 6.1.2 a system following the synthetic aperture principle will have an azimuth spatial resolution, which is equal to half the length of the antenna.  $V_s$  expresses the relative speed,  $B_D$  the Doppler bandwidth,  $\lambda$  the wavelength and  $D$  the length of the real antenna.

$$\rho_a = \frac{V_s}{B_D} = \frac{V_s \lambda 2D}{4V_s \lambda} = \frac{D}{2} \quad 6.1.2$$

Another advantage of a wide beam and thus the high overlap between the individual measurements is that more images can be combined to increase the signal to noise ratio. The changes in frequency can be used to classify strips of equal frequency within one measurement. To increase the signal to noise ratio, overlapping images can then be combined.

The radar backscatter is influenced by a number of different parameters. One can distinguish between radar observation parameters and surface parameters.

### **Radar observation parameters:**

- Frequency

The frequency of the incidence radiation influences the penetration depth of the waves into target surface and the relative roughness of the surface considered. Penetration depth however, is also related to the moisture content of the target

- Polarisation

The linear polarisation configurations HH (Horizontal-Horizontal), VV (Vertical-Vertical), HV (Horizontal-Vertical) and VH (Vertical-Horizontal) are commonly used. The first letter refers to the polarisation of the emitted wave, the second to the polarisation of the received wave. The polarisation can provide information about the orientation of small scattering elements.

- Incidence angle

The incidence angle is defined by the angle between the perpendicular to the imaged surface and the direction of the incident radiation. For most targets the backscatter coefficient varies with the incidence angle.

### **Surface parameters:**

- Roughness

As mentioned above, the surface roughness is a relative concept depending on wavelength and incidence angle. According to the Rayleigh criterion a surface is considered smooth if it fulfils Equation 6.1.3 ( $h$  being the mean height of surface variations,  $\lambda$  wavelength and  $\theta$  the incidence angle).

Generally, it can be said that the smoother a surface, the lower the received backscatter radiation. This is due to specular reflection, comparable to a mirror in the optical domain. Water usually has a very smooth surface; therefore water typically has low backscatter values, in radar images water surfaces appear dark, and can be distinguished from other land cover easily. Wind-stress or current can cause an increase of

surface roughness (e.g. waves) and decrease the detectability of water bodies.

- **Moisture**

The relative permittivity of a material is given by the dielectric constant.

It consists of two parts, the permittivity and the conductivity, both are highly dependent on the moisture content of the material considered.

In the microwave region, most natural materials have a dielectric constant between 3 and 8, in dry conditions. Water has a high dielectric constant (80) - at least 10 times higher than for dry soil. As a result, a change in moisture content generally provokes a significant change in the dielectric properties of natural materials; increasing moisture is associated with an increased radar reflectivity. The electromagnetic wave penetration is an inverse function of water content. [ESA, 2012]

$$h < \frac{\lambda}{8 \cos \theta} \quad 6.1.3$$

## **6.2 Image segmentation methods**

Many different image-processing techniques and algorithms exist to derive flood area and extent from image data. All of which exploiting the fact that due to its smooth surface, water has significantly lower backscatter intensities than other land cover classes (see, influence of surface roughness on the backscatter signal in section 6.1) There are three main techniques commonly used for water body delineation and remote sensing classification applications.

Firstly, there are the so-called “Histogram Threshold Methods”. There are many different algorithms for histogram thresholding. Statistical attributes of the histogram are analysed to find an optimal threshold to divide the image in two or more classes. In the case of flood detection in grey-scale images there are usually two classes only, water and non-water land-cover. Therefore, as a result one gets a binary image representing the areas covered by water and areas that are dry. The advantage of this technique is that other than the image itself and the grey-level histogram, respectively, no further input is needed. Albeit, there can be



some difficulties, most of the algorithms can only work reliably if the histogram roughly represents a bi-modal distribution, which means the histogram is based on the mixture of two 1-D Gaussian probability distributions and the two distributions are well distinguishable (high contrast). The segmentation result can be heavily biased if the portion of water in an image scene is very small or the contrast between water and non-water surfaces is low. In both cases it is hard to distinguish between two different classes within the histogram. In a best case scenario the histogram will have two lobes, approximately equally sized, and separated by a distinguished “valley”.

The problem of an unequal distribution of water and non-water surfaces within an image scene can be overcome by utilising a so-called split-based approach. In a pre-processing step the image is split into tiles of equal size. Only tiles containing a sufficient portion of both classes and showing a sufficient amount of between-class contrast should be selected. Bazi et al. (2007) propose that a minimum amount of 10% per class is sufficient to achieve a reliable classification result. To identify tiles with both sufficient amounts of each class and sufficient level of contrast a measure for the variability of data within an image has to be introduced (section 8.2).

Another image segmentation technique utilises histogram texture measures to find pixels belonging to a certain class. In the process, it is tried to imitate the functionality of the human visual system, which is able to identify differences in texture even if there are only small differences in contrast. Widely used algorithms rely on simple statistical features of a pixel and its neighbouring area, which are then computed for each pixel using a moving window. For instance, commonly used are the variance and the mean Euclidian distance as texture measures. The problem with histogram texture measures, used as an image segmentation technique, is that a training or reference area is needed to calculate the target statistical criteria. In an automated classification process, as to be implemented in this study, the only possibility for such a training area would be the outcome of another segmentation technique such as histogram thresholding. In the research leading to this study, it was found that in this case the combination of different approach does not significantly improve

classification results. For example, if due to a poor configuration of the image histogram, image thresholding is unable to deliver meaningful results. An incorrect classification would be used, for the definition of histogram texture measure and thus, would again result in an incorrect classification.

Another method which is widely used in water body delineation is segmentation by an active contour model (often called a statistical snake). Active contour modelling uses a dynamic curvilinear contour to search the edge image space until it settles upon image region boundaries, driven by an energy function that is attracted to edge points, so that a “snake” will settle on a line feature even if it is incomplete [M. S. Horrit *et al.*, 2001]. The problems with an active contour model however, are similar to the problems with histogram texture measures. The statistical “snake” requires a source region to start growing from. Again, in an automated classification approach, the use of a technique that requires the manual input of source areas is not feasible. Another drawback is that this technique only works reliably for connected water surfaces; it has been successfully used for coastline delineation. However, in most flooding scenarios there are a lot of unconnected, smaller water surfaces, which would significantly complicate the processing.

Due to the difficulties stated above, the focus in this thesis is set on histogram thresholding techniques. They can be categorized in six groups according to the information they are exploiting [M. Sezgin, B. Sankur, 2004]:

1. Histogram shape-based methods, where, for example, the peaks, valleys and curvatures of the smoothed histogram are analysed.
2. Clustering-based methods, where the grey-level samples are clustered in two parts as background and foreground (object), or alternately are modelled as a mixture of two Gaussians.
3. Entropy-based methods result in algorithms that use the entropy of the foreground and background regions, the cross-entropy between the original and binarized images, etc.

4. Object attribute-based methods search a measure of similarity between the grey-level and the binarized images, such as fuzzy shape similarity, edge coincidence, etc.
5. The spatial methods use higher-order probability distribution and/or correlation between pixels
6. Local methods adapt the threshold value on each pixel to the local image characteristics

#### 6.2.1 Otsu's Algorithm

The Otsu Method is an automated histogram shape-based threshold selection method named after Nobuyuki Otsu. It is assumed that the image to be thresholded contains two classes of pixels or bi-modal histogram (i.e. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal. It is a technique, which has been widely used for SAR image classification.

The so-called Otsu-Method belongs to the category of clustering-based segmentation methods, where the grey-level samples are clustered in two parts as a result a binary image is produced, containing one class for foreground and background, respectively.

The Otsu-Method evaluates the “goodness” of any given grey value to find the optimal threshold for image segmentation. The advantage of this method is that only the grey-level histogram is needed to derive an image threshold without any other a-priori knowledge.  $L$  grey levels represent the pixels of a given image. The number of pixels at level  $i$  is denoted by  $n_i$ . The grey-level histogram is normalized and regarded as probability distribution, as seen in Equation 6.2.1 to Equation 6.2.3.

$$p_i = n_i/N \tag{6.2.1}$$

$$p_i \geq 0 \tag{6.2.2}$$

$$\sum_{i=1}^L p_i = 1 \tag{6.2.3}$$

The pixels are assigned to two classes separated by the threshold-value at level  $k$ . The optimal value for  $k$  is found by maximizing the between-class variance. Therefore the criterion function  $\eta$  is introduced.  $\eta$  is defined in Equation 6.2.4.

$$\eta(k) = \sigma_B^2(k)/\sigma_T^2 \quad 6.2.4$$

With  $\sigma_B^2$  and  $\sigma_T^2$  expressing the between-class variance, and the total variance of levels, respectively. They can be expressed by the equations 6.2.5 to 6.2.6.

$$\sigma_B^2 = \omega_0\omega_1(\mu_1 - \mu_0)^2 \quad 6.2.5$$

$$\sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p_i \quad 6.2.6$$

With  $\omega_0$  and  $\omega_1$  being the probabilities of class occurrence for background and foreground, separated by the threshold value  $k$ .  $\mu_0$  and  $\mu_1$  represent the class mean levels. The total mean grey-level of the original picture is expressed by  $\mu_T$ . Equation 6.2.7 and Equation 6.2.8 defines the probabilities of class occurrence and the class mean levels.

$$\omega(k) = \sum_{i=1}^k p_i \quad 6.2.7$$

$$\mu(k) = \sum_{i=1}^k i p_i \quad 6.2.8$$

### 6.3 Temporal Stability Approach

As discussed in section 6.1, radar backscatter strongly depends on the dielectric constant of the observed object. The dielectric constant is the relative permittivity of a dielectric material. It is an important parameter in characterizing capacitors. Vacuum has, per definition, a dielectric constant of 1. Water has a very high constant of up to 88. Thus, the water content of a certain material (e. g. soil moisture) strongly influences the permeability. Therefore,

there is a strong correlation between surface soil moisture and radar backscatter intensities.

Originally the concept of temporal stability was used for explaining spatio-temporal patterns of surface soil moisture. It was introduced by G. Vachaud et al. (1985) for the purpose of determining regional soil moisture contents by evaluating few in-situ point measurements. It was found that spatial soil moisture patterns tend to persist in time and that therefore soil moisture measured at single in-situ stations actually is often highly correlated with the mean soil moisture content over an area. J. Enin et al. (2000) have shown that there exists a strong large scale spatial correlation of surface soil moisture which has lengths of 400 to 800 km, caused by atmospheric forcing, namely precipitation and evaporation processes. This approximately corresponds to the area which later is going to be used as image subset.

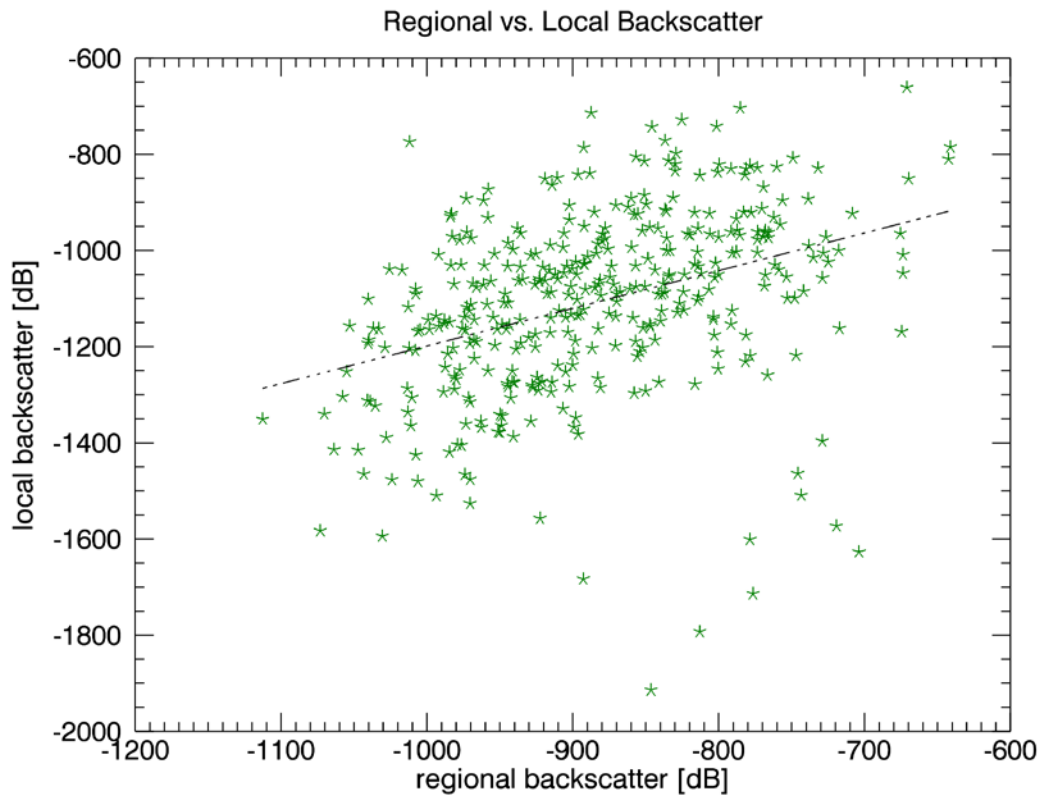
W. Wagner et al. (2008) have shown that radar backscatter, due to the high surface soil moisture dependency, as well shows a strong spatio-temporal correlation between regional and local radar backscatter. In this study, this fact is exploited for the detection of flooding and water bodies, respectively. Therefore ASAR time-series for each individual pixel are going to be analysed.

Radar backscatter over land surfaces is a complex composition of many different influences like sensor parameters or the dielectric and geometric characteristics of land cover [F. T. Ulaby et al., 1982]. However, over bares soil and moderately vegetated terrain soil moisture has a very high influence on radar backscatter measurements at all spatial scales [W. Wagner et al., 2008]. To prove the hypothesis that temporally stable soil moisture leads to temporally stable radar backscatter, Wagner et al. (2008) analysed long term backscatter time series acquired by the Advanced Synthetic Aperture Radar (ASAR). They found that the relationship between regional and local backscatter time series can be modelled linearly which can be formulated as seen in Equation 6.3.1.

$$\sigma_l^0(x, y, t) = a(x, y, \bar{t}) + b(x, y, \bar{t})\sigma_r^0(t) \quad 6.3.1$$

Equation 6.3.1 shows the linear relationship between local ( $\sigma_l^0$ ) and regional ( $\sigma_r^0$ ) backscatter. The relation is expressed in form of a linear equation the so called regression coefficients  $a$  and  $b$ .

Originally, the determination of soil moisture content from a regional measurement is based on the relation formulated above. In this case, the concept of temporal stability will be used for the detection of water bodies and flood areas. If the regional and local values from all pixels in a time series are expressed in a scatter plot, the relation can be modelled by a simple linear regression (least square fit). Usually the relation between regional and local values should be approximately linear, i.e. if the regional backscatter goes down, so does the local backscatter. The hypothesis is the following: If a pixel is covered with water during a flooding event, which isn't usually covered with water, it should have a significantly lower local backscatter then it would have in a dry state (specula reflection over water surfaces [*I. H. Woodhouse, 2006*]) and thus it would be detectable as an outlier from the usual trend. Outliers can be detected by calculating the distance from the line of regression in the fraction of standard deviations, for each date. Pixels covered with water can then be classified by defining a threshold fraction of standard deviations (e.g. if a pixel has a distance of more then -2 standard deviations from the regression line, it is covered with water).



*Figure 6.3.1: Relation between regional and local backscatter with line of regression*

Figure 6.3.1 shows a scatter plot for regional and local backscatter. On the x-axis regional backscatter values are plotted, the y-axis shows local backscatter values. The values in the plot represent the same location in an ASAR WS time series for different acquisition dates. There are a few significant negative outliers. According to the hypothesis, these could be dates on which the pixel was flooded.

### 6.3.1 Regression Analysis

As discussed in the preceding section (section 6.3) the temporal relation of a local radar backscatter value with the regional backscatter value can be modelled linearly. Regression analysis is used for the description and estimation of dependencies between different datasets. The aim is to approximate variations of a certain measure through the variability of one or several different factors. Thus, regression analysis is used to describe functional dependencies.

In our case a linear regression will be used, following the findings of Wagner et al. (2008) (section 6.3). We want to discuss two different approaches of regression analysis. First, the so called simple least squares fit which is the standard and most commonly used method. Most software packages offer an implementation of this method. Alternatively the line of regression can be calculated using a so called total least squares fit. In the case of a simple least squares fit, the  $x$ -values are considered as a fixed quantity and the linear relationship is used to predict the  $y$ -values. Thus, the residuals are measured in parallel to the  $y$ -axis which means (see Figure 6.3.2); they represent the discrepancy between measured and predicted  $y$ -values. In a total least squares fit, both variables are considered uncertain. Therefore the residuals are measured perpendicular to the line of regression.

#### 6.3.1.1 Simple least squares regression

We consider the simple two-dimensional case with  $n$  data points given in the  $\mathbb{R}^2$  plane (see Equation 6.3.2).

$$\{(x_i, y_i) \in \mathbb{R}^2 | i = 0, \dots, n\} \quad 6.3.2$$

As mentioned earlier, in the case of the simple least squares fit, the  $x$ -values are considered as fixed. The  $y$ -values are called ‘target measures’. If we want to describe a linear relationship between two measurements, the line of regression can be described using a basic linear equation (Equation 6.3.3).

$$y(x) = a + bx \quad 6.3.3$$

The goal of the regression analysis is to find the coefficients  $a$  and  $b$  which minimize the squared residuals between the line of regression and the measured  $y$ -values. The sum of the absolute values (instead of the squared) of distances might seem a more appropriate quantity for minimization (the use of squared residuals results in a disproportionately large weighting of outliers). However, the use of the absolute value results in discontinuous derivatives which cannot be treated analytically. The squared deviations from each point are summed and the resulting residual is then minimized to find the best fitting line of regression. The minimization can be formulated as seen in Equation 6.3.4.



$$f(a, b) := \sum_{i=0}^n (y_i - (a + bx_i))^2 \quad 6.3.4$$

The residuals are minimized by calculating the first partial derivative of the function  $f(a, b)$  for both coefficients and setting it equal 0.

$$\frac{\partial(f(a, b)^2)}{\partial a} = 0 \quad 6.3.5$$

$$\frac{\partial(f(a, b)^2)}{\partial b} = 0 \quad 6.3.6$$

This can be interpreted as the search for the line ‘nearest’ to the data points, minimizing vertical distances and making the assumption that model errors in the data are confined to the observer y-coordinates. After partially deriving both parameters, the solution for the fitted parameters  $a$  and  $b$  can be written as seen in Equation 6.3.7 and Equation 6.3.8 (with  $\bar{x}$  and  $\bar{y}$  being the respective mean value).

$$b = \frac{\sum_{i=1}^n (\bar{x} - x_i)(\bar{y} - y_i)}{\sum_{i=1}^n (\bar{x} - x_i)^2} \quad 6.3.7$$

$$a = \bar{y} - b\bar{x} \quad 6.3.8$$

The parameter  $a$  usually known as ‘intercept’, it represents the intersection between y-axis and the line of regression. The parameter  $b$  is called ‘slope’ it determines the inclination of the line of regression.

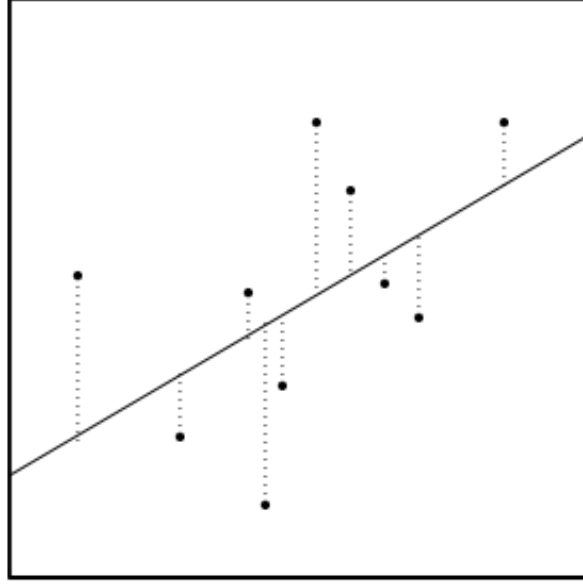


Figure 6.3.2: Graphic to illustrate the measurement of residuals in a simple least squares fit. Residuals represented by dotted lines, data points by solid point

For quality estimation of the fit, following measures are used: standard deviation from the line of regression (Equations 6.3.9, 6.3.10 and 6.3.12), standard error of the coefficients  $a$  and  $b$  (Equations 6.3.11, 6.3.13 and 6.3.14) and the correlation coefficient (Equation 6.3.15) between regional and local backscatter. The latter gives an idea of how well the data can be represented by a linear fit in general. The other factors describe the quality of the actual fit.

$$ss_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2 \quad 6.3.9$$

$$ss_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad 6.3.10$$

$$ss_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 \quad 6.3.11$$

$$\sigma = \sqrt{\frac{ss_{yy} - b * ss_{xy}}{n - 2}} \quad 6.3.12$$

$$SE(a) = \sigma \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{SS_{xx}}} \quad 6.3.13$$

$$SE(b) = \frac{\sigma}{\sqrt{SS_{xx}}} \quad 6.3.14$$

$$r^2 = \frac{SS_{xy}^2}{SS_{xx}SS_{yy}} \quad 6.3.15$$

[Acton, 1959]

### 6.3.1.2 Total least (orthogonal) squares regression

The section above describes the problem of locating a line nearest to a collection of points. This is done by measuring the vertical distances between line of regression and data points. In practice, this technique is commonly used. It provides a fitting function for the independent variable  $X$  that estimates  $y$  for a given  $x$ . Also, it provides a much simpler analytic form for the fitting parameters, than the orthogonal least squares fit does (as seen below). Alternatively, the shorter Euclidean distance can be used to measure the residuals. In the simple least squares fit the  $x$  variable is considered as fixed, there is no variation allowed. The total least squares method allows uncertainties in both  $x$  and  $y$  variables.

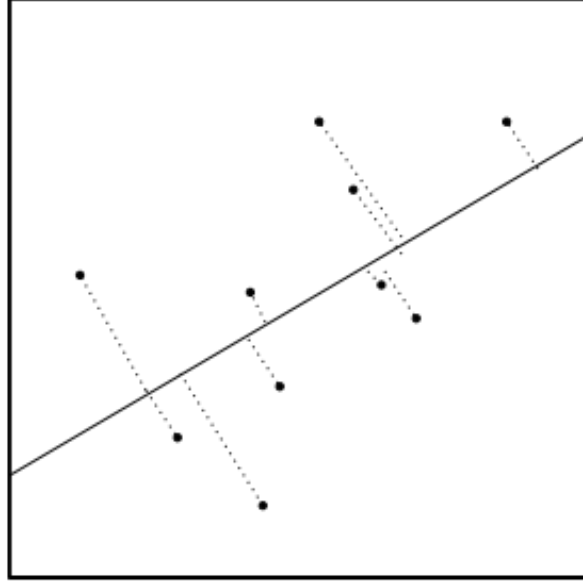


Figure 6.3.3: Schematic representation of a point cloud including the line of regression with residuals measured perpendicularly to the line of regression.

Thus, the total least squares fit is considered as the problem of finding the line  $l$  that minimizes the sum of squares of ‘true’ distances (Equation 6.3.16) [P. de Groen, 1996].

$$f(l) = \sum_{i=1}^n \text{dist}((x_i, y_i), l)^2 \quad 6.3.16$$

The perpendicular distance from a line with the parameters  $a$  and  $b$  to a point  $i$  is given by Equation 6.3.17 [Weisstein, 2012].

$$d_i = \frac{|y_i - (a + bx_i)|}{\sqrt{1 + b^2}} \quad 6.3.17$$

Like in the case of the simple least squares fit, the aim is to find the parameters  $a$  and  $b$  that minimize Equation 6.3.18 which represents the squared, perpendicular residuals from the line of regression. Weisstein (2012) proposes the following solution (Equation 6.3.18 to Equation 6.3.24). The calculation of the parameters is done by setting the respective first partial derivative to 0.

$$f(a, b) \equiv \sum_{i=1}^n \frac{(y_i - (a + bx_i))^2}{1 + b^2} \quad 6.3.18$$

$$\frac{\partial f(a, b)}{\partial a} = \frac{2}{1 + b^2} \sum_{i=1}^n (y_i - (a + bx_i))(-1) = 0 \quad 6.3.19$$

$$\begin{aligned} \frac{\partial f(a, b)}{\partial b} &= \frac{2}{1 + b^2} \sum_{i=1}^n (y_i - (a + bx_i))(-x_i) \\ &\quad + \sum_{i=1}^n \frac{(y_i - (a + bx_i))^2 (-1)(2b)}{(1 + b^2)^2} = 0 \end{aligned} \quad 6.3.20$$

The solution for the first partial derivative for  $a$ , can be solved as seen in Equation 6.3.21.

$$a = \bar{y} - b\bar{x} \quad 6.3.21$$

There is no straight forward solution for the first partial derivative for  $a$ . It can be brought into the form of a quadratic equation and solved with the quadratic equation solution formula (Equation 6.3.22 and Equation 6.3.23).

$$b^2 + \frac{\sum_{i=1}^n y_i^2 - \sum_{i=1}^n x_i^2 + \frac{1}{n}((\sum_{i=1}^n x_i)^2 - (\sum_{i=1}^n y_i)^2)}{\frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i - \sum_{i=1}^n x_i y_i} = 0 \quad 6.3.22$$

$$B \equiv \frac{\frac{1}{2} \left( \sum_{i=1}^n y_i^2 - \frac{1}{n} (\sum_{i=1}^n y_i)^2 \right) - \left( \sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2 \right)}{\frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i - \sum_{i=1}^n x_i y_i} \quad 6.3.23$$

Finally, the solution for  $b$  can be written as follows (Equation 6.3.24).

$$b = -B \pm \sqrt{B^2 + 1} \quad 6.3.24$$

This solution cannot be extended to a higher order polynomial regression [Weisstein, 2012]

## Principal Component Analysis

Alternatively, principal component analysis (PCA) can be used to calculate an orthogonal regression. PCA is mathematically defined as an orthogonal linear transformation, the result of a PCA are the so called eigenvectors. The first eigenvector gives the direction of the highest variance; the second explains the

remaining variance and so on. The goals of a PCA is to extract the most important information from the data table, compress the size of the data set by keeping only the important information, simplify the description of the data set and analyse the structure of the observations and the variables [H. Abdi, 2010]. At the same time the first eigenvector corresponds to the fitted line, minimising the orthogonal distances between the model fit and the measurements. PCA is implemented in most mathematical software packages and thus the application is fairly straight forward.

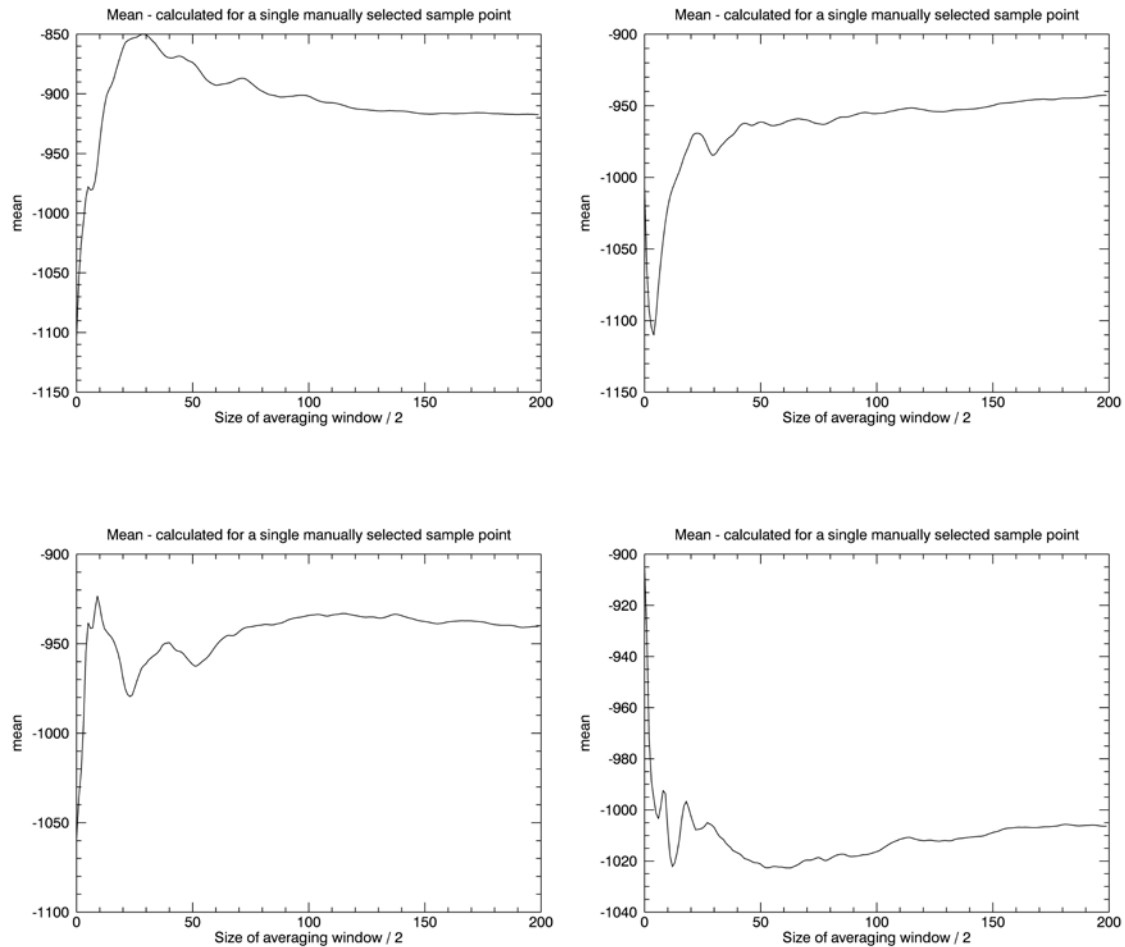
### 6.3.2 Averaging Methods

In the temporal stability approach the regional value of a certain area and acquisition date is compared with individual local values. The local value is always represented by an area of 150 by 150 m which is the size of one ASAR WS pixel. However, there are different methods of calculating the regional value. In this work we want to discuss the following possibilities:

- Calculating a global average. This means that for each date in the time-series there is going to be one regional value for all pixels. In this case, calculated for the entire Severn region (image size: 225 km by 135 km)
- Calculating an average for individual image tiles. This means, for each date there is going to be several regional values, one for each image tile
- Using a moving window for the averaging process. In this case an individual regional value is calculated for each pixel by defining an averaging area around each pixel

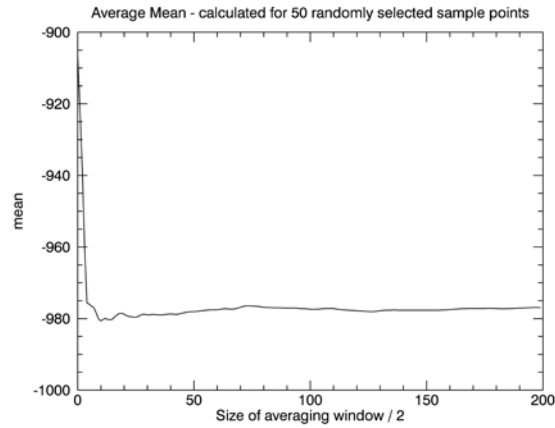
In the case of the averaging of the moving window and the image tiles, an appropriate window size has to be defined.

The regional average should represent the overall regional trend. Therefore, the size has to be chosen large enough that the average value is dominated by this trend and local variations become insignificant. To determine the 'correct' size multiple pixels were selected randomly and the mean was calculated for a growing area around each pixel.



*Figure 6.3.4: Mean value for image-subset of variable size*

The hypothesis is that when the window dimensions reach an optimal size, the average converges towards a stable value – the regional trend is stronger than local irregularities. Figure 6.3.4 shows the results plotted for four randomly selected pixels. Figure 6.3.5 shows the average mean, plotted for 50 randomly selected centre pixels. Both graphics show that at around 100 pixels the average value becomes very stable. Based on these findings we used a 200 by 200 pixel moving window and image tiles, respectively. The window size must not be selected too large to keep the processing costs reasonable.



*Figure 6.3.5: Average mean values for 50 image-subsets of variable size with 50 randomly selected centre points.*

We can also distinguish between the different methods of average computation. The two possibilities we want to discuss are the arithmetic mean (Equation 6.3.25) and the median (Equation 6.3.26). The mean value is the arithmetic average of a set of values. However, for a non-normally distributed dataset, the mean is not necessarily the same as the middle value (median). The median represents the value where exactly 50 % of the values are above and 50 % are below. Thus, the median represents a more robust approximation to the expected value.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad 6.3.25$$

$$P(X \leq m) \geq \frac{1}{2} \text{ and } P(X \geq m) \geq \frac{1}{2} \quad 6.3.26$$



## 7 Data

### 7.1 Datasets

A summary of all datasets used in this study, is given below:

- ENVISAT ASAR WSM Level 1b Data
  - Spatial resolution: 150 m
  - Extent: Severn Region – Latmin: 51°42' N, Lonmin: 1°48' W; Latmax: 52°18' N, Lonmax: 2°48' W
  - Acquisition date: 12/07/2005 to 07/01/2012
  - Total number of images: 380
- SRTM (Shuttle Radar Topography Mission) DEM
  - Spatial resolution: ~90 m
  - Extent: Subset for the Severn Region
  - Acquisition date: 02/2000
- DORIS Precise Orbit State Vectors
  - Precise orbit estimate for the geocoding of ENVISAT ASAR data
- Flood extent marks derived from aerial photography
  - Acquisition date: 23/07/2007

### 7.2 Advanced Synthetic Aperture Radar

The Advanced Synthetic Aperture Radar (ASAR) instrument on board of the ENVISAT satellite extends the mission of the Active Microwave Instrument (AMI) Synthetic Aperture Radar (SAR) instruments flown on the European Remote Sensing (ERS) Satellites ERS-1 and ERS-2. ASAR uses an active phased-array antenna, with incidence angles between 15 and 45 degrees. Applications for this sensor include the study of ocean waves, sea ice extent and motion and land surface studies such as deforestation and desertification, to name a few. [ESA, 2012]

The main technical features are:

- Instrument enhancements that include a digital chirp generator (programmable from 200 kHz to 16 MHz) and an improved linear dynamic range.
- Flexible swath positioning; offering the choice between several image swath positions at various distances from the sub-satellite track, with different incidence angles.
- Dual polarisation; offering horizontal (HH) and vertical (VV) or cross polarisation (HH&HV or VV&VH) operation.
- Wide swath coverage; 405 km swath with 150 m or 1 km spatial resolution.
- Enhanced Wave Mode with images acquired at 100 km interval along-track.
- Extended operating time at high-resolution (30 minutes of operation; 10 minutes in eclipse).
- Global SAR coverage; possible using the solid-state recorder or data relay satellite.

[ESA, 2012]

Envisat flies in a sun-synchronous polar orbit of about 800-km altitude. The repeat cycle of the reference orbit is 35 days, and for most sensors, being wide swath, it provides a complete coverage of the globe within one to three days. The exceptions are the profiling instruments MWR and RA-2 which do not provide real global coverage, but span a tight grid of measurements over the globe. This grid has the same 35-day repeat pattern which has been well established by ERS-1 and ERS-2.

In order to ensure an efficient and optimum use of the system resources and to guarantee the achievement of the mission objectives ENVISAT reference mission operation profiles are established and used for mission and system analyses to define the instrument operational strategies, the command and control, and the data transmission, processing and distribution scenarios.

## **Mission and operation requirements**

- Sun-synchronous polar orbit (SSO): Nominal reference orbit of mean altitude 800 km, 35 days repeat cycle, 10:00 AM mean local solar time (MLST) descending node, 98.55 deg. inclination.
- The orbit is controlled to a maximum deviation of  $\pm 1$  km from ground track and  $\pm 5$  minutes on the equator crossing MLST.
- Recording of payload data over each orbit for low bit rate (4.6 Mps) on tape recorders or solid state recorder (SSR).
- High rate data (ASAR and MERIS) to be accessible by direct telemetry or recording on SSR.

In this report we are using ENVISAT ASAR - Level 1b data in “Wide Swath Mode”, the polarisation is vertical-vertical (VV). ASAR, operating in WSM has a swath width of 406 km the individual datasets have a spatial resolution of up to approximately 75m. The resolution allows the detection of relatively small features. To the level 1b data product, three processing steps are applied automatically in pre-processing:

- Preprocessing: ingest and correct the raw ASAR data
- Doppler Centroid Estimation: estimate the centre frequency of the Doppler spectrum of the data, related to the azimuth beam centre
- Image Formation: process the raw data into an image (using range-Doppler and SPECAN algorithms)

Pre-processing includes a basic data check, in which the data-headers are checked. If the headers can be read correctly the data are decompressed and statistically tested to estimate the bias introduced by receiver electronics. In the next step bias and errors induced by the electronics are corrected. In a last step a replica of the transmitted pulse is extracted and the noise packets are analysed.

### 7.3 Study Area

The study area for this report is the Severn-region on the west-coast of the United Kingdom, between Cardiff and Bristol. With about 354 kilometres the river Severn is the longest River in Great Britain [BBC, 2006].

The region can be divided into two main areas, the River Severn Catchment and the Severn Tidal Tributaries Catchment. The River Severn Catchment covers the entire river until Gloucester (downstream) from where the Severn becomes increasingly tidal influenced.

The overall River Severn catchment (Figure 7.3.1) area covers approximately 11000 km<sup>2</sup>, it has a population of around 2.3 million. 90% of the area is currently used for agriculture. However, there also are a number of major urban settlements including Worcester, Tewkesbury, Gloucester and Warwick, to name only a few. The upper Severn catchment is predominantly hilly, enclosed by the Cambrian Mountains on the western edge. Further downstream, at the eastern edge of the Severn Uplands, the topography becomes flatter, with its widest part around the Severn-Vyrnwy confluence. From here the catchment can be divided into four topographically distinct areas. These are; the relatively flat Shropshire Plain to the north and west, the Kerry Hills and Shropshire Hills to the southwest (the Teme sub-catchment), the undulating and wooded Midlands Plateau to the centre (including Ironbridge Gorge) and the broader and flatter floodplain of the Severn towards Worcester. [Environment Agency UK, 2009]

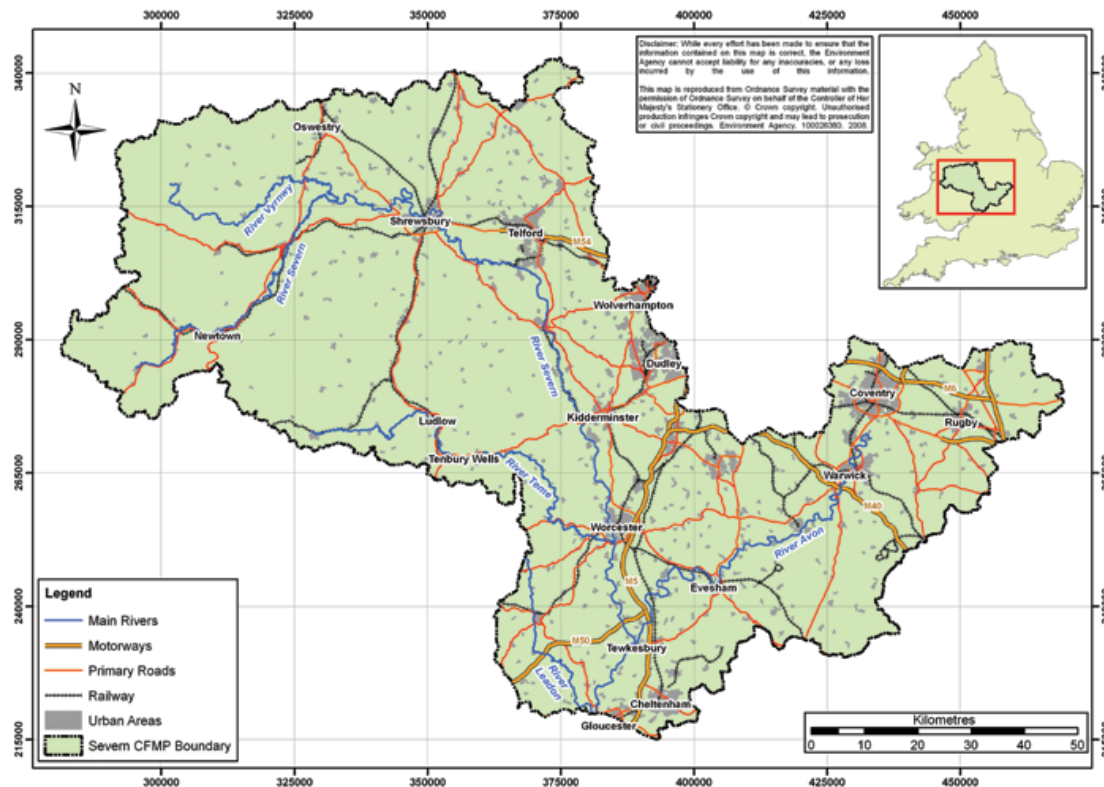


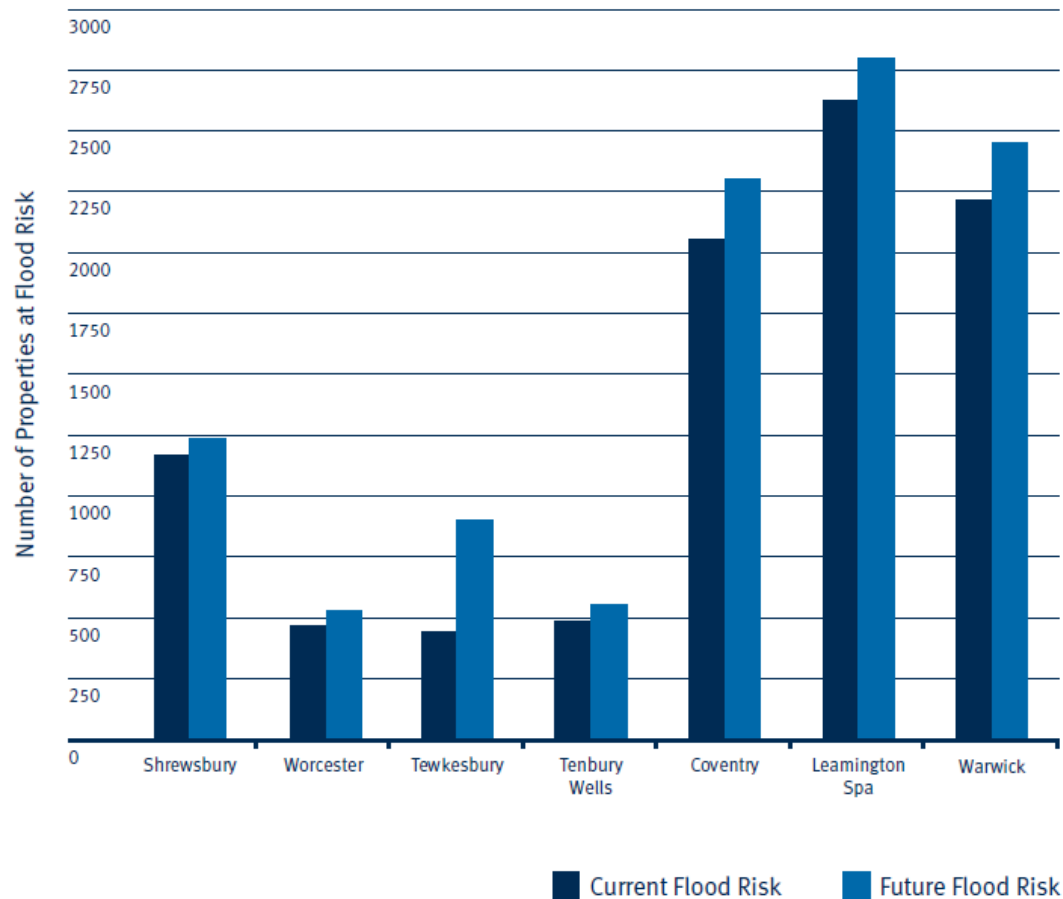
Figure 7.3.1: Map of the River Severn Catchment (northern part)<sup>1</sup>

The Severn Tidal Tributaries (Figure 7.3.3) is largely rural too, with the exception of Gloucester in the north of the catchment and Avonmouth in the south. Approximately 8% of the catchment is classified as urban and the entire area has a population of approximately 313000. The catchment covers an area of over 1000 km<sup>2</sup> which can be divided into six sub-catchments each draining into the tidal River Severn. [Environment Agency UK, 2009]

The flood risk usually is split up into two components: the chance or probability of a particular flood and the impact or consequence that the flood would have if it occurred. The probability of a flood is typically expressed as the likelihood of the flood occurring in one year. The likelihood is given as a percentage. For example, a 1% flood has a 1% chance of occurring in any one year. In the River Severn catchment area there are currently around 60,000 people and 29,000 commercial and residential properties which would be affected from a 1% annual probability river flood. Therefore, 2.5% of the total population of that area are at risk from flooding. [Environment Agency UK, 2009]

<sup>1</sup> Environment Agency United Kingdom: <http://publications.environment-agency.gov.uk/PDF/GEMI0909BQYM-B-E.pdf>

The most significant event in recent years took place in June/July 2007 after a period of exceptional rainfall which is the event we investigated for the purpose of automated water body classification.



*Figure 7.3.2: Current and future (2100) flood risk to property from a 1% annual probability river flood<sup>2</sup>*

According to the report by the Environment Agency United Kingdom (2009), an increase of severe flooding events has to be expected in the future, due to climate change. In a scenario with a 20% increase in peak flow in all watercourses, until 2100, the number of people affected by a 1% annual probability river flood would increase to approximately 68,000 and the number of properties to around 33,000 (Figure 7.3.2)

<sup>2</sup> Environment Agency United Kingdom: <http://publications.environment-agency.gov.uk/PDF/GEMI0909BQYM-B-E.pdf>

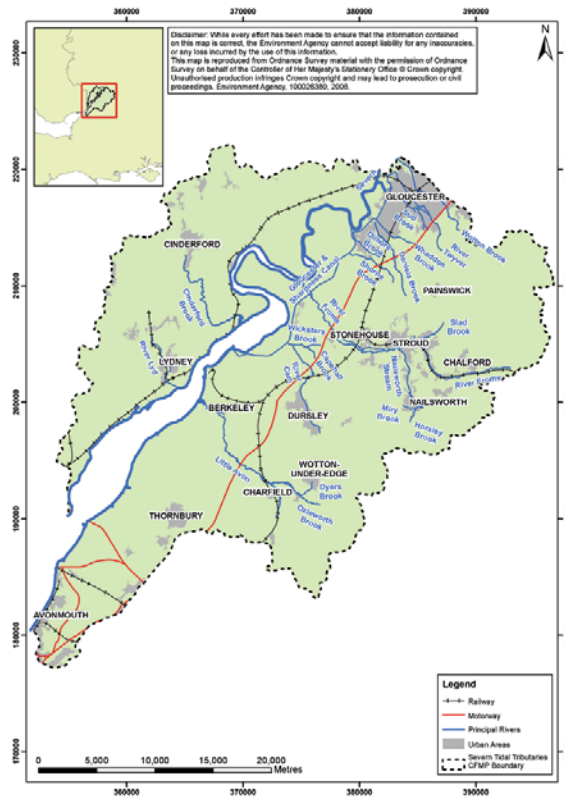
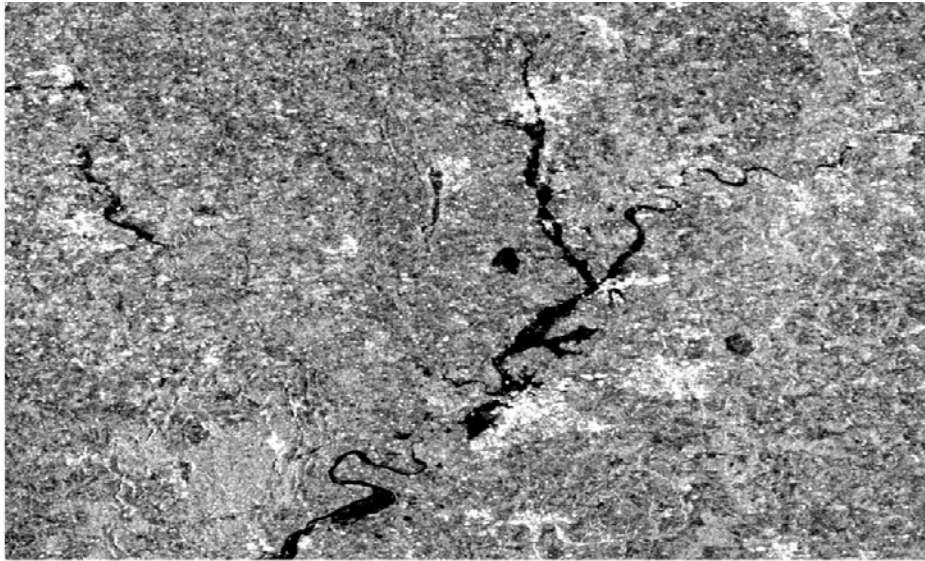


Figure 7.3.3: Map of the tidal River Severn Catchment (southern part)<sup>3</sup>

The water body classification in this work was based on a normalised ASAR WSM time series which covers an approximately 69km by 67km area around the River Severn region. The geographic extent of which is: upper left corner – 52° 17' 59"N, 2°47'59"W; lower right corner – 51°41'59"N, 1°48'14"W. The time-series contains recordings within the area of interest which were taken between the 12<sup>th</sup> of July 2005 and the 7<sup>th</sup> of January 2012, a total of 380 datasets. The water body classification itself will be based on the image from 23<sup>rd</sup> of July 2007, this falls into the time of the severe flooding from 2007 (Figure 7.3.4).

<sup>3</sup> Environment Agency United Kingdom: <http://publications.environment-agency.gov.uk/PDF/GEMI0909BQYN-E-E.pdf>



*Figure 7.3.4: Normalised ASAR WSM image from 23<sup>rd</sup> of July, 2007*

## **8 Processing**

Most of the processing was carried out with Exelis' Interactive Data Language (IDL). This is a programming language, widely used for image processing in the remote sensing community. The geocoding is performed with the Next ESA SAR Toolbox (NEST), which is an open software tool that bundles functionality for SAR processing. NEST was developed by the Canadian software developing company ARRAY. Finally, for the quality assessment, R, a programming language for statistical computing was used.

### **8.1 Pre-Processing**

Figure 8.1.1 presents the processing chain for ASAR WS raw data. In the first step ARRAY'S NEST software is used for the range-doppler geo-rectification. Then the data is resampled, normalised and stored, as time-series, in a global grid from which the data can be extracted as an image stack, representing a certain area of interest.



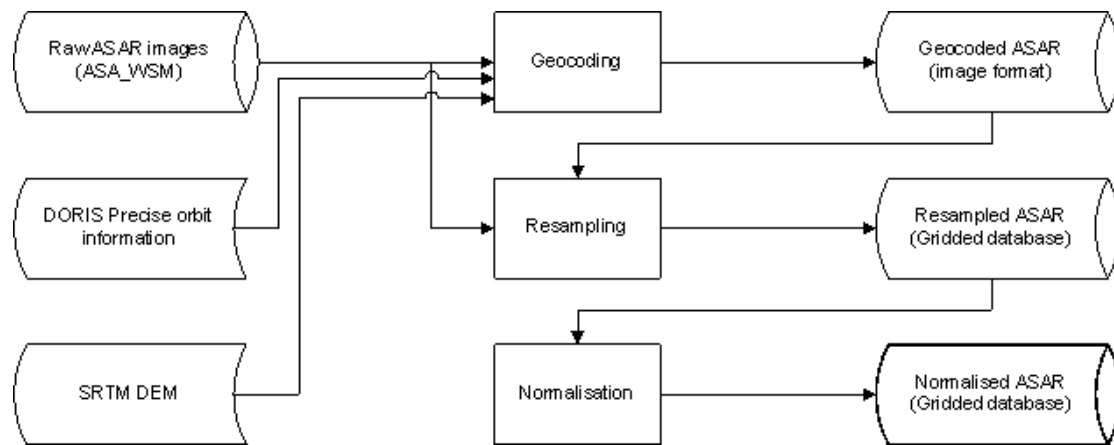


Figure 8.1.1: Processing chain for ASAR WS raw data

### 8.1.1 Geo-Rectification

The side-looking viewing geometry of synthetic aperture radar systems causes a number of geometric and radiometric distortions (Figure 8.1.2). The first distorting factor is due to the fact that in the imaging process not the ground range but the slant range is measured. Thus, image data not directly in the sensors nadir position will have some distortions; the further away the larger will be the difference between slant range and ground range. This source of distortion is rather straightforward to compensate, since it can be described by a small number of variables. Some side-looking radar systems incorporate a correction within the radar's signal processing itself.

Since the earth's surface usually differs from an entirely flat plane, further geometric distortions will occur as a result of the imaging method. One can distinguish between two different effects, which can be induced by different terrain shapes. These phenomena are called *layover* and *shadowing*. Layover arises from the fact that the pulse delay time is used to determine the across track coordinate of a scatterer. In the case of a mountain viewed from the side; the mountain slope facing the radar sensor will appear shortened while the opposite slope will appear stretched. An uncorrected image of an area of high relief has a characteristically appearance in which the mountains appear to lean towards the radar. Additionally, the number of scatterers per unit length, measured on the sensor facing mountain slope, is higher than usual. Thus, this side of the mountain will appear unusually bright. On the other hand, the slope on the opposite side of the mountain has a lower number of scatterers per unit

length. Thus, it will appear darker. This phenomenon is often called *low-lighting* and it is purely geometrical. Layover can be corrected if the surface geometry of the observed region is known, e.g. by including a digital terrain model in the geocoding process [Woodhouse, 2006].

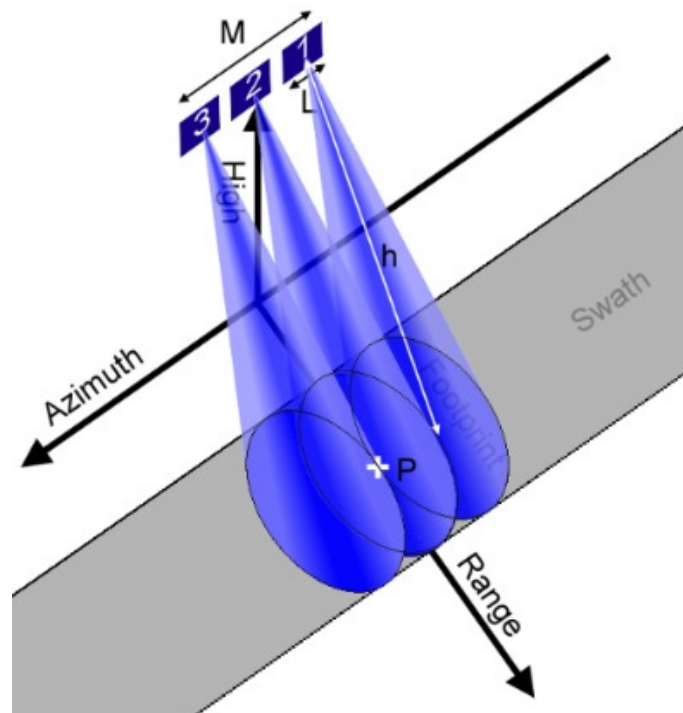


Figure 8.1.2: Synthetic Aperture Radar viewing geometry<sup>4</sup>

In areas with a very high relief, e.g. very high mountains with particularly low valleys, the phenomenon of shadowing occurs, which means that one part of the surface is hidden from the radar's view. The obscured surface area receives no illumination at all, and consequently no signal is returned from it. Scattering of microwave radiation from the atmosphere into the shadowed region is entirely negligible. Shadowing cannot be compensated in any processing step, since for the shadowed areas no information is available. In the radar image this regions will appear completely black [Woodhouse, 2006].

Geometric distortions, as described above, can be corrected by a terrain correction (the Range Doppler Terrain Correction) process. Some effects, like the

<sup>4</sup> Dantor (2005):  
<http://de.wikipedia.org/w/index.php?title=Datei:Sarresrp.jpg&filetimestamp=20120209134912#filelinks>

shadowing of pixels are irreversible. For this thesis the software toolbox released by the European Space Agency (Next ESA SAR Toolbox – NEST) was used for the terrain correction process.

The *Range Doppler Terrain Correction Operator* implements the backward geocoding approach for geocoding SAR images from single 2D raster radar geometry. It uses available orbit state vector information in the metadata or external precise orbit (in this work DORIS orbit files were used) to determine the exact platform location in space. Also it utilizes the radar timing annotations and the slant to ground range conversion parameters together with the reference DEM data to derive the precise geo-location information. As a reference surface used the DEM from NASA's 'Shuttle Radar Topography Mission' (SRTM).

For each pixel two unknowns have to be determined, the range distance and the azimuth distance. The azimuth can be determined by the Doppler shift history in frequency. The position of each object on the ground is then localized relative to the nadir line (by the range distance) and relative to the broadside by the azimuth distance. Combined with the parameters from the DORIS precise orbit files, it is now possible to determine the exact platform position and therefore the position of each image pixel, projected onto the reference DEM, without the use of tie points.

#### **8.1.2 Resampling**

To improve the possibilities of both temporal and spatial analysis, all images are resampled to a fixed grid and stored in a database structure. Based on the resolution of the ASAR WS data, a global grid with a sampling interval of 2.4 arc-seconds, corresponding to the distance of about 75 m (half the resolution of the ASAR WS data) at the equator, was defined.

#### **8.1.3 Normalisation**

Radar backscatter in general, shows a very strong dependency on the local incidence angle (specular reflection, corner reflectors, etc.). This means, images recorded from different orbits, and thus different viewing angles, look different. To be able to perform temporal, time-series based analysis and compare images

from different acquisitions, the angular dependency of backscatter values has to be normalized. This is done in the normalization step.

For each 2.4 arc second location, a linear model is fitted to the backscatter measurements as a function of local incidence angle  $\theta$ . All radar backscatter measurements are then adjusted to a reference incidence angle of  $30^\circ$  [Sabel, 2012]. In Equation 8.1.1, the radar backscatter before normalisation is given by  $\sigma^0(\theta)$ ,  $\sigma^0(30)$  is the normalised backscatter.

$$\sigma^0(30) = \sigma^0(\theta) - a(\theta - 30) \quad 8.1.1$$

## 8.2 Otsu Method

In this section we want to discuss the implementation of the Otsu algorithm, which was used for auto thresholding. In Figure 8.2.1, the algorithm is described schematically with the help of a flow chart.

### 8.2.1 Image pre-processing for the Otsu Method

To optimise the input image data for an optimal auto grey-level thresholding, the following processing steps were performed.

- 1.) Normalisation of the image histogram: The original grey-level histogram has to be normalized to contain only positive values. In this case the image will be transformed to contain values from 0 to 255 only.
- 2.) To overcome the problem of an uneven distribution of the classes water and no-water within the original dataset, the image is divided into overlapping sub-images – each having a dimension of 100 by 100 pixels. The aim is to select sub-images containing a sufficient proportion of both classes only. Martinis et al. (2009) introduced a measure for the contrast within an image (Equation 8.2.1), on which the selection of appropriate sub-images was based. Only the selected parts were used for further processing and histogram thresholding.

$$C_{VX} = \left| \frac{\mu_W}{\sigma_W} \right|, R_{XN} = \left| \frac{\mu_W}{\mu_T} \right| \quad 8.2.1$$

- 3.) Only sub-images with a  $C_{VX}$  value greater or equal to 0.7 and with an  $R_{XN}$  value between 0.4 and 0.9 [Martinitis et al., 2009] are selected for further processing.
- 4.) For each sub-dataset that qualifies for processing an individual threshold is calculated. In the end, the final threshold is derived from the mean threshold-value.
- 5.) Application of a so-called Lee-Filter. The purpose of which is to reduce speckle noise in the radar image. Speckle noise in SAR is a multiplicative noise; it is caused by coherent processing of backscattered signals from multiple distributed targets [B. Tso and P. Mather, 2009]. The Lee-Filter belongs to the group of adaptive speckle filters.
- 6.) In this step a Canny-Edge filter is applied on the image. The Canny-Edge filter detects edges within the image. The areas around edges often contain a large number of mixed pixels, which can distort the thresholding result. To minimize this effect, a buffer zone is calculated around the detected edges. A mask, consisting of these zones, is applied to ignore respective regions for further processing. [H. Liu and K. C. Jezek, 2004]

### 8.2.2 Otsu Algorithm

In the following section there is a step-by-step description of the grey-level thresholding algorithm itself, based on Otsu's method. The only input needed is the pre-processed radar image and the grey-level histogram, respectively.

- 1.) Computing the image histogram and the probability density function
- 2.) Iterating through all possible thresholds and computing the between class variance. Using the between class variance the selection criterion can be determined (section 6.2.1)
- 3.) In the first iteration the calculated criterion is selected as the maximum and as the temporary threshold-value.

- 4.) The newly calculated criterion function value is compared with the previous maximum, in each iteration step.
- 5.) In case the new value is higher then the old, the old will be replaced and the corresponding threshold value is selected as a temporary solution
- 6.) After iterating through the complete range of grey-scale values the grey-level, maximising the criterion function, will be found.

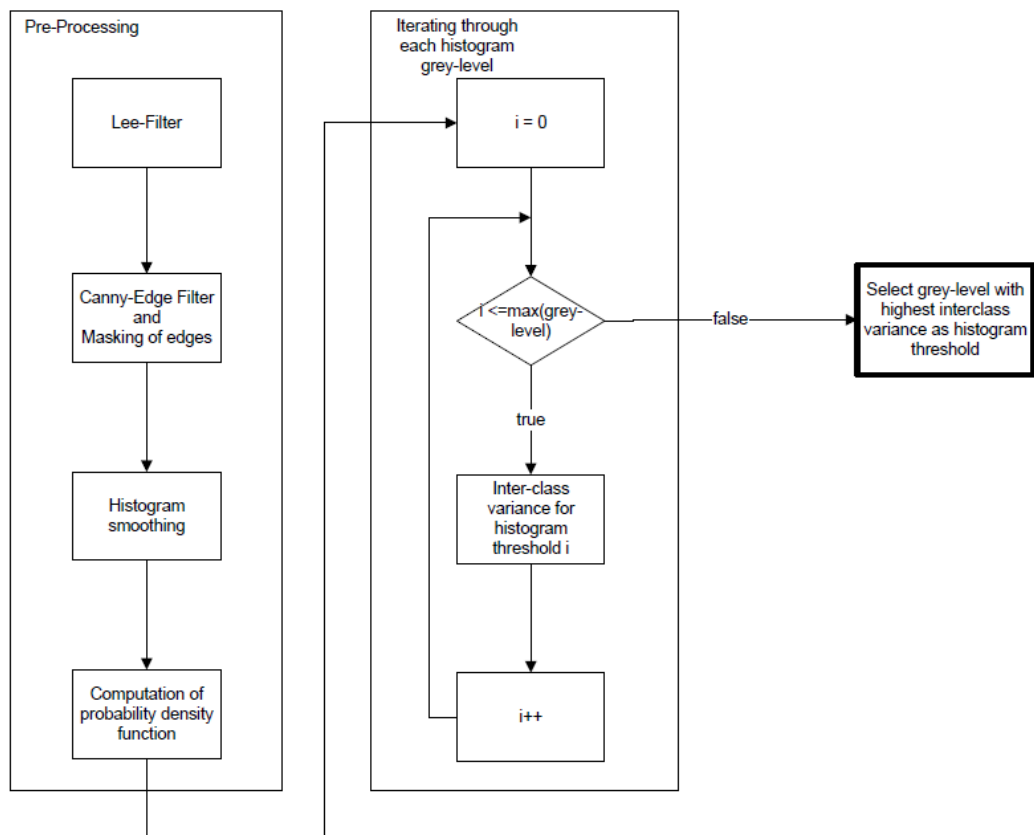


Figure 8.2.1: Flow chart for Otsu's Method

### 8.3 Temporal stability approach

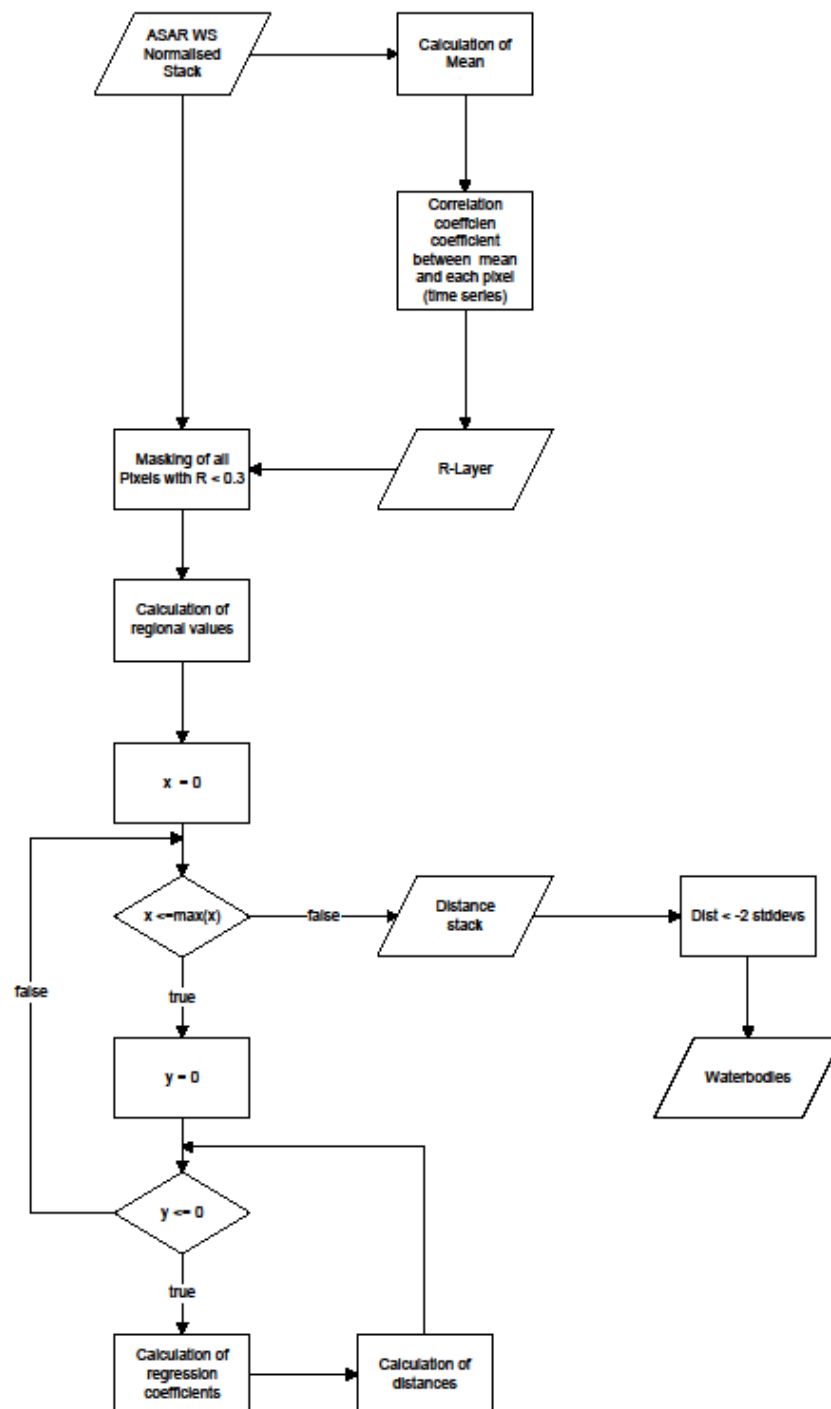


Figure 8.3.1 schematic illustration of the temporal stability approach

Figure 8.3.1 shows a schematic illustration of the temporal stability approach algorithm.

Other than the Otsu method, which requires a single radar image only, the temporal stability approach uses the ASAR WS normalised stack, which has gone through the processing steps described in section 8.1, as an input. Then the following steps are executed for water body delineation.

1. Calculation of a mean (regional) value for each image in the stack
2. Computation of the correlation coefficient between the time series of each pixel and the time series of regional values
3. The correlation coefficient for each pixel was stored as the so called 'R-Layer' [W. Wagner *et al*, 2008]
4. From the R-Layer and image mask was derived, covering all the pixels with a correlation coefficient of 0.3 or smaller
5. Based on the masked ASAR WS normalised stack, the regional values, for each image, were calculated
6. Iterating through each pixel, the regression coefficients were calculated, for the regional/local distribution.
7. Based on the regression coefficients from Step 6, for each data point, the distance to the line of regression could be calculated. The distance was measured as the number of standard deviations
8. The output of Step 7 was stored in an image stack where for each pixel the distance to the line of regression is given as the number of standard deviations
9. In the final step, pixels with  $\leq -2$  standard deviations ( $-3$  in the case of the orthogonal regression) were classified as water bodies.

#### **8.4 Post classification filtering**

In a classification result, there are a number of small, most likely wrongly classified areas (in our case water bodies), which are usually caused by measurement errors or noise. To delete these areas, a filtering routine was applied to the final classification. The routine considers both the area and the value of pixel segments.



A region growing algorithm is used to calculate the size of each segment. Segments with a size smaller or equal to 5 pixels will be deleted. Additionally the average value of a segment is taken into account, following the hypothesis that a higher deviation from the line of regression also increases the probability that a certain pixel is actually covered with water. In the initial classification process, all pixels with a standard deviation of -2 or higher were selected as water pixels. In the post classification, the average value of a segment is considered. For example: If a water segment, smaller or equal to 5 pixels, is detected, the mean value is calculated. If the average standard deviation is -4 or higher, the segment will not be omitted.

The flowchart in Figure 8.4.1 illustrates schematically the functionality of the post-classification algorithm. It filters small segments of water. The same filtering routine is applied to all non-water areas to decrease the amount of noise within the classification areas. This means, the same process is applied again with the filtering criterion of standard deviation  $> -2$  (first decision in the flowchart).

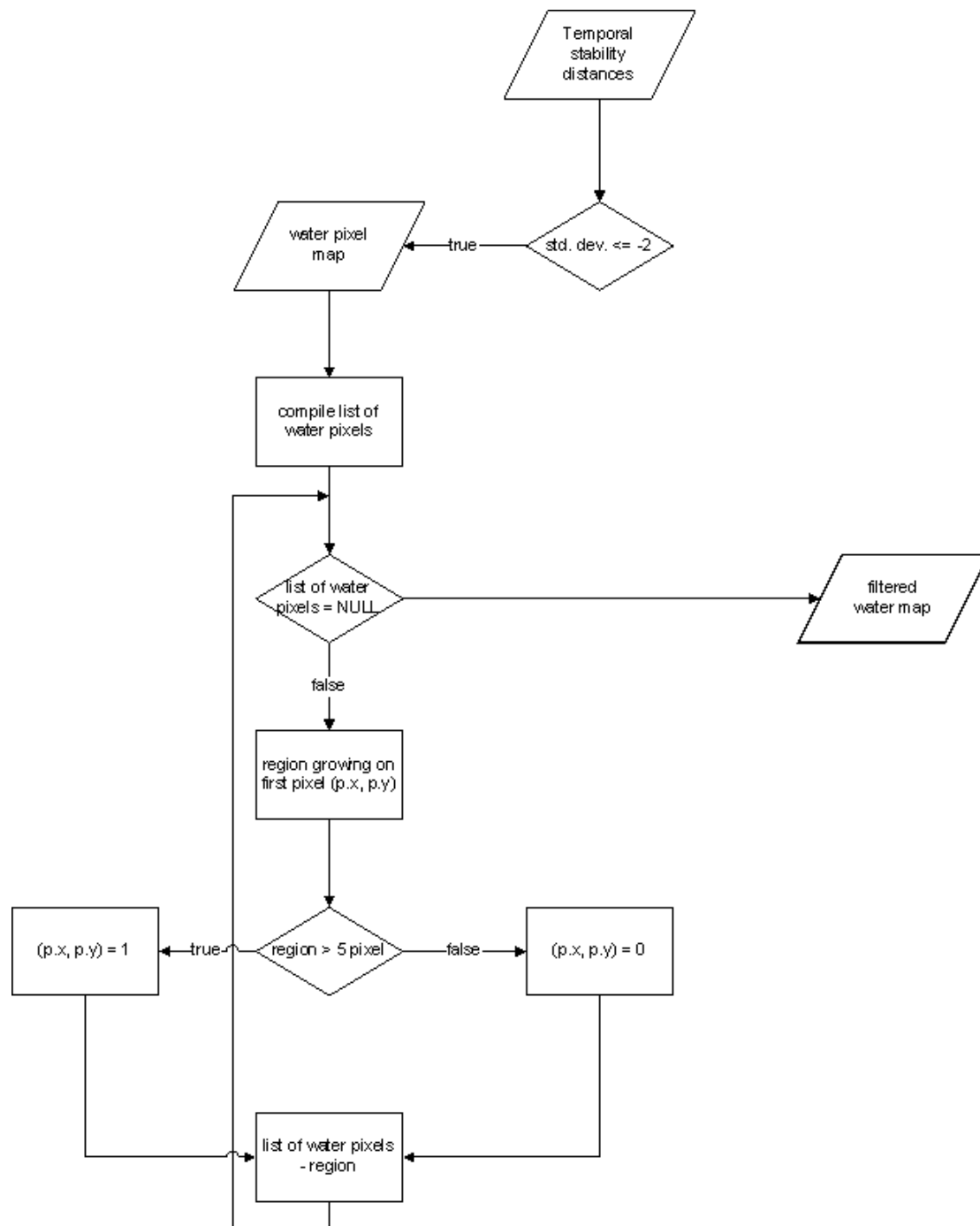


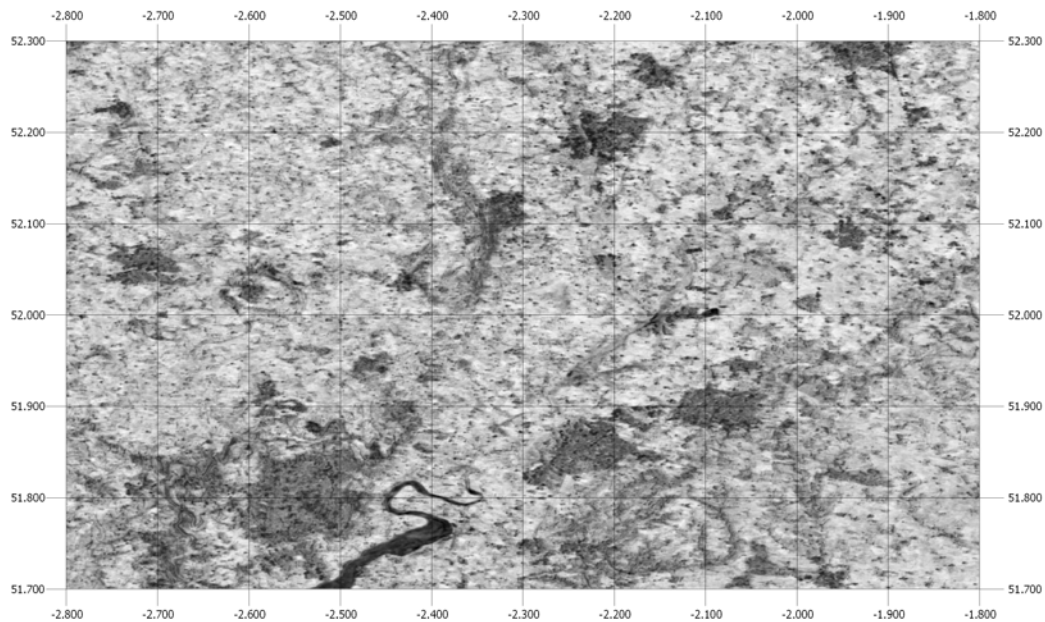
Figure 8.4.1: Flowchart for the illustration of the post-classification algorithm

## 9 Results

As a basis for the water body classification the ASAR WS evening scene from the 23 of July, 2007 was used. Each of the classification approaches was applied to the same scene.

## 9.1 R-Layer

The R-Layer should help to identify areas, like forests or very dense shrub land, urban areas or permanent water bodies, where no reasonable classification is possible or in case of permanent water bodies are not prone to flooding. The R-Layer is generated by calculating the correlation coefficient between the time series of a single pixel value (local value) and the regional value [W. Wagner, 2008]. Theoretically, this method replaces the need for a-priori knowledge in terms of land cover. By increasing the overall correlation between regional and local backscatter and therefore the ability to detect outliers, the accuracy of the results from the temporal stability approach can be improved.



*Figure 9.1.1: Subset of the R-Layer of the flooding area. Pixels with low correlation appear dark pixels with high correlation appear brighter*

Figure 9.1.1 shows a visual representation of the R-Layer for the river Severn area, Figure 9.1.5 shows the corresponding area in an optical satellite image, acquired from Google Earth. These images show how the R-Layer highlights differences between some land cover types. For example, the two dark areas in the lower centre and the dark area in the top centre of the R-Layer correspond nicely to the urban areas of Gloucester, Cheltenham and Worcester. According to W. Wagner et al. (2008), the correlation is generally very high over agricultural

areas and other sparsely vegetated terrain with values up to about 0.8. The values decrease with increasing vegetation density and becomes smaller than 0.2 over dense forests and urban areas. Following these findings, the masking threshold was chosen at 0.3. In processing scenarios which are using the R-Layer, pixels with a correlation below that value were ignored.

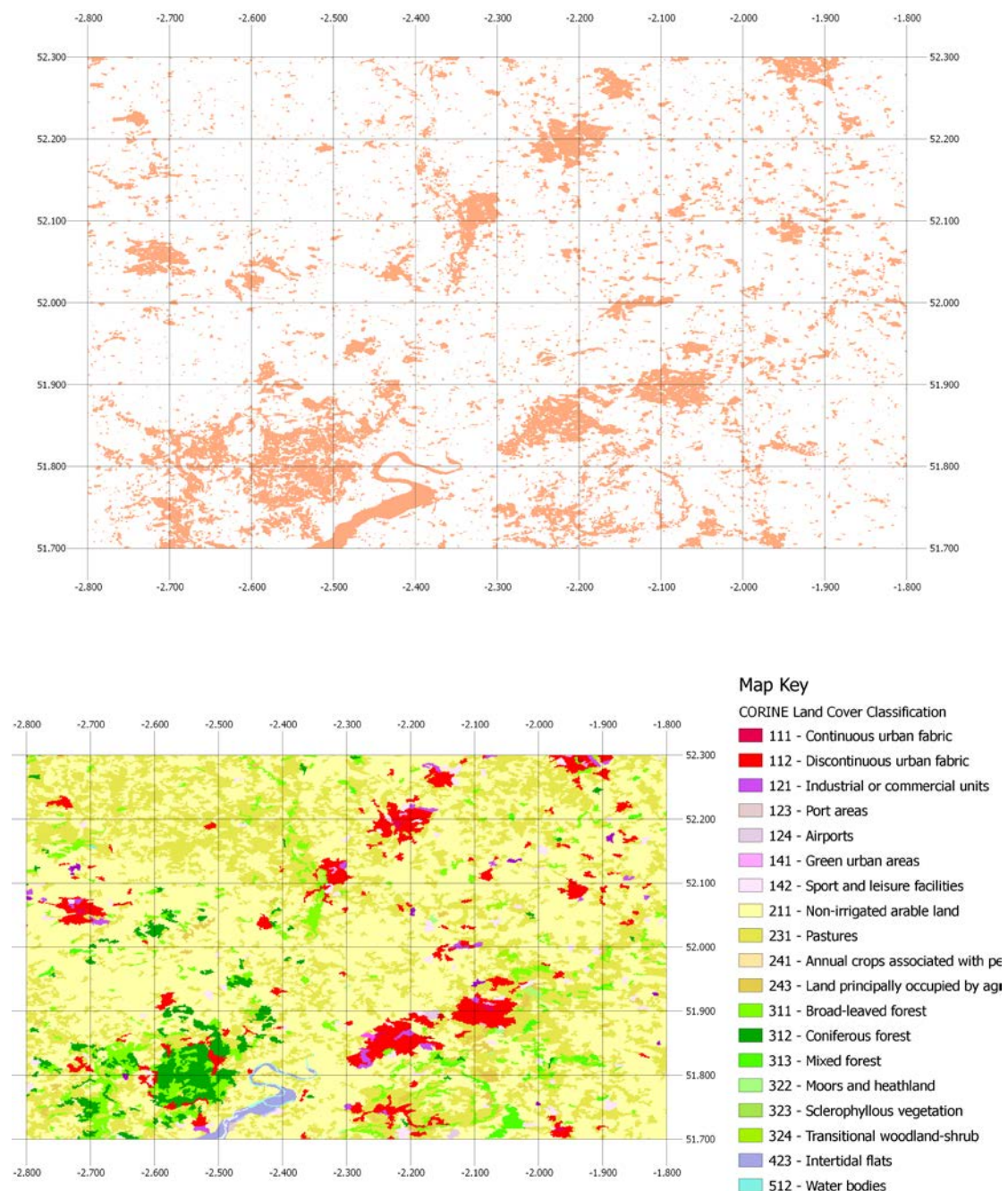
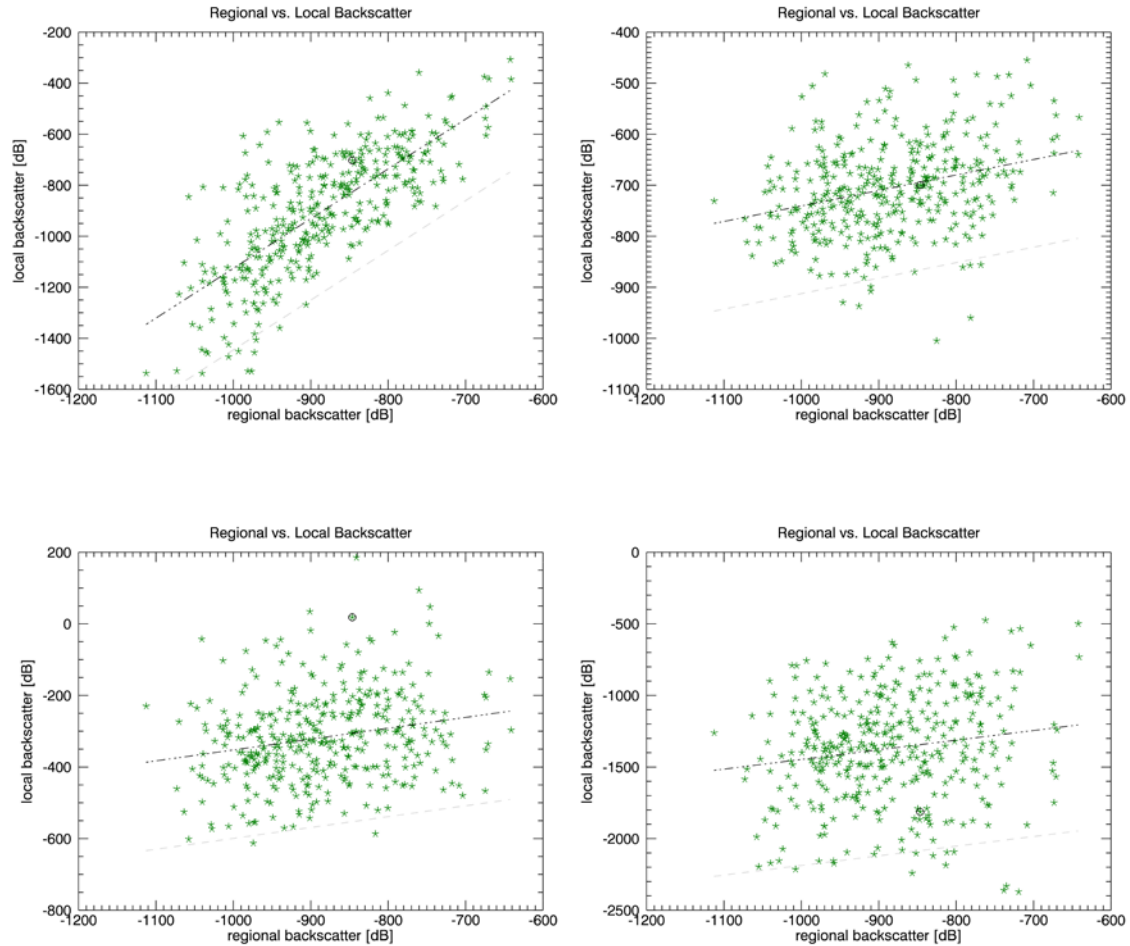


Figure 9.1.2: top – subset of the R-Layer for the flooding area. Pixels with an R-Value of below 0.3 are coloured in pink; bottom – CORINE land cover classification for the River Severn area

The relations between R-Layer and urban or forest areas respectively, clearly shows in Figure 9.1.2, where all the pixels with a correlation value below 0.3 are coloured in orange.



*Figure 9.1.3: scatter plots between regional and local backscatter values, regional values were calculated as global mean for the entire image. Top-left: values for a pixel representing the class agricultural; top-right: values from a forest-pixel; bottom-left: values from an urban pixel; bottom-right: values from a permanent water body pixel*

Figure 9.1.3 illustrates the correlation differences between pixels of different classes. The class “agricultural-areas” shows a significantly narrower scattering than the other classes. The classes urban, forest and permanent water bodies show a very high scattering, illustrated also by the large 2-standard deviation distance (grey-line).

**Correlation coefficients:**

- Agricultural: 0.743
- Forest: 0.306
- Permanent water bodies: 0.163
- Urban: 0.218

The numbers above support the conclusions from the visual interpretation in Figure 9.1.3. Also they correspond to the numbers given by W. Wagner et al. (2008) in their report. The low correlation is caused by a temporally relatively static backscatter of the classes “forest”, “urban” and “permanent water bodies”. Unlike in the case of open agricultural areas the backscatter is not influenced by soil moisture and thus doesn’t follow the regional trend - hence the low correlation. The backscatter from “forest-areas” is mainly influenced by properties of the tree canopy, only very little radiation can penetrate through to the ground. In urban areas, the surface is sealed and covered mainly by artificial structures – the backscatter has almost no seasonal component at all. Corner reflectors and specular reflection dominate the backscatter signal. There is a high viewing angle dependency. Backscatter on water surfaces is generally very low (specular reflection) it can be influenced by wind and waves. In these land-cover classes no reasonable classification of flooding will be possible. The correlation coefficient can be used to automatically mask those pixels. By excluding pixels with a very low correlation, the overall correlation increases.



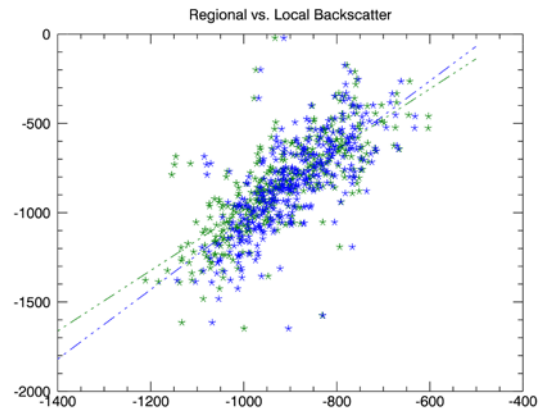


Figure 9.1.4: regional versus local backscatter: blue – all pixels were included for the calculation of the regional values; green – pixels with a correlation coefficient with a correlation of 0.3 or less were omitted

The influence of omitting low correlation data, on the line of regression, can be seen in Figure 9.1.4.

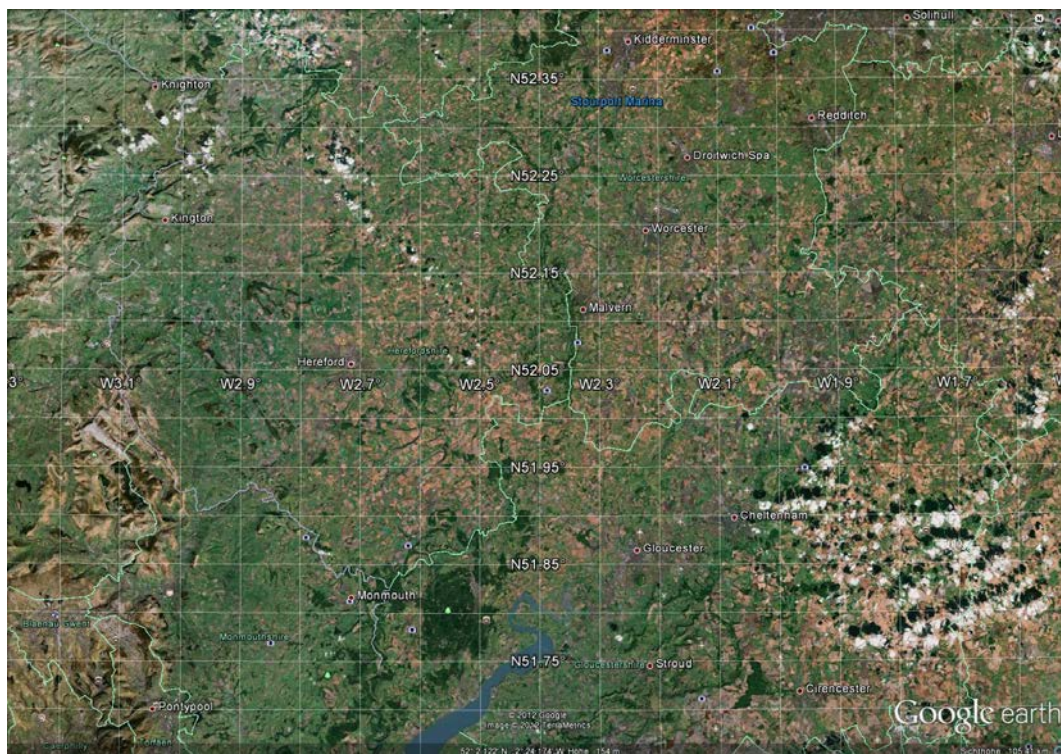


Figure 9.1.5: Google Earth optical satellite imagery. The red rectangle corresponds to the same subset as can be seen in Figure 9.1.5

## 9.2 Averaging results

As discussed in section 6.3.2, there are several different ways to calculate the regional value, which is required for the temporal stability approach. Due to the very high processing costs of the moving window averaging method (section 6.3.2), we chose a subset which covers the area affected by the main river Severn flood only (Figure 9.1.2). In case of the global mean, there is only one value for each image i.e. acquisition date. If average values are calculated with the image tiling or the moving window method, there are multiple values for each date.

Figure 9.2.1 shows the time-series for regional values, calculated with different averaging methods. The plots and the images (Figure 9.2.2 to Figure 9.2.3), respectively show that there are slight differences between the different averaging methods as well as between the results with and without the application of the correlation layer.

The small differences between the results of the different averaging methods do influence the distances, calculated for the temporal stability approach.



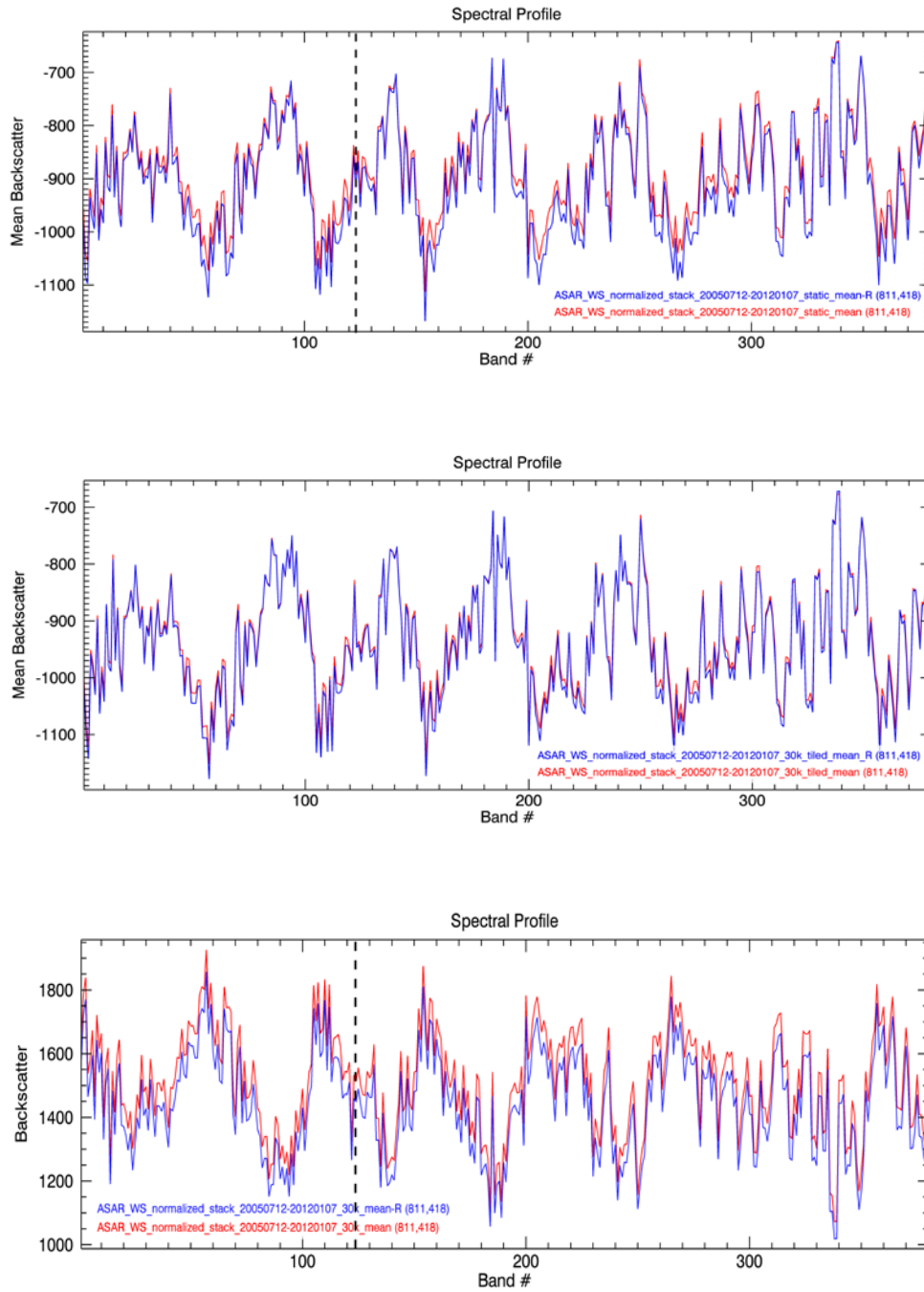
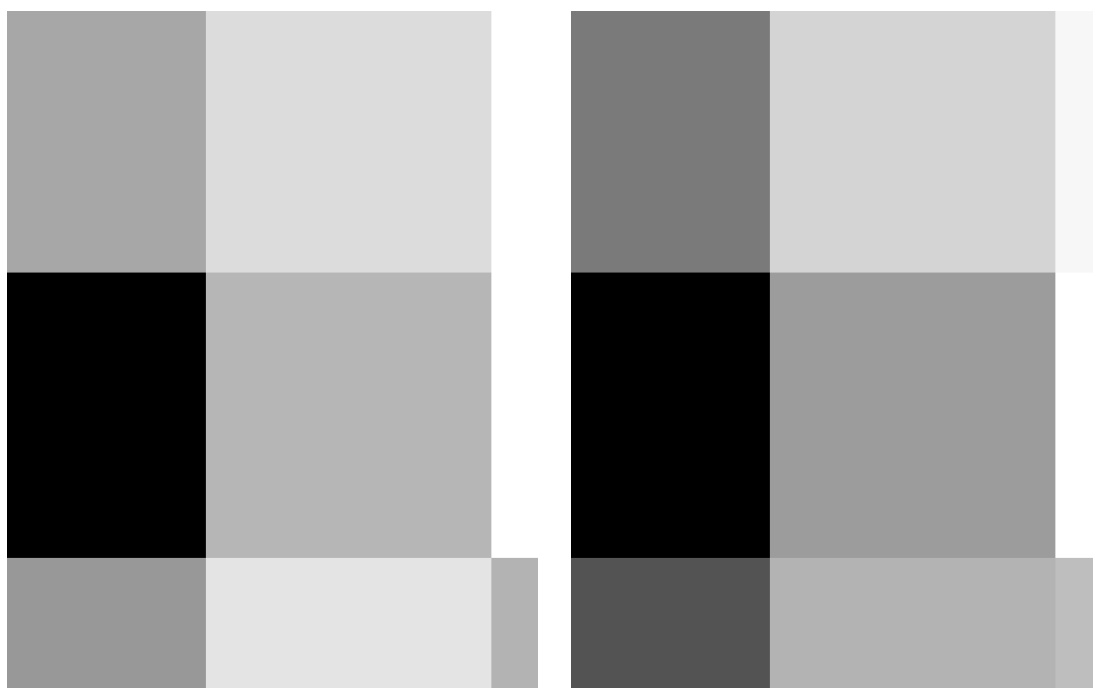
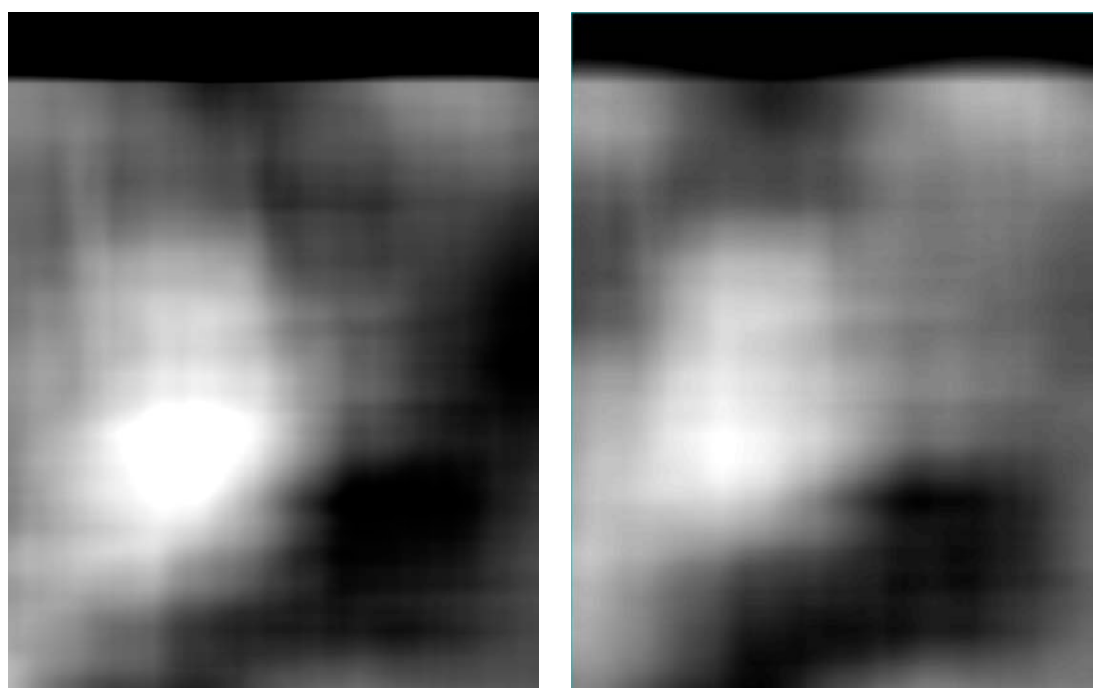


Figure 9.2.1: Time series of the regional value, for one sample pixel. Top – global mean, centre – regional value for image tiles, bottom – moving window. The red line represents the mean calculated over all pixels, for the calculation of the blue line pixels with a correlation coefficient of 0.3 or smaller were omitted. The vertical dashed line indicates the location of the image from the 23 of July 2007.



*Figure 9.2.2: left – mean values derived from individual image tiles; right – mean values derived from individual image tiles, incorporating the correlation layer mask*

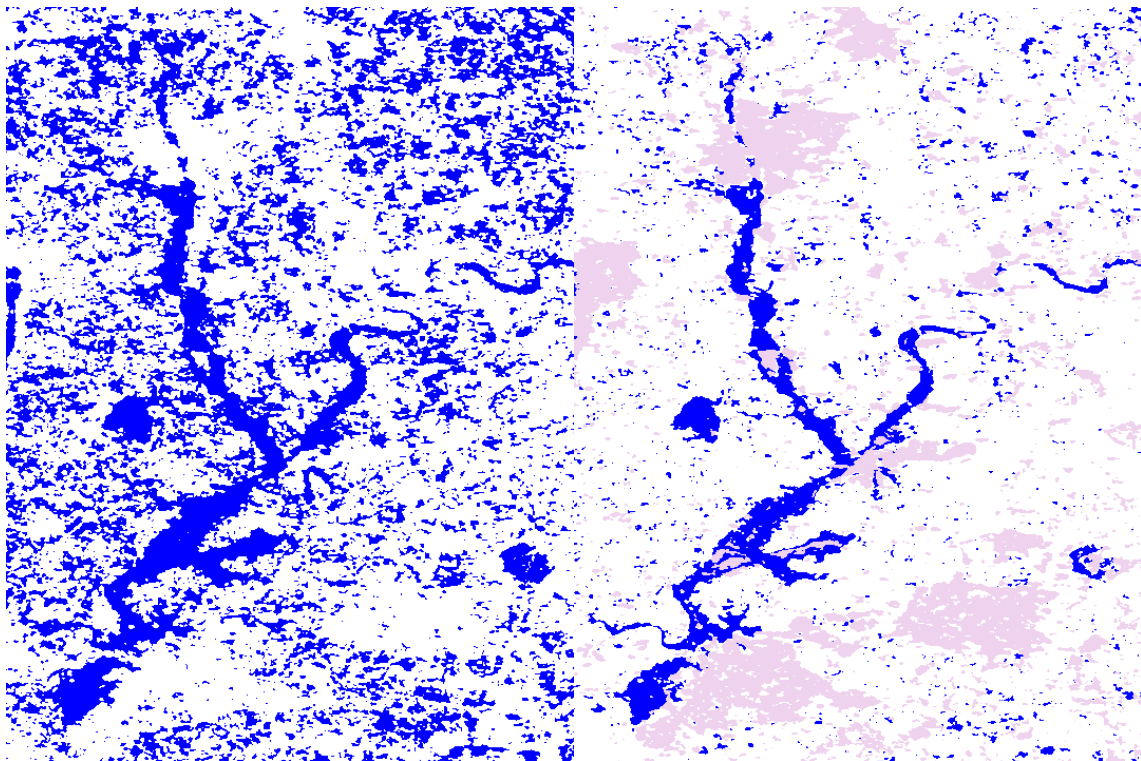


*Figure 9.2.3: Mean values calculated for individual pixel. Left – all pixels were used to calculate the average; right – pixels with a correlation of 0.3 or smaller were masked*

### 9.3 Otsu's Method

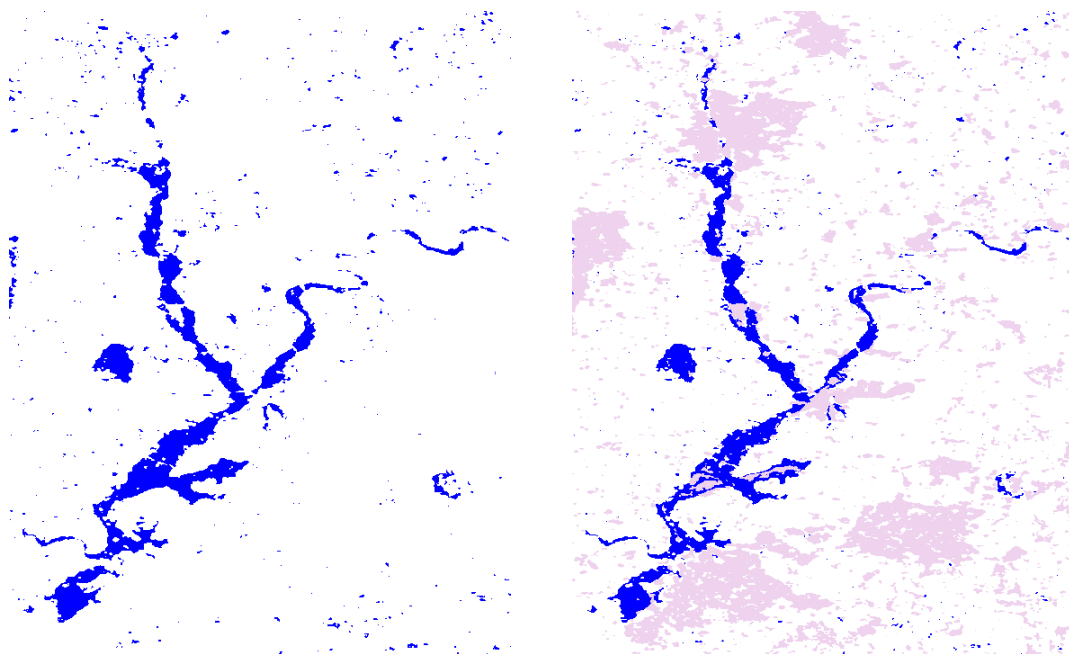
As we discussed before there are several different approaches to the Otsu Method:

- Application of the algorithm to the entire radar scene
- Application to the entire scene, masking the areas with R-layer values of below 0.3 which means omitting pixels from permanent water bodies, urban areas and forests.
- Application of a pre-selection routine (section 8.2.1), to find image subsets with a high probability of containing both water and non-water surfaces
- Application of the same pre-selection routine as above. Additionally, pixels with a value below 0.3, from the R-Layer, are masked which means omitting pixels from permanent water bodies, urban areas and forests.



*Figure 9.3.1: Left: No pre-selection algorithm applied to the scene; right: There was no pre-selection applied to the scene. Pixels with R-Layer values of below 0.3 were omitted.*

In the images above (Figure 9.3.1) the results for the Otsu classification without the pre-selection routine are shown. Blue pixels represent flooding areas, no water pixels are white and areas with a correlation value of below 0.3 are shown in pink. There is a significant improvement between the image on the left (without inclusion of the R-Layer to the processing routine) and the image on the right (with inclusion of the R-Layer to the processing routine). The Otsu method is very sensitive to the distribution of water and non-water areas within the area of interest. There are hardly any permanent water bodies in this scene, i.e. in this particular case omitting pixels with a correlation of below 0.3 increases the amount of water pixels, compared to non-water areas. Urban areas, forests and dense shrub works are masked out, for the most part, which causes a more equal distribution of water and non- water areas. In a hypothetical situation with an area without any urban areas or dense vegetation but with larger areas of permanent water bodies, the exact opposite would be the case. The amount of water pixels, compared to non-water pixels, would be decrease which could cause a worse classification result. The inclusion of the R-Layer improves the classification result in the case of the River Severn region, in different locations the exact opposite could be the case.



*Figure 9.3.2: The pre-selection routine was applied to both images above. Left: without inclusion of the R-Layer. Right: Pixels with a correlation coefficient of below 0.3 were omitted, in the classification process*

The pre-selection routine as described in section 8.2.1 was applied to the images in Figure 9.3.2 before applying the Otsu method. There is a significant improvement between these results and the results from Figure 9.3.1. The pre-selection routine selects image subsets which are likely to contain water and non-water surfaces, this is done by selecting subsets based on a measure for the image variance. This method decreases the sensitivity to an uneven distribution between water and non-water pixels. However, it can not distinguish between variance due to water and non-water pixels and the variance due to urban areas or dense vegetation, which can be similarly high. In this case the inclusion of the R-Layer improves the classification result because image tiles with a high variation, due to urban areas or vegetation, can not be selected as a classification basis (those areas are masked out).

If we again consider a hypothetical case, of an image scene without urban areas or dense vegetation but a large amount of permanent water bodies, the number of detected image subsets suitable for the Otsu classification would decrease through inclusion of the R-Layer. This means, the effect on the classification result would be negative. However, the effect would not be as significant as in the case of the basic approach (without the pre-selection routine).

The application of a pre-selection routine improves the robustness of the classification in general (independent of the land cover composition of the area of interest). In some cases, further improvement can be achieved, by the application of the correlation layer mask.

In this specific case the best result was achieved with the application of the pre-selection routine and the R-Layer. The final resulting threshold is at -12.6651 dB.

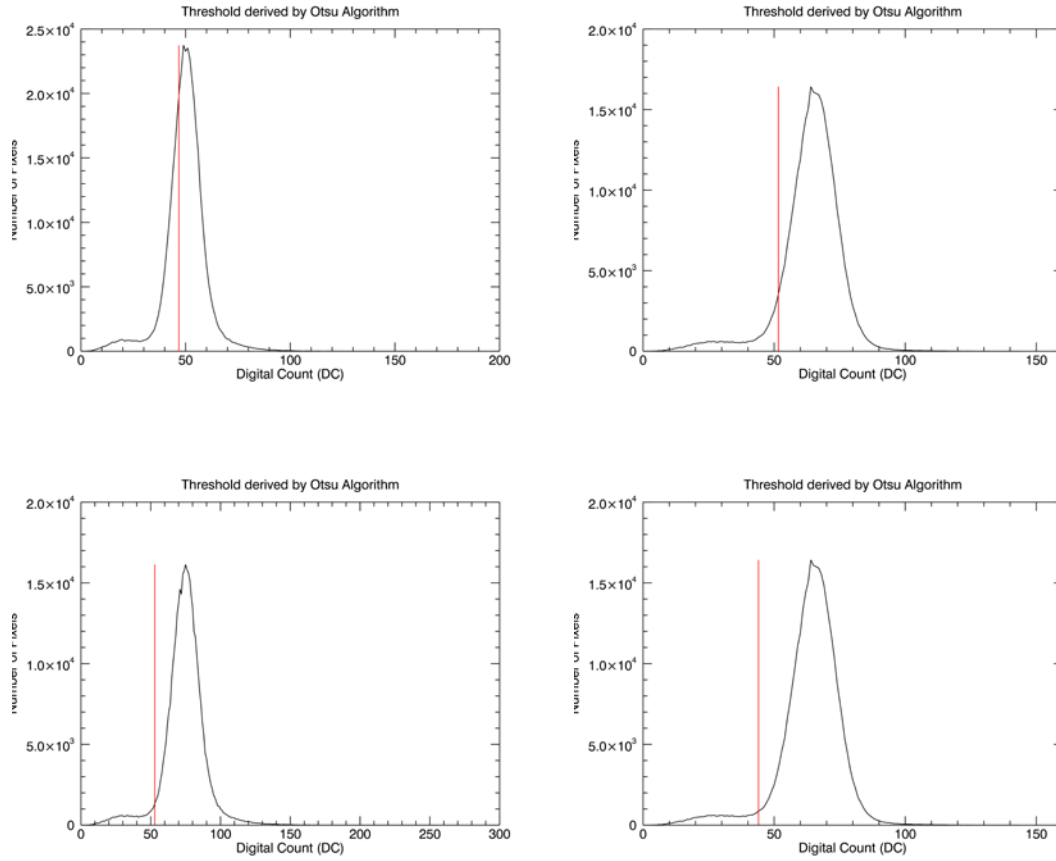


Figure 9.3.3: Histogram plot with auto-generated threshold for all the cases mentioned above – top-left: no pre-selection routine, no correlation layer; top-right: no pre-selection routine, with correlation layer; bottom-left: with pre-selection routine, no-correlation layer; bottom-right: with pre-selection routine, with correlation layer

The plots in Figure 9.3.3 show the grey-level histograms of the radar image, it illustrates the separation between water and no-water pixels in a bi-modal distribution. The number of no-water pixels is significantly higher, thus the right lobe of the distribution is considerably taller. Figure 9.3.3 supports the findings from the classification that the ‘best’ result can be reached with the Otsu method combined with the pre-selection routine and the correlation layer.

## 9.4 Temporal Stability Approach

### 9.4.1 Comparison of different averaging areas

For the computation of the temporal stability method, a regional value is required. As discussed before there are several possibilities of how to calculate that value, each method caused a slightly different classification outcome. In the

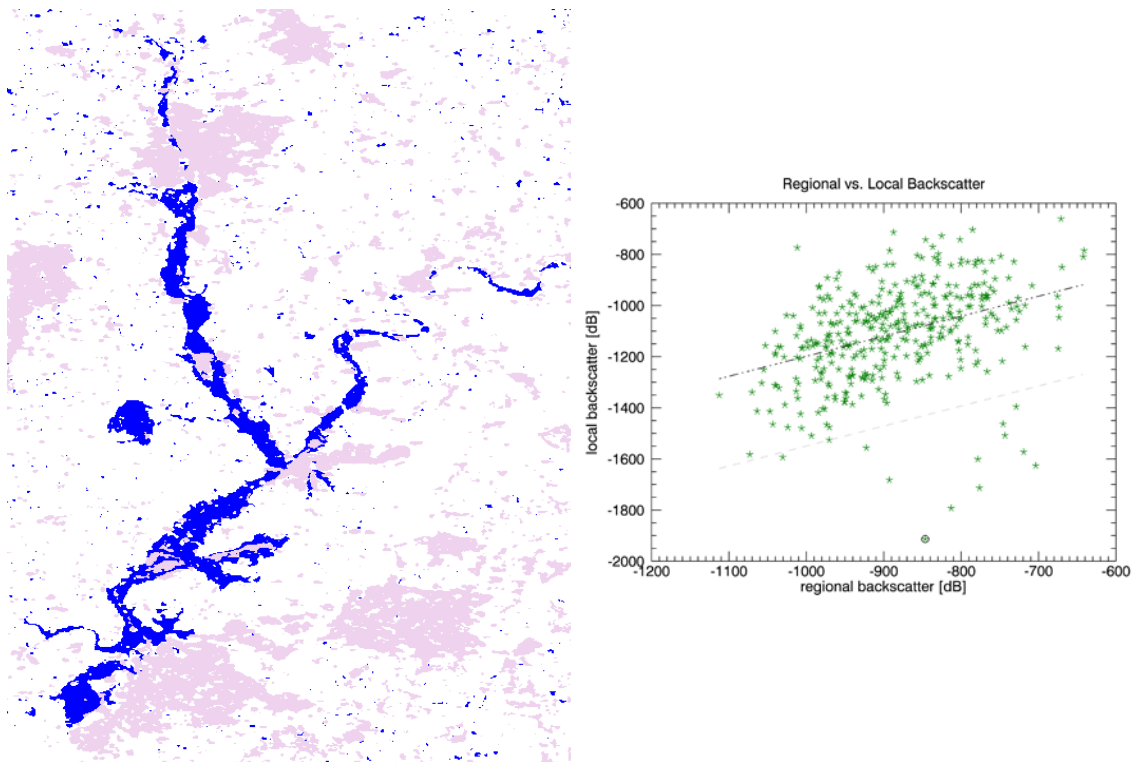
first case the mean was used as an average value and a simple least squares regression was used. This section should demonstrate the differences between different averaging areas. Moreover, the results will be compared to those obtained by the application of the median, instead of the mean, and the application of a total least squares fit, instead of a simple least squares fit.

The following averaging methods were applied (see section 9.2):

- Global mean (one mean value for each date)
- Global mean, pixels with a correlation coefficient of less than 0.3 were masked
- Calculation of the mean for image tiles of an area of approximately 30 by 30 km
- Calculation of the mean for image tiles of an area of approximately 30 by 30 km, pixels with a correlation coefficient of less than 0.3 were masked
- Calculation of the mean over a moving window with an approximate size of 30 by 30 km
- Calculation of the mean over a moving window with an approximate size of 30 by 30 km, pixel with a correlation coefficient of less than 0.3 were masked

The following figures are all based on the water body classification of the ASAR WS evening scene from the 23<sup>rd</sup> of July, 2007.

## Global mean

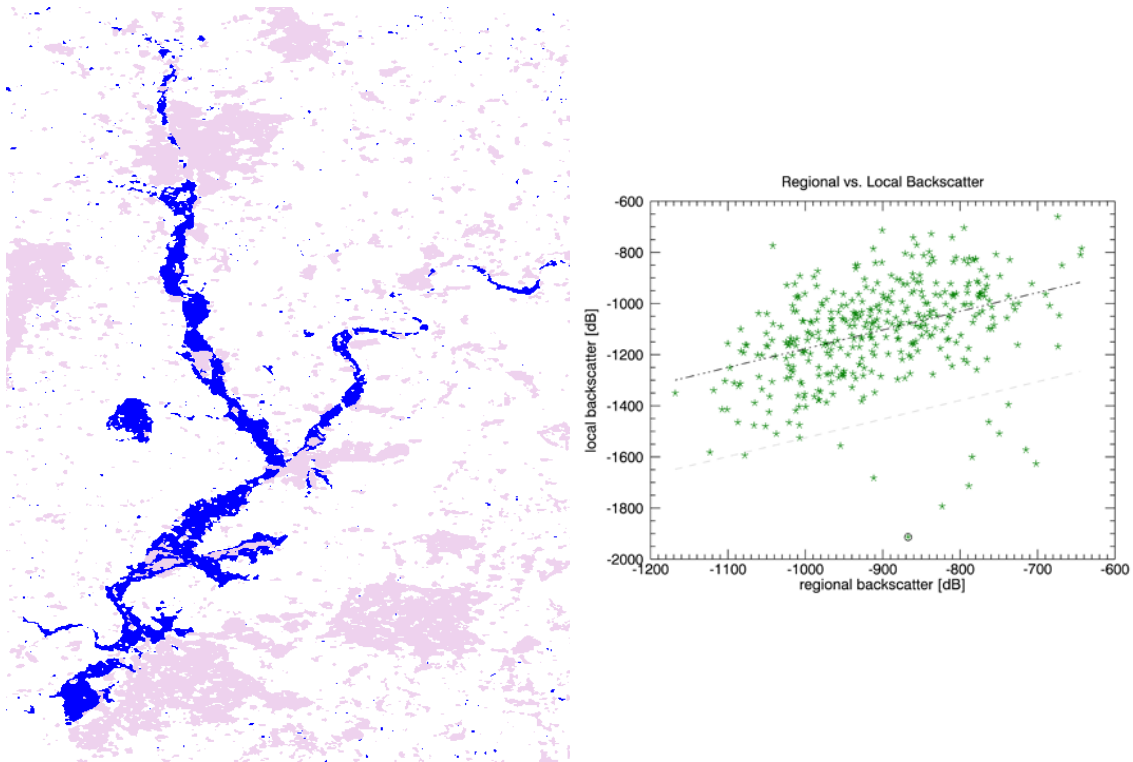


*Figure 9.4.1: Left – classification results using a global mean; right – scatter plot between regional backscatter and local backscatter for the time series of one pixel*

Figure 9.4.1 shows the classification of the temporal stability distances. Pixels with a standard deviation, from the line of regression, equal or smaller then -2 are coloured in blue, they represent water bodies or flooding, respectively. White pixels represent non-water surfaces and pink are all pixels with a correlation coefficient smaller then 0.3. However, the correlation layer was not regarded in this averaging process. On the right side one can see the scatter plot between local and regional backscatter. The dots represent the backscatter values of one single pixel for the entire time series. The line of regression is coloured black; the grey line represents the regression line minus two standard deviations. All pixels below that line were classified as water. The circled data point represents the pixel values from the 23<sup>rd</sup> of July 2007, the date of the flooding, which means that this particular pixel was flooded at that time.



## Global mean including the correlation layer



*Figure 9.4.2: classification results using a global mean, pixels with a correlation coefficient smaller than 0.3 were masked. Left – classification result; right – scatter plot of regional versus local backscatter*

Figure 9.4.2 shows the classification result using a global mean. Unlike in the first case, pixels with a correlation coefficient below 0.3 were masked in the averaging process. The same colour scheme as in Figure 9.4.1 was used. There is a significant improvement of the classification result between Figure 9.4.1 and Figure 9.4.2. There is less noise in the result.

Initial State				
		background	water	no-water
Final State	background	0.000	0.005	0.000
	water	0.000	87.940	0.006
	no-water	0.000	12.055	99.994
	class changes	0.000	12.060	0.006

Table 9.4.1: Differences between both global mean approaches

### Mean for image tiles

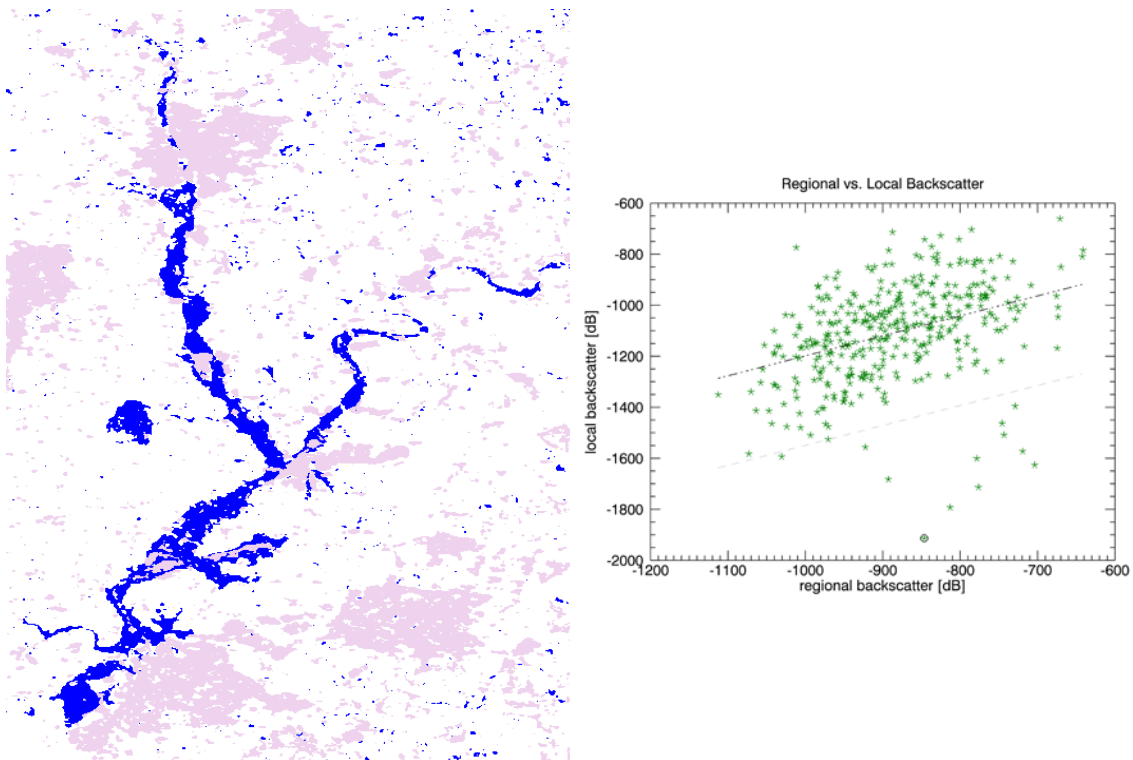


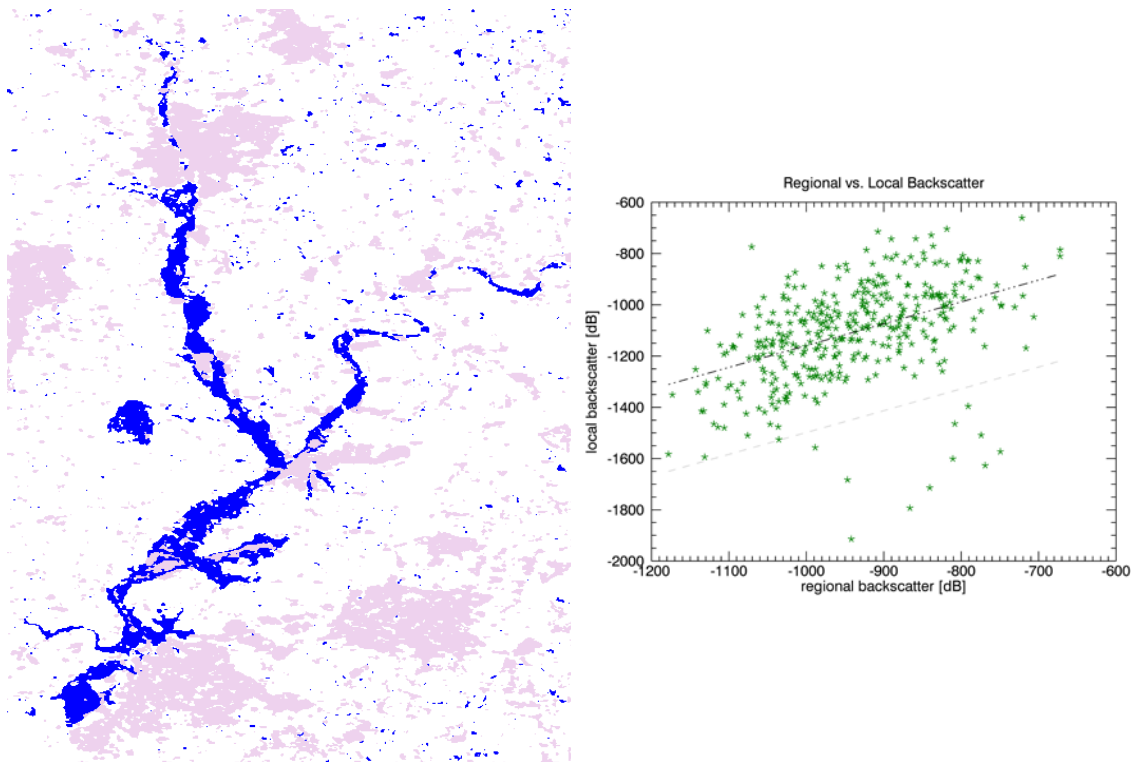
Figure 9.4.3: classification result using a mean calculated for image tiles. Left – classification results; right – scatter plot between regional and local backscatter

In contrast to the global mean, not only one average value was used for each acquisition date but average values were calculated for individual image tiles. Figure 9.4.3 shows the classification outcome using this averaging method. In comparison to the results of the two methods before, there is a slight change in

the slope of the line of regression. The classification result did not improve – there is more noise. However, the area of interest in the Severn region features a very homogenous land cover, in situations with a higher diversity of land cover types or stronger topographical features, the use of a tiling approach could have some advantages. Single, strong features like mountains would not influence the entire scene.

### Mean for image tiles, including the correlation layer

In this case, the average is calculated following the same principle as described above including an image mask covering all pixels with a correlation coefficient of 0.3 or smaller.



*Figure 9.4.4: classification result using mean values calculated for individual image tiles: left – classification map; right – scatter plot between regional and local backscatter*

There is no significant change between Figure 9.4.4 and Figure 9.4.3. Due to the fact that the regional value is not calculated over the entire image, urban areas, forests, densely vegetated areas or permanent water bodies cannot influence the regional value globally. Therefore, changes in the classification result, induced by

low correlation pixels are not as significant as if a global average is used. **Error! Reference source not found.** shows the results of change detection between the classification result from Figure 9.4.3 and Figure 9.4.4. The classifications are almost identical.

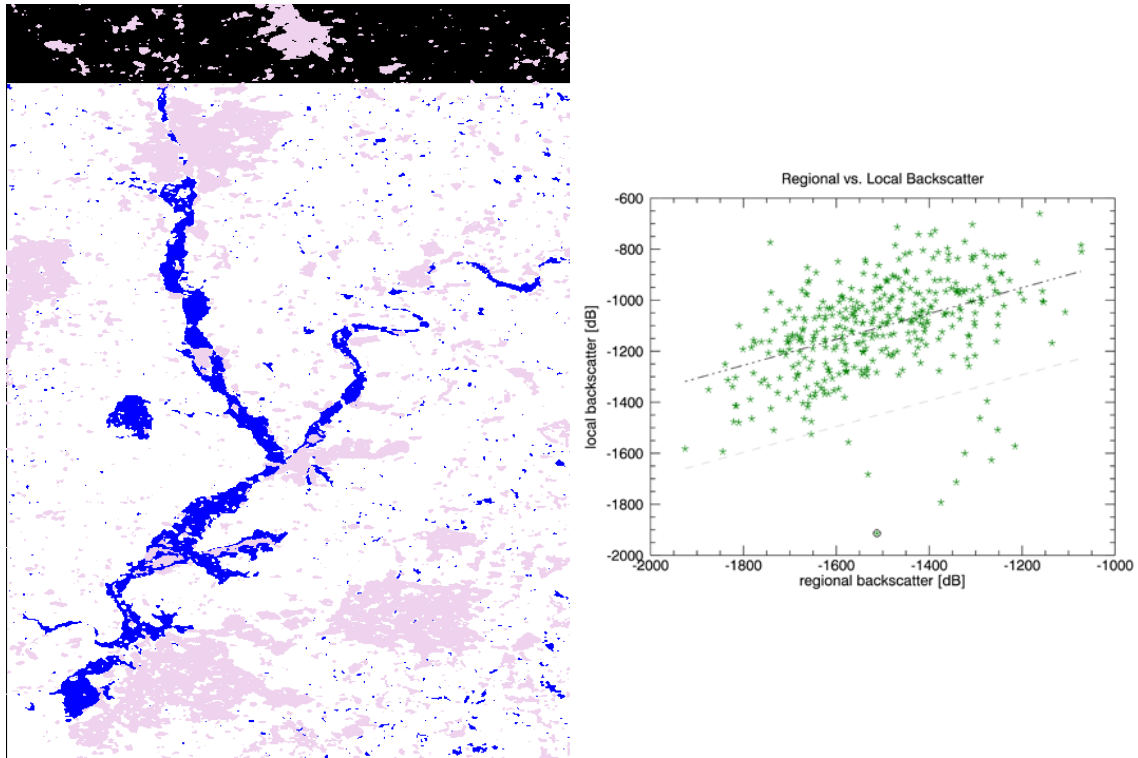
		Initial State		
		background	water	no-water
Final State	background	0.000	0.005	0.000
	water	0.000	98.627	0.015
	no-water	0.000	1.368	99.985
	class changes	0.000	1.373	0.015

*Table 9.4.2: change statistics between mean for image tiles without and with correlation layer. All numbers are given in percent*

### **Moving window**

The following results were generated, using a regional value calculated by a so called 'moving window'. In this case an individual regional value for each pixel is calculated by averaging over a 30 by 30 km area around it.

Using the 'moving window' averaging approach, a certain amount of data is lost (depending on the size of the moving window). For pixels within the distance of half the size of the moving window from the image borders, the mean cannot be calculated (for a 30 km by 30 km moving window, approximately 15 km on each side of the image are lost). This can be seen as a black strip on top of the classification result in Figure 9.4.5.



*Figure 9.4.5: classification results for mean from moving window: left – classification map; right – scatter plot between regional and local backscatter*

### **Moving window, including the correlation layer**

The final averaging method, we want to discuss, is the mean by moving window method combined with the correlation layer mask. Figure 9.4.6 presents the classification results. The differences between Figure 9.4.5 and Figure 9.4.6 are similarly small as they are between Figure 9.4.3 and Figure 9.4.4. The improvements induced by the application of the R-Layer mask are insignificant.

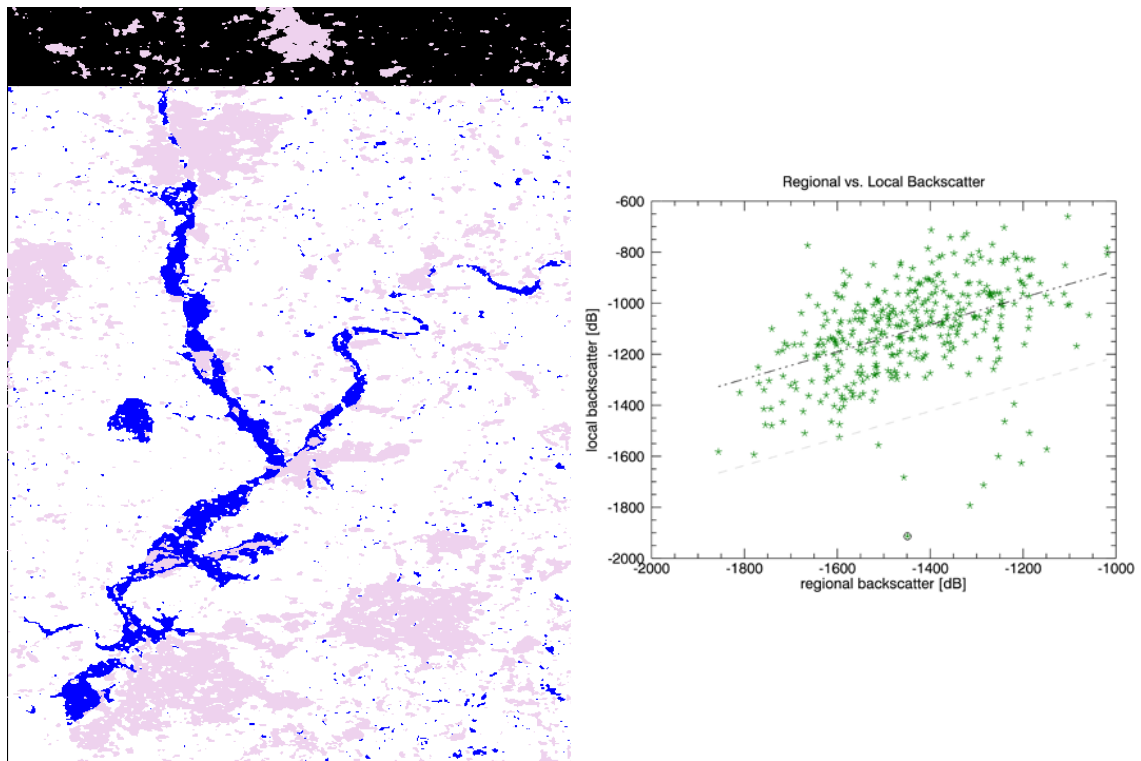


Figure 9.4.6: classification results from the mean by moving window method, including the correlation layer mask: left – classification map; right – scatter plot between regional and local backscatter

	Std.Dev.	SE(a)	SE(b)	Corr. Coeff.
<b>Global Mean</b>	155.184	77.719	0.088	0.555
<b>Global Mean + correlation layer</b>	154.705	71.061	0.078	0.556
<b>Mean for image tiles</b>	152.971	74.079	0.082	0.569
<b>Mean for image tiles + correlation layer</b>	152.668	69.219	0.074	0.570
<b>Moving window</b>	151.523	70.809	0.049	0.577
<b>Moving window + correlation layer</b>	151.075	66.067	0.052	0.578

Table 9.4.3: Quality parameters for the linear fit for different averaging methods

The visual comparison of the different averaging methods indicates that the use of a global mean produces the best classification outcome. The result can be further enhanced by masking all pixels with a correlation coefficient of 0.3 or less for the averaging process. However, the regression analysis accuracy measures do not support those findings (Table 9.4.3). The error measures give an estimation of how well the data is explained by the linear fit. They were calculated by averaging the fit quality parameters of 30, randomly selected, pixels. The standard deviation describes the standard distance between a single data point and the line of regression. The standard errors for  $a$  and  $b$  describe the discrepancy between the line of regression and a line that would go through each data point. The correlation coefficient provides an estimate of how well the relation between two datasets can be described linearly. On the one hand the results support the assumption that by masking pixels with low correlation, the fit quality and thus the classification is improved. On the other hand they show that a good fit does not necessarily produce an accurate classification.

The potential quality of a linear regression increases with the correlation between the two input variables. The kind of averaging method already implies an increasing correlation if viewed from global mean to moving window. In the case of a global mean, each pixel in an image has the same regional value. The moving window method assigns a different regional value for each pixel, based on the area surrounding it. Thus local variations have a stronger influence on the moving window regional value and therefore the correlation between it and the local value is higher. As Table 9.4.3 and the classification graphics show, this does not imply a better classification.

#### **9.4.2 Comparison between mean and median**

As discussed in section 6.3.2, due to its stronger resistance to outliers, the median often offers a better representation of the expected value than the mean.

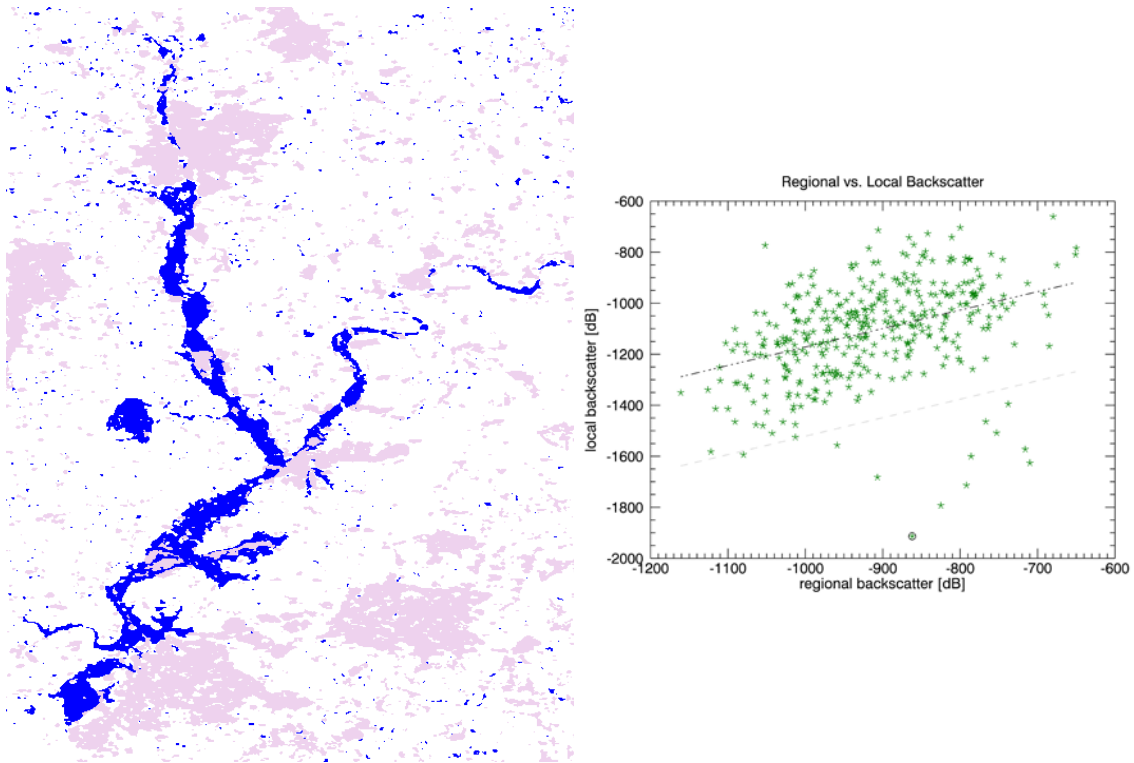


Figure 9.4.7: classification results using the median for the calculation of the regional value. Right – classification map; right – scatterplot with line of regression and -2 standard deviation distance

The image above (Figure 9.4.7) shows the results with the median used as an estimator for the regional value. The regional value was determined as one global average for each date. Also in the calculation included was the correlation layer mask (pixels with a correlation coefficient of 0.3 or lower were masked). The direct comparison of the classification results from Figure 9.4.2 and the results in Figure 9.4.7 show a decrease in classification quality. In Figure 9.4.7 a large number of small water surfaces were detected outside the main flooding area.

	Minimum	Maximum	Standard Dev.
<b>Regional value - mean</b>	-1168.12	-643.32	101.451
<b>Regional value - median</b>	-1161.00	-650.00	101.326

Table 9.4.4: statistical distribution of regional values calculated by mean and median, respectively



Despite the fact that the median is a more robust measure for the average value, the differences are insignificant (Table 9.4.4).

	Std. Dev.	SE(a)	SE(b)	Corr. Coeff.
<b>Regional value - mean</b>	154.705	71.061	0.078	0.556
<b>Regional value - median</b>	154.834	71.660	0.078	0.556

*Table 9.4.5: linear fit quality measures*

Table 9.4.5 shows the quality measures for the linear fit between regional and local value. The standard deviation in this table must not be confused with the standard deviation in Table 9.4.4. While the latter describes the standard deviation within the distribution of regional values, the first describes the standard deviation of each data point from the line of regression. The application of the median for the calculation of the regional average does not improve the linear relation between regional and local values; in this case the mean produces a higher quality fit. Even though the differences are insignificant, the numbers from Table 9.4.5 support the conclusion from the visual interpretation, which suggests that the mean as regional average produces a better classification result than the median.

### 9.4.3 Simple least squares or total least squares (orthogonal) regression

Usually, the goal of a linear regression analysis is to find a straight line that minimises the residuals between the predicted and the measured y coordinate. Thus, the residuals are measured parallel to the y-axis (distance between measured and fitted value). In a total least squares fit, the residuals are calculated perpendicular to the line of regression and none of the variables is considered fixed.

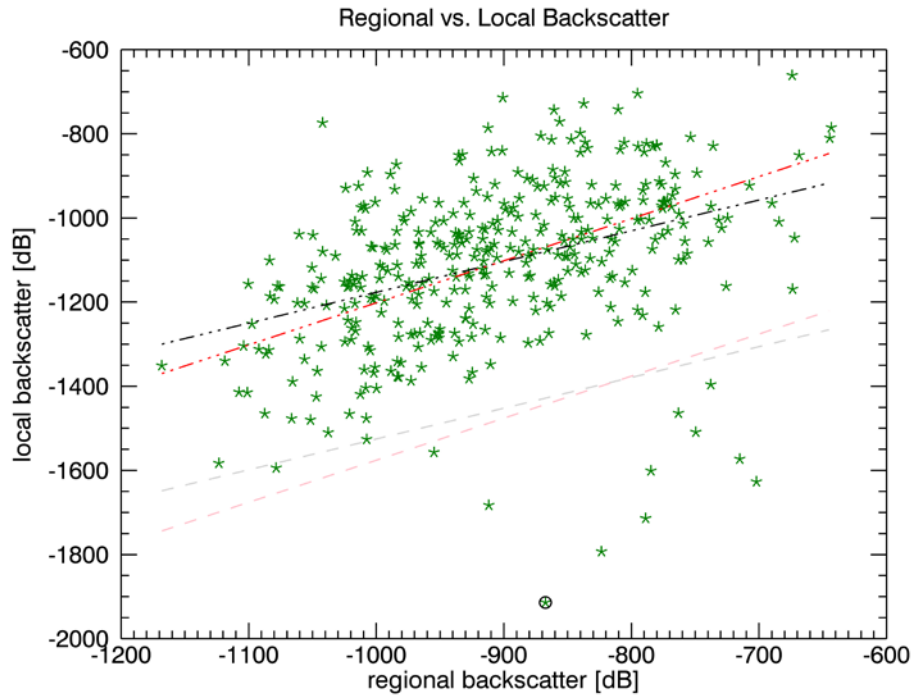
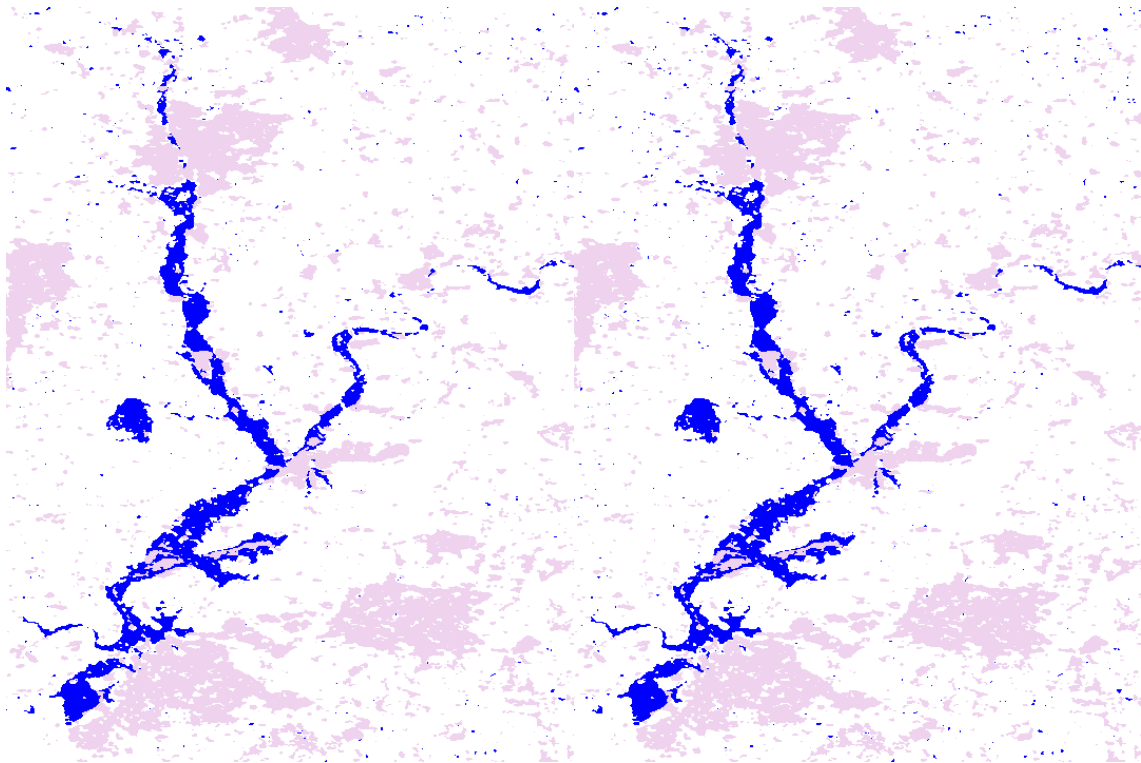


Figure 9.4.8: Scatterplot between regional and local backscatter. The black line represents the line of regression calculated by a linear fit; the grey line is the -2 standard deviation distance. The red line represents the line of regression calculated with a total least square fit; the light red line shows the respective -3 standard deviations distance.

Figure 9.4.8 shows a plot of the line of regression calculated with a simple linear fit and with an orthogonal fit, respectively. Visually, the line produced by the orthogonal fit (red) represents the data more accurately than the simple regression line (black). The figure shows that, for this particular pixel, there is only one date which would not be classified as water, based on the orthogonal regression, but would be, based on the simple linear regression. In case of the orthogonal regression, -3 standard deviations were used as a threshold, instead of -2 standard deviations for the linear regression. This means, in total there will be fewer pixels selected as water pixels. In a normally distributed dataset 2.1 % of the data lie beyond two standard deviations, only 0.1 % of the data have a higher standard deviation than 3.

The differences in the classification result are small (Figure 9.4.9). There is an overestimation of flood pixels outside the main flooding area; the result based on the orthogonal regression has more noise.



*Figure 9.4.9: Direct comparison between classification results: left – distances calculated, based on the simple linear regression; right – distances calculated based on the orthogonal regression*

Theoretically, a data sample where both axes are prone to uncertainties can be modelled better with an orthogonal regression [W. Niemeier, 2008]. Table 9.4.6 shows the quality measures of the two different fitting methods. The numbers prove that, in this case, measuring the residuals perpendicularly leads to shorter distances and thus ‘better’ error measures. Standard deviation, as well as both standard errors, has improved significantly between the two methods. The difference between the error measure for the simple linear regression and the orthogonal linear regression is approximately 27 %.

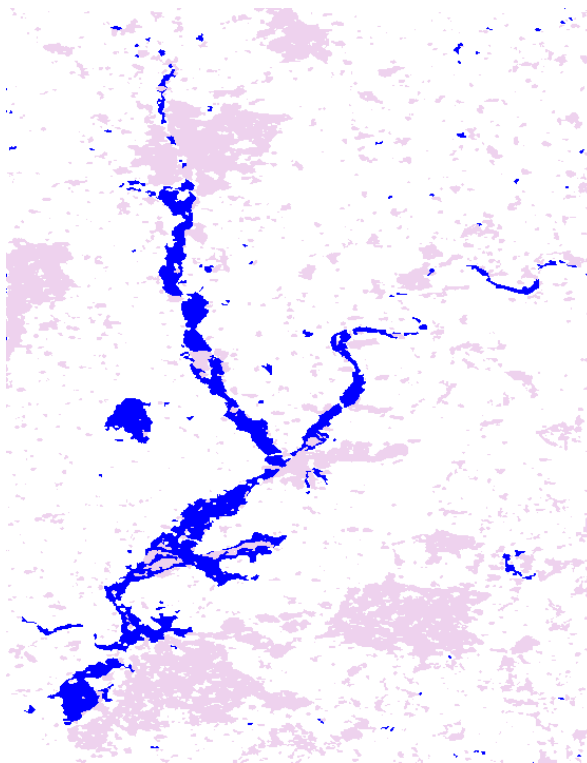
	Std. Dev.	SE(a)	SE(b)	Corr. Coeff.
<b>Simple linear regression</b>	154.705	71.061	0.078	0.556
<b>Orthogonal regression</b>	113.211	52.396	0.057	0.556

*Table 9.4.6: Quality measures for simple linear regression and orthogonal regression. Standard deviation, standard error-a, standard error-b and correlation coefficient*

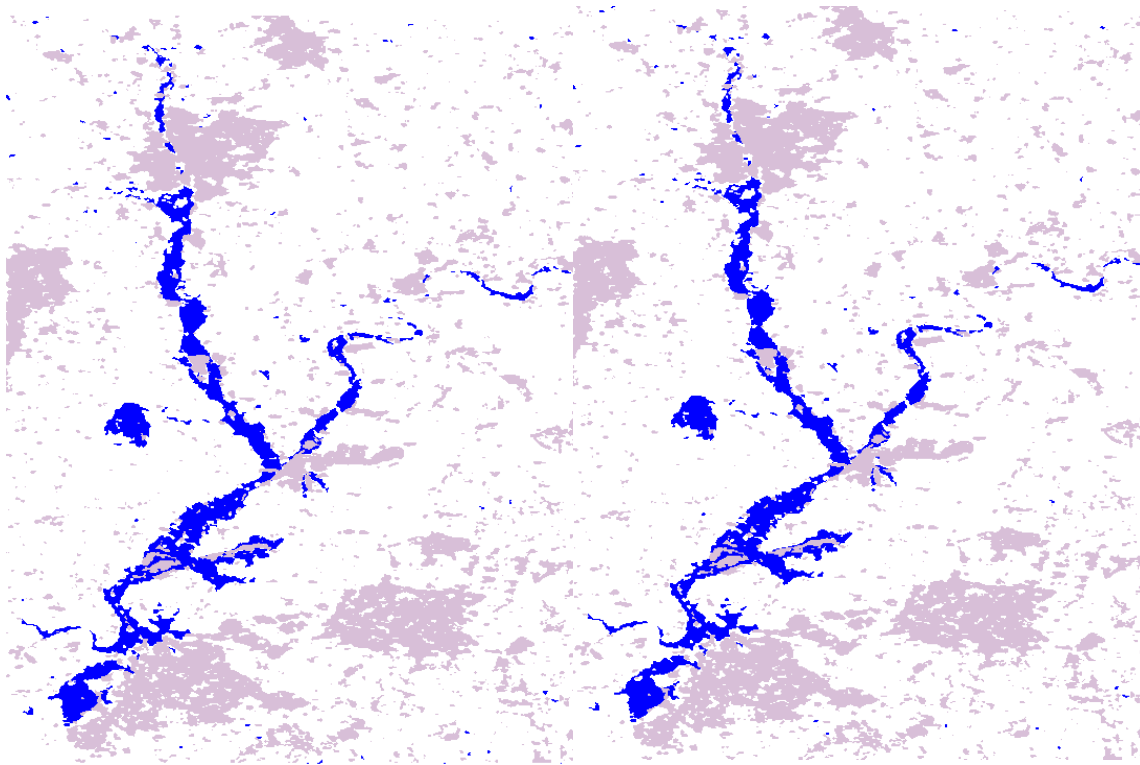
## 9.5 Post classification filtering

As a final improvement of the results, the post classification filtering routine described in section 8.4 was applied. The routine filters segments of water or non-water pixels, respectively based on their size. In our case small water surfaces outside the main flooding area and islands within the flooding area shall be removed.

After the application of the method there are hardly any visual differences between the results from the Otsu method and the results from the temporal stability approach (Figure 9.5.2 and Figure 6.3.1). There are some larger flooding areas which couldn't be removed by the post classification algorithm in the Otsu result.



*Figure 9.5.1: results from the Otsu method*



*Figure 9.5.2: results from the temporal stability approach, left – simple linear regression; right – orthogonal regression*

## 10 Accuracy Assessment

The results presented in section 5 suggest that the temporal stability approach, in general, produces a higher quality classification than Otsu's method. Furthermore it was shown that the global mean leads to the best results and that it can be improved by masking pixels with a low correlation.

The best results for the Otsu method were achieved with the split based approach including the correlation layer to mask pixels with a low correlation, i.e. areas where no flood detection is possible due to dense vegetation cover or urban areas.

### 10.1 Flood extent marks

The River Severn flood extent marks were used as a reference for a quality estimation of the temporal stability approach water body classification. We differentiated between marks in open areas and marks in urban areas. For that

purpose, we used the correlation layer, derived from the ASAR WS time series. Extent marks covered by the correlation layer mask (pixels with a correlation  $< 0.3$ ) were omitted for the accuracy assessment. In total there were 2310 marks of which we selected 1573. Figure 10.1.1 shows the selection of extent marks. The size of the circles represents the distance between an extent mark and its closest water pixel from the temporal stability water body classification. Green circles indicate an underestimation of the flood extent in the classification, red circles indicate an overestimation (i. e. the extent mark lies within the area classified as water surface).

The graphics show that both temporal stability results are very similar, despite the application of different regression methods.

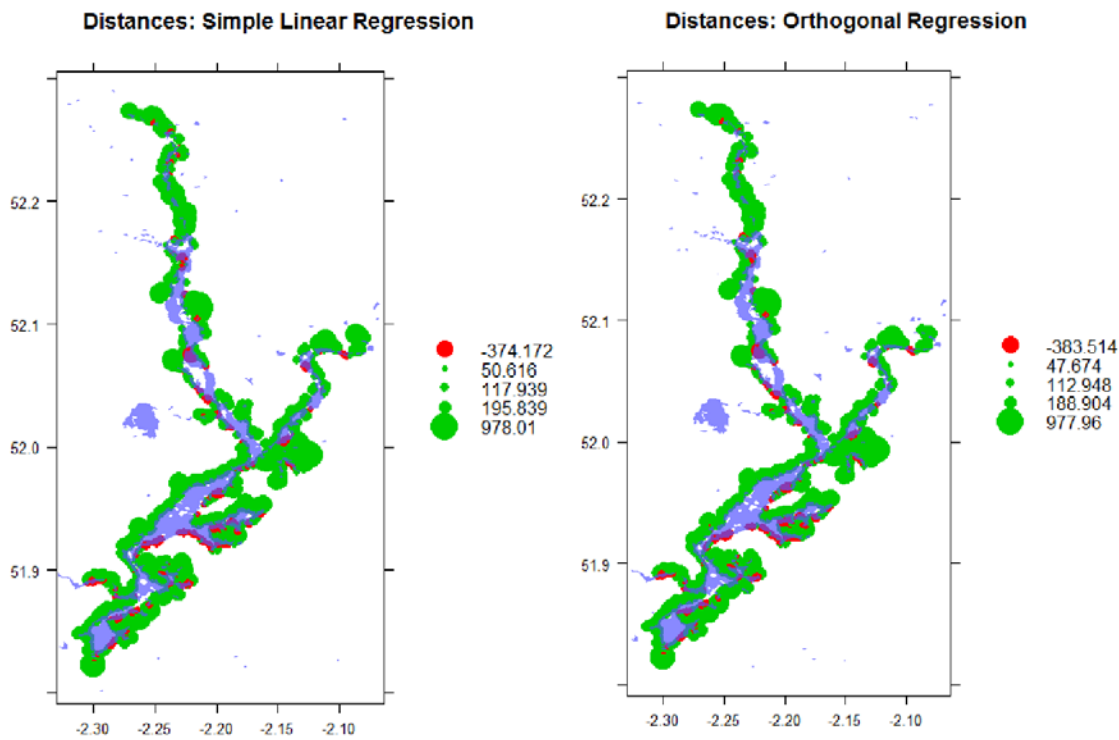


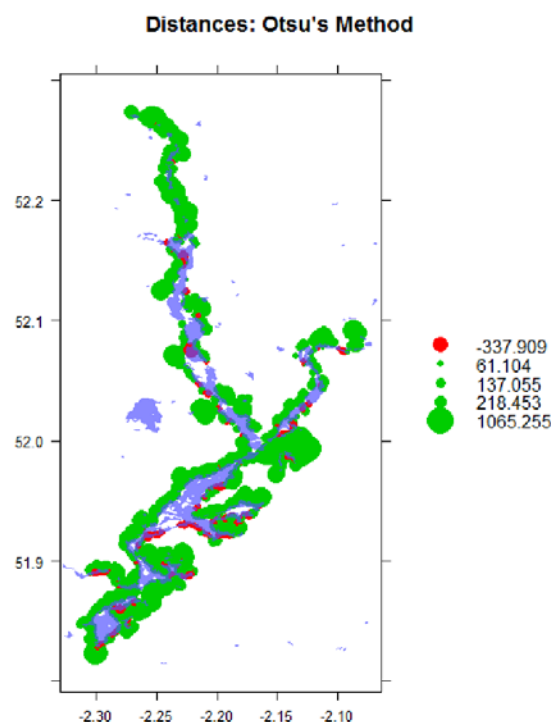
Figure 10.1.1: Flood extent marks and distances to classification result from the temporal stability approach.

Table 10.1.1 shows the corresponding numbers. According to the results the root mean square error (RMS), for the simple linear regression is at 199.02 m and for, the orthogonal regression at 194.85. The RMS, also known as the quadratic mean, is a measure of the magnitude of a varying quantity. In this case this is the

distance between extent mark and classification, thus the RMS is standard error of the classification.

The original spatial resolution of the ASAR WS dataset, used as a classification basis, is 150 m. A RMS of 199.02 m or 194.85 m, respectively, means that the standard error of the classification is only slightly above the dimensions of one pixel.

Figure 10.1.2 presents the quality estimation for the Otsu method. The graphical representation does look very similar to the results from the temporal stability approach (Figure 10.1.1). The underestimation of the flood extent is slightly higher (maximum distance of 1065.26) while the overestimation is slightly smaller (maximum negative distance if -337.91). However, Table 10.1.1 shows that the Otsu Algorithm produces an RMS of 228.94 which is significantly higher than the standard error for both temporal stability results.



*Figure 10.1.2: Extent marks and water body classification derived with Otsu's method*

	Minimum Distance [m]	Maximum Distance [m]	RMS [m]
<b>Temporal Stability</b>	0.00	978.01	199.02
<b>Temporal Stability, Orthogonal Regression</b>	23.63	977.96	194.85
<b>Otsu</b>	0.00	1065.26	228.94

*Table 10.1.1: Distances between classification and flooding extent marks*

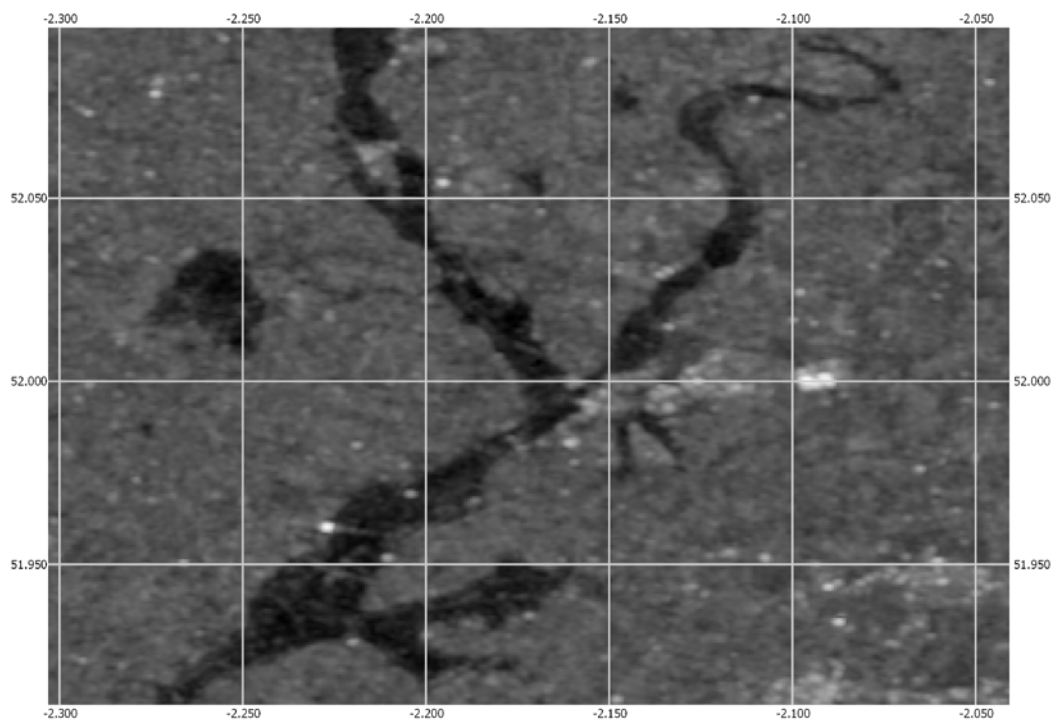
## 10.2 Geocoding

Certain inaccuracies are added by the geocoding process. The ASAR WS time series is composed from images, recorded from multiple satellite orbits. Therefore images are taken from different viewing angles. On the one hand this can cause radiometric distortions which should be largely corrected, by the normalisation process. On the other hand it can cause geometric distortions; this should be compensated by the range Doppler geocoding process. However, to fully correct the geometric influence of different viewing angles, a high accuracy digital elevation model would be required.

The SRTM digital elevation model, which was used for geocoding, has a 90% absolute height error of 6.6 m [Rodriguez, 2005].

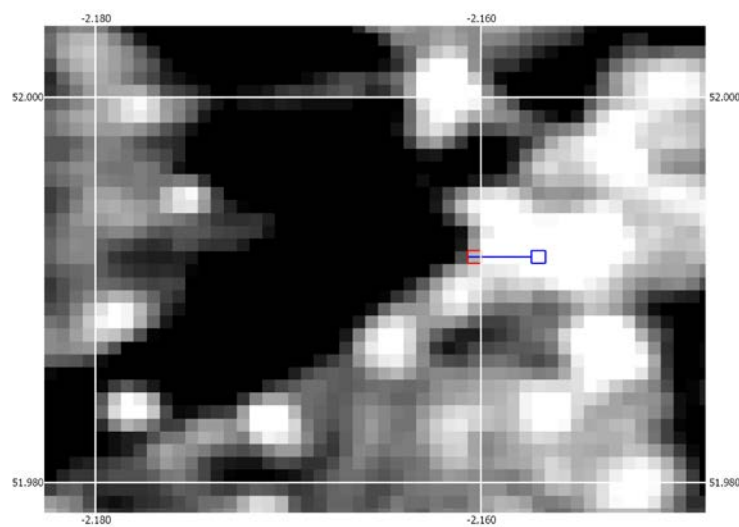
One factor that is adding additional variation to the ASAR WS time series are inaccuracies which are induced by the geocoding process, i.e. shifts between individual scenes. The combination of error in the digital elevation model and sudden changes in the topography can cause errors in the geocoded image, e.g. shifts between images recorded from different viewing angles.



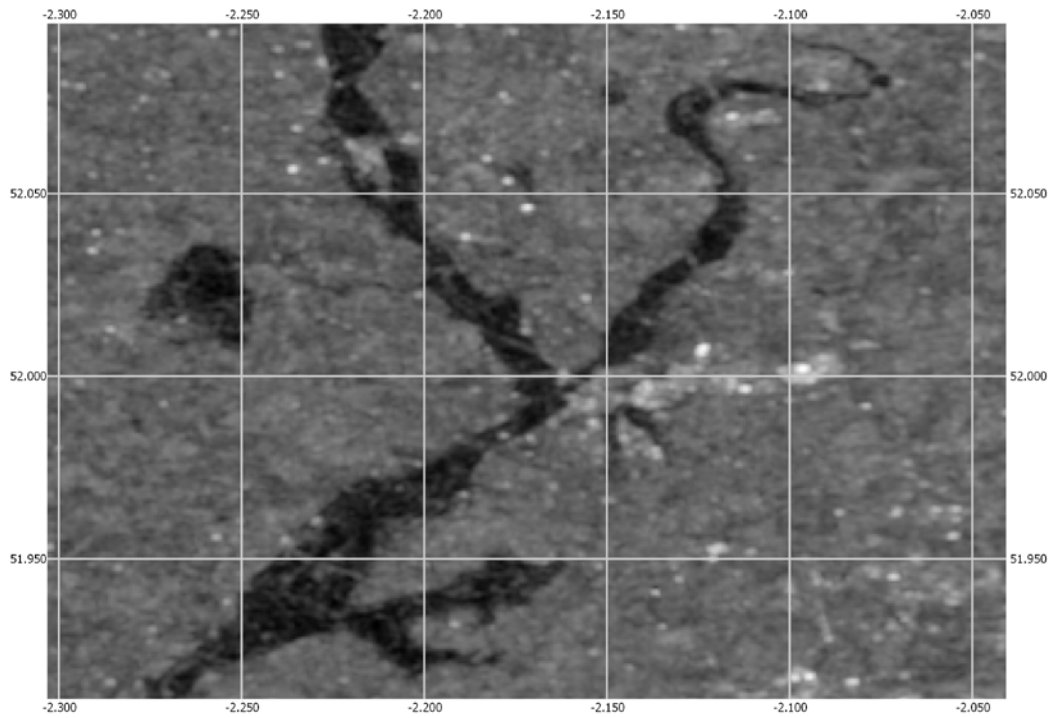


*Figure 10.2.1: ASAR WS image recorded on the 23<sup>rd</sup> of July 2007 – morning*

A significant east – west shift was discovered between the morning and the evening scene from the 23<sup>rd</sup> of July 2007 (Figure 10.2.1 and Figure 10.2.3). The difference between the two images is approximately 221.5 m. However, the error varies within the area of interest. Figure 10.2.2 shows a detailed view of the shift of one pixel. The blue square represents the approximate position of the pixel marked by the red square, in the morning scene.



*Figure 10.2.2: Shift of one pixel between morning and evening scene*



*Figure 10.2.3: ASAR WS image recorded on the 23 of July - evening*

## 11 Conclusion

One of the main problems in an unsupervised classification process is that conditions can vary significantly between different regions on the earth. For instance, it can be very hard to distinguish between water and non-water surface when the area is generally very dry and smooth (e.g. sand). SAR images consisting of such land-cover show little contrast.

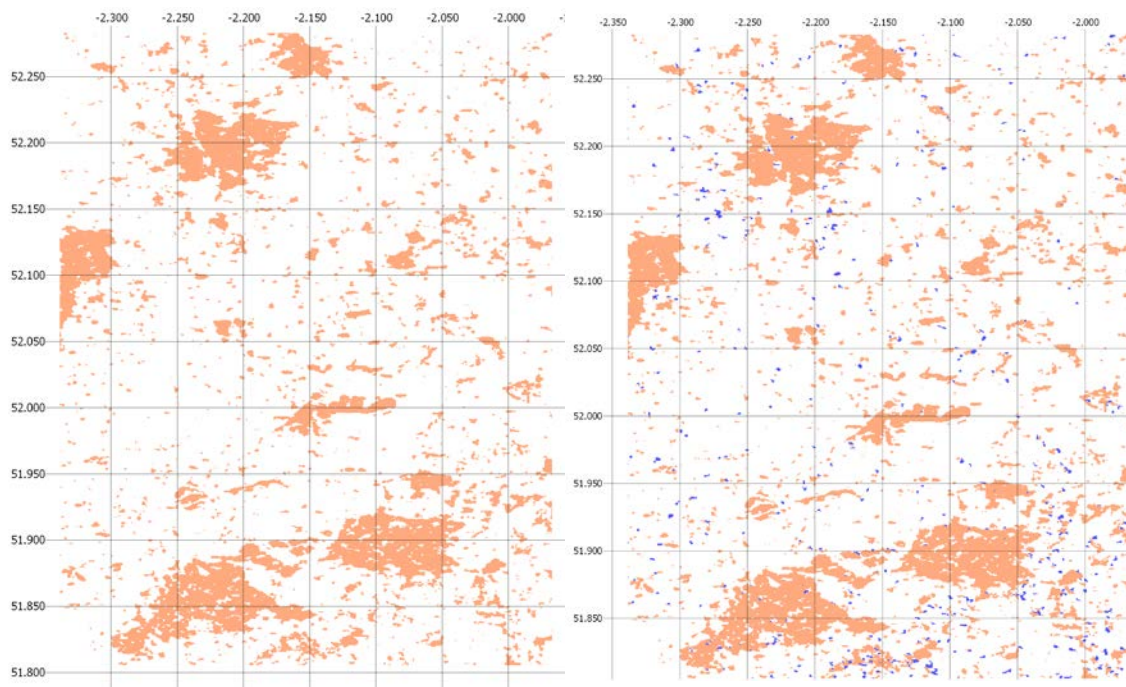
Overall the findings presented in this work show that it is possible to achieve a water body classification automatically and without any required interaction during the processing phase.

Two different classification methods were tested: the temporal stability approach and the Otsu algorithm. The temporal stability of radar images was originally used by Wagner et al. 2008, for the characterisation of surface soil moisture patterns. In this work, it was proven that the approach can be successfully applied for the classification of water bodies or flooding, respectively. In terms of accuracy it delivers slightly better results than Otsu's

Method which is well established and has been applied for such purposes many times.

One major advantage of Otsu's Method is that the classification can be based on one single geocoded radar image only, which means, other than the digital elevation model for geocoding, no additional data is needed (the computation based on a single image is not possible if the correlation layer is used).

Additionally, the computational effort is significantly smaller than for the temporal stability approach. This makes it easier to implement in a near real time processing environment. The algorithm finds a threshold between water and non-water surfaces, based on the grey level histogram of an image. Due to this approach the method is highly sensitive to the shape of the histogram and thus the land-cover composition within an image. If, for example, the amount of water compared to other surfaces is too low, it can be hard for the Otsu algorithm to determine a correct threshold. Another problem can occur if there is no water in the area of interest; when the algorithm is applied to a histogram, it always finds a threshold, even if no water is present. Partially those problems can be compensated by the pre-selection routine which splits up the image into overlapping tiles and only selects those which have a high variation. However, there can be high variations between other land-cover classes as well (e.g. agricultural areas and forest or urban areas). These areas can be masked with the R-Layer (section 9.1).



*Figure 10.2.1: Left – classification result, obtained by Otsu’s method, for a non-flooding scene; right – classification result, obtained by temporal stability approach, for the same non-flooding scene.*

The temporal stability approach has the disadvantage that it requires not only a single radar image but a time series of images. This method is not based on the grey-level histogram, like Otsu’s method, but on the temporal stability between regional and local value (under normal circumstances). If a pixel is flooded it can be detected as an outlier from this otherwise stable relation. The generation of the image time-series requires a significant amount of pre-processing; geocoding, resampling, normalisation and averaging (for the regional value) have to be performed for each image in the stack. The advantage of this approach is that it is less sensitive to inhomogeneity of the area of interests land cover. Additionally, other than Otsu’s Method, the performance of the temporal stability approach is not affected by the ratio between water and non-water surfaces. The classification is based on the time series of each pixel - if there are no pixels in an image that excess the threshold distance from the line of regression, none are classified as water.

Figure 10.2.1 shows the classification results for a non-flooding date obtained by Otsu’s method and the temporal stability approach, respectively (both including the correlation layer). The Otsu algorithm pre-selection routine didn’t find any

image subset which qualified for thresholding, which is why there are no pixels, classified as water. This approach requires relatively large water surface to work. In the results from the temporal stability approach small areas were classified as water. This could be caused either by minor flooding during that time (the image was recorded on the 30<sup>th</sup> of July 2009 which was a time of high precipitation, approximately 11 cm during the month of July) or by image noise.

Both approaches (temporal stability and Otsu's method) could be successfully applied for the automated detection of flooding in the river Severn area. Based on our findings, the temporal stability approach is more stable than Otsu's method. However, it requires a higher processing effort; it can only work if a sufficient amount of data is available for the area of interest. If this not the case, the Otsu algorithm can still provide a good alternative. It has to be noted that if only a single dataset is available, it is impossible to generate the R-Layer. Thus, a lower classification quality, as we were able to achieve in this work, would have to be expected.

## 12 Bibliography

Abdi, Hervé, and Lynne J. Williams. "Principal Component Analysis." *Wiley Interdisciplinary Reviews: Computational Statistics* 2.4 (2010): 433-59.

Acton, Forman S. *Analysis of Straight-line Data*. New York: Wiley, 1959.

Allan, R. P., and B. J. Soden. "Atmospheric Warming and the Amplification of Precipitation Extremes." *Science* 321.5895 (2008): 1481-484.

Bazi, Yakoub, Lorenzo Bruzzone, and Farid Melgani. "Image Thresholding Based on the EM Algorithm and the Generalized Gaussian Distribution." *Pattern Recognition* 40.2 (2007): 619-34.

Bhaskaran, B., and J. F. B. Mitchell. "Simulated Changes in Southeast Asian Monsoon Precipitation Resulting from Anthropogenic Emissions." *International Journal of Climatology* 18.13 (1998): 1455-462.

Canny, John. "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986): 679-98.

"Chapter 1 The ASAR User Guide." *The ASAR User Guide*. ESA. 19 Mar. 2012.  
<<http://envisat.esa.int/handbooks/asar/CNTR1.htm>>.

De Groen, P. "An Introduction to Total Least Squares." *Nieuw Archief Voor Wiskunde* 4 (1996): 237-253.

Entin, J., A. Robock, K. Y. Vinnikov, S. E. Hollinger, S. Liu, and A. Namkhai. "Temporal and Spatial Scales of Observed Soil Moisture Variations in the Extratropics." *Journal of Geophysical Research* 105 (2000): 11865-1877.

ESA. "Level 1B Products." *Level 1B Products*. ESA. 05 Sept. 2012.  
<<http://envisat.esa.int/handbooks/asar/CNTR2-6.htm>>.

ESA. "Parameters Affecting Radar Backscatter." 36. *Parameters Affecting Radar Backscatter*. ESA. 17 July 2012.  
<[http://earth.esa.int/applications/data\\_util/SARDOCS/spaceborne/Radar Courses/Radar Course III/parameters affecting.htm](http://earth.esa.int/applications/data_util/SARDOCS/spaceborne/Radar_Courses/Radar_Course_III/parameters_affecting.htm)>.

Giorgi, Filippo, Linda O. Mearns, Christine Shields, and Larry McDaniel. "Regional Nested Model Simulations of Present Day and 2XCO<sub>2</sub> Climate over the Central Great Plains of the United States." *Climate Change* 40.3-4 (1998): 457-93.

Gonick, Larry, and Woollcott Smith. *The Cartoon Guide to Statistics*. New York, NY: HarperPerennial, 1993.

Hennessy, K. J., J. M. Gregory, and J. F. B. Mitchell. "Changes in Daily Precipitation under Enhanced Greenhouse Conditions." *Climate Dynamics* 13.9 (1997): 667-80.

Horritt, M. S., D. C. Mason, and A. J. Luckman. "Flood Boundary Delineation from Synthetic Aperture Radar Imagery Using a Statistical Active Contour Model." *International Journal of Remote Sensing* 22.13 (2001): 2489-507.

- Kapur, J., P. Sahoo, and A. Wong. "A New Method for Gray-level Picture Thresholding Using the Entropy of the Histogram." *Computer Vision, Graphics, and Image Processing* 29.3 (1985): 273-85.
- Kittler, J., and J. Illingworth. "Minimum Error Thresholding." *Pattern Recognition* 19.1 (1986): 41-47.
- Knutson, T. R., and R. E. Tuleya. "Increased Hurricane Intensities with CO<sub>2</sub>-induced Warming as Simulated Using the GFDL Hurricane Prediction System." *Climate Dynamics* 15.7 (1999): 503-19.
- Kothavala, Z. "Extreme Precipitation Events and the Applicability of Global Climate Models to the Study of Floods and Droughts." *Mathematics and Computers in Simulation* 43.3-6 (1997): 261-68.
- Liu, H., and K. C. Jezek. "Automated Extraction of Coastline from Satellite Imagery by Integrating Canny Edge Detection and Locally Adaptive Thresholding Methods." *International Journal of Remote Sensing* 25.5 (2004): 937-58.
- Martinis, S., A. Tuele, and S. Voigt. "Towards Operational near Real-time Flood Detection Using a Split-based Automatic Thresholding Procedure on High Resolution TerraSAR-X Data." *Natural Hazards and Earth System Science* 9.2 (2009): 303-14.
- Matgen, P., R. Hostache, G. Schumann, L. Pfister, L. Hoffmann and H.H.G. Savenije. "Towards an Automated SAR-based Flood Monitoring System: Lessons Learned from Two Case Studies." *Physics and Chemistry of the Earth* 36 (2011): 241-252.
- Niemeier, Wolfgang. *Ausgleichungsrechnung: Statistische Auswertemethoden*. Berlin: De Gruyter, 2008.
- Nwanna, Gladson I. *Natural Disasters and Other Emergencies, What You Should Know: A Family Planning & Survival Guide*. Baltimore, MD: Frontline, 2004.
- Otsu, N. "A Threshold Selection Method from Gray-Level Histograms." *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979): 62-66.

R., Barstow. *ENVISAT-1 Products Specifications*. Tech. no. PO-RS-MDA-GS-2009. 4th ed. Vol. 8.: ESA, 2007. ASAR Products Specifications.

Reynard, Nick, Sue Crooks, Rob Wilby, and Alison Kay. *CLIMATE CHANGE AND FLOOD FREQUENCY IN THE UK*. Rep. Climate Change Unit and Environment Agency UK, 2004. Web. 10 Aug. 2012.

<<http://nora.nerc.ac.uk/2976/1/N002976C0%5B1%5D.pdf>>.

"The River Severn Facts." *BBC News*. BBC. 05 Sept. 2012.

<<http://www.bbc.co.uk/england/sevenwonders/west/severn-river/>>.

Rodriguez, E., C. S. Morris, J. E. Belz, E. C. Chapin, J. M. Martin, W. Daffer, and S. Hensley. *An Assessment of the SRTM Topographic Products*. Rep. JPL and NASA, 2005. 18 Aug. 2012. <[http://www2.jpl.nasa.gov/srtm/SRTM\\_D31639.pdf](http://www2.jpl.nasa.gov/srtm/SRTM_D31639.pdf)>.

Rosich, B., and P. Meadows. *Absolute Calibration of ASAR Level 1 Products*. Rep. N.p.: ESA, 2004.

Sabel, Daniel. *ENVISAT ASAR Processing at IPF*.

Schreier, Gunter. *SAR Geocoding: Data and Systems*. Karlsruhe: Wichmann, 1993.

Schumann, G., G. Di Baldassarre, and P.d. Bates. "The Utility of Spaceborne Radar to Render Flood Inundation Maps Based on Multialgorithm Ensembles." *IEEE Transactions on Geoscience and Remote Sensing* 47.8 (2009): 2801-807.

Sezgin, Mehmet, and Bülent Sankur. "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation." *Journal of Electronic Imaging* 13.1 (2004): 146.

Simoncelli, E. P. *Least Squares Optimization*. Rep. N.p., Sept. 2004. Web. 11 July 2012. <<http://www.cns.nyu.edu/~eero/NOTES/leastSquares.pdf>>.

Sitton-Kent, Mark. *River Severn Catchment Flood Management Plan*. Rep. Vol. 1.1. Solihull: Environment Agency UK, 2009.

Small, D., and A. Schubert. *Guide to ASAR Geocoding*.: ESA, 2008. Print.



Tsao, Brandt, and Ppaul Mather. *Classiciation Methods for Remotely Sensed Data*. 2nd ed.: Taylor & Francis, 2009.

Ulaby, Fawwaz T., Adrian K. Fung, and Richard K. Moore. *Microwave Remote Sensing: Active and Passive. Volume 2: Radar Remote Sensing and Surface Scattering and Emission Theory*. Vol. 2. Reading, MA: Addison-Wesley, 1982.

Ulaby, Fawwaz T., Richard K. Moore, and Adrian K. Fung. *Microwave Remote Sensing: Active and Passive*. Vol. 1. Reading, MA: Addison-Wesley Pub., Advanced Book Program/World Science Division, 1981.

UNISDR. *2011 Disasters in Numbers*, 2011.

Vachaud, G., A. Passerat De Silans, P. Balabanis, and M. Vauclin. "Temporal Stability of Spatially Measured Soil Water Probability Density Function1." *Soil Science Society of America Journal* 49.4 (1985): 822.

Wagner, Wolfgang, Carsten Pathe, Marcela Doubkova, Daniel Sabel, Annett Bartsch, Stefan Hasenauer, Günter Blöschl, Klaus Scipal, José Martínez-Fernández, and Alexander Löw. "Temporal Stability of Soil Moisture and Radar Backscatter Observed by the Advanced Synthetic Aperture Radar (ASAR)." *Sensors* 8.2 (2008): 1174-197.

Webster, M., J. Ginetti, P. Walker, D. Coppard, and R. Kent. *The Humanitarian Costs of Climate Change*. Rep. Feinstein International Center, 2008. 21 Aug. 2012. <<http://www.indiaenvironmentportal.org.in/files/FeinsteinTuftsclimatechange.pdf>>.

Weisstein, Eric W. "Least Squares Fitting--Perpendicular Offsets." *Least Squares Fitting--Perpendicular Offsets*. MathWorld--A Wlofram Web Resource. 10 July 2012. <<http://mathworld.wolfram.com/LeastSquaresFittingPerpendicularOffsets.html>>.

Woodhouse, I. H. *Introduction to Mircowave Remote Sensing*. Boca Ranton: CRC, 2006

## 13 Appendix 1

In the following section, the IDL code written for the execution of the Otsu Method can be found:

```
;This function finds segments of high variance in an image and calls
;classification function
pro pre_otsu, infile, r_layer_infile, outfile, band,
histo_plot=histo_plot

;reading the image file
image = read_envi_new(infile)
r_layer = read_envi_new(r_layer_infile)
hdr = image.hdr
image = float(image.data[*,*,band])
r_layer = r_layer.data
nodata= -9999.0
xy = size(image, /DIMENSIONS)

;pixels with a value equal to the no-data-value or with a correlation
;coefficient below 0.3 are masked
if (size(where(image eq nodata), /DIMENSIONS))[0] gt 0 then
image[where(image eq nodata)] = !Values.F_NAN
if (size(where(r_layer lt 0.3), /DIMENSIONS))[0] gt 0 then
image[where(r_layer lt 0.3)] = !Values.F_NAN
DELVAR, r_layer

;Mapping all image value to a range from 0 to 255
orig_min = min(image, /NAN)
neg_mask = where(~finite(image), neg_count)
if neg_count gt 0 then image[neg_mask] = -9999.0
image = image + abs(orig_min)
range = (max(image, /NAN) - orig_min)
image = round((image / range) * 255.0, /L64)
image = float(image)
if neg_count gt 0 then image[neg_mask] = !VALUES.F_NAN
DELVAR, range, neg_mask, neg_count

;initialising the pre-selection routine
mask = finite(image)
xy = DOUBLE(size(image, /DIMENSIONS))
winddim = 200 ;dimensions for the moving window
x = xy[0] / winddim * winddim
y = xy[1] / winddim * winddim
h = HISTOGRAM( image, LOCATIONS=h_loc, /NAN, BINSIZE=1.0)
T1 = 0.0
threshold = 0.0
subset_count = 0
image_statistics, image, STDDEV = stddev_image, MINIMUM =
min_image, MAXIMUM = max_image, MEAN = mean_image, MASK = mask

;in the following part a window with high variance is selected and
;the threshold for this window calculated. In the end an average
;threshold is calculated
for i = 0, x-winddim, winddim/2 do BEGIN
  for j = 0, y-winddim, winddim/2 do BEGIN
    if (j lt y-winddim) and (i lt x-winddim) then imgwindow =
image[i:i+(winddim-1),j:j+(winddim-1)]
```

```

        if (j ge y-winddim) and (i lt x-winddim) then imgwindow =
        image[i:i+(winddim-1),j:xy[1]-1]
        if (j lt y-winddim) and (i ge x-winddim) then imgwindow =
        image[i:xy[0]-1,j:j+(winddim-1)]
        if (j ge y-winddim) and (i ge x-winddim) then imgwindow =
        image[i:xy[0]-1,j:xy[1]-1]
        mask_window = finite(imgwindow)
        image_statistics, imgwindow, STDDEV = stddev_window, MINIMUM =
        min_window, MAXIMUM = max_window, MEAN = mean_window, MASK =
        mask_window
;calculating the selection criterion
        cvx = abs(mean_window / stddev_window)
        rxn = abs(mean_window / mean_image)
        if (cvx ge 0.7) and (rxn le 0.9) and
        (size(where(finite(imgwindow)), /DIMENSIONS))[0] gt 9000 then
        BEGIN
            threshold = threshold + otsu(imgwindow, mask_window)
            subset_count = subset_count + 1
        ENDIF
    ENDFOR
ENDFOR

;calculating the final threshold
        threshold = threshold / subset_count
        watercount = size(where(image le threshold), /DIMENSIONS)
        water_percentage = watercount/(xy[0]*xy[1])

        if subset_count eq 0 then threshold = 0

;histogram plot output
        if keyword_set(histo_plot) then begin
            p1 = PLOT( h_loc, h )
            p1.title = 'Threshold derived by Otsu Algorithm
            p2 = PLOT( [threshold,threshold+0.00000001], [0,MAX( h )],
            /OVERPLOT )
            p2.COLOR = 'red'
            p1.XTITLE = 'Digital Count (DC)'
            p1.YTITLE = 'Number of Pixels'
        endif

;calculate the original backscatter value
        threshold_orig = ((threshold/255) * range) - abs(orig_min)

;output to the screen
        print, 'Otsu: ' + string(threshold_orig)
        print, 'Number of selected subsets: ' + string(subset_count)
        print, 'Percentage of surface covered with water (Otsu): ' +
        string(water_percentage)

;image classification and output
        threshold_Image = image
        DELVAR, image
        lowIndex = WHERE( image LE threshold, count )
        IF ( count GT 0 ) THEN threshold_Image[lowIndex] = 1
        highIndex = WHERE( image GT threshold, count )
        IF ( count GT 0 ) THEN threshold_Image[highIndex] = 0
        no_dataIndex = WHERE( ~finite(image), count )
        if count gt 0 then threshold_Image[no_dataIndex] = 0

        outhdr = { param_descr : 'Otsu Classification', $

```

```

        samples : hdr.samples, $
        lines   : hdr.lines, $
        bands    : 1, $
        header_offset : 0, $
        file_type : 'ENVI standard', $
        data_type : 1, $
        interleave : 'BSQ', $
        map_info : hdr.map_info, $
        data_ignore_value : -9999, $
        band_names : 'Water' $
    }

    write_envi_new, outfile, threshold_Image, outhdr

END

FUNCTION OTSU, image, mask, THRESHOLDED_IMAGE=thresholdedImage

;Preprocessing:
;step 1: application of the Lee-Filter for speckle reduction
;step 2: extraction of edges with the Canny-Edge filter, a buffer
;zone is calculated around the edges to avoid mixed pixels which
;could bias the threshold value.
    no_data = where(~finite(image))
    if (size(no_data, /DIMENSIONS))[0] gt 0 then image[no_data] = 0
    image = LEEFILTER(image, 2)
    if (size(no_data, /DIMENSIONS))[0] gt 0 then image[no_data] =
    !VALUES.F_NAN
    minvalue = min(image, /NAN)
;transform the input into a high-variance-image
    image_statistics, image, STDDEV = stddev_image, MAXIMUM =
    max_image, MASK = mask
    imagecanny = CANNY(image)
    histo_mask = dilate(imagecanny, [[1, 1, 1, 1], [1, 1, 1, 1], [1, 1,
    1, 1]])
    histo_mask = histo_mask or mask
    IMAGE_STATISTICS, image, MEAN = centre, MASK = histo_mask
    image_h = image
    image_h[where(histo_mask eq 0)] = !VALUES.F_NAN
    h = HISTOGRAM( image_h, OMIN=minDN, OMAX=maxDN, LOCATIONS=h_loc,
    /NAN)
    h_size = long(size(h, /DIMENSIONS))
    h_size = h_size[0]
    for i = 0, (h_size - 1) do BEGIN
        if h_loc[i] ge centre then BEGIN
            centre_pos = i
            break
        ENDIF
    ENDFOR
;Histogram-Smoothing
    for i = 2, h_size - 1 - 3 do h(i) = (h(i-2) + 2*h(i-1) + 3*h(i) +
    2*h(i+1) + h(i+2))/9

;-----
;compute the image histogram and probability density function
    minvalue = min(image, /NAN)
    pdf = h / total( h, /DOUBLE )
    h_size = long(size(h, /DIMENSIONS))
    h_size = h_size[0]-1

```

```

;iterate through all possible thresholds and compute the interclass
;variance
;at each level
    threshold = 0
    interclassVariance = 0
    ;maxDC2= maxDC - (maxDC / 100 * 80)
    n = indgen(h_size + 1)
    FOR k = 0, h_size-1 DO BEGIN
        om0 = total(pdf[0:k])
        om1 = total(pdf[k+1:h_size])

        omk = total(pdf[0:k])
        muk = total(n[0:k]*pdf[0:k])
        muT = total(n[0:h_size]*pdf[0:h_size])

        mu0 = muk/omk
        mu1 = (muT - muk)/(1-omk)

        sb = (muT*omk - muk)^2 / (omk*(1 - omk))

        if sb gt interclassVariance THEN BEGIN
            interclassVariance = sb
            threshold = k
        ENDIF
    ENDFOR
    xy = size(image, /DIMENSION)
    totalcount = xy[0] * xy[1]

;return the threshold to the calling routine
    RETURN, threshold+minvalue

END

```

## 14 Appendix 2

In this section the source code written for the execution of the temporal stability method can be found

```
;the following procedure calculates the regional mean for each image  
;in an image stack
```

```
pro static_mean, infile, outfile, rlayer=rlayer, nodata=nodata,  
med=med, tiff=tiff, corr=corr  
  
;If the keyword 'tiff' is set, the geotiff header is converted to a  
;envi header  
if keyword_set(tiff) then begin  
    ok = query_tiff(infile, stackinfo, GEOTIFF = geoinfo)  
    header = geotiff2envihdr(geoinfo, stackinfo, PROCESSING_STAGE  
        = 'normalised')  
endif else begin  
    headerpath = filext(infile, /PATH, /NAME) + '.hdr'  
    header = read_envi_header_new(headerpath)  
endelse  
if ~keyword_set(corr) then rlayer = read_envi_new(rlayer)  
temp = make_array(header.samples, header.lines, /FLOAT)  
  
;cleanup  
if (file_test(outfile)) then FILE_DELETE, outfile  
  
;iterating through each band within the image stack  
for i = 0, header.bands-1 do begin  
  
    if keyword_set(tiff) then temp = read_tiff(infile, CHANNELS = i)  
    else begin  
        case header.interleave of  
        'bsq': begin  
            data_start = ulong64((i * (header.samples*header.lines)) *  
                header.data_type)  
            temp = read_binary(infile, DATA_DIMS = [header.samples,  
                header.lines], DATA_START = data_start, DATA_TYPE =  
                header.data_type)  
        end  
  
        'bip': begin  
            for j = 0, header.samples-1 do begin  
                data_dims = [1, header.lines]  
                data_start = (j * header.bands + i) * header.data_type  
                temp[j,*] = read_binary(infile, DATA_DIMS = data_dims,  
                    DATA_START = data_start, DATA_TYPE = header.data_type)  
            endfor  
        end  
  
        'bil': begin  
            for j = 0, header.lines-1 do begin  
                data_dims = [header.samples, 1]  
                data_start = ((j * header.bands + i) + header.samples) *  
                    header.data_type  
                temp[*,j] = read_binary(infile, DATA_DIMS = data_dims,  
                    DATA_START = data_start, DATA_TYPE = header.data_type)  
            endfor  
        end  
    end  
end
```

```

        end
        endcase
    endelse

;calculate the mean for the read band
    meanvalues = make_array(header.samples, header.lines, /INTEGER,
        VALUE=-9999)
    if keyword_set(med) then begin
        if ~keyword_set(corr) then mask_r = where(rlayer.data lt 0.3,
            count_r)
        mask_nodata = where(temp eq nodata, count_nodata)
        if ~keyword_set(corr) then begin
            if count_r gt 0 then temp[mask_r] = !VALUES.F_NAN
        endif
        if count_nodata gt 0 then temp[mask_nodata] = !VALUES.F_NAN
        meanvalues[*,*] = round(median(temp[*,*,i], /EVEN))
    endif else begin
        mask_nodata = temp ne nodata
        if ~keyword_set(corr) then begin
            mask_r = rlayer.data ge 0.3
            mask = mask_nodata and mask_r
        endif else mask = mask_nodata
        image_statistics, temp, MEAN = im_mean, MASK = mask
        if im_mean ne 0 then meanvalues[*,*] = round(im_mean)
    endelse

;output to hard disk
    if (total(strmatch(tag_names(header), 'byte_order', /FOLD_CASE))
        eq 1) then begin
        if (header.byte_order eq 1) then $
            openw, unit, outfile, /APPEND, /GET_LUN,
                /SWAP_IF_LITTLE_ENDIAN $
        else $
            openw, unit, outfile, /APPEND, /GET_LUN, /SWAP_IF_BIG_ENDIAN
        endif else begin
            openw, unit, outfile, /APPEND, /GET_LUN, /SWAP_IF_BIG_ENDIAN
        endelse
        writeu, unit, meanvalues
        close, unit
        FREE_LUN, unit
    endfor

    outhdr = { param_descr : 'Global Mean', $
        samples : header.samples, $
        lines : header.lines, $
        bands : header.bands, $
        header_offset : 0, $
        file_type : 'ENVI standard', $
        data_type : 2, $
        interleave : 'BSQ', $
        map_info : header.map_info, $
        data_ignore_value : -9999 $
    }
    outhdr.map_info.units = 'units=Degrees'

    outhdrpath = filext(outfile, /PATH, /NAME) + '.hdr'
    write_envi_header_new, outhdrpath, outhdr

end

```

```

;calculate mean values for tiled images
pro mean_image_tiled

;data input
imagestack =
read_envi_new('U:\FAAPS\Mexico\temp\
ASA_WSMIMM_normalised_Tabasco_subset.dat')
r_layer = read_envi_new('U:\FAAPS\Mexico\temp\R_tiled')

meaning = fltarr(imagestack.hdr.samples, imagestack.hdr.lines,
imagestack.hdr.bands)
meaning[*,*,*] = -9999
r_mask = r_layer.data ge 0.3

DELVAR, r_layer

;iterate through each band within the stack and calculate mean values
;for each individual tile
for i = 0, imagestack.hdr.bands - 1 do begin
    mask = imagestack.data[*,* ,i] ne -9999
    mask = mask and r_mask
    for k = 0, imagestack.hdr.samples - 300, 300 do begin
        for l = 0, imagestack.hdr.lines - 300, 300 do begin
            if (k + 2 * 300) ge imagestack.hdr.samples then begin
                dx = imagestack.hdr.samples - 1
                dy = l + 299
            endif
            if (l + 2 * 300) ge imagestack.hdr.lines then begin
                dx = k + 299
                dy = imagestack.hdr.lines - 1
            endif
            if ((k + 2 * 300) ge imagestack.hdr.samples) and ((l + 2 *
300) ge imagestack.hdr.lines) then begin
                dx = imagestack.hdr.samples - 1
                dy = imagestack.hdr.lines - 1
            endif
            if ((k + 2 * 300) lt imagestack.hdr.samples) and ((l + 2 *
300) lt imagestack.hdr.lines) then begin
                dx = k + 299
                dy = l + 299
            endif
            temp_img = imagestack.data[k:dx,l:dy,i]
            temp_mask = mask[k:dx,l:dy]
            image_statistics, temp_img, MASK = temp_mask, MEAN = temp
            meaning[k:dx,l:dy,i] = temp
        endfor
    endfor
endfor

header = imagestack.hdr
header.data_type = 2
header.data_ignore_value = -9999
meaning = fix(round(meaning))
write_envi_new, 'U:\FAAPS\Mexico\temp\tiled_mean', meaning, header

end

```



```

;This script calculates the correlation or R-Layer, respectively
pro scaling, imagestack, regionalfile, outfile, tiff=tiff

;read the header, correlation layer and the image stack containing
;the regional values
if keyword_set(tiff) then begin
    ok = query_tiff(imagestack, stackinfo, GEOTIFF = geoinfo)
    imheader = geotiff2tag2envihdr(geoinfo, stackinfo,
    PROCESSING_STAGE = 'normalised')
    imheader.data_ignore_value = -32768
endif else begin
    headerpath = filext(imagestack, /PATH, /NAME) + '.hdr'
    imheader = read_envi_header_new(headerpath)
endelse

headerpath2 = filext(regionalfile, /PATH, /NAME) + '.hdr'
regheader = read_envi_header_new(headerpath2)

;cleanup
if (file_test(outfile)) then FILE_DELETE, outfile

;the image-stack is split into several strips to save memory, the
;size of each strip is calculated using the function 'blocksize'
bs = blocksize(imheader.lines, imheader.samples, imheader.bands,
imheader.data_type, MINBLOCKS= 1)

;creating the arrays for the correlation coefficient
rlayer = make_array(imheader.samples, imheader.lines, /FLOAT,
VALUE=-9999)

;read the image-stack strip-wise
for i = 0, bs.nblocks-1 do begin
;initialise arrays
    imarr = make_array(imheader.samples, bs.nrows[i], imheader.bands,
    TYPE = imheader.data_type, VALUE = -9999)
    meanarr = make_array(imheader.samples, bs.nrows[i],
    imheader.bands, TYPE = regheader.data_type, VALUE = -9999)

;read each band within the image and mean value stack
    for m = 0, imheader.bands - 1 do begin
        if ~keyword_set(tiff) then data_start = ulong64(((bs.rows[i] *
        imheader.samples) + (m * imheader.samples * imheader.lines)) *
        imheader.data_type)
        mean_start = ulong64(((bs.rows[i] * regheader.samples) + (m *
        regheader.samples * regheader.lines)) * regheader.data_type)

;image
        if ~keyword_set(tiff) then begin
            imarr[:,*,m] = read_binary(imagestack, DATA_START =
            data_start, DATA_TYPE = imheader.data_type, DATA_DIMS =
            [imheader.samples,bs.nrows[i]])
;reverse image if file was written top down (ENVI standard)
            if (total(strmatch(TAG_NAMES(imheader), 'read_order',
            /FOLD_CASE)) eq 1) then begin
                if (imheader.read_order eq 1) then imarr[:,*,m] =
                reverse(imarr[:,*,m], 2)
            endif

;swap bytes to conform with system, if necessary (byte order tag must
;be available)

```

```

        if (total(strmatch(TAG_NAMES(imheader), 'byte_order',
/FOLD_CASE)) eq 1) then begin
            case imheader.byte_order of
                1 : swap_endian_inplace, imarr[*,*,m],
                    /SWAP_IF_LITTLE_ENDIAN
                0 : swap_endian_inplace, imarr[*,*,m],
                    /SWAP_IF_BIG_ENDIAN
            else :
            endcase
        endif
    endif else begin
        temp = read_tiff(imagestack, CHANNELS = m)
        imarr[*,*,m] = temp[*,bs.rows[i]:bs.rows[i]+bs.nrows[i]-1]
    endelse

;regional values
    meanarr[*,*,m] = read_binary(regionalfile, DATA_START =
    mean_start, DATA_TYPE = regheader.data_type, DATA_DIMS =
    [regheader.samples,bs.nrows[i]])
;reverse image if file was written top down (ENVI standard)
    if (total(strmatch(TAG_NAMES(regheader), 'read_order',
/FOLD_CASE)) eq 1) then begin
        if (regheader.read_order eq 1) then meanarr[*,*,m] =
        reverse(meanarr[*,*,m], 2)
    endif

;swap bytes to conform with system, if necessary (byte order
;tag must be available)
    if (total(strmatch(TAG_NAMES(regheader), 'byte_order',
/FOLD_CASE)) eq 1) then begin
        case regheader.byte_order of
            1 : swap_endian_inplace, meanarr[*,*,m],
                /SWAP_IF_LITTLE_ENDIAN
            0 : swap_endian_inplace, meanarr[*,*,m],
                /SWAP_IF_BIG_ENDIAN
        else :
        endcase
    endif

endfor

;calculate the correlation between regional and local values, for
;each pixel in the stack
    nodata_mean = fix(regheader.DATA_IGNORE_VALUE)
    nodata_stack = fix(imheader.DATA_IGNORE_VALUE)
    for k = 0, imheader.samples - 1 do begin
        for l = 0, bs.nrows[i] - 1 do begin
            no_data_pos = WHERE((imarr[k,l,*] eq nodata_stack) or
            (meanarr[k,l,*] eq nodata_mean))
            data_pos = WHERE((imarr[k,l,*] ne nodata_stack) and
            (meanarr[k,l,*] ne nodata_mean), data_count)

            if (data_count/float(imheader.bands) gt 0.3) then begin
                local = fltarr(data_count)
                regional = fltarr(data_count)
                local[*] = float(imarr[k,l,data_pos])
                regional[*] = float(meanarr[k,l,data_pos])

                corr = correlate(regional, local)
                rlayer[k, bs.rows[i] + 1] = corr
            end
        end
    end

```

```

        endif
    endfor
endfor

endfor

paramhdr = { param_descr : 'R-Layer', $
              samples   : imheader.samples, $
              lines     : imheader.lines, $
              bands      : 1, $
              header_offset : 0, $
              file_type  : 'ENVI standard', $
              data_type  : 4, $
              interleave : 'BSQ', $
              map_info   : imheader.map_info, $
              data_ignore_value : -9999, $
              band_names : ['R-Layer'] $
            }
paramhdr.map_info.units = 'units=Degrees'

;output to hard disk
write_envi_new, outfile, rlayer, paramhdr

end

```

```

;This script calculates the distance between the line of regression
;and each data point in a time series.
pro temp_stab_regression, imagestack, regionalfile, outfile,
ortho=ortho, tiff=tiff

;read the header, correlation layer and the image stack containing
;the regional values
if keyword_set(tiff) then begin
    ok = query_tiff(imagestack, stackinfo, GEOTIFF = geoinfo)
    imheader = geotifftag2envihdr(geoinfo, stackinfo,
PROCESSING_STAGE = 'normalised')
    imheader.data_ignore_value = -32768
endif else begin
    headerpath = filext(imagestack, /PATH, /NAME) + '.hdr'
    imheader = read_envi_header_new(headerpath)
endelse

    headerpath2 = filext(regionalfile, /PATH, /NAME) + '.hdr'
    regheader = read_envi_header_new(headerpath2)

;cleanup
if (file_test(outfile)) then FILE_DELETE, outfile

;to reduce memory consumption, the image stack is processed
;stripwise, to calculate an appropriate strip size the function
;blocksize is used
bs = blocksize(imheader.lines, imheader.samples, imheader.bands,
imheader.data_type, MINBLOCKS=1)

;create the arrays for the regression coefficients
a = make_array(imheader.samples, imheader.lines, /FLOAT, VALUE=
9999)
b = make_array(imheader.samples, imheader.lines, /FLOAT, VALUE=
9999)

;read the image-stack strip wise
for i = 0, bs.nblocks-1 do begin
;initialisation of arrays
    imarr = make_array(imheader.samples, bs.nrows[i], imheader.bands,
TYPE = imheader.data_type, VALUE = -9999)
    meanarr = make_array(imheader.samples, bs.nrows[i],
imheader.bands, TYPE = regheader.data_type, VALUE = -9999)
    temp_img = make_array(imheader.samples, bs.nrows[i],
imheader.bands, /INTEGER, VALUE = -9999)

;iterate through each layer and store the strip-subset of the image
;and the mean values stack, respectively
for m = 0, imheader.bands - 1 do begin
    if ~keyword_set(tiff) then data_start = ((bs.rows[i] *
imheader.samples) + (m * imheader.samples * imheader.lines)) *
imheader.data_type
    mean_start = ((bs.rows[i] * regheader.samples) + (m *
regheader.samples * regheader.lines)) * regheader.data_type

;image
if ~keyword_set(tiff) then begin
    imarr[:,*,m] = read_binary(imagestack, DATA_START =
data_start, DATA_TYPE = imheader.data_type, DATA_DIMS =
[imheader.samples,bs.nrows[i]])

```

```

;reverse image if file was written top down (ENVI standard)
  if (total(strmatch(TAG_NAMES(imheader), 'read_order',
/FOLD_CASE)) eq 1) then begin
    if (imheader.read_order eq 1) then imarr[*,*,m] =
      reverse(imarr[*,*,m], 2)
    endif
  endif

;swap bytes to conform with system, if necessary (byte order tag must
;be available)
  if (total(strmatch(TAG_NAMES(imheader), 'byte_order',
/FOLD_CASE)) eq 1) then begin
    case imheader.byte_order of
      1 : swap_endian_inplace, imarr[*,*,m],
        /SWAP_IF_LITTLE_ENDIAN
      0 : swap_endian_inplace, imarr[*,*,m],
        /SWAP_IF_BIG_ENDIAN
    else :
    endcase
  endif
endif else begin
  temp = read_tiff(imagestack, CHANNELS = m)
  imarr[*,*,m] = temp[*,bs.rows[i]:bs.rows[i]+bs.nrows[i]-1]
endif

;regional values
meanarr[*,*,m] = read_binary(regionalfile, DATA_START =
mean_start, DATA_TYPE = regheader.data_type, DATA_DIMS =
[regheader.samples,bs.nrows[i]])
;reverse image if file was written top down (ENVI standard)
  if (total(strmatch(TAG_NAMES(regheader), 'read_order',
/FOLD_CASE)) eq 1) then begin
    if (regheader.read_order eq 1) then meanarr[*,*,m] =
      reverse(meanarr[*,*,m], 2)
    endif
  endif

;swap bytes to conform with system, if necessary (byte order tag must
;be available)
  if (total(strmatch(TAG_NAMES(regheader), 'byte_order',
/FOLD_CASE)) eq 1) then begin
    case regheader.byte_order of
      1 : swap_endian_inplace, meanarr[*,*,m],
        /SWAP_IF_LITTLE_ENDIAN
      0 : swap_endian_inplace, meanarr[*,*,m],
        /SWAP_IF_BIG_ENDIAN
    else :
    endcase
  endif
endif

endfor

;-----

;calculate the regression between regional and local values, for
;each pixel within the stack
nodata_mean = fix(regheader.DATA_IGNORE_VALUE)
nodata_stack = fix(imheader.DATA_IGNORE_VALUE)
for k = 0, imheader.samples - 1 do begin
  for l = 0, bs.nrows[i] - 1 do begin
    no_data_pos = WHERE((imarr[k,l,*] eq nodata_stack) or
(meanarr[k,l,*] eq nodata_mean))

```

```

data_pos = WHERE((imarr[k,l,*] ne nodata_stack) and
(meanarr[k,l,*] ne nodata_mean), data_count)

if (data_count/float(imheader.bands) gt 0.3) then begin
    local = fltarr(data_count)
    regional = fltarr(data_count)
    local[*] = float(imarr[k,l,data_pos])
    regional[*] = float(meanarr[k,l,data_pos])

;calculate the simple linear regression
    if ~keyword_set(ortho) then begin
        line = poly_fit(regional, local, 1, YERROR = yerror, YFIT
= yfit, YBAND = yband, /DOUBLE)
        a[k, bs.rows[i] + 1] = line[0]
        b[k, bs.rows[i] + 1] = line[1]

        residuals = local - yfit
        residuals = residuals / yerror
    endif

;if the keyword 'ortho' is set, the line regression can be calculated
;by a total least squares fit
    if keyword_set(ortho) then begin
        data = fltarr(2,data_count)
        data[0,*] = regional - mean(regional)
        data[1,*] = local - mean(local)
        dervar = pcomp(data, COEFFICIENTS = coeffs, /DOUBLE,
EIGENVALUES = eigenvals)

        line = [0, (coeffs[0,0] / coeffs[1,0])]
        yfit = line[0] + line[1]*(regional-mean(regional)) +
mean(local)
        yerror = sqrt(total((abs(yfit - (line[0] +
line[1]*local)) / sqrt(1+line[1]^2))^2) / 378.0)
        residuals = local - yfit
        residuals = residuals / yerror
    endif

    residuals = fix(round(residuals*100))
;check if pixels were omitted
    if n_elements(residuals) ne imheader.bands then begin
        residuals_full = make_array(imheader.bands, /INTEGER,
VALUE = -9999)
        for j = 0, n_elements(residuals) - 1 do begin
            residuals_full[(data_pos[j])] = residuals[j]
        endfor
    endif else residuals_full = residuals

    temp_img[k,l,*] = residuals_full
endif
endfor
endfor

;-----

;write output as BIL (band interleaved by line) file
openw, u, outfile, /APPEND, /GET_LUN
for j = 0, bs.nrows[i]-1 do writeu, u, temp_img[*,j,*]
close, u
FREE_LUN, u
DELVAR, temp_img, imarr, meanarr

```

**endfor**

```
outheader = filext(outfile, /PATH, /NAME) + '.hdr'
disthdr = { param_descr : 'Temporal Stability: Distances', $
  samples : imheader.samples, $
  lines : imheader.lines, $
  bands : imheader.bands, $
  header_offset : 0, $
  file_type : 'ENVI standard', $
  data_type : 2, $
  interleave : 'BIL', $
  map_info : imheader.map_info, $
  data_ignore_value : -9999}; $
  ;band_names : imheader.band_names $
; }
disthdr.map_info.units = 'units=Degrees'
write_envi_header_new, outheader, disthdr

parampath = filext(outfile, /PATH)
paramhdr = { param_descr : 'Regression Parameters', $
  samples : imheader.samples, $
  lines : imheader.lines, $
  bands : 2, $
  header_offset : 0, $
  file_type : 'ENVI standard', $
  data_type : 4, $
  interleave : 'BSQ', $
  map_info : imheader.map_info, $
  data_ignore_value : -9999, $
  band_names : ['a', 'b'] $
}
paramhdr.map_info.units = 'units=Degrees'
params = [[a],[b]]
write_envi_new, parampath + 'parameters', params, paramhdr
```

**end**

## 15 Appendix 3

In this section, the source code for the post-processing routine can be found:

```
;This script performs a classification and filtering of the temporal
stability output image
pro post_classification, distances, rlayer_path, outfile, thresholdv,
band

    tempstab = read_envi_new(distances)
    rlayer = read_envi_new(rlayer_path)

    rlayer = rlayer.data
    flood_img = tempstab.data[:,*,band]
    header = tempstab.hdr
    DELVAR,tempstab, rlayer

;pixels with a distance below a given threshold are classified as
;water
    water_class = (flood_img le thresholdv) and (flood_img ne -9999)

;water "islands" smaller than 6 pixels are removed
    water = where(water_class eq 1, water_count)
    while water ne !NULL do begin
        area = region_grow(water_class, water[0], THRESHOLD=[1,1],
/ALL_NEIGHBORS)
        area_mean = mean(flood_img[area])
        if (n_elements(area) lt 6) and (area_mean gt thresholdv*4)
        then water_class[area] = 0

;delete already visited elements from the array
    if (n_elements(area) gt 1) then begin
        first = 0
        for j = 0, n_elements(area) - 1 do begin
            last = N_Elements(water) - 1
            CASE where(water eq area[j]) of
                first: water = water[1:*]
                last: water = water[first:last-1]
            else: water = [water[first:where(water eq area[j])-1],
                    water[where(water eq area[j])+1:last]]
            endcase
        endfor
    endif else begin
        case n_elements(water) of
            1: water = []
            else: water = water[1:*]
        endcase
    endelse
endwhile

;no water "islands" smaller than 6 pixels are removed
    nowater = where(water_class eq 0, nowater_count)
    while nowater ne !NULL do begin
        area = region_grow(water_class, nowater[0], THRESHOLD=[0,0],
/ALL_NEIGHBORS)
        if n_elements(area) lt 6 then water_class[area] = 1
```



```

    if (n_elements(area) gt 1) then begin
        first = 0
        for j = 0, n_elements(area) - 1 do begin
            last = N_Elements(nowater) - 1
            CASE where(nowater eq area[j]) of
                first: nowater = nowater[1:*]
                last: nowater = nowater[first:last-1]
            else: nowater = [nowater[first:where(nowater eq
                    area[j])-1], nowater[where(nowater eq
                    area[j])+1:last]]
            endcase
        endfor
    endif else begin
        case n_elements(nowater) of
            1: nowater = []
            else: nowater = nowater[1:*]
        endcase
    endelse
endwhile

r_mask = where(rlayer lt 0.3)
if total(r_mask) ne -1 then water_class[r_mask] = 0

;data output to hard-disk

outhdr = { param_descr : 'Water Classification', $
            samples   : header.samples, $
            lines     : header.lines, $
            bands      : 1, $
            header_offset : 0, $
            file_type  : 'ENVI standard', $
            data_type  : 1, $
            interleave : 'BSQ', $
            map_info   : header.map_info, $
            data_ignore_value : -9999, $
            band_names  : ['water classification'] $
          }

write_envi_new, outfile, water_class, outhdr

END

```