SPATIO-TEMPORAL DATA FUSION TOOLKIT –VERSION 2

Deliverable D4.2.2

Circulation:	PU - Public
Lead partner:	TU Delft
Contributing partners:	CNR-IMATI-GE, IGN, UCL, FOMI
Authors:	Roderik Lindenbergh, Beril Sirmacek, Jinhu Wang (TU Delft) , Roberto Giachetta (Foemi), Simone Pittaluga, Michela Spagnuolo (Imati)
Quality Controller:	Andre Stumpf (UBO)
Version:	2.0
Date:	31.10.2014

Consisting of

SINTEF	STIFTELSEN SINTEF, Department of Applied Mathematics, Oslo, Norway
Fraunhofer	Fraunhofer Institute for Computer Graphics Research, Darmstadt, Germany
CNR-IMATI-GE	Institute for Applied Mathematics and Information Technologies of the National Research Council (CNR-IMATI), Genova, Italy
MOSS	M.O.S.S. Computer Grafik Systeme GmbH (MOSS), Munich, Germany
HRW	HR Wallingford Ltd (HRW), Wallingford, UK
FOMI	Hungarian National Mapping and Cadastral Agency (FÖMI), Institute of Geodesy, Cartography and Remote Sensing, Budapest, Hungary
UCL	University College London (UCL), Research centre for Photogrammetry, 3D Imaging and Metrology, London, UK
TUDelft	Delft University of Technology (TUDelft), Department of Geoscience and Remote Sensing & Man-Machine Interaction Group, Delft, The Netherlands
IGN	Institut National de l'Information Géographique et Forestière (IGN), Paris, France
UBO	Université de Bretagne Occidentale (UBO), European Institute for Marine Studies, Brest, France
Ifremer	L'Institut Francais de Recherche pour l´Exploitation de la Mer (Ifremer), Brest, France
Liguria	Regione Liguria, Genova, Italy

DOCUMENT HISTORY

Version[1]	Issue Date	Stage	Content and Changes
1.0	Oct 28 2014	Draft	
2.0	Oct 31 2014	Final	Incorporated comments from quality control

[1] Integers correspond to submitted versions

1 EXECUTIVE SUMMARY

This report describes the second version of the IQmulus *Spatio-Temporal Data Fusion Toolbox* (Deliverable D4.2.2). This toolbox actually describes a series of implemented algorithms, spread over 9 different services, developed by three different IQmulus partners, Fömi, TU Delft and Imati. In the introduction of this report the process that led to the identification of these services is described. Services are either needed by specific workflows in the IQmulus showcases, are basic services that are needed in many workflows or belong to a novel research domain, linked to IQmulus themes.

To develop these services, some use has been made of open source libraries and compilers that are therefore shortly discussed. All services have been developed in relation to certain test data that again relate to the IQmulus showcases. Also this test data is shortly described.

The main part of this toolbox report is, however, dedicated to the different services themselves. For each service a short description of its purpose and motivation of its development is given, followed by a more detailed description of the algorithm and a short manual page summarizing its basic functionality. Almost every service is demonstrated on test data. For each service it is described how its performance can be evaluated using so-called performance and evaluation criteria. For all services for which this is relevant it is in particular described what strategy is used to handle big data.

In addition, service tables are provided for each service in an appendix that lists in a uniform way all information relevant for a service. In a second appendix a glossary is included that describes technical terms.

2 TABLE OF CONTENTS

1 Executive Summary 3
2 Table of contents 4
3 Table of figures 6
4 Introduction 7
4.1 Motivation of Service Development 7
4.2 Service characteristics 8
4.3 Test results 9
5 Libraries 11
5.1 Libgeotiff 11
5.2 Libtiff 11
5.3 Mono runtime 11
6 Data Sets 12
6.1 TU Delft campus data 12
6.1.1 LMMS, Delft Campus 12
6.1.2 IPHONE IMAGE DATA SETs, DELFT 12
6.2 Fömi satellite and parcel data 13
6.3 Liguria precipitation data 14
7 Big data strategy in year 2 15
8 Outlook to the third-year Toolbox 16
9 Glossary Terms 17
10 New Services 22
10.1 Radiometric Enhancement, Fömi, #70 23
10.1.1 Motivation for inclusion 23
10.1.2 Current algorithm and possible improvements 23
10.1.3 Manual text 23
10.1.4 Big data strategy 25
10.1.5 Example 26
10.2 Convolution Filtering, Fömi, #71 27
10.2.1 Motivation for inclusion 27
10.2.2 Current algorithm and possible improvements 27
10.2.3 Manual text 28
10.2.4 Big data strategy 29
10.2.5 Examples 29
10.3 2D on 2D Image Registration, Fömi, #72 31
10.3.1 Motivation for inclusion 31
10.3.2 Current algorithm and possible improvements 31
10.3.3 Manual text 31
10.3.4 Big data strategy 34
10.3.5 Examples 34
10.4 Preprocessing of raster data (version 1), Fömi, #73 35
10.4.1 Motivation for inclusion 35
10.4.2 Current algorithm and possible improvements 35
10.4.3 Manual text, Service #73 35

10.4.4 Big data strategy ... 37
10.4.5 Examples ... 37
10.5 Topological Analysis of Land 2D data, Fömi, #74 ... 38
10.5.1 Motivation for inclusion ... 38
10.5.2 Current algorithm and possible improvements ... 38
10.5.3 Manual text ... 38
10.5.4 Big data strategy ... 39
10.5.5 Examples ... 39
10.6 Extraction of rain data, IMATI, #96 ... 40
10.6.1 Motivation for inclusion ... 40
10.6.2 Current algorithm and possible improvements ... 40
10.6.3 Manual text ... 41
10.6.4 Big data strategy ... 41
10.6.5 Example ... 41
11 Updated Services ... 43
11.1 Point cloud intersection (Service #98) ... 43
11.1.1 Motivation for inclusion ... 43
11.1.2 Current algorithm and possible improvements ... 43
11.1.3 Manual Text ... 44
11.1.4 Big data strategy ... 44
11.1.5 Examples ... 45
11.2 Geotiff to LAS, TU Delft, #99 ... 47
11.2.1 Motivation for inclusion ... 47
11.2.2 Current algorithm and possible improvements ... 47
11.2.3 Manual Text ... 47
11.2.4 Big data strategy ... 47
11.2.5 Example ... 47
11.3 Registering 2D images on point clouds, TU Delft, #100 ... 49
11.3.1 Motivation for inclusion ... 49
11.3.2 Current algorithm and possible improvements ... 49
11.3.3 Manual Text ... 49
11.3.4 Big data strategy ... 50
11.3.5 Example ... 50
12 References ... 51
13 Appendix: Service Tables ... 52
13.1 Radiometric Enhancement, Fömi, #70 ... 52
13.2 Convolution Filtering, Fömi, #71 ... 54
13.3 2D on 2D Image Registration, Fömi, #72 ... 56
13.4 Top of the atmosphere refelectance, Fömi, #73 ... 59
13.5 Topological Analysis of 2D data, Fömi, #74 ... 61
13.6 Extraction of rain data, IMATI, #96 ... 63
13.7 point cloud Intersection, TU Delft, #98 ... 65
13.8 Geotiff to LAS, TU Delft, #99 ... 67
13.9 Registering 2D images on point clouds, TU Delft, #100 ... 69

3 TABLE OF FIGURES

Figure 1: LMMS dataset TU Delft campus 12

Figure 2: Photos from the Iphone data set sampling the TU Delft OTB building, Delft, The Netherlands 13

Figure 3: Photos from the Iphone data set of the TU Delft CITG building, The Netherlands. 13

Figure 4: Photos from the Iphone data set of the TU Delft architecture building, The Netherlands. 13

Figure 5: Photos from the Iphone data set sampling a residential building in Delft, The Netherlands. 13

Figure 6: Greyscale version of a SPOT 5 spectral image used to generate the examples in Services #70 to #72. 14

Figure 7: Liguria cumulative precipitation over a 24 hours period. (Image source: http://www.severe-weather.eu/) 14

Figure 8: Voxelized point cloud. 15

Figure 9: Example results of scaling test on two different data sets. The horizontal axis indicates the number of points in the point cloud considered, the vertical axis is the running time in seconds. 16

Figure 10: Radiometric enhancement techniques 26

Figure 11: Filtering methods 30

Figure 12: Image registration 34

Figure 13: Example of input dataset: precipitation on Septmber 29, 2013 between 00.00 and 00.15 AM. 42

Figure 14: Example of output dataset: time series of precipitation organized per weather station. 42

Figure 15: Experimental computation time as function of input data size given as a number of points. 45

Figure 16: Airborne laser data (green and blue) and LMMS data (white) before intersection 46

Figure 17: Left: AHN2 airborne (white) and LMMS data (green) after intersection. Right: point cloud distances between airborne and LMMS on the intersected area. 46

Figure 18: Left, input 2D geotiff; Right, output 3D las file 48

Figure 19: 2D images projected on a 3D facade of a point cloud of a TU Delft campus building 50

4 INTRODUCTION

In this Introduction a short overview is given of the status of the work within Task 4.2 that eventually lead to this toolbox report. The general scope of Task 4.2 has already been described in last years' Toolbox Report, D.4.2.1.

4.1 MOTIVATION OF SERVICE DEVELOPMENT

In the second year of the IQmulus project, service development has been driven by the three showcases Marine, Land and Urban. Within each showcase several workflows are defined that are described in D1.2.3. Therefore all new services described in this Toolbox report were initiated by analysing the needs in these workflows.

Services #70, #71, #72, #73 and #74 are related to Integrated Land Showcase Workflow 3 (LS3): Flood and Waterlogging detection. In addition, service #71 is also related to the Integrated Land Showcase Workflow 4 (LS4): Detection and characterization of landslides, while service #72 is related to LS4 as well, and also to the Marine Showcase Workflow 4 (MS4): Measuring submarine dune migration. In particular Service #72 replaces Service #96 mentioned in D.1.23. The development of Service #93 has been cancelled since Service #72 matrured more rapidly and already covers all functionalities required for the implementation of the showcases. Service #96, Extraction of rain data is related to the Land Showcase, Workflow 2, (LS2), Precipitation Analysis. An overview of the new services together with responsible partners is given in Table 1: Newly Developed Services

TABLE 1: NEWLY DEVELOPED SERVICES

Number	Service Name	Lead	Showcase
#70	Radiometric Enhancement	Fömi	LS3
#71	Convolution Filtering	Fömi	LS3, LS4
#72	2D on 2D Image Registration	Fömi	LS3, MS4
#73	Pre-processing of raster data	Fömi	LS3
#74	Topological Analysis of land 2D data	Fömi	LS3
#96	Extraction of rain data	IMATI	LS2

In addition, three of the services released in project year 1 have been renewed. An existing service is typically renewed because a new algorithm is implemented that outperforms the previous version in terms of quality of the results or computational efficiency. The three renewed services are summarized in Table 2: Updated Services.

It should be noticed that the existing services that are no further described in this Toolbox Report are still subject to (minor) change, which often are quite time consuming. Bugs may be identified because of internal WP4 testing or because of the testing procedures from WP7. Also the further tuning of service development as described in D4.1.2 (Development Guidelines for Data Integration and Processing – Version 3) have lead to changes in the online help descriptions, the requirements on the targeted operating system and in the release cycle. The

release cycle refers to the processing starting from identifying the need of a service in a showcase, via its approval and development towards internal testing and documentation. This process is finalized by a so-called smoke test in which the task leader runs the service on the Fraunhofer cloud.

TABLE 2: UPDATED SERVICES

Number	Service Name	Lead	Showcase
#98	Point Cloud Intersection	TU Delft	urban
#99	Geotiff to LAS	TU Delft	urban
#100	Registering 2D images on point clouds	TU Delft	urban

4.2 SERVICE CHARACTERISTICS

With respect to the first years toolbox some changes in the service descriptions have been made. The use of each service is this year in addition described by a man page, which will in principle also be used in the integrated prototype. For evaluating the performance of a Service this year metrics are required that describe the robustness, scalability and accuracy of the result of a service. For accuracy think for example on the average distance between two point clouds after their mutual alignment while scalability can be described in theory by the computational complexity of the underlying algorithm and in practice by a scalability test. Finally, for each Service for which this is relevant a strategy is described on how to operate the Service on big data. In the individual service descriptions below, for each service a motivation for inclusion is given, a description of the underlying algorithm, the manual text, the big data strategy and some typical example. The metrics are included in the Service Tables in the Appendix.

The status at the time of this writing of the services covered in this Toolbox Report is summarized in Table 3. For all services at least a prototype is running. Services #96 and #98 passed the smoke test, which means that they are released, although for both services the handling of unexpected input should be further improved. Services #70 to #74, which are designed in a similar way seemed OK, but faced an unexpected error during the smoke test on the Fraunhofer cloud. Finally Services #99 and #100 were developed under Windows and work fine there, but need fine tuning of their compilation under linux.

TABLE 3: SERVICE RELEASE STATUS

Service	Language	Meets IQmulus Architecture	Execution nodes	Metadata	Github	Smoke test	Released
#70	C#	yes	single	yes	yes	fail	no
#71	C#	yes	single	yes	yes	fail	no
#72	C#	yes	single	yes	yes	fail	no
#73	C#	yes	single	yes	yes	fail	no
#74	C#	yes	single	yes	yes	fail	no
#96	Python	yes	single	yes	yes	pass[2]	yes
#98	C++	yes	single	yes	yes	pass[2]	yes
#99	C++	No, windows	single	yes	yes	-	no
#100	C++	yes	single	yes	no	-	no

4.3 TEST RESULTS

In the second year of IQmulus also the testing of the IQmulus services started with Individual Component Testing, compare D7.2. In this paragraph we shortly list some feedback we got on the TU Delft Year 1 services and the action the TU Delft developers took accordingly. Note that feedback was directly provided to the developers, therefore at the time of this writing, feedback provided to Task 4.2 developers outside TU Delft is not available to the author.

SERVICE 8 POINT CLOUD TO POINT CLOUD DISTANCE
Remark: the GUI is too complicated to use. A command line function is recommended.
Action: a comman line linux version has been developed meanwhile

SERVICE 11 SPATIAL EXTENT
Remark: The algorithm doesn't consider points with elevation below 0.
Action: This will be fixed
Remark: The program cannot handle unexpected user entries.
Action: Algorithms will from now on be tested unexpected arguments, both in number and in syntax

SERVICE 17, 3D KEYPOINT EXTRACTION
Remark: the program cannot handle unexpected user entries.
Action: Algorithms will from now on be tested unexpected arguments, both in number and in syntax

SERVICE 31, POINT CLOUD GENERATION FROM MULTIVIEW IMAGES

[2] Works fine, but handling exceptions should still be improved

Remark: the more images you use, the better results you get
Action: we will improve the documentation and possibly also let the program itself give feedback on this point.
Remark: results are as expected.
Action: none
Remark: the program cannot handle unexpected user entries.
Action:. Algorithms will from now on be tested unexpected arguments, both in number and in syntax

5 LIBRARIES

In the following, we briefly mention two libraries that were used for the first time in the second year of IQmulus within Task 4.2. For libraries used before the reader is referred to the Y1 report.

5.1 LIBGEOTIFF

Libgeotiff is a library is an open source library normally hosted on top of libtiff for reading, and writing GeoTIFF information tags. This library is used in Service #99.

5.2 LIBTIFF

Libtiff is an open source library which provides support for reading and writing Tag Image File Format (TIFF) images which is a widely used format for storing data. This library is used in Service #99.

5.3 MONO RUNTIME

The mono runtime allows to directly run programs developed under windows on a linux machine. This environment is used in the Services #70 to #74.

6 DATA SETS

Here an overview is given of new data sets used during the development of Year 2 Task 4.2 services. Again, the reader is referred to D4.2.1 for descriptions of data sets already used during the development in IQmulus Year 1.

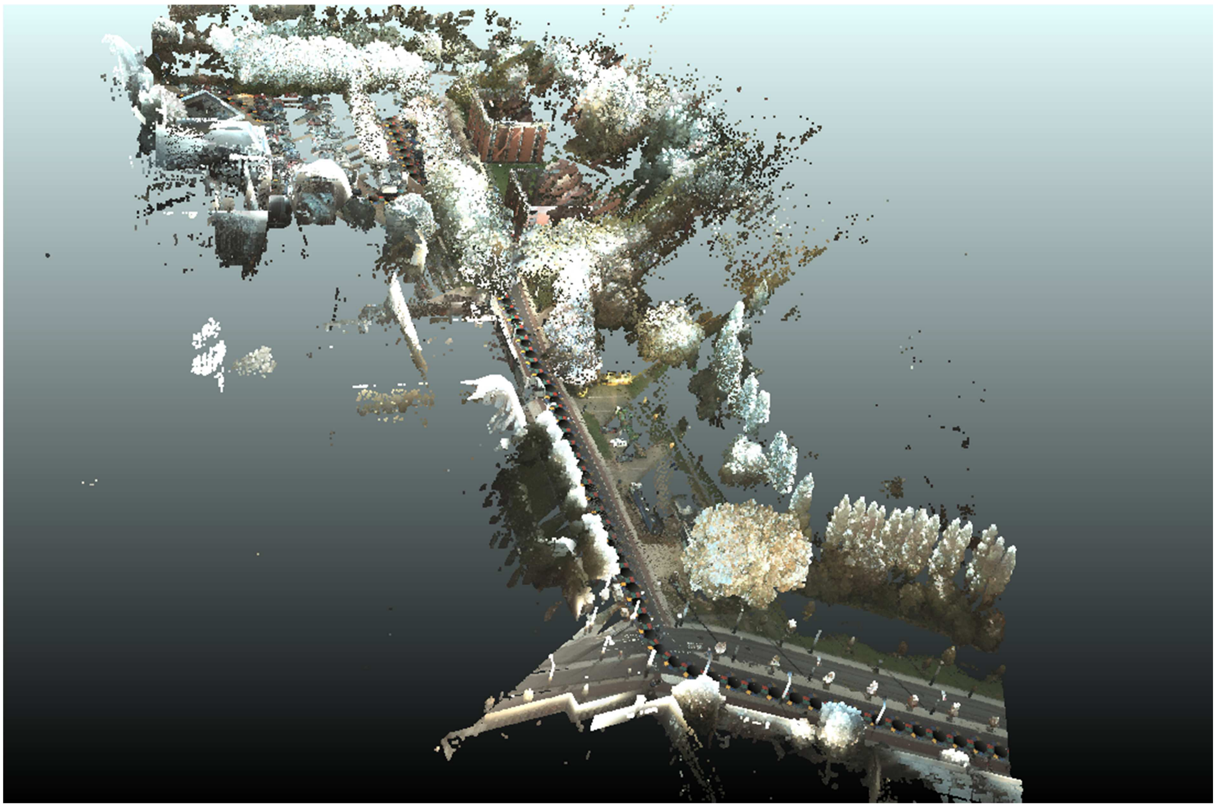
6.1 TU DELFT CAMPUS DATA

All kind of different data sets are available for the TU Delft campus area, making it a good test case for IQmulus Servise development. Below a mobile mapping data set and several image data sets are introduced.

6.1.1 LMMS, Delft Campus

This data set is used to test the services with ID numbers; #11, #98, #17, #18, #26. It consists of 53 958 666 points in total and covers an area of approximately 750 x 1200 m. It is in *.las and *.laz. format and came in 509 tiles of 25 x 25m. It has been captured by the Drivemap laser mobile mapping system of Fugro.

FIGURE 1: LMMS DATASET TU DELFT CAMPUS



6.1.2 IPHONE IMAGE DATA SETs, DELFT

In Figure 2 to Figure 5 we show multi-view photosets sampling buildings at the TU Delft campus obtained by IPhone.

FIGURE 2: PHOTOS FROM THE IPHONE DATA SET SAMPLING THE TU DELFT OTB BUILDING, DELFT, THE NETHERLANDS

FIGURE 3: PHOTOS FROM THE IPHONE DATA SET OF THE TU DELFT CITG BUILDING, THE NETHERLANDS.

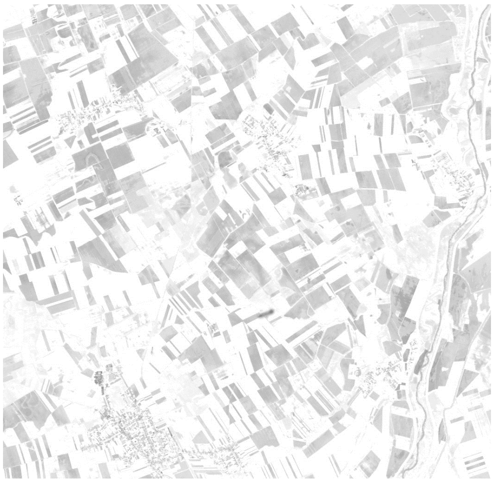
FIGURE 4: PHOTOS FROM THE IPHONE DATA SET OF THE TU DELFT ARCHITECTURE BUILDING, THE NETHERLANDS.

FIGURE 5: PHOTOS FROM THE IPHONE DATA SET SAMPLING A RESIDENTIAL BUILDING IN DELFT, THE NETHERLANDS.

6.2 FÖMI SATELLITE AND PARCEL DATA

The services working on raster data (i.e. Services #70, #71, #72 and #73) have been tested on Orthophoto (dataset #5, high resolution RGB images), Landsat 8 (dataset #47, high resolution multispectral satellite images) and SPOT 5 imagery (high resolution multispectral satellite images containing 4 bands). All images have a file size of up to 300MB in GeoTIFF format without compression. Service #74 (Spatial/topological operations on 2D data) has been tested on Shapefiles extracted from the Land Parcel Dataset (dataset #4) with file size up to 50MB. The example images present in the description are extracted from a SPOT 5 image, see also Figure 6

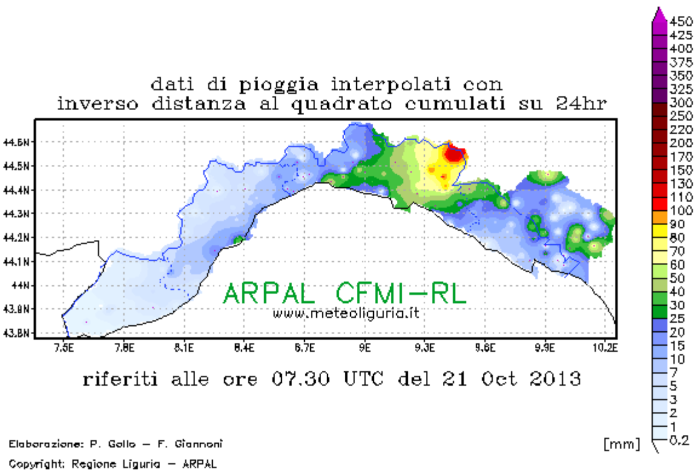
FIGURE 6: GREYSCALE VERSION OF A SPOT 5 SPECTRAL IMAGE USED TO GENERATE THE EXAMPLES IN SERVICES #70 TO #72.



6.3 LIGURIA PRECIPITATION DATA

The precipitation data used in Service #96 is described in detail in D1.2.3. To summarize, it consists of precipitation data of the regional Arpal network, which spans 145 stations, and of precipitation data of the Genova municipality network, which has 30 stations. Cumulative precipitation over Liguria at October 21, 2013 is illustrated in Figure 7.

FIGURE 7: LIGURIA CUMULATIVE PRECIPITATION OVER A 24 HOURS PERIOD. (IMAGE SOURCE: HTTP://WWW.SEVERE-WEATHER.EU/).

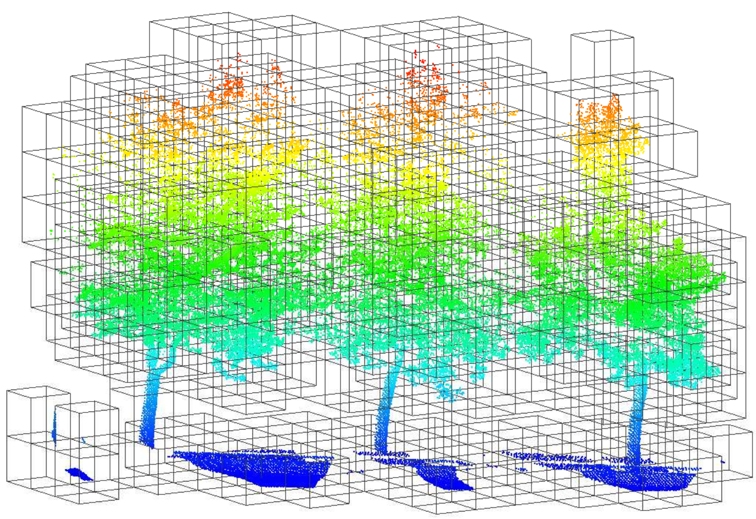


7 BIG DATA STRATEGY IN YEAR 2

In the second year of IQmulus the need to operate on big data became more urgent. Showcases were developed around big data examples and cloud infrastructure became available that actually enables the processing of big data. Finally big data sets as available to some IQmulus partners were shared among partners. In the first version of this Toolbox, Hadoop was used as a strategy to partition large data in manageable chunks for efficient cloud processing by partner UCL. Typically a Service based on Hadoop aims at processing all available data (for example all points in a point cloud) albeit in an efficient way. For the services #70 to #74 described in this Toolbox report a Hadoop implementation is under development by partner Fömi. These Hadoop versions will be released in project year 3. These services all consider spectral satellite image analysis.

For point cloud data also an alternative strategy will be considered. Typically, point clouds are highly redundant. A point cloud obtained by a laser scanner representing a straight wall may exists of millions of points, where three points suffice to define position and orientation of the wall plane. Therefore an alternative strategy to process big data is to only consider that part of the data that suffices to extract the information required. On way to do so is by using voxels. Voxels are regular 3D cubes that are used to subdivide the spatial extent of a point cloud in units of for example 25 cm, compare Figure 8. In this years Toolbox TU Delft experiments with voxel processing by designing a new algorithm for one new Service, Point cloud intersection, Service #98. The result of voxel processing is scalable in the sense that finer results are obtained at higher computational costs using smaller voxels. An effective way of processing could therefore be to work from coarse to fine. More information on this specific algorithm can be found in the Service descripton below.

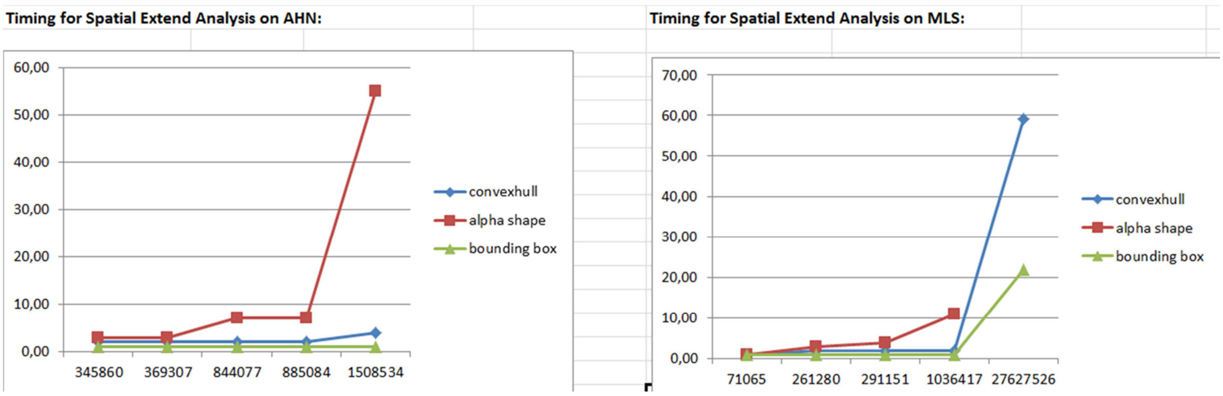
FIGURE 8: VOXELIZED POINT CLOUD.



Within IQmulus we have defined big data not only as data that is large in size, but also data that is complex or heterogeneous in nature. Heterogeneous data is notably relevant for this Toolbox, as here data fusion is considered. Service #100, registering 2D images on point clouds is an example where two data sets of completely different nature are connected.

In the second year of IQmulus also the performance of services started to be evaluated. A test that is very relevant when working with big data is to check how the performance evaluates when increasing the input size. To perform such a test in practice a convenient way is to start with a large data set, and create randomly thinned versions, consisting of 10%, 1% and 0.1% random points of the original data set. The results of such a test on AHN airborne laser scan data and LMMS mobile mapping data, both sampling part of the TU Delft campus are visualized in Figure 9. Both results show that performance strongly reduces at higher data sizes, but one should be careful in drawing conclusion from such figures. Often performance is most affected by the computer's internal memory, and a good solution to increase performance would be to adapt a suitable data partitioning strategy. This particular test was performed on a standard desktop computer

FIGURE 9: EXAMPLE RESULTS OF SCALING TEST ON TWO DIFFERENT DATA SETS. THE HORIZONTAL AXIS INDICATES THE NUMBER OF POINTS IN THE POINT CLOUD CONSIDERED, THE VERTICAL AXIS IS THE RUNNING TIME IN SECONDS.



8 OUTLOOK TO THE THIRD-YEAR TOOLBOX

We conclude here with a ahort outlook towards the third-year Toolbox. During the second year some essential developments took place that will have their impact on further service development and maintenance. Notably all newly released and existing services run together on one environment, the Fraunhofer IQmulus cloud. This will enable developers and additional IQmulus users to easily evaluate workflows consisting of a series of Services. It is expected that such evaluation will expose possible weak parts in services which will indicate to developers where to focus their efforts. In addition much more services will be parallelized, as described above in Chapter 7, and more focus will be on how services should interact with solutions for storing and querying big data sets.

9 GLOSSARY TERMS

Below a list of glossary terms is given. The final goal of this list is to provide non-expert end users of the IQmulus system with understandable explanations of IQmulus functionalities as implemented by the services described in this toolbox report. The explanations in the glossary list will be made visible to the end users by means of the Domain Specific Language, compare deliverable D3.3 on the DSL tool chain prototype release. It should be noted that the list below is only a first starting point. The list was compiled by the developers and no feedback from end users has been collected so far. The terms in the list were selected by analysing the Showcase descriptions in deliverable D1.2.3 and by analysing the Service tables in the Appendix below. Further explanation on the glossary can also be found in the Development Guidelines for Data Integration and Processing, D4.1.3.

3D Data	Observations in 3D space
3D Data acquisition	The process of obtaining observations in 3D space
Alpha shape	An alpha shape is a family of piecewise linear simple curves in the Euclidean plane associated with a set of points. The alpha shape associated with a set of points is a generalization of the concept of the convex hull, i.e. every convex hull is an alpha shape but not every alpha shape is a convex hull.
Bounding box	The smallest enclosing box which includes all points of a point cloud inside of it. For a point set (S) in N dimensions, it refers to the box with the smallest measure within which all the points lie.
Calibrated	An instrument is calibrated if the measurements it collects are accurate (i.e. not biased).
Camera pose	Camera pose identification is also known as extraction of extrinsic parameters of a camera which means determination of the camera location and orientation with respect to the other cameras (if exist) or the objects in the input data.
Cloud to cloud distance	Distance between the closest points between two different point clouds in an intersection area.
Cloud-mask	A binary image denoting whether the specified area is unobstructed between the surface and the remote sensing device (usually satellite). The mask is applied in case of image processing algorithms, which result can be influenced in case the clouded area is also processed. The cloud mask can be creating using a variety of algorithms, for example thresholding in different spectral domains.
Cluster of points	A group of 2D or 3D points in a point cloud which belong together according to a proximity criterion.

Convex hull	Convex hull is a set of points in the Euclidean space which has the smallest convex set that it can contain.
Co-register	To co-register is to apply any method for aligning images or point clouds.
Data extent	The range of the data values in each of its dimensions.
Data set	A collection of numerical values carrying information on a process or object
Geotiff	Geotiff refers to Tiff file which has geographic (or cartographic) data embedded as tags within the Tiff file. The geographic data can then be used to position the image in the correct location and geometry on the screen of a geographic information display.
Gridding	Generating a 2D or 3D mesh which has constant size intervals in every axis. Gridding generally refers to assigning input data points or pixels into the mesh which they fall into.
ICP	ICP (Iterative Closest Point) is an algorithm for minimizing the difference between two point clouds. The algorithm keeps one of the point clouds fixed and transforms the coordinates ofo the other point cloud to find the best match. The algorithm iteratively revises the transformation (translation and rotation) needed to minimize the distance between two point clouds, compare [8].
Image	A matrix which holds 2D earth coordinates of the measurement samples (pixels) and measurement attributes for each sample.
Isocontour	Contour line of points or pixels in the data which contain the same attribute value (generally height).
Isosurface	An isosurface is a three-dimensional surface layer which consists of points with the same attribute value.
Key point	3D point or 2D pixel coordinates of the samples which indicate important descriminitive characteristics of the surface or the object which is observed.
Land Parcel Identification System (LPIS)	The land-parcel identification system (LPIS) is a system to identify land use for a given country using orthophotos and high resolution satellite imagery. A unique number is given to each land parcel to provide a unique identification in space and time. In Hungary, the software is titled MePAR, and came into operation in 2004.
LAS	A LAS file format is a public file format for the interchange of three-dimensional point cloud data

	between data users. The LAS file format is a binary file format that maintains information specific to the lidar nature of the data while not being overly complex.
LIDAR data	LIDAR (Light Detection and Ranging) is a remote sensing method that uses light in the form of a pulsed laser to measure ranges (variable distances) to the Earth. These light pulses generate precise, three-dimensional information about the shape of the Earth and its surface characteristics.
Matching	Registering vectors of one data set to the vectors of another data set by considering the similarities of some certain attributes.
Multi-angle images	Multi angle images are the images which are taken from different location with different looking angle to the same region or object of interest.
Multi-temporal	Data which is collected at different times over approximately or exactly the same land/object coverage area.
NDSI	The Normalized Difference Soil Index is a spectral index to delineate soil areas in the remotely sensed image. NDSI is computed as a combination of short-wave infrared (SWIR) and near infrared (NIR) values.
NDVI	The Normalized Difference Vegetation Index is a spectral index to assess the live green vegetation of the specified area. NDVI is computed as a combination of near infrared (NIR) and visible red values (NDVI is functionally, but not linearly, equivalent to the near infrared, red ratio).
NDWI	The Normalized Difference Water Index is a spectral index to delineate open water features of the specified area. NDWI is computed as a combination of short-wave infrared (SWIR) and visible red values.
Optical image	An image that is created by using an imaging device, such as camera capturing the electromagnetic waves in different domains of the electromagnetic spectrum. The domains are typically in the infrared, visible and ultraviolet range, which can all be divided into multiple subdomains, resulting in multiple bands within the image (known as multispectral image).
Point cloud	A set of points in a 3D coordinate system typically sampling the surface of a solid object. Point clouds considered here are obtained by laser scanning, multi-view photogrammetry or echo sounding
Point cloud intersection	The group of points from two or more different point

	clouds which sample the same geographical area on earth surface.
Point set	A collection of points describing the outline of an object in 2D or 3D
Pre-process	Pre-process refers to algorithms which apply certain desired modifications on the raw input data such that the input data meet requirements of other algorithms.
Registration	The process of transforming data sets from different coordinate systems into one common coordinate system. It is a necessary step for comparing data or visualizing them together in a realistic way.
Satellite images	Satellite imagery refers to remote sensing images collected by artificial satellites. Different satellite missions and sensors can provide remote sensing imagery for various purposes with various properties (spatial, spectral, radiometric and geometric resolution).
Shape	the form of an object
SONAR data	Sonar (Sound NAvigagtion and Ranging) data is collected by measuring the reflected sound from a surface. This data acquisition method is mostly used for scanning the bottom layer of sea and ocean.
Spatial extent	Structural measures extracted from a rigid form input based on a shape measuring criterion (i.e. convex hull, alpha shape, bounding box etc.)
Spectral Indices	Indices are graphical indicators that can be used for analyzing remote sensing measurements. Several indices are available for measurement of different characteristics, such as vegetation health and status or fire severity.
Subsampling	Reducing the resolution by choosing only one value to represent the samples in larger sampling area.
Survey coverage	The spatial extent sampled by a survey
Survey number	Identificaton number of a specific survey
Threshold	A certain value to judge the input information as accepted or rejected.
Tie point	3D point or 2D pixel coordinates which are used to calculate translation and rotation matrices to register two input data in one common coordinate system.
Tile	One piece of identically shaped areas which all together cover the input data completely.
ToA reflectance	Top of Atmosphere Reflectance represents the solar

	radiation incident on the satellite sensor in standard unitless terms, independent of the position of the sun with respect to the earth. The ToA reflectance is computed as a preprocessing step in multiple satellite image analysis operations to obtain better results in further processing steps.
Tolerance	Accepted limit of variation in data values.
Topographic Map	A Topographic map is a detailed and accurate map of a certain region representing topographic features such as hills, rivers, roads and houses.
Topographic Object	Objects on the earth surface with known elevation values.

10 NEW SERVICES

In this chapter the IQmulus Task 4.2 services that were newly developed in project year 2 are described in detail. For each service the following information is provided:

- *Motivation of inclusion;* Describing why a particular service is developed, connecting to the showcases.
- *Current algorithm and possible improvements;*
- *Manual text;* This is the manual text that will be provided to the technical users within the IQmulus system.
- *Big data strategy;* What strategy is adapted now or near future to cope with big data.
- *Examples;* On notably IQmulus data.

In addition, for each service a service table is provided in the Appendix that contains a full description of all aspects of a service. The ID of each service is used for internal communication purposes, and it refers to the ID used in the eRoom table of services.

10.1 RADIOMETRIC ENHANCEMENT, FÖMI, #70

Radiometric enhancement or simply image enhancement is the process of making an image more interpretable for a particular application and to highlight particular features of the image [1]. The enhancement alters the image based on the values of individual pixels, usually interpreting the values separately for each image band. Thus, radiometric enhancement of a multispectral image can be usually considered as a series of independent band enhancements.

10.1.1 Motivation for inclusion

Integrated Land Showcase, Workflow 3 (LS3): Flood and Waterlogging detection.

10.1.2 Current algorithm and possible improvements

The main goal of radiometric enhancement is the alteration of the image histogram in order for specific regions of intensity to be more distinguishable, thus usually creating more contrast. The enhancement techniques for the first version include the following methods, compare also Figure 10.

- *Linear contrast enhancement*: linear enhancement refers to the affine transformation of image histogram using an additive (a) and a multiplicative (b) value, which are applied for all intensity values.

$$I'(x,y) = aI(x,y) + b$$

- *Saturating contrast enhancement*: as in case of linear enhancement, the histogram is transformed in an affine manner, but values are automatically computed using histogram properties so that the transformation is optimal (no values are excluded).

$$I'(x,y) = \frac{I(x,y) - I_{min}}{I_{max} - I_{min}} \cdot 2^R$$

- *Histogram equalization*: this method effectively spreads the most frequent intensity values by using the cumulative distribution function (cdf) of the histogram. This method usually creates the highest contrast within the image.

$$I'(x,y) = \frac{cdf\big(I(x,y)\big) - cdf(I_{min})}{cdf(I_{max}) - cdf(I_{min})} \cdot 2^R$$

- *Inversion*: this method inverts spectral intensities, creating a negative image. As it does not modify image histogram or image contrast, it is primarily used in case of human interpretation.

Future versions will include more techniques and the ability to execute the operations in an Apache Hadoop cloud environment.

10.1.3 Manual text

Radiometric enhancement service, Version 1 (FÖMI)

DESCRIPTION
Radiometric enhancements transform image histogram as a preprocessing operation for further analysis. The technique can differ, and there are also special forms of enhancement.

Version 1 contains the following methods: linear contrast enhancement, saturating contrast enhancement, histogram equalization and inversion.

INSTALLATION
1. Install Mono 3.8 or .NET Framework 4.5 (or later).
2. No additional installation of the service is required: just copy the content of the "bin" folder. All used libraries are included.

USAGE
This is a command line (console) executable.
Run "AEGIS.Console.RadiometricEnhancement.exe" with .NET Framework (Windows), or "mono AEGIS.Console.RadiometricEnhancement.exe" with Mono (Linux, OS X, Windows).

Execution is fully automated, only errors and help information are displayed during execution. If no messages are displayed, execution is successful.
Options must be included beforehand.
The following options available:

-help	Display help text.
-input	Specifies the input file path. (Only TIFF/GEOTIFF format is supported.)
-output	Specifies the output file path. (Only TIFF/GEOTIFF format is supported.)
-method	Specifies the enhancement method. (See below.)
-band-index	Specifies the zero based band index which should be processed. (Optional, if none specified, all bands will be processed. Indexing is zero based, thus the first band has index of 0.)
-offset	Specifies the offset used in linear contrast enhancement. (Optional, the default value is 0.)
-factor	Specifies the factor used in linear contrast enhancement. (Optional, the default value is 1.)

The following methods are supported:
- linear contrast enhancement
 (using the manually specified offset and factor values for all pixels)
- histogram equalization
(automated enhancement using the commutative distribution function)
- saturating contrast enhancement
 (automated enhancement using affine transformation of the histogram)
- inversion
(for creating a negative image)

ERROR CODES
Beside the error message, the following exit codes are used for notification:

1	Error during parsing of options (e.g. required options not specified or do not have the right format).
2	Error during input reading (e.g. file was not found, or is not in a supported format).
3	Error during operation execution (e.g. options do not have the right value).
4	Error during output writing (e.g. output directory does not exist).

EXAMPLES

1. mono AEGIS.Console.RadiometricEnhancement.exe -input="/user/me/data/input.tif" -output="/user/me/data/output.tif" -method="histogram equalization"
Specifies that all bands of the input should be equalized (individually), thus the output will have the same number of bands as the source.
2. mono AEGIS.Console.RadiometricEnhancement.exe -input="/user/me/data/input.tif" -output="/user/me/data/output.tif" -method="linear contrast enhancement" -offset=30 -band-index=0
Specifies that the first band of the input should be processed by adding 30 to all values. The output image will only have the specified band.

INPUT DATA
Example input file is available in the "data" folder.

SOURCE CODE
The complete source code including unit tests is available in the "src" folder. The solution file "sln" can be opened using Visual Studio 2013.

CONTACT
Roberto Giachetta (groberto@inf.elte.hu)

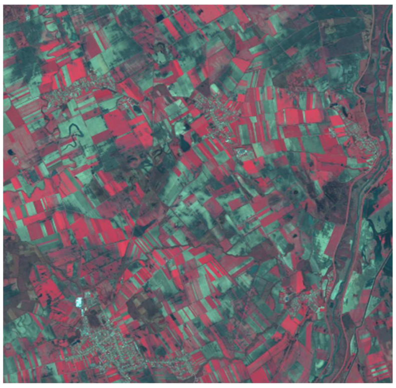
10.1.4 Big data strategy

The version of the services #70 to #74 described here are non-parallel prototypes. The parallel/distributed versions of the services working with Hadoop MapReduce will be released during IQmulus Year 3.

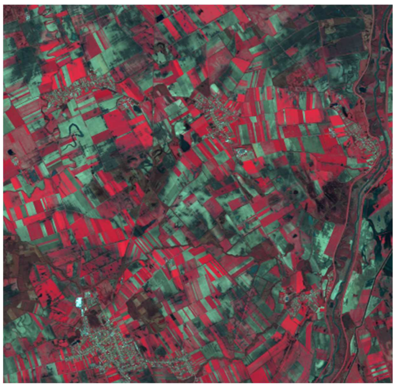
10.1.5 Example

An example is shown in Figure 10.

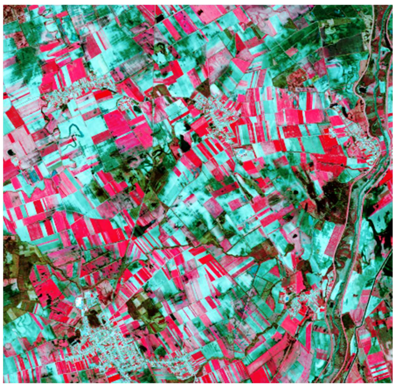
FIGURE 10: RADIOMETRIC ENHANCEMENT TECHNIQUES



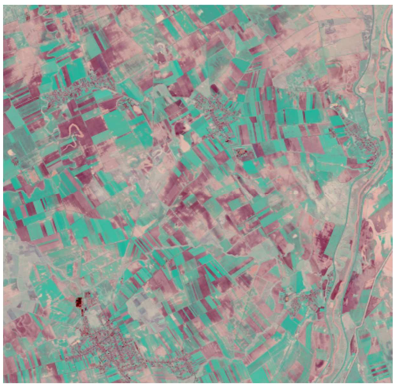
a) Original image



b) Saturating contrast enhancement



c) Histogram equalization



d) Inversion

10.2 CONVOLUTION FILTERING, FÖMI, #71

Image filtering is the application of various effects on raster imagery [1]. The type of filter determines the impact on the image that can range from image smoothing through edge detection to any specific application.

10.2.1 Motivation for inclusion

Integrated Land Showcase Workflow 3 (LS3): Flood and Waterlogging detection; Integrated Land Showcase Workflow 4 (LS4): Detection and characterization of landslides.

10.2.2 Current algorithm and possible improvements

The filtering is performed by performing convolution of the image and an odd sized matrix, which is known as kernel ($k \in \mathbb{R}^{(2n-1)\cdot(2n-1)}$). The matrix is applied over each pixel of the image, and all values underlying the matrix are multiplied by the matrix value. The central pixel under the matrix is then replaced by the sum of computed values. Hence, the spectral information of neighbouring pixels influences the value of the central pixel.

$$I'(x,y) = \sum_{i=1}^{2n-1} \sum_{j=1}^{2n-1} I(x-n+i, y-n+j) \cdot k[i,j]$$

The first version of the service includes the following filters (see Figure 2).

- *Box*: The box, box blur or mean filter the simplest matrix that computes the mean of filter values. It can be used as a fast approximation of the Gaussian-blur, as no actual matrix is required to be computed.
- *Gaussian-blur*: this filter performs blurring an image by a Gaussian function. It is used for enhancing image structures, removing noise, and it reduces the image's high-frequency components.
- *Median*: The median filter simply selects the median of the underlying values. It is primarily used for smoothening, as under some conditions, it preserves edges while removing noise.
- *Usharp masking*: An image sharpening filter that uses a blurred (unsharp) positive image to create a mask of the original image. The unsharped mask is then combined with the negative image, creating an image that is less blurry than the original.
- *Mean removal*: This image filter sharpens the image by changing the values of neighboring pixels with approximately the same value, so that small differences are evened out.
- *Sobel*: An edge detection algorithm that computes an approximation of the gradient of the image intensity function at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction
- *Prewitt*: Another edge detection algorithm using the same approach as the Sobel filter, with a slightly different kernel.

Further version will include more methods and the ability to execute operation in Apache Hadoop cloud environment.

10.2.3 Manual text

CONVOLUTION FILTERING SERVICE, VERSION 1 (FÖMI)

DESCRIPTION
In convolution filtering an odd sized template (known as kernel) is applied to image regions, and the central spectral value is computed based on the neighbouring values. By using different kernels, several results can be obtained, for instance, smoothing, sharpening, edge detection, line detection, etc.

INSTALLATION
1. Install Mono 3.8 or .NET Framework 4.5 (or later).
2. No additional installation of the service is required, just copy the content of the "bin" folder. All used libraries are included.

USAGE
This is a command line (console) executable.
Run "AEGIS.Console.Filtering.exe" with .NET Framework (Windows), or "mono AEGIS.Console.Filtering.exe" with Mono (Linux, OS X, Windows).

Execution is fully automated, only errors and help information are displayed during execution. If no messages are displayed, execution is successful.
Options must be included beforehand.

The following options available:

-help	Display help text.
-input	Specifies the input file path. (Only TIFF/GEOTIFF format is supported.)
-output	Specifies the output file path. (Only TIFF/GEOTIFF format is supported.)
-method	Specifies the filtering method. (See below.)
-band-index	Specifies the zero based band index which should be filtered. (Optional, if none specified, all bands will be filtered.)
-radius	Specifies the radius of the filter (when applicable). (Optional, the default value is 1.)
-gaussian-standard-deviation	Specifies the standard deviation value for the Gaussian blur filter. (Optional, the default value is 1.)
-sharpening-amount	Specifies the amount of sharpening using unsharp masking. (Optional, the default value is 0.8.)
-sharpening-threshold	Specifies the threshold at which sharpening takes effect using unsharp masking. (Optional, the default value is 0.)
-weight	The weight of the central value in mean removal. (Optional, the default value is 1.)

The following methods are supported:
- box (with specified radius)
- median (with specified radius)
- gaussian-blur (with specified radius and std. deviation)
- unsharp-masking (with specified amount and threshold)

- mean-removal (with specified weight)
- prewitt
- sobel

ERROR CODES
Beside the error message, the following exit codes are used for notification:

1 Error during parsing of options
(e.g. required options not specified or do not have the right format).

2 Error during input reading
(e.g. file was not found, or is not in a supported format).

3 Error during operation execution
(e.g. options do not have the right value).

4 Error during output writing
(e.g. output directory does not exist).

EXAMPLES

1. mono AEGIS.Console.Filtering.exe -input="/user/me/data/input.tif" -output="/user/me/data/ouput.tif" -method="prewitt"
Specifies that band 1 (second band) of input.tif should be filtered with the Prewitt filter.

2. mono AEGIS.Console.Filtering.exe -input="/user/me/data/input.tif" -output="/user/me/data/ouput.tif" -method="gaussian-blur" -band-index=1 -radius=2 -gaussian-standard-deviation=1.7
Specifies that band 1 (second band) of input.tif should be filtered with Gaussian blur of 2 radius and 1.7 std. deviation.

INPUT DATA
Example input file is available in the "data" folder.

SOURCE CODE
The complete source code including unit tests is available in the "src" folder. The solution file "sln" can be opened using Visual Studio 2013.

CONTACT
Roberto Giachetta (groberto@inf.elte.hu)

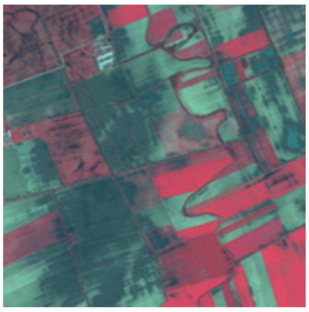
10.2.4 Big data strategy

The version of the services #70 to #74 described here are non-parallel prototypes. The parallel/distributed versions of the services working with Hadoop MapReduce will be released during IQmulus Year 3.

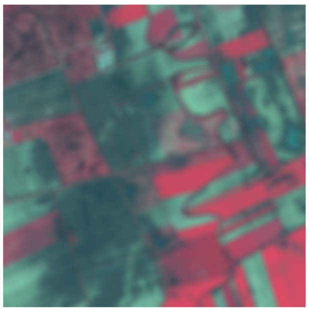
10.2.5 Examples

Some examples of Service #71 are shown in Figure 11.

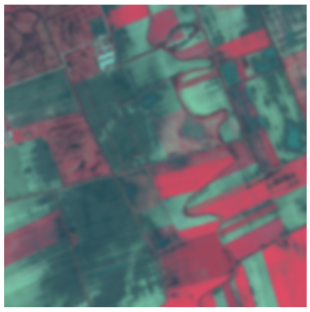
FIGURE 11: FILTERING METHODS



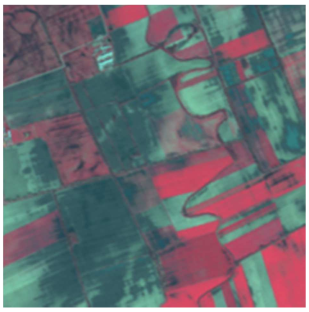
a) original image



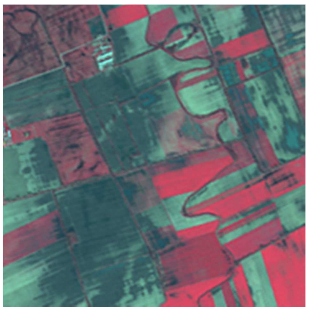
b) box filter



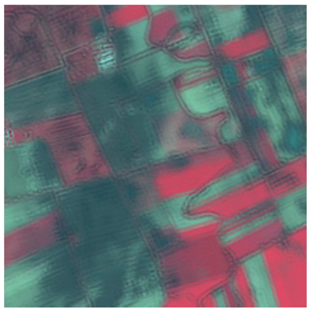
c) Gaussian filter



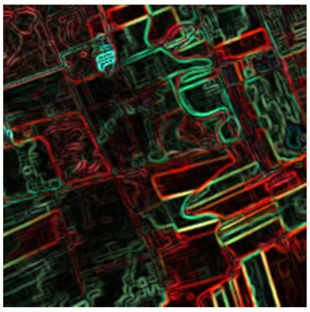
d) median filter



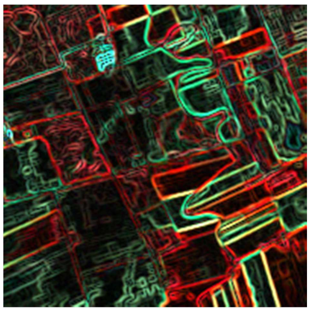
e) unsharp masking



f) mean removal



e) Prewitt filter



f) Sobel filter

10.3 2D ON 2D IMAGE REGISTRATION, FÖMI, #72

In image registration, the task is to transform images into a single spatial reference system [1]. Either the imagery is not initially referenced, or the reference system of the image does not match that of the dataset, the image has to be transformed in order to be processable with other data available in the dataset (see Figure 3).

10.3.1 Motivation for inclusion

Integrated Land Showcase Workflow 3 (LS3): Flood and Waterlogging detection; Integrated Land Showcase Workflow 4 (LS4): Detection and characterization of landslides; Marine Showcase Workflow 4 (MS4): Measuring submarine dune migration.

10.3.2 Current algorithm and possible improvements

The simplest method for performing registration is the allocation of ground control points (GCP), which have coordinates within the image space, and coordinates in the destination reference system. Another approach is to provide referenced imagery witch is matched to the source image using pattern recognition, resulting in GCP-s for the matched coordinates (this functionality is not supported in the initial release). The transformation can be computed by polynomial approximation of the GCP-s. In case the source data is referenced, the transformation can be computing using formula of the specified reference systems.

Registration is followed by resampling the image, which can also be performed independently, when scaling of the image is required. Resampling is performed using interpolation. The first version of the service supports three interpolation techniques:

- *nearest neighbour*, which simply uses the nearest pixel value;
- *bilinear*, which computes the linear combination of four neighbouring pixels;
- and *Lanczos* [3], which applies the Lanczos kernel to the values $L(x) = \mathrm{sinc}(x)\,\mathrm{sinc}(x/a)$.

Further version will include more resampling methods, automated registration using reference imagery and the ability to execute operation in Apache Hadoop cloud environment.

10.3.3 Manual text

Image registration and resampling service, Version 1 (FÖMI)

DESCRIPTION
Image registration is the process of transforming different sets of data into one reference system (by using ground control points or reference imagery).
Registration is followed by resampling (scaling) with multiple supported interpolations.
Version 1 contains control point based registration and reference system matching with linear, bilinear and Lanczos resampling techniques.

INSTALLATION
1. Install Mono 3.8 or .NET Framework 4.5 (or later).
2. No additional installation of the service is required, just copy the content of the "bin" folder. All used libraries are included.

USAGE

This service is provided with two executables. One executable for resampling only, and one for registration and resampling.
Both are command line (console) executables.

RESAMPLING
Run "AEGIS.Console.Resampling.exe" with .NET Framework (Windows), or "mono AEGIS.Console.Resampling.exe" with Mono (Linux, OS X, Windows).

Execution is fully automated, only errors and help information are displayed during execution. If no messages are displayed, execution is successful.
Options must be included beforehand.

The following options are available:

-help	Display help text.
-input	Specifies the input file path. (Only TIFF/GEOTIFF format is supported.)
-output	Specifies the output file path. (Only TIFF/GEOTIFF format is supported.)
-method	Specifies the resampling method. The following methods are supported: nearest neighbour, bilinear and Lanczos. If none specified, nearest neighbour resampling is used.
-number-of-rows or -rows	Specifies the numbers of rows (height) in the output image. If not specified, the value is computed based on the number of columns to preserve aspect ration.
-number-of-cols or -cols	Specifies he number of columns (width) in the output image. If not specified, the value is computed based on the number of rows to preserve aspect ration.

REGISTRATION
Run "AEGIS.Console.Registration.exe" with .NET Framework (Windows), or "mono AEGIS.Console.Registration.exe" with Mono (Linux, OS X, Windows).

Execution is fully automated, only errors and help information are displayed during execution. If no messages are displayed, execution is successful.
Options must be included beforehand.

The following options are available:

-help	Display help text.
-input	Specifies the input file path. (Only TIFF/GEOTIFF format is supported.)
-output	Specifies the output file path. (Only TIFF/GEOTIFF format is supported.)
-reference-system or ref	The EPSG reference system identifier of the target system. If not provided, reference system information will not be stored.
-ground-control-point or -gcp	Specifies a ground control point on the input image. Control points are specified as a sequence of 4 numbers divided by semicolon (row, column, X, Y).
-column-width or -colw	Specifies the width of the column in the output image with respect to the reference system. If not specified, it is automatically computed based on the input image.
-row-height or -rowh	Specifies the height of the row in the output image with respect to the reference system. If not specified, it is automatically computed based on the input image.

-method	Specifies the resampling method. The following methods are supported: nearest neighbour, bilinear and Lanczos. If none specified, nearest neighbour resampling is used.

Registration uses either reference system matching or ground control points (GCP). If at least one GCP is provided, GCP will be used. At least 3 control points must be specified, if less are specified, error will occur.
If no GCP is specified, reference system matching will match the source reference system to the target. If the source image has no reference system, an error will occur.

ERROR CODES
Beside the error message, the following exit codes are used for notification:

1 Error during parsing of options
(e.g. required options not specified or do not have the right format).
2 Error during input reading
(e.g. file was not found, or is not in a supported format).
3 Error during operation execution
(e.g. options do not have the right value).
4 Error during output writing
(e.g. output directory does not exist).

EXAMPLES

1. mono AEGIS.Console.Resampling.exe -input="/user/me/data/input.tif" -output="/user/me/data/output.tif" -rows=500 -cols=1000

   Specifies that the input image should be resized to 1000x500 pixels using nearest neighbour resampling method.

2. mono AEGIS.Console.Resampling.exe -input="/user/me/data/input.tif" -output="/user/me/data/output.tif" -method="lanczos" -rows=350

   Specifies that the input image should be resized using the Lanczos method to 350 pixel height and the width should be computed to preserve aspect ratio.

3. mono AEGIS.Console.Registration.exe -input="/user/me/data/input.tif" -output="/user/me/data/output.tif" -ref="23700" -method="bilinear"

   Specifies that the input image should be transformed into HD72 / EOV reference system using reference system matching and bilinear resampling. The dimensions of the output image will match the dimensions of the input.

4. mono AEGIS.Console.Registration.exe -input="/user/me/data/input.tif" -output="/user/me/data/output.tif" -ref="23700" -gcp="0;0;15000;5430" -gcp="0;100;15200;5430" -gcp="100;0;15000;5330" -rowh=2 -colw=2 –method ="lanczos"

   Specifies that the input image should be transformed into HD72 / EOV reference system based on coordinates for pixels (0,0), (0,100) and (100, 0) using Lanczos resampling with 2 row height and 2 column width. Thus, the output image will be half as high as the original.

INPUT DATA
Example input file is available in the "data" folder.

SOURCE CODE
The complete source code including unit tests is available in the "src" folder. The solution file "sln" can be opened using Visual Studio 2013.

CONTACT
Roberto Giachetta (groberto@inf.elte.hu)

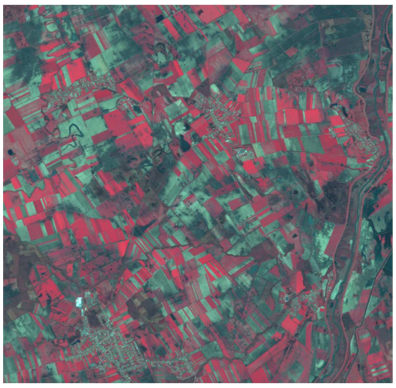
10.3.4 Big data strategy

The version of the services #70 to #74 described here are non-parallel prototypes. The parallel/distributed versions of the services working with Hadoop MapReduce will be released during IQmulus Year 3.

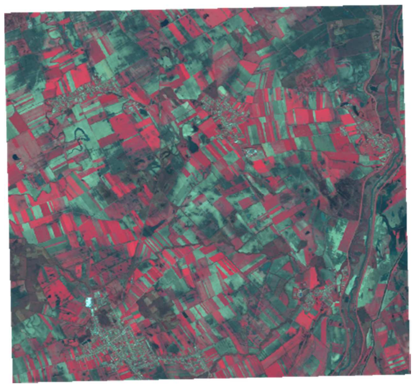
10.3.5 Examples

An example of the result of the image registration service is shown in Figure 12.

FIGURE 12: IMAGE REGISTRATION



a) original image



b) referenced image

10.4 PREPROCESSING OF RASTER DATA (VERSION 1), FÖMI, #73

Preprocessing is a complex operation that is provided to enable multiple operations to be executed in a single process.

10.4.1 Motivation for inclusion

Integrated Land Showcase Workflow 3 (LS3): Flood and Waterlogging detection.

10.4.2 Current algorithm and possible improvements

The executable operations contain radiometric enhancements, geometric corrections, removal of atmospheric effects and instrumental errors, image registration and resampling. Multiple operations can be specified using methods defined in services #70, #71 and #72.

Further versions of the service include an automation process that detects and applies the required corrections based on the image metadata (e.g. mission and sensor type, imaging time and location, sensor information). To enable the automated correction, a predefined procedure must be provided for the specified sensor.

10.4.3 Manual text, Service #73

Preprocessing of raster data, Version 1 (FÖMI)

DESCRIPTION
Corrections of radiometric and geometric properties must be applied to all remotely sensed images before further processing and analysis can be performed. These corrections include the removal of atmospheric effects, instrumental errors and geometric effects (such as earth rotation, curvature and panoramic distortion).

INSTALLATION
1. Install Mono 3.8 or .NET Framework 4.5 (or later).
2. No additional installation of the service is required, just copy the content of the "bin" folder. All used libraries are included.

USAGE
This is a command line (console) executable.
Run "AEGIS.Console.Preprocessing.exe" with .NET Framework (Windows), or "mono AEGIS.Console.Preprocessing.exe" with Mono (Linux, OS X, Windows).

Execution is fully automated, only errors and help information are displayed during execution. If no messages are displayed, execution is successful.
Options must be included beforehand.

The following options available:

-help	Display help text.
-input	Specifies the input file path. (Only TIFF/GEOTIFF format is supported.)
-output	Specifies the output file path. (Only TIFF/GEOTIFF format is supported.)
-method	Specifies the preprocessing method. Multiple methods can be specified. (See below.)

-band-index	Specifies the zero based band index which should be processed. (Optional, if none specified, all bands will be processed. Indexing is zero)
-offset	For enhancement. Specifies the offset used in linear contrast enhancement. (Optional, the default value is 0.)
-factor	For enhancement. Specifies the factor used in linear contrast enhancement. (Optional, the default value is 1.)
-radius	For filtering. Specifies the radius of the filter (when applicable). (Optional, the default value is 1.)
-gaussian-standard-deviation	For filtering. Specifies the standard deviation value for the Gaussian blur filter. (Optional, the default value is 1.)
-sharpening-amount	For filtering. Specifies the amount of sharpening using unsharp masking. (Optional, the default value is 0.8.)
-sharpening-threshold	For filtering. Specifies the threshold at which sharpening takes effect using unsharp masking. (Optional, the default value is 0.)
-weight	For filtering. The weight of the central value in mean removal. (Optional, the default value is 1.)
-reference-system or ref	For registration. The EPSG reference system identifier of the target system. If not provided, reference system information will not be stored.
-ground-control-point or -gcp	For registration. Specifies a ground control point on the input image. Control points are specified as a sequence of 4 numbers divided by semicolon (row, column, X, Y).
-column-width or -colw	For registration. Specifies the width of the column in the output image with respect to the reference system. If not specified, it is automatically computed based on the input image.
-row-height or -rowh	For registration. Specifies the height of the row in the output image with respect to the reference system. If not specified, it is automatically computed based on the input image.
-resampling-method	For registration and resampling. Specifies the resampling method. The following methods are supported: nearest neighbour, bilinear and Lanczos. If none specified, nearest neighbour resampling is used.
-number-of-rows or -rows	For resampling. Specifies the numbers of rows (height) in the output image. If not specified, the value is computed based on the number of columns to preserve aspect ration.
-number-of-cols or -cols	For resampling. Specifies he number of columns (width) in the output image. If not specified, the value is computed based on the number of rows to preserve aspect ration.

The following methods are supported:
- linear contrast enhancement (using the manually specified offset and factor values for all pixels)

- histogram equalization
- saturating contrast enhancement
- inversion
- registration
- resampling
- box filter (with specified radius)
- median filter (with specified radius)
- gaussian-blur filter (with specified radius and std. deviation)
- unsharp-masking filter (with specified amount and threshold)
- mean-removal filter (with specified weight)
- prewitt filter
- sobel filter

ERROR CODES
Beside the error message, the following exit codes are used for notification:

1 Error during parsing of options
(e.g. required options not specified or do not have the right format).
2 Error during input reading
(e.g. file was not found, or is not in a supported format).
3 Error during operation execution
(e.g. options do not have the right value).
4 Error during output writing
(e.g. output directory does not exist).

EXAMPLES
mono AEGIS.Console.Preprocessing.exe -input="/user/me/data/input.tif" -output="/user/me/data/ouput.tif" -method="histogram equalization" -method="unsharp-masking filter" -method="resampling" -resampling-method="lanczos" -rows=100 -cols=100

Specifies that input.tif will be preprocessed by applying histogram equalization, unsharp-masking, and finally resampling to 100x100 pixels using Lanczos resampling.

INPUT DATA
Example input file is available in the "data" folder.

SOURCE CODE
The complete source code including unit tests is available in the "src" folder. The solution file "sln" can be opened using Visual Studio 2013.

CONTACT
Roberto Giachetta (groberto@inf.elte.hu)

10.4.4 Big data strategy

The version of the services #70 to #74 described here are non-parallel prototypes. The parallel/distributed versions of the services working with Hadoop MapReduce will be released during IQmulus Year 3.

10.4.5 Examples

No example

10.5 TOPOLOGICAL ANALYSIS OF LAND 2D DATA, FÖMI, #74

Spatial operations on 2D vector data is a common feature in most geospatial toolkits. The methods are described in the OGC Simple Feature Access standard [4] and are composed of query (e.g. equals, intersects, touches) and analysis (e.g. union, difference, buffer) operations.

10.5.1 Motivation for inclusion

Integrated Land Showcase Workflow 3 (LS3): Flood and Waterlogging detection.

10.5.2 Current algorithm and possible improvements

The operations are computed by transforming the vector geometries to a topological representation, such as the half-edge model. The half-edge data model stores the faces, edges, vertices and half-edges of the geometries and their topological structure in a planar graph. As the name implies, a half-edge is a half of an edge and is constructed by splitting an edge down its length. In order to transform a geospatial dataset, the clipping of geometries in the dataset must be evaluated first, which can be performed using the Bentley-Ottmann [5] and Weiler-Atherton [6] algorithms.

After the construction of two (or multiple) topological half-edge graphs, their merge can easily be performed through applying the before mentioned algorithms similarly. By tagging each vertex in the result graph to which input graph(s) did it belong, the topological relations between the inputs graphs can easily be determined afterwards based on a similar model like the (Dimensionally Extended) Nine-Intersection Model (DE-9IM).

The first version of the service includes topological relationship queries, including equals, intersects, disjoint, touches, crosses, within, and overlaps. The result is the relation between the specified geometries. Further versions will include analysis operations, support for topological model of raster data (in form of a hybrid raster-vector topology model), and cloud based execution of operations stored within the Hadoop Distributed File System. Additionally, caching support will be enabled to increase the processable data amount.

10.5.3 Manual text

Spatial/topological operations on 2D data sets, Version 1 (FÖMI)

DESCRIPTION
The topological representation and analysis of 2D vector geometries is a common feature in most geospatial toolkits. Combined with the analysis of raster data, this forms a hybrid raster-vector topology model. On the model, several queries (equals, intersects, overlaps, …) and analysis operations (distance, intersection, buffer, …) can be performed.

INSTALLATION
1. Install Mono 3.8 or .NET Framework 4.5 (or later).
2. No additional installation of the service is required, just copy the content of the "bin" folder. All used libraries are included.

USAGE
This is a command line (console) executable.
Run "AEGIS.Console.Topology.exe" with .NET Framework (Windows), or "mono AEGIS.Console.Topology.exe" with Mono (Linux, OS X, Windows).

Execution is fully automated, only errors and help information are displayed during execution. If no messages are displayed, execution is successful.
Options must be included beforehand.

The following options available:

-help	Display help text.
-input1	Specifies the first input file path. (Only Shapefile format is supported.)
-input2	Specifies the second input file path. (Only Shapefile format is supported.)
-operation	Specified the topological analysis operation.

The following operations are supported: equals, disjoint, intersects, touches, crosses, within, contains, overlaps.
The return value of the service contains the result of the operations, namely:
0 false
1 true

ERROR CODES
Beside the error message, the following exit codes are used for notification:
2 Error during parsing of options
(e.g. required options not specified or do not have the right format).
3 Error during input reading
(e.g. file was not found, or is not in a supported format).
4 Error during operation execution
(e.g. options do not have the right value).
5 Error during output writing
(e.g. output directory does not exist).

EXAMPLE
mono AEGIS.Console.Topology.exe -input="/user/me/data/input1.shp" -
output="/user/me/data/input2.shp" -operation="overlaps"

Checks for any overlaps between shapes of input1.shp and input2.shp.

INPUT DATA
Example input file is available in the "data" folder.

SOURCE CODE
The complete source code including unit tests is available in the "src" folder. The solution file "sln" can be opened using Visual Studio 2013.

CONTACT
Roberto Giachetta (groberto@inf.elte.hu)

10.5.4 Big data strategy

The version of the services #70 to #74 described here are non-parallel prototypes. The parallel/distributed versions of the services working with Hadoop MapReduce will be released during IQmulus Year 3.

10.5.5 Examples

No example

10.6 EXTRACTION OF RAIN DATA, IMATI, #96

This service preprocess the rainfall data gathered from the meteorological service and prepares the dataset for further elaborations.

10.6.1 Motivation for inclusion

Integrated Land Showcase, Precipitation analysis (LS2).

10.6.2 Current algorithm and possible improvements

The input data sets are in a comma separated value format, where each rainfall data is recorded in a row. Each row has seven fields:

ID of weather station (ANAG.CODE); time stamp (DTRF); rainfall (RAINC); name of weather station (NAME); longitude (LON); latitude (LAT); elevation (ELEV);

Some fields, like the geographical position, are always the same for a given weather station but are repeated for every recorded time step. The aim of this service is to optimize these data by making them easily readable for other services. The service reads the input and groups the rainfall observations per weather station ID. Each observation is identified by a timestamp; The service computes the cumulative rainfall over a range of time steps as selected by the user.

The output of the service is an enriched point set, where each row has a set of j+4 fields, where j is the number of time steps. The first four fields are fixed and related to the description of the weather station (ANAG.CODE LAT LON ELEV); the others are the j rainfall observations RAINC_j.

Libraries used

The software is implemented using Python without external modules.

Underlying Algorithm

The service execution is described by the following pseudo code:

```
# Read the file name of the rainfall data and the time step
# (in minutes) for computing the cumulative rainfall;
Read_input (In_file_name,time_step);

# Load the input file in a 2D dictionary where the primary key is the ID of
# weather station (ANAG.CODE) and the secondary key is the timestamp (DTRF)
Load(In_file_name)

# For each weather station output the ID and the geographical information
Foreach ANAG.CODE
      Output ANAG.CODE LAT LON ELEV
      Foreach DTRF

            # For each timestep compute the cumulative rainfall data and
            # output it
            Compute_the_cumulative_rainfall
            Output cumulative_rainfall
```

10.6.3 Manual text

NAME
extractrainfall.py – convert the rainfall data provided by Regione Liguria Meteorological service to a format more suitable for further elaboration and compute the cu
mulative rainfall.

SYNOPSIS
[python] extractrainfall.py input_filename cumulative_time_step [time_span]

DESCRIPION
input_filename : the name of rainfall data file

cumulative_time_step: the size of the time interval over the service compute the cumulative rainfall. Set the parameter to one to output the data without compute the
cumulative rainfall.

time_span: is the interval of time where the service have to compute the cumulative rainfall. It is expressed as STARTDATATIME/ENDDATATIME; DATATIME must be express a
s YYYYMMDDHHMM

The output file have the same name of the input but the service append the suffix _processed at the name, and it is placed automatically in the folder OUTPUT that mus
t be yet present.

EXAMPLE
python extract_raindata.py DATASET/observed_rain_2013092900_24h.csv 60

or for TEST:
python extract_raindata.py TEST/small_dataset.csv 5

the output must have the same rainfall values of the input

10.6.4 Big data strategy

As weather stations are relative sparse, a big data strategy is not required for this particular service with respect to data size. It does contribute though to the aspect of data complexity by making heterogeneous data less complex.

10.6.5 Example

The functionality of the service is demonstrated in the two figures below. In Figure 13 input data as obtained for a given time step is shown. For each station a acronym, a time stamp, the amount of precipitation, the full station name, the stations location and its elevation with respect to a reference geoid is given. In Figure 14 precipitation observations from one day are summarized in one row per station.

FIGURE 13: EXAMPLE OF INPUT DATASET: PRECIPITATION ON SEPTMBER 29, 2013 BETWEEN 00.00 AND 00.15 AM.

```
ANAG.CODE;DTRF;RAINC/10;NAME;LON/100000;LAT/100000;ELEV
CALGR;201309290000;0;Calice Ligure - Ca rosse;8.30205;44.19573;50
CAMPL;201309290000;0;Campo Ligure;8.6939;44.54256;335
CARPE;201309290000;0;Carpe;8.16447;44.14819;610
CARRO;201309290000;0;Carro;9.59772;44.27525;437
CASON;201309290000;0;Casoni di Suvero;9.76581;44.30542;1070
BELEN;201309290000;0;Colle Belenda ;7.69983;43.98152;1357
BESTA;201309290000;0;Bestagno;8.0007;43.9333;281
BMARO;201309290000;0;Borgomaro;7.95315;43.97713;250
BONUO;201309290000;0;Borgonuovo;7.62177;43.84803;115
BRZON;201309290000;0;Borzone;9.41013;44.42429;386
```

FIGURE 14: EXAMPLE OF OUTPUT DATASET: TIME SERIES OF PRECIPITATION ORGANIZED PER WEATHER STATION.

```
BESTA 8.0007 43.9333 281.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2.0 2.2
SEGIA 9.18206 44.48758 895.0 0.0 0.0 0.0 5.2 26.2 2.0 0.0 0.0 0.0 0.0 0.0 0.0 0.2 1
SCIAR 8.62464 44.40645 235.0 0.0 0.2 1.2 3.0 7.2 1.0 0.2 0.2 0.2 0.0 0.0 0.0 0.4 18
MIGNA 8.93816 44.54028 270.0 0.0 0.0 0.0 0.0 1.2 0.6 0.2 0.0 1.4 0.2 0.0 0.0 0.0 0.
GEQUE 8.9726 44.42367 200.0 0.0 0.0 0.0 0.2 0.2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.4 0.8
MTECA 8.95422 44.54856 620.0 0.0 0.0 0.0 0.2 1.4 1.0 0.0 0.0 0.4 0.2 0.0 0.0 0.0 0.
VREGI 8.94744 44.52038 407.0 0.0 0.0 0.0 0.0 0.8 0.6 0.2 0.2 0.0 0.0 0.0 0.0 0.0 0.
MROCC 9.93842 44.07129 405.0 0.0 0.2 0.0 0.2 2.0 6.6 1.2 0.0 0.0 0.0 0.8 0.0 0.8 0.
CISNE 8.15823 44.08053 40.0 0.2 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.2 7.4
SEBOR 7.6985 43.82756 550.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.2 0.0
TRIOR 7.76242 43.99451 789.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.
CCHER 9.32194 44.42107 615.0 0.0 0.0 0.4 0.8 1.0 0.0 0.0 0.0 0.0 0.0 2.4 0.2 0.4 3.
ALPIC 8.52604 44.40667 435.0 0.0 0.4 25.0 20.4 6.6 6.2 0.4 0.0 0.0 0.0 0.0 0.0 0.4
PREMA 9.00911 44.41681 234.0 0.0 0.0 0.0 0.0 0.2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.4 1.
PRTTI 9.27102 44.39365 70.0 0.0 0.0 3.0 0.8 0.8 0.0 0.0 0.0 0.0 0.0 3.6 0.0 0.8 3.8
MAGSG 9.89656 44.36182 180.0 0.0 0.0 0.2 0.4 2.4 5.8 3.8 0.0 0.0 0.2 0.2 0.6 3.2 1.
MTPOP 9.16912 44.32709 610.0 0.0 0.0 0.0 0.2 1.0 0.2 0.0 0.0 0.0 0.0 0.0 1.8 4.2 0.0 3.0 3.
```

11 UPDATED SERVICES

In this chapter the IQmulus Task 4.2 services that got a major update in project year 2 are described. The presented information is the same as for the new services.

11.1 POINT CLOUD INTERSECTION (SERVICE #98)

Given two spatial point cloud datasets A and B, the dataset intersection consists of the region covered by both datasets A and B.

11.1.1 Motivation for inclusion

A first version of this service, Service #12, was already released in project year 1. This is a basic service that is for example used for change detection and for the evaluation of a registration at overlapping areas. It can be used in different showcases but notably in the urban showcase. The implementation in Year 1 used a brute force approach which was computationally not efficient enough. To improve on this year a new algorithm was developed based on voxels.

11.1.2 Current algorithm and possible improvements

The current algorithm is based on voxels. First, voxelization is performed on each of the two imported point cloud datasets A and B, resp. Second, inclusion relations are analysed and stored for all eight vertices of each voxel of point cloud A in B, and vice versa. Next, voxels are categorized into two classes, inside and outside of the area of intersection. Finally, points corresponding to voxels inside the intersection are obtained and exported.

In detail, the steps of the algorithm are as follows:

1. A pair of overlapping point clouds (Source and Target respectively) is loaded into the system.
2. The user specifies a voxel size or a default size is provided by the algorithm.
3. Both point clouds are subdivided into voxels of a given size. This subdivision is driven by the bounding box of each data set. In general the subdivision is not the same.
4. For all the voxel cells in the Source dataset, the relative position of their vertices within the Target voxels is determined: Source vertices, which are inside a target voxel cell are labelled +1; The other source vertices are labelled -1.
5. A source cell is considered to be in the intersection if the majority of its eight vertices is labelled positive;
6. All points in the voxel cell are then exported as belonging to the area of intersection.
7. After that the same procedure is repeated with Source and Target interchanged.

Note that the last stap is required as the subdivision of the two input point clouds is in general not the same. A possible improvements would be to consider three types of voxels. Currently only voxels inside and outside the intersection are distuinguished. In addition voxels partly in- or outside could be maintained in an additional category on which further processing is performed with a smaller voxel size (at the cost of a longer computational). What should be notable considered is how the algorithm should cope with data sets that are subdivided into tiles. In principle it easy to parallelize this approach.

11.1.3 Manual Text

NAME
intersection -- determines the area of intersection of two overlapping point clouds

SYNOPSIS
Intersection [OPTIONS] [input-file-1][input-file-2][output-file-1][output-file-2]

DESCRIPTION
The two output files are the points from the two input files in the overlapping region

Both input and output are *las files

-voxelsize specifies the voxel size, if not specified a default value is used
-help displays this manual page

CONTACT
Jinhu Wang, Jinhu.Wang@tudelft.nl

11.1.4 Big data strategy

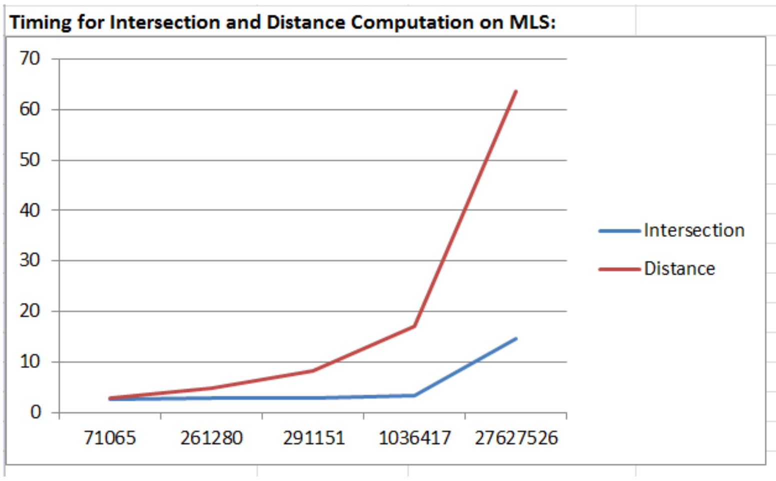
The strategy to use big data in this service is to use voxels. In this way not all individual point cloud points have to be processed.

For understanding big data processing possibilities, it is important to check how the computation time scales with the data size. For this purpose, we have applied intersection and distance computation tests using AHN and LMMS point clouds of different sizes. As can be seen in Figure 15 the computation time seems to increase exponentially with the size of the input data, as specified in Table 4. Therefore, intelligent tiling methods will be considered in addition to avoid overloading of the memory.

TABLE 4: INTERSECTION (SERVICE #98) AND DISTANCE (SERVICE #8) COMPUTATION TIME (IN SECONDS) FOR DIFFERENT LMMS POINT CLOUD SIZES.

Number of LMMS Points	Intersection (Seconds)	Distance (Seconds)
71065	2,694	2,8
261280	2,892	4,876
291151	2,946	8,206
1036417	3,251	16,976
27627526	14,568	63,561

FIGURE 15: EXPERIMENTAL COMPUTATION TIME AS FUNCTION OF INPUT DATA SIZE GIVEN AS A NUMBER OF POINTS.



11.1.5 Examples

Figure 16 shows example input data for determining the intersection from two overlapped point cloud datasets. The white point cloud is point cloud data of TU Delft campus obtained with the LMMS from Fugro. While the green point cloud data is AHN2 airborne point cloud data. The two point cloud data is overlapped and the service is employed to determine the intersected region. The result of the intersection service is visualized in Figure 17.

FIGURE 16: AIRBORNE LASER DATA (GREEN AND BLUE) AND LMMS DATA (WHITE) BEFORE INTERSECTION.

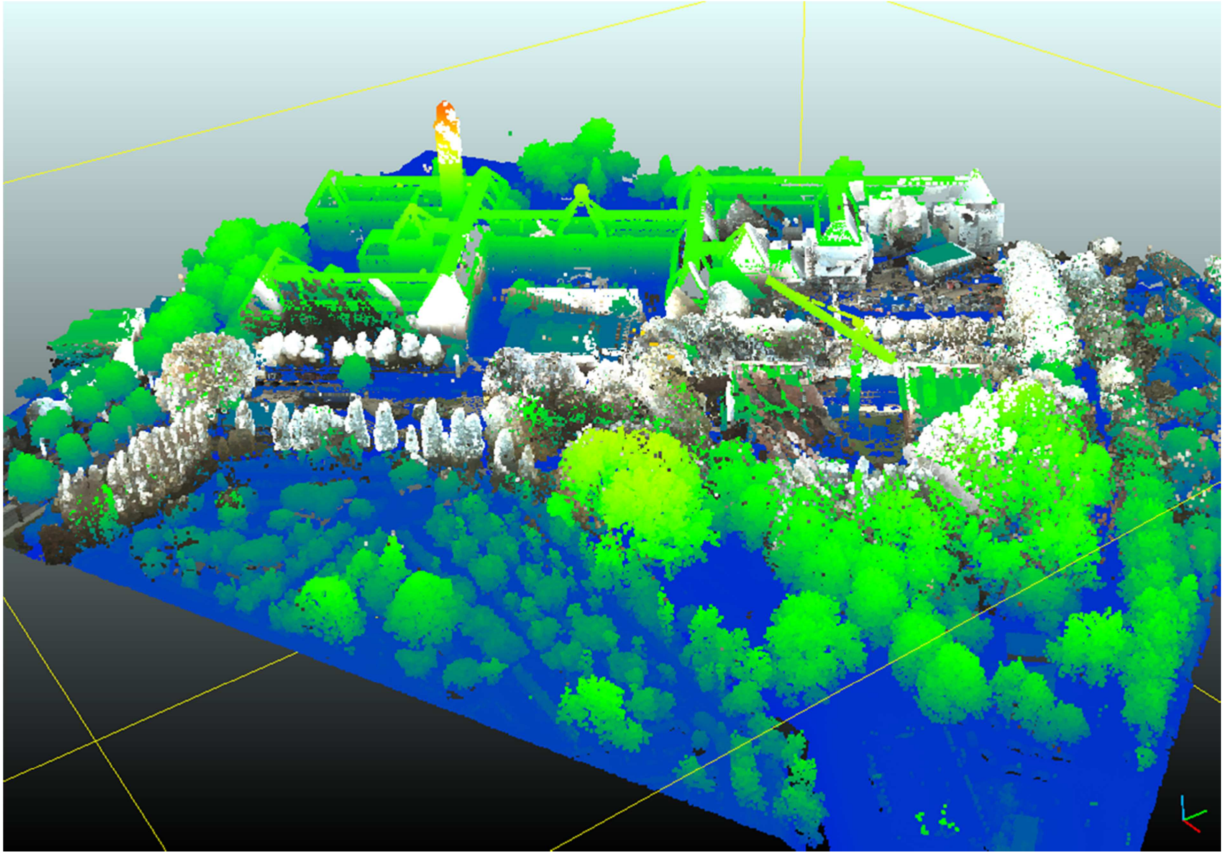
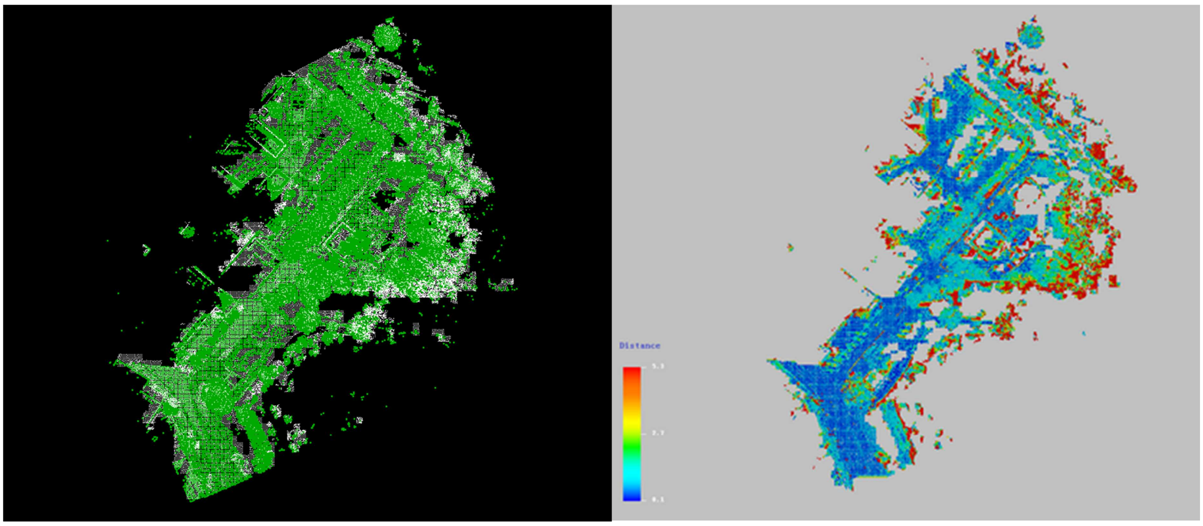


FIGURE 17: LEFT: AHN2 AIRBORNE (WHITE) AND LMMS DATA (GREEN) AFTER INTERSECTION. RIGHT: POINT CLOUD DISTANCES BETWEEN AIRBORNE AND LMMS ON THE INTERSECTED AREA.



11.2 GEOTIFF TO LAS, TU DELFT, #99

This service converts a geotiff file to a 3D point cloud in LAS format.

11.2.1 Motivation for inclusion

In Year 1 there was a Service #23 described as 'data conversion' which was not very specific. As a start, last year we have developed two exe files for .xyz to .pcd and for pcd to .xyz data conversion. During this year, considering the developments and needs of the other services, we have decided that it is important to have a basic service for Geotiff to LAS conversion as well. Currently LAS files are notably used in the urban showcase. Therefore this service is related to Service #23, but is actually completely new.

11.2.2 Current algorithm and possible improvements

This services works as follows. First a a geotiff image is read, which is consecutively converted into a point cloud. The result is saved in .las format.

11.2.3 Manual Text

The manual text look as follows:

NAME
    geotiff2las – converts a geotiff file into a las file

SYNOPSIS
    geotiff2lass [OPTION] ... [input-file] [output-file]

DESCRIPTION
converts argument [input-file] into a .las file where [input-file] is required to be a geotiff file.
When [output-file] is not specified argument geotiff2lass will write its output to input-file.las

-h   will show this man page

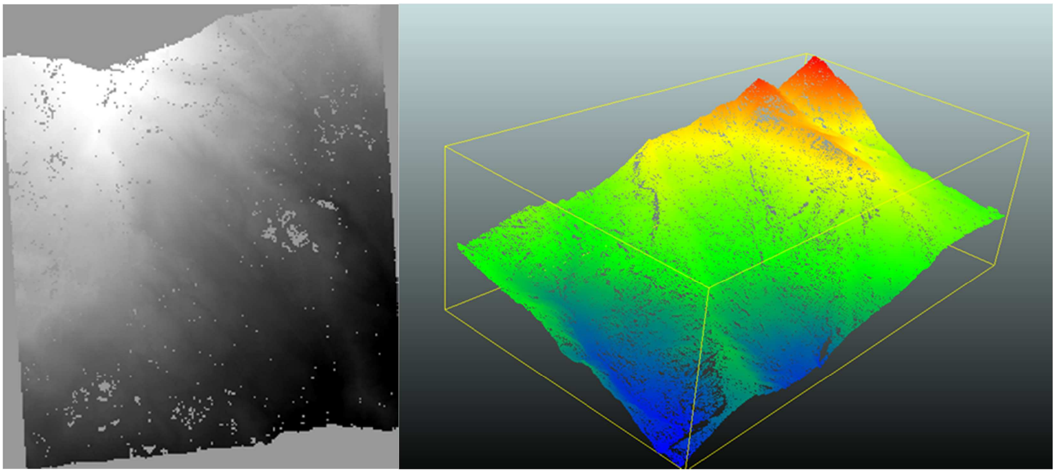
11.2.4 Big data strategy

If the input image has a height and/or width larger than 1000 pixels, we tile the input image into pieces of at mosts 1000 pixels long in both dimensions.

11.2.5 Example

In Figure 18 we show an example Geotiff image (left side) and a point cloud (right side) obtained by the algorithm.

FIGURE 18: LEFT, INPUT 2D GEOTIFF; RIGHT, OUTPUT 3D LAS FILE



11.3 REGISTERING 2D IMAGES ON POINT CLOUDS, TU DELFT, #100

Goal of this service is to cover mesh surfaces with a given texture. When one wants to increase details of a 3D structure and make it look more realistic, it is important to be able to add texture to a 3D surface. In order to solve this problem, with service #100 we have developed a service which generates a textured mesh when a 3D surface mesh and a picture to be used for texturing is provided.

11.3.1 Motivation for inclusion

This service belongs to the urban showcase. As a service initiated in year 1 it does not belong to a specific workflow in year 2.

11.3.2 Current algorithm and possible improvements

The algorithm needs at least 4 tie point couple in order to find the projection matrix, that projects the 2D image on the 3D point cloud. These tie points must be selected by the user and written to a text file as specified below in the manual. For the best texturing results, it is important to choose the tie points accurately and to choose keypoint locations such as window corners and points on the façade borders.

11.3.3 Manual Text

The manual text will look approximately as follows:

NAME

texturing – textures a point cloud

SYNOPSIS

texturing [OPTION] [input-file1] [input-file2] [input-file3] [input-file4] [output-file]

DESCRIPTION

Textures a point cloud or mesh, [input-file1] in pcd format, by draping [input-file2], and image in jpg format over it at a location specified by [input-file3] using camera parameters given in [input-file4]. [input-file3] and [input-file4] should be *.txt files.

If the input [input-file1] is a point cloud, then a mesh is generated first and saved as a file in .VTK format. If the input is already a mesh, then the texturing process runs straightforward. [input-file3] holds the corresponding tie points between mesh and image. This text file must contain these values as follows;

xp1 yp1 zp1 xi1 yi1
xp2 yp2 zp2 xi2 yi2
xp3 yp3 zp3 xi3 yi3
xp4 yp4 zp4 xi4 yi4
xp5 yp5 zp5 xi5 yi5

Here (xp, yp, zp) values are the 3D coordinates of the tie points in the mesh and (xi,yi) are the 2D coordinates of the tie points in the image which is going to be used for texturing. The algorithm requires at least 4 tie point couples.

If the cameraparameters are not known and not provided as input, the algorithm uses iPhone camera internal parameter values.

The textured result mesh is saved into the same work folder as a .VTK file.

-h

will show this man page

EXAMPLE

texturing mypointcloud.pcd image.jpg tiepoints.txt cameraparameters.txt

CONTACT
Beril Sirmacek, B.Sirmacek@tudelft.nl

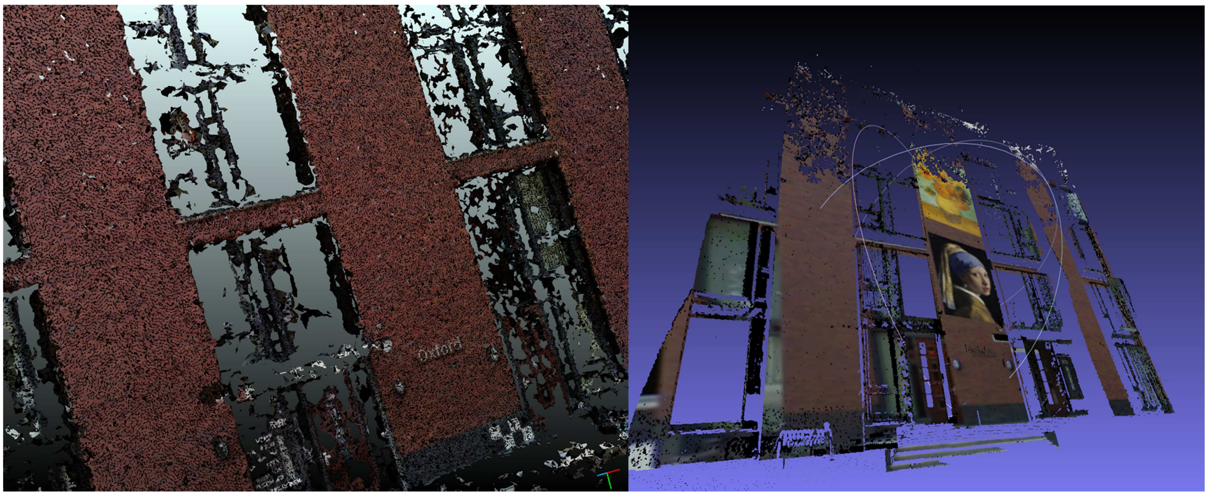
11.3.4 Big data strategy

This service is a typical example where the big data is rather characterized by its complexity then by it size. In the current version, tie points are given by the user. It is thinkable that in future a generic 2D to 3D matching strategy will be used to automatically drape 2D images on the corresponding part of a 3D point cloud. To do this is in a generic way is currently still challenging however.

11.3.5 Example

In Figure 19 we show two texturing results. The right hand side example shows that also unrealistic images can be used for texturing. It is also visble in this example, that texturing results can be in very high resolution.

FIGURE 19: 2D IMAGES PROJECTED ON A 3D FACADE OF A POINT CLOUD OF A TU DELFT CAMPUS BUILDING.



12 REFERENCES

[1] J. A. Richards. *Remote Sensing Digical Image Analysis*, 5th ed. Stringer-Verlag (2013).

[2] R. Giachetta. AEGIS - A state-of-the-art spatio-temporal framework for education and research. In *OSGeo Journal*, 13. OSGeo Foundation (2014).

[3] B. Madhukar, R. Narendra. Lanczos Resampling for the Digital Processing of Remotely Sensed Images. In *Proceedings of International Conference on VLSI, Communication, Advanced Devices, Signals & Systems and Networking* (VCASAN-2013); vol. 258 of *Lecture Notes in Electrical Engineering*. Springer India (2013).

[4] J. R. Herring (ed.). *OpenGIS Implementation Standard for Geographic Information: Simple Feature Access - Common Architecture*, version 1.2.1. Open Geospatial Consortium (2011). http://www.opengeospatial.org/standards/sfa

[5] J. L. Bentley, T. A. Ottmann. Algorithms for reporting and counting geometric intersections. In *IEEE Transactions on Computers*, 28 (9). IEEE (1979).

[6] K. Weiler, P. Atherton. *Hidden Surface Removal using Polygon Area Sorting*. In *Computer Graphics*, 11 (2). ACM (1977).

[7] Vosselman, G. and Maas, H.-G. (Eds). Airborne and Terrestrial Laser Scanning, Whittles, (2010).

13 APPENDIX: SERVICE TABLES

13.1 RADIOMETRIC ENHANCEMENT, FÖMI, #70

IQmulus Service information: #70			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	Radiometric enhancement		*Radiometric enhancement is used in many processing algorithms to enhance image processing and visualization.*
Description	Radiometric enhancements transform image histogram as a preprocessing operation for further analysis. The technique can differ, and there are also special forms of enhancement.		*User is able to choose from the specified enhancement methods.*
Service functionality	**Input:**	Raster image (GeoTIFF).	*Can work on any general TIFF file.*
	Input parameters: *[optional]*	Enhancement method, and additional parameters (for the specified method). Optionally the band number.	
	Output:	Enhanced raster image (GeoTIFF).	
	Functionality of the service:	Spectral transformation.	
Algorithm	The radiometric enhancement is applied to all spectral values of the image. For the execution, the generation and analysis of image histogram may also be required. The algorithm can be performed in-place and out-place. Each value modification can be performed independently (in parallel).		
Implementation details	**Implementation language**	C#	*The implementation is performed using the AEGIS [2] framework.*
	Dependencies with other libraries	Mono runtime environment	
	Operating	Any operation system	

	system	running .NET Framework or Mono.	
	Visualization modalities of the output	-	
IQmulus Data	• #5 (Orthophotos)		
Service characteristics	**Accuracy:** Methods are accurate to pixel level, as no heuristics are used in the operations. The result should match the result of other image processing software.		
	Robustness: The service handles all user errors (option errors) with appropriate error messages. Errors with respect to input data, operation execution and output writing are also handled. The operation can be executed on any image up to 32 bits radiometric resolution per band even containing floating point values. The allowed size of the input data is dependent on available memory (the input data is loaded into memory).		
	Computational time in relation to data size: The operations have linear algorithmic complexity (O(n)) with respect to image dimensions (width, height, number of bands, radiometric resolution).		
	Locality/globality of the algorithm: The operation is local for every spectral value (on each band of the image), but the histogram must be computed for each band of the image, which is a global operation. The execution time is usually within a minute.		
Alternatives	Use functionality from other toolkit (e.g. Orfeo Toolbox).		
Related use cases	**[Generic data manipulation service]** • Land Showcase User Story 2 (1.2.2_SC2_2) • User story 1.1_35		
Responsible Partner	FOMI, Roberto Giachetta		
Involved Partners	FOMI, Eötvös Loránd University (through contracted experts)		

13.2 CONVOLUTION FILTERING, FÖMI, #71

IQmulus Service information: #71			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	Convolution filtering		*Convolution filtering usually servers as a step in complex image operations (e.g. feature detection).*
Description	In convolution filtering an odd sized template (known as kernel) is applied to image regions, and the central spectral value is computed based on the neighbouring values. By using different kernels, several results can be obtained, for instance, smoothening, sharpening, edge detection, line detection, etc.		*User must be able to choose the method, the size of the filter (in some cases), and additional parameters of the kernel (e.g. the amount of sharpening).*
Service functionality	**Input:**	Raster image (GeoTIFF).	*Can work with other raster data source even without spatial reference.*
	Input parameters: *[optional]*	Convolution method, filter size (with some filter), and additional parameters of the kernel (with some filters). Optionally the band number.	
	Output:	Filtered image (GeoTIFF).	
	Functionality of the service:	Spectral transformation.	
Algorithm	The kernel is applied to all spectral values, requiring the operation to be out-place (neighbourhood values are also needed for computation). Each value modification can be performed independently (in parallel).		
Implementation details	**Implementation language**	C#	*The implementation is performed using the AEGIS [2] framework.*
	Dependencies with other libraries	Mono runtime environment.	
	Operating	Any operation system running .NET	

	system	Framework or Mono.	
	Visualization modalities of the output	-	
IQmulus Data	• #5 (Orthophotos) • #6 (Digital Surface Model) • #7 (Digital Elevation Model) • #8 (Digital Terrain Model) • #9 (Digital Terrain Model)		
Service characteristics	**Accuracy:** Methods are accurate to pixel level, as no heuristics are used in the operations. The result should match the result of other image processing software. Metrics: The usage of 0 radius filters should not modify the image.		
	Robustness: The service handles all user errors (option errors) with appropriate error messages. Errors with respect to input data, operation execution and output writing are also handled. The operation can be executed on any image up to 32 bits radiometric resolution per band even containing floating point values. The allowed size of the input data is dependent on available memory (the input data is loaded into memory).		
	Computational time in relation to data size: The operations have linear algorithmic complexity (O(n)) with respect to image dimensions (width, height, number of bands, radiometric resolution) and filter size (radius).		
	Locality/globality of the algorithm: The operation is focal for every spectral value, thus the pixel neighbourhood (matching the size of the kernel) is required. The filter can be applied individually for each band.		
Alternatives	Use functionality from other toolkit (e.g. Orfeo Toolbox).		
Related use cases	**[Generic data manipulation service]** • Land Showcase User Story 2 (1.2.2_SC2_2) • User story 1.1_35		

13.3 2D ON 2D IMAGE REGISTRATION, FÖMI, #72

IQmulus Service information: #72			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	2D on 2D image registration.		*Image registration is required for non-referenced rasters.*
Description	Image registration is the process of transforming different sets of data into one reference system (by using control points or reference imagery). Registration is followed by resampling (scaling) with multiple supported interpolations.		*User may input the control points, or use a set of referenced images to perform the registration.*
Service functionality	**Input:**	Raster image (GeoTIFF) (can be general TIFF image without reference system).	*Can work with other raster data source.*
	Input parameters:	Target reference system, ground control points (with reference system data), or referenced images (in GeoTIFF format).	
	Output:	Raster image (GeoTIFF) with spatial reference.	
	Functionality of the service:	Geometric transformation.	
Algorithm	If the control points are not specified, the referenced images are looked up for possible control points. There are then located on the source image resulting the control points. A polynomial transformation, and resampling is applied to the image (based on the number of control points).		
Implementation details	**Implementation language**	C#	*The implementation is performed using the AEGIS [2] framework.*
	Dependencies with other libraries	Mono runtime environment	

	Operating system	Any operation system running .NET Framework or Mono.	
	Visualization modalities of the output	-	
IQmulus Data	• #5 (Orthophotos)		
Service characteristics	**Accuracy:** Depending on the transformation and resampling method, the resulting image should contain values that within the range of the matched region of the source image. The result should match the result of other remotely sensed image processing software.		
	Robustness: The service handles all user errors (option errors) with appropriate error messages. Errors with respect to input data, operation execution and output writing are also handled. The operation can be executed on any image up to 32 bits radiometric resolution per band even containing floating point values. The allowed size of the input data is dependent on available memory (the input data is loaded into memory).		
	Computational time in relation to data size: The operations have linear algorithmic complexity (O(n)) with respect to image dimensions (width, height, number of bands, radiometric resolution) with minor additional cost of computing the transformation (independently of image size). However, the computation of Lanczos resampling is computation intensive.		
	Locality/globality of the algorithm: The lookup and allocation of control points is performed globally on the referenced images and the source image (but can performed in parallel). Resampling is performed locally on each pixel.		
Alternatives	Use functionality from other toolkit (e.g. Orfeo Toolbox).		
Related use cases	**[Generic data manipulation service]** • Land Showcase User Story 2 (1.2.2_SC2_2)		

	o User Story 1.1_35 o User Story 1.1_22 o User Story 1.1_30 o User Story 1.2.1_18 o User Story 1.2.1_13 o User Story 1.1_28	
Responsible Partner	FOMI, Roberto Giachetta	
Involved Partners	FOMI, Eötvös Loránd University (through contracted experts)	

13.4 TOP OF THE ATMOSPHERE REFELECTANCE, FÖMI, #73

IQmulus Service information: #73			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	Top of the atmosphere refelectance.		*Preprocessing is required to remove the effect of radiometric and geometric distortion.*
Description	Corrections of radiometric and geometric properties must be applied to all remotely sensed images before further processing and analysis can be performed. These corrections include the removal of atmospheric effects, instrumental errors and geometric effects (such as earth rotation, curvature and panoramic distortion).		*It is possible to have pre-defined correction parameters for sensors. Therefore the user only needs to input the sensor type and image capture parameters (e.g. time, look angle).*
Service functionality	**Input:**	Raster image (GeoTIFF).	*Can work with other raster data source.*
	Input parameters: *[optional]*	Sensor type and image capture parameters.	
	Output:	Preprocessed raster image (GeoTIFF).	
	Functionality of the service:	Geometric and spectral transformation.	
Algorithm	For atmospheric effects radiometric enhancement techniques are applied. For geometric distortion affine transformations, or control-point based polynomial transformations may be used. The image is resampled in the process.		*The used techniques heavily depend on the type of sensor used. Hence, a predefined sensor parameter collection must be specified.*
Implementation details	**Implementation language**	C#	*The implementation is performed using the AEGIS [2] framework.*
	Dependencies with other libraries	None.	
	Operating system	Any operation system running .NET Framework or Mono.	

	Visualization modalities of the output	-	
IQmulus Data	• #5 (Orthophotos)		
Service characteristics	**Accuracy:** Methods are accurate to pixel level, as no heuristics are used in the operations. Metrics: The result of application of multiple methods should be the same, as the individual application of methods in Services #70, #71 and #72.		
	Robustness: The service handles all user errors (option errors) with appropriate error messages. Errors with respect to input data, operation execution and output writing are also handled. The operation can be executed on any image up to 32 bits radiometric resolution per band even containing floating point values. The allowed size of the input data is dependent on available memory (the input data is loaded into memory).		
	Computational time in relation to data size: The operations have linear algorithmic complexity (O(n)) with respect to image dimensions (width, height, number of bands, radiometric resolution) and filter size (radius) with minor additional costs.		
	Locality/globality of the algorithm: Most corrections can be applied locally, but it depends on the methods used.		
Alternatives	Use functionality from other toolkit (e.g. Orfeo Toolbox).		
Related use cases	**[Generic data manipulation service]** • Land Showcase User Story 2 (1.2.2_SC2_2) • User story 1.1_35 • User story 1.1_22		
Responsible Partner	FOMI, Roberto Giachetta		
Involved Partners	FOMI, Eötvös Loránd University (through contracted experts)		

13.5 TOPOLOGICAL ANALYSIS OF 2D DATA, FÖMI, #74

IQmulus Service information: #74			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	Topological analysis of 2D data		*Topological analysis is a common feature in GIS toolkits.*
Description	The topological representation and analysis of 2D vector geometries is a common feature in most geospatial toolkits. Combined with the analysis of raster data, this forms a hybrid raster-vector topology model. On the model, several queries (equals, intersects, overlaps, …) and analysis operations (distance, intersection, buffer, …) can be performed.		*The analysis can be extended to 3D vector data.*
Service functionality	**Input:**	2D vector data (Shepfile), or feature coordinates. Raster data (GeoTIFF) in further versions.	*Can work with other raster and vector data sources. The operation (id) can be passed as a parameter with operation-specific values (e.g. the distance in case of buffering).*
	Input parameters: *[optional]*	The operation, and optional parameters.	
	Output:	Result of analysis (Shapefile or GeoTIFF), or topological relations.	
	Functionality of the service:	Topological analysis.	
Algorithm	The topological (graph-form) representation is computed from the inputs using multiple algorithms. Relations are calculated using the DE-9IM model.		
Implementation details	**Implementation language**	C#	*The implementation is performed using the AEGIS [2] framework.*
	Dependencies with other libraries	None.	
	Operating system	Any operation system running .NET Framework or Mono.	

	Visualization modalities of the output	-	
IQmulus Data	• #4 (Land Parcel Database) • #14 (Building Boundary Database) • #19 (Road Network) • #20 (Forestry Map) • #21 (Land Cover Map)		
Service characteristics	**Accuracy:** Methods are accurate to a certain level defined by the precision model of the geometries (at least 16 digits using double precision floating point values). The methods follow the requirements of the SFA standard.		
	Robustness: The service handles all user errors (option errors) with appropriate error messages. Errors with respect to input data, operation execution and output writing are also handled. The allowed size of the input data is dependent on available memory (the input data is loaded into memory, and additionally, the topology graph is also constructed within O(n^2) size with respect to number of points in the input).		
	Computational time in relation to data size: The execution time of the operations depends on the number of shapes and size of shapes from input (hence the total number of points (vertices) and connections (edges) in the source) and the number of intersections between the shapes. The operations are time consuming, as the topology graph of the source data has to built and analysed with O(n log n) complexity algorithms.		
	Locality/globality of the algorithm: Some executed operations are rather global, than local.		
Alternatives	Use functionality (mainly for vector data) from other toolkit (e.g. GeoTools).		
Related use cases	• Land Showcase User Story 2 (1.2.2_SC2_2) • User Story 1.1_22		
Responsible Partner	FOMI, Roberto Giachetta		
Involved Partners	FOMI, Eötvös Loránd University (through contracted experts)		

13.6 EXTRACTION OF RAIN DATA, IMATI, #96

<table>
<tr><th colspan="4">IQmulus Service information</th></tr>
<tr><td>Name of the metadata</td><td colspan="2">Content expected</td><td>Motivation/comments</td></tr>
<tr><td>Service Acronym</td><td colspan="2">Extraction of rain data</td><td>Unique identifier of the service; necessary to call it within user-defined workflows</td></tr>
<tr><td>Description</td><td colspan="2">This service reads the input data provided by the official Meteorological service and prepares the dataset for later elaborations. The service transposes the input table, computes the cumulative rainfall over the selected time interval, and groups the dataset by weather station.</td><td>Brief textual description of the service: what it provides, what can be used for. This text could be used as a short “help text” in the User Interfaces of IQmulus</td></tr>
<tr><td rowspan="4">Service functionality</td><td>Input: <data representation and format></td><td>Enriched point set (MAT); Each row is represented as follow:

ID of weather station (ANAG.CODE); time stamp (DTRF); rainfall (RAINC); name of weather station (NAME); longitude (LON); latitude (LAT); elevation (ELEV);</td><td rowspan="4">Include here the input/output of the service, the name of the functionality (e.g., registration, fusion), and input parameters (if any). In the future, we may consider defining a taxonomy of the functionalities to harmonize the terminology used to fill this field.</td></tr>
<tr><td>Input parameters: [optional]</td><td>• Time interval over which the cumulative rainfall has to be calculated</td></tr>
<tr><td>Output: <data representation and format></td><td>Enriched point set (MAT); Each row is represented as follow:

ANAG.CODE LAT LON ELEV RAINC_j

With j between 0 and the number of time interval.</td></tr>
<tr><td>Functionality of the service: <text></td><td>Pre-process input data;</td></tr>
<tr><td>Algorithm</td><td colspan="2">The input data set is stored in a 2D dictionary where the primary key is the station code and the secondary key is the time stamp. The data set will be sorted to fulfill the desired output structure.</td><td>The same functionality may in principle be implemented by different algorithms.</td></tr>
<tr><td>Implementation details</td><td>Implementation language</td><td>Python</td><td>Include information related to the implementation of the service, such as language (e.g.,</td></tr>
</table>

	Dependencies with other libraries		*C, C++, etc); dependencies with other libraries (e.g., ANN library); constraints on the operating systems, and visualization modalities of the output.*
	Operating system	Linux, Windows	
	Visualization modalities of the output		
IQmulus Data	**Available IQmulus input data**: #27;#42		*If there are examples of data that could serve as an example, then include here their identifiers as in the data table in eRoom (useful to test the service]*
Service characteristics	**Accuracy:** Machine precision. It depends on the accuracy of the input dataset. The service perform only sorting and arithmetic operations (sum) over the dataset.		*These fields are necessary to document all the characteristics of the services that are important to assess the quality of the results. Also, these fields could be used to select a specific service among more services that implement the same functionality but with different characteristics. See the following box.*
	Robustness: Depends on the input data quality and smoothness		
	Computational time in relation to data size: $O(N^2)$		
	Locality/globality of the algorithm: Each weather station can be processed regardless of the other		
Alternatives	No alternatives		*List other services (if any) that have the same functionality, have the same input/output but use a different algorithm or have different features*
Related use cases	Land scenario		*Mention the user stories related to the usage of the service (see D4.1.1) and related use cases uploaded on Red Mine*
Responsible Partner	IMATI simone.pittaluga@vs30.it		*Partner ID and responsible person (include email)*
Involved Partners			

13.7 POINT CLOUD INTERSECTION, TU DELFT, #98

IQmulus Service information			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	Point Cloud Intersection		*Determine intersected area from the two imported point cloud datasets.*
Description	Two overlapping point cloud datasets are imported and then the points of the intersecting area are determined in each dataset.		*Users can directly know if the two imported point cloud data have intersection region and where the intersection is.*
Service functionality	**Input:**	ASCII format *.xyz and *.las point cloud data	*Quickly acquire the intersected area of two overlapped imported point cloud data.*
	Input parameters: *[optional]*	File path of the two point cloud data.	
	Output: <data representation and format>	Two *.las files storing the points of the intersected area in each input dataset	
	Functionality of the service:	Acquire the intersected area of the two imported point cloud datasets.	
Algorithm	This service is implemented based on voxels. Firstly, two voxels are generated with the imported point cloud datasets. Then the relative position of the vertices of each voxel cell is determined based on the geometry of a cell. Thirdly, all labelled vertices are categorized and a decision is made whether the cell is a overlapping voxel cell. Finally, the points in the voxel cell are then classified and exported.		*OpenGL and OSG, which are also open source and can also be used to display the computed results.*
Implementation details	**Implementation language**	C/C++	*ANN and FLANN can also be used for the KNN searching.*
	Dependencies with other libraries	Nanoflaan	
	Operating system	Linux	
	Visualization modalities of the output	Points from the intersected area could also be outputted and displayed in a new	

		window.	
IQmulus Data	**Available IQmulus input data**: For tests we have used the following data; (1) AHN-2 data. Also in *.xyz format. (2) TU Delft campus LMMS point cloud data		
Service characteristics	**Accuracy:** a) The median or mean of the point to point distances (Service #8) between the two output point clouds is small. b) The spatial extent (Service #11) of the two output point clouds is almost the same.		
	Robustness: Reliability is verified by testing several different point cloud data.		
	Computational time in relation to data size: About fifteen seconds for approximately 1 million points[3]		
	Locality/globality of the algorithm: The program can be used to quickly obtain if the point cloud data have intersection area, if have , obtain the intersected region.		
Alternatives	Perform the internal computation and then output the intersected area and display in other point cloud visualization software, such as openmesh , open flipper, and QuickTerrain. Within IQmulus: Service #12		
Related use cases	For point cloud mosaic and then		
Responsible Partner	TUDelft, Jinhu Wang (Jinhu.wang@tudelft.nl)		
Involved Partners	• TUDelft, Roderik Lindenbergh (algorithm development support) • TUDelft, Beril Sirmacek [b.sirmacek@tudelft.nl] testing.		

[3] Practical tests have been done, but the developer could not yet indicate the computational complexity of his method

13.8 GEOTIFF TO LAS, TU DELFT, #99

IQmulus Service information			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	GeoTiff to LAS conversion		
Description	*Reading the image which shows the elevation, converting it to the point cloud and saving the point cloud as a LAS format data.*		*User can apply quick file format conversion.*
Service functionality	**Input:**	Geotiff image	
	Input parameters: *[optional]*	Path of the folder where the input 2D image is located.	
	Output:	Screen display of the input image and the generated point cloud (optional). Point cloud in LAS format	
	Functionality of the service:	Quick 2D to 3D conversion.	
Algorithm	We use libgeotiff library for reading the geotiff image. For file conversion, we use our own function. If the display option is chosen the input image and the generated point cloud are displayed using VTK and opencv library. Liblas library helps to write the generated point cloud in LAS file format.		
Implementation details	**Implementation language**	C++	
	Dependencies with other libraries	Liblas, GDAL, OpenCV, PCL, VTK, Boost, libgeotiff, libtiff Libraries are needed.	
	Operating system	Windows and Linux	
	Visualization modalities of the output	Visualization and rendering of the point cloud.	
IQmulus Data	**Available IQmulus input data**: For tests we have used the following data; (3) UBO – TLS Topography of a small beach in Brittany (ID-10-11) (4) UBO – Bathymetry in Brittany (ID-12-13) (5) Linguria – Lidar data for the study area of Vernazza, Liguria, Italy (ID-22-23)		
Service characteristics	**Accuracy:** Visual inspection if features in 3D		

	point cloud match features in 2D image.	
	Robustness: reliable to work with large point clouds	
	Computational time in relation to data size: Less than two seconds even for point cloud data with ~600000 points.	
	Locality/globality of the algorithm: Currently the program can convert xyz, ascii and las data formats to pcd format. We plan to implement more file format conversion options.	
Alternatives	It is also possible to display data and apply file format conversion using commercial software like QuickTerrain, MeshLab, CloudCompare, etc. Within IQmulus, compare Service #23	
Related use cases	Each user story in the Redmine list can benefit from file format conversion.	
Responsible Partner	TUDelft, Beril Sirmacek [b.sirmacek@tudelft.nl]	
Involved Partners	• TUDelft, Roderik Lindenbergh (algorithm development support) • TUDelft, Jinhu Wang (testing the developed algorithms)	

13.9 REGISTERING 2D IMAGES ON POINT CLOUDS, TU DELFT, #100

IQmulus Service information			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	Registering 2D images on point clouds		*Mapping texture on point cloud*
Description	3D surface mesh is generated from input point cloud if the mesh is not provided. User submits a list of tie point couples selected from the point cloud and the 2D image which is going to be used for texturing. Using the tie points and the estimated camera parameters, 2D texture is projected on the 3D mesh surface. Textured mesh is saved as a result.		
Service functionality	**Input:**	.pcd point cloud file or .ply format mesh .jpg 2D input image	
	Input parameters: *[optional]*	Path of the folder where the input point cloud and the 2D image are located. A list of tie point coordinates	
	Output: <data representation and format>	.ply format textured mesh file.	
	Functionality of the service: <text>	Adding textures on 3D surface meshes	
Algorithm	3D ranslation and rotation matrix is computed by using the list of matching tie points.		
Implementation details	**Implementation language**	C++	
	Dependencies with other libraries	PCL, VTK, FAST, OpenGL, OpenCV, Boost Libraries are needed.	
	Operating system	Windows and Linux	
	Visualization modalities of the output	Visualization of the 3D textured mesh surface can be provided automatically.	
IQmulus Data	**Available IQmulus input data**: For tests we have used the following data; (6) LMMS TUDelft Campus (7) IPhone point cloud of TUDelft OTB Building		*eRoom id's are provided.*

	(8) IPhone point cloud of TUDelft CiTG Building (9) IPhone point cloud of TUDelft Architecture Building (10) IPhone point cloud of residential building in Delft	
Service characteristics	**Accuracy**: The tie points must be selected accurately and well distributed on the surface, otherwise the registration accuracy will be lower. Some metrics to evaluate the accuracy: Façade edge fitting accuracy. (Minimal and maximal values of the distances between the point cloud edges and the registered image edges. Window & door edge fitting accuracy. (Minimal and maximal values of the distances between windows and doors in the point cloud and the same features in the registered images	
	Robustness: Reliable to work with large point clouds. Reliable, even if the 2D image camera parameters are not known.	
	Computational time in relation to data size: == several minutes for the residential building example. With increasing point cloud size, the process time increases exponentially.	
	Locality/globality of the algorithm: The program can easily be integrated to work with the other services.	
Alternatives	Commercial software like (QuickTerrain) can help to register the texture on 3D models if matching tie points are provided. Within IQmulus: compare Service #29.	
Related use cases	User Story (IQmulus) #1037, Land Showcase User Story 2 (1.2.2_SC2_2); User can register up-to-date scene on to the 3D models.	
Responsible Partner	TUDelft, Beril Sirmacek [b.sirmacek@tudelft.nl]	
Involved Partners	• TUDelft, Roderik Lindenbergh (algorithm development support) • TUDelft, Jinhu Wang (testing the developed algorithms)	