Independent Research Project Report - Coral Reefscape 3D Reconstruction

Dong WANG
Florida Institute of Technology
August 31, 2024

Abstract

This research explores the integration of the ReScape computer-vision algorithm with the Deep Anything monocular depth estimation model to enhance scene understanding and reconstruction of the coral roofscape images. ReScape removes the perspective distortion from reef scape images by transforming them into top-down views, making them usable for quantitative analysis of reef conditions. Deep Anything's advanced monocular depth estimation provides accurate depth information from single images. By leveraging these technologies together, our approach aims to improve the precision and reliability of depth perception in various applications. This integration is particularly valuable in scenarios where depth sensors are impractical or unavailable.

In the research, we also explore the integration of the Segment Anything algorithm. At the end of the report, the potential reasons for the depth discontinuities are discussed that occur where two surfaces meet at different depths. Additionally, the report discusses the kurtosis of the depth value distribution as the camera moves across the object.

The code is available at https://github.com/wang0dong/3DReScape.git.

1. Introduction

In the realm of computer vision, depth estimation is a crucial task that enables machines to understand the three-dimensional structure of the world from two-dimensional images. One of the advancements in this field is the ”Deep Anything” model, which leverages deep learning techniques to infer depth information from a single image. Unlike traditional depth estimation methods that rely on stereo vision or specialized depth sensors, Deep Anything utilizes convolutional neural networks (CNNs) to predict depth maps with high accuracy directly from monocular inputs. Developed to address the limitations of conventional depth estimation methods, Deep Anything integrates advanced neural network architectures and extensive training on diverse datasets to provide robust depth predictions in a wide range of scenarios. This model is particularly advantageous in applications where depth sensors are impractical or unavailable.

The model's key innovation lies in its ability to generalize across various environments and lighting conditions, making it suitable for real-world applications including autonomous driving, augmented reality, and robotics. By transforming single images into detailed depth maps, Deep Anything opens up new possibilities for spatial understanding of the coral reefscapes images.

1.1. Background

Coral reefscapes are complex marine environments that are both ecologically significant and visually intricate. Traditional methods of modeling these ecosystems often require multiple images or complex sensor data, which can be resource-intensive and challenging to obtain. As a result, there is a growing

need for methods that can accurately reconstruct 3D models from a single 2D image to facilitate ecological studies, conservation efforts, and virtual simulations of coral reefs.

This research is conducted as part of an independent summer project in collaboration with the Institute for Global Ecology PhD student Zack Ferris and Professor Robert van Woesik, and with the supervision of Professor Eraldo Ribeiro, this study explores the application of advanced algorithms to address the challenges of 3D model reconstruction. Specifically, the research focuses on utilizing the ReScape algorithm.

1.2. Objectives

The accurate reconstruction of 3D models from single 2D images presents a significant challenge in computer vision and graphics, particularly in the context of complex environments such as coral reefscapes. This research aims to develop a method for generating detailed and accurate 3D models of coral reefscapes from a single image. To address this challenge, we propose leveraging the ReScape algorithm, which is designed to create a top-down view of the reefscape image. By integrating the ReScape algorithm into the modeling process, we seek to improve the accuracy and fidelity of the generated 3D models, thereby advancing the state of the art in 3D reconstruction from limited visual data.

2. Method

The Figure 1. Content Diagram of Research Methodology below illustrates the methodology employed in this study. The process begins with the pre-processing of 2D coral-reefscape raw image files. This involves the creation of a depth map to represent the image's depth information, segmentation to identify and isolate different features within the image, and the application of the ReScape algorithm to produce a 2D top-down view of the image and the image compression factor.

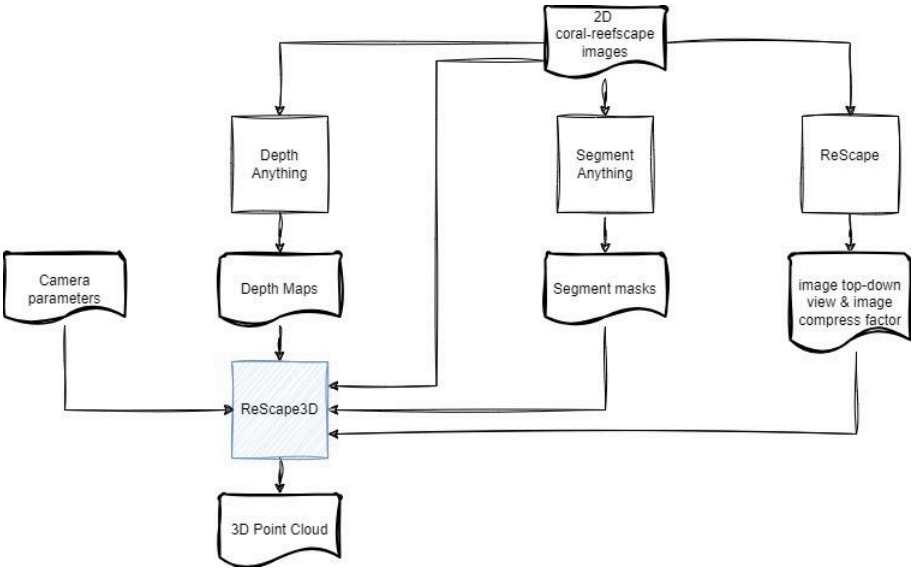

Figure 1. Content Diagram of Research Methodology

Using the outputs generated from the pre-processing step, the ReScape3D algorithm constructs a 3D point cloud iteratively. The process then rotates the point cloud such that the camera's coordinate axis, ***w***, is aligned perpendicular to the plane fitted to the point cloud, ensuring a consistent orientation for further

analysis. Following this rotation, the algorithm generates a 2D top-down view of the point cloud. Next, the algorithm calculates the compression factor, a crucial metric that quantifies the degree of compression or expansion experienced by the point cloud when projected onto the 2D plane. This compression factor is then rigorously compared against a baseline value, which has been previously estimated by the ReScape algorithm. The comparison is essential for determining the best relative distance between the camera and the image plane, ensuring that the 3D reconstruction accurately reflects the object's geometry. By fine-tuning this distance, the ReScape3D algorithm enhances the precision of the 3D model, minimizing distortions and improving the overall fidelity of the reconstructed scene.

2.1. Research Design

- **Depth and Inverse Projection**

  When a camera captures an image of a scene, depth information is inherently lost because three-dimensional objects and points are projected onto a two-dimensional image plane. This process, known as projective transformation, involves converting spatial points into pixels on a 2D plane.

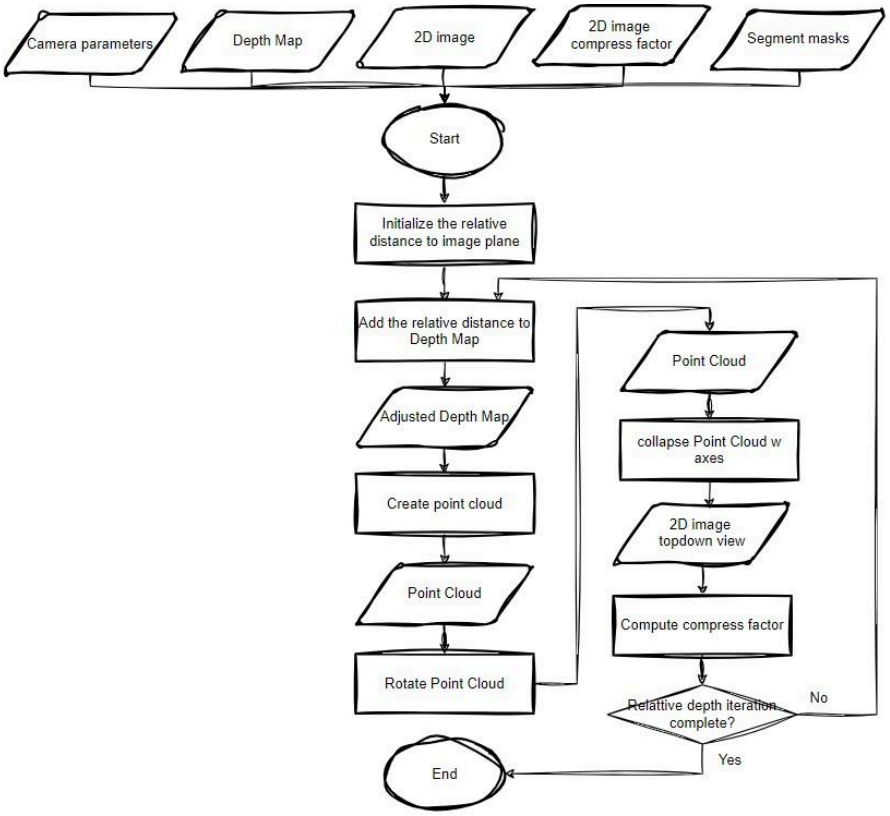

Figure 2. ReScape3D Algorithm

However, in scenarios where we need to reverse this process—recovering and reconstructing the scene from a 2D image, we must also ascertain the depth or Z-component of each corresponding pixel. Depth information can be represented as a separate image, as illustrated in Figure 3. Inverse projection (center), where darker intensities indicate greater distances from the camera.

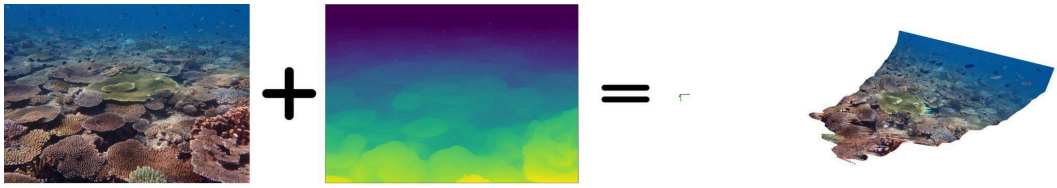

Figure 3. Inverse projection

Depth perception is crucial for various computer vision applications. For instance, in autonomous vehicles, accurate depth measurement enables more informed decision-making by providing a clear understanding of distances between the vehicle, other vehicles, and pedestrians. In this research, depth maps are stored as heat maps, where normalized relative distances are represented. In this representation, a value of 255 signifies the maximum distance within the scene, while 0 denotes the minimum distance.

First and foremost, understanding the geometrical model of the camera projection serves as the core idea. What we are ultimately interested in is the depth, parameter Z. Here, we consider the simplest pinhole camera model with no skew or distortion factor. 3D points are mapped to the image plane $(u,\ v)\ =\ f(X,\ Y,\ Z)$. The complete mathematical model that describes this transformation can be written as $p\ =\ K[R|t] \star P$.

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Equation 4

Where

- p is the projected point on the image plane.
- K is the camera intrinsics matrix.
- $[R|t]$ is the extrinsic parameter describing the relative transformation of the point in the world frame to the camera frame.
- $P,\ [X,\ Y,\ Z,\ 1]$ represents the 3D point expressed in a predefined world coordinate system in Euclidean space.
- Aspect ratio scaling, s: controls how pixels are scaled in the x and y direction as focal length changes.
- The matrix K is responsible for projecting 3D points to the image plane. To do that, the following quantities must be defined as
    - Focal length $(f_x,\ f_y)$: measure the position of the image plane wrt to the camera center.
    - Principle point $(u_0,\ v_0)$: the optical center of the image plane.
    - Skew factor: the misalignment from a square pixel if the image plane axes are not perpendicular. In our example, this is set to zero.

Instead of creating the function from scratch, the method $o3d.geometry.PointCloud.create\_from\_rgbd\_image$ from the Open3D library is imported for generating 3D point clouds from RGBD (Red, Green, Blue, Depth) images. This method utilizes both color and depth information to reconstruct the spatial layout of a scene. The method

processes the RGB and depth images along with the camera intrinsics to compute a 3D point cloud. Each pixel in the depth image is translated into a 3D point in space based on the depth information and the RGB values are assigned to these points.

Once the points are represented in 3D, a useful application is to project them onto a top-down view of the scene. This projection allows for comparison with the top-down view generated by the ReScape algorithm, providing an indirect validation of the 3D model's accuracy.

- **Brute force the distance to the image plane**

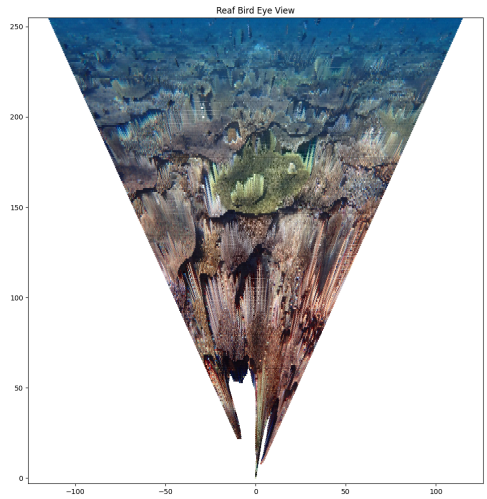

Figure 5. Reefscape Top-down view

Figure 5. Reefscape Top-down view displays the top-down view generated from the initial depth map produced by Depth Anything. The 2D image reveals reef scapes converging towards the optical center. Notably, there is significant distortion near the optical center, resulting in a considerable deviation from the baseline 2D image.

This issue arises because the depth map produced by Depth Anything does not account for the distance between the object and the image plane, as illustrated in Figure 6. Distance to image plane.

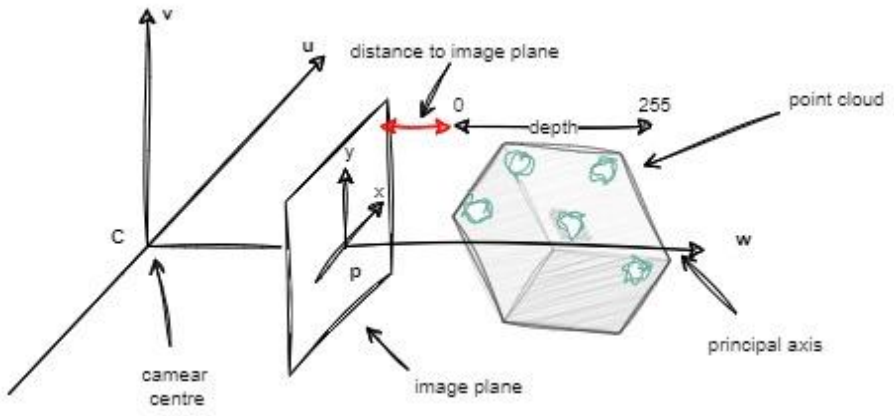

Figure 6. Distance to the image plane

To solve the image distortion issue, the distance to the image plane is incorporated into the depth map and the following additional steps are introduced when the 3D point cloud is created.

- Fit a plane to the point cloud using least squares fitting
  The method is truncated SVD. Truncated Singular Value Decomposition (Truncated SVD) is a variant of the Singular Value Decomposition (SVD), commonly used in dimensionality reduction, data compression, and noise reduction. Truncated SVD computes only the top k singular values and corresponding singular vectors. This significantly reduces the memory and computational requirements.

$$A_k = U_k \sum_k V_k^T$$

Equation 7.

  Where
  - U_k consists of the first *k* columns of *U*.
  - $\sum_k$ is the $k \times k$ diagonal matrix of the top *k* singular values.
  - V_k^T consists of the first *k* rows of V^T.

  This step will return the fit plane and the plane's normal vector.

- Rotate the camera
  Given the fit plane's normal vector n from the previous step and the camera coordinate system's w-axis normal vector $t = [0, 0, 1]^T$. The rotation axis r is computed as the cross product of the vector n and the t, which is perpendicular to both n and t and thus serves as the axis around which the rotation will occur. The rotation angle θ required to align n with t can be determined using the dot product and the magnitude of the cross product:

$$cos(\theta) = \mathbf{n} \cdot \mathbf{t}$$

Equation 8.

$$sin(\theta) = ||\mathbf{r}||$$

Equation 9.

Here, *cos(θ)* is calculated directly from the dot product of n and t, and *sin(θ)* is given by the magnitude of the rotation axis r. The rotation angle θ is determined using the arctangent function.

$$\theta = arctan2(sin(\theta), cos(\theta))$$

Equation 10.

With the normalized rotation axis r and the angle θ, the rotation matrix R is constructed using the axis-angle representation:

$$R = get\_rotation\_matrix\_from\_axis\_angle(r \cdot \theta)$$

Equation 11.

The function *get_rotation_matrix_from_axis_angle* from the Open3D library generates the 3D rotation matrix from the axis-angle vector $r \cdot \theta$. This matrix effectively rotates the normal vector n into alignment with the target normal vector t.

The figure below illustrates the steps ‘Fit a plane’ and ‘Rotate the camera’.

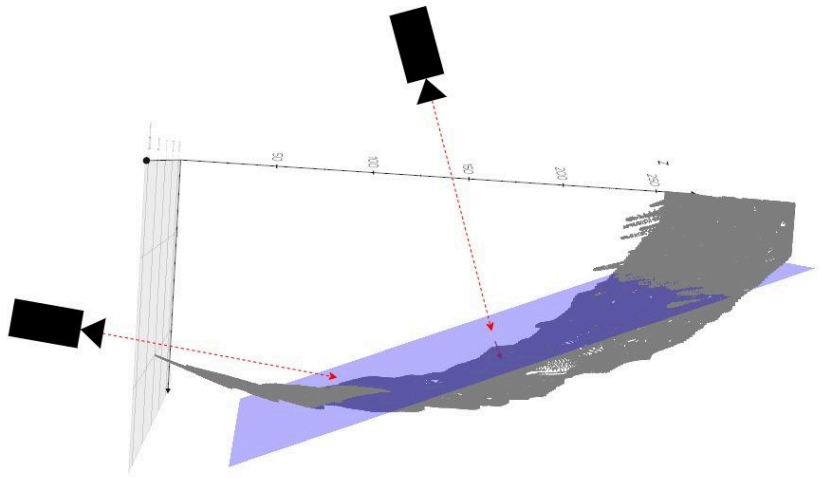

Figure 14. Fit a plane and rotate the camera

- Parallel project
  Parallel projection provides a straightforward method to visualize and analyze 3D point clouds in a 2D plane while preserving the relative proportions of objects.
- Calculate compression factor
  The *scipy.spatial.ConvexHull* method is employed to determine the boundary points that enclose the 2D points. From these boundary points, the largest possible quadrilateral is identified. Once the four corner points are determined, the Euclidean distance between the top and bottom corners is calculated to derive the compression factor. The figure below illustrates an example of this step result.

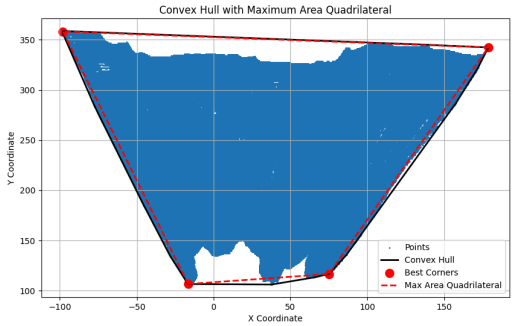

Figure 15. 2D image quadrilateral

- **Optimized depth**
  The brute force search produces a list containing both the relative distances and the corresponding compression factors of the 2D images. We choose the compression factor that best aligns with the results obtained from the ReScape algorithm's Brute Force Inverse Perspective Mapping to identify the optimal depth value. This method ensures that the selected depth value is the most accurate, enhancing model fidelity and precision. The figure below shows the relative distance brute force search results for the image TMFM0126.

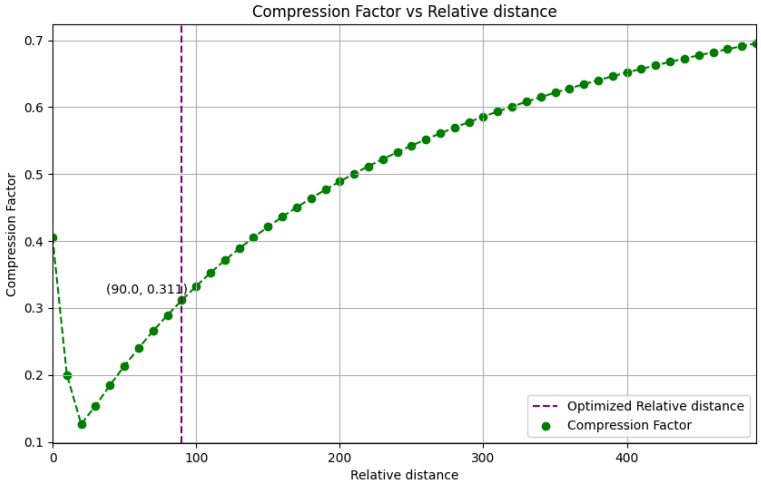

Figure 16. Relative distance brute force search

Figure 17 illustrates the dimensional changes in the 3D point cloud during depth map offset iterations. As the depth map offset increases, the 3D point cloud expands across all three dimensions, resulting in a larger and more dispersed structure. This expansion reflects the increased distance between points within the cloud, affecting the overall spatial configuration and potentially influencing the accuracy of subsequent analyses. The relationship between depth map offset and point cloud dimensionality highlights the sensitivity of the model to depth adjustments, emphasizing the importance of precise calibration for accurate 3D reconstruction.

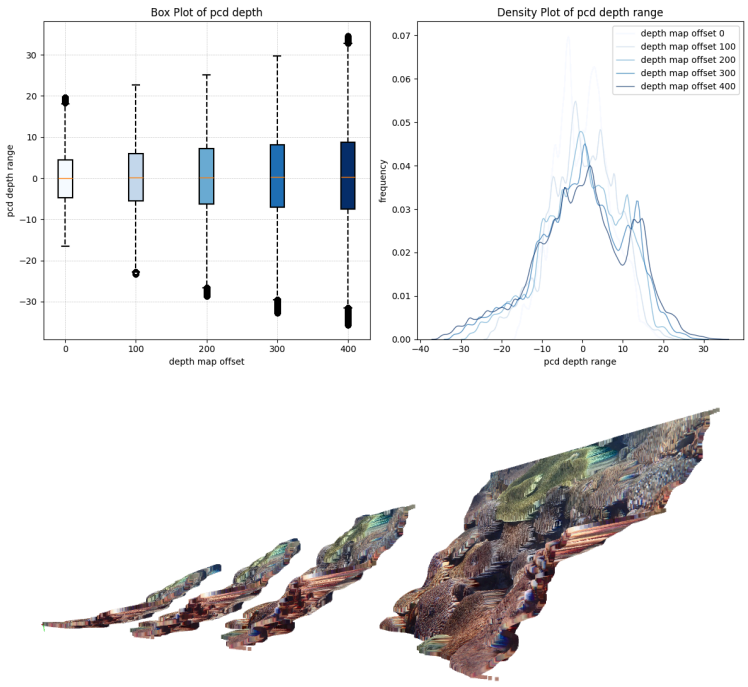

Figure 17. 3D point cloud dimension change in depth iteration

3. Discussion

3.1. Depth Discontinuities

Depth Discontinuities: Edges often correspond to depth discontinuities, where two surfaces meet at different depths. Accurately estimating depth at these points is difficult because the depth can change sharply, leading to ambiguities in depth estimation algorithms. The image below shows the 2D image TMFM0126 top view result from two algorithms ReScape3D and ReScape.

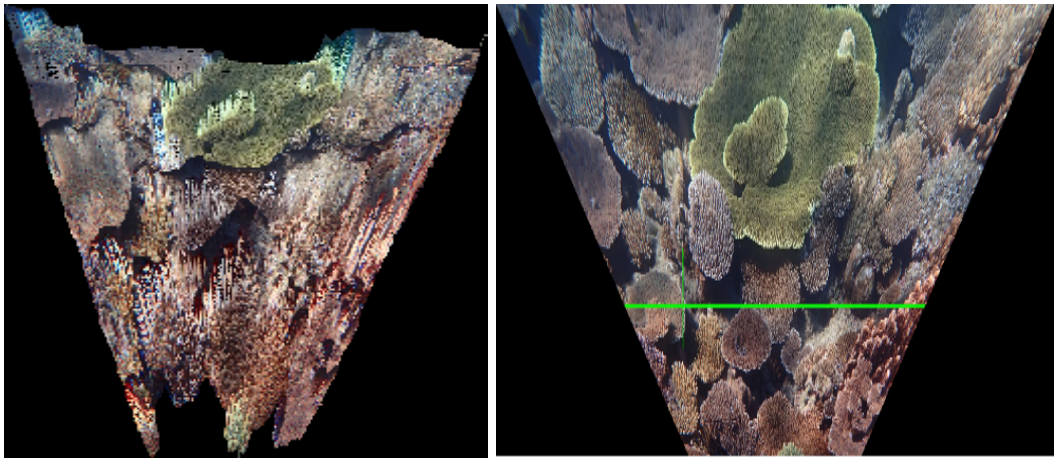

Figure 18. Top-down view compare (ReScape3D vs ReScape)

The left image in Figure 18. Top-down view compare (ReScape3D vs ReScape indicates the depth estimation algorithms may smooth out these edges, leading to visual artifacts, such as

improper shadows or inaccurate object boundaries, which detract from the realism of the rendered scene.

Here are two potential problems associated with depth discontinuities:

- **Edge Detection**: Depth discontinuities frequently correspond with the edges of objects, making accurate edge detection essential yet challenging. The Depth Anything algorithms may inadvertently smooth these edges, resulting in inaccurate depth maps. For instance, in the image below, the algorithm needs to correctly identify the edges between Reef A and Reef C and recognize the depth gap between them. Without this, the algorithm might mistakenly smooth the depth transition between Reef A and Reef C.

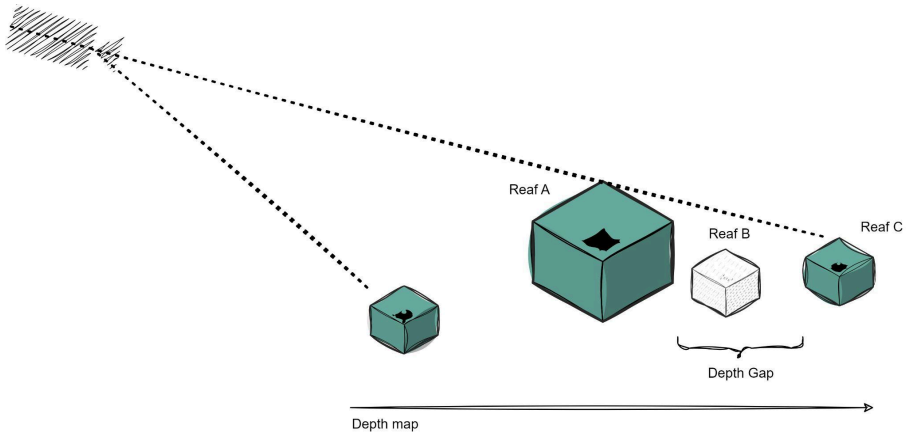

Figure 21. Edge detection

- **Occlusion Handling**: At depth discontinuities, parts of one object might occlude another. Depth estimation algorithms must correctly handle these occlusions to avoid artifacts, such as depth "bleeding" from one object to another. The image below illustrates the object 15 depth “bleeding” to the adjacent object.

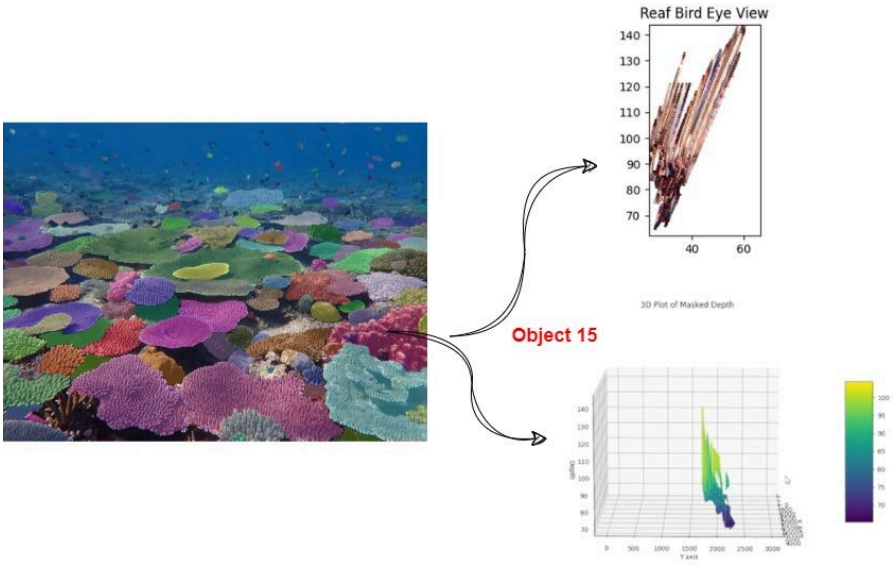

Figure 22. Depth "bleeding"

3.2. Segmentation

Segmentation is a critical step in computer vision that involves partitioning an image into distinct regions or segments, each representing different objects or areas of interest. In this research project, we utilize Meta's Segment Anything technology, a state-of-the-art solution for object segmentation. This technology employs advanced machine learning models and extensive data sets to provide high-quality results across diverse applications. We can construct three-dimensional models from segmented two-dimensional image data using the image segments and enhanced depth information from the previous study. This approach bridges the gap between 2D imagery and 3D representations, facilitating detailed analysis and visualization of objects and scenes.

Figure 23. Image segmentation

The segmentation mask can be applied to the image's depth map to isolate specific regions or components within the scene. By accurately separating these parts, the mask enables the reconstruction of a 3D model representing the individual object in the image. This process ensures that only the targeted object is included in the final 3D model, while extraneous elements are excluded, resulting in a precise and clean representation of the object in three dimensions. This process of using segmentation masks on depth maps to create 3D models of individual objects has several potential applications across various fields:

- Autonomous Systems and Robotics:
  Robots can utilize segmented 3D models to identify and interact with specific objects in their environment, improving object manipulation, navigation, and task execution.
- Augmented Reality (AR) and Virtual Reality (VR):
  Segmented 3D models can be used to accurately overlay digital objects onto real-world environments, enhancing AR experiences. In VR, these models enable the creation of realistic virtual environments populated with precise 3D representations of real-world objects.
- Medical Imaging and Diagnostics:
  Segmentation of depth maps from medical scans (e.g., MRI or CT scans) allows for the isolation and 3D reconstruction of specific anatomical structures, aiding in diagnosis, surgical planning, and treatment.

3.3. Kurtosis

Kurtosis is a statistical measure that describes the shape of a distribution's tails in relation to its overall shape. It quantifies whether the data points in a distribution are more or less concentrated around the mean, particularly in the tails.
There are three types of kurtosis:

- Leptokurtic (positive kurtosis): This type of distribution has fatter tails, meaning there are more outliers. The peak of the distribution is also higher and sharper compared to a normal distribution. A leptokurtic distribution has a kurtosis value greater than 3.
- Mesokurtic (normal kurtosis): This refers to a normal distribution with a kurtosis value of 3. It serves as a benchmark to compare other distributions. The tails are moderate, and the peak is neither too sharp nor too flat.
- Platykurtic (negative kurtosis): This distribution has thinner tails and fewer outliers, with a lower, broader peak. A platykurtic distribution has a kurtosis value less than 3.

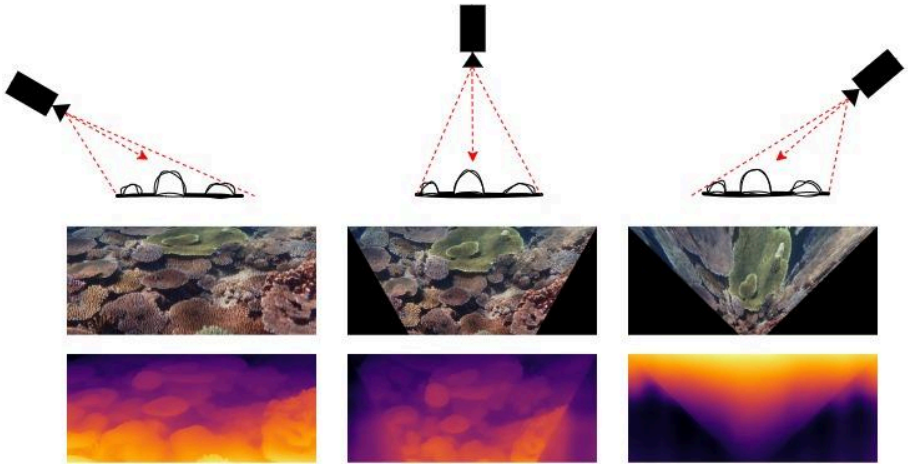

Figure 24. Kurtosis

Figure 24. Kurtosis illustrates the theoretical change in the kurtosis of the depth value distribution as the camera passes over an object. According to the theory, when the camera is positioned directly above the object's surface, the depth distribution should exhibit normal kurtosis, indicating the depth values distributed in a way that has a peak and tails similar to a normal distribution, without extreme outliers or a concentration of depth values at a particular distance.
As the camera is rotated away from this central position, the kurtosis decreases, which indicates that the scene has more variability in depth, with objects at varying distances. There may be less uniformity in depth, possibly reflecting a more complex scene with diverse surfaces and structures at different depths. In this context, kurtosis could be another valid indicator of the destination camera position besides the perspective-grid angle, which the ReScape algorithm has used in the Brute-Force Inverse-Perspective-Mapping function.
We are using the kurtosis function in the scipy.stats library in Python to calculate the excess kurtosis, which is the kurtosis relative to a normal distribution (i.e., it subtracts 3 from the raw kurtosis value). A result of 0 indicates mesokurtic behavior.

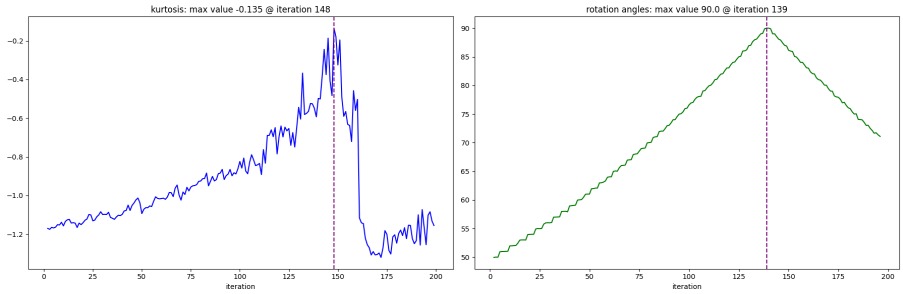

Figure 25. Kurtosis & perspective-grid angle vs compression factor

The results from the Brute-force searches conducted for both the ReScape3D and ReScape algorithms show that the optimal configurations identified are relatively close to each other. Specifically:

- **ReScape3D Search Results**: The Brute-force search guided by the maximum Kurtosis value yielded a result with a maximum Kurtosis of -0.135 at iteration 148, with a compression factor of 0.26.

- **ReScape Search Results**: The Brute-force search guided by the perspective-grid angle identified a 90-degree angle at iteration 139, with a compression factor of 0.305.

Despite the different guiding metrics used — Kurtosis for ReScape3D and perspective-grid angle for ReScape — the results indicate that the compression factors are relatively close, suggesting that the parameter settings leading to optimal outcomes are not vastly different. This close alignment implies that both metrics provide comparable insights into optimizing the algorithms and achieving effective results.

The similarity between the two search results highlights the robustness of the ReScape3D and ReScape algorithms in adapting to different parameter settings, reinforcing the validity of both guiding approaches. This alignment also suggests that the chosen metrics for guiding the searches are effective and offer complementary insights into the optimization process.

3.4. Future research

Building upon the findings of this study, several avenues for future research emerge, which could further enhance the accuracy and applicability of 3D modeling from images using the ReScape3D algorithm.

- **Advanced Edge Detection Techniques** Future research should explore the development and integration of advanced edge detection algorithms to better identify depth discontinuities. Current methods may benefit from enhancements in detecting and accurately representing the boundaries where surfaces meet at different depths. Investigating techniques such as edge-aware filtering, gradient-based methods, or machine learning approaches could improve the precision of depth discontinuity detection of the Depth Anything algorithm.

- **Enhanced Occlusion Handling** Occlusions, where objects partially or fully obscure other objects, pose significant challenges in 3D modeling. Future studies could focus on developing robust methods for handling occlusions, including:
    - Multi-View Integration: Combining data from multiple viewpoints to reconstruct occluded regions and improve model completeness.
    - Predictive Models: Using predictive models to estimate occluded parts based on visible data, potentially leveraging AI techniques to infer hidden structures.
    - Adaptive Algorithms: Designing adaptive algorithms that can dynamically adjust to varying levels of occlusion and improve depth estimation in complex scenes.

- **The guidance to optimize the depth value** Future research should explore other guidance to optimize the depth value map created by the Depth Anything algorithm, in this research, I'm using the compression factor found by ReScape in 2D image's Brute-force perspective-grid angle search and the 3D model's fidelity has not been verified.

Once a high-fidelity 3D model is available, the Coral Reefscapes Rugosity Study can proceed with several advanced analyses and applications:

- Detailed Rugosity Analysis
  Utilize the accurate 3D model to perform a detailed analysis of the coral reef's rugosity. This includes measuring the surface area and planar area of the reef to calculate various rugosity metrics, such as average rugosity, surface roughness, and texture complexity.

- Habitat Assessment
  Assess the relationship between rugosity and biodiversity. Use the high-fidelity model to identify complex structures and microhabitats that may support a diverse range of marine species. Correlate rugosity measurements with species distribution and abundance data.

- Visualization and Interpretation
  Create detailed visualizations and interactive models to better understand and communicate the reef's structure. Use these visualizations to illustrate rugosity patterns, habitat complexity, and ecological interactions.

Reference

1. Z. Ferris, E. Ribeiro, T. Nagata and R. van Woesik. ReScape: transforming coral-reefscape images for quantitative analysis. In *Nature Scientific Report (2024) 14:8915.*
2. Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, Hengshuang Zhao. Depth Anything V2. In *arXiv 13 Jun 2024*.
3. Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollar, Ross Girshick. Segment Anything. In *arXiv 05 Apr 2023*.
4. Daryl Tan. Inverse Projection Transformation. In *Medium Dec 15, 2019*.
5. Aqeel Anwar. What are Intrinsic and Extrinsic Camera Parameters in Computer Vision? In *Medium Feb 28, 2022*.