3D OBJECT SHAPE RECONSTRUCTION FROM UNDERWATER MULTIBEAM DATA AND OVER GROUND LIDAR SCANNING

Marek Kulawiak
Zbigniew Łubniewski
Gdańsk University of Technology, Poland

ABSTRACT

The technologies of sonar and laser scanning are an efficient and widely used source of spatial information with regards to underwater and over ground environment respectively. The measurement data are usually available in the form of groups of separate points located irregularly in three-dimensional space, known as point clouds. This data model has known disadvantages, therefore in many applications a different form of representation, i.e. 3D surfaces composed of edges and facets, is preferred with respect to the terrain or seabed surface relief as well as various objects shape. In the paper, the authors propose a new approach to 3D shape reconstruction from both multibeam and LiDAR measurements. It is based on a multiple-step and to some extent adaptive process, in which the chosen set and sequence of particular stages may depend on a current type and characteristic features of the processed data. The processing scheme includes: 1) pre-processing which may include noise reduction, rasterization and pre-classification, 2) detection and separation of objects for dedicated processing (e.g. steep walls, masts), and 3) surface reconstruction in 3D by point cloud triangulation and with the aid of several dedicated procedures. The benefits of using the proposed methods, including algorithms for detecting various features and improving the regularity of the data structure, are presented and discussed. Several different shape reconstruction algorithms were tested in combination with the proposed data processing methods and the strengths and weaknesses of each algorithm were highlighted.

Keywords: multibeam sonar; laser scanning; three dimensional reconstruction; point cloud; triangulation

INTRODUCTION

Acoustic sensing of seafloor and underwater objects used to obtain their 3D shape and relief representation has several applications, which include acquiring the underwater information with respect to safe navigation, positioning of offshore installations such as oil platforms or oil and gas pipes, marine archaeology where it aids to investigate and visualise in 3D the underwater objects like wrecks and other man-made constructions, recognition of topographical features of the seabed for maps construction and in the context of applications where the 3D mesh of bottom terrain is needed for modelling various processes occurring in waters, as well as research, monitoring and visualization of marine environment pollution [3].

At the same time, the over terrestrial LiDAR (*Light Detection And Ranging)* scanning applications include the creation of detailed 3D topographic maps [4], modelling of various processes and phenomena related to the area of research in urban areas [3] or local prediction of the accuracy of satellite navigation systems when taking into account obstructions such as satellite coverage and multipath signal propagation, the protection of critical infrastructures, the creation of 3D terrain visualization systems dedicated for various purposes (e.g. supporting the training of emergency services with the use of real-field simulation [8]), the recreation

of detailed digital models depicting architectural monuments, allowing for their later restoration in the case of damage or destruction.

The data describing the shape of researched areas, obtained by either probing the seafloor with a multibeam echo sounder or scanning over-ground terrain with a LiDAR device, are usually available in the form of groups of separate points located at irregular intervals in three-dimensional space, known as point clouds. This data model has several disadvantages, i.e.: it may be difficult to spot some features in a point set (such as small objects) which would be easier to notice on the surface of a solid object, some datasets may need to be viewed from specific angles if their point density is low, even primitive surfaces such as flat building roofs have to be represented by a dense group of points (see Fig. 1) which causes the data to require unnecessary large amounts of disk storage and system memory.
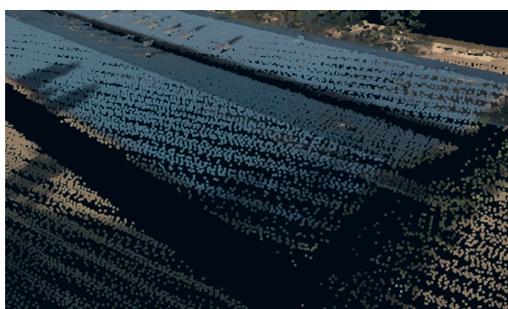
Fig. 1. Sample dense point cloud representing a set of buildings of relatively simple shape. Image captured with a visualization application created by one of the authors

For these reasons, a more practical solution involves the use of three-dimensional spatial objects in the form of surfaces consisting of higher order geometric structures like edges and facets, known as triangulated irregular network (TIN) models. What is more, the overall shape of many of the researched objects is mostly regular and features fragments such as flat walls and smooth seabed. It means that representing them in the form of solid meshes results in simplifying their visualization, also in the context of memory usage and the performance of various algorithms used to process the data during later steps.

Many solutions exist for recovering the shape of the seafloor and underwater objects which can be used for data acquired by multibeam echo sounders and side scan sonars. Also, the dedicated, commercial software packages like Caris HIPS and SIPS [6], Kongsberg Seafloor Information System (SIS) [21] or QINSy Software [19] support to some extent such functionality with respect to acquired multibeam datasets. However, the utilised approaches rely mainly on straightforward application of TIN surface construction or other meshing procedure directly from point cloud data, what frequently causes an occurrence of numerous artefacts and in consequence produces minor quality results.

Attempts to recover the higher order geometric structures of underwater objects are not widely reported in literature. Notable methods include analysing the adjacency of points in a dataset and detecting the acoustic shadow zones [4], dividing the data into several blocks and constructing Delaunay triangulations in sub-blocks individually [16], integration of data obtained by both acoustic and optical sensors through geometrical correspondences and registration [10] as well as adopting a stereo-like vision approach based on image matching [17].

In case of over ground areas, the existing solutions for three dimensional reconstruction of investigated objects shape are primarily used with LiDAR data and include approaches such as: assembling building blocks from a set of standard roof shapes [11], roof plane segmentation performed by minimizing an energy function [13], merging LiDAR and hyperspectral image data and performing shape reconstruction using the implicit geometry method [18], decomposing building footprints and estimating roof models with the use of RANSAC technique [9], as well as reconstructing building roofs from LiDAR data integrated with optical multi-view images [7].

From more detailed analysis of the state of the art in the mentioned subject, it may be concluded that the existing solutions are partially satisfactory and still need extensive research. Up to date, a large number of various algorithms dedicated to transforming point cloud data into more complex TIN models [12][2][23][1] is known. Unfortunately, due to the nature of data acquisition methods based on the use of technologies related to acoustic sounding and laser scanning, the end results often end up being unsatisfactory when standard surface reconstruction techniques are used. Common problems include significant amounts of noise (i.e. existence of large numbers of points which should be excluded from further processing), frequent lack of data, as well as strong variability of local point density and data accuracy inside a single point set. To a certain extent, the data in this form can be used in order to recreate the surface of simple objects [5] and in some cases also buildings [22] [20]. However, in case of more complex and varied objects, obtaining satisfactory results is often much more difficult to achieve [14].

The creation and application of a new method for three-dimensional reconstruction specialized for the aforementioned types of data may enable the creation of more accurate and at the same time less complex models of the researched objects, which can later be used in various geographic information systems and other software offering the visualization of three-dimensional scenes. In this paper, the authors propose a new approach to 3D shape reconstruction from both multibeam and LiDAR measurements. It is based on a multiple-step and to some extent adaptive process, in which the chosen set and sequence of particular stages may depend on the current type and characteristic features of the processed data.

DATA DESCRIPTION

The input data described in this article have the structure of georeferenced three-dimensional point clouds. Most of the

underwater data were acquired with a multibeam echosounder with 160 beams, mounted to the bottom of a vessel moving at an average speed of 1 m/s with a ping rate equal to 0.96 Hz. Since most of the research was performed in 43 m water depth, the average distance between consecutive points in a single swath is equal to 0.28 m. The terrestrial data consists of multiple laser scans obtained with the use of LiDAR devices, resulting in a large dataset with the average resolution of around 19 points/m². Even though the data originate from many different sources (Gdańsk University of Technology, Kongsberg Maritime AS, Maritime Office in Gdynia, Polish Centre of Geodesic and Cartographic Documentation), their spatial structures are quite similar and have the form of unorganised point clouds with varying density, which is caused by the fact that the method used for acquiring the data involves the use of vehicles such as ships and aircrafts which gather the data while remaining mostly at the same altitude. Because of this, the surfaces which are parallel to the vehicle's track of movement, like ground and rooftops, are represented in greater detail than other surfaces, such as walls and masts. This means that the spatial structure of these data are actually similar to raster height maps. The LiDAR data sets contain additional information, such as the colour value of each point, as well as classification information where every point is assigned to a single class representing objects such as buildings, ground, water as well as different types of vegetation.

DATA PROCESSING METHODS

The proposed solution for reconstructing data obtained by multibeam sonars and LiDAR systems is a multiple-step process which is presented in Fig. 2.
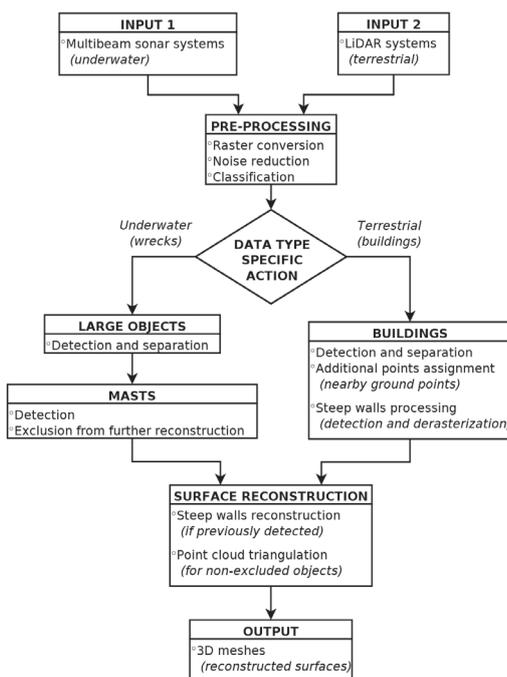

Fig. 2. The flow chart of the proposed solution

Since both of these types of data usually have a similar spatial structure, many of the proposed methods can be applied to datasets obtained from various different sources after performing proper pre-processing. After that, the pre-processed point clouds are split into separate objects using methods dedicated to specific types of data (described in section 3.2). Finally, the shape of each object is reconstructed by specialized triangulation algorithms (section 3.3) designed for the proposed file structure and 3D models of surfaces are generated and saved on the disk.

PRE-PROCESSING

Since the input data originate from different sources, they have to be converted into a common file structure so the same processing algorithms can be used for different types of data. One of the major steps performed during this process is converting the input data from the form of irregular point clouds into the structure of regular raster grids. This greatly simplifies further steps of the processing pipeline, such as the application of feature detection algorithms, and allows for the use of point cloud triangulation methods designed to work in two-dimensional space. Two approaches are used for determining the optimal resolution for creating raster network points from input data:

I. Since most underwater datasets were obtained with the use of a vessel maintaining mostly the same heading (the difference to the desired direction was usually not greater than 15 degrees) during the seabed measuring, these data are treated as a set of raster rows (where each row is represented by a single swath), and the number of columns is equal to the number of the echosounder beams.

II. For other data (including terrestrial datasets), the process of generating raster network points is explained in greater detail in further parts of this section.

During this step, additional noise reduction methods are applied, which remove or minimize the distortions caused by specific types of irregularities while preserving as much information as possible. In general, three types of problems are taken under consideration [15]:

I. Lack of data in some areas, which may affect the results of automatic object detection and cause the creation of unnecessary spaces lacking data in the process of surface reconstruction. For this reason, the proposed approach is to generate new points for these areas in order to make the data structure more regular. The values for the newly introduced points can be provided by coping the height values from non-empty neighbours.

II. Strong variability of local point density, which can be solved by converting the entire data into the form of a regular raster. This is done by dividing the point set into a grid of two-dimensional sectors of equal size and assigning each point to a single element. Every sector is then converted into a single point in the resulting dataset, where its height will be equal to the average height of all points located inside this sector.

Figure 3 provides an additional explanation of solving the problem of varying point density combined with the idea of filling empty areas with generated points. The tile size is decided by empirically testing different values for the output number of rows and columns. The chosen resolution is considered optimal if the number of no-data points in the output raster (prior to filling empty areas) does not exceed 5% of all points in the converted dataset.

III. Large amounts of local noise, which strongly affects the final reconstruction process and causes the creation of surfaces of highly undesirable shapes. This can be solved by using an algorithm such as the "envelope filter", which divides the distorted area in small groups of points and preserves only the most significant ones which would describe the outer surface of the data in the best possible way. Figure 4 shows the results of applying the envelope filter while introducing new points with values calculated by the method of linear interpolation.
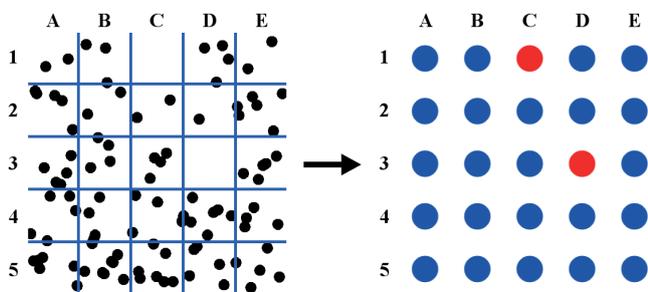

Fig. 3. An example of dividing a dataset into a regular grid of sectors (outlined in blue) and converting it into a two-dimensional array of points. The empty sectors C2 and D3 are replaced with newly generated points (depicted in red) calculated from height values of non-empty neighbour points

Fig. 4. A sample set of points (a) located in two-dimensional space is processed by the envelope filter, resulting in a point set (b) representing the outer surface of the object

The last important step which has to be performed during the pre-processing stage is detecting individual objects in the data and assigning them to separate classes. Data classification was proven to be a successful technique for improving the quality of various surface reconstruction methods for underwater objects [14] and can offer the same benefits for similar types of data, such as the possibility of using different methods and their parameters for recognized objects depending on their type (additional operations can be performed on complicated objects such as buildings and shipwrecks which would not be required for simpler objects like ground and seabed), as well as the overall improvement of the results provided by various shape reconstruction algorithms, as sometimes they fail to properly preserve the gaps found between separate objects if they lack classification information and are treated as a single objects.

DATA TYPE SPECIFIC ACTION

Different procedures are applied to the data depending on their contents. For this reason, the following section is split into two categories, one dedicated to underwater environment and the other one to terrestrial objects.

Underwater environment

A specialized algorithm for dividing the data into separate objects is applied to files representing contents of an underwater environment. This algorithm consists of several steps explained in the following section and has customizable parameters which make this method more flexible. The results of performing each step of the algorithm on a sample dataset using default values for input parameters are presented in Figure 6.

I. Temporary removal of details: A copy of the dataset representing the height of each point is created and any local irregularities in the data are reduced by processing them with a median filter algorithm using a 3×3 kernel. This copy is then used as reference for further steps of the algorithm, otherwise the results would be heavily affected by any local irregularities present in the original data.

II. Initial classification: The dataset is divided into two classes depending on the height delimiter defined by one of the algorithm's customizable parameters. By default, each point with a height less or equal to a set value (function parameter) of the dataset's height delta is assigned to the class representing the seabed, while all remaining points are preliminarily classified as a part of some underwater objects. The default value of this delimiter is equal to 10%, which is satisfactory for data describing large underwater objects like shipwrecks, while for other data the algorithm should eventually assign all points to the seabed class.

III. Classification correction: The algorithm enters a correction loop, which attempts to change the assignments of some of the points which were not classified as a part of the seabed in the previous step. In each iteration of the loop the algorithm checks every point located on the border of the seabed class and compares it with its neighbours from the other class – if the height difference between a neighbour point and the border point is not greater than a set value (by default it is 5% of the dataset's height delta), then this neighbour is also assigned to the seafloor class. The loop continues as long as there are potential points which could be assigned to the seabed class. A detailed step-by-step example of applying the correction loop to the sample point set is presented and explained in Fig. 5. The initial state of the point set (after applying preliminary classification) is shown in Fig. 5 a), where the blue points represent seabed while

the red points represent the underwater objects class. In Fig. 5 b) the algorithm checks the height of points A, C and F, which are the nearest neighbours of point B and classifies point F as a part of the seafloor. In Fig. 5 c) the neighbours B, D, and G of point C are checked by the algorithm and as a result point G is assigned to the seabed class. In Fig. 5 d) the nearest neighbours of point F are tested for the possibility of shifting classes, but none of them are assigned to the seafloor class, as the height difference between the pairs of points FE and FJ are too high while points B and G are already classified as parts of the seafloor class. In Fig. 5 e) and Fig. 5 f) points K and L are assigned to the seabed class. In Fig. 5 g) the point classification remains unchanged. The final result of applying the correction process is shown in Fig. 5 h).
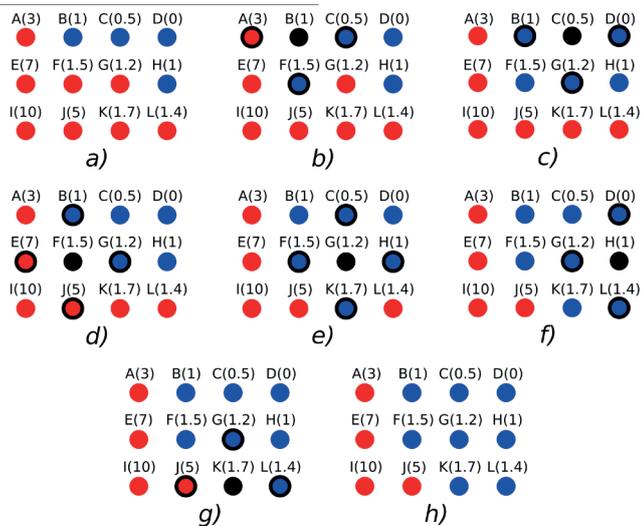
Fig. 5. A step-by-step process of performing each iteration of the correction loop part of the object detection algorithm, where the delimiter for assigning neighbour points is equal to 0.5 units

IV. Object separation: Separate groups of points which are surrounded by points classified as seabed are assigned to new classes, where each class is unique to a single group. After this step, every class should represent a different underwater object, such as a shipwreck. Figure 6 presents the results of applying two steps of the classification algorithm to a point cloud featuring a single shipwreck (Fig. 6 a): initial classification (Fig. 6 b) and classification correction (Fig. 6 c) using the default parameter values.

Among the objects described by the input data there are some which are not suitable for further processing, mainly because they are represented by a very small number of points, which is insufficient for automatic reconstruction of high-quality meshes. In case of files representing the approximate surface of underwater environments, the shipwreck masts are a notable example of such objects. In case of land data, the objects which are not suitable for reconstruction include different types of vegetation, as well as other objects such as lamp posts. These types of objects are removed from further processing in order to improve the overall quality of mesh reconstruction.
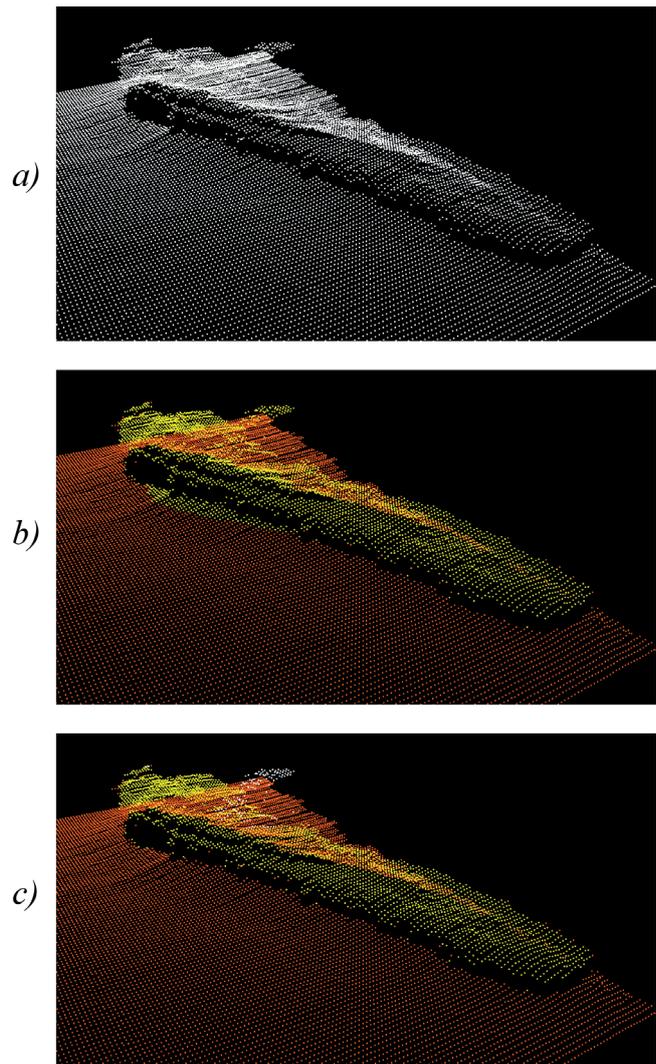

Fig. 6. The results of applying several steps of the classification algorithm to a single point cloud featuring a shipwreck lying in the Gulf of Gdansk at 43 m water depth

The algorithm for detecting masts in an object describing a shipwreck is as follows. First of all, the object is divided into regular sectors, where each sector represents a group of points and is assigned a value equal to the position of the highest point in this group. After this step, each sector whose value is larger or equal to 60% of the object's height is preliminarily classified as a fragment of a mast (an assumption is made that the shipwreck is oriented in such a way that its bottom is mostly lying on the seabed). Those sectors which are connected with each other are then merged into larger groups. Finally, these groups whose number of points is larger than 5% of all points of the processed object are considered to be something entirely else than masts and are assigned to the base "non-masts" class. All points classified as masts are then copied into a separate object, while the originals are removed from the source object and replaced with newly generated points to fill up the created gaps. Fig. 7 compares the results of reconstructing a sample point set with and without applying the masts detection algorithm.
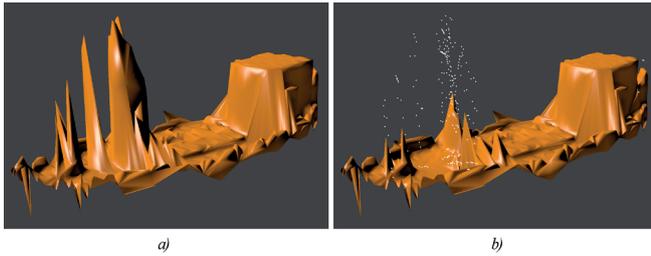
Fig. 7. The results of performing surface reconstruction on the point set describing the Cleona shipwreck (lying near Asker in 18 m water depth), before (a) and after (b) applying the masts detection algorithm

Over ground objects

Since all of the files representing land objects used in this work already contain classification information, the amount of processing for these data is significantly lower than in case of underwater environments. Additional steps are performed for land objects classified as buildings:

I. Distant objects are separated from each other and processed independently.

II. Ground points located next to a building are assigned to that building. This is done to make sure that each wall of a reconstructed object actually touches the ground.

III. Walls are detected for each building with the following algorithm. First of all, the minimum and maximum height for each building is determined. After that, the algorithm iterates through each row and column of the grid describing current building and searches for areas where the height difference between neighbour points is very large. If a pair of points is found where the height difference between them is larger or equal to 25% of the height difference between the highest and the lowest point of the building, then both points are classified as a part of a wall. The results of applying this wall detection algorithm can be seen in Fig. 8.
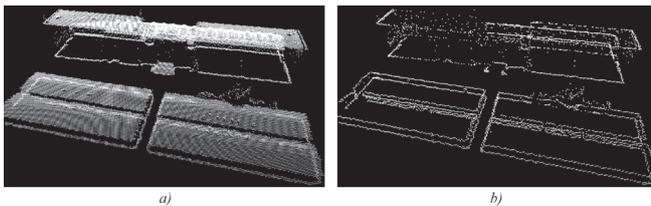

Fig. 8. Sample dataset containing several buildings of the Gdansk University of Technology (a) processed by the wall detection algorithm (b)

IV. Additional edge de-rasterization algorithm is performed in order to reduce irregularities caused by raster characteristics of the data structure by altering the positions of some of its points. For this reason, a special dictionary of patterns was prepared, which features a number of various shapes which can be used to describe straight lines in raster graphics, along with proper vector representations of each line. This dictionary is then used by the algorithm in order to find groups of points representing walls of a building and modify their positions in such a way that they form actual straight lines in 3D space. Several notable examples of representing lines in raster graphics are shown in Figure 9 a), while Figure 9 b) shows proper replacements of these

shapes in vector graphics. The algorithm also affects the positions of points located near detected walls by moving them slightly towards these walls. This is performed mainly in order to improve the shading of the output models when calculating per-vertex normals. The final results of the edge de-rasterization algorithm combined with previous steps explained in this paragraph can be seen in Fig. 10.
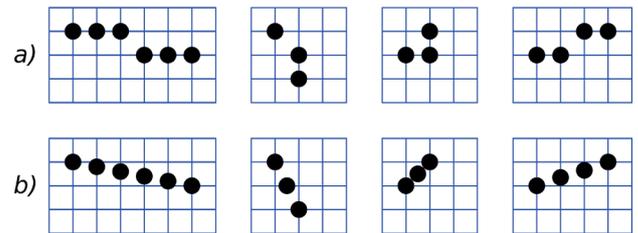

Fig. 9. Replacement tables used for altering raster representations of sample lines shown in a) along with the desired shapes of these lines presented in b)
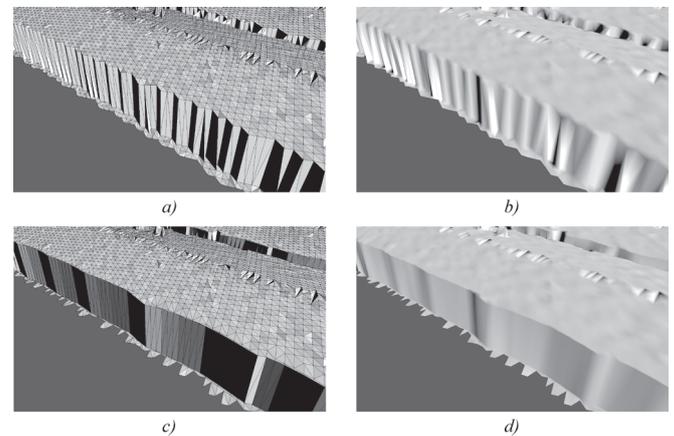

Fig. 10. Surface reconstruction results for data prior (a-b) and after (c-d) the application of the edge de-rasterization algorithm

SURFACE RECONSTRUCTION

For underwater data, as well as for terrestrial objects for which no walls were detected in previous steps, the entire surface of the object is recreated by using a specialized algorithm which accepts a two-dimensional grid of points as input and builds triangles on top of chosen points after comparing their heights. The algorithm analyses the data by iterating through points located in the same row (for underwater data, each row describes the contents of a single swath) and looking for potential candidates for forming a triangle in the next (neighbour) row. Given the nature of the data structure, assuming that it was properly pre-processed, this limitation should not exclude valuable candidates from the process and it also grants the possibility of performing parallel computations. During each iteration, the current point and its next neighbour in the same row are automatically picked for creating the next triangle, while the third required point is chosen from the next row. The third point is selected from a subset of points in the next row which are located

near the first two points and the best candidate must fulfil the following requirement: its height being the closest to the average value of the first two points. Finally, a single triangle is added on top of the two current points in the first row and the third point from the second row. After that, the algorithm fills any holes which might have been created during that step. A step-by-step example of reconstructing a sample grid of heights is presented in Figure 11. Fig. 11 a) depicts the input point set, where the height of each point is given in brackets. Fig. 11 b) shows the first step of the algorithm, where the triangle CFE was created, since points E and F were the first points in the first row and the height of point C from the second row was closest to the average value of the pair EF. Fig. 11 c) represents the results of performing the next step of the algorithm, where the holes between points CEA were filled with two new triangles: ABE and BCE. Fig. 11 d) shows the next iteration of the algorithm using the same logic as in Fig. 11 b). Fig. 11 e) and Fig. 11 f) depict the next steps which the algorithm performs in order to fill the remaining holes by introducing new triangles CDF and DHG. Fig. 11 g) represents the final result of applying the triangulation algorithm, where the sub-figure CDGFE is distinguished to point out that the algorithm is capable of generating surfaces with properly placed edges between steep walls which makes the output model look more natural.
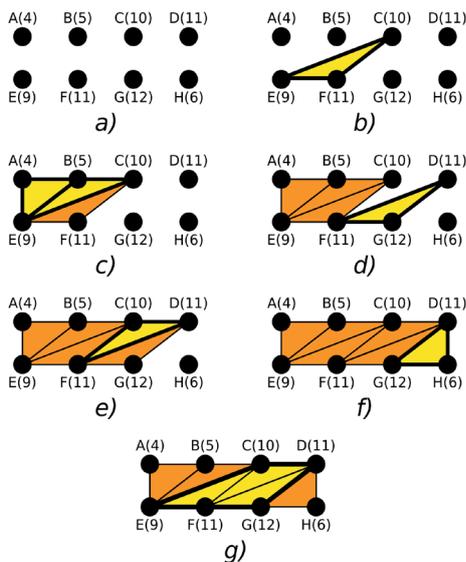

Fig. 11. An example of reconstructing a small point set using the triangulation algorithm explained above.

Otherwise, if the data describe terrestrial objects with detected walls, a multi-step approach is used, which consists of dividing each object into smaller rectangular sectors (each consisting of up to four points) and reconstructing each sector with the proper triangulation algorithm. The default algorithm is used for sectors which do not contain any points classified as walls. The other sectors are reconstructed with an algorithm which performs different actions depending on the number of points in each sector which were classified as a part of walls. If the number of "wall points" in a sector is equal to 4, then there are only two different ways of triangulating

this sector possible as shown in Figure 12. Both of them offer correct methods of building triangles from given points, but better quality of the reconstructed surface can be achieved if the proper method is chosen with the following condition. The choice between methods shown in Fig. 12 a) and Fig. 12 b) is made by comparing the following pairs of points: BD and AC, and determining which pair should form an edge. If points B and D are the two highest or the two lowest points in this sector, then option a) is chosen for the reconstruction process, otherwise option b) is used instead.
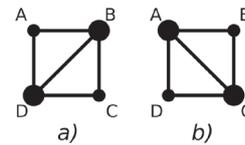

Fig. 12. Two different methods of triangulating a sector of four non-empty points.

If the number of "wall points" in a sector is equal to three, then the solution is straightforward and involves two steps: creating a triangle from these three points and if the fourth point is not empty, introducing an additional triangle in a way that fills the remaining hole in this sector. There are exactly four different scenarios which have to be considered in this step, as presented in Fig. 13.
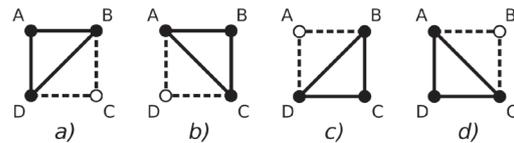

Fig. 13. Possible methods of triangulating a sector featuring exactly three "wall points"

All remaining scenarios of triangulating a sector which are considered by the algorithm are presented in Fig. 14. If the number of "wall points" is equal to two and they form the diagonal BD, then up to two triangles are created, depending on whether the points A and C are empty or not, as seen in Fig. 14 a) and Fig. 14 b). If points B and D are both not classified as a part of a wall and there are no empty points in the sector, then two triangles are built using the method shown in Fig. 14 c). Otherwise, if the sector contains only three non-empty points in total, then only a single triangle can be created using one of four possible methods presented in Fig. 14 d) e) f) and g).
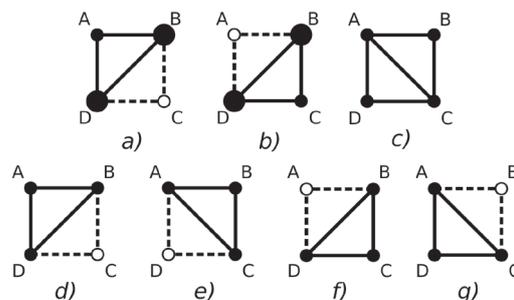

Fig. 14. Remaining methods of triangulating a sector featuring no more than two "wall points"

The difference between results obtained by the multi-step triangulation algorithm and the single-step version in many cases can be a subtle one, but is clearly noticeable in situations where the reconstructed object contains regular plain walls, as seen in Fig. 15.
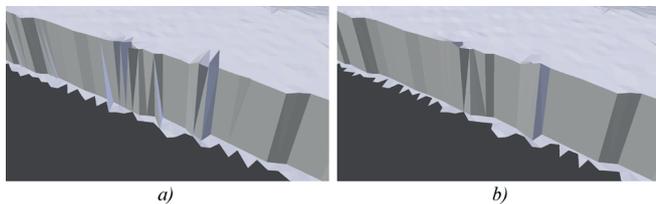

Fig. 15. The subtle difference between performing the single-step (a) and the multi-step triangulation.

RESULTS

In the research the proposed triangulation algorithms have been tested for reconstructing the shape of several objects represented by point cloud datasets processed with the use of methods described in paragraph 3.1-3.2. The same data were then used to obtain meshes using known methods for 3D surface reconstruction, including two-dimensional Delaunay triangulation, the Ball-Pivoting algorithm and the Poisson Surface Reconstruction method. The results shown for these known methods are the best ones, obtained by empirically testing the available parameters upon which they depend.

The first set of tests was performed on the low-resolution dataset shown earlier in Fig. 6, and the results of these tests can be seen in Fig. 16.
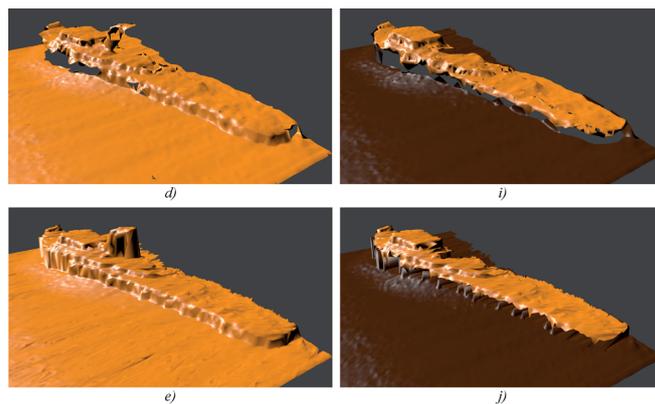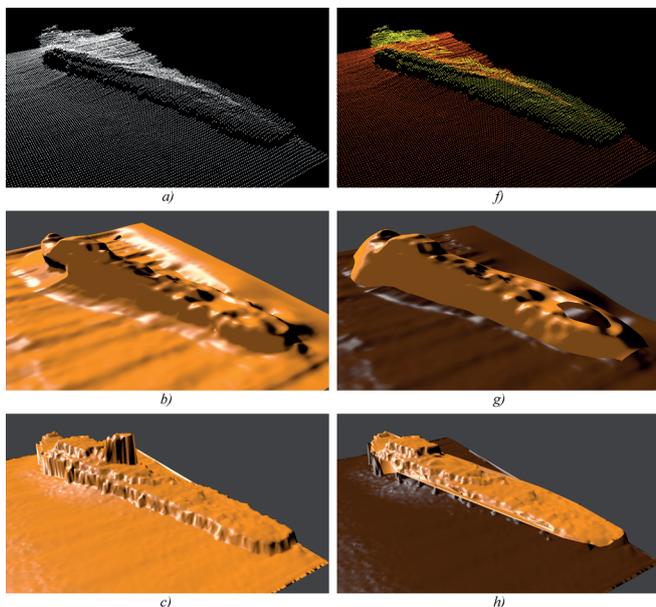

Fig. 16. The results of applying several surface reconstruction methods to two versions of the same dataset representing a low-resolution shipwreck (a – unmodified, f – pre-processed): Poisson surface reconstruction (b, g), 2D Delaunay Triangulation (c, h), Ball Pivoting algorithm (d, i) and the proposed triangulation method (e, j).

In this case, the result obtained with the Poisson surface reconstruction (Fig. 16 b, Fig. 16 g) did not preserve any of the details seen in the original data. On the other hand, the remaining methods offered noticeably better results. Using these methods, three very similar models were generated, where the mesh created by Ball-Pivoting algorithm (Fig. 16 c) contains a significant amount of holes, while the 2D Delaunay triangulation introduced additional polygons near the boundaries of the wreck (Fig. 16 d). The proposed reconstruction algorithm offers a result (Fig. 16 e) similar to the one created with the Ball-Pivoting method (Fig. 16 d), but it features a fewer amount of irregularities.

The second batch of tests were performed on high-resolution data representing the Cleona shipwreck (Fig. 17a) which were processed by the same methods as the previous sample, but this time the envelope filter was used as well. The results of applying the Ball-Pivoting method were omitted as the reconstructed model of the shipwreck represented only a very small fragment of the input data.
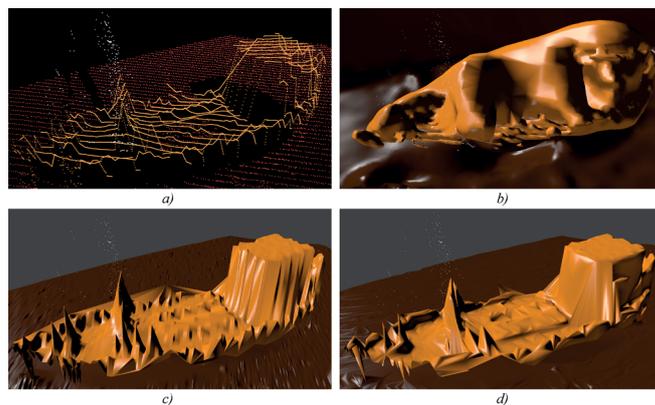

Fig. 17. The results of applying several surface reconstruction methods to a pre-processed dataset (a) representing a high-resolution representation of the Cleona shipwreck: Poisson surface reconstruction (b), 2D Delaunay Triangulation (c) and the proposed reconstruction method (d).

During this experiment, the worst result was obtained by the Ball-Pivoting algorithm which could not handle the input data properly and as such managed to reconstruct only a small fragment of the object, so for this reason it is not featured among the results obtained by other methods. Once again, the Poisson surface reconstruction performed poorly (Fig. 17b), while a fair reconstruction of the object was provided by the 2D Delaunay triangulation (Fig. 17c). The proposed reconstruction algorithm offered a result similar to the one created with the Delaunay triangulation, but it also featured smoother surfaces as seen in Fig. 17d.

The final tests were performed on a LiDAR dataset featuring one of the buildings of the Gdańsk University of Technology, which was processed in a similar way as the previously discussed data files, but with the additional application of the wall detection and de-rasterization algorithm, resulting in a regular point set as seen in Figure 18 a.
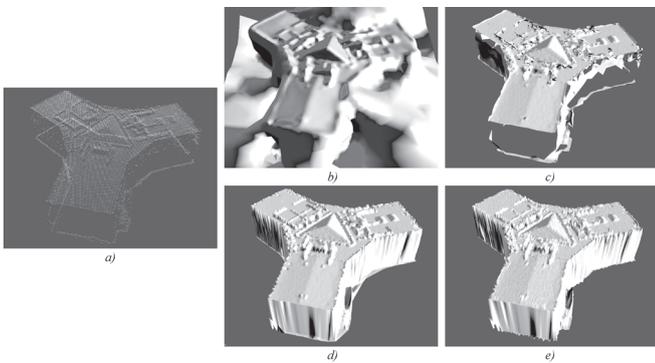

Fig. 18. The results of applying several surface reconstruction methods to a pre-processed high-resolution dataset (a) representing one of the buildings of the Gdańsk University of Technology: Poisson surface reconstruction (b), Ball Pivoting algorithm (c), 2D Delaunay triangulation (d) and the proposed reconstruction algorithm (e).

This time the Poisson surface reconstruction method provided a mesh (Fig. 18 b) which features many details seen in the input point cloud, although it also reduced the sharpness of their edges. The Ball-Pivoting algorithm managed to reconstruct most parts of the building (Fig. 18 c), although the walls were mostly left empty. Both the 2D Delaunay triangulation and the proposed reconstruction method managed to fully recreate the provided building (Fig. 18 d and 20 e) and in this case the mesh generated by the Delaunay triangulation algorithm features slightly more regular surfaces than the mesh reconstructed by the proposed triangulation method. Nevertheless, both methods benefit from using the wall de-rasterization algorithm mentioned before.

CONCLUSIONS

A multiple step solution for processing, classification and reconstruction of point cloud data was presented, dedicated to datasets acquired with the use of multibeam sonars and LiDAR systems, although it should also be adaptable to other sources as long as the input data have similar spatial structure. The benefits of using the proposed methods, including algorithms for detecting various features and improving the regularity of the data structure, were presented and discussed. Several different shape reconstruction algorithms were tested in combination with the proposed data processing methods and the strengths and weaknesses of each algorithm were highlighted.

REFERENCES

1. Amenta N., Choi S., Kolluri R. K., The Power Crust, Proceedings of the sixth ACM symposium on Solid modeling and applications, 2001, pp. 249-266.

2. Bernardini F., Mittleman J., Ftushmeier H., Silva C., Taubin G., The Ball-Pivoting Algorithm for Surface Reconstruction, IEEE Transactions on Visualization and Computer Graphics, vol. 5, No. 4, 1999, pp. 349-359.

3. Chybicki, A., Kulawiak, M., Lubniewski, Z., Dabrowski, J., Luba, M., Moszynski, M. and Stepnowski, A., 2008, May. GIS for remote sensing, analysis and visualisation of marine pollution and other marine ecosystem components. In Information Technology, 2008. IT 2008. 1st International Conference on (pp. 1-4). IEEE. DOI: 10.1109/ INFTECH.2008.4621628

4. Kulawiak, M. and Kulawiak, M., 2017. Application of Web-GIS for Dissemination and 3D Visualization of Large-Volume LiDAR Data. In *The Rise of Big Spatial Data* (pp. 1-12). Springer International Publishing. DOI: 10.1007/978-3-319-45123-7_1

5. Campos R., Garcia R., Nicosevici T., Surface reconstruction methods for the recovery of 3D models from underwater interest areas, OCEANS, 2011 IEEE - Spain, 2011.

6. Caris HIPS and SIPS, http://www.caris.com/products/hips-sips/ (accessed on 03.11.2017)

7. Cheng L., Tong L., Chen Y., Zhang W., Shan J., Liu Y., Li M., Integration of LiDAR data and optical multi-view images for 3D reconstruction of building roofs. Optics and Lasers in Engineering, 51(4), 2013, pp. 493-502.

8. Cohen, D. and Gotsman, C., 1994. Photorealistic terrain imaging and flight simulation. IEEE Computer Graphics and Applications, 14(2), pp.10-12.

9. Henn A., Gröger G., Stroh V., Plümer L., Model driven reconstruction of roofs from sparse LIDAR point clouds. ISPRS Journal of photogrammetry and remote sensing, 76, 2013, pp. 17-29.

10. Hurtós N., Cufí X., Salvi J., Calibration of optical camera coupled to acoustic multibeam for underwater 3D scene reconstruction, OCEANS 2010 IEEE-Sydney, 2010, pp. 1-7.

11. Kada M., McKinley L., 3D building reconstruction from LiDAR based on a cell decomposition approach. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 38, W4, 2009.

12. Kazhdan M., Bolitho M., Hoppe H., Poisson Surface Reconstruction, Eurographics Symposium on Geometry Processing, 2006, pp. 61-70.

13. Kim K., Shan J., Building roof modeling from airborne laser scanning data based on level set approach. ISPRS Journal of Photogrammetry and Remote Sensing, 66(4), 2011, pp. 484-497.

14. Kulawiak M., Łubniewski Z., Reconstruction Methods for 3D Underwater Objects Using Point Cloud Data, Hydroacoustics vol. 18, 2015, pp. 95-102.

15. Kulawiak M., Łubniewski Z., Processing of LiDAR and Multibeam Sonar Point Cloud Data for 3D Surface and Object Shape Reconstruction, Baltic Geodetic Congress (BGC Geomatics), Gdańsk, 2016. DOI: 10.1109/BGC. Geomatics.2016.41

16. Lu D., Li H., Wei Y., Shen T., An Improved Merging Algorithm for Delaunay Meshing on 3D Visualization Multibeam Bathymetric Data, Information and Automation (ICIA), 2010 IEEE International Conference on. IEEE, 2010, pp. 1171-1176.

17. Lu Y., Oshima M., On the 3-D Reconstruction of Seabed Using Multiple Sidescan Sonar Images, IAPR Workshop on Machine Vision Applications, 2002, Nara, Japan.

18. Nikic D., Wu J., Pauca P., Plemmons R., Zhang Q., A novel approach to environment reconstruction in lidar and hsi datasets, Advanced Maui Optical and Space Surveillance Technologies Conference Vol. 1, 2012, p. 81.

19. QINSy Knowledge Base, https://confluence.qps.nl/display/KBE/QINSy+Knowledge+Base (accessed on 10.02.2017)

20. Rottensteiner F., Automatic generation of high-quality building models from lidar data, IEEE Computer Graphics and Applications 23.6, 2003, pp. 42-50.

21. Seafloor Information System SIS Operator Manual, Release 3.6, Kongsberg Maritime AS, 2009

22. Tse R. O. C., Gold C. M., Kidner D. B., Building reconstruction using LIDAR data, Proceedings 4th ISPRS Workshop on Dynamic and Multi-dimensional GIS, 2005.

23. Tsai V. J. D., Delaunay triangulations in TIN creation: an overview and a linear-time algorithm, International Journal of Geographical Information Systems, vol. 7 iss. 6, 1993, pp. 501-524, DOI 10.1080/02693799308901979.

CONTACT WITH THE AUTHORS

Marek Kulawiak
e-mail: marek.kulawiak@eti.pg.gda.pl

Gdańsk University of Technology
Faculty of Electronics,
Telecommunications and Informatics,
Department of Geoinformatics
Narutowicza 11/12, 80-233 Gdansk
Poland

Zbigniew Łubniewski
e-mail: lubniew@eti.pg.gda.pl

Gdańsk University of Technology
Faculty of Electronics,
Telecommunications and Informatics,
Department of Geoinformatics
Narutowicza 11/12, 80-233 Gdansk
Poland