MULTIVARIATE SURFACE GENERATION (MONTH 30)

Deliverable D4.4.2

Circulation:	PU: Public
Lead partner:	SINTEF
Contributing partners:	SINTEF, CNR-IMATI, IGN
Authors:	Vibeke Skytt, Simone Pittaluga, Davide Sobrero, Laurent Caraffa
Quality Controllers:	Roderik Lindenbergh
Version:	1.0
Date:	04.05.2015

DOCUMENT HISTORY

Version[1]	Issue Date	Stage	Content and Changes
0.1	23.03.2015	20%	First incomplete draft
0.2	16.04.2015	60%	Partial restructuring and added content
0.3	20.04.2015	75%	More content
0.4	22.04.2015|	85%	Preliminary version sent to quality control
0.5	26.04.2015	99%	More content and update after first quality control
1.0	04.05.2015	100%	Update after second quality control. Ready for submission

[1] Integers correspond to submitted versions

EXECUTIVE SUMMARY

This report describes the progress made in the second year of the development of the toolbox for Task 4.4 *Multivariate Surface Generation.*

Eight new services have been implemented. Three services which are under development are also presented. The selection of the services has mainly been based on the workflows in the IQmulus show cases. The actual context is described for each individual service. Also the method applied in the service is described along with an assessment covering quality, performance and scalability, and sensitivity with respect to user input. Several services from the previous deliverable have been modified, from which two have been subject to major updates. The motivation for the update is presented along with the actual content of the update. These services also assessed with respect to the same criteria as the current ones. Three partners have been involved in the service development: SINTEF, CNR-IMATI and IGN. A uniform, tabular description of new and significantly updated services can be found in Appendix C.

Three different approaches are selected for surface generation:

- Triangulated surfaces
- Locally refined spline surfaces, i.e. LR B-spline surfaces
- A volumetric representation with an associated confidence metric

Data sets used for testing the services and external libraries contributing to the services are described in the report.

The services, both from the previous and from the current deliverable, are classified according to whether or not they are exposed to big data and in what form: volume, rapidly varying input data and whether the service leads to a reduction in data size or an organization of the data that is in particular appropriate for volumes of data. Some services are prioritized for demonstration of big data capabilities and the relevant strategies for handling big data are presented. In the description of each service, it is reported on how far this development has come at the present time and the report finishes with an outlook into how the work will proceed.

TABLE OF CONTENTS

1 INTRODUCTION

Surface generation from point data is in essence the operation of enhancing the data with structure and pursuing a more convenient way of representation. In GIS context, the traditional way of introducing structure is to transfer point clouds into rasters or grids.

A raster is a uniform point set organized in cells where each cell is represented by one point. The point can either be selected from the original point cloud or computed from it. This operation is denoted interpolation. Several different methods can be applied. The format is discrete, no information on the behaviour between the grid points exists so this has to be interpreted during exploration.

IQmulus task 4.4 is going beyond the raster format. Three different surface representations and corresponding methods for construction are pursued. The formats are:

- Triangulations. In D4.4.1 Delaunay type triangulations were delivered, now extensions to multi level triangulations are pursued.
- LR B-spline surfaces. The first related services were delivered in D4.4.1. This delivery contains improvements to the previously delivered services, some new services and a path towards management of larger data sets is introduced.
- A volumetric representation with a confidence metric associated to each cell in the representation. It is obtained by a carving procedure. A triangular surface can be extracted by setting a threshold to the confidence label. The first version of services related to this work will be delivered in D4.4.2, but the work will continue towards the next release.

The three different approaches are connected to different showcases, triangulations to the Land showcase, spline surfaces to the Marine showcase and the volumetric representation to the Urban showcase. This does not exclude the use of a particular format in other showcases. The motivation is mainly to concentrate resources and be able to create complete work flows.

This report corresponds to the second delivery of T4.4. The selection of services in this deliverable relate to the showcases defined in WP1. A short description will be given in the remainder of this section. In Section 2 we will discuss service evaluation in view of service quality and scalability. In particular the ability to handle large data sets is in focus. Section 3 and 4 gives an update on libraries and data sets used in implementation and testing of services. Section 5 presents new services developed in this period and Section 6 services which have been subject to major updates. Section 7 presents services which are still under development. Finally, a conclusion and an outlook into the next project period are presented. The report has 3 appendices, two of them related to Section 7. The last contains service metadata information for new and significantly updated services.

1.1 RELATED SHOWCASES

The showcases are defined from the User Requirements presented in D1.1, D1.2.1 and D1.2.2. The showcases have been updated in D1.2.3 and are expected to evolve further during the project. The showcases provide the major guidelines for the service definition of WP4. There are three showcases which again are divided into a number of workflows.

Land showcase:

The text below describes a usage scenario of IQmulus from the perspective of expert users who support decision makers for risk management in critical hydro-meteo events.

> *As hydrologist or geo-morphologist supporting decision makers in civil protection, I want to analyse data measured during critical events to prepare better prediction and monitoring of floods and landslides. To this end, I want to study the evolution of measured precipitation data as well as slope deformation from optical images, compute parameters to produce high-quality input for hydrological and mechanical modelling and simulation, and compare the results to reference measurements obtained for flooding events and landslides.*

The goal of the scenario above is to demonstrate the usefulness of IQmulus to compute rapidly, even roughly, parameters needed to run flooding simulation or landslide models. To this end, it is necessary to elaborate quickly large and heterogeneous datasets, the solution being not only implementing a high-performance computing environment, but primarily devising an intelligent handling and storage of the data.

The showcase is split into 5 workflows where the first two are addressed by services in this deliverable:

- (LS1) Organization of the terrain data according to their membership to main alert area (terrain model preparation)
- (LS2) Analysis and modelling of the observed rain (precipitation analysis)

Marine showcase:

The Marine Showcase is formulated as follows:

> *I want to create a seamless land/underwater elevation model by the integration of land and underwater data sources (topographic and bathymetric LIDAR and SONAR point clouds, existing digital elevation models and surface models in different formats - data models - resolutions), and want to extract the shoreline based on it, to obtain a seamless data product that can also be used in further analysis and processing tasks.*

The showcase is demonstrating how the current process for fusing multiple surveys and surface generation can be improved both in terms of automation and processing spent. These improvements will enable surfaces to be generated for much larger areas and also in response to user control. *The user control requires a new approach to the current deconfliction method so surveys selected by users can be combined without the need for human interaction.* The *user – controlled* or *interactive deconfliction* planned to be implemented in IQmulus will combine the metadata of the different datasets with geometrical properties of the different datasets as well as their spatial relationships.

The current T4.4 deliverable is focused on the following work flows :

- (MS1) Marine DEM generation: rapid and flexible generation of a single seamless surface from multiple disparate point cloud source data;
- (MS2) Inspection of the quality of the representation;
- (MS4) Change detection by comparing DEMs from different dates;

Urban showcase:

The urban show case is formulated in D1.2.3 and is described by:

I want to update my existing 3D catalogue of urban topographic objects given a new data set. For that purpose I want to be able to

- *remove non-static objects (like cars, rubbish bins, bikes, people,...),*
- *characterize changes (addition, removal, deformation) in static objects (like trees, bus-stops, facade elements, chimneys,...),*
- *possibly include new or exclude existing categories (e.g. remove category of roof-top antennas; add category of charge points for electric cars).*

It is divided into two work flows:

- (US1): Detection of buildings for monitoring and cadastral updating (long workflow)
- (US2): Individual tree extraction from urban LMMS data (short workflow).

The show case is not directly focused on surface generation, but does still have components from T4.4.

2 SERVICE QUALITY AND SCALABILITY

The showcases presented in the previous section are expected to be innovative and to demonstrate progress beyond the state-of –the art in one or more of the following categories:

- Usability: The solution should meet user needs
- Quality: The results should be trustworthy
- Performance and scalability: Handling of big data

Individual service are not all expected to deliver equally well with respect to each of these categories. Some services need to address for instance large data sizes, other work on already processed data. However, the services need to fit together into a work flow which fulfils some of the criteria above. In this section we will discuss service quality, in particular performance and scalability.

2.1 SERVICE QUALITY

Performance indicators for multivariate surface generation are

- quality meaning the accuracy in the adaptation of the surfaces to the point clouds, smoothness and curvature of the reconstructed surfaces;
- scalability meaning notably processing time and the possibility to parallelize the computation through data partitioning. For some services scalability refers also to the ability to handle rapid variation in the data

Surface quality is an established concept. Still, different approaches for surface generation will address the quality aspect differently and T4.4 also includes services that do not in itself generate a surface, but are related to surface generation. In the presentation of the individual services, the quality concept will be elaborated for that particular service and an assessment will be presented, when possible by comparing with "ground truth".
User input may influence the quality of the service result and the scalability properties of the services. The sensibility of a service with respect to human intervention will also be discussed.

2.2 BIG DATA

Current data acquisition methods produce large amounts of data, but volume is not the only unit for big data. Also data heterogeneity where the data points are produced by different methods with varying quality characteristics can be seen as an aspect of big data as well as rapidly varying input data. For rainfall data, the number of data acquisition sites is small so the number of data points for each time step is low. On the other hand the number of time steps is high and the time interval between them may be short. Thus, the challenge is not necessarily to handle large amounts of points, but to handle each time step fast enough to be ready for the next time step. The service results may have to be elaborated at a large number of positions.

The different services are exposed to the big data problem in different ways. Table 1 gives a summary on how services relate to big data. Services being exposed to large data sizes need a strategy for it. This can be:

- Multi-core, single node parallelization
- Multi-node parallelization using Hadoop and MapReduce or Spark
- Tiling and stitching of point clouds letting the infrastructure handle the parallelization. The tiling strategy will not be identical for all services.

- Streaming to adapt to structure in the input data and reduce memory usage.

A typical approach for services of some complexity is to initially implement a single core version of the service for later extending it to handle larger data sets. Several of the services in this delivery are in progress of being able to handle big data, but are currently limited to smaller data sets. The status column indicate the present situation with regard to big data

Id	**Purpose**	**Large data volume**	**High velocity**	**Data reduction /organiza tion**	**Big data strategy**	**Status**
			Services delivered at D4.4.1			
9	Spline surface from point cloud	Yes	Not expected	Reductio n	Parallelization Tiling and stitching	Ongoing work
40	Rainfall using kriging	No	Yes	No	Parallelization Tiling and stitching	Updated for D4.4.2
49	Constrained triangulation	Possibly	Not expected	No	Operate on selected points*	Used from #107
51	Triangulation from raster	Possibly	Not expected	No	Parallelization	Not prioritized
55	Spline surface from raster	Possibly	Not expected	Possible reduction	Parallelization Tiling and stitching	Not prioritized
56	Parameterize triangulated points	No	No	No	Thinned point clouds	Not relevant
57	Update spline surface with point cloud	Possibly	No	Yes	Parallelization Operate on tiled surfaces	For D4.4.3
66	Point dimensionality	Yes	No	No	Tiling and stitching	
67	Rainfall using radial basis functions	No	Yes	No		Not prioritized
84	Parameterization by spline surface projection	No	No	No	Operate on tiled surfaces	Not prioritized
			D4.4.2 services			
58	Rainfall using splines	No	Yes	No		Not prioritized
85	Stitching of LR B-spline surfaces	No	No	No	Operate on reduced data size	Not prioritized
86	Classify points from surface distance	Possibly	No	No	Operate on tiled surfaces Parallelization	Need integration with

						infrastructure
88	Distance field with respect to surface	Possibly	No	No	Operate on tiled surfaces Parallelization	Need integration with infrastructure
90	Trim spline surface	Yes	No	No	Operate on tiled surfaces	Need integration with infrastructure
91	Spline surface to raster	No	No	No	Operate on tiled surfaces	
95	Watertight surface reconstruction	Yes	No		Sub modular optimization. MapReduce	For D4.4.3
110	3D Delaunay triangulation	Yes	No	No	Tiling and stitching MapReduce Streaming	For D4.4.3
Services under development						
48	Multi-resolution triangulation	Yes	Not expected	Organize to improve usability	Tiling and stitching	For D4.4.3
107	Extract triangle mesh for visualization					Coupled with #48
108	Multi-resolution to raster					Coupled with #48

TABLE 1: T4.4 SERVICES AND BIG DATA

* The service is single core. The size of the data set that can be handled depends on hardware resources, mainly RAM. For instance with 16 GB RAM, the service can handle about 15 million points. The service is used in a chain of services with #48/#107. Then it will be fed with an appropriate number in data points.

Many services are exposed to big data sizes. As presented in D4.13, the group of services is analyzed and a number of the services are selected for special emphasis with regard to big data. For T4.4 the selected services are:

#9 Spline approximation from point cloud – Aspect: Size and data reduction

#40 Approximation of rainfall data using kriging – Aspect: Velocity and heterogeneity

#48 Multi-resolution for land monitoring – Aspect: Size and data organization

#66 Point cloud dimensionality – Aspect: Size

#95 Watertight surface reconstruction – Aspect: Sizeand heterogeneity

#110 3D Delaunay triangulation – Aspect: Size and heterogeneity

Activities to meet the scalability requirements for these services are ongoing. Some results are presented in this deliverable and the effort continues towards D4.4.3.

3 LIBRARIES & PROGRAMMING LANGUAGES

This section provides an overview of generic libraries, software tools and programming languages that currently support the development of T4.4 services. Only libraries and tools that are new compared to D4.4.1 are listed here.

3.1 CGAL

CGAL is short for *The Computational Geometry Algorithms Library.* CGAL is a software project that provides access to geometric algorithms in the form of a C++ library. CGAL is used in several areas needing geometric computation including geographic information systems..The software is available under open source licenses (LGPL or GPL depending on the component) when used for open source software.

CGAL v4.5 released in September 2014 provide a parallel 3D Delauney triangulation implementation.

3.2 GEOTIFF

GeoTIFF represents an effort by over 160 different remote sensing, GIS, cartographic, and surveying related companies and organizations to establish a TIFF based interchange format for georeferenced raster imagery. Libgeotiff is an open source library normally hosted on top of libtiff for reading, and writing GeoTIFF information tags.

3.3 ARMADILLO

Armadillo is a high quality C++ linear algebra library aiming at a good balance between speed and ease of use.

3.4 PCL

The point cloud library (PCL) is an open-source library of algorithms for point cloud processing tasks and 3D geometry processing.

3.5 GOTOOLS

GoTools is an inhouse SINTEF library. It is used for the LR B-spline related services. The g2-format which is used for representation of LR B-spline surfaces is related to this library. Some services (#9, #86, #88 and #90) can also read point cloud information in the g2 format. This option is included for debugging purposes.

4 DATA SETS

This section describes data sets used in the description of services in this report.

4.1 TOULOUSE

The dataset is a land-based mobile mapping data set corresponding to a 10km acquisition in downtown Toulouse (~1 hour of acquisition) with IGN's STEREOPOLIS mobile mapping system. It contains 1 012 160 278 LiDAR points (83GB). It was acquired with a multi-return Riegl VQ-250 laser scanner and consists of the following attributes:

- Absolute and relative measured 3D positions
- Absolute and relative sensor 3D positions
- GPS time
- Multi-return echo number (i/N)
- Measured range and angles
- Measured quantities (amplitude, reflectance, deviation)

The data set is used in the context of the Urban Scenario and the services #110 and #95 utilize the the fact that the data set offers additional information to the pure point cloud.

4.2 FOUR, SUBMARINE SAND DUNES

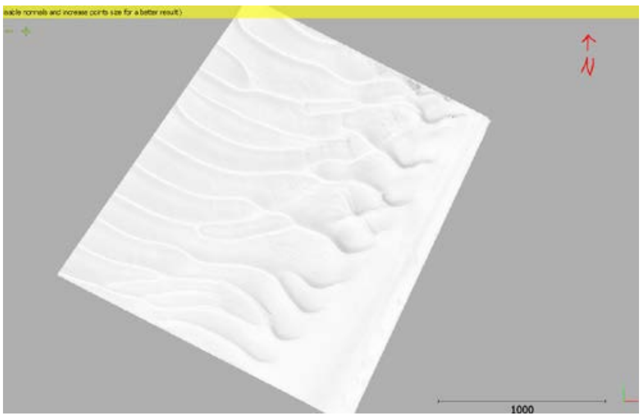


FIGURE 1: 2010 POINT CLOUD ACQUIRED FROM SAND DUNES AT FOUR

This data set contains two acquisitions of submarine sand dunes in Brittany obtained by Bathymetry. The data exists as point clouds and as Digital Terrain Models with a resolution of 2m. The data is acquired in 2010 and 2011. The gridded point clouds cover different sized areas. The 2010 raster is a 13 M ENSI ASCII file, while the 2011 raster is of size 93 M. The original point clouds are represented as .txt and the 2010 and 2011 point clouds have size 135 M (4.6 million points) and 119 M (4 million points), respectively.

The point cloud from 2010 is used in the description of service #88.

4.3 SUBMARINE SURVEY CLOSE TO BRITAIN

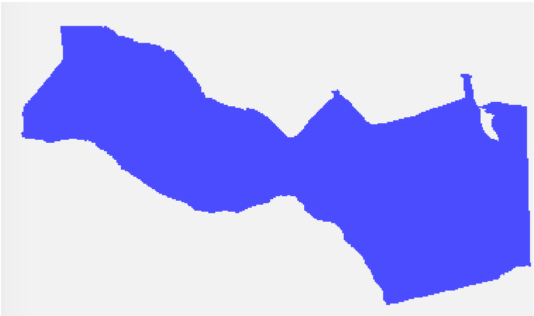


FIGURE 2: SURVEY OFF THE COAST OF BRITAIN PROVIDED BY HRW

FIGURE 3: THE LEFT HALF OF THIS POINT CLOUD

The data set is originally provided by HRW and contains one survey. The original data was given by longitudes, latitudes and elevation data. It has been translated into UTM coordinates to give better correspondence between figures in x and y and elevation. The translated data set is shown in Figure 2. The point cloud consists of 131 million points and in most cases only the left part of this point cloud is used for examples in this report, see Figure 3. Information regarding this sub cloud is given in the table below.

Data set	Size (las)	No of points	Elevation	Domain
HRW 143513 1	1.1 G	58.58 mill.	-27.22,2.63	[619003,631565]x[5622915,5631291]
HRW 143513 2	559 M	29.29 mill.	-27.22,2.63	[619003,631565]x[5622915,5631291]
HRW 143513 3	280 M	14.64 mill.	-27.22,2.63	[619003,631565]x[5622915,5631291]
HRW 143513 4	140 M	7.32 mill.	-27.22,2.63	[619003,631565]x[5622915,5631291]
HRW 143513 5	70 M	3.66 mill.	-27.22,2.63	[619003,631565]x[5622915,5631291]

TABLE 2: PROPERTIES OF HRW SUB CLOUD. THE FILE FORMAT IS LAS.

This data set is used in testing of the services: #90, #85 and #9.

4.4 LIGURIA PRECIPITATION DATA

The data set is used to test service #58. It was also used for service #40 and #67 of D4.4.1. Rainfall data visualized in Figure 3 is gathered from the Regione Liguria network (red stars) and from the Genova municipality network (purplecircles) for a total of about 178 measurement stations. The rainfall is cumulated every 30 minutes. In addition to the dataset delivered with D4.4.1 the updated service #40 could take as input rain gathered from weather radar (dataset #43). The radar data have to be translated from binary format to the same format as the data sets gathered by the rain gauges.

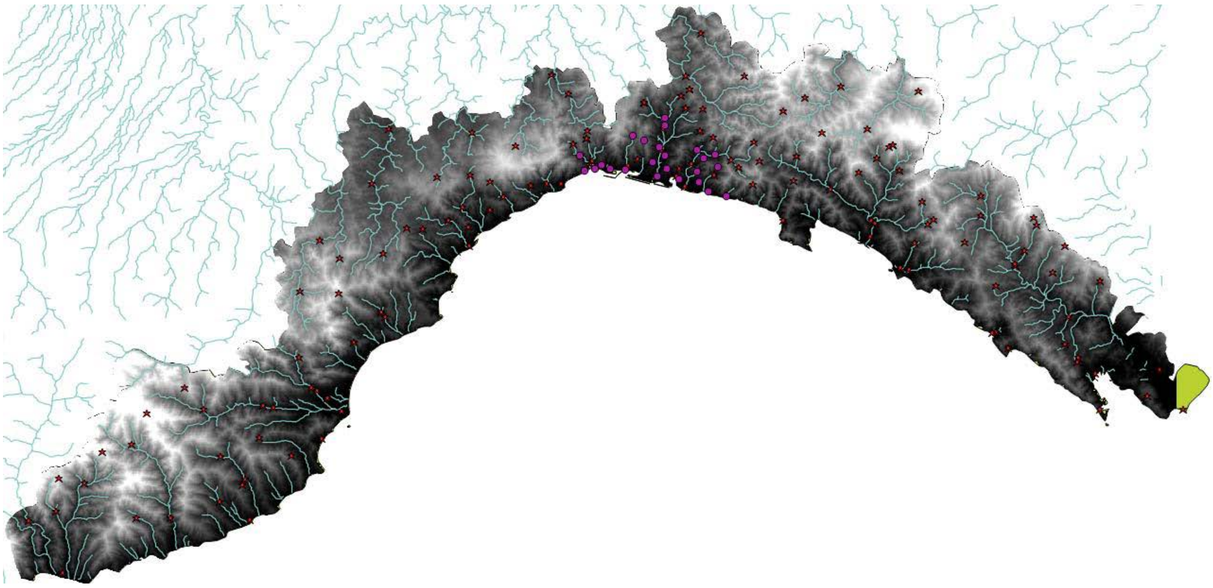


FIGURE 4: THE RAIN FIELD AT STEPS OF 30 MINUTES FOR POINTING OUT THE EVOLUTION OF A THUNDERSTORM OVER THE CITY OF GENOVA

4.5 REUNION

The Reunion data is provided by UBO and consists of a bi-temporal MBES point cloud data set acquired in 2009 and 2013. The two data sets overlap partially. The point clouds are shown in the Figures 5 and 6. The initial point clouds are thinned to get data of different sized. The figures are given in the table below.

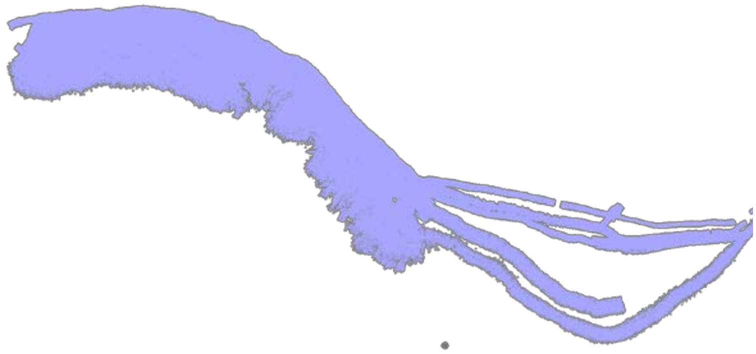


FIGURE 5: REUNION 2009

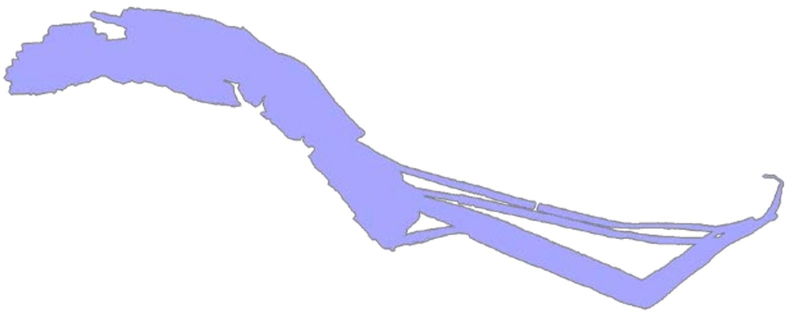


FIGURE 6: REUNION 2013

Data set	Size (las)	No of points	Elevation	Domain
REUNION 2009 1	559 M	29.29 mill.	-245.55,-3.96	[329345,342275]x[7636723,7645220]
REUNION 2009 2	280 M	14.64 mill.	-245.55,-3.96	[329345,342275]x[7636723,7645220]
REUNION 2009 3	140 M	7.32 mill.	-245.55,-3.96	[329345,342275]x[7636723,7645220]
REUNION 2009 4	70 M	3.66 mill.	-245.55,-3.96	[329345,342275]x[7636723,7645220]
REUNION 2013 1	605 M	31.69 mill.	-241.46,-0.99	[329296,342273]x[7637267,7645077]
REUNION 2013 2	303 M	15.84 mill.	-241.46,-0.99	[329296,342273]x[7637267,7645077]
REUNION 2013 3	152 M	7.92 mill.	-241.46,-0.99	[329296,342273]x[7637267,7645077]
REUNION 2013 4	76 M	3.96 mill.	-241.46,-0.99	[329296,342273]x[7637267,7645077]

TABLE 3: PROPERTIES OF REUNION POINT CLOUDS

This data set is used in testing of the services: #86, #90, #91 and #9.

4.6 LIGURIA-LAS

The Liguria-LAS dataset is a collection of raw point clouds acquired by airborne flights organized

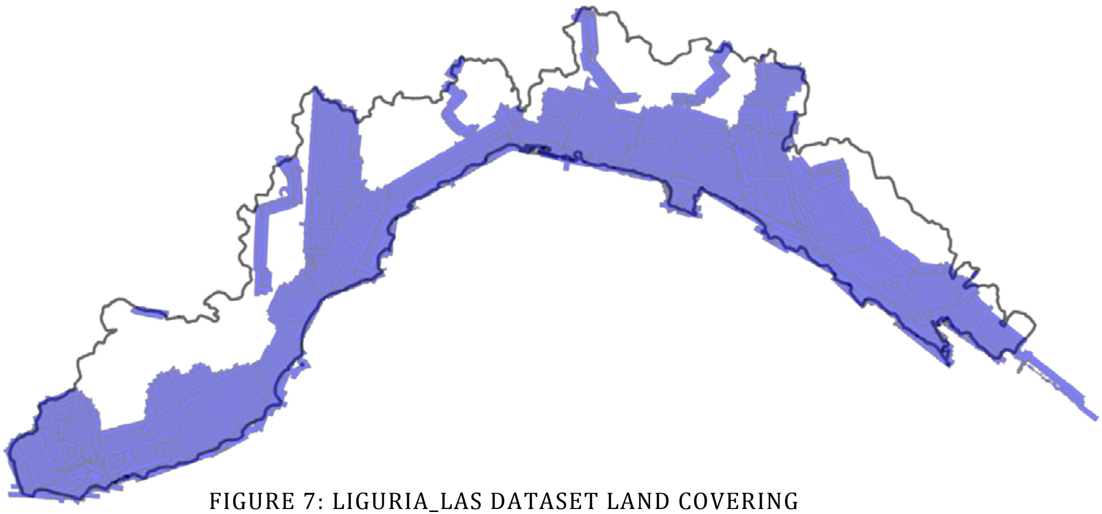


FIGURE 7: LIGURIA_LAS DATASET LAND COVERING

by the italian Ministry of Environment.

It consists of 927 LAS file for a total size of 599 GB. It covers around 65% of the land of Regione Liguria. Since each LAS file is the result of one strip acquired during a single flight, there are overlapping between different strips. The reference system is GCS-WGS84.

The Liguria-LAS dataset is the dataset used for services within the MultiResolution for Hydrology framework, as described in Section 7.

5 NEW SERVICES

This section describes in detail the services developed in Task 4.4 for D4.4.2 and their experimental evaluation on the above mentioned sample datasets.

Service #	Langu age	Os	Serial / parallel / distributed	Metadata available	On artifactory	Smoke test passed	Logging enabled	Scalability test performed
#58	C++	Linux	Serial	Yes	Yes	OK	Yes	Locally
#88	C++	Linux	Parallel	Yes	Yes	OK	Yes	Locally
#86	C++	Linux	Parallel	Yes	Yes	OK	Yes	Locally
#90	C++	Linux	Serial	Yes	Yes	OK	Yes	Locally
#91	C++	Linux	Serial	Yes	Yes	OK	Yes	Locally
#85	C++	Linux	Serial	Yes*	Yes	OK	Partially	No
#110	C++	Linux	Serial	No	No	No	No	No
#95	C++	Linux	Serial	No	No	No	No	No

TABLE 4: STATUS OF D4.4.2 SERVICES WITH RESPECT TO RELEASE CRITERIA

Table 4 gives an overview of the newly developed T4.4 services.
* The service including metadata information will be updated.

5.1 #58 RAINFALL BY LR B-SPLINE APPROXIMATION, SINTEF

The service performs an approximation of rainfall data measured at a number of rain gauges sites. The data is measured in a number of time steps and for each time step the measurement values are sparse and irregularly situated. The service approximates the rainfall values for each time step separately and produces a number of LR B-spline surfaces. For each time step, the number of data points is low, thus high volume is not an issue for this service.

The service is related to the Land Scenario (LS2). The input data are the rainfall data gathered from a network of rain gauges and/or sampled from radar measures maps. The output are the estimated rainfall fields represented as one LR B-spline surface for each time step. Approximately the same functionality is covered by the services #40, approximation of rainfall data with ordinary kriging, and #67, approximation of rainfall data with radial basis functions. The three services participated in the IQmulus context on rainfall data in June 2014.

5.1.1 Method and implementation

The core of the algorithm is the approximation functionality which also is the core of the algorithm for service #9 presented in Section 6.1, but the governing parameters given to the adaptive approximation algorithm differ. For the purpose of this service only approximation with the so-called LR-MBA method is applied. This alternative is more appropriate than least squares approximation for very scattered and highly varying data points. The core approximation algorithm is shortly presented in Section 6.1.2.1. Special care is taken to avoid negative rain field results.

5.1.2 Example

The test data set is the same as the one used for service #40 and #67 in D4.4.1 and has been described in Section 4.4. Rainfall data gathered from the Region Liguria and from the Genova

municipality networks gives a total of 178 measurement stations. Data for every 30 minutes are considered. The selected rain fall is measured at a day with a thunderstorm. Thus, the data contains large differences in the amount of rain within small geographical areas. The approximation results will be discussed in Sections5.1.3 to 5.1.5..

5.1.3 Quality

This service has been evaluated according to the following quality criteria:

- Accuracy of the approximation scheme. The value of the LR B-spline surfaces are compared to the input rainfall values
- Cross-validation. Every 30 minutes, the rain gauges have been turned off, one by one, and the surface generation has been performed with the reduced input data. The resulting surface is then compared to the unused rain gauge.
- Comparison with a different scale. The surface generation is performed based only on the data from Region Liguria and then compared to the data from Genova municipality which is denser than the regional data, but contained in a smaller area.

The difference between estimated and measured rain values is summarized by the quantities maximum and average distance for all time steps and all measurement values and maximum and average Mean Squared Error (MSE) given as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i^* - p_i)^2$$

MSE is computed for each time step. Here p denotes the estimated value and p^* the measured value. n is the number of measurement values for each time step.

Criteria	**Maximum error**	**Average error**	**Maximum MSE**	**Average MSE**
Accuracy	0.056 mm (0.093%)	0.0046 mm	0.00033	0.00018
Cross validation	27.21 mm (45.4%)	0.043 mm	31.79	7.51
Different scale	30.4 mm (50.7%)	0.6 mm	206.61	20.31

TABLE5: ACCURACY OF LR B-SPLINE SURFACES APPROXIMATING RAIN FIELDS

The percentage values show the maximum error compared with the maximum measured rain fall value which is 60mm. The approximation algorithm relates to a prescribed tolerance of 0.05 mm and refines the spline space corresponding to the rain field surfaces where the required accuracy is not met. A maximum of 20 iterations of refinement and re-approximation is performed. The cross-validation accuracy and the different scale comparison give good results where the rain fall is smoothly varying, but show larger discrepancies when the thunderstorm hits. In the comparison with the other two services, service #40 gave the overall best results, but at that time it also was the most time consuming. Service #40 has been chosen for further development and the updated version is presented in Section 6.2.

5.1.4 Scalability

The service treats each time step separate and is expected to perform linearly with regard to the number of time steps. However, highly varying rainfall values requires more computation time than time steps with less variation. The relative number of time steps with large variation in the rainfall values will, thus, influence the shape of the graph in figure 7. Also the number of iterations allowed in the generation of the rain field approximation influenced the computation time as can be seen in the figure.Each time step consists of a small number of rain fall measurements so the data size at one point in time is not an issue with regard to scalability.

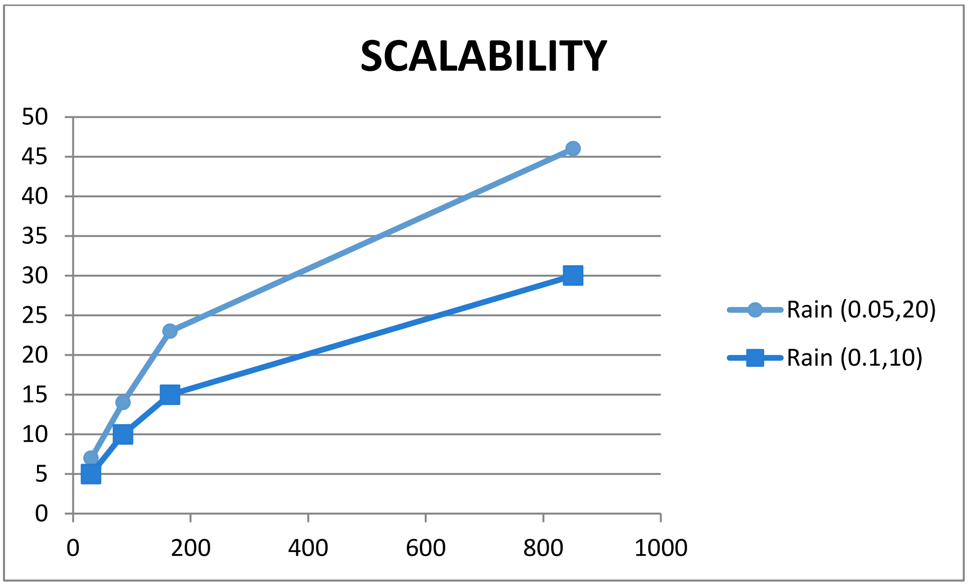


FIGURE 7: SCALABILITY OF SERVICE WITH RESPECT TO DATA SIZE. THE GRAPHS ARE MARKED WITH THE PARAMETERS APPLIED IN THE TEST: TOLERANCE AND NUMBER OF ITERATIONS

Figure 7 illustrates the scalability of the service. The data size along the x-axis is given in KB and the time along the y-axis in seconds. Two different sets of input parameters are applied. Accuracy figures of one time step with respect to these parameters can be found in the next paragraph. For the scalability testing only data from the Region Liguria are used while in investigating the degree of human intervention, also Genova municipality data is included. This should not change the outcome of the investigation. The computations are performed on a Laptop with 15.6 GB memory and 8 Intel@Core i7-4702HQ CPU cores.

5.1.5 Degree of human intervention

The service requires user input specifying a tolerance and a maximum number of iterations in the adaptive approximation algorithm. Selecting a too small tolerance may lead to a too concentrated peak at the rainfall values, leaving the overall smoothness of the resulting rainfall function to suffer. Too many iterations may result in the same effect. On the other hand, large tolerances and/or a small number of iterations will give poor accuracy at the rainfall measurements. In the following, we will illustrate the effect of the user input through an example.

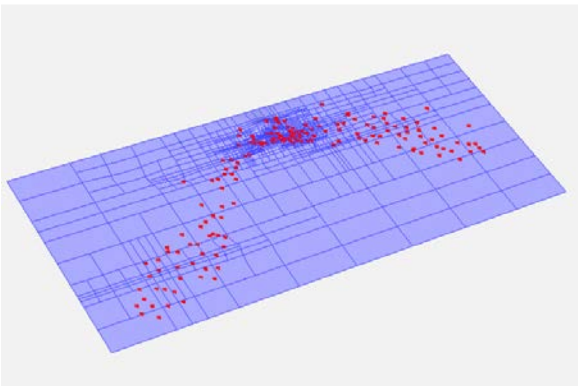


FIGURE 8: OVERVIEW OVER MEASURED RAIN VALUES AND APPROXIMATING RAIN FIELD

Figure 8 shows the rainfall measurements (red dots) and approximating rain field function for time step number 41 in the data set. The number of rain fall values is 168. The rain values are parameterized on a scaled version of the xy-domain of the corresponding terrain. The scaling enables better correspondence between distances in x and y and rain values, and also the possibility of visualizing rain data and field as 3D entities.

Figure 9 shows a detail of the rain field for different input parameters. The blue lines show boundaries between polynomial elements in the rain field function. The shapes of the rain fields are very similar, but the second field has got more polynomial elements. The actual measurements of the accuracy show a slightly larger effect

Tolerance, no of iterations	No of elements	Max. distance	Average distance	No of points > tolerance
0.1 mm, 10	1411	0.552	0.0159	2
0.1 mm, 20	1555	0.095	0.0096	0
0.05 mm, 10	1741	0.552	0.01	2
0.05 mm, 20	1904	0.046	0.0043	0

TABLE 6: ACCURACY OF RAIN FIELD APPROXIMATION WITH REGARD TO INPUT PARAMETERS

The most critical parameter with regard to distance is the number of iterations. It governs the limit of available polynomial elements and the algorithm will always approximate the points as close as possible given the current spline space. The tolerance governs the refinement strategy

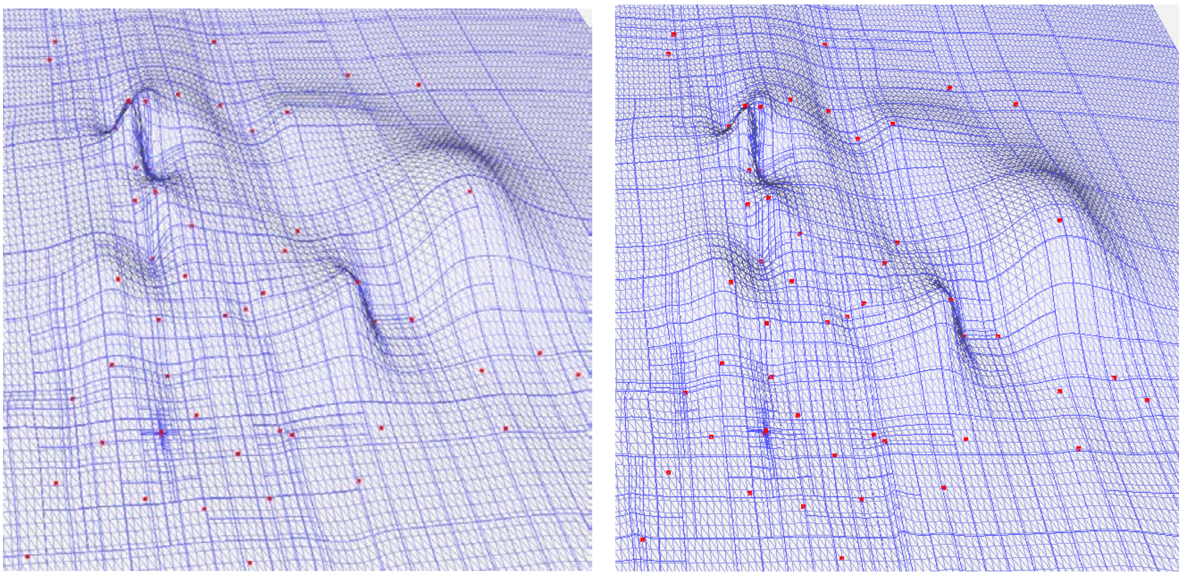


FIGURE 9: DETAIL OF RAINFIELD USING TOLERANCE 0.1 MM AND 10 ITERATIONS (FIRST) AND TOLERANCE 0.05 MM AND 20 ITERATIONS (SECOND)

5.2 #88 DISTANCE BETWEEN POINT CLOUD AND LR B-SPLINE SURFACE, SINTEF

This service measures the distance between a point cloud and an LR B-spline surface representing the same area. The current version is single-threaded, but it is currently ported to a multi-threaded version. The point cloud size is limited by memory usage, but in practice the surface domain is limiting the point cloud size to a larger extent. The surface domain is defined from point clouds given as input to service #9 and these point clouds can be handled by this service.

The surface is representing elevation and is parameterized on the x- and y-values of the point cloud from which it was generated. For each point the distance to the surface is computed and the result is either exported in text format as x, y, z and distance or as a ply file. The point cloud

can be the one originally used to create the surface or another point cloud sampling the same area obtained at a different time or by a different sensor.

The service is a part of the Marine Scenario where it is relevant in several situations:

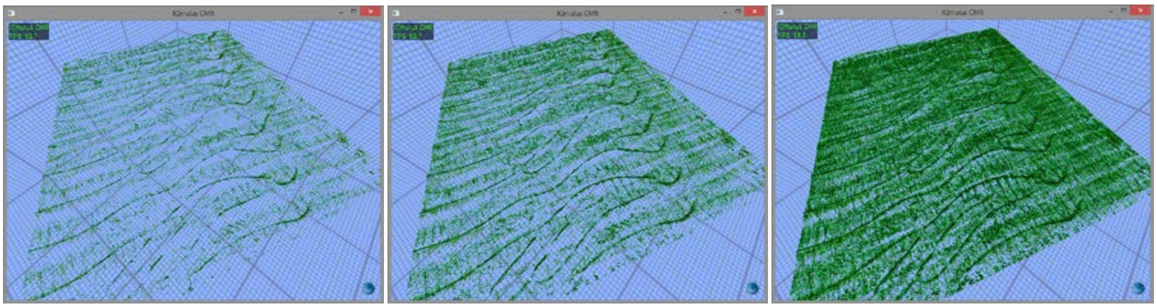
- Quality control. The surface is compared to the point set from which it was created. Knowing the distances associated to each single point makes it possible to identify outliers, points lying in steep areas where the distance in the z-coordinate gives a false impression of the real distance and points where the surface does not give an adequate representation of the terrain. In such areas additional representation formats like triangulations could be used to improve the model.
- Change detection. If a surface created from one point cloud is compared to another point cloud obtained from the same area at a different time, domains where the terrain have changed will stand out by having large distances between the point cloud and the surface. This offers a tool to decide where effort to compute the changes more accurately should be applied.
- Deconfliction. Different data sets from different dates and with varying quality cover the same area. They overlap, but have varying extents. This service can be used to detect areas and data sets where these give inconsistent information.

5.2.1 Method and implementation

The distance field is obtained by evaluation. To improve the performance in manoeuvring in the data structure of the LR B-spline, an intermediate data structure with direct access to the polynomial elements in the surface is created. The points are sorted according to which element they belong to and evaluation is performed by evaluating all B-splines in the parameter value corresponding to the point and multiplying the B-splines with the corresponding coefficients and scaling functions, see the surface expression in Section 6.1.2.1. Given polynomial degrees of d_1 and d_2 in the x and y direction, respectively, $(d_1 + 1) * (d_2 + 1)$ basis functions are computed recursively. The polynomial elements have access to all B-splines overlapping the element so no further transversal of the data structure is required.

5.2.2 Example

The 2010 point cloud from the sand dunes at Four has been approximated by an LR B-spline surface. This service has been used to compute the distance field between the point cloud and the surface. Figure 10 shows the distance field applying different distance thresholds for whether or not a point should be shown. The tips of the sand dunes clearly stand out as areas where the approximation is less accurate than the general result.



FIGUR 10: DISTANCE FIELD WHERE POINTS ARE SELECTED FOR VISUALIZATION DEPENDING ON THE DISTANCE TO THE ASSOCIATED SURFACE

5.2.3 Quality

Spline evaluation is a numerically stable operation that in essence consists in computing convex combinations of the value of B-splines of lower degrees in the parameter values of the points. The service eactly computes the difference in elevation between the point cloud and the surface for each point. However, the distance is measuring the length of the height axis between the point and the surface and this is in some cases an overestimate compared to 3D distance, see Section 5.3.3 and 5.3.4 for more information.

5.2.4 Scalability

Due to the local support of the B-splines only a limited number of B-splines are non-zero at each data point so the evaluation performed in this service is not a resource demanding operation. The manageable size of the point cloud is limited by the memory size of the computer. However, in practice the data size is limited by the domain represented by the LR B-spline surface involved in the computations. No further tiling is required for this service. The surface is typically created using service #9. Thus, the point cloud size is limited by the properties of that service.

Operations on large LR B-spline surfaces are in general time consuming due to a complex data structure. This effect is avoided by the pre process creating a direct access to the polynomial elements.

The service is multi-threaded and it scales linearly with respect to the size of the point cloud. The scalability is equal to that of service #86 and the graph from Section 5.3.4 is applicable also for this service.

5.2.5 Degree of human intervention

The service is parameter free and does not require any human intervention.

5.3 #86 CLASSIFICATION OF POINTS WITH RESPECT TO LR B-SPLINE SURFACE, SINTEF

The service is related to work flow 2 of the Marine Scenario. The intended use is the same as for service #88, but the presentation of the result is different. In this case the points are grouped according to the distance to a given surface and to application specified thresholds. The result of

this service can be used in a visual inspection of a surface created from a point cloud. The service makes it for instance is possible to check if outlying points belong to steep areas where the distance computation along the height axis does not reflect the real distance or if the accuracy of the surface approximation is poor in an area. The service serves as a quality test for LR B-spline surfaces created for instance by service #9.

This service will be applied to point clouds suitable for being processed by service #9 or tiled point clouds being pre processed according to that service. No further tiling of the point cloud is required.

5.3.1 Method and implementation

The first part of this service is the same as for service #88. Next the points are classified according to categories as defined by the input parameters of the service.

5.3.2 Example

FIGURE 11: LR B-SPLINE REPRESENTATION OF THE 2013 REUNION DATA SET

In this example, we will investigate the quality of an LR B-spline representation of the REUNION data set from 2013 (Section 4.5). The approximation tolerance was 0.5 m. The surface is shown in Figure 11.

Accuracy information from the approximation service (#9) revealed that for some points the distance to the surface was quite large. To get an interpretation of this information we apply this service. The points are classified and visualized in Figure 12 according to the following categories:

- > 20 m below the surface: Bright green and large dots
- 5-20 m below the surface: Lighter green and medium sized dots
- 1-5 m below the surface: Light green and small dots
- 0.5-1 m below the surface: Very light green and small dots
- 0-0.5 m below the surface: Small white dots
- 0-0.5 m above the surface: Small white dots
- 0.5-1 m above the surface: Very light red and small dots
- 1-5 m above the surface: Light red and small dots
- 5-20 m above the surface: Lighter red and medium sized dots
- > 20 m above the surface: Bright red and large dots

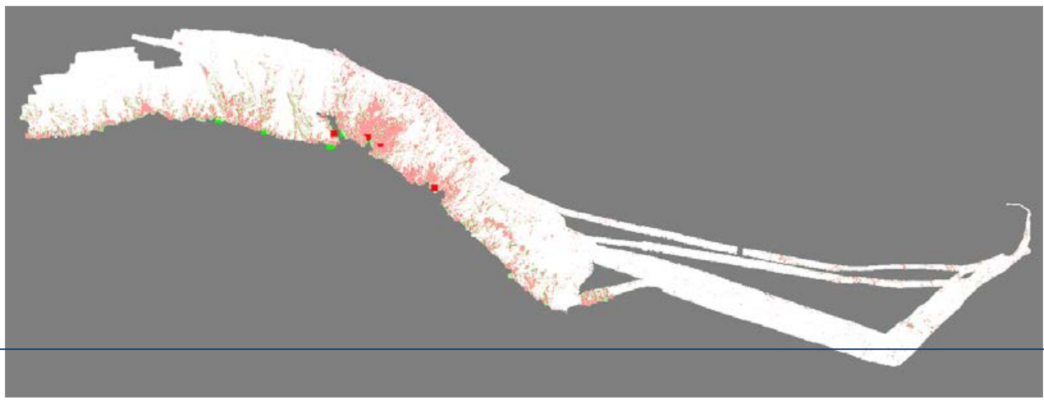


FIGURE 12: CLASSIFIED POINT SET

We can see that in some areas, the distances are large. In Figure 13 we zoom in on the most critical area

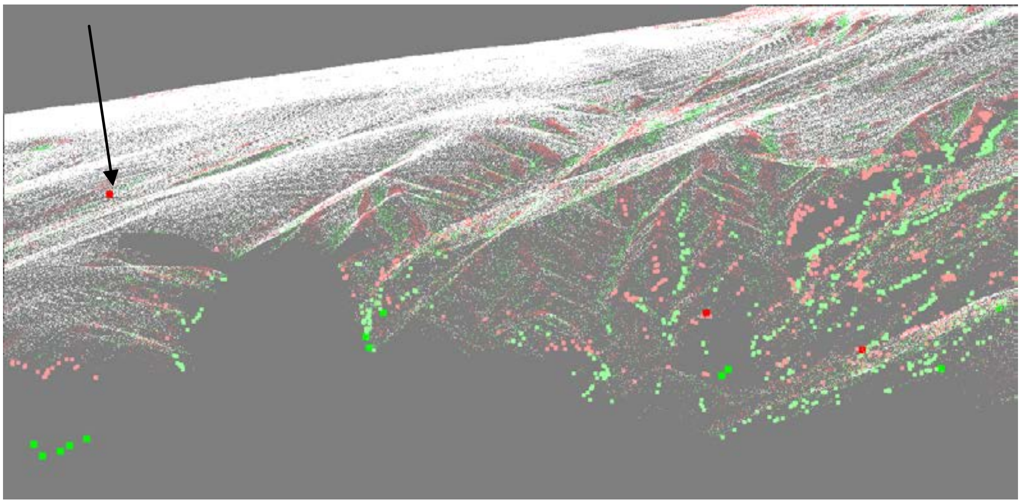


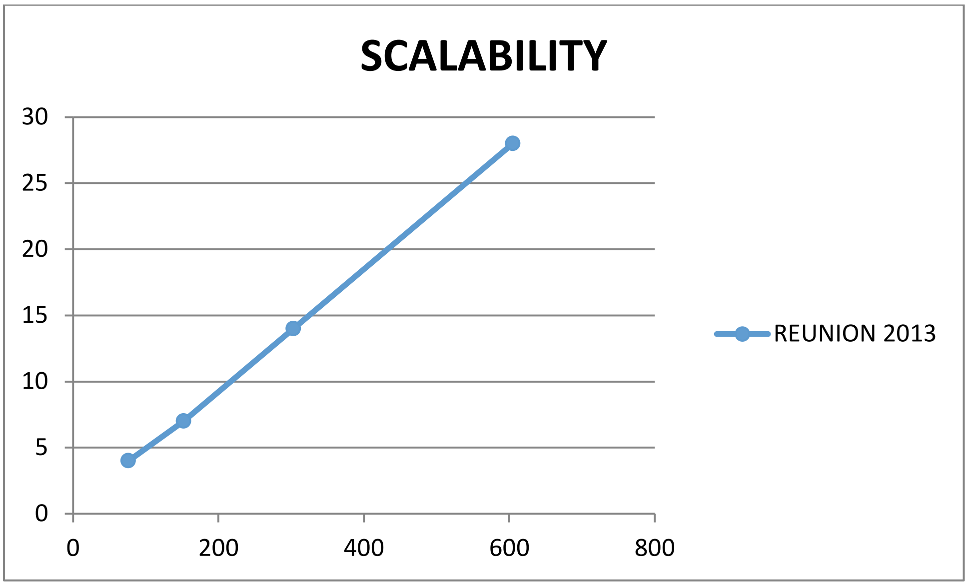
FIGURE 13: CLASSIFICATION DETAIL. AN OUTLIER IS INDICATED BY AN ARROW.

One point is clearly an outlier. The other most distant points lie in steep areas where the distance computation gives a large overestimate compared to a 3D distance computation. However, the investigation also suggests that if the areas of this data set with the largest point-surface distances are critical for further computations, then a more accurate approximation (a 3D surface would be able to reproduce the steep areas more accurately) or a different surface representation (triangulation) could be beneficial.

5.3.3 Quality

As mentioned in Section 5.2.3 the accuracy of the evaluation is not an issue. However, Figure 13 illustrates that the measured distance will give an overestimate compared to the 3D distance in steep areas. This service will not be modified to take 3D distances into account, but in the next project period, we will provide some functionality to analyze points being identified by this service with more detail.

5.3.4 Scalability



FIGUR 12: COMPUTATIONAL EFFORT WITH RESPECT TO DATA SIZE

Distances between the REUNION 2013 data sets of varying sizes, see Section 4.5, and the surface shown in Figure 11 are computed. The results are summarized in Figure 14. The data size is given in MB along the x-axis and the time in seconds along the y-axis. The computations are performed on a Laptop with 15.6 GB memory and 8 Intel@Core i7-4702HQ CPU cores. The service consists of a small pre process which is dependent on the surface size, but not on the point cloud, but most of the time is spent in evaluation. This is done point-by-point and scales completely linearly. The service is multi-threaded. The computational time shown in Figure 12 excludes the time needed for reading the input data and writing the result to file.

5.3.5 Degree of human intervention

This service is designed for interactive use and human intervention. The classification parameters should be set according to the needs of the user and output information from the applied approximation service give input regarding how to classify the points. Two classification options exist:

- Uniform classification with regard to distance. A maximum distance and the number of levels between zero distance and the given maximum are specified. Points being more distant than the given maximum on both sides of the surface are classified into separate groups.
- Distance specification. All distance levels are given from the application and mirrored around zero distance.

5.4 #90 TRIMMING OF LR B-SPLINE SURFACE, SINTEF

An LR B-spline surface has a rectangular domain. Assume the surface model elevation and is parameterized on the xy-domain of the point cloud from which it was created. Then the surface will extend beyond the domain of the point cloud unless the cloud happens to be rectangular in x

and y. To ensure that the surface corresponds to the point cloud, a bounded surface is defined by trimming the initial surface with regard to the extent of the input points.

LR B-spline representation of terrains is used in the Marine Scenario. In MS4, these surfaces are evaluated to define rasters which again serves as input for change detection. If the domain of the surface does not correspond to the domain of the initial point cloud, false results may be obtained in the context of change detection.

5.4.1 Method and implementation

The service defines the trimming curves from a point cloud where no particular structure is expected. The surface represents the elevation of the point cloud and is parameterized on x and y. No additional boundary information is required.

The trimming curves bounds the point cloud in x and y. They are computed through a recursive procedure where the point cloud is divided regularly into sub clouds. Boundary sub clouds are divided further until a recursion level deduced from an input parameter (the tightness parameter) is reached. Each final sub cloud is bounded by a rectangle and these rectangles are combined to create an outline of the point cloud. This results in one or more polygons. The polygons are split at sharp corners in the underlying directional trend and each piece is approximated by a spline curve in x and y. Finally a bounded surface is defined from an initial non-trimmed LR B-spline surface approximating the given point cloud and the constructed trimming curves.

5.4.2 Example

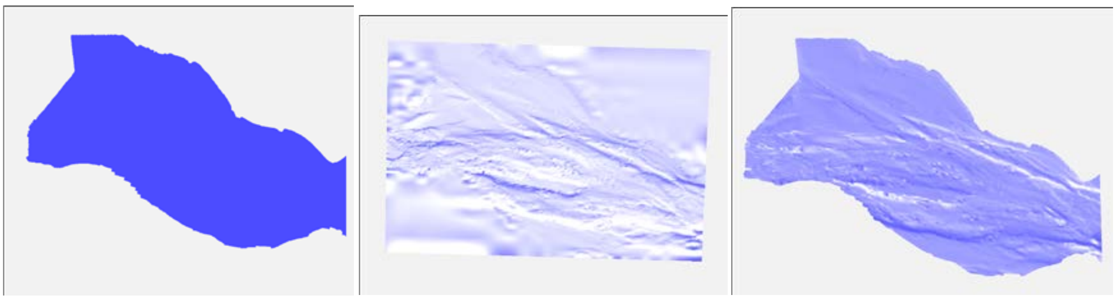


FIGURE 15: POINT CLOUD, APPROXIMATING LR B-SPLINE SURFACE AND THE FINAL TRIMMED SURFACE

In Figure 15, the point cloud HRW 143513 4 from Table 2 (Section 4.3) is approximated with an LR B-spline surface. This service is used to trim the surface according to the extent of the point cloud. A tightness parameter of 7 is used.

5.4.3 Quality

In the context of trimming, quality is related to how well the trimming curve follows the outline of the point cloud. The trimming curve should follow the point cloud boundary closely, but not adapt to single points in areas where the point cloud is sparse. The tightness is governed by a user parameter, see Section 5.4.4. Points with a reasonable distance to other points in the cloud should not be excluded from the trimmed surface. However, the service cannot guarantee this feature. In the first step in the algorithm the point cloud is bounded by poly lines, in the second step these lines are approximated by curves using a tolerance dependent on the length of the

line segments. The resulting curve may exclude some points. To get an impression of the quality, consider the following example. The last trimmed surface in Figure 20 (Section 5.4.5) is compared to the point cloud REUNION 2013 4 from Table 2 (Section 4.5). In this case the trimming curves are computed with tightness set to its maximum allowed parameter (tightness parameter = 8). This point cloud represents a worst case scenario due to its complex shape.

Figure 16 presents an overview. 239 points out of 3.96 millions are found to lie outside the trimming loop. Looking at the details in Figure 17, the points are scattered and lie very close to

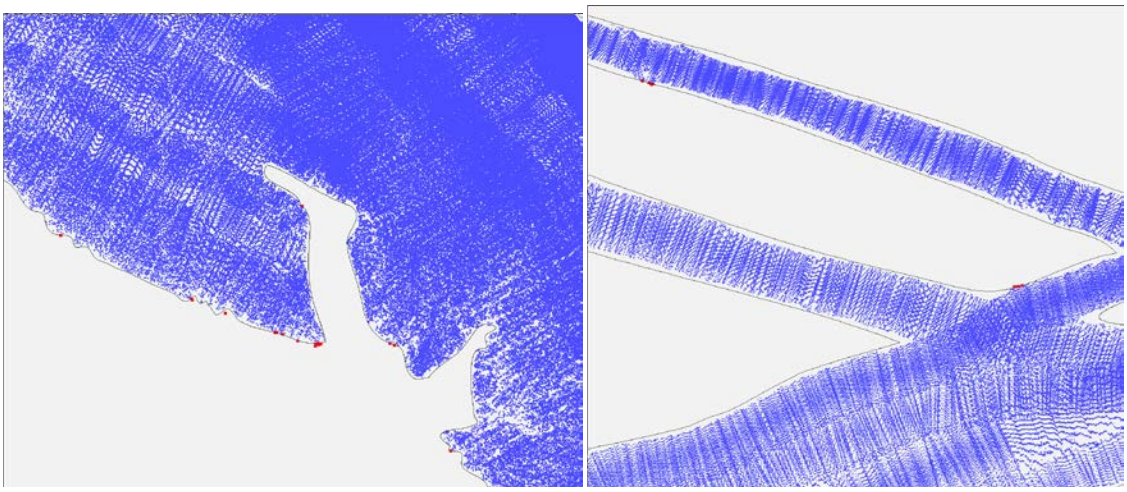


FIGURE 17: DETAILS, TRIMMING LOOP, POINTS INSIDE THE SURFACE (BLUE) AND OUTSIDE (RED)

the trimming loop. In most cases, the distance between these points and their nearest neighbours are larger than the typical distance within the point cloud. The trimming loop is quite close to the point cloud.

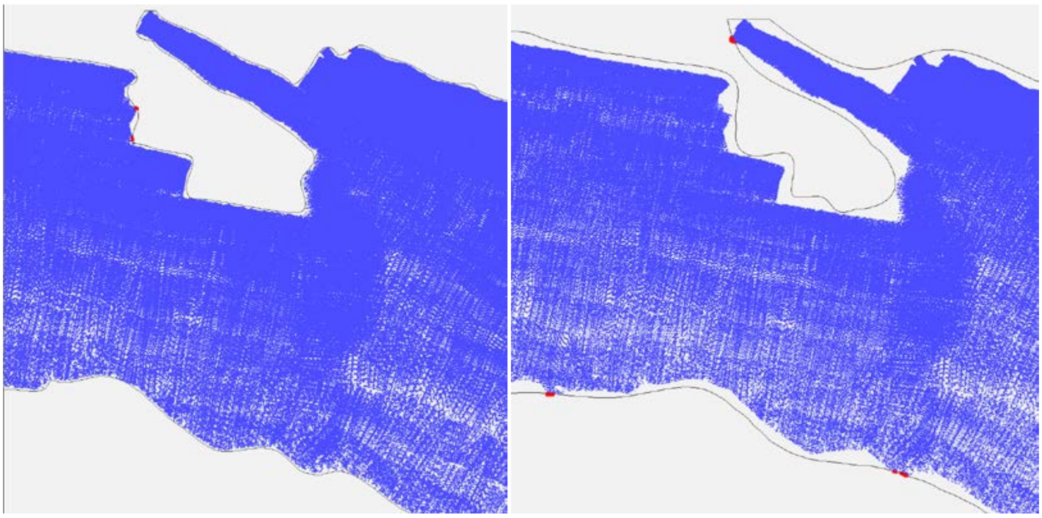


FIGURE 18: DETAIL FOR TIGHTNESS=8 AND TIGHTNESS=6, TRIMMING CURVE, POINT CLOUD (BLUE) AND POINTS OUTSIDE THE TRIMMING LOOP (RED)

Testing the trimming result for tightness=6 results in 412 points outside the trimmed surface. The polygonal bounding the surface tend to lie further away from the point cloud, but the tolerance used in the curve approximation is increased compared to tightness=8. In Figure 18, we can see the same trimming curve details for the two trimmed surfaces.

The result is good, but not perfect. Possible extensions/modifications to the service is:

- Apply service #11 from T4.2 to compute the spatial extent of the point cloud. This service uses Alfa Shape Extraction. The result of service #11 is used to trim the surface.
- Provide an additional service that takes a shape file describing the point cloud domain as input
- Move the trimming curve further away from the point cloud. The distance should reflect the tolerance used in the trimming curve approximation. This will, however, result in a less tight bound.
- Extend the service to take exclusion information as presented in this test into consideration. It would require some effort to achieve a reasonable performance of the function checking the result.

A decision regarding the options above will be taken when detailed planning for D4.4.3 is taking place.

5.4.4 Scalability

The scalability of this service is related to the size and configuration of the input point cloud. Also an LR B-spline surface is input to the service, but its purpose is only to be the underlying surface in the resulting trimmed surface. It does not enter the computations. To be able to distinguish the data size from the data configuration, we will look at two different point clouds and corresponding non-trimmed surfaces.

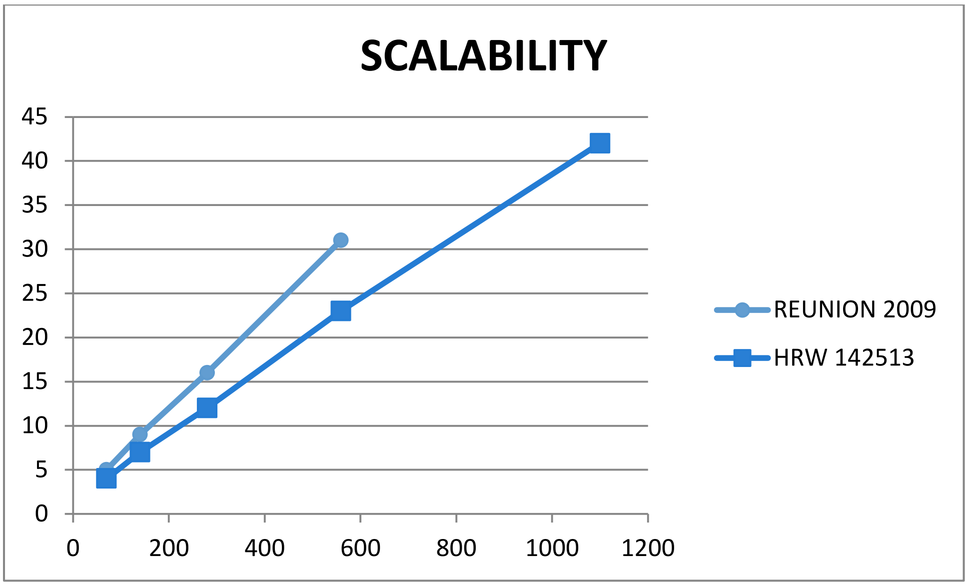


FIGURE 19: TRIMMING WITH RESPECT TO POINT CLOUD, DATA SIZE IN MEGABYTES ALONG THE X-AXIS, COMPUTATION TIME IN SECONDS ALONG THE Y-AXIS

Figure 19 illustrates the relation between size of the point cloud and computation time for the 2009 REUNION data set presented in Section 4.5 and the data set originating from HRW, see Section 4.3. The HRW point cloud is compact and has a relatively simple outline while the REUNION point cloud has several holes and a very complex shape. The data sizes are measured in megabytes. The surface size corresponding to REUNION 2009 is 3.0 M and to the HRW point cloud is 6.5 M. In both cases a tightness parameter of 7 is applied. As we can see, the computations with the point cloud having the simpler outline are faster. The scalability is different, but stays within the same magnitude. The main conclusion is that the service scales

linearly. The computations are performed on a Laptop with 15.6 GB memory and 8 Intel@Core i7-4702HQ CPU cores.

5.4.5 Degree of human intervention

The service has one sensibility parameter influencing the density with which the trimming curve bounds the point cloud. This parameter is translated into a maximum recursion level for the computations and information on splitting the point cloud into sub clouds. The parameter should lie in the interval [1,8]. For point clouds of high and relatively uniform density, it should

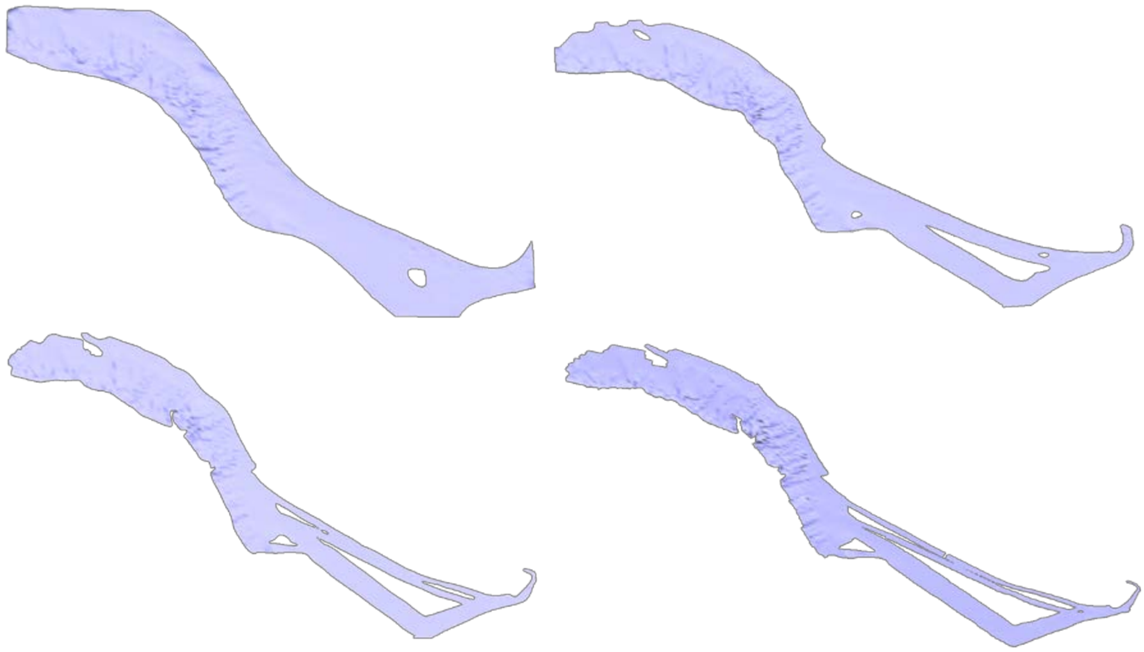


FIGURE20: TRIMMING OF SURFACE APPROXIMATING THE REUNION 2013 POINT CLOUD USING TIGHTNESS PARAMETER 2, 4, 6 AND 8.

be in the upper end of this range, but be decreased for point clouds with very varying density.

Figure 20 illustrates the effect of this tightness parameter on the resulting surface. For the value 2 the trimmed surface only roughly follows the outline of the point cloud. The accuracy is improved gradually and for a tightness parameter of 8, the outline of the trimmed surface corresponds quite tightly to the domain of the point cloud, see also Section 5.4.3.

5.5 #91 LR B-SPLINE SURFACE TO RASTER, SINTEF

An LR B-spline surface representation is capable of representing local details without a global increase in surface size. Thus, one such surface can be harvested to produce rasters of varying cell sizes without sacrificing the accuracy.

This service belongs to work flow 4 of the Marine Scenario where the LR B-spline representation of the sea bottom needs to be prepared to give appropriate input for the change detection service which is the essence of this work flow.

5.5.1 Method and implementation

A regular grid is defined from a requested cell size and the parameter domain of the surface or from a user specified domain. The LR B-spline surface is evaluated regularly with respect to this grid applying the same pre processing step as for service #86 and #88. If the surface is trimmed, the points lying outside the trimming loop(s) are "removed" by assigning them a no-data value specified by the application. Finally, the obtained raster is written to a geotiff file using the library geotiff2 as a front end.

5.5.2 Example

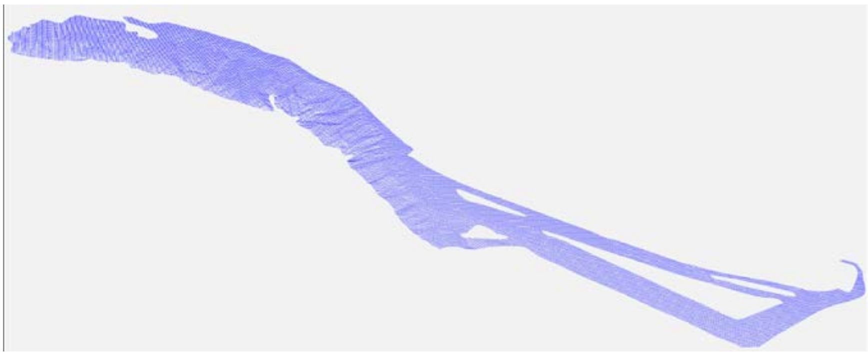


FIGURE 21: 10 X 10 RASTER CORRESPONDING TO THE 2013 REUNION DATA SET

The trimmed LR B-spline surface computed with tightness parameter 6 is used to generate the raster in Figure 21. The cell size is 10 by 10 meters.

5.5.3 Quality

This service is merely a translation between representations. The quality aspect is related to the services prior to this one in the work flow chain, namely service #9 and service #90, see Section 6.1.4 and 3.4.4. The services #86 and #88 can be used to check the quality of the surface representation, see the Sections 5.2 and 5.3.

5.5.4 Scalability

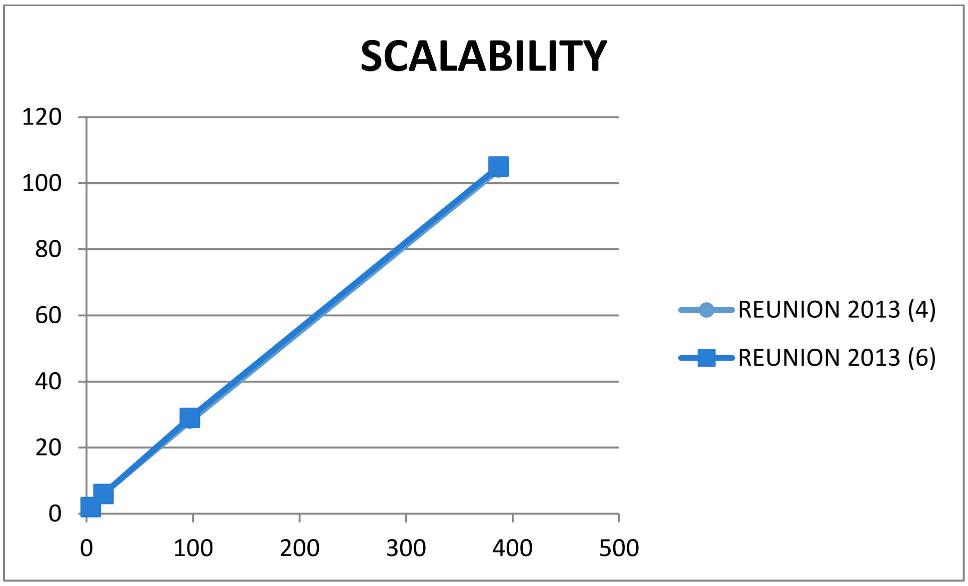


FIGURE 22: RASTER COMPUTATION WITH REGARD TOTHE REUNION 2013 DATA SET

Figure 22 shows the scalability of the raster computation. The input surface is approximating the 2013 REUNION survey and the tightness parameters 4 and 6 is applied for the trimming of this surface, see Section 5.4.5 for the effect of this parameter. The unit along the x-axis is the size of the output raster while the unit along the y-axis is the computation time in seconds. The complexity of the trimming loop has minimal effect on the computation time. The computations are performed on a Laptop with 15.6 GB memory and 8 Intel@Core i7-4702HQ CPU cores.

The HDFS file system is block based and this does not correspond well to the random file access used by geotiff. To support locally mounted HDFS for file access, the possibility of storing the output file locally and then move it to the final destination is provided. This solution does not scale if the file size grows beyond the available local memory. However, the problem of scalability is not believed to be a problem for this service as the size of the LR B-spline surface will be limited. In practice the service scales linearly.

5.5.5 Degree of human intervention

The application must specify the cell size and the no-data value. Note that geotiff2 which is used as a front end towards the geotiff file does not support a no-data value entity. Thus, it is important to specify a value which is recognized by the receiver of this file.

The service is intended for use in change detection. Thus several surfaces created from surveys of different dates are given as input to this service. As the surveys may have different domains, see Section 4.5, the parameter domains of the associated surfaces will also differ. To enable consistent raster representations, this service offers the possibility of specifying the domain from which the raster is generated. The intersecting domain of the relevant surfaces should be used for the creation of the rasters.

5.6 #85 STITCHING OF LR B-SPLINE SURFACES, SINTEF

A seamless elevation model is central in the Marine Scenario. This model may consist of more than one surface. In the LR B-spline surface case, this service ensures that a regular collection of surfaces build a seamless model.

LR B-spline approximation reaches a limit with regard to the size of the point cloud being approximated by one surface and the size of this surface, see Section 6.1. The recipe to handle large data sizes is tiling and stitching. This service represents the last part of this process. The context is the Marine Scenario, but the service is relevant in all situations where LR B-spline surfaces are used to approximate large point clouds or represent large geographical areas with high resolution.

The service will become a part of a chain of services performing LR B-spline approximation of large point clouds in the next project period.

5.6.1 Method and implementation

The starting point is a regular collection of *n x m* LR B-spline surfaces. They do all have the same degree and non-overlapping and continuous parameter domains. Not all surfaces need to be defined and the surfaces may be trimmed.

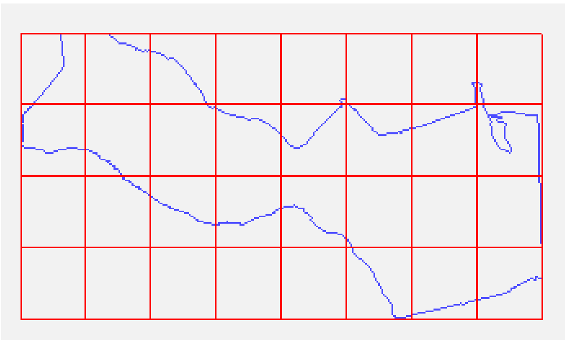


FIGURE 23: REGULAR DOMAIN CORRESPONDING TO SURFACE COLLECTION

Figure 23 shows the configuration of the surface domains (red) along with the outline of trimming loop (blue). The point cloud is the complete 473513 survey from HRW translated to UTM coordinates, see Section 4.3. The surfaces are modified along the boundaries and in the corners to obtain a seamless elevation model.

C^0 continuity between two surfaces is achieved if corresponding surface coefficients at the common boundary are identical and C^1 continuity if boundary coefficients and the next rows of coefficients in both surfaces are positioned on a line when a coefficient and the associated parameter values are viewed as a 3D point. Around a corner, the closest coefficients from all surfaces must lie in a plane. This construction presupposes a certain regularity of the spline spaces which is obtained in two steps:

- The closest knot lines along the common boundary are extended to become continuous throughout the surface. This is illustrated in Figure 24
- The two surfaces are ensured to have corresponding knot lines orthogonal to the boundary in the vicinity of this boundary. This is illustrated in Figure 25 and 26.

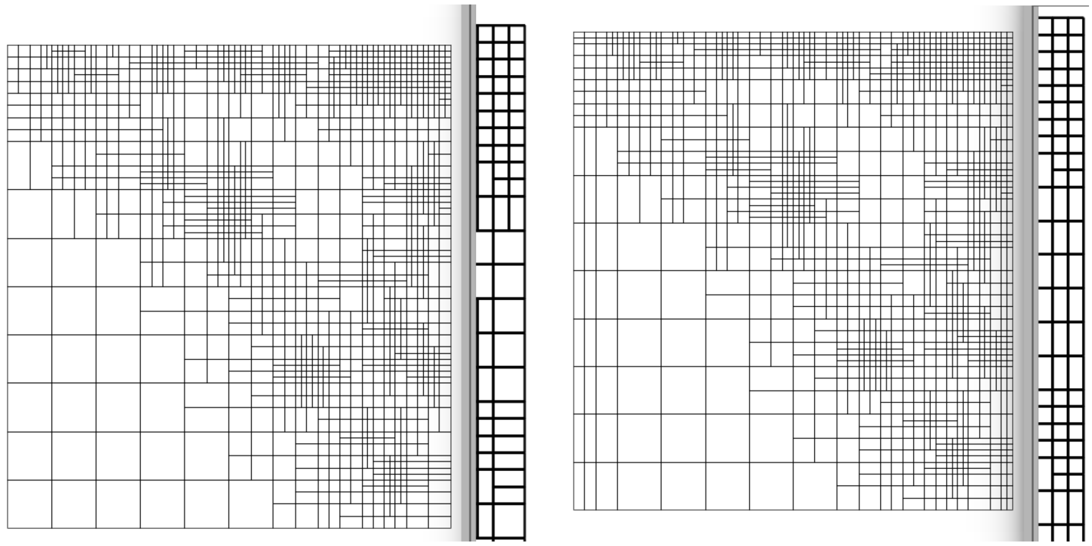


FIGURE24: KNOT LINES OF SURFACE BEFORE AND AFTER INSERTING KNOTS ALONG BOUNDARIES, ONE PART OF THE BOUNDARY IS HIGHLIGHTED

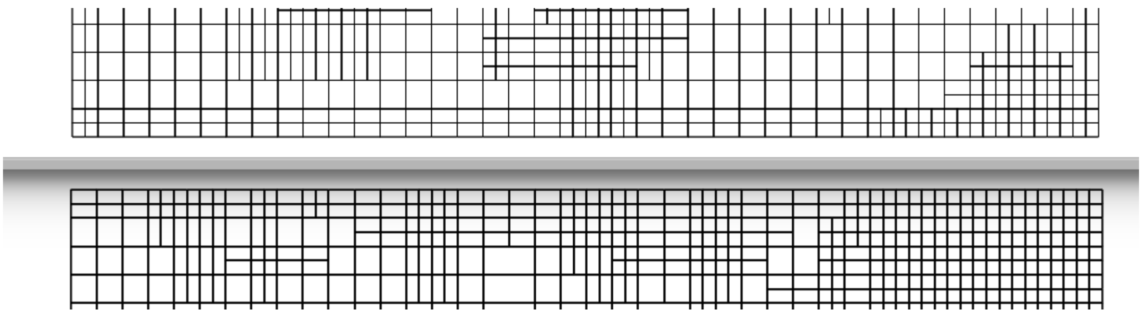


FIGURE 25: KNOT LINES CLOSE TO COMMON BOUNDARY BEFORE THE SURFACES ARE REFINED

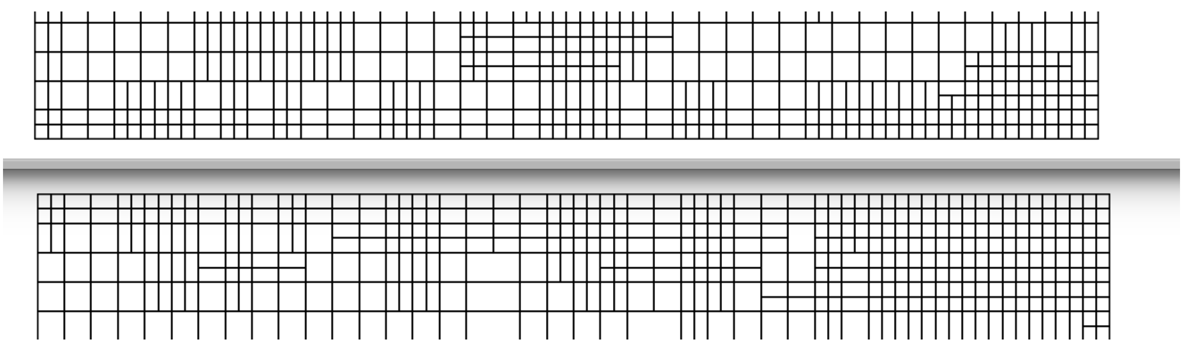


FIGURE 26: KNOT LINES AFTER THE SURFACES HAVE GOT COMMON SPLINE SPACES CLOSE TO THE COMMON BOUNDARY

5.6.2 Example

The complete 473513 survey has got 131 million points and this is too much to be approximated by one LR B-spline surface. The data set is split regularly into 8 times 4 overlapping tiles. Due to the shape of the point cloud, some tiles are empty while the others contain up to about 12.8 million points. Each tile is approximated using service #9 with tolerance 0.5 and 3 iterations, see Section 6.1. Then the surface is restricted to the domain of non-overlapping tiles and trimmed using service #90. This is where the current service starts and the setup is visualized in Figure 27 and 28.

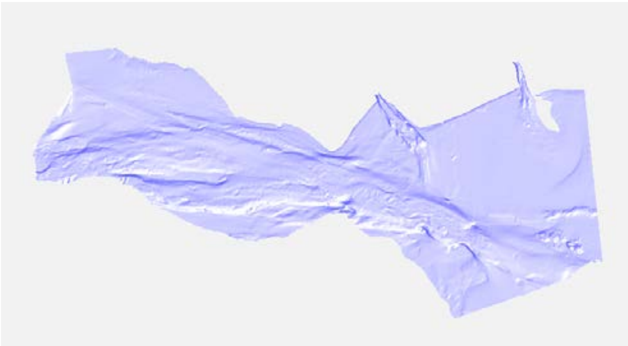
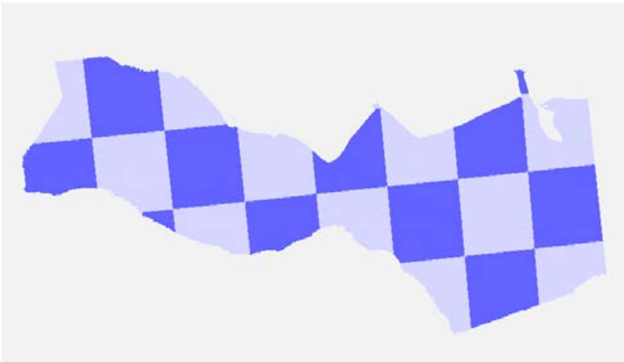


FIGURE 27: THE SURFACE SET APPROXIMATING THE POINT CLOUD



FIGUR 28: SURFACE CONFIGURATION

The surface set after stitching is shown in Figure 29. Visually, there is very little difference between the surfaces in Figure 27 and the surfaces in Figure 29. The accumulated file size has increased from 3.1 M to 3.4 M. This data increase may be reduced. We will look more into this subject in the next period.

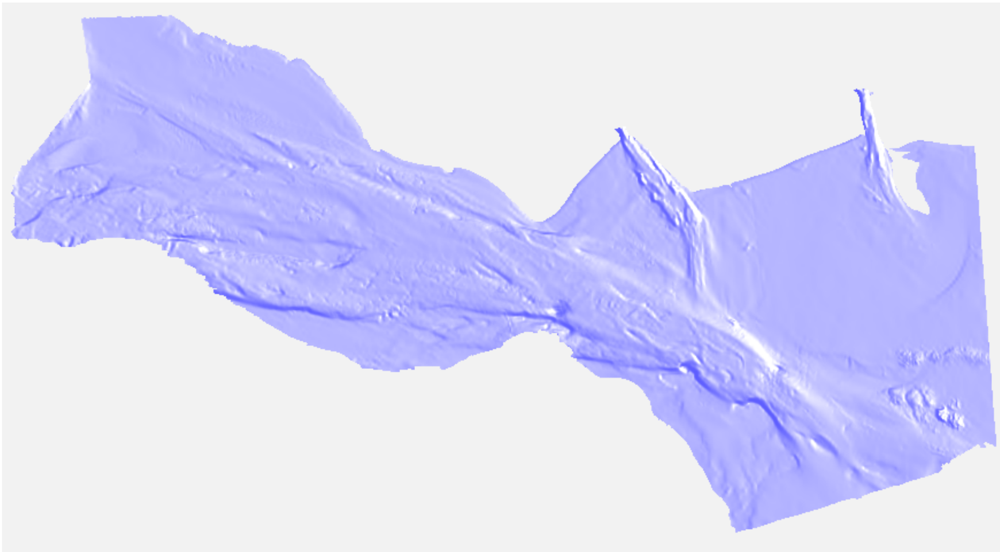


FIGURE 29: SURFACE SET AFTER STITCHING

5.6.3 Quality

A high quality result of this service is exact C^1 continuity between the surfaces with a minimum change in elevation. The figures for the example above is

Surface set	Maximum C^0 discontinuity	Maximum C^1 discontinuity
Before stitching	13.2269 m	0.178585 m
After stitching	$1.28e^{-10}$ m	$2.07e^{-11}$ m

TABLE 7: SURFACE DISCONTINUITIES BEFORE AND AFTER STITCHING

The discontinuity measurements are computed with regard to the non-trimmed surfaces approximating the tiles and the largest numbers before stitching appears outside the trimming loops where the surfaces don't adapt to any points. The discontinuities are, thus, not visible in Figure 27. The C^1 discontinuity figures represent the difference between the surface derivatives along common boundaries.

The modification changes the surface, but the extent of the change is not investigated. This service will be subject to further development. Then the service quality will be analyzed further.

5.6.4 Scalability

The execution time depends on the number of surfaces in the grid and the size of the input surfaces. The surface size should be kept at a moderate level due to all operations performed on the surface, not only this one.

5.6.5 Degree of human intervention

The service requires no human intervention, but has one optional parameter. When this parameter is set, the accuracy information reported in Table 7 is computed and written to standard output. The computation is sampling based and affects the performance of the service slightly.

5.7 #110 3D DELAUNAY TRIANGULATION, IGN

This service is a pre-process to service #95, see Section 5.8.

5.7.1 Method and implementation

The mesh is computed as a 3D Delaunay triangulation using CGAL. (in this resume, the distinction between vertex and point is made : a vertex is a structure of the triangulation which is represented by a 3d point) Figure 30 shows the triangulation data structure of CGAL.

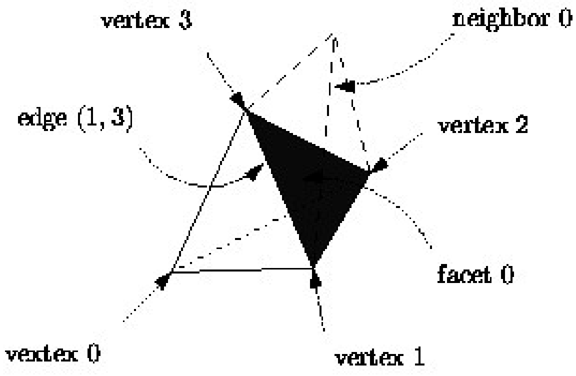


FIGURE 30: CGAL TRIANGULATION DATA STRUCTURE

In 3D, the worst case complexity of a triangulation is quadratic in the number of points. For Delaunay triangulations, this bound is typically reached when points are equally distributed on two non-coplanar

lines. However, the complexity of a Delaunay triangulation is linear or close to linear in the number of points. Several articles have proved such good complexity bounds for specific point distributions, such as points distributed on surfaces under some conditions.

5.7.2 Example

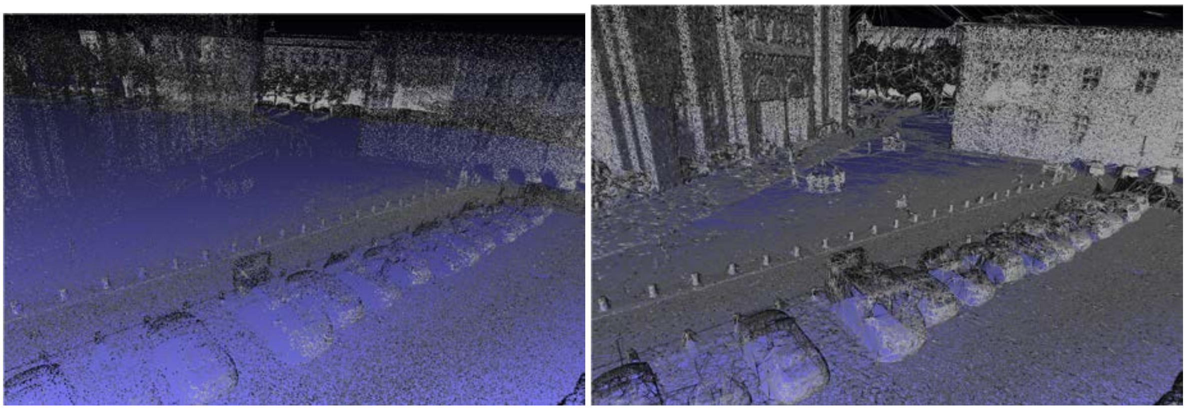


FIGURE 31: A PART OF A POINT CLOUD CONSISTING OF 3791823 POINTS AND THE CORRESPONDING 3D TRIANGUALATION

5.7.3 Scalability

We are currently working on a common representation for 3D triangulations that is adapted to big data. The proposed format is designed to store a set of simplicial meshes (2D and 3D) representing a big simplicial complex tiled according to some spatial or semantic rule.

In order to scale the Delaunay triangulation algorithm, we will adapt the method introduced in [5]: an approach that allows to process the Delaunay triangulation for a well-distributed data sets in 2D and 3D.

The method computes a billion-triangle terrain representation from 11.2 GB of LIDAR data in 48 minutes using only 70 MB of memory on a laptop.

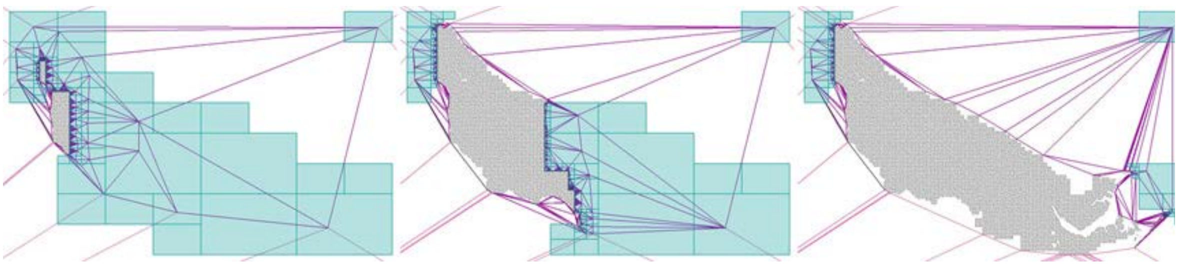


FIGURE 32 : EXAMPLE OF LARGE SCALE TRIANGULATION FRAMEWORK FROM THE ORIGINAL PAPER

5.7.4 Degree of human intervention

The proposed approach is fully automatic.

5.8 #95 WATERTIGHT SURFACE RECONSTRUCTION, IGN

Surface reconstruction from points has already been studied extensively. The first reason is the ill-posed characteristic of the surface reconstruction problem. The second reason of this interest is the constant increase of both the number of applications that use reconstructed surfaces (Digital elevation models, computation for flood simulation or pathplanning) and the data acquisitions sizes (Lidar, images...). These aspects result in multiple and various methods based on different assumptions related to data perturbations. The IQmulus context is relevant for these cases. It is against this background that we propose, instead of choosing a given data representation, surface prior and specific output, a method that takes generic input data, and produces a watertight surface of the scene with a confidence metric.

Key contributions are :

- Joint and unified handling of Lidar and photogrammetric depth measurements as evidences that a 3D segment from the sensor (camera or lidar) to the measurement is free space and that space is occupied immediately behind the measurement
- Handling of the heterogeneous sampling rates (airborne, MMS, static, image, lidar...) and location uncertainties of all measurements
- Production of a watertight surface with a confidence metrics associated to each triangle

The result of the method is a volumetric segmentation with a multi-label representation where each label is related to the confidence of the segmentation. A surface can be extracted with an empty/occupied segmentation by defining a threshold for the confidence label at e.g. 50%, which enables the extraction of the resulting mesh as the set of triangles separating an empty-labelled tetrahedron from a occupied-labelled one.

5.8.1 Method and implementation

Our formulation is based on the segmentation of 3D space. The goal of this approach is to segment the space into volumes that are inside (objects, trees, ground, buildings...) and outside (air) with a global smoothness criterion that penalizes solutions with important surface. This leads to a global optimization formulation that can be solved efficiently with a graph-cut approach. Instead of producing a binary segmentation, the result is a segmentation of the space where each label is a confidence to be inside or outside an object. This gives a quality criterion for each area of the reconstructed surface. The algorithm is decomposed into three steps:

- First, the scene is segmented in a tetrahedral mesh using the Delaunay triangulation (service #110).
- Then for each cell, a score that shows the confidence of the cell to be inside or outside is computed (Image c in Figure 33).
- The global optimization problem is minimized with the alpha-expansion algorithm (Image d).

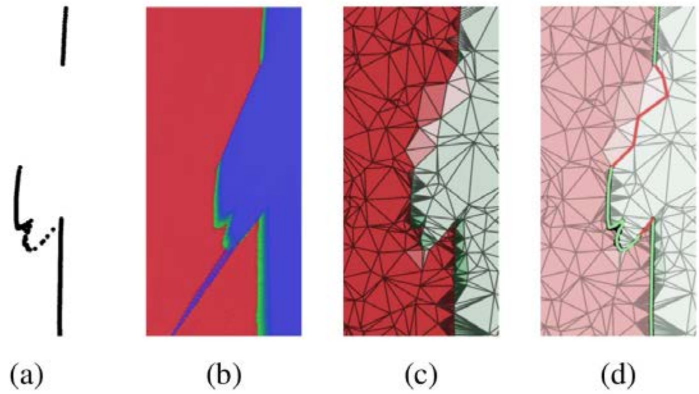


FIGURE 33: DECIDING WHETHER A CELL IS INSIDE OR OUTSIDE A BOUNDARY, CURVE VIEW

5.8.1.1 Details

The aim of the proposed method is to find, for a set of inputs *I,* a segmentation of the space as *occupied* or *empty* following a suitable model. Inputs are a set of mass functions m_i defined by

$$m_i(P) \rightarrow [0,1]^3; P \rightarrow (e, o, v) \text{ s.t. } e + o + u = 1$$

where *e* ,*o* and *u* are respectively the occupancy masses for a 3D point *P*∈*R3* to be *empty*, *occupied* or *unknown*. The mass functions are then merged to have a global mass function

$$m(P) = \bigoplus_{i \in I} m_i(P)$$

m(*P*) is the final mass that gives, for a point *P*∈*R3* , the confidence to be *empty*, *occupied* or *unknown* for the set of mass function *I.*
The problem is then defined as a multi-label segmentation of the space regarding to *m* where the labels set is $L = \{0, \frac{1}{(N-1)}, \frac{2}{(N-1)}, \dots, \frac{(N-2)}{(N-1)}, 1\}$. Each label *l*∈*L* represents the occupancy confidence where *l=1* represents a strong confidence to be *occupied* and *l=0* a strong confidence to be *empty*.
The space is discretized with a 3D Delaunay triangulation, where
T is the set of tetrahedra and $F \subset T^2$ the set of facets represented by a pair of adjacent tetrahedra.
We note l_t the label of the tetrahedron $t \in T$ and $l_T = (l_t)_{t \in T}$ the set of label l_t of the triangulation *T.*
Finally, the goal is to find, for each tetrahedron of the triangulation, a label l_t that encodes the confidence of the tetrahedron in being either *occupied* or *empty* for a given set of input mass functions m_i . To this end, the problem is formulated as an energy minimization framework composed of two terms: the data term $E_{data}(l)$, which represents how a label is close to the global mass function *m* and the smoothness term $E_{prior}(l)$, which penalize solutions with important interfaces between labels. The final energy is:

$$\min_{l_t \in L^T} E_{data}(l_t) + \lambda E_{prior}(l_t)$$

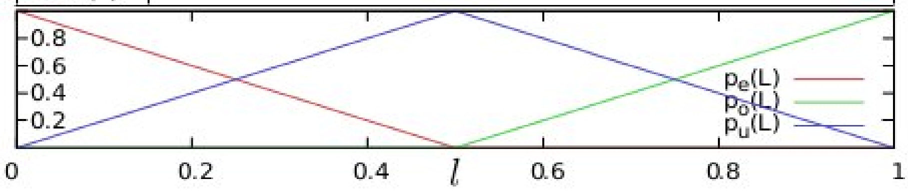
where λ is the weight of the prior compare with the data term. Now the two terms will be described in details.

Data term

The data term models how a label *l* fits to the global mass function *m* defined above. To take into account the *unknown* label, the confidence of a label *l* to be *occupied* is decomposed into 3 functions $p_e(l)$, $p_o(l)$ and $p_u(l)$ defined as:

$$p_i(l) = \begin{cases} 2(0.5 - l) & \text{if i=e} \quad \text{and } l < 0.5 \\ 2(l - 0.5) & \text{if i=o} \quad \text{and } l > 0.5 \\ 1 - 2(|l - 0.5|) & \text{if i=u} \\ 0 & \text{else} \end{cases}$$

l	0	0.2	0.4	0.6	0.8	1
$p_e(l)$	1	0.6	0.2	0	0	0
$p_o(l)$	0	0	0	0.2	0.6	1
$p_u(l)$	0	0.4	0.8	0.8	0.4	0



The previous figure shows an example with 6 labels. When =0.2, the corresponding scores are $p_e(l) = 0.6$, $p_o(l) = 0$, and $p_u(l) = 0.4$. An interesting situation is the non-inclusion of the case $p_e(l) = 0.6$ and $p_o(l) = 0.4$, which can also be represented by the input $l = 0.6$. It tells that the solution is a conflict or a fuzzy zone. This allows to move the *CONFLICT* case into the *UNKNOWN* case. However, one can imagine another cost function for conflict, change detection, accuracy measurement.

Finally, the data term is the integral of the difference between the function $p_i(l)$and the mass function $m_i$for each label i ∈ (i,o,u)over the volume of the tetrahedron, which gives:

$$E_{data}(l) = \sum_{t \in T} \sum_{f \in \{e,o,u\}} \int_t |p_i(l_t) - m_f(v)| \mathrm{dv}$$

Prior term

The prior term in surface reconstruction forces solutions with small surface. This idea is extended to the multi-label case by minimizing the surface multiplied by the norm between label probabilities, which leads to the following term:

$$E_{prior}(l) = \sum_{t_1,t_2 \in F^2} \mathrm{area}(t_1 \cap t_2) |l_{t_1} - l_{t_2}|$$

Here $\mathrm{area}(t_1 \cap t_2)$ denotes the area of the triangle $t_1 \cap t_2$. The important property is that the distance from the *OCCUPIED* area to the *EMPTY* area is always equal to one regardless of the number of labels.

5.8.2 Example

FIGURE 34: JOINT RECONSTRUCTION FROM AIRBORNE AND TERRESTRIAL (MMS) LIDAR OF THE SAME SCENE

In this example, we show how the DST helps for modelling different types of data. The data input is both airborne and terrestrial Lidar data. Two airborne data sets are overlapped to cover a wide area. A terrestrial point cloud is included at the center of the scene for a final 5.2M points set. Airborne and terrestrial Lidar are both represented as a beam. Results show that both data sets are taken into account to produce a mesh with both fine details from the ground based Lidar and a global watertight surface from the airborne data. The small thickness parameter on airborne data allows reconstruction of high buildings. Thanks to the small thickness and extra points on the terrestrial data, pillars are well meshed. The close up view shows that the merging between airborne and terrestrial point is well managed.

5.8.3 Quality

The multi label segmentation enables the output of a confidence metrics on each triangle of the reconstructed surface that states to which degree using this triangle in the final mesh was a clear decision. Due to the confidence metric, information on the quality of the service results is automatically provided.

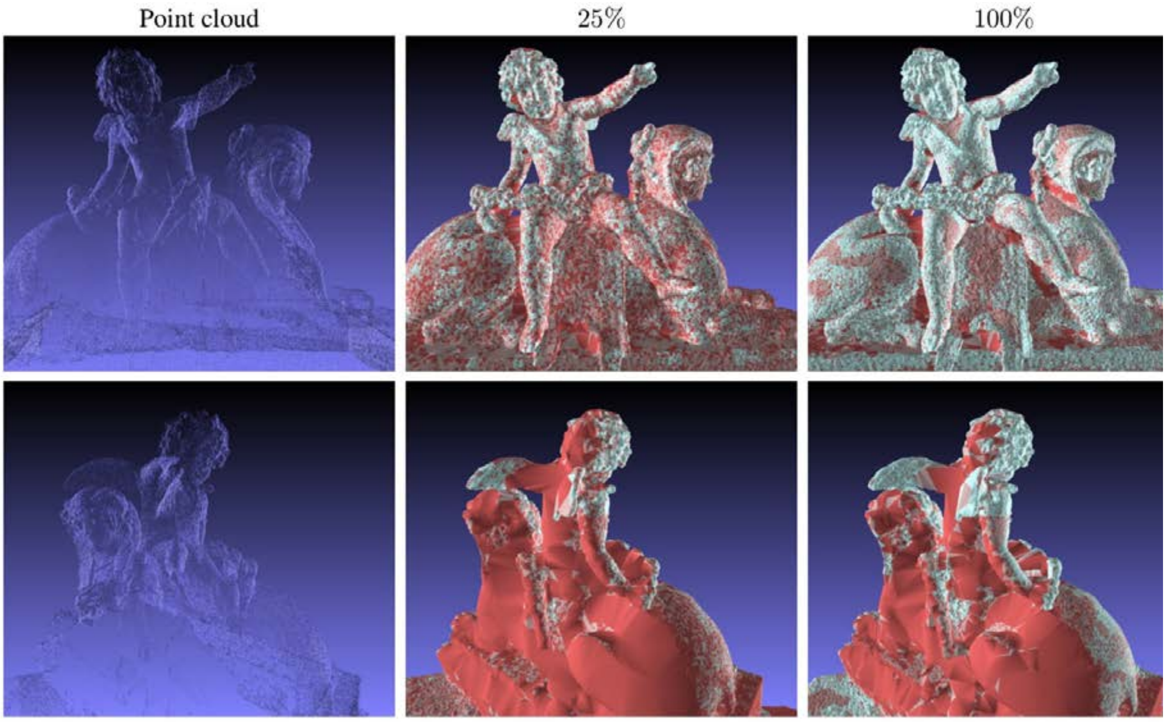


FIGURE 35: RESULT OF THE SPACE SEGMENTATION WITH 4 LABELS ON STRUCTURE FROM MOTION DATA. FIRST COLUMN : THE 3D POINT CLOUD GENERATED, SECOND COLUMN : THE SPACE SEGMENTATION GENERATED WITH ALL POINTS AND THIRD COLUMN : WITH 25% OF POINTS WITH THE SAME PARAMETER. THE FACET COLOR INDICATES THE DIFFERENCE OF PROBABILITY BETWEEN THE TWO NEIGHBORING TETRAHEDRA. THE WHITER THE FACET, THE BIGGER THE MORE CERTAIN THE SURFACE IS LOCALLY.

5.8.4 Scalability

The three steps of the approach need specific implementations in order to take into account an arbitrary number of inputs.

- The 3D triangulation : see Section 5.7.3
- The score computation : This task is embarrassingly parallel
- The multi-label optimization : this may be tackled on arbitrarily large graphs using open source libraries such as GraphLab

The score computation is processed by a ray tracing into a Delaunay triangulation. The current version of this service is single core, but will be extended to handle big data in the next project period using MapReduce. Each ray tracing can be easily done in parallel (map function) and the global score computed with the reduce function .

No scalability results are currently obtained.

5.8.5 Degree of human intervention

The proposed approach is fully automatic.

6 UPDATED SERVICES

This section presents services from D4.4.1 which has been subject to major revisions during the last year. Many services have been extended to accept a broader range of input and output formats. For instance service #9 does now accept point clouds on las, xyz and g2 formats while the distance field which is an optional outcome from the service can be represented in a simple text format and as ply. The g2 format is mainly used for debugging purposes, see Section 3.5. Service #55 which creates an LR B-spline surface from a raster can now take input as ENSI Ascii or geotiff format. Most services have been updated with bug fixes and smaller adjustments. This is seen as normal maintenance and will not be given further attention in this report.

Service #	Language	Os	Serial / parallel / distributed	Metadata available	On artifactory	Smoke test passed	Logging enabled	Scalability test performed
#9	C++	Linux	Parallel	Yes	Yes	OK	Yes	Locally
#40	C++, Fortran	Linux-like	Parallel*	Yes	Yes	For D4.4.1	Yes	Locally

TABLE 8: OVERVIEW OVER SERVICES FROM D4.4.1 WHICH HAVE BEEN SUBJECT TO MAJOR UPDATES

* The service may be executed as a multi-node

The table above gives an overview of the services updated for D4.4.2. In this section, the services will be presented in some detail.

6.1 #9 LR B-SPLINE APPROXIMATION FROM POINT CLOUD, SINTEF

Creating an LR B-spline surface from a point cloud is an important functionality in the Marine Scenario and service #9 is the basis for most of the IQmulus services related to LR B-splines. LR B-splines is a new representation format that was introduced in [1].

6.1.1 Motivation for the update

The version of service #9 which was delivered in D4.4.1 provided a proof of concept for representation of geographical information by LR B-spline surfaces. The service showed the ability to extract the general landscape structures into a smooth surface. Also smooth formations were well represented by an adaptive procedure where an initially lean surface were refined and updated to encompass more and more detail. However, the chosen approximation method (least squares approximation with a smoothing term) did not facilitate maintaining this procedure sufficiently long to give an accurate representation of detail. Furthermore, the data sizes handled by the service were too small for practical use.

6.1.2 Modifications

The main modifications are related to the introduction of an alternative approximation method which is used in combination with least squares approximation and adoptions of the algorithm to handle larger data sizes. In addition bug fixes and smaller modifications increase the stability and usability of the service.

6.1.2.1 Multilevel B-spline approximation applied in an LR B-spline setting

In the report of D4.4.1, LR B-splines are described in some detail. To repeat, we see that an LR B-spline surface F is expressed with respect to parameters u and v as

$$F(x,y) = \sum_{i=1}^{K} s_i P_i \, N_i^{d_1,d_2}(x,y),$$

where P_i are the surface coefficients and N_i are the associated B-splines. K is the number of B-splines. The B-splines are defined on a set of knots in both parameter directions and have polynomial degrees d_1 and d_2 in the first and second parameter direction, respectively. The B-splines have limited support and the extent is given from the polynomial degrees. s_i are scaling factors associated with the B-spline functions in order to ensure partition of unity. The surface is parameterized on the xy-domain of the point cloud used in the surface generation.

In contrast to the least squares approximation which is described in D4.4.1, multilevel B-spline approximation (MBA) is a local approximation method [2]. The surface approximation is, as before, applied adaptively and for each iteration step the surface is refined in areas where a prescribed tolerance is not met and updated to achieve a better approximation of the point cloud. Using the LR-MBA method each surface coefficient is updated with respect to the distance between the data points in its support and the function. Let (x_c, y_c, z_c), $c = 1, \ldots, C$ be the data points in the support of a B-spline. The corresponding new coefficient P_i is determined by

$$P_i = \frac{\sum_c (s_i N_i(x_c, y_c))^2 \varphi_c}{\sum_c (s_i N_i(x_c, y_c))^2},$$

where P_c is computed for each data point as

$$\varphi_c = \frac{s_i N_i(x_c, y_c) z_c}{\sum_i (s_i N_i(x_c, y_c))^2}$$

The sum in the denominator is taken over all B-splines which have (x_c, y_c) in their support.

In the multilevel tensor-product setting where the method was originally developed, the difference between the function and the data values is, at each step, used to compute a difference function approximating the error. The final surface is evaluated by computing the sum of the initial surface and all the difference functions. In the LR B-splines setting, the computed difference function is added to the initial surface at each step giving a unified expression without a global increase in data size.

The method of least squares approximation posses some best approximation properties, see [3]. Applying least squares approximation with a smoothing term reduces the approximation power, but the method still gives good accuracy with respect to the data size. However, in areas with high variation in the data and little regularity in the point distribution this can come at the cost of surface oscillations in areas without data points. On the other hand, the MBA method offers a slightly lower accuracy, but behaves better in low regularity areas. Moreover, as a local method it is not subject to numerical problems at higher iteration steps as is the case for the least squares method. The two methods are used together in the adaptive algorithm, starting with least squares and continuing with MBA after a number of steps. The interplay is studied in [4].

6.1.2.2 The big data challenge

The most important property of LR B-spline surfaces in the context of big data is the ability to reduce the data size drastically and while keeping control over the approximation error.

However, to achieve this gain, it is necessary to be able to handle large data sets. For the current deliverable a number of updates to improve big data handling have been introduced:

- Removed bottlenecks in the implementation
- Reduced the intermediate data storage during execution
- Made operations particularly affected by the number of data points multi-threaded.
  - Evaluate distance between points and surface
  - Evaluating B-splines during the build of the mass matrix for least squares approximation.
  - Computing the contribution to the updated surface while applying the LR-MBA method for approximation

The multi-threading implemented for this service carries over to service #55 (spline surface from raster) and service #57 (update spline surface with point cloud) from D4.4.1.

The largest data set being tested by this service after the listed modifications is the largest data set of 1.1 GB consinsting of 58.58 million points presented in Section 4.3.

This is still not denoted as "big data". Even though a typical survey is smaller than this data set, the service should be able to handle collections of surveys such that the data sizes add up to much higher numbers. The strategy here is tiling and stitching:

1. Divide the data set assembled from several surveys into regular tiles with an overlap
2. Approximate each tile with an LR B-spline surface (this service). Withdraw the surface to remove the overlap and perform trimming if necessary (service #90)
3. Stitch the surfaces together with C^1 continuity to get a seamless result (service #85)

The activities in point 2 can be distributed over several nodes using the power of the IQmulus infrastructure. This approach is ongoing work which will be delivered in D4.4.3. Some preliminary results are shown in Section 5.6.

6.1.3 Example

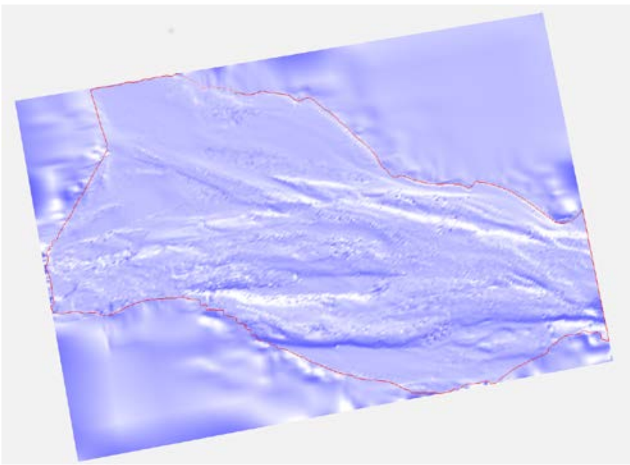


FIGURE 36: SURFACE APPROXIMATING THE HRW 473513 1 POINT CLOUD

Figure 36 shows an LR B-spline surface approximating the HRW 473513 1 point cloud with 58 million points (Table 2, Section 4.3). The surface file is of size 11 M. The size of the las point cloud is 1.1 GB, and an ascii representation corresponding to the surface representation is of size 7.2 GB. The accuracy measures of this approximation can be found in line 3 of Table 9.

The surface has a larger domain than the point cloud. To bound the surface with respect to the cloud, service #90 must be applied. The red curve indicates the extent of the non-trimmed surface.

6.1.4 Quality

In this context quality mainly refers to approximation accuracy and surface quality as measured by surface curvature. The surface should be smooth in areas with few data points. Such areas can be found in the parts of the surface in Figure 30 lying outside the trimming loop. The expected accuracy necessarily depends on the properties of the input point cloud. Outliers should not be approximated with a high degree of accuracy, otherwise the surface quality will suffer. In general, this service is expected to be applied in the context of relatively smooth terrains or when the aim of the surface generation is to extract the essence of the landscape. An estimate would be that given a tolerance reflecting the point cloud quality and the terrain (in the magnitude of 0.3-0.6 meters) and a sufficient number of iterations, 97-99% of the points should be closer to the surface than the tolerance.

The table below summarizes the accuracy results of a number of surface approximations applied to the HRW and REUNION 2009 data sets of size 140 M described in Section 4.3 and 4.5. The number of points is approximately 7.3 millions. For the HRW data set a tolerance of 0.3 m is applied while a tolerance of 0.5 m is used in the REUNION case.

	Data set, no of iterations	Max distance	Average distance	No of points outside tolerance	Average dist for out-of-tolerance points	No of elements
1	HRW 473513 4, 5	5.31 m	0.12 m	548489 (7.5%)	0.44 m	26863
2	HRW 473513 4, 6	5.31 m	0.09 m	242845 (3.3%)	0.42 m	66889
3	HRW 473513 4, 7	5.31 m	0.07 m	97450 (1.3 %)	0.4 m	169056
4	HRW 473513 4, 8	5.31 m	0.06 m	33878 (0.46%)	0.38 m	364606
5	REUNION 2009 3, 5	45.5 m	0.39 m	1521914 (21%)	1.35 m	18144
6	REUNION 2009 3, 6	27.14 m	0.21 m	721990 (10 %)	1.04 m	48375
7	REUNION 2009 3, 7	20.58 m	0.13 m	281528 (3.8 %)	0.87 m	124253
8	REUNION 2009 3, 8	12.97 m	0.09 m	101047 (1.4%)	0.74 m	307900

TABLE 9: ACCURACY VERSUS MAXIMUM NUMBER OF ITERATIONS

The figures are clearly different for the two data sets. The HRW data set is relatively smooth while the REUNION data set has some steep areas with a high level of variation in the points. In such areas the error measurement produces an overestimate of the 3D distance between the surface and the point as discussed in Section 5.3.3. The terrain is also less suited for being represented by a smooth surface. For the HRW data set, the maximum distance between a point and the surface did not decrease. This indicates an outlier. For both data sets, the average error and the average error for out-of-tolerance points decrease steadily.

The approximation accuracy depends heavily on the properties of the point cloud. To remove these factors from the analysis of the approximation, we apply the service to a clean data set. It is constructed from the surface corresponding to line 2 in Table 9. The surface is evaluated in a raster using service #91 and this raster is viewed as a point cloud. The data size is 131 M for ascii xyz format (6.58 million points). This point cloud should in theory be reproducible by an LR B-spline surface with the same spline space. However, the spline space is adaptively defined by an automatic procedure based on the distance field of the points which makes it unlikely to reach exactly the same spline space. In addition, the algorithm is approximate, it finishes when a required accuracy is met. Thus, interpolation cannot be expected, but the accuracy is expected to be good.

Tolerance, no of iterations	Max distance	Average distance	No of points outside tolerance	Average dist for out-of-tolerance points	No of elements
0.1, 6	0.87 m	0.012 m	23769 (0.36%)	0.14 m	65498
0.05, 7	0.19 m	0.0047	2611 (0.04%)	0.07 m	156714

TABLE 10: APPROXIMATION ACCURACY FOR A CLEAN DATA SET

Applied to a point cloud acquired by measurements these results would be very good, concerning this test case they are not quite satisfactorily. The average distance between the point cloud and the surface is well below the tolerance, but the maximum distance is larger than expected. We see that the number of elements does not correspond to the number for the initial surface. An investigation on the development of the spline space throughout the execution will be performed to find out why the maximum distance between the point cloud and the surface remains larger than the tolerance.

6.1.5 Scalability

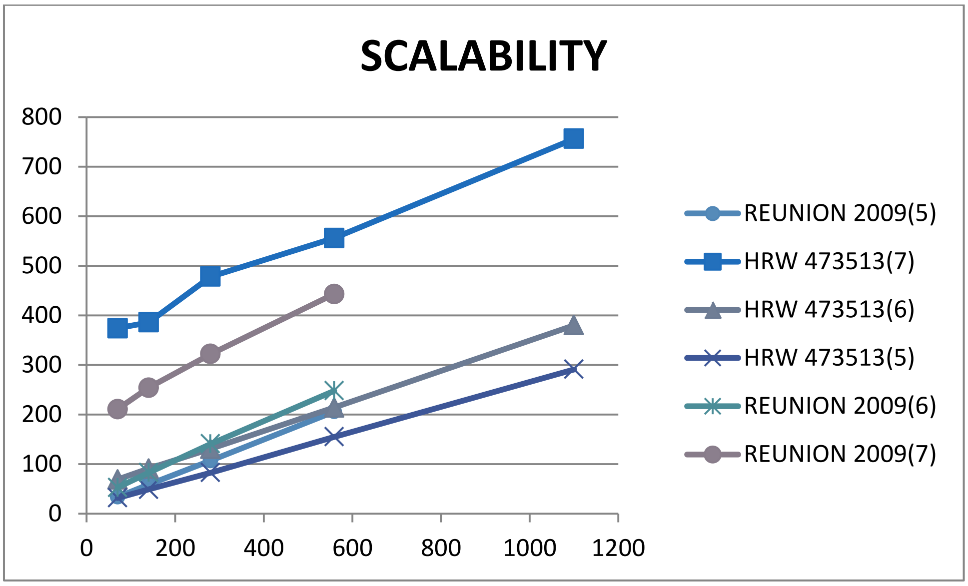


FIGURE 37: SCALABILITY FOR APPROXIMATION OF POINT CLOUD WITH LR B-SPLINE SURFACE

Figure 37 illustrates the scalability of the service. The performance and the acceptable size of the point cloud have improved since D4.4.1. Then the processing time for the Four data sets (~4 million points) was comparable with the current processing time for the HRW 143513 3 point cloud (14.64 million points) The 2009 version of the REUNION data set presented in Section 4.5 (Table 3) and the HRW 143513 data sets from Table 2, Section 4.3, are approximated with varying sizes of the input point cloud and with varying number of iterations. For the REUNION point cloud a tolerance of 0.5 meters is used and for the HRW point cloud, the tolerance is 0.3 meters. The size of the point cloud (las) is the unit of the x-axis and the execution time in seconds the unit of the y-axis. The number of iterations applied in the approximation algorithm is given in parenthesis along with the data set name. The service is single node, but multi-threaded. The computations are performed on a Laptop with 15.6 GB memory and 8 Intel@Core i7-4702HQ CPU cores.

As indicated Figure 37, the scalability is good with respect to data size, but there is a jump in execution time when the number of iterations increases from 6 to 7. This is related to the surface size and the effort required for refining the surface.

Data set/data size	70	140	280	559	1100
REUNION 2009 (5)	18092	18045	18106	17953	
REUNION 2009 (6)	48006	48032	48333	47862	
REUNION 2009 (7)	123401	124253	124582	123756	
HRW 473513 (5)	27070	26821	26452	25368	26569
HRW 473513 (6)	66260	65911	65869	66035	66894
HRW 473513 (7)	163974	166161	168353	170241	172826

TABLE 11: SURFACE SIZE WITH RESPECT TO DATA SET, DATA SIZE AND NUMBER OF ITERATIONS

Table 11 presents the surface size with respect to the size of the point cloud and the number of iterations used in the approximation algorithm. The number in parenthesis following the data set name indicates the number of iterations used when creating the surface. The data set size in megabytes can be found in the upper row in the table. The surface size is measured in polynomial elements in the LR B-spline surface. When this size becomes large, the effort to navigate in the data structure increases drastically. An important reason for introducing tiling for this service is to avoid large surfaces. They are time consuming to generate and to exploit. The table above shows that the surface size is related to the number of iterations and not the size of the input point cloud. Thus, the main parameters for defining tile sizes should be the geographical extent of the data set and the complexity of the shape. However, the data size must be kept within manageable limits.

6.1.6 Degree of human intervention

The service has two mandatory parameters which influence the result, the tolerance and the maximum number of iterations, and several optional ones. The mandatory parameters do not have any default values. The number of iterations is advised to lie in the range [3,7] depending on the wanted accuracy. The tolerance should reflect the smoothness of the point cloud and the purpose of applying this service. However, even for an accurate reproduction, the tolerances should not be set lower than ~2.5 m. The maximum number of iterations has most impact and the effect on the accuracy can be seen in Table 9.

In Table 6 in Section 5.1.5 we see that the average distance is decreased when a tighter tolerance is applied to rain fall data while the maximum distance remains the same. Using a tolerance of 0.5 and 5 iterations in an approximation of the HRW data set used in Section 6.1.4 and comparing with line 1 in Table 9, the maximum distance decreases slightly to 5.19 m while the average distance remain the same. The number of elements has decreased with about 5%. The tolerance influences the refinement strategy, but the approximation at a given step in the iteration will always be as accurate as possible given the spline space. Thus, the sensitivity of the tolerance is low when there are points more distant than the tolerance scattered over large parts of the surface. Using the same tolerance and 7 iterations, the average distance becomes 0.083 m which is a 20% increase when we compare with line 3 in the table. The decrease in the number of elements is about 35%. When the approximation is allowed to be more accurate the size of the tolerance becomes more important for the result.

The service has two optional parameters which influence the performance, the smoothing weight and a flag governing the interplay between the two approximation methods, i.e. the least squares approximation and the multi level B-spline approximation applied to LR B-spline surfaces (LR-MBA). The smoothing weight has a default value of $1.0e^{-10}$ which is almost negligible. The sensitivity of this parameter is low, but increasing it drastically will lead to improved surface quality and lower accuracy in areas with high variations in elevation and unevenly distributed points. A typical scenario is a point cloud where deconfliction has failed.

The interplay between the approximation methods is much more influential. The default setting is to start with least squares approximation and switch to LR-MBA after the first occurrence of 5 iterations and the last iteration step. The algorithm will also switch to LR-MBA if the approximation errors increase using the least squares method. The service is now set us use pure least squares method unless the accuracy decreases and pure LR-MBA. The data set is the one constructed for the investigation reported in the Table 8, and the tolerance is 0.1. The data set has got 6.58 million points. The development of the approximation accuracy during the execution can be found in Table 12.

Iteration	Least squares				LR-MBA			
	Max dist	Average dist	No of pts outside tol	Average dist for out-of-tol pts	Max dist	Average dist	No of pts outside tol	Average dist for out-of-tol pts
Initial	14.66 m	1.22 m	5761597	1.39 m	14.66 m	1.22 m	5761597	1.39 m
1	12.14 m	0.67 m	4786191	0.9 m	11.65 m	0.88 m	5531635	1.03 m
2	7.5 m	0.29 m	3681382	0.5 m	9.98 m	0.49 m	4772347	0.6 m
3	5.0 m	0.12 m	2264768	0.3 m	5.04 m	0.2 m	3339187	0.36 m
4	3.91 m	0.054 m	1203254	0.2 m	4.27 m	0.09 m	1890498	0.24 m
5	1.89 m	0.02 m	251177	0.16 m	2.78 m	0.04 m	717533	0.18 m
6	0.87 m	0.012 m	23679	0.14 m	0.89 m	0.013 m	43702	0.15 m

TABLE 12: APPROXIMATION ACCURACY WITH RESPECT TO NUMBER OF ITERATIONS

The final number of elements was 65498 for the least squares method and 73868 for the LR-MBA method. The results when demanding the pure least squares method are the same as for the default setting.

6.2 #40 RAINFALL USING ORDINARY KRIGING, CNR-IMATI

This service provides approximation of rainfall data, measured at a number of rain gauges sites or from radar reflectivity. The problem is to use in a correct way data at different resolution (very sparse rain data with over the area of interest, different resolution with respect to DEM where the interpolation is needed). The DEM is used to get the points where the field will be approximated. The DEM should have enough details (resolution) to highlight main topographic features.

6.2.1 Motivation for the update

The first release of the service was able to handle only a single input datasets. Often the user has different rain measurements gathered at the same time from different instruments. With the first release, the user had to merge each input dataset before calling the service. However, it is likely that the users have datasets from different sources and keep them distinct. The updated service takes different input datasets into account and it computes the contribution that each dataset provides in the estimation of each interpolated point; the process is shown in Figure 38.

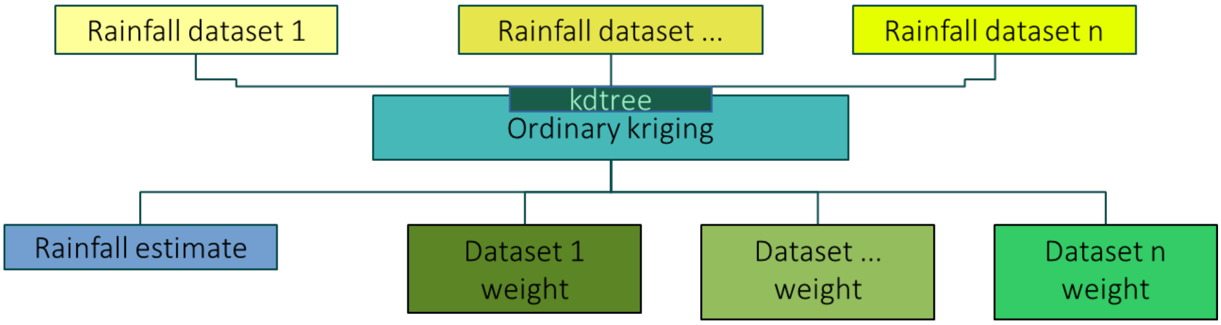


FIGURE 38: THE UPDATED SERVICE WILL READ INPUT FROM DIFFERENT DATASETS AND WILL COMPUTE THE WEIGHT THAT EACH DATASET GETS IN THE ESTIMATION PROCESS

6.2.2 Modifications

Recalling the explanation of Ordinary kriging (OK) exposed in D4.4.1 we see that the OK is a point estimator algorithm in the "best linear unbiased estimator" family. The OK estimate is a linear combination of the available data. The estimation of the variable is expressed as:

$$p_0 = \sum_{j=1}^{n} \omega_j \times p_j$$

Where p_0 is the point at which the surface has to be estimated, p_j are the known samples and ω_j are the weights for all the known points.

During the OK process the service will store for each input data information about the source dataset. To compute the contribution of each dataset (*l*), W_l, we need to compute the following ratio for each dataset:

$$W_l = \frac{\sum_{j \in M_l} |\omega_j|}{\sum_{j=0}^{n} |\omega_j|}$$

Where ω_j is the OK weight, *n* is the number of samples, *l* is the index of dataset and M_l is the set of data belonging to dataset *l*. For example, if we guess to have *n* samples from only two different datasets (*l=0; l=1*), the first datasets have sample p_j and weight ω_j with *j* from *0* to *m* and the second one with *j* from *m* to *n*. The weight W_l will express the contribution of *l* dataset to the estimated value. The approach is a modification of ordinary kriging. Figure 39 shows the estimation result, respectively on the left the results computed with only one input dataset (only rain gauges) and on the right the one computed with two different dataset (radar and rain gauges). The different datasets are measurements of the same variable with different instruments. Figure 40 shows the contribution of each dataset in the estimated map plotted in 39-b. The colour scale varies from blue to red corresponding to a contribution varying from null to full, respectively. In the considered example we used only two different dataset and for that reason the two maps in Figure 40 are complementary.

.

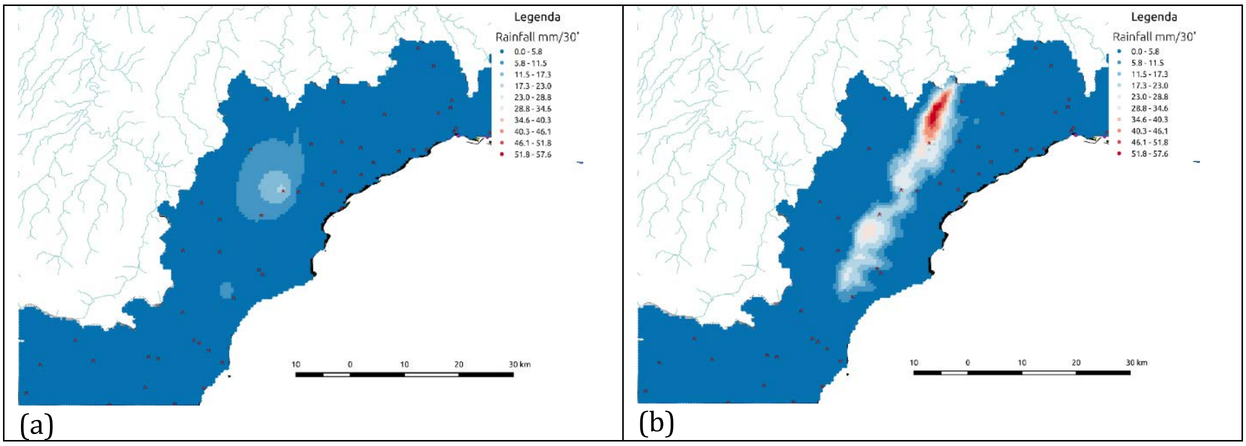


FIGURE 39: THE MAP ON THE LEFT HAS BEEN COMPUTED USING THE OLD VERSION OF THE SERVICE WITH AS INPUT ONLY THE RAINGAUGES WHEREAS THE MAP ON THE RIGHT HAS BEEN ESTIMATED CONSIDERING BOTH RAIN GAUGES AND RADAR MEASUREMENTS.

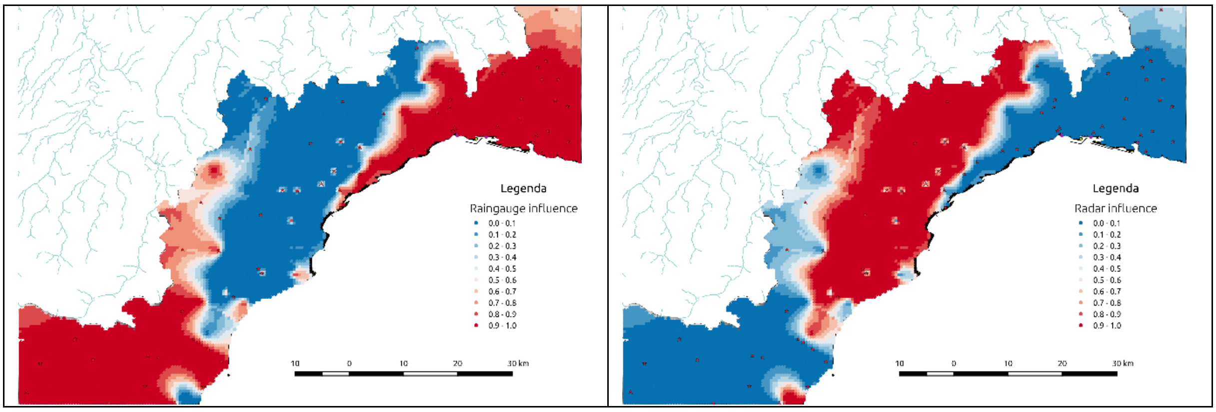


FIGURE 40: THE LEFT MAP SHOWS THE WEIGHT OF THE RAIN GAUGES DATASET WHERERAS THE RIGHT MAP SHOWS THE WEIGHT OF THE RADAR DATASET MAPPED IN 39 (B).

6.2.3 Quality

The algorithm update does not modify the quality performance of the service. As we already have seen in D4.4.1 the service has been tested with different techniques in order to discuss the following items.

- *Evaluation of the accuracy of each approximation scheme* through the comparison between approximated and true values at every sample point.
- *Cross-validation of the method.* Every 30 minutes, each rain gauge has been turned off and the approximation function has been sampled at this position. Then, the obtained value has been compared with the true value.
- *Evaluation of the accuracy of the algorithms through the comparison with the ground-truth*. For the validation of the approximation techniques, the rainfall data measured by the Genova municipality has been used as ground truth to validate the values interpolated from the regional data set. The two observation networks cover an overlapped region of the study area. The network from Genova municipality is located within the boundary of the city and is denser than the one from Regione Liguria that

covers the whole study area. Some stations of the regional networks are located in the Genova area and allow us to interpolate the rainfall field over the city within a certain accuracy, which has been further estimated comparing the approximation results at these two scales.

The following table outline the result of quality testing:

Technique	Max [mm]	Mean [mm]	Std. dev. [mm]	MSE [mm²]
Evaluation of the accuracy	3.3E-07 (5E-6%)	4.3E-11	3.8E-9	1.44E-17
Cross-validation	32.44 (54.1%)	0.02	2.38	5.64
Comparison with the ground-truth	28.62 (47.7%)	0.59	4.45	20.21

TABLE 13: TESTING WITH RESPECT TO THE ASSESMENT CRITERIA

6.2.3 Scalability

The scalability of service has been tested on an eight cores desktop computer with 16 GB of RAM running Ubuntu 14.04. To test scalability we used the same rainfall record for all test and varied the size of the point cloud where the service will estimate values. The size range of the tested point cloud varies from 20 thousand points (about 1.4 MB) up to 210 millions points (about 17 GB). The results of the tests are plotted in Figure 41. The x-axis indicates the number of evaluated points and the y-axis shows the execution time expressed in seconds.

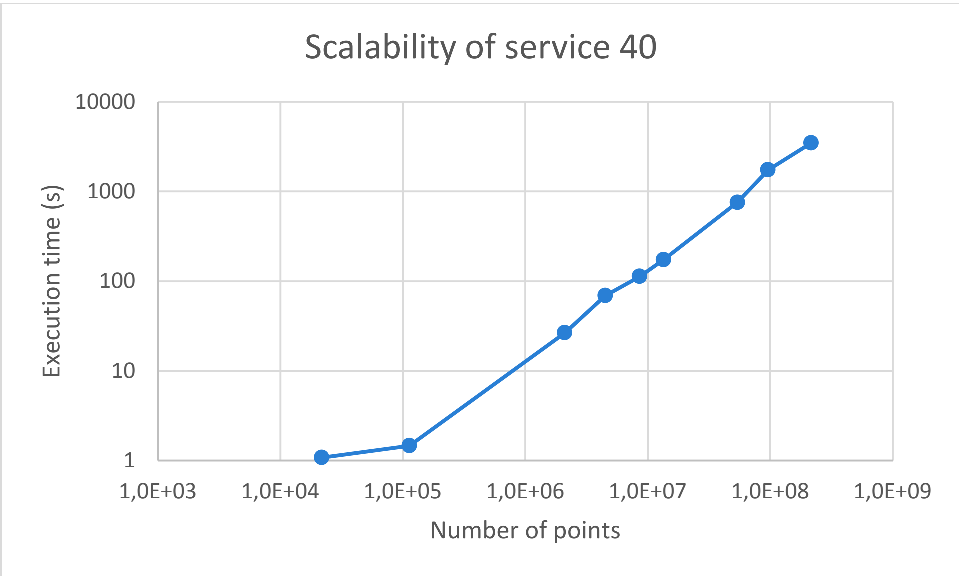


FIGURE 41: TIME NEEDED FOR EXECUTION OF THE SERVICE (Y AXIS) RESPECT DIFFERENT NUMBER OF INPUT POINTS (X-AXIS)

The service code is multi threaded and may be executed multi node with the help of script. The point cloud representing the positions where the rain field will be estimated, is tiled in chunks. Each thread elaborates one chunk. At the end the chunks will be stitched together.

6.2.4 Degree of human intervention

The updated service have tuning parameter optimized for this kind of datasets already hardcoded and do not need human intervention.

7 SERVICES UNDER DEVELOPMENT, CNR-IMATI

This section contains a description of 3 related services which is expected to be delivered within 2 months starting from PM30.

7.1 THE MULTIRESOLUTION FOR HYDROLOGY FRAMEWORK

The MultiResolution for Hydrology framework is part of the Land Showcase and it is designed to demonstrate how intelligent processing of geometric data is a solution to handle big data.

Big data, such as LIDAR point clouds, are typically stored as collections of files, each containing either tiles of the whole dataset or raw measurements (e.g., the LIDAR strips acquired during a flight). The indexing of the tiles in the collection is usually defined according to *standard* conventions, typically by indexing the tiles according to the min/max values of the data bounding box, in particular in the case of regular tiles organized with a grid approach. For LIDAR raw strips acquisition, the collection is often organized with respect to time and date of acquisition, without any specific spatial indexing. Moreover, within each tile, all data are exactly at the same level of representation: it is not possible to distinguish data/samples along feature lines; it is not possible to understand what points contribute to reaching a given resolution, and so on. Obviously, it is possible to have an insight into the information contained in the data after an actual analysis of the data.

We argue that analyzing data beforehand is an important way to approach big data. Insight into the data can indeed suggest novel modalities to organize and represent geospatial information, which make it possible to handle big data. For what this service is concerned, the idea is worked out for big point clouds, stored as collections of files, for which building a single high-resolution triangulation is in practice impossible due to the complexity of the resulting model. Indeed, one of the main bottlenecks is the visualization, which is limited by the capabilities of the graphic hardware available on the market. The size and the complexity may also be a problem for processing, in particular in the case of global algorithms that are tricky to be parallelized. In the case of LIDAR data, we are considering billion of point to be managed and processed: currently there are algorithms to build and manage triangulations of millions of points (see for examples [5]), but for higher number of points, it is difficult to build, store and then process and visualize a triangulation at full resolution.

A multi-resolution model, that is a triangulation that loads from the available data only those points that are necessary to achieve a prescribed accuracy, is the most appropriate solution to address this issue. The techniques at the state-of-the-art address this problem by considering that the approximation accuracy, or level of detail, is uniform over the whole domain of interest. Resolutions varying locally are used typically for rendering purposes, in order to have the highest resolution closer to the point of view of the user. Here, we would like to generalize the concept of multi-resolution to handle and varying the resolution in localized areas, defined both by the semantics of the usage context of the model and by the user interest.

The first step is the organization and prioritization of the data within the original files according to semantic information. After this organization/indexing, data can be easily retrieved to produce a triangulation with variable resolution in different regions of interest, as specified by the user interest and by the semantics, namely in this case by the hierarchy of the basin induced by the river network[2]. The framework is implemented in the domain of hydrologic studies but the validity is more general and can be applied whenever a hierarchy on the data can be derived from any other conceptual level/knowledge.

The system is composed of two main modules:

- the BasinTree library, to construct and query the index of the data that reflect the hierarchy of basins;
- the LasSort module, to reorganize the point cloud associated to the basin map.

7.1.1 Representing the drainage basins hierarchy

The aim of the BasinTree library is to convert a drainage basins map, which codes the hierarchy of interest in the domain of application, into a tree structure representing the containment relations defined by the drainage basin hierarchy. The drainage map encodes indeed a partition of the interested region, defined as a set of polygons contained into each other. We name the partition a "vector-layer partition" meaning that the hierarchy is encoded by a vector-layer, that is, the polygons defining the boundaries of the partition.

The assumption on the input is that we have, alongside with the basin map that subdivides the land map with all the top-basins, a representation of all the rivers associated with the basin map (river network). We need the rivers to retrieve automatically the hierarchy of basins.

7.1.2 The BasinTree structure.

The BasinTree is the way we use to encode and represent the hierarchy of the watershed, which correspond to the hierarchy of the rivers associated to the drainage map.

Moreover, we designed the BasinTree to retrieve and manage the containment and adjacency relations among basin at different scales, which will be crucial to select the area of interest during the query/extraction phase.

In detail, a BasinTree is represented as a generic tree, with a possibly variable number of children for every internal node. For all the nodes, we store a vector of points representing the boundary of the region

In the leaf nodes, we also store additional information to encode the Region-Region (RR) relation and a link to the metadata file associated to the basin, which contains the link to the LAS files with the points that are inside the basin. The index of the LAS files comes from the LasSort module described below, and it is the analogous of the metadata file generated by Service 94, the Vector Layer Partitioning of a Point Cloud, described in D4.3.2.

7.1.3 The Region-Region (RR) relation.

To encode the RR relation we decompose the boundary of the region in polylines, and we use these parts to encode the relation as in a boundary representation.

[2] http://en.wikipedia.org/wiki/Strahler_number

The subdivision of the boundary is given by the KeyPoints, i.e. points on the boundary where a change of the adjacent region occurs: in the example in Figure 42 the KeyPoints of δ are [C, D, E], where the regions adjacent to δ change.

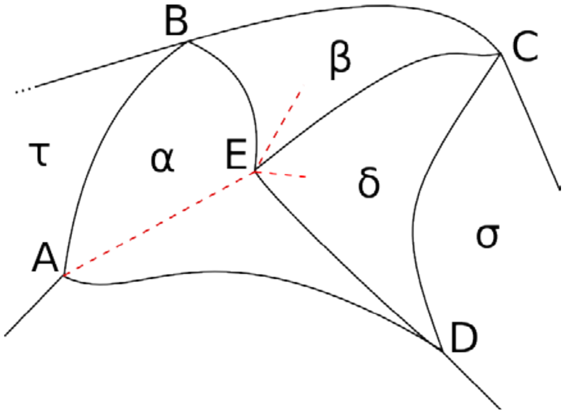


FIGURE42

We encode the KeyPoints with an array of indices referring to the array of points representing the whole boundary; this array is stored in clockwise order, according with the convention on the whole boundary induced by the ESRI Shapefile format definition[3].

The adjacent regions are encoded in an array with the same size of the KeyPoints one, according to the following rule: given an array of keyPoints $V=[v_1..v_n]$, and a n adjacency array $RR=[r_1..r_n]$, the r_i region is the region adjacent through the polyline defined by point $[v_i, v_{i+1}]$. Notice that the KeyPoints vector in interpreted as a circular array.

In the example of Figure 42, we have:

V(α) : [A, B, E, D] RR(α) :[τ, β, δ, -]
V(β) : [B, C, E] RR(β):[-, δ, α]
V(δ) : [C, D, E] RR(δ) : [σ, α, β].

The RR relation is explicitly stored only for the leaf nodes: starting from the information on the leaves, it is possible to retrieve the adjacency relation for the internal nodes.

7.1.4 Generation of the tree structure

The generation of the tree structure is straightforward under the assumption we made on the input: each river as a unique reference to a basin, such as the BasinID or the name of the basin. Following the river starting from the outlet, at every fork we insert the new basin subdivision in the tree, until all sources are reached.

3 *https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf*

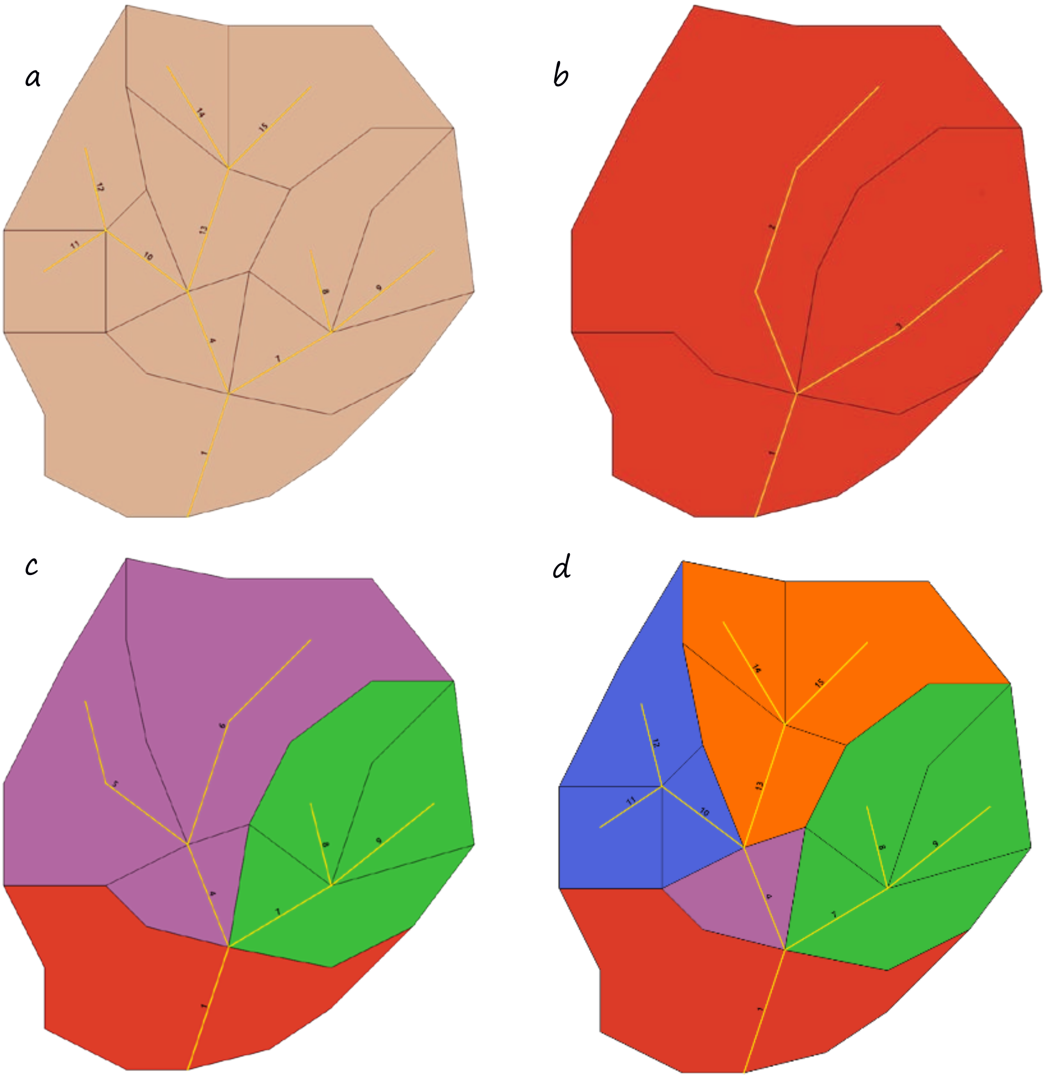


FIGURE 43 BUILDING THE BASINTREE: AN EXAMPLE

In Figure 43, we show an example of the process performed to build the BasinTree starting from the input, represented in figure 43a: notice that, for visualization purpose, we show only the top-

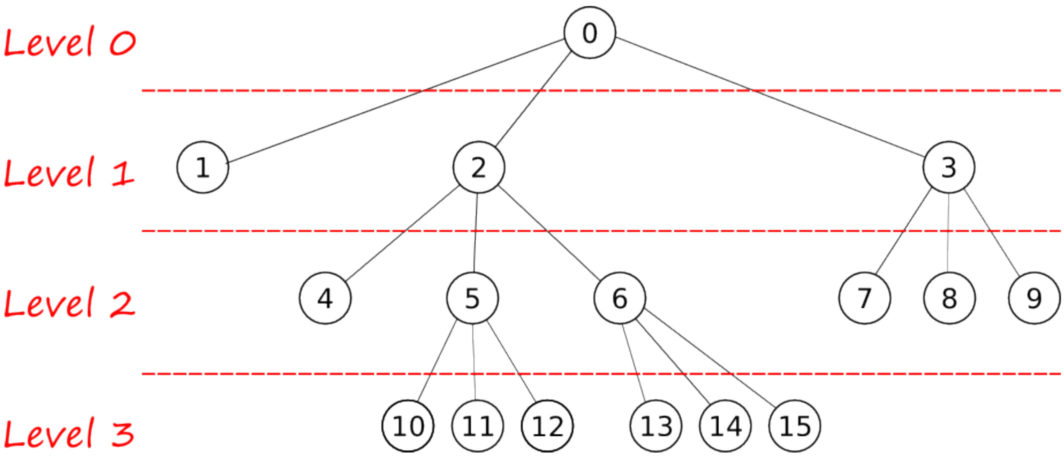


FIGURE 44: THE TREE STRUCTURE RELATE TO FIGURE 2

level basin, i.e. the decomposition at its deepest level. In the real input we also have the intermediate regions, such as 2, 3, 5 and 6. In Figure 43b, 43c and 43d we show the iterative subdivision of the basins induced by the forks of the rivers. The resulting tree is shown in Figure 44.
In the showcase context, we repeat the algorithm for all the top basins, i.e. the ones that are not part of other basin because they have the outlet of the associated river.

7.1.5 Encoding the tree

Since from the processing of the whole basin map we do not obtain a single tree but a forest of trees, whose roots are the top-level basins, we insert a virtual super-node representing the union of all the regions: in our reference showcase, the root will be the complete LIGURIA region.
It is also possible to insert other "virtual" nodes to group basins at the top level in wider area: for example in the case of Liguria, it may be useful for experienced end-users to have a level associated with the "Ambiti di Bacino" maps, as shown in Figure 45.

FIGURE 45 "AMBITI DI BACINO" MAP

Since the result of this processing step is a forest of generic trees enriched with the adjacency relation, and there is no standard format for this kind of data, we store the BasinTree in a JSON file, writing all the nodes of the tree in the order induced by a depth-first traversal of the tree. We store different information for internal nodes and leaves, since the union of all the leaves represents the complete subdivision of the space at the maximum scale level.

We encode every node of the tree as a JSON Object that contains the information needed to rebuild the tree structure with all the different attributes of the basins. A detailed description can be found in the JSON-schema in Appendix A.

In the following table, we describe the attributes and some constraints that cannot be inferred from the schema.

Name	Type	Description
ID	Integer	Unique identifier of the basin, used also to identify the node in the tree
parent	Integer	Unique identifier of the parent of the node, to encode the tree structure. If the parent is the virtual root, this value is -1
type	String	to indicate if the node is internal or if it is a leaf
shp_uri	String	Link to the ESRI Shapefile with the boundary of the region
json_las_uri	String	Link to the Json metadata file (ID.json) with the link to the LAS files with points inside the region. Valid only if type is "leaf".
asdj_basins	Array of integer	The unique ID of the adjacent regions at the maximum scale. Valid only if type is "leaf".
key_points	Array of integer	Indexes in the set of boundary points identifying the key points. Valid only if type is "leaf".

TABLE 14: EXPLANATIONS TO THE JSON SCHEMA IN APPENDIX A

7.1.6 The LasSort module

The aim is to have a well organized ordering/indexing of the original LAS collection that allows to access directly only those points necessary to build a triangulation of the terrain which respects both the requirements imposed by users selections (e.g., scale of basins to be analyzed and focus of interest) and by available hardware and/or cloud configurations. The user may want to analyze, for example, only one basin in detail, or all the basins at a given scale (e.g., 2 km^2 extension). Given the parameters selected by the users, the triangle model has to be constructed. For this, we do not plan to use all available high-resolution data, but to access only the data points that are needed to (i) provide a triangulation in the area of interest of the user with the required accuracy; (ii) in the areas outside the focus of interest, provide a triangulation with a lower accuracy that takes into account the context (hardware support/configuration) in which the triangle model has to be produced (e.g., available RAM, available nodes).

The idea is therefore to organize the hierarchy of basins and the associated original LAS files so that, for a given set of user parameters and hardware settings, it is easy to access directly only to the points needed to fulfill the request. If the selected point sets is manageable, as a single triangle mesh, then the extraction process will work on a single data structure/file, otherwise it will adopt a different strategy, triangulating the terrain in a basin-wise manner and patching the results for analysis purposes.

The LasSort module can be considered as an evolution of the Vector Layer Partitioning of a Point Cloud (VLPPC) service (IQmulus Service 94). But, while the VLPPC service provides an information about what LAS file contains data belonging to a given basin, the "well ordered LAS file" is aimed at performing further indexing (and re-ordering) of the original LAS file according to rules that influence the resolution of the triangle mesh to be constructed at the end of the pipeline.

Each LAS file can potentially contain points from different basins, identified by metadata in the tree structure: we group points from the same region in adjacent records of the LAS file (see the description of the VLPPC service in D4.3.2).

Within each group of points belonging to the same basin, we can prioritize the points as:

1. Points in the area of influence of a boundary segment
2. Points of a feature we want to preserve
3. Points in the area of influence of a feature
4. Generic points

The key idea here is to sort the all the point in (inverse) “importance” order: in this way, in the extraction phase, according to the resolution request, we do have to take into account only the number of points to be selected. To build this “well ordered” LAS (i.e. to sort the points), we have to run a simplification algorithm on the point cloud to iteratively remove points from the original highest resolution. The order of deletion of points is reversed, i.e. generic points are removed first. Points that survive at the end of the simplification are the most significant and marked as low-resolution points: this means they will be always displayed, even when low resolution rendering is required. It is important to notice that, for our scope, we do not want to insert new points.

Since the first points deleted are the least important, points will be arranged in reverse dilation order in the LAS blocks. Therefore, they will be arranged from the most significant to the lowest, in order to optimize the triangulation construction

We choose to encode this information in JSON format, which is simple to understand and easy to manipulate: the format is under definition, but is clearly related to the JSON format defined for the VLPPC service (updated version with respect to the description of D4.3.2).

7.1.6.1 The Extraction Phase

To extract a variable resolution terrain from this framework, we use the tree of the drainage hierarchy as "spatial index" to identify the area of interest, supporting geometric selection (for example a rectangular or a circular area) or direct selection through basins, macro-area (i.e. Ambiti di Bacino) labels or scale request.

This step is performed with the C++ library implemented for BasinTree building and handling.

The service has two different ways of selecting the region of interest:

- via a list of Basins
- via a bounding box

To fulfil the first query type, we search the tree for the desired basins and gather the information on adjacent regions and total number of points. In this case, we also check if the selected regions are adjacent and, eventually, we select the missing patches to have a proper coverage.

In this first case, we stop when we have completed the search of the basins in the desired list at the level where the basins resides in the tree.

The second type of query is performed by visiting the tree and checking the nodes against the bounding box, to retrieve a list of basins inside the selected bounding box. In this second case, we try to go deep in the tree to select the higher resolution possible.

Another parameter is the measure of the "resolution": we use a value from 0.0 to 1.0 to indicate the percentage of points to be retrieved.

This value, combined with the information retrieved from the tree, allows us to identify the actual number of points to be triangulated.

Once the process of selecting the desired region is done, we have a list of basins and a number of points to be triangulated.

Her we have two possibilities:

1. the number of points is manageable in-core from service 49: we (will) know an estimate of the higher number of points feasible from the scalability test;
2. the number of point is clearly impossible for the in-core algorithm.
   In this case, we can triangulate the basins separately (according to the list of selected basins) and write down the output in the Partitioned Triangle Mesh format which is under definition: a preliminary technical description is in Appendix B

7.2 MAPPING THE MODULES OF THE FRAMEWORK TO IQMULUS SERVICES

48	Multi-resolution triangulation for land monitoring	This is actually the initialization of the system. Input: • A drainage basin map, with the associated rivers • A set of Point Clouds covering the area. Output: • An encoding of the BasinTree structure for basin hierarchy • A set of sorted LAS files, indexed from the leaves of the BasinTree.
107	Extract triangle mesh for visualization	Extract a triangle mesh from the Multi-resolution framework, by properly select the points according to user request and the by using service 49 to generate the triangle mesh. Input: • A set of basin in which the user is interested and a resolution request. Output: • A single triangle mesh if feasible of a Partitioned Triangle Mesh otherwise.
108	Multi-resolution to raster	Convert the triangle mesh extracted by service 107 to a raster (GEOTiff)

TABLE 15: THE MULTI RESOLUTION TRIANGULATION SERVICES

7.3 CURRENT STATUS AND DEVELOPMENT PLAN

Unfortunately, although the Multiresolution framework was supposed to be delivered at PM30, we are experiencing a delay, mostly for some underestimation during the design and the development phase of (i) the work load; (ii) technical problems.

The framework is currently under development; we implemented a beta version of the BasinTree structure and we are testing the library on real data.

For the LasSort module, we are implementing and testing different approaches to get a proper order, also considering an adaptive approach to minimize the approximation error for the underlying terrain surface. In this direction, we are currently implementing and testing some method described in literature [6,7,8].

We plan to have a functional beta version of the complete system within **2 months** starting from PM30.

8 CONCLUSIONS

Most of the services delivered in D4.4.1 and D4.4.2 are single threaded and single node. Some services are multi threaded. The main strategy for handling large data sets is to tile the data and let the job manager manage distribution of the single node services in the cluster. There is ongoing work in preparation for tiling and stitching for many services, but the results are only to a limited extent available in this deliverable.

The multi-resolution triangulation services related to the Land Scenario is delayed. The services are still under development and will be released in short time.

The toolkit pursues three very different surface generation strategies which are tied to the three scenarios. The T4.4 functionalities required in the Land Scenario are previously delivered or ongoing work. With regard to the Marine Scenario, deconfliction is not addressed, but services like #9, #88 and #57 (update of an LR B-spline surface with regard to a given point cloud) are important pre requisites. No services related to MS3 are delivered. The T4.4 services connected to the Urban Scenario are delivered in D4.4.2, but more effort is required to reach their potential with regard to big data.

The T4.4 toolkit is being populated with a broad range of services that fit into the specified work flows with services from other toolkits. However, an increased attention with regard to big data is required.

9 OUTLOOK ON VERSION 3

The main emphasis for T4.4 towards the next deliverable is big data. Several services have to be updated to fulfil the data size requirements of IQmulus. This applies in particular to the services being selected for demonstrating big data capabilities, see Section 2.2, but also other services may be influenced. Depending on the characteristic for the different services, the following approaches are relevant:

- Tiling and stitching. Combining this approach with the infrastructure gives multi-node capacity both with regard to computational power and memory use.
- Multi-threading. Some services, for instance #86 and #88, are easily ported to a multi-threaded version. This approach can be combined with tiling and stitching.
- Utilize the expected data distribution of input points for streaming computation of Delaunay triangulations.
- Hadoop MapReduce. The method is most appropriate for non-global computations. We will further analyze our services to see where it can make a difference.

The current deliverable addresses the defined work flow well, but some functionality is still missing. The multi-resolution triangulation services are being delayed and we will return to these services in D4.4.3. In Marine Scenario deconfliction is a major issue and this has not yet been addressed in a systematic way. Deconfliction will be in focus for the next period. The strategy will be:

- Make a low resolution collection LR B-spline surfaces approximating the area of interest. This functionality coincides with the tiling and stitching approach for LR B-spline approximations (service #9).
- Compare the surface set to the initial data to identify conflict areas. This is an extension to service #88.

- Prioritize the initial surveys according to date and quality to decide which initial data should be prioritized in the conflict areas. Acquisition method may also be relevant in this context.
- Update the surface set to the requested accuracy with respect to the selected initial data. This functionality will build upon service #57 from D4.4.1.

The IQmulus show cases will be updated within a short time frame. The result of that work may influence the plans for T4.4. The realization of MS3 and feature detection based on LR B-spline approximation will be dependent on the results of the show case update.

During this project, it has become steadily clearer that spline surfaces and triangulated surfaces have different pros and cons and they complement each other very well. We will look into the potential for combining the two surface representations in the IQmulus work flows.

10 REFERENCES

1. T. Dokken, K.F. Pettersen, T. Lyche. Polynomial splines over locally refined box-partitions. Computer Aided Geometric Design 2013; 30(3): 331-356; http://dx.doi.org/10.1016/j.cagd.2012.12.005
2. S. Lee, G. Wolberg and S.Y. Shin. Scattered data interpolation with multilevel B-splines. IEEE Transaction on Visualization and Computer Graphics, 3(3): 229-244, 1997
3. M.J.D. Powell. Approximation theory and methods. Cambridge Univercity Press (1981)
4. V. Skytt, O. Barrowclough and T. Dokken. Locally refined spline surfaces for representation of terrain data. Comput Graph (2015), http://dx.doi.org/10.1016/j.cag.2015.03.006i
5. Isenburg, M.; Liu, Y.; Shewchuk, J. R. & Snoeyink, J. (2006), 'Streaming computation of Delaunay triangulations.', ACM Trans. Graph. 25 (3) , 1049-1056 . Web.
6. Pauly, Mark, Markus Gross, and Leif P. Kobbelt. "Efficient simplification of point-sampled surfaces." Proceedings of the conference on Visualization'02. IEEE Computer Society, 2002.
7. Alexa, Marc et al. “Point Set Surfaces.” Proceedings of the Conference on Visualization ’01. Washington, DC, USA: IEEE Computer Society, 2001. 21–28. Web. VIS ’01.
8. Linsen, Lars. Point Cloud Representation. Univ., Fak. für Informatik, Bibliothek, 2001. Print.

11 APPENDICES

APPENDIX A – BASINTREE JSON SCHEMA

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "items": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "node_type": {
            "type": "string",
            "enum": ["internal"]
          },
          "id": {
            "type": "number"
          },
          "parent": {
            "type": "number"
          },
          "shp_uri": {
            "type": "string"
          }
        },
        "required": [
          "node_type",
          "id",
          "parent",
          "shp_uri"
        ]
      },
      {
        "properties": {
          "node_type": {
            "type": "string",
            "enum": ["leaf"]
          },
          "id": {
            "type": "number"
          },
          "parent": {
            "type": "number"
          },
          "shp_uri": {
            "type": "string"
          },
          "json_las_uri": {
            "type": "string"
          },
          "adj_basins": {
            "type": "array",
            "items": {
              "type": "integer"
            }
          },
          "key_points": {
            "type": "array",
            "items": {
              "type": "integer"
```

```
          }
        }
      },
      "required": ["node_type", "id", "parent", "shp_uri",
                   "json_las_uri", "adj_basins", "key_points"]
    }
  ]
 }
}
```

APPENDIX B- PARTITIONED TRIANGLE MESH FORTMAT (DRAFT)

This format is designed to store a set of simplicial meshes (2D and 3D) representing a big simplicial complex tiled according some spatial or semantic rule. For sake of simplicity we refer to the 2D case

Given a set of N triangle mesh, for each patch we store it in indexed format decoupling the geometry from the connectivity. So, we encode points and simplexes in two different binary files : XX.points for the geometry and XX.simplex for the connectivity.

We have also to encode the connectivity graph of the adjacency relation of the patches, and for every adjacent border portion, we have to store a "points matching" file.

Since we have a "points matching" file for every arc of the connectivity graph, we encode a textual "correspondence file" of the boundary points that matches.

File XX_YY.lookup represent the arc from XX and YY (and reverse).

For every corresponding point, we store a couple of indexes, the first referring to the XX.points file and the second to the YY.points file.

```
.point file format
HEADER
LIST_OF_POINTS

Header spec:
  #magic_number                [string format identifier] [to be specified]
  #number_of_points_record     [long 4 bytes]
  #X_scale_factor              [double 4 bytes]
  #Y_scale_factor              [double 4 bytes]
  #Z_scale_factor              [double 4 bytes]
  #X_offset_factor             [double 4 bytes]
  #Y_offset_factor             [double 4 bytes]
  #Z_offset_factor             [double 4 bytes]
Point record spec:
  X                            [long 4 bytes]
  Y                            [long 4 bytes]
  Z                            [long 4 bytes]
```

To retrieve the actual coordinates, the apply the following equation:

X_coordinate = (X * X_scale) + X_offset

Y_coordinate = (Y * Y_scale) + Y_offset

Z_coordinate = (Z * Z_scale) + Z_offset

```
.simplex file format

#number of simplexes           [long , 4 bytes]

#dimension_of_simplex          [char, 1 byte]

#list_of simplexes
```

Every simplex is encoded with the indexes of its vertices in the .points file.

```
Simplex Record
      V1          [long, 4 bytes]
      ……
      Vn          [long, 4 bytes]
```

where *n* is the dimension of the simplexes

APPENDIX C– SERVICE INFORMATION TABLES

This appendix contains service metadata table for the new services in D4.4.2 and the services from D4.4.1 which have been subject to a major update.

<table>
<tr><th colspan="4">IQmulus Service #09</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">ParamPoints2Spline
Generate spline surfaces from parameterized 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">Compute one B-spline or LR B-spline surface by approximating a point cloud with parameter information. The parameterization can be given by the xy-data of an elevation data set or be computed using service #56, possibly in combination with service #84.</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>Point cloud (.las, .xyz or .g2)</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>• Approximation tolerance
• Number of iterations in adaptive procedure</td></tr>
<tr><td>Output: <data representation and format></td><td>• LR B-spline surface in g2-format
• Accuracy information and possible distance field (.txt or .ply)</td></tr>
<tr><td>Functionality of the service: <text></td><td>Surface approximation</td></tr>
<tr><td>Algorithm</td><td colspan="2">An adaptive procedure is applied. At each step the current surface is refined in areas where a prescribed tolerance is not met and an updated approximating surface is computed. Two different approximation methods are used depending on the stage of the computation and optional input parameters: least squares approximation and multi-level B-spline approximation applied on LR B-spline surfaces (LR-MBA)</td><td>The same functionality may in principle be implemented by different algorithms.</td></tr>
<tr><td rowspan="4">Implementation details</td><td>Implementation language</td><td>C++</td><td rowspan="4">Include information related to the implementation of the service, such as language (e.g., C, C++, etc); dependencies with other libraries (e.g., ANN library); constraints on the operating systems, and visualization modalities of the output.</td></tr>
<tr><td>Dependencies with other libraries</td><td>liblas, rply</td></tr>
<tr><td>Operating system</td><td>Linux</td></tr>
<tr><td>Visualization modalities of the output</td><td>Visualization of LR B-spline surface</td></tr>
<tr><td>IQmulus Data</td><td colspan="2">Available IQmulus input data:
Original version of point clouds from #12 and #13 (Four), #23 (Liguria), REUNION data sets from UBO, various HRW data sets</td><td>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</td></tr>
</table>

		the service]
Service characteristics	**Accuracy:** Depends on the characteristics of the input points and the number of iterations in the adaptive procedure. In most cases 98-99% of the points will lie closer to the surface than 0.5m. **Robustness:** Small perturbations in the point set will give small changes in the surface. **Computational time in relation to data size:** Provided that the complexity of the underlying terrain, the maximum number of iterations and the tolerance is kept fixed, the computations time scales linearly with respect to the size of the point cloud. 14 million points, 280M las-file takes 3 minutes (Intel processor, 4 kernels, 11.7GiB memory) **Locality/globality of the algorithm:** Global, but some operations are local Data partitioning will be necessary to address the scalability of the service for large point clouds and geographical domains.	*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.*
Alternatives	<List of Service IDs>	*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	Marine 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	SINTEF Vibeke Skytt, Vibeke.Skytt@sintef.no	*Partner ID and responsible person (include email)*
Involved Partners		

IQmulus Service #58			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	Spline rainfall		*Unique identifier of the service; necessary to call it within user-defined workflows*
Description	Approximate a number of time steps of rainfall data by LR B-spline surfaces		*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*
Service functionality	**Input: <data representation and format>**	Matrix with rainfall data (.csv or .mat)	*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*
	Input parameters: *[optional]*	Approximation tolerance Maximum number of iterations	
	Output: <data representation and	Approximating LR B-spline surfaces (.g2)	

	format>	Associated information (.txt)	*this field.*
	Functionality of the service: <text>	Approximation of rainfall data	
Algorithm	An adaptive procedure is applied. At each step the current surface is refined in areas where a prescribed tolerance is not met and an updated approximating surface is computed using multi-level B-spline approximation applied on LR B-spline surfaces (LR-MBA)		*The same functionality may in principle be implemented by different algorithms.*
Implementation details	**Implementation language**	C++	*Include information related to the implementation of the service, such as language (e.g., C, C++, etc); dependencies with other libraries (e.g., ANN library); constraints on the operating systems, and visualization modalities of the output.*
	Dependencies with other libraries		
	Operating system	Linux	
	Visualization modalities of the output	Visualization of LB B-spline surface	
IQmulus Data	**Available IQmulus input data**: Liguria observed rainfall data		*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:** Depends on the number of iterations. Expecting an accuracy of about 0.05m for 20 iterations		*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: Small perturbation in the input data will give small differences in the surface.		
	Computational time in relation to data size: The data set for each time step is small. The computation time scales approximately linearly with the number of time steps		
	Locality/globality of the algorithm: Global for each time step		
Alternatives	<List of Service IDs>		*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	SINTEF Vibeke Skytt, Vibeke.Skytt@sintef.no		*Partner ID and responsible person (include email)*
Involved Partners	IMATI for information, contest on rainfall data and preparation of data		

IQmulus Service #85

Name of the metadata	Content expected		Motivation/comments
Service Acronym	StitchSurfaceSet		*Unique identifier of the service; necessary to call it within user-defined workflows*
Description	Ensure C^1 continuity between LR B-spline surfaces ordered in a regular grid (corner-to-corner configuration). The surfaces may be trimmed and some surfaces may be missing.		*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*
Service functionality	**Input: <data representation and format>**	Metafile providing path and name for the surface file (.json). File root for modified surface files	*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.*
	Input parameters: *[optional]*	Optional parameter enabling computation of continuity before and after applying this service	
	Output: <data representation and format>	Metafile providing path and name for modified surfaces (.json) Modified surfaces (.g2)	
	Functionality of the service: <text>	Stitching of LR B-spline surfaces	
Algorithm	Insert new knots to ensure corresponding spline spaces of two surfaces at common boundaries and to ensure a tensor-product configuration close to the boundary. Average boundary coefficients. Ensure that the two rows of coefficients closest to an edge lie along a straight line and the corner coefficients lie in a plane.		*The same functionality may in principle be implemented by different algorithms.*
Implementation details	**Implementation language**	C++	*Include information related to the implementation of the service, such as language (e.g., C, C++, etc); dependencies with other libraries (e.g., ANN library); constraints on the operating systems, and visualization modalities of the output.*
	Dependencies with other libraries		
	Operating system	Linux	
	Visualization modalities of the output	Visualization of LR B-spline surfaces	
IQmulus Data	**Available IQmulus input data**: HRW survey 473513		*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:** Resulting in discontinuities in the magnitude of $1.0e^{-10}$.		*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.*
	Robustness: Expected to be robust. Requires more testing.		
	Computational time in relation to data size: The computational time depends on the number and size of the input surfaces. Less computational		

	expensive than the surface generation itself, needs more testing.	*See the following box.*
	Locality/globality of the algorithm: Semi-local. The interfaces between two and two surfaces along edges and four and four surfaces in corners are treated separately.	
Alternatives	<List of Service IDs>	*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	Marine 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	SINTEF Vibeke Skytt, Vibeke.Skytt@sintef.no	*Partner ID and responsible person (include email)*
Involved Partners		

IQmulus Service #86			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	ClassifyPntSfDist		*Unique identifier of the service; necessary to call it within user-defined workflows*
Description	Classify points according the distances between a point cloud and a corresponding LR B-spline surface		*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*
Service functionality	**Input: <data representation and format>**	LR B-spline surface, possibly trimmed (.g2) Point cloud (.las, .txt, .ply, .g2)	*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.*
	Input parameters: *[optional]*	Specification of classification	
	Output: <data representation and format>	Classified point cloud (.txt or .ply) Classification informations (.json)	
	Functionality of the service: <text>	Classification based on computed distance	
Algorithm	Define an intermediate data structure to enable fast access to the basis functions of the LR B-spline surface. Evaluate the surface in the x- and y-values of all data points and compute the distance. The points are classified according to this distance.		*The same functionality may in principle be implemented by different algorithms.*
Implementation details	**Implementation language**	C++	*Include information related to the implementation of the service, such as language (e.g.,*

	Dependencies with other libraries	liblas, rply and GoTools are integrated in the executable	*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	
	Visualization modalities of the output		
IQmulus Data	**Available IQmulus input data**: Original version of point clouds from #12 and #13 (Four), #23 (Liguria), REUNION data sets from UBO, various HRW data sets		*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:** Evaluation according to machine precision		*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: Small changes in the surface or point cloud will lead to small changes in the classification		
	Computational time in relation to data size: The computation is linear in the number of data points. Pre-processing time depends on the surface size.		
	Locality/globality of the algorithm: The evaluation is local, the pre-processing is global		
Alternatives	<List of Service IDs>		*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	Marine 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	SINTEF Vibeke Skytt, Vibeke.Skytt@sintef.no		*Partner ID and responsible person (include email)*
Involved Partners			

IQmulus Service #88			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	ComputeDistPointsSurf		*Unique identifier of the service; necessary to call it within user-defined workflows*
Description	Compute the distance field between a point cloud and a corresponding LR B-spline surface		*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*
Service functionality	**Input: <data representation and format>**	LR B-spline surface, possibly trimmed (.g2) Point cloud (.las, .txt, .ply, .g2)	*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.
	Input parameters: *[optional]*		
	Output: <data representation and format>	Summary of distance information Distance field given as x, y, z and distance (.txt or .ply)	
	Functionality of the service: <text>	Distance field computation	
Algorithm	Define an intermediate data structure to enable fast access to the basis functions of the LR B-spline surface. Evaluate the surface in the x- and y-values of all data points and compute the distance.		*The same functionality may in principle be implemented by different algorithms.*
Implementation details	**Implementation language**	C++	*Include information related to the implementation of the service, such as language (e.g., C, C++, etc); dependencies with other libraries (e.g., ANN library); constraints on the operating systems, and visualization modalities of the output.*
	Dependencies with other libraries	liblas rply and GoTools are integrated in the executable	
	Operating system	Linux	
	Visualization modalities of the output	Visualization of distance field	
IQmulus Data	**Available IQmulus input data**: Original version of point clouds from #12 and #13 (Four), #23 (Liguria), REUNION data sets from UBO, various HRW data sets		*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:** Evaluation according to machine precision		*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: Small changes in the surface or point cloud will be exactly reflected in the distance field.		
	Computational time in relation to data size: The computation is linear in the number of data points. Pre-processing time depends on the surface size.		
	Locality/globality of the algorithm: The evaluation is local, the pre-processing is global		
Alternatives	<List of Service IDs>		*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	Marine 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	SINTEF Vibeke Skytt, Vibeke.Skytt@sintef.no		*Partner ID and responsible person (include email)*
Involved Partners			

IQmulus Service #90			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	TrimSurfWithPoints		*Unique identifier of the service; necessary to call it within user-defined workflows*
Description	Bound an LR B-spline surface with one or more loops describing the extent of the point cloud from which the surface was generated. The result will be that the domain of the trimmed surface will correspond to the area covered by the point cloud.		*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*
Service functionality	**Input: <data representation and format>**	LR B-spline surface: g2 format Point cloud: las, xyz or g2-format	*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.*
	Input parameters: *[optional]*	Tightness parameter. How close to the point cloud should the trimming curve go? A number in the interval [1,8].	
	Output: <data representation and format>	Trimmed B-spline surface: g2-format	
	Functionality of the service: <text>	Trimming of LR B-spline surface	
Algorithm	The point cloud is recursively divided into sub clouds. Each cloud is bounded by a rectangle and the outer boundaries of these rectangles are collected into inner and outer trimming loops. These loops are approximated by parameter domain spline curves which are used to limit the surface.		*The same functionality may in principle be implemented by different algorithms.*
Implementation details	**Implementation language**	C++	*Include information related to the implementation of the service, such as language (e.g., C, C++, etc); dependencies with other libraries (e.g., ANN library); constraints on the operating systems, and visualization modalities of the output.*
	Dependencies with other libraries	liblas	
	Operating system	Linux	
	Visualization modalities of the output	Visualization of the trimmed LR B-spline surface	
IQmulus Data	**Available IQmulus input data**: Point clouds and corresponding approximating LR B-spline surfaces from #12 and #13 (Four, las version more appropriate), #23 (Liguria), REUNION data sets from UBO, various HRW data sets		*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**: The domain is bounded by the constructed trimming loop. It is not interpolated. Accuracy depends on the tightness parameter.		*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.*
	Robustness: Unexpected results may occur for very scattered data sets. Small changes in the data points will induce none or small changes in the result.		
	Computational time in relation to data size: The computation is linear with respect to data size provided that the tightness parameter and the complexity of the domain represented by the point		

	cloud are kept fixed. **Locality/globality of the algorithm:** In essence global. Local results are merged in a recursive procedure.	*See the following box.*
Alternatives	None. In the triangulation context, the services #49 and #51 creates triangulated surfaces adapting to the domain of the input data. Service #49 takes additional input describing the domain boundary.	*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	Marine 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	SINTEF Vibeke Skytt: Vibeke.Skytt@sintef.no	*Partner ID and responsible person (include email)*
Involved Partners		

IQmulus Service #91			
Name of the metadata	**Content expected**		**Motivation/comments**
Service Acronym	SplineSurf2Raster		*Unique identifier of the service; necessary to call it within user-defined workflows*
Description	Extract raster model from LR B-spline surface. The raster is exported to geotiff.		*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*
Service functionality	**Input: <data representation and format>**	LR B-spline surface, g2 format	*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.*
	Input parameters: *[optional]*	Cell size, nodata value Optional: sub domain for raster generation	
	Output: <data representation and format>	Raster, geotiff	
	Functionality of the service: <text>	Create raster from surface	
Algorithm	Define a regular grid based on the surface domain and cell size. Evaluate the surface in this grid and remove grid points lying outside eventual trimming loops.		*The same functionality may in principle be implemented by different algorithms.*
Implementation details	**Implementation language**	C++	*Include information related to the implementation of the service, such as language (e.g., C, C++, etc); dependencies with other libraries (e.g., ANN library); constraints on the operating systems, and visualization modalities of the output.*
	Dependencies with other libraries	geotiff2	
	Operating system	Linux	
	Visualization modalities of the output	Visualization of geotiff raster	
IQmulus Data	**Available IQmulus input data**:		*If there are examples of data*

	LR B-spline surfaces approximating point clouds from #12 and #13 (Four, las version more appropriate), REUNION data sets from UBO	*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:** The raster interpolates the surface **Robustness:** Perturbation in the surface will be exactly carried over to the raster **Computational time in relation to data size:** Linear in the number of cells provided that the input surface size and the complexity of the surface domain are kept fixed. Trimmed surfaces require slightly more effort than non-trimmed ones. **Locality/globality of the algorithm:** Not relevant	*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.*
Alternatives	Given the format of the input: none	*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	Marine 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	SINTEF Vibeke Skytt: Vibeke.Skytt@sintef.no	*Partner ID and responsible person (include email)*
Involved Partners	Testing by UBO in the context of Marine workflow 4	

IQmulus Service #40			
Name of the metadata			**Motivation/comments**
Service Acronym	krigingRAIN		*Unique identifier of the service; necessary to call it within user-defined workflows*
Description	Approximation of rain fall data, measured at a number of raingauge sites, using as additional information the topography of the area. The problem is to use correctly data with different resolution (very sparse raindata with respect to the DEM). The DEM should have enough details (resolution) to highlight main topographic features without loading the system too much.		*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*
Service functionality	**Input: <data representation and**	Raindata: Enriched PC – ASCII - DEM: Enriched PC – ASCII or	*Include here the input/output of the*

	format>	PLY	*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.*
	Input parameters: *[optional]*	Model variogram structure; the structure should be expressed as type (e.g. Gaussian, spherical, etc) and as its parameters (e.g. nugget, sill, isotropic or anisotropic rangw and orientation of anisotropy. User may select between different algorithms	
	Output: <data representation and format>	Rain field: Gridded point cloud (GeoTIFF) – Enriched PC – PLY	
	Functionality of the service: <text>	Interpolation of point data with kriging paradigm.	
Algorithm	• Ordinary kriging with external drift • Simple kriging with local varying mean • Ordinary cokriging		*The same functionality may in principle be implemented by different algorithms.*
Implementation details	**Implementation language**	C++, Fortran	*Include information related to the implementation of the service, such as language (e.g., C, C++, etc); dependencies with other libraries (e.g., ANN library); constraints on the operating systems, and visualization modalities of the output.*
	Dependencies with other libraries	Read and write libraries for GeoTIFF and shapefile, GSLIB, GsTl	
	Operating system	Linux-like	
	Visualization modalities of the output	• 3D visualization of the rainfall map with different colours, possibly in relation to rain value • possibility to overlap the rainfall map with 3D DEM and watershed map.	
IQmulus Data	**Available IQmulus input data**: Dataset 22, Dataset 27, Dataset 26, Dataset 42, 43		*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:** The ordinary kriging estimator is an exact estimator.		*These fields are necessary to document all the characteristics of the services that are important to assess*
	Robustness: Small perturbations in the point set will give small changes in the estimate.		

	Computational time in relation to data size: It is linear respect the size of DEM, it has computational cost of $O(M \log N)$ respect the size of rain fall record (N) and the number of neighbours samples (M) **Locality/globality of the algorithm:** The estimation process is local and it involves only the M nearest samples of the point we are going to estimate.	*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.*
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	Use case (IQmulus) #1107	*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	• CNR-IMATI, Simone Pittaluga (simone.pittaluga@ge.imati.cnr.it) • TUDelft	*Partner ID and responsible person (include email)*
Involved Partners	• Regione Liguria	