

Generation of a Generic tetrapod model from photogrammetric point clouds of breakwaters

N. Plantier^a, A.Li^b, S.Travadel^c

a. Mines Paris PSL - nicolas.plantier@etu.minesparis.psl.eu

b. Mines Paris PSL - antoine.li@etu.minesparis.psl.eu

c. C.R.C. - Centre de recherche sur les Risques et les Crises - sebastien.travadel@minesparis.psl.eu

Keywords : Tetrapods; wave-dissipating blocks; point cloud segmentation; watershed algorithm; point cloud registration; Iterative Closest Points algorithm (ICP)

Abstract :

Industrials start to substitute the manual inspection of wavebreakers by point cloud acquisition. However, the analysis of this data is not mature yet. Having the knowledge of the generic model of tetrapods constituting a wavebreaker is useful for modeling and the monitoring of its structure. However, this model is rarely known in real world situations and requires invasive methods to extract it. In this paper, we introduce a non invasive method that reconstructs the generic model of tetrapods constituting a wavebreaker from photogrammetric recordings only. Our algorithm can be decomposed into two main steps : 3D segmentation of portions of tetrapods and reconstruction of the generic model using point cloud registration.

Source Code :

The Python source code for this algorithm is available on the following github repository <https://github.com/nicolasplantier/Find-generic-model-tetrapod>. Compilation and usage instruction are included in the README.txt file of the archive.

1 Introduction

The use of tetrapod-based wavebreakers is becoming increasingly widespread, and maintaining the structural integrity of these wavebreakers is a matter of great interest. Traditionally, the structural analysis of wavebreakers and the identification of potential vulnerabilities has been carried out through visual inspection. However, the acquisition of digital data for wavebreakers through photogrammetric or bathymetric technologies offers a less burdensome technique for monitoring. The ultimate goal of this research is to create a digital twin of the wavebreaker, where each tetrapod is clearly identified and tracked. Having access to this digital twin allows municipalities to detect with centimetre-level precision very small deformations or structural movements. While some companies have made progress in this area and there are scattered examples of success in the literature, they often rely on neural networks that operate opaquely and are prone to a certain error rate.[6] Additionally, such approaches typically require prior knowledge of the tetrapod's generic mesh for the reconstruction of the digital twin. In this article, we present a novel approach that focuses on extracting a generic model of tetrapod from point cloud data as a preliminary step towards future reconstruction.

Compared to the few algorithms that already exist, this algorithm is highly interpretable. At each step of the algorithm, results can be visualized on the associated point clouds.

2 Dataset

Numerous techniques are used to obtain point clouds of tetrapods both above and below the sea. Photogrammetric methods are typically used for the emergent areas of wave dissipating blocks in order to generate point clouds of very high quality. With this method, one tetrapod is typically represented by 250,000 points. Conversely, data beneath the sea is obtained using sonars. This method suffers from significant occlusions due to data being acquired by a boat on the surface of the sea, resulting in lower quality data. Employing this technique, a tetrapod is represented by approximately 3,000 points.

As such, the algorithm only uses UAV photogrammetric data acquired in Ajaccio to infer the generic model due to its superior quality. The two datasets used gather approximately 200 tetrapods, with many occlusions or damages. Furthermore, sonar data was used solely for detecting subaqueous displacement.

To minimize the size of the dataset, the two point clouds were divided into patches, each containing between 5 and 10 tetrapods. The next section presents the segmentation algorithm that was applied to every patch in order to obtain many instances of the same generic model.

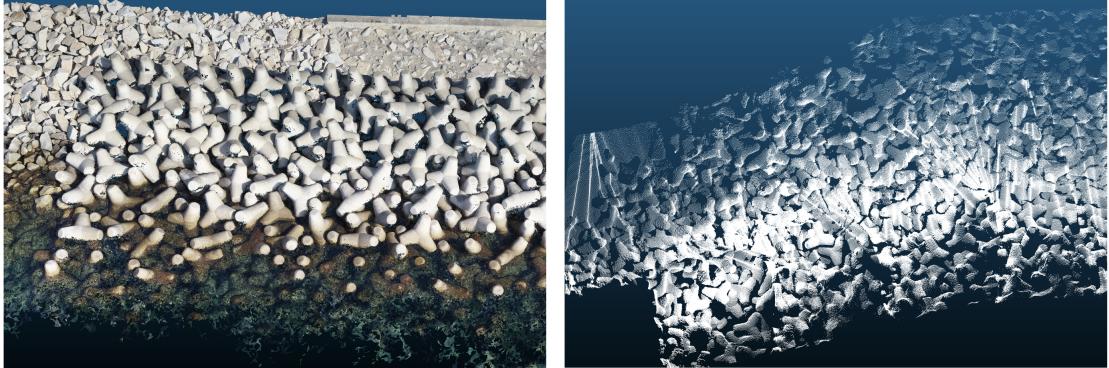


Figure 1: UAV Photogrammetric data above the sea (left) and sonar data under the sea (right), Ajaccio 2019

3 Instance segmentation

In order to obtain the generic tetrapod model, inferred from a sample of instances of the same model with damage and occlusions, it is necessary to segment the original point cloud. This segmentation is based on the computation of a flatness measure on the set of patches, a preliminary step to the application of the watershed algorithm for the final segmentation.

3.1 Flatness Measure

In all the patches, tetrapods have in common the regularity of their surface. On the other hand : the sea, frontiers between tetrapods and other objects (like rocks, sand etc...) do not share this feature.

Let m be the number of points in a patch and let $(x_i)_{i \in I_m}$ be the point cloud P of this patch, where $I_m = [1, m]$ and $x_i \in \mathbb{R}^3$. Each patch is splitted into K voxels where K is defined such as the volume of each voxel is approximately 6cm^3 .

Then :

$$P = \bigcup_{i \in I_m} x_i = \bigcup_{K_i \in P} K_i = \bigcup_{K_i \in P} \bigcup_{x_j \in K_i} x_j$$

We define the flatness measure for each voxel in all patches as :

$$\forall i \in I_K, \mu(K_i) = \frac{\sum_{x_j \in K_i} \alpha(d(x_j, p_i)) \times d(x_j, p_i)}{\sum_{x_j \in K_i} \alpha(d(x_j, p_i))}$$

Where $d(x_j, d_i)$ is the euclidian distance between a point x_j and its corresponding regression plane p_i . And where α is a function that grows with the distance between a point and the regression plane p_i of the patch number i . Adding this weight is necessary in order to put a higher flatness measure for patches which contain a few points far from the regression plane and thus unlikely to belong to the surface of a tetrapod.

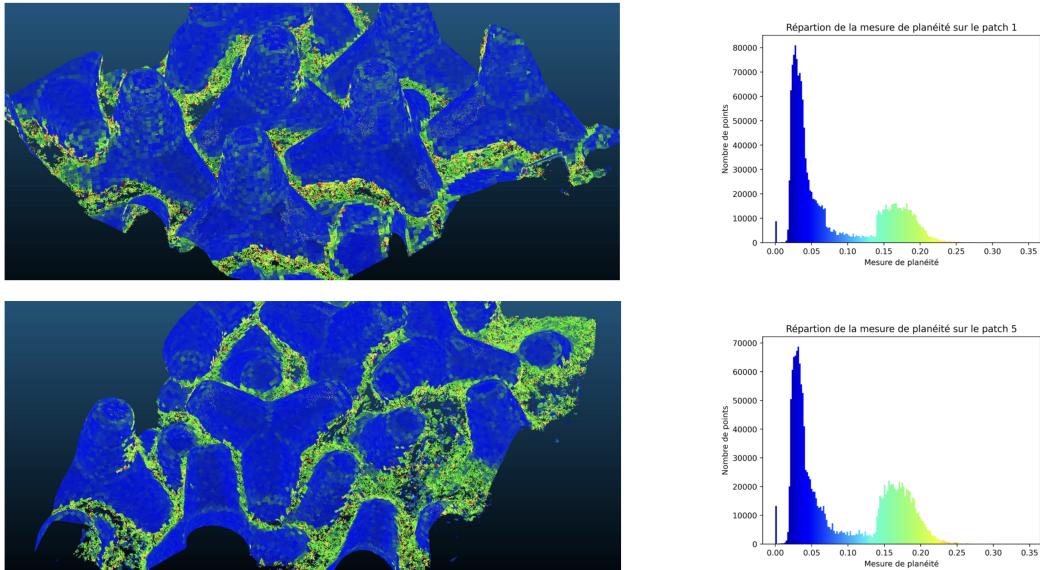


Figure 2: Flatness measure for 2 different patches

In figure 2, we represented the histograms of flatness measures for two different patches. Two distinct behaviors are primarily observed : the points that belong to a tretrapod (blue section) and those located on the area between two tetrapods (green and yellow sections).

3.2 Segmentation via watershed algorithm

The flatness measure is subsequently transformed into 2D images, wherein regions of darkness generally correspond to tretrapod, while brighter regions do not. These 2D images correspond to the projection of all the points contained in every patches on the (x,y) plane. The color of each pixel is determined by the corresponding flatness measure. The pixel size was chosen in order to correspond approximately to the previous voxel size. The loss of information due to this projection is however limited because of the occlusions.

3.2.1 Watershed algorithm : general principle

The watershed algorithm is a commonly used algorithm for 2D image segmentation in traffic monitoring, hydrology or medical image analysis. It was first introduced by S. Beucher [2] and, even if many progresses were made since then, the general principle of the algorithm remains similar and can be described as follow. [5]

When applied to a grayscale image, the watershed algorithm interprets the image as a topographical map, where the pixel intensities represent the elevation of the corresponding points on the map. In the

context of tetrapod detection, the grayscale image is a 2D representation of the patch's flatness measure, where low values correspond to tetrapods and high values correspond to regions in between them. The approach for identifying tetrapods involves identifying the local minima on the map and gradually flooding these basins until the water from neighboring basins meets, creating 'watershed lines'. These lines are then used to define the boundaries between the different tetrapods.

The watershed algorithm used in practice is as follows :

1. Pre-processing : To improve the segmentation quality, a local gradient and a smoothing are applied to the image. In figure 3 (left), we have illustrated this pre-processing step.
2. Markers labelling : This step aims to label the objects to be segmented. Initially, the local minima are identified as the pixels where the local gradient is below a chosen threshold. Then, the connected components of local minima are labeled, and these markers act as catchment basins i.e. starting points for the flooding process that defines the objects. Results can be visualized in figure 3 (right).
3. Labeling of the remaining pixels : We use a hierarchical queue (HQ) process. It is a dynamic process that aims at ordering the pixels to operate the 'flooding'. It sets a priority level for each classified pixel depending on their elevation and attributes a label accordingly on their neighbors.

The application of the watershed algorithm may present practical challenges, specifically regarding the selection of suitable pre-processing methods or the establishment of marker definitions. The choices made are detailed in the next subsection.

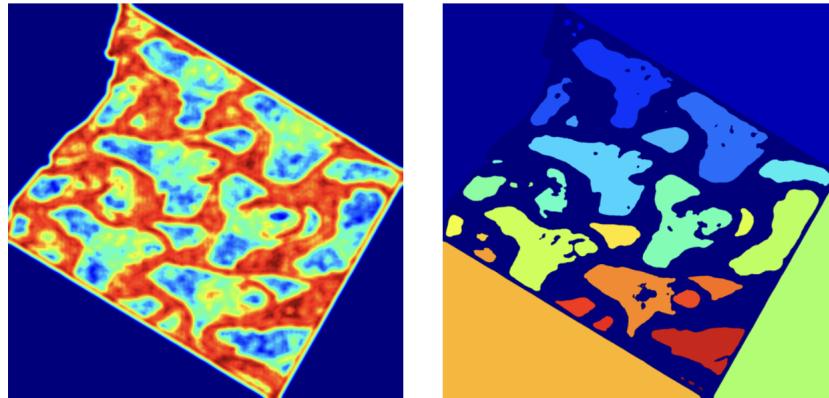


Figure 3: Pre-processing : gradient (left) and markers (right)

3.2.2 Pre-processing and parametrization

Our purpose is to reconstruct a tetrapod model using partial tetrapod point clouds. In this context, smaller samples are preferred over larger ones, as long as significant portions of the tetrapod are captured to enable reconstruction. Oversegmentation is a common bias of watershed algorithms but for this matter it is less problematic than sub-segmentation.

The choice of our pre-processing has been done accordingly. An inverse Gaussian gradient was used and the parameters were adjusted to increase blurring, which helped to create wide boundaries between tetrapods. Averaging was then applied to smoothen the image and enhance this effect. To initialize markers, a criterion was defined, which involved setting the local minima threshold based on the size of the fifth biggest component, as typically, there are around 5 exploitable tetrapods on a patch. The resulting array contained multiple components, but the background was removed and only the largest components were selected.

3.2.3 Results

The watershed algorithm provides a 2D segmentation that is satisfying regarding our goal of having tetrapod parts that do not exceed the real tetrapod. We also set a threshold to only select the bigger objects that represent significant portions of tetrapods.

In order to obtain a 3D instance segmentation, it is defined that points whose projection onto the x,y plane is considered to belong to a tetrapod are themselves considered to belong to a tetrapod.

We obtain satisfying tetrapod portions despite having some parasite portions on the boundaries. Results are represented in figure 4.

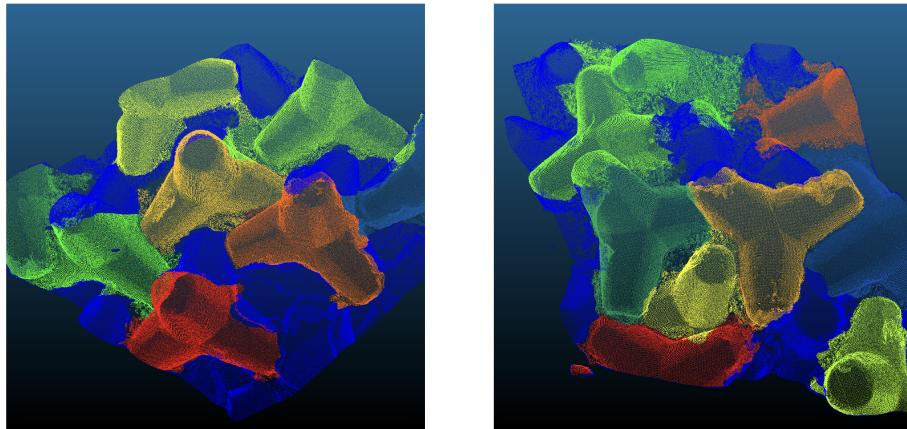


Figure 4: Instance segmentation for 2 patches after applying te watershed algorithm

4 Inferring the generic model

4.1 General methodology for identifying points belonging to the main axes.

Having access to various instances of the generic model, it becomes necessary to identify and locate the axes of the tetrapod. As a matter of fact, the identification of the generic model from which these instances originate requires a thorough understanding of the various tetrapods, as will be demonstrated later on.

Each axis of the same tetrapod can be characterized by two parameters $(a, b) \in (\mathbb{R}^3, \mathbb{R}^3)$, where a represents the directional vector of the axis and b represents its center of gravity; therefore, b is located at the center of the corresponding tetrapod's foot. The algorithm for identifying these two parameters can be summarized in the following steps :

1. Identification of the points belonging to one of the tetrapod's axes
2. Application of the Iterative Hough Transform algorithm to identify the tetrapod's axes

4.1.1 Identification of the points belonging to the tetrapod's axes

The central point of this method consists in noticing that, for a tetrapod foot presenting rotational invariance around its axis, the incoming normals of the surface intersect on a single straight line: the axis of the tetrapod's foot.

Firstly, the tetrapod's point cloud is divided into 10^6 voxels. For each of these voxels, the regression plane is computed along with the normal line to this plane passing through the center of gravity of the points in the voxel. However, due to the finite point density and non-zero standard deviation around the regression plane, the normals intersect in the vicinity of the real axis. Therefore, a probability map of voxels belonging to the axes is established by summing the number of axis intersections on each of the tetrapod's voxels.

The following notations will be used :

- $v_{i,j,k}$ for $i, j, k \in \llbracket 1, 100 \rrbracket$ as the 10^6 voxels of the tetrapod, where the i-coordinate corresponds to the x-coordinate, j corresponds to the y-coordinate, and k corresponds to the z-coordinate.
- $n_{i,j,k}$ for $i, j, k \in \llbracket 1, 100 \rrbracket$ as the 10^6 normals for each voxel of the tetrapod, with the convention that $n_{i,j,k} = 0$ if the voxel contains 2 points or fewer.
- $S_{i,j,k}$ for $i, j, k \in \llbracket 1, 100 \rrbracket$ as the number of normal crossings from other voxels that occur in the voxel i, j, k .
- $p_{i,j,k}$ for $i, j, k \in \llbracket 1, 100 \rrbracket$ as the probability for the voxel to contain a point from the axis.
- $x_{i,j,k}$ for $i, j, k \in \llbracket 1, 100 \rrbracket$ as the points in \mathbb{R}^3 at the center of the voxel i, j, k .

Additionally, the following relationship holds for $i, j, k \in \llbracket 1, 100 \rrbracket$ and S_{max} as the maximum of the $S_{i,j,k}$:

$$p_{i,j,k} = \frac{S_{i,j,k}}{S_{max}}$$

For all heights, only voxels with a sufficient probability of containing a point from the axis are kept. Formally, denoting V as the set of retained voxels and V_k as the retained voxels at altitude k , we have:

$$V = \bigcup_{k \in \llbracket 1, 100 \rrbracket} V_k$$

$$V_k = \{v_{i,j,k} | p_{i,j,k} > 0.8 \times p_k \text{ et } S_{i,j,k} > 10 \text{ pour } i, j \in \llbracket 1, 100 \rrbracket\}$$

where :

$$p_k = \max(\{p_{i,j,k} | i, j \in \llbracket 1, 100 \rrbracket\})$$

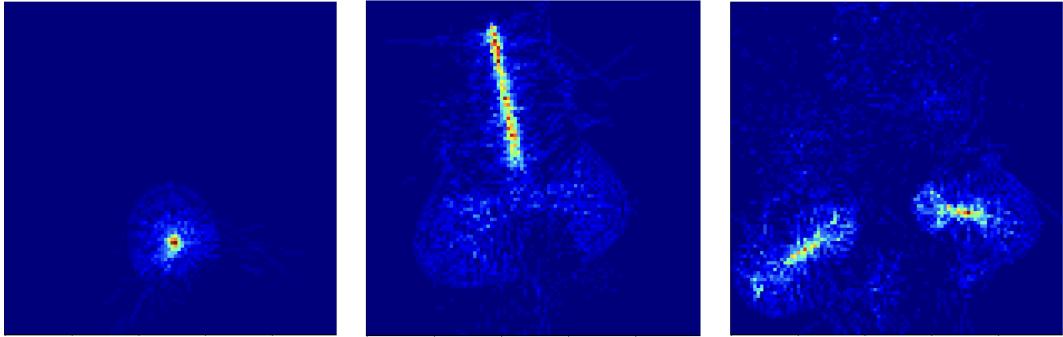


Figure 5: Probability of point presence on tetrapod axes at different heights

In Figure 5, high probabilities can be observed on all four axes of the tetrapod. The leftmost photograph helps identify the first central axis, while the following two photographs identify three additional axes, each forming a 120-degree angle with the others.

Let T be the set of triplets (i, j, k) of the remaining voxels and P_{axe} be the set of the remaining points. Then, we have:

$$P_{\text{axe}} = \bigcup_{t \in T} x_t$$

However, preserving only the voxels that exceed a certain probability threshold does not account for the original probability dispersion seen in Figure 5. To address this issue, we superimpose the centers of the voxels representing the highest probabilities multiple times. More specifically:

$$P_{\text{axe}} = \bigcup_{t \in T} X_t$$

where, with for all $l \in \mathbb{N}$, $x_{t,l} = x_t$:

$$X_t = \{x_{t,l} \mid l \in [\![1, p_t * S_{\max}]\!] \}$$

Results for two distinct tetrapods - having respectively 4 and 3 feet - can be visualized in figure 6.

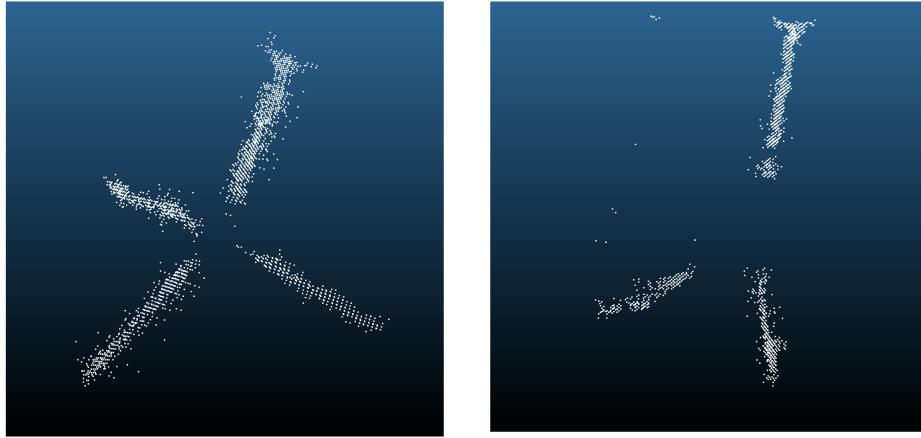


Figure 6: Remaining points to find the axes on two distinct tetrapods

4.1.2 Detection of the axes by Iterative Hough Transform

Iterative Hough transform allows us to detect the main axes in the point clouds from the previous scatter points. This algorithm was first implemented in 2017 [3] and as demonstrated by the authors it is not very sensitive to outliers. The general principle of the method can be described by the following steps:

1. The parameter space is discretized for all lines passing through the point cloud. Each element of this parameter space is a pair $(b_i, a_i) \in (\mathbb{R}^3, \mathbb{R}^3)$ representing a point belonging to the line and its direction.
2. An accumulation array is established: each point is associated with one or several lines to which it may belong. The elements of the parameter space with the highest values in the accumulation array correspond to lines that are most likely to contain a large number of points.
3. The line with the highest value in the accumulation array is retained.
4. For a given line, only points within a certain distance threshold are kept. From this set of points, the line that passes closest to these points is determined by least squares method.
5. These points are removed from the accumulation array, and steps 3 and 4 are repeated until the element with the highest value in the accumulation array has too few associated points.

The Iterative Hough transform algorithm for line detection is very efficient but often detects multiple lines on the same axis. Therefore, it is necessary to introduce a distance threshold in the parameter space to properly detect lines corresponding to distinct axes. The parameter pairs resulting from the Hough transform algorithm are sorted in ascending order of the number of points. For all lines whose distance in the parameter space is below a threshold set empirically, only the line with the largest number of points from the initial point cloud is retained. This threshold criterion is particularly effective in the specific context of detecting the axes of a tetrapod, as these axes form angles of 120 degrees with each other and their respective centers of gravity are regularly spaced.

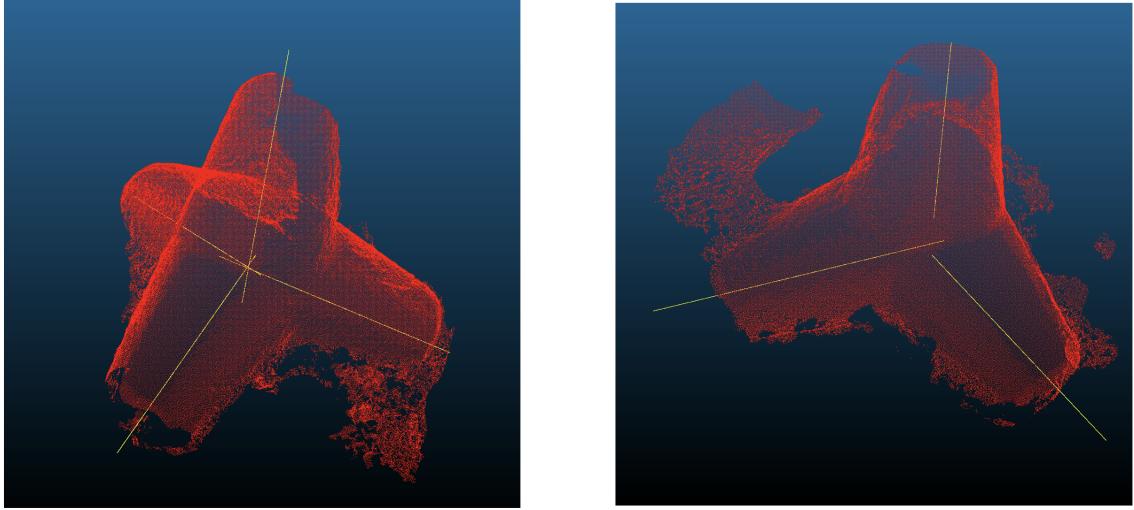


Figure 7: Axes identified with Iterative Hough Transform

4.2 Location of the feet on the tetrapod

The first major use of identifying the axes of the tetrapod (center of gravity and direction) is to roughly determine the location of its feet. The projection of the tetrapod's point cloud onto the plane perpendicular to an axis allows us to distinguish the boundary of a foot. Assuming circular symmetry, the boundary is characterized on this plane by a high density of points, as illustrated in Figure 8 (left). In figure 8 (right), we also represented the plot of the point density as a function of the euclidian distance (in \mathbb{R}^2) to the center of gravity of the foot. More specifically, we represented the following function, where L is the distance between the center of the tetrapod's foot and the surface of the tetrapod : $L = \max_i |x_i - a|$

$$\begin{aligned} d &: [0, L] \longrightarrow \mathbb{R} \\ r &\longmapsto \frac{N(r, r + \Delta r)}{\Delta r} \times \frac{1}{r} \end{aligned}$$

where r represents the distance from the axis and Δr represents the thickness of the ring over which $N(r, r + \Delta r)$ points are counted.

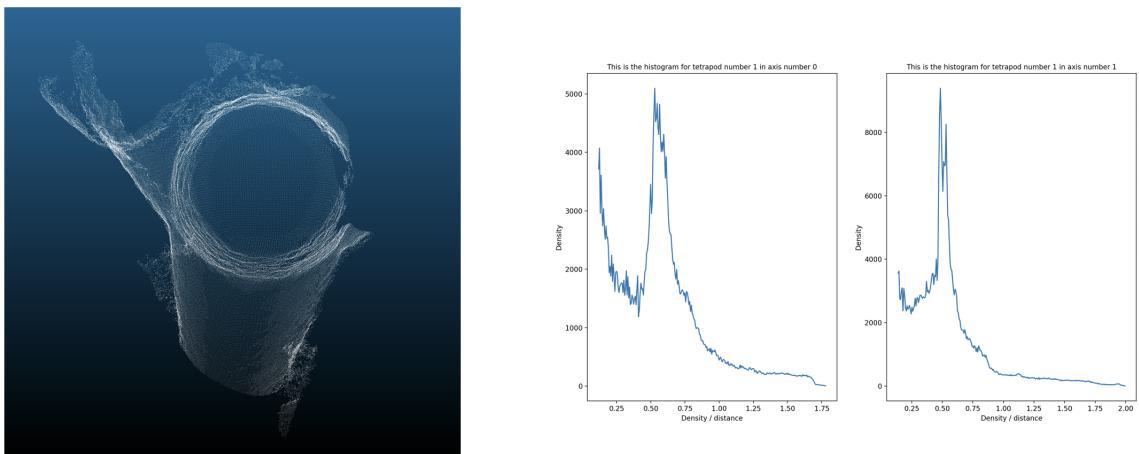


Figure 8: View of a tetrapod in the direction of the main axis (left) and graph of the point density as a function of distance from the center of gravity for two different tetrapods (right)

In figure 8 (right), the point density as a function of the distance from the center of gravity of a tetrapod's foot is depicted. The discontinuity in the derivative of this curve after the global maximum enables us to infer a sudden change in point density at the end of the boundary. Thus, we identify all the feet of the tetrapod, as illustrated in Figure 9.

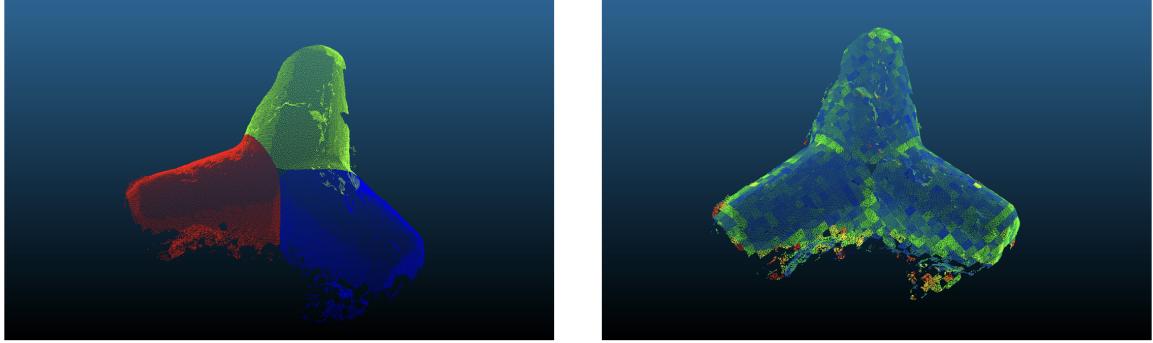


Figure 9: Feet of a tetrapod (left) and associated flatness measure (right)

Identifying the feet of all tetrapods enables the identification of outliers through two methods:

1. Points that do not belong to any foot are considered as outliers.
2. Points with a flatness measure exceeding a certain threshold are deemed not to belong to the tetrapod's surface and are therefore considered as outliers. Figure 9 (right) illustrates the flatness measure on the points of the feet.

4.3 Inferring the generic model

It is now possible to inferate the generic model of tetrapod from the different instances. Two essential steps can be identified.

1. Global registration is carried out in order to align the different tetrapod models with each other.
2. Removal of remaining outlier values is performed on the global model by means of density measure.

4.3.1 Global registration

In order to obtain the generic model of tetrapods we proceed as follows:

1. Initialization with the 4-axis model that has the largest number of points.
2. Each additional model is then overlaid on this model one by one through global registration.
3. The final tetrapod corresponds to the overlay of all instances previously obtained.

The global registration is performed by the Iterative Closest Point (ICP) algorithm [1]. Let P be the set of points of cardinal N_p to be aligned with the set of points X of cardinal N_x . The algorithm can be described by the following steps :

1. Identify the closest N_p points in X to the N_p points in P_k . We denote these points as Y_k .
2. Find the optimal translation and rotation of P_k that minimize the mean squared error between P_k and Y_k . Denote this as $q_k = (q_R, q_T)^t$.
3. Apply the transformation to the initial point cloud P_k as follows: $P_{k+1} = q(P_k)$.

4. Terminate after a fixed number of iterations.

The main weakness of this algorithm lies in its high sensitivity to outliers. The first step of the algorithm involves finding the closest neighbors of a point cloud on another and overlaying these two point clouds. If there are some outliers in either of the two point clouds, the algorithm may be biased and not converge to the true solution. Therefore, the pre-processing steps (section 4.1 and 4.2) are necessary to converge properly to the correct solution.

4.3.2 Density on the final model to delete outliers

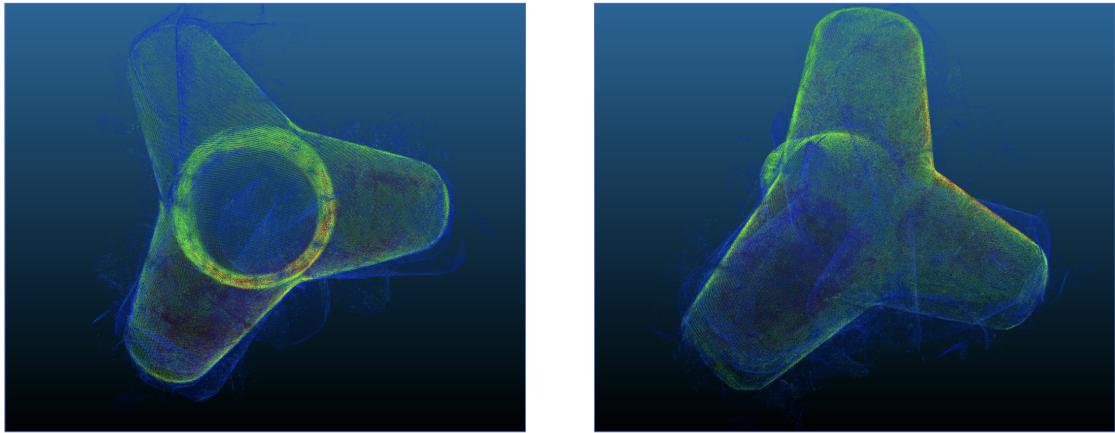


Figure 10: Overlay of tetrapod models. Areas with low point densities are represented in blue, while those with high densities are represented in red.

Figure 10 shows the overlay of the tetrapod models obtained previously. The scalar field represented in Figure 10 corresponds to the point density in the vicinity of each point. This density was obtained using a k-nearest neighbor algorithm. High point density corresponds to an area where a large number of models have overlapped, and thus corresponding to the boundaries of the generic tetrapod model. Conversely, a low-density area (represented in blue in the figure) corresponds to an area where very few models have overlapped. These low-density areas correspond to a global registration that has not converged well, mainly due to the presence of outliers that were not eliminated during the pre-processing steps. The tetrapods that do not overlap correctly with the initial tetrapod will then be uniformly distributed in the space around the tetrapod. They are therefore easily identifiable and can be immediately eliminated since they are located in areas with very low point densities.

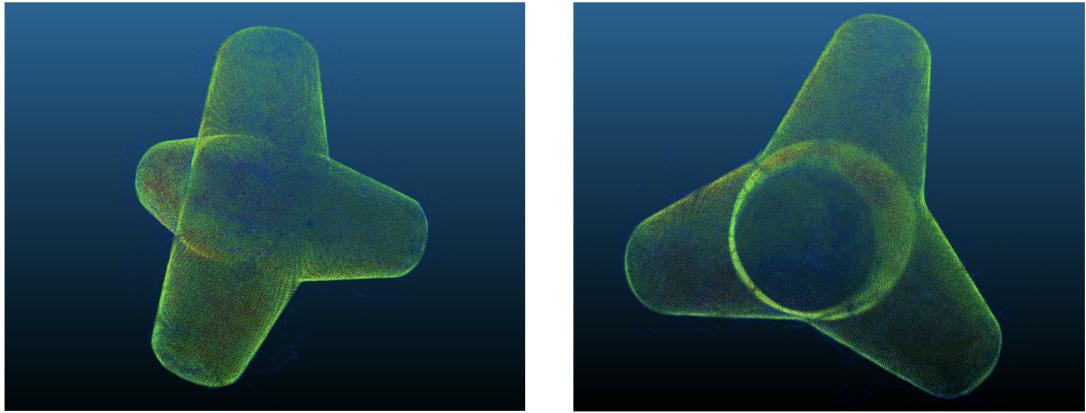


Figure 11: Final tetrapod model

The result after removing outliers is shown in Figure 11.

4.3.3 Quantifying Model Performance with Error Measures

In order to evaluate the algorithm's performance, the focus of this study will be restricted to measuring the error between the final model obtained and the theoretical one. It is important to note that obtaining the theoretical model is impossible as the algorithm aims to approximate it. However, it is possible to select the tetrapod that appears to come closest to the theoretical model from the entire dataset and, in particular, select the best foot. This study will focus on comparing this foot, which is considered to be similar to the theoretical foot, to one of the feet of the generic tetrapod obtained as output of the algorithm.

The error between the points of this foot and those from the ideal tetrapod is solely due to the error in photogrammetric measurement. This error is estimated by cutting the foot in a plane orthogonal to its axis of revolution. Assuming a circular cut, this error is equivalent to the standard deviation of the distance between the points and the axis. (See Figure 12, left)

For the foot of the generic model obtained from the algorithm, the uncertainty is associated with the thickness of the obtained envelope. (See Figure 12, right)

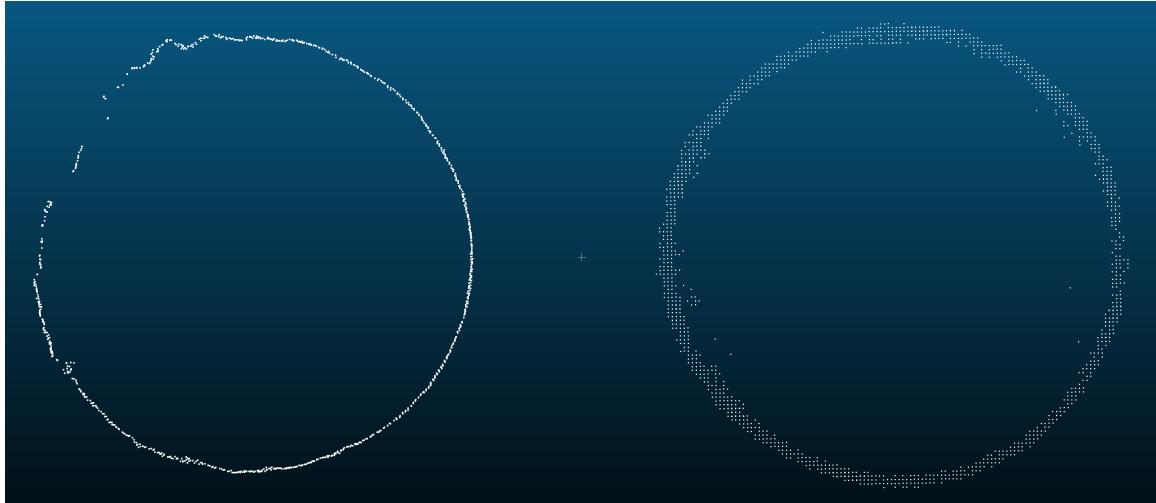


Figure 12: Transverse section of tetrapod feet. Ideal foot considered (left), foot obtained from the algorithm (right)

For a foot with a circular cut, the information on the radius (distance between the axis and the border) as a function of the distance from the center of the tetrapod, is an information that fully characterizes the foot. Such a graph is illustrated in Figure 13.

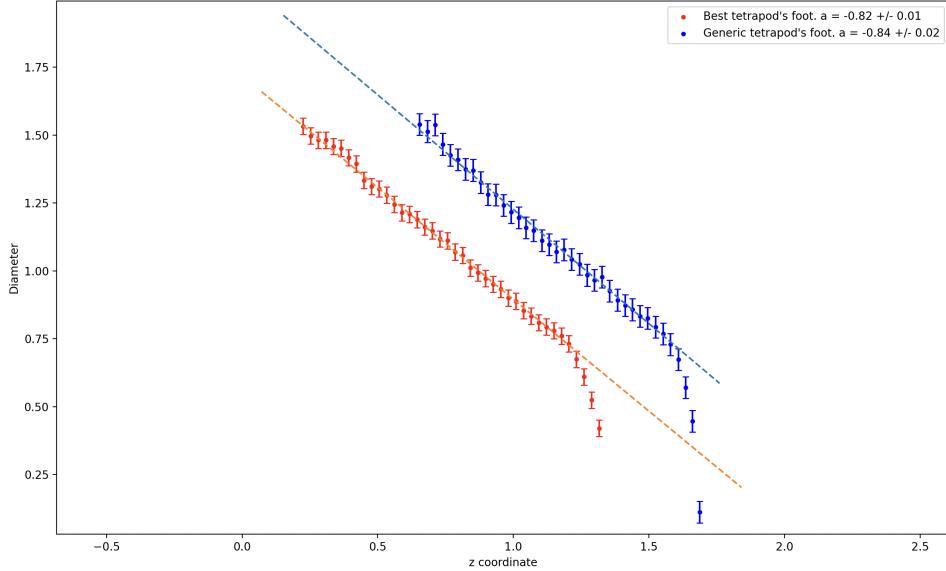


Figure 13: Diameter as a function of the z-coordinate for the ideal tetrapod and the tetrapod obtained from the algorithm

In red, the diameter as a function of its z-coordinate (which increases as we approach the tip of the foot) is depicted. By performing a least squares linear regression for the coefficients and Monte Carlo simulations for the uncertainty on these same coefficients [4], a slope of -0.82 ± 0.01 is obtained for the ideal foot. Similarly, a slope of -0.84 ± 0.02 is obtained for the generic foot. Moreover, on the nonlinear part of the curve, the points from both models follow the same trend.

We also performed a global registration on the two tetrapod feet, and the results are robust. The ideal foot, consisting of 66306 points, was aligned with the foot obtained from the algorithm with an RMSE of 0.00776.

5 Conclusion

With a high-quality dataset of point clouds acquired via photogrammetric methods we successfully managed to extract a generic model of surface tetrapods. The first obstacle was to segment the wavebreaker point cloud into individual tetrapods. This was possible thanks to the regularity of the tetrapods' and the irregularity of the interstices. Combined with a fine tuned watershed algorithm, we could extract samples of tetrapods. It is important to notice that the segmentation is not perfect but of sufficient quality for our purpose. However, those samples were hard to manipulate but given the rounded shape of all tetrapods that are the most commonly used (invariance by rotation around each axis), we were able to extract the axes of all tetrapods. We were then able to reconstruct the model of tetrapod using point cloud registration via the Iterative Closest Point (ICP) algorithm. A point of interest of our method is that it only requires a small number of samples to be able to generate a satisfying model allowing the extraction of a model even for small datasets.

5.1 Prospects

The extraction of a tetrapod model is intended to identify it within a point cloud of such instances structure. Although it is not the object of the present paper, we tried to use our model to locate

tetrapods in a wavebreaker point cloud. We did it on bathymetric data because there are less precise and with a lesser point density. The algorithm applied consists of successive point cloud registration of a tetrapod model over the wavebreaker. Once a tetrapod is identified, it is removed from the point cloud and the process is repeated. However, this algorithm has only been applied on a single patch and needs more development and refinement to be applicable. Figure 14 (left) shows the creation of a digital twin that identifies most of the tetrapods except those that are not visible enough. Figure 14 (right) the overlay of the original point cloud and the positioned models of tetrapods is observable and validates the soundness of our tetrapod model.

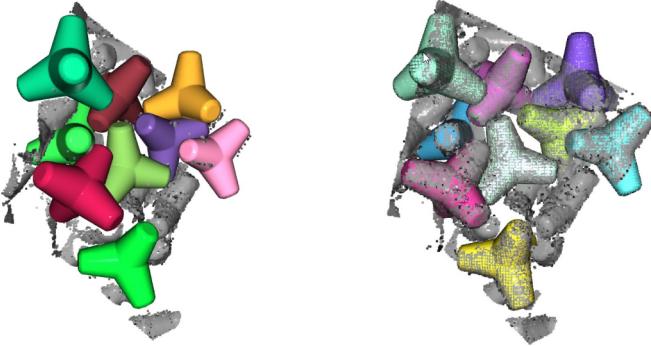


Figure 14: Identification of tetrapod instances on a bathymetric point cloud : reconstruction of the wavebreaker with our tetrapod model (left) and overlay of the original point cloud with the reconstructed wavebreaker (right)

References

- [1] P.J. Besl and Neil D. McKay. “A Method for Registration of 3-D Shapes”. In: *IEEE* (Feb. 1992).
- [2] S. Beucher. “Image segmentation and mathematical morphology”. PhD thesis. École Nationale Supérieure des Mines de Paris, 1990.
- [3] C. Dalitz, T. Schramke, and M. Jeltsch. “Iterative Hough Transform for Line Detection in 3D Point Clouds”. In: *Image Processing On Line* (2017), pp. 184–196.
- [4] Legrand F. *Mesures et incertitudes*. <https://www.f-legrand.fr/pynum/capacites/incertitudes.html>. 2021.
- [5] Roerdink J. and A. Meijster. “The Watershed Transform: Definitions, Algorithms and Parallelization Strategies”. In: *Fundamenta Informaticae* (Jan. 2000).
- [6] Xu Y. et al. “Deep-Learning-Based Three-Dimensional Detection of Individual Wave-Dissipating Blocks from As-Built Point Clouds Measured by UAV Photogrammetry and Multibeam Echo-Sounder”. In: *Remote Sens* (2022).