Advice and Recommendations for Sizing Large Siphonophores with Photogrammetry

Contributors: Joost Daniels, Kakani Katija, and Nicole Kaiser
Bioinspiration lab at Monterey Bay Aquarium Research Institute

****NOTE: files and videos referenced in this paper are only available upon request to authors****

Introduction

What we were trying to achieve and why

Siphonophores are diverse gelatinous soft-bodied creatures which are present from the bottom to the surface of the Pacific Ocean. They play important ecological roles, such as predation for some species. Because of their semi-transparent biological structures, it can be difficult to capture these species in the wild, both from an imagery and specimen collection standpoint. In addition, it also is challenging to accurately make sizing measurements. Several traditional methods for sizing large animals are not effective for large siphonophores. For example, net collection destroys their delicate biological structures and CT imaging requires non-movement. Although laser line scanning and structured light sizing have shown potential for smaller siphonophores and other gelatinous creatures, they cannot scan the full body length of a large siphonophore (Daniels et al. 2023).

Benefits of approach

Photogrammetry has potential benefits:

- Can capture overlapping biological features and piece together a full length body scan of an organism
- Low cost and minimal hardware requirements
- Access to free and open source software
- Non-invasive
- Has been rigorously tested and validated by scientific and general community

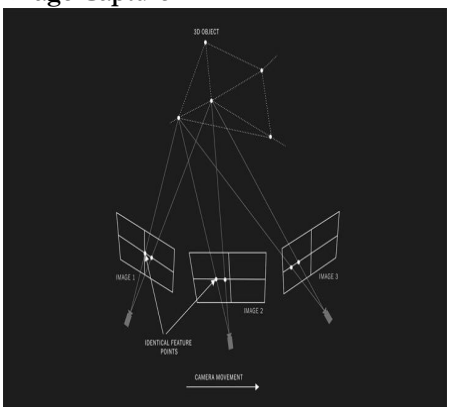
Hence, this project sought to explore photogrammetry as a potential method for sizing large siphonophores.

How do we size big siphonophores in the ocean using photogrammetry?

The objectives of this project were the following:
1. Investigate the pipeline and workflow of traditional photogrammetry on several selected large siphonophore datasets
2. If there seemed to be some potential, develop a baseline photogrammetry pipeline and workflow for our imagery
3. Produce results that could be reproducible and serve as a foundation for biological measurements (ex. Length and width)

Overview of Traditional Photogrammetry Methodology

Image Capture



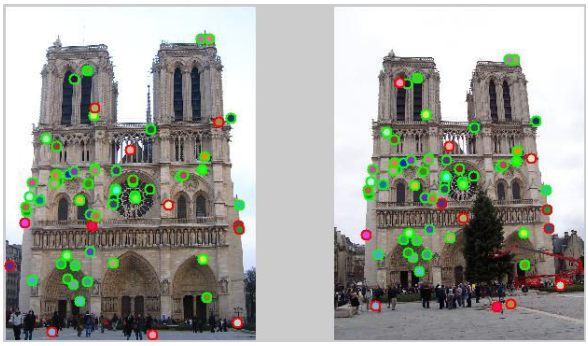
Cohrs et al, R & D (2023)

Typical method for image capture

When creating a 3D model using photogrammetry, photographs should be taken from different points (different heights, different angles) in the scene. This difference in apparent position is called parallax, which allows the photogrammetry software to calculate depth, making it possible to render a 3D model. This results in a 360 degree turnaround of an object of interest.

Pipeline Steps of Photogrammetry

After **image capture**, **features** from the collection of images are **extracted.** Features are essentially linkages between the object and the image, they are characteristic or striking parts of objects (ex. zooids). Group of pixels in images which look the same for different camera positions. **Features are matched** and **images are reconstructed** from the matched features. In this step, there is an association of homologous points based on shape and color discontinuities. Images are matched based on overlap of recurring patterns or features.



Representation of feature extraction: Plene, Github (2023)

Structure from motion is when there is the reconstruction of a 3D point cloud with 2D points. This is achieved by identifying a set of features, or group of pixels, in images and identifying image pairs which hold those features to infer scene structure (3D representation) and the internal calibration of cameras. There is **Dense point cloud generation**, where a set of points in space are created that represent reconstruction of the scene or object. Then, there is the calculation of distance of every pixel to internal cameras that were identified. There is **Mesh construction** from dense point clouds, or the structural geometrical build consisting of triangles and 3D representation of the object. Finally, there is the **texturing** step, which is when materials are laid over the mesh and it resembles how it appears in the original image capture.

See video below for overview of the whole process described here: https://youtu.be/vh_11y-8i_A

Our Pipeline Recommendations

Software and Hardware

The free software that was implemented for photogrammetry use was Meshroom (AliceVision | Photogrammetric Computer Vision Framework). Other photogrammetry softwares such as Agisoft were experienced with but ultimately Meshroom was selected as it was found to be much more time efficient and user friendly for the authors. However, there were inherent challenges as not all of the authors of this paper had access to a high quality GPU hardware in their computer. This prevented the advancement of data past the dense point cloud generation of photogrammetry. Hardware limitations were circumvented by sending semi-completed pipelines to high powered lab computers stationed at MBARI, which resulted in full pipeline completions. Results of object files were visualized in the free 3D creation software Blender (blender.org - Home of the Blender project - Free and Open 3D Creation Software).

Our datasets

The datasets that were utilized for the project were collected from ROVs (remotely operated vehicles) owned and operated by the Monterey Bay Aquarium research institute (MBARI) located in Moss Landing, California. Underwater visual footage of several siphonophore species (*praya* and *mezzanine*) was collected around the Monterey Bay area at around 200-1000m (midwater environment). The camera that was used for recording the siphonophores was an Insite Mini Zeus II camera. Of these videos, two were selected for analysis (see below).

Video 1: https://drive.google.com/file/d/17bZZ3arW98x68f_ofhzi_aHWp2rnNKsA/view?resourcekey

Video 2: https://drive.google.com/file/d/1QMA42yv4fcJI1oc_XzShxWra8aNAxXG-/view?resourcekey

It is important to note that these data sets differed from images traditionally used in photogrammetry in that the subject matter is semi-transparent, there is not a 360 degree view of the siphonophore, and there is movement and particulate matter in the background, which creates noise.

Best image density (seconds apart or spacing)

Video 1 (first dataset) was 24 seconds of an original 6 minute video at 60 frames per second. This led to an image density of 1440 images. Video 2(second dataset) was a total duration of 3 minutes and 42 seconds from an original 6 minute video at 60 frames per second. This led to an image density of 13320.

Maximum total number of images

For video 1, it was found that a maximum subset of a total number of 1349 images (25 % of the original images), was sufficient to produce a mesh and textured result. Originally, a subset of 5000 images was used as the frames per second was increased to see if this was relevant to output, but this did not yield desired results in terms of density and composition. For video 2, a subset of around 400 images was selected. It was found that a maximum total number of 200 images (50 % of total dataset) was sufficient to produce a decent mesh result, although texturing was not always successful. It can be concluded that the number of images does affect final mesh output, mainly because of the reconstruction aspect of the pipeline and the number of final features extracted.

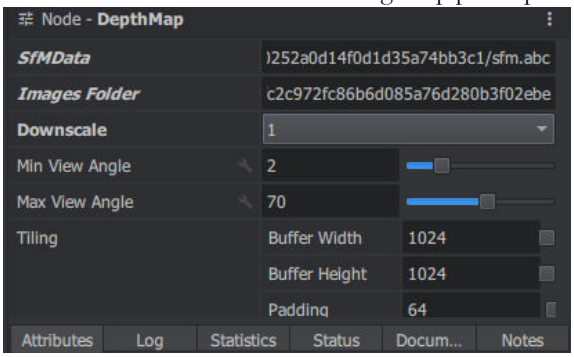
Imaging Culling

Potentially, image culling could be beneficial to improving the reconstruction aspect of the pipeline but it would make the photogrammetry process more tedious and time consuming. In addition, culling too many 'imperfect photos' may lead to poor reconstruction results in itself if there is not enough coverage of the organism. Recommended criteria for guiding imaging culling would be to eliminate photos with noticeably poor resolution, that are 'blurry,' have low contrast between siphonophore and the environment, and a high amount of background interference as this can lead to visibility constraints and impacted reconstruction.

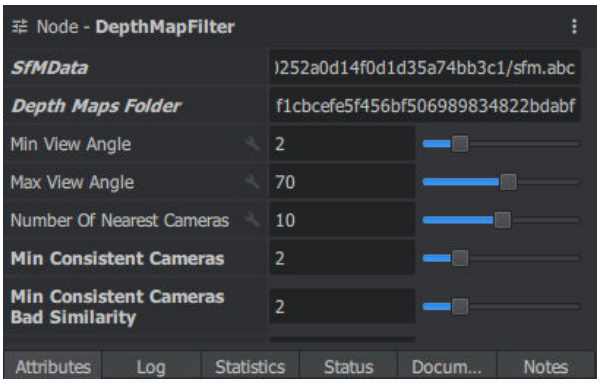
Best pipeline settings and texturing settings

Base pipeline settings for decent 3D Reconstruction that could be used for measurements

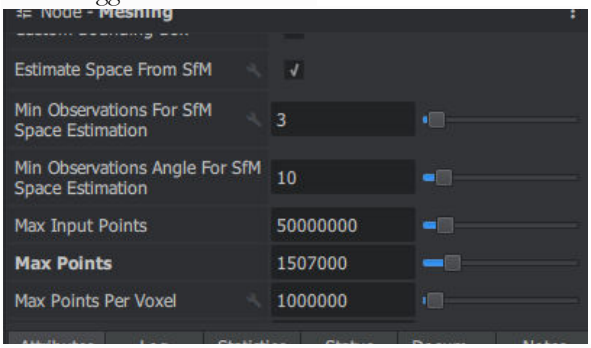
The base pipeline that was determined for the project was entitled "siphonophore-dense_reconstruction_JD_25pc_eldorado." All photogrammetry computations were done in Meshroom (version 20232.0). The base pipeline was modified based on Meshroom's dense reconstruction pipeline settings. This meant focusing on the depthmap, texturing, and meshing steps of the pipeline. The images used in this pipeline were from video 1. The modifications to the original pipeline provided in Meshroom were the following:



In the **DepthMap** node, the **downscale** setting was lowered to 1. Successful settings around downscale allowed the most detail to be preserved in the mesh.



In the **DepthMapFilter** node, the min consistent cameras and min consistent cameras bad similarity were lowered from default to 2, as it helps to reduce complications from blurry photos and previous issues with holes in the meshes, which were struggles that were faced.

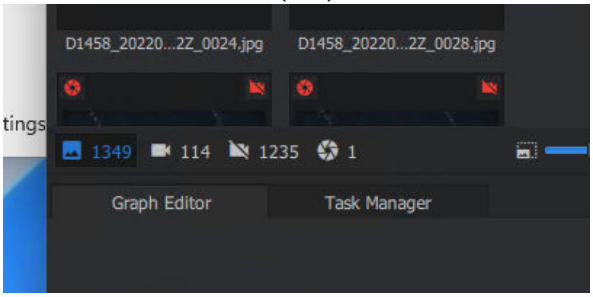


For meshing, the **max points** were augmented and reduced significantly from the default value. This was to increase mesh precision. The max input point value was determined through trial and error, and eventually 50 million was selected as it produced a desirable amount of mesh detail and did not sacrifice the density of the point dense cloud. It also enabled the ability to stay within RAM limitations of the computers being utilized. In the **meshfiltering** node, **filter large triangles factor** was selected and was adjusted to avoid holes and to limit large, irrelevant triangles that might be added to the mesh. **Keep Only the Largest Mesh** was selected to eliminate unconnected fragments from the mesh. In **texturing**, the **downscale** was changed to 1 to improve the final resolution of the mesh.

Results of this Pipeline based on key elements and features of successful photogrammetry

In order to achieve a good result from photogrammetry, it is essential that there is high feature extraction and strong matching, as there needs to be enough features that can be extracted to set up the foundation for meshes. One key parameter to determining these elements is high reconstruction. High reconstruction is criteria for quality image data. If fewer camera angles are reconstructed, it means that there are less 2D points to be transformed into a 3D point cloud that can be used for depth maps, meshing, and texturing. In addition, it is desired to have clean meshes (no holes, distortions, or inaccuracies) and texturing success.

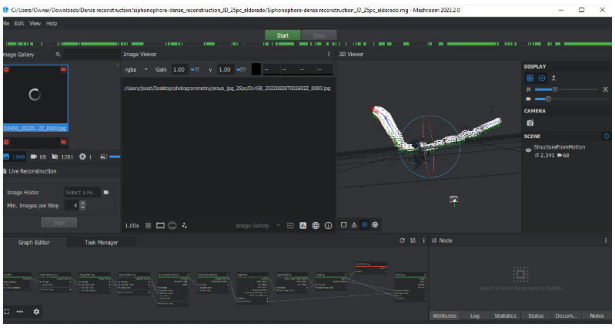
1. **Reconstruction (8 %)**



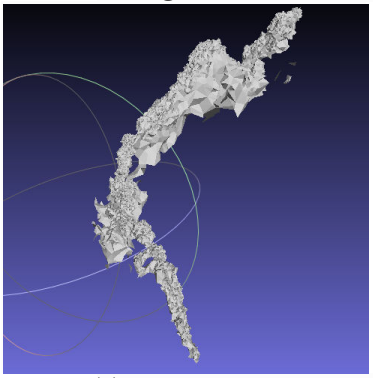
The reconstruction of the images could have been low due to several factors. One was that there seemed to be a disconnect between rendering camera metadata in meshroom. Several times, the software either defaulted to standard camera metadata settings or said that there was missing metadata, and research revealed that this could have an impact

on reconstruction. This could be improved by tracking camera metadata and manually inputting the data such as focal length. However, decent reconstruction results were still achieved so it should not take high priority over other strategies.

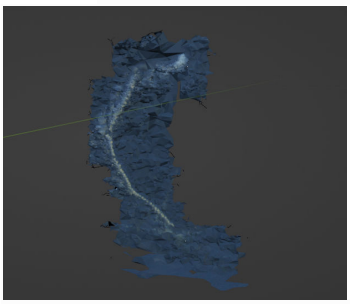
In addition, poor matching can lead to not only limited reconstruction results but failing at the meshing step. For example, in video 1 salps swam in front of the camera and this led to feature mapping with the salps instead of with the good siphonophore data. This can be improved by more specific input image filtering. For example, removing images that have high background or foreground interference that could interfere with the features of the main object of interest, in this case the large-bodied siphonophore.



2. **Mesh: The mesh that was produced had decent resolution and was not distorted. Measurements such as length and width could be made after calibration.**

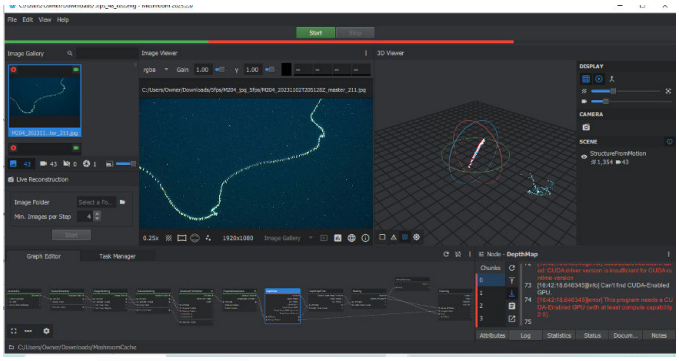


3. **Texturing: Materials were laid on mesh so that visible zooids could be counted and distance calculated between them.**

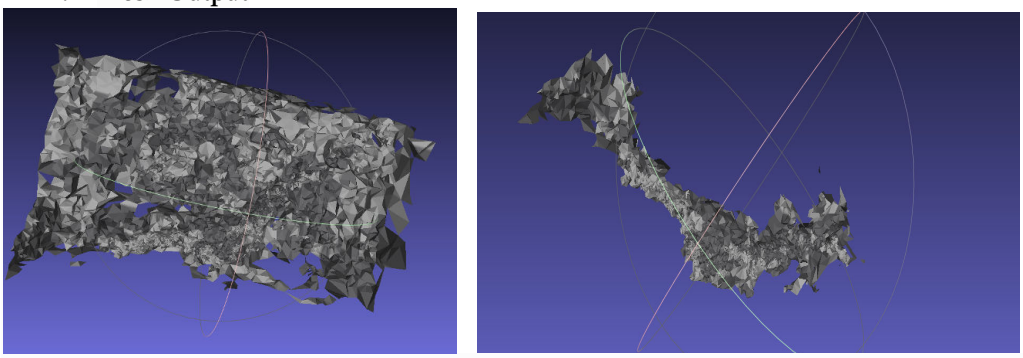


Other results: Default settings but with dataset from video 2

1. **Reconstruction (100 %)**



2. **Mesh Output**



The mesh output is considered to be decent because although there is a lot of noise and interference, the body of the siphonophore has been identified from the background (left is original unedited mesh and right is the mesh edited to include mainly the mesh of the siphonophore with some background). The body is being connected even though there are some gaps. In bad meshes, the siphonophore mesh would be disjointed and the mesh would be broken into parts and outputs that were not coherent.

3. **Texturing: did work for** "siphonophore_dense_5fps_whole_JD_eldorado" and "siphonophore_dense_50pc_25pc_eldorado" pipelines" according to meshroom outputs. The first pipeline contained 400 photos of the video 2 dataset, while the second pipeline contained 71 photos of the video 2 dataset. These pipelines differed in the number of input photos. Photos that were blurry were culled from each of these two pipelines.

Note: * Materials from the textures did not transfer over to the visualization platforms "Meshlab" and "Blender." Next steps should involve optimizing texturing settings for these pipelines as well as aligning outputs with visualization software for textured meshes.

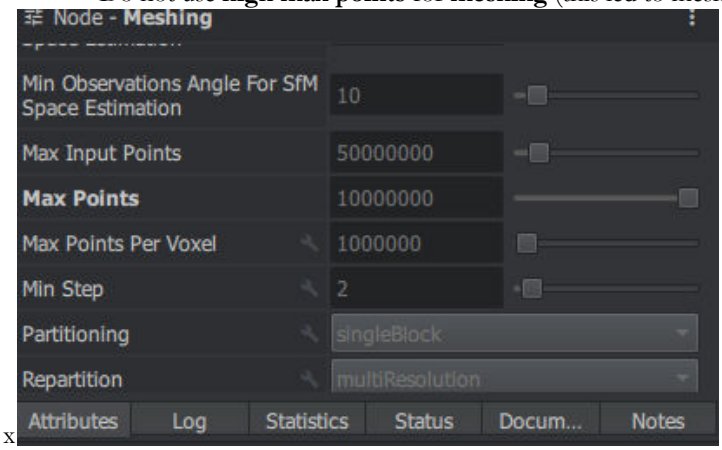
In order to achieve these results, the base pipeline that was determined for the project, "siphonophore-dense_reconstruction_JD_25pc_eldorado," was used as a guide in terms of relevant dense and sparse reconstruction parameters to modify and adjust for optimized reconstruction and mesh results. However, instead of the input photos from video 1, input photos were selected from video 2. A subset of around 43 images yield the above reconstruction results. This showed promise, since the previous video 1 dataset had less optimal reconstruction results. Therefore, the "siphonophore-dense_reconstruction_JD_25pc_eldorado" parameters were utilized for a subset of 398 (around 400) images in video 2 to yield the mesh output depicted above. This meshroom pipeline was entitled "siphonophore_dense_5fps_whole_JD_eldorado." This pipeline produced a more optimal mesh than "siphonophore_dense_50pc_25pc_eldorado," which was similar to the latter except it used only 25 % of the photos from the other pipeline.

Failed Pipelines, Not important or what doesn't work, avoid unless certain situations

Most of the 'failed' pipelines which did not work focused on modifications based on Meshroom's "Sparse Reconstruction Settings" or settings which focused on the front half of the traditional photogrammetry pipeline. These pipelines failed because they either had low reconstruction, distorted meshes with many holes and inaccuracies, or

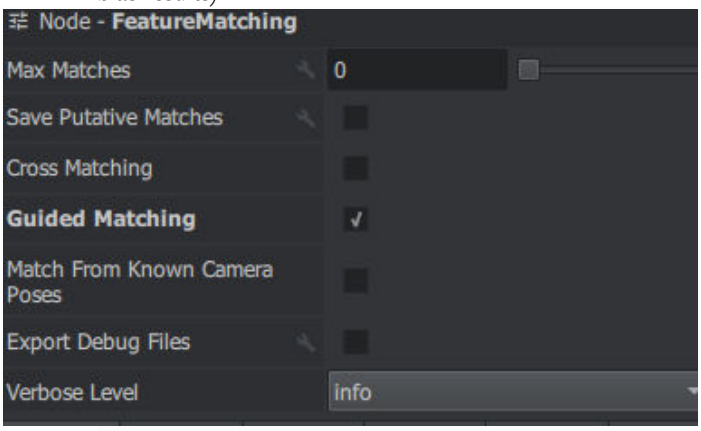
texturing did not work. Therefore, the pipeline settings listed below are what we would recommend be avoided in the future for siphonophore ROV collected images with similar conditions as our video 1.

- Do not use **high max points** for **meshing** (this led to mesh inaccuracies and distortions)

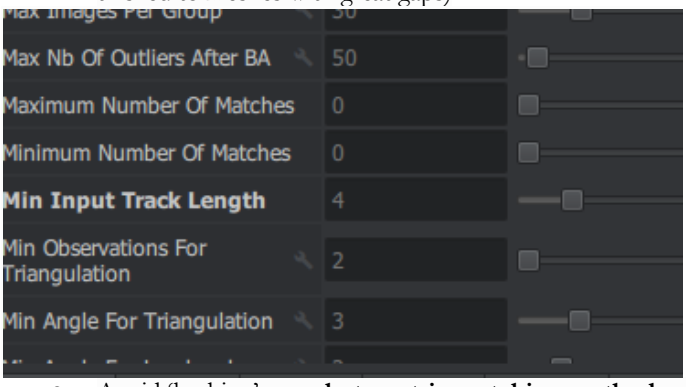


x

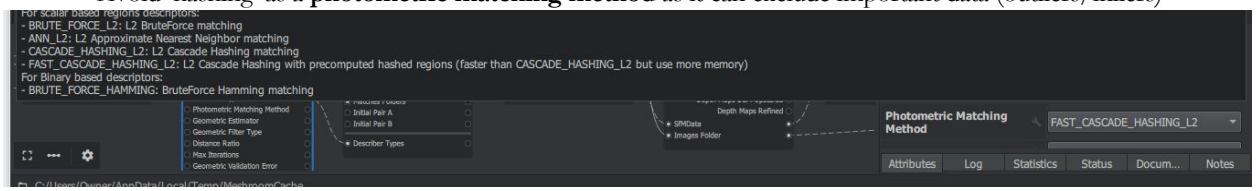
- Do not select **guided matching for feature extraction (**supposed to aid with repetitive structures but this can bias results)



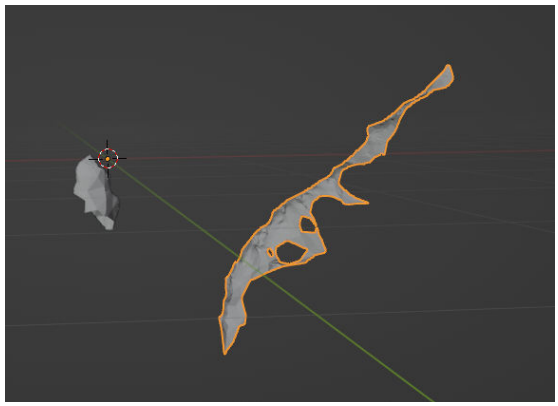
- Do not reduce **MinInputTrackLength** (supposed to keep only most robust matches and remove outliers but this led to meshes with great gaps)



- Avoid 'hashing' as a **photometric matching method** as it can exclude important data (outliers/inliers)



Poor mesh results (see below)



***Important Note about Reconstruction**

It is believed that for the video 1 dataset, reconstruction failed due to background interference and the fact that species of other underwater organisms traveled across the camera and blocked the siphonophore mid-video. However, it is believed that not all video datasets of siphonophores will have this issue. Therefore, some of our lack of success with these pipelines may come down to the reconstruction of the features that were inherent to our image datasets.

Conclusions/Key Takeaways (new intern)

From this project, photogrammetry was able to be achieved successfully under certain conditions despite significant differences with typical photogrammetry approaches. Unique reconstructions for large, soft-bodied underwater organisms were achieved. Due to the fact that this was not a comprehensive look at all possible siphonophore datasets, there is much future direction to be made regarding other key parameters and settings in Meshroom and for utilizing the photogrammetry method. Future interns can use our pipelines as a base but continue to make modifications as deemed necessary for different siphonophore datasets. There were vastly different results depending on when there were environmental changes in the clips, such as lighting and water conditions. These changes led to immediate differences in reconstruction, meshing, and texturing results in Meshroom.

How to build off this work or dataset, Advice and information

Photogrammetry parameters are sensitive to images or the environment, and that an input clip has a large impact on the output. It is vital that future work is focused on capturing effective input images. Dense reconstructions worked best for our datasets, however, for other datasets sparse reconstruction settings may have relevance. Future work should focus on modifications to 'sparse reconstruction' parameters, or steps on the front end of the pipeline, once quality image input has been established. Finally, image calibration and establishing camera metadata can allow for robust measurements to be made, as this project stepped a tad short from generating such results. In order to aid in image capture, ROV footage of siphonophores should strive to increase different camera angles and views of these organisms in the water, even though there are inherent limitations due to where these organisms are located.

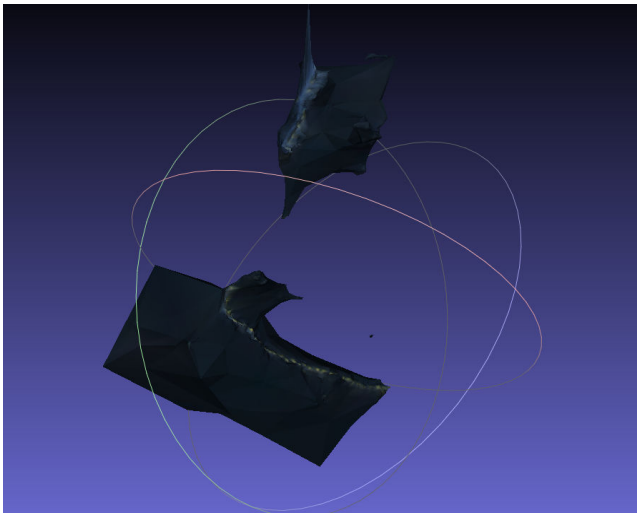
Alternative methods in the future, Neural radiance Fields and 3D Gaussian Splatting

Neural radiance fields is the use of a fully connected neural network that can generate novel views of complex 3D scenes based on a partial set of 2D images. Although this process is computationally expensive and requires knowledge of deep learning techniques, it could be potentially used for sizing large siphonophores. Python and Cuda toolkits are important to compiling the code needed to run these methods. Future direction could explore this method as an alternative to photogrammetry. Neural Radiance fields has its advantages to traditional photogrammetry in that it uses its AI driven ability to generate any angle of a scene, filling in photo gaps, and blending the information of existing photos. Hence, it does not require as many images from every angle as photogrammetry does, and is effective for complex scenes. It is ideal for scenarios with incomplete data and where flexibility in viewpoint generation is required, as with large scale siphonophores.

The data outputs were achieved through a method known as view synthesis. View synthesis involves taking a series of photos that show an object from multiple angles, creating a hemispheric plan of the object, and placing each image in

the appropriate space around the object so as to predict the depth of a series of images that describe the different perspectives of an object. It accurately represents 3D scenes for computing image rendering.

A NeRF essentially optimizes a continuous volumetric scene function. A NeRF is created from each viewpoint through creating a sampled set of 3D points, producing an output set of densities and colors, and accumulating these colors and densities into a 2D image that eventually will be used to render new views of a mesh 3D object. To generate a 3D mesh object file from NeRF, various pipelines and strategies can be used (Ratkotosaona et al. 2023 (NeRFMeshing (m-niemeyer.github.io); Mildenhall et al. 2022 (NeRF (acm.org). However, most methods are optimized for traditional neural radiance field and photogrammetry datasets. A common platform for computing neural radiance fields is Nerfstudio(nerfstudio), and this video goes through the logistics of training a model and producing a textured 3D mesh object similar to the image seen below: How to Make 3D Models from NeRFs using Nerfstudio (youtube.com) Essentially, it is difficult to generate mesh object files from neural radiance fields alone and to do so requires additional computational work and pipelines. However, The 3D object that was created from this data was created with RealityCapture on PC.



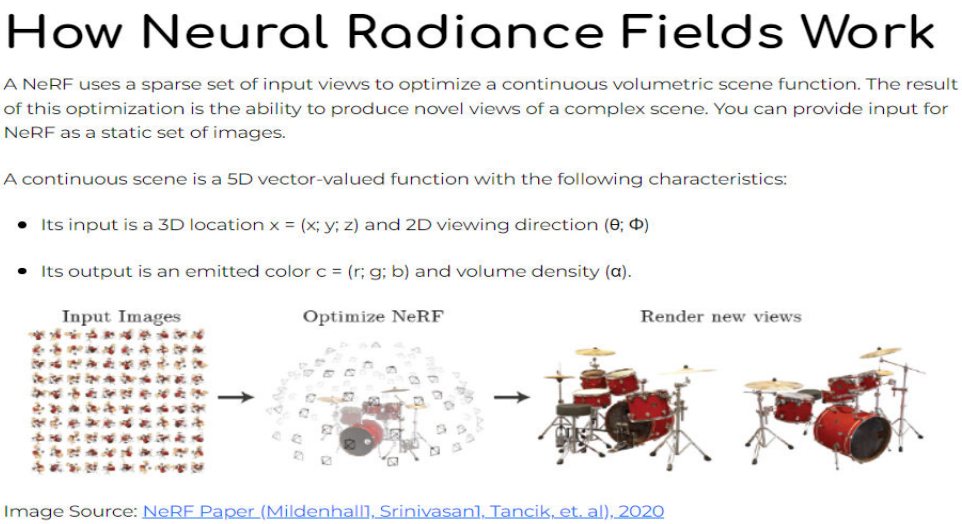


Image Source: NeRF Paper (Mildenhall, Srinivasan, Tancik, et. al), 2020

Preliminary results do show promise as well with 3D Gaussian Splatting (see below). 3D Gaussian splats were used in addition to the data provided by the neural radiance fields. The SPLAT result shown below is entitled "Sea Creature Blue." Splats use the mathematical Gaussian function to generate a point cloud visualization onto a 3D space. These dots, or 'splats' blend together with their colors to create a cohesive 3D scene. These splats were used to generate what

appears to be the shape, form, and color of the siphonophore from video 1, with surrounding background or 'blue water.' Collectively these dots paired with the gaussian function map out the body of the siphonophore.

Essentially, Gaussian 3D Splatting uses a rasterization technique that allows real-time rendering of photorealistic scenes from small samples of images. It begins with estimating a point cloud from the initial set of images using the Structure from Motion method. Each point is then converted to a Gaussian, which is described by parameters such as position, covariance, color, and transparency. It has advantages to photogrammetry and neural radiance fields in that it can generate high quality, ultra realistic scenes with rich detail, as well as capture thin surfaces like hair. However, it has high VRAM usage.

Upload these results in supersplat to view them: SuperSplat (playcanvas.com)
Tutorial for how to effectively view these results in Supersplat: GitHub - playcanvas/super-splat: 3D Gaussian Splat Editor

Acknowledgements of external contribution:

The authors of this white paper would like to sincerely thank Steven Hernandez, a graduate student currently serving as a research assistant at the Media Arts and Sciences and Herberger Institute at Arizona State University, for his production and contribution of this data. For more specific explanations of methods used in Gaussian Splatting and Neural Radiance Fields, he can be contacted at the following email: sherna86@asu.edu. The results can be found in the "Final_Results" folder which contains the neural radiance fields "SPLAT" folder.