PROBABILISTIC ROADMAPS FOR VIRTUAL CAMERA PATHING WITH CINEMATOGRAPHIC PRINCIPLES

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Katherine Davis

April 2017

COMMITTEE MEMBERSHIP

TITLE:	Probabilistic Roadmaps for Virtual Camera Pathing with Cinematographic Principles
AUTHOR:	Katherine Davis
DATE SUBMITTED:	April 2017
COMMITTEE CHAIR:	Zoë Wood, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Aaron Keen, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Maria Pantoja, Ph.D Assistant Professor of Computer Science

ABSTRACT

Probabilistic Roadmaps for Virtual Camera Pathing with Cinematographic Principles

Katherine Davis

As technology use increases in the world and inundates everyday life, the visual aspect of technology or computer graphics becomes increasingly important. This thesis presents a system for the automatic generation of virtual camera paths for fly-throughs of a digital scene. The sample scene used in this work is an underwater setting featuring a shipwreck model with other virtual underwater elements such as rocks, bubbles and caustics. The digital shipwreck model was reconstructed from an actual World War II shipwreck, resting off the coast of Malta. Video and sonar scans from an autonomous underwater vehicle were used in a photogrammetry pipeline to create the model.

This thesis presents an algorithm to automatically generate virtual camera paths using a robotics motion planning algorithm, specifically the probabilistic roadmap. This algorithm uses a rapidly-exploring random tree to quickly cover a space and generate small maps with good coverage. For this work, the camera pitch and height along a specified path were automatically generated using cinematographic and geometric principles. These principles were used to evaluate potential viewpoints and influence whether or not a view is used in the final path. A computational evaluation of 'the rule of thirds' and evaluation of the model normals relative to the camera viewpoint are used to represent cinematography and geometry principles.

In addition to the system that automatically generates virtual camera paths, a user study is presented which evaluates ten different videos produced via camera paths with this system. The videos were created using different viewpoint evaluation methods and different path generation characteristics. The user study indicates that

users prefer paths generated by our system over flat and randomly generated paths. Specifically, users prefer paths generated using the computational evaluation of the rule of thirds and paths that show the wreck from a large variety of angles but without too much camera undulation.

ACKNOWLEDGMENTS

Thanks to:

- The Cal Poly graduate students for creating and uploading this thesis template.
- Dr. Zoë Wood and Shinjiro Sueda for their CSC 471 and CSC 474 base code.
- Keenan Reimer for the underwater scene created for his CSC 471 final project.
- The Cal Poly Computer Science Department and all my amazing professors for the classes, support, and knowledge that made this thesis possible.
- The computer science graphics thesis group who gave me invaluable advice, resources, and debugging help.
- My friends and roommates: Lana, Allie, Vivian, Katelyn, Denise, and so many more who have given me unending support and encouragement.
- Sara Bilich and Sebastian Siebert von Frock for caustics code, the initial PRM node class, and shipwreck model creation.
- Sara, Sebastian and the rest of the ICEX 2016 team: Chris Clark, Zoë Wood, Vai Viswanathan, Zayra Lobo, and Jessica Lupanow who inspired the idea for this project, helped create the shipwreck model, and taught me about PRMs.
- My parents for their encouragement, patience, and trust that I would eventually graduate college.
- Cameron, for being my rock to stand on.
- Zoë, for being my thesis adviser, professor, boss, travel buddy, occasional Mom, chocolate provider, advice giver, cause of countless late night, and most importantly, friend. You gave me so many opportunities and believed in me even when I didn't– I truly would not be here without you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

There exist millions of exquisite computer graphics models and scenes carefully crafted by artists and computer scientists. In order to show off these models and scenes, animators must hand create camera paths that orbit the model, often called fly-throughs. They do this by carefully choosing camera viewpoints that display specific features of the models and interpolating smooth paths between these points. This is a time consuming and painstaking process that requires artistic ability to do well. This work seeks to generate automatic fly-throughs for three-dimensional (3D) scenes focused on a single model. Meanwhile, there is a rich history of cinematographic principles examined in film studies and long standing history of geometry used in art and mathematics. We use these cinematographic and geometric considerations to generate visually pleasing paths. These considerations evaluate viewpoints from heights and pitches generated by our system along a specified path. Users are pleased with the paths we create and prefer the paths generated with cinematographic principles over simple, flat paths. The "rule of thirds" is used to characterize cinematographic principles, and object normals are representative of geometric evaluation.

This specific project was born out of the International Computer Engineering Experience (ICEX). This is a joint project between California Polytechnic State University, San Luis Obispo and Harvey Mudd College. Computer graphics and robotics students travel to Malta to perform research on shipwrecks off the Maltese coast. Underwater robots, sonar scans, video footage, and photogrammetry are used to create 3D computer graphics models of shipwrecks as explained in Chapter 2. One of the goals of this project is to create educational videos about these shipwrecks. To this end, this thesis focuses on generating animations that feature visualizations of the

shipwreck in a virtual underwater setting. Specifically, we take the shipwreck model generated by the ICEX team of the X127, an actual World War II shipwreck, and place it in a virtual underwater world. With such a unique and historically significant model, this work aims to create automatic visualizations that display it in a compelling way.

Inspired by this multi-disciplinary robotics and computer graphics project, this thesis explores the use of a robotics motion planning algorithm to generate the fly-through camera paths. Motion planning algorithms seek to solve the "find path problem" or how to move from point A to point B in a specified space. This body of algorithms includes A*, Dijkstra's shortest path, trajectory planning, and many more. A popular motion planning algorithm known for generating small maps with good coverage and connectivity is the probabilistic roadmap (PRM) algorithm [8]. PRMs are ideal for classic robotics problems because they are highly customizable and can create paths that follow kinematic constraints of different robots. PRMs have also been used in a variety of other applications such as video games, character animation, and even computation biology [11]. We use a PRM to generate the fly-through camera paths in our underwater shipwreck scene given that PRMs have good coverage while keeping maps small.

The contributions of this thesis are as follows:

- Image processing system that renders a given 3D scene from a specified camera position and direction then analyzes the image using computer vision with regards to cinematographic and geometric principles.

- Probabilistic roadmap algorithm that generates customizable, visually appealing camera paths for a fly-through of a 3D scene focused on a single model.

- A user study showing which characteristics of fly-through paths are preferable to users.

Chapter 2

BACKGROUND

This work combines elements from a variety of research fields. We will give necessary background in robotics, image composition, and scientific modeling.

2.1 Probabilistic Roadmaps

Probabilistic Roadmaps are used to find an optimal path to move a robot through a configuration space. They quickly explore the space by using a rapidly-exploring random tree (RRT). Two primary strategies are used to create PRMs. These are a multi-query PRM which generates a single roadmaps that is used many times and a single-query PRM that creates a new roadmap every time a search is done through the space. We will focus on single-query PRMs in this work. In a typical PRM we have a roadmap, which is a set of nodes and edges, and a configuration space which is the space we are trying to move through. The general algorithm shown in Algorithm 1 adds nodes to the roadmap if they are collision free until an ending condition is met. This ending condition can be a path length, time limit, or ending position. The node generation step on line 5 can be done so that the generated path follows the kinematic constraints of a particular robot.

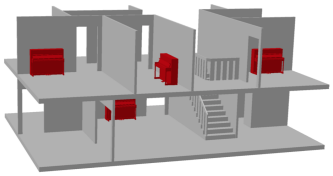
Algorithm 1: General Probabilistic Roadmap Algorithm.

```
1  R(N, E) = Roadmap(Nodes, Edges);
2  cStart = start configuration;
   Data: cStart and configuration space
   Result: Path from cStart to end
3  while end condition not reached do
4      select random node c to expand from;
5      randomly generate c' from c;
6      if edge e from c to c' is collision free then
7          R.add(c', e);
8          if c' reaches end condition then
9              return path;
10         end
11     end
12 end
```

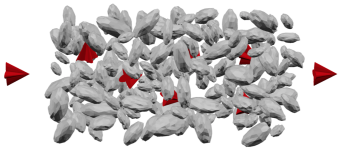
2.1.1 Configuration Spaces

A configuration defines the exact state of a robot, for example the robot position, camera tilt, or direction of rudder. The configuration space is "the space of all possible configurations of the robot" [8]. Traditional PRMs can work in a variety of configuration spaces. Ratanen studies neighborhood-based PRM construction methods in his work using a variety of test environments [18]. These environments are shown in Figure 2.1 with possible robot configurations in red and obstacles in gray. The environments include a house environment where the robot is a piano that has to move through the house without colliding with walls, stairs, roofs, and floors, and an asteroid environment with many small shapes that crowd the space and only leave

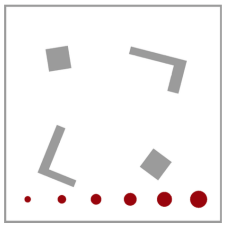
narrow passages. He also examined a two dimensional environment that has only four obstacles in it but the robot consists of six pieces that must avoid contact with each other as well as the obstacles.



(a) House environment with piano robot.



(b) Asteroid field environment with narrow passages.



(c) Two Dimensional space with few obstacles but six part robot.

Figure 2.1: Example PRM environments [18]. Possible robot configurations shown in red, obstacles shown in grey.

2.1.2 Degrees of Freedom

An object in 3D space can be defined by degrees of freedom, for example position and orientation. The first three degrees of freedom define the position of the object in space. This is the x, y, z position of the objects with relation to the coordinate frame it is in. The next three degrees of freedom define the orientation of an object. These are often referred to as yaw, pitch, roll and can be seen in Figure 2.2. Visualizing these with regard to an airplane, the yaw is rotation around the vertical axis, or how far left and right the plane is turning. Pitch is rotation around the axis that runs from wing to wing, defining the angle the plane is diving down or flying up at. Roll is rotation around the axis that runs from tip to tail, so literally the amount the plane is rolling over its wings.

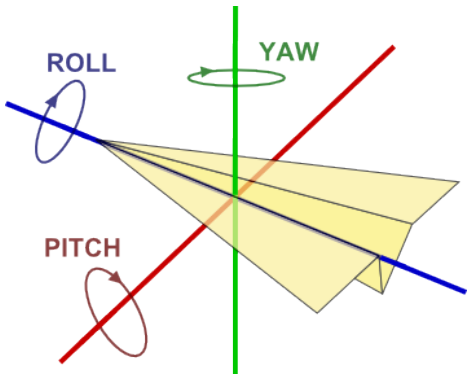


Figure 2.2: Yaw, pitch, and roll of an object. [4]

2.2 Cinematography

"Beauty is in the eye of the beholder" is an age old proverb. Aesthetics is a highly subjective topic so it is difficult to determine objectively if something is visually pleasing. However, there are some generally accepted rules of composition that tend to increase the aesthetic quality of images and videos. These rules include the rule of thirds, diagonal dominance, visual balance, and size region [16]. The rule of thirds can be explained by splitting an image up into three sections, both horizontally and vertically. The subject of the photo or horizon line should be placed along one of these lines or at the intersection of two lines as seen in Figure 2.3. This rule can be seen in many famous photographs and paintings and many photo editing tools provide such a grid when users enter cropping mode so that they may crop their pictures to conform to the rule of thirds. In this work, we use the rule of thirds to help us evaluate generated camera paths for their potential to be visually appealing. We chose the rule of thirds as one example of a cinematographic rule that could be used but discuss in the future work, Section 7.1, that the aesthetics of generated paths could be further increased by including additional rules in path creation.

Figure 2.3: Image composition "rule of thirds" shown in practice. Subject of the image and horizon line both fall along one of the thirds lines [1].

2.3 Shipwreck Data Collection

The digital model of a shipwreck used in this work is created from an actual World War II wreck in Malta. This ship, the X127, was hit on the stern by a torpedo in April 1942 and sunk immediately [5]. It has sat at the same position off the coast of Manoel Island, Malta ever since. This location is shown in Figure 2.4 and is a popular scuba diving site. The wreck is 30 meters long and lies on a slant, putting the bow 5 meters underwater and the stern 22 meters deep [6]. A group of Cal Poly and Harvey Mudd students and professors traveled to the wreck as part of the "International Computer Engineering Experience" in June 2016. They used an OceanServer Iver2 Autonomous Underwater Vehicle (AUV) with a BlueView Multibeam Sonar and GoPro Hero 2 attached as seen in Figure 2.5 to capture sonar and video data of the shipwreck. This process is further described by Viswanathan et al. in [21].

Figure 2.4: Location of the X127 shipwreck off the coast of Manoel Island in Malta.

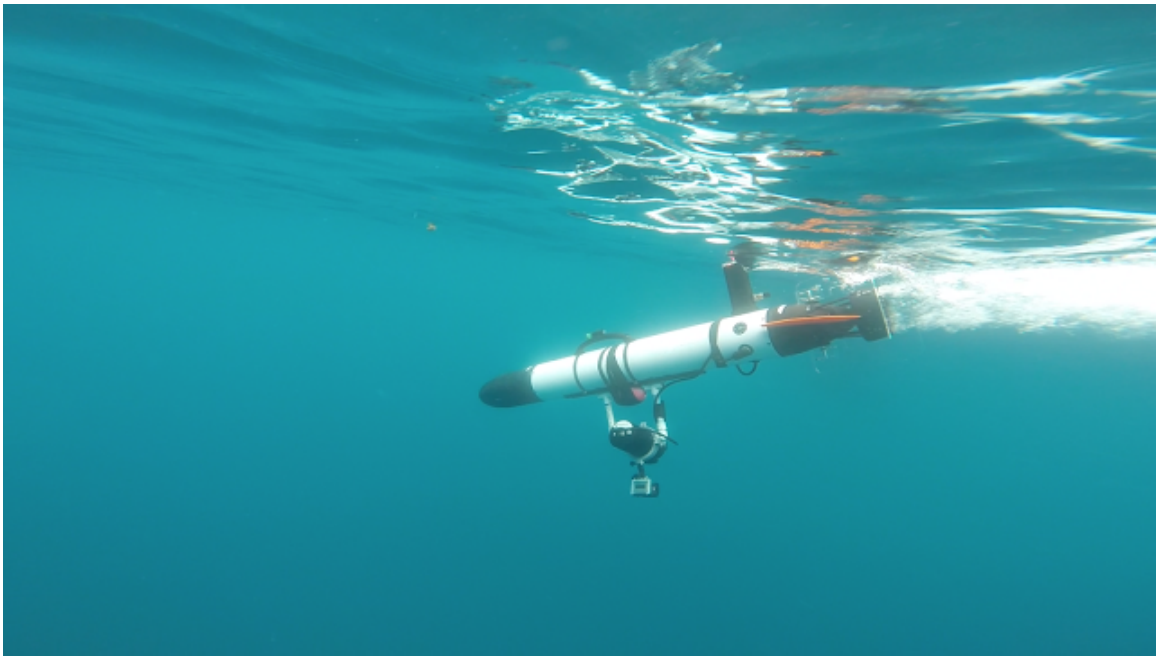


Figure 2.5: Autonomous Underwater Vehicle with sonar and GoPro attached to capture shipwreck data.

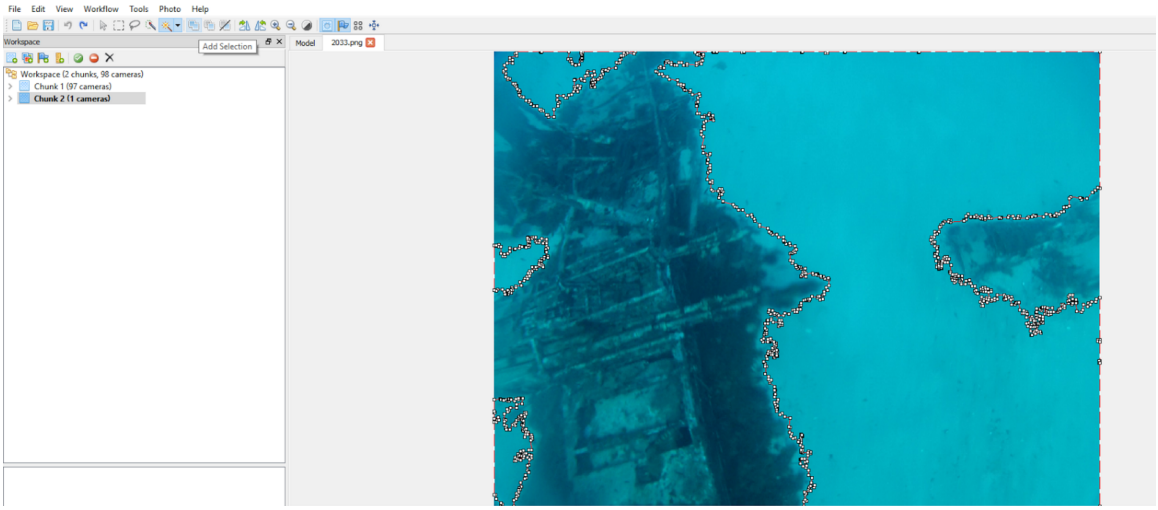


Figure 2.6: Agisoft Photoscan Professional software being used for photogrammetry, the process of generating a digital 3D models from images.

2.4 Photogrammetry

The video data collected by the AUV was used to create a digital 3D model of the shipwreck through a process called photogrammetry. Photogrammetry is generally the science of making measurements from photographs. It attempts to find the exact position of surface points to generate digital 3D models from photographs. Video frames that contain the shipwreck were pulled from the AUV video and color corrected. These images were put into Agisoft Photoscan Professional, shown in Figure 2.6, which utilizes photogammetry to create 3D models from still images. In Photoscan the photos are aligned, a point cloud is generated, and finally a mesh and texture are produced. Siebert von Frock et al. gives a more detailed description of this process [22].

Chapter 3

RELATED WORKS

A large body of work exists involving probabilistic roadmaps, virtual cameras, and image composition but none combine them in the unique way this thesis does to generate visually pleasing camera fly-throughs.

3.1 Motion Planning Algorithms for Camera Control

One of the most closely related works is that of Joubert et al. in "Towards a Drone Cinematographer: Guiding Quadcopter Cameras using Visual Composition Principles" [13]. This work uses a motion planning algorithm to control a physical drone camera, rather than a virtual camera as in our work. The authors focus on taking well-composed video footage from a quadcopter of human subjects interacting with each other. The shooting takes place outside with the subjects wearing Real Time Kinematic GPS and Inertia Measurement Unit sensors so the drone knows the exact location and orientation of the subjects. To transition between static shots, the drone uses a real-time trajectory planning algorithm that tries to maintain visually pleasing shots while moving.

Another related work is that of Li and Cheng who navigate virtual environments with a real-time camera control module [14]. This module generates camera motions that automatically control a third-person camera for use in video games. They maintain a real-time camera by using a lazy PRM that follows the game avatar. The PRM uses intercuts to avoid occlusion and collisions while preserving cinematography rules. This work and ours both use PRMs concerned with cinematographic principles but

they focus on third-person videos game cameras using geometric evaluation while we focus on fly-through camera paths using screen space evaluation.

3.2 Cinematography and Image Composition

A canonical work in the area of combining cinematography with computer science is that of Christianson et al. in 1996 [7]. Back in that time period, cameras tended to be first person or from a small set of viewpoints. Christianson et al. wanted to move towards placing cameras to help tell the story of a 3D computer graphics worlds. This paper discussed cinematography principles and created the Declarative Camera Control Language encoding 16 idioms from a film textbook to formalize cinematography principles for use in computer graphics. The idioms encoded focus on character interaction and movement rather than composition rules such as the rule of thirds used in our work.

Another work related to the aesthetics side of this thesis looks at detecting the rule of thirds in still images [17]. The paper utilizes saliency and generic objectness analysis to determine the subject of a photo. It then analyzes where the subject is in regards to the thirds lines of the photo. Mai et al. apply machine learning techniques to their output and determine their algorithms have 80% accuracy. Other works detecting the rule of thirds utilize it along with other image composition rules to create aesthetically pleasing images through cropping and retargeting [16, 19]. Since our work focuses on scenes with a single main object we use a less advanced object detection technique through the Open Computer Vision library to identify use of the rule of thirds.

3.3 Probabilistic Roadmaps

A vast body of work exists examining probabilistic roadmap theory and their applications in robotics. Below, we examine a few that relate most closely to this thesis.

Li and Shie explain their approach to the classic motion-planning problem in [15]. They create a probabilistic roadmap that expands on the Reconfigurable Random Forest data structure which is based on the traditional Rapidly-exploring Random Tree. The planner can be updated quickly to create paths that are optimal for multiple situations. If the environment being mapped is dynamic, this approach works well because the planner is called multiple times. The roadmap is kept small by periodically trimming unnecessary nodes. They also mention limiting the number of children each node can have to keep the roadmap small. This is an idea we used during part of our development cycle but did not make it into the final version of our algorithm. Our scene is given a dynamic appearance through randomly generated bubbles appearing from the seafloor and constantly moving caustics overlayed on the entire scene. However, the models in our scene are actually static so our roadmap algorithm does not take dynamic objects into consideration.

Viswanathan et al. uses a PRM planner and an AUV to help create the shipwreck model used in this thesis. A high level scan of the shipwreck area is done first and used to create a bathymetry map. This bathymetry map and is used in a PRM that creates additional AUV missions that make low altitude flyovers of the area. The PRM is optimized for "maximizing information gain" or getting good video data of the shipwreck. Model creation done of the shipwreck with photogrammetry requires photos of the shipwreck from as many angles as possible. Therefore, the PRM paths optimize for this while keeping kinematic constraints of the AUV in consideration. Our "high weight node" list described in Section 4.4 was inspired by this work's PRM that maximizes information gain. However, this work is optimizing for a high number

of camera angles to use photogrammetry while we focus on maximizing viewpoints from a cinematographic and geometric perspective.

Chapter 4

IMPLEMENTATION

This chapter provides implementation details of our camera path generation system. This system generates fly-through animations using a robotics motion planning algorithm, probabilistic roadmaps. Cinematographic and geometric principles, specifically the rule of thirds and model normals, are used to generate visually pleasing paths. The sample scene used here features a digital 3D model of an actual shipwreck in Malta, the X127, placed in an underwater scene. An overview of the path generation system is given in Figure 4.1 and each section is discussed in detail below. The path playback system and general virtual world aesthetics are then discussed.

4.1 Terminology

Common motion planning terms such as node, path, and roadmap are used frequently throughout this chapter. We explain this terminology as used in our project. A configuration space is the space the robot moves through as explained in detail in Section 2.1.1. Nodes are used to describe potential camera configurations. They contain the position and direction of a camera viewpoint, along with other pertinent information about that node, listed below.

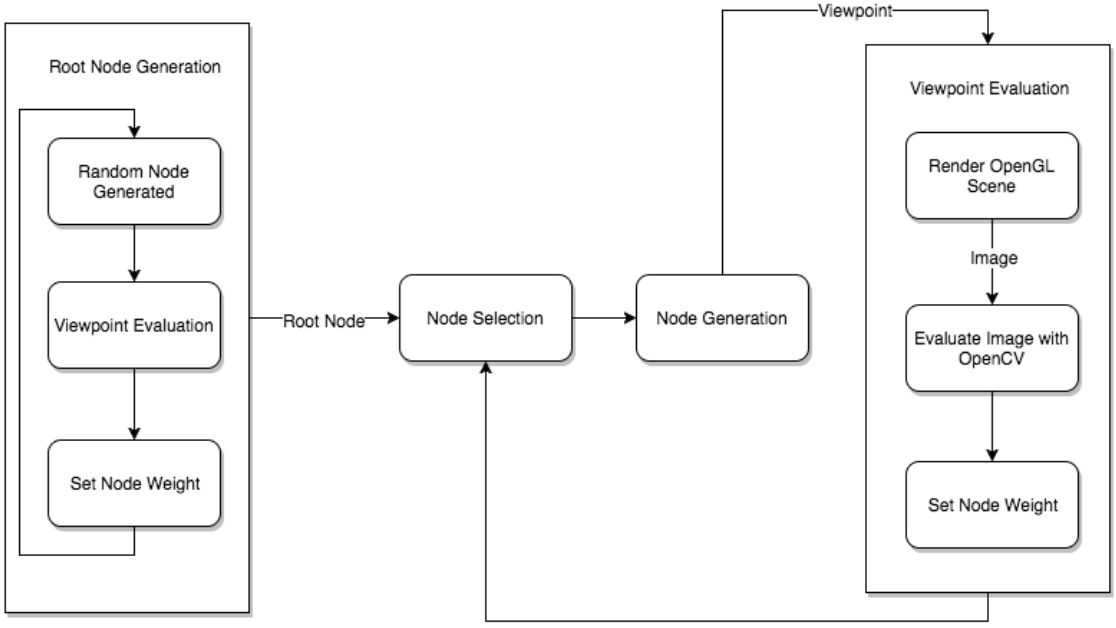


Figure 4.1: Overview of the PRM Camera Path Generation System.

```
class Node {
    Vector3f position;
    Vector3f direction;
    Node *parent;
    int pathLength;
    int roadmapNdx;
    double weight;
};
```

Nodes are placed on a tree starting from a single root node, which is the robot's starting configuration. An entire tree of nodes is called a roadmap. This roadmap contains all nodes in one run of the PRM algorithm, regardless of whether or not these nodes end up on the final fly-through path. We use the term path to describe a set of

nodes on our roadmap that form a complete course from start to end configuration of the desired length.

4.2 PRM Algorithm

Algorithm 2 gives the high level algorithm used to generate PRM Camera Paths. PRMs use a rapidly-exploring random tree to explore a configuration space with small maps that provide good coverage.

The configuration space used in this work is the height-pitch space in an underwater world. The camera paths are generated along a set elliptical path with variation in radius, height, and pitch being generated by our PRM. We know that a generally circular path is used in good fly-throughs so the x and z coordinates of our current viewpoint are calculated using the polar coordinates of a circle and how far around the circle the current viewpoint is. An example path shown from a top down view can be seen in Figure 4.2. The underwater world has various rock and seaweed models throughout and features a central shipwreck model that the camera is hoping to view from interesting angles. Ideal paths have between 10 and 15 nodes which represent different camera viewpoints. This many viewpoints allows the wreck to be seen from a variety of angles through generated pitches and heights without creating a path that moves up and down too frequently.

Broadly, the algorithm adds nodes to a roadmap until a path of the desired length is generated. Line 1 assumes a root node is selected, which is discussed in Section 4.3. Selecting a node to expand from, lines 4 through 11 are explained in Section 4.4 and line 12 refers to node generation, discussed in Section 4.5. Line 16 refers to the complicated viewpoint evaluation process, specified in Section 4.6.

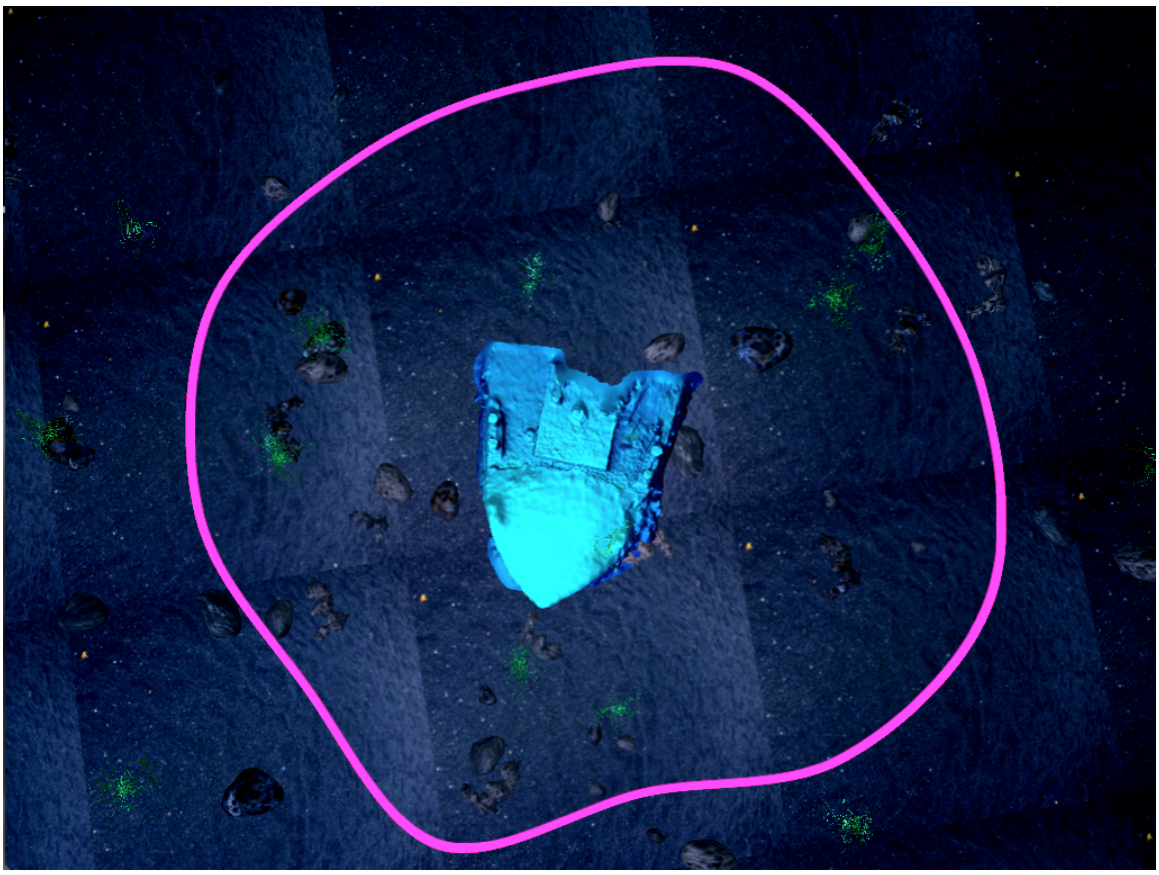


Figure 4.2: Top down view of the the shipwreck scene. Semi-elliptical path of varying radius shown in pink. Casutics turned off for clarity.

Algorithm 2: High Level PRM Camera Path Generation Algorithm.

```
1  R(N, E) = Roadmap(Nodes, Edges);
   Data: rootNode, pathLength
   Result: Path of pathLength from rootNode
2  highWeightNodes[];
3  while currentPathLength < pathLength do
       // select node c to expand from
4      if even iteration of loop then
5          c ← highWeightNodes[iteration];
6      end
7      else
8          do
9              c ← rand node in R;
10         while c < weightThreshold;
11     end
12     randomly generate c' from c;
13     calculate edge e from c to c';
14     R.add(c', e);
15     currentPathLength ← c'.pathLength;
16     c'.weight ← calculateWeight(c');
17     if c'.weight > highWeightThreshold then
18         highWeightNodes.add(c');
19     end
20 end
```

4.3 Root Node Generation

All paths and roadmaps must begin with a starting configuration, specified in a root node. We want this root node to be the best viewpoint possible, so 200 potential root nodes are generated by selecting random heights and pitches. The weight of all these nodes are calculated based on the current viewpoint evaluation criteria as described below in Section 4.6 and the best node is selected as the root node. This is the node with either the highest or lowest score, depending on the evaluation criteria currently being used. Once the root node is chosen, it is added to the roadmap, thus making the roadmap size one and the algorithm proceeds with the node selection step.

4.4 Node Selection

As shown on lines 4 through 11 in Algorithm 2, node selection is an involved process. A node to expand from must be selected for the algorithm to proceed, but we want to select a node with a better weight to generate more aesthetically pleasing paths. Throughout the entire path generation process, a list of "high weight" nodes is maintained that have values above a certain threshold as discussed in Section 4.6.3. On half of the iterations through the path generation main loop, a node from this list is selected to expand from if there are unused nodes on this list. This expansion process is detailed below in Section 4.5. On the other half of the loop iterations, a random node in the roadmap is selected to expand from. Since we do not want low weight nodes ending up in our final path, a node over a certain weight threshold is selected as shown on line 10. This threshold is lower than the threshold for the high weight node list. Additionally, this threshold decreases slightly each iteration through the line 10 loop so that eventually a node will be selected even if the roadmap only contains low quality nodes. Using node weight thresholds is a pruning step that helps keep

the roadmap size small by not expanding the roadmap in directions with low quality nodes.

4.5 Node Generation

In order to display our scene in a compelling way, the roadmap must expand out through our configuration space to show the shipwreck at a variety of angles. Once a node to expand from has been selected as described above, a new node must be generated based off this old node. This node generation is done by taking the height of the old node and adding and subtracting a height delta. A random height is then chosen within this new range. The same process happens for pitch, selecting a random pitch within a delta of the old node's pitch.

The specifics of the height and pitch deltas are described in Chapter 5 but height delta was typically in the range of 1-3 and pitch delta was generally between 5°and 15°. The full X127 shipwreck is 30 meters long, with the front portion shown in our digital model being about 1/4 of the total ship length or 7.5 meters long. Since the digital wreck model is 12 units long in our world, this means 1 meter in the real world is about 1.6 units in our virtual world.

We know that good fly-throughs follow a mostly circular path so we determine the x and z coordinates and yaw and roll degrees of the node by calculating the polar coordinates of a circle. The current node's position along the path is used to perform sine and cosine calculations. These calculations tell us our x and z coordinates based on how far the current viewpoint is around a circle centered on the wreck. Typical radii of generated paths in our system are between 25 and 30 units. That circle is sometimes multiplied by a small factor in the x or z direction to create a semi-elliptical path. Variation in the radius is also typically added to create more interesting paths and increase the degrees of freedom. This can be seen in Figure 4.2, which shows

the top down view of a path generated by our system. The height is constrained to be above the seafloor and below a specified maximum height. Once a new node is generated, we must evaluate the scene from that node's position and direction.

4.6 Viewpoint Evaluation

The process of evaluating the camera viewpoint specified in a given node is complex. The actual node object must be passed from the PRM generation code to the OpenGL code, rendered using OpenGL and written out as an image. Next the image must change formats and be passed to OpenCV code to actually evaluate the image. This calculated weight must then be set on the node and used in the node selection process for the next iteration of the PRM loop.

4.6.1 OpenGL Scene

The 3D underwater scene includes sand, rocks, seaweed, bubbles, caustics, and the X127 shipwreck model. This scene is rendered using OpenGL and described thoroughly in Section 4.9. Once a node is passed to the OpenGL code via return parameter, the node's position and direction vectors are used to set a virtual camera in the scene. In order to render the scene, all model data must be transferred to the graphics processing unit (GPU) then rendered from the specified position. The resulting scene is then written to a framebuffer and transferred to the OpenCV image format.

4.6.2 OpenCV Image Evaluation

Once the scene is rendered from the node's viewpoint and in OpenCV's image format, mat, computer vision algorithms are run on it. We are looking to see how closely the image follows the rule of thirds and how many model normals are facing the camera.

4.6.2.1 Rule of Thirds Detection

One of the canonical rules of aesthetics from a cinematographic perspective is the rule of thirds as described in Section 2.2. For automatic camera path generation, this means a good view of the scene should follow the rule of thirds. Thus our algorithm must be able to evaluate if a given view follows this rule.

Methods exist in the OpenCV library that allow for drawing bounding boxes and circles based on contours in an image. These methods were used to help evaluate the rule of thirds in our system. The existing methodology begins by converting an input image to grayscale and blurring it. This is done to reduce noise and detail so shapes can better be detected. Next OpenCV's threshold() function is called to detect edges in the image and the output of this function call is passed to the findContours() function. This function finds the contours or outlines of the shapes. The contours are then approximated as polygons with the approxPolyDP() function and bounding rectangles and circles are formed with boundRect() and minEnclosingCircle(). An example of this methodology can be seen in Figure 4.3.

To use this methodology in our program, we must make a number of modifications. First, the level of homogeneous blur and threshold values must be adjusted for our scene. A kernel size of five by five is used in the normalized box filter. Bounding circles are completely eliminated and only one bounding rectangle is used. All generated bounding rectangles are searched through and the largest one is stored while the rest are discarded. The y coordinate of the top of this largest bounding rectangle is found and used in the thirds calculation. The absolute value of this y coordinate minus the y value of the top horizontal 1/3 line of the image is calculated. This determines how close the bounding box is to following the rule of thirds as seen in Figure 4.4. This measure is set as the weight of the node to be used in node selection. Using this

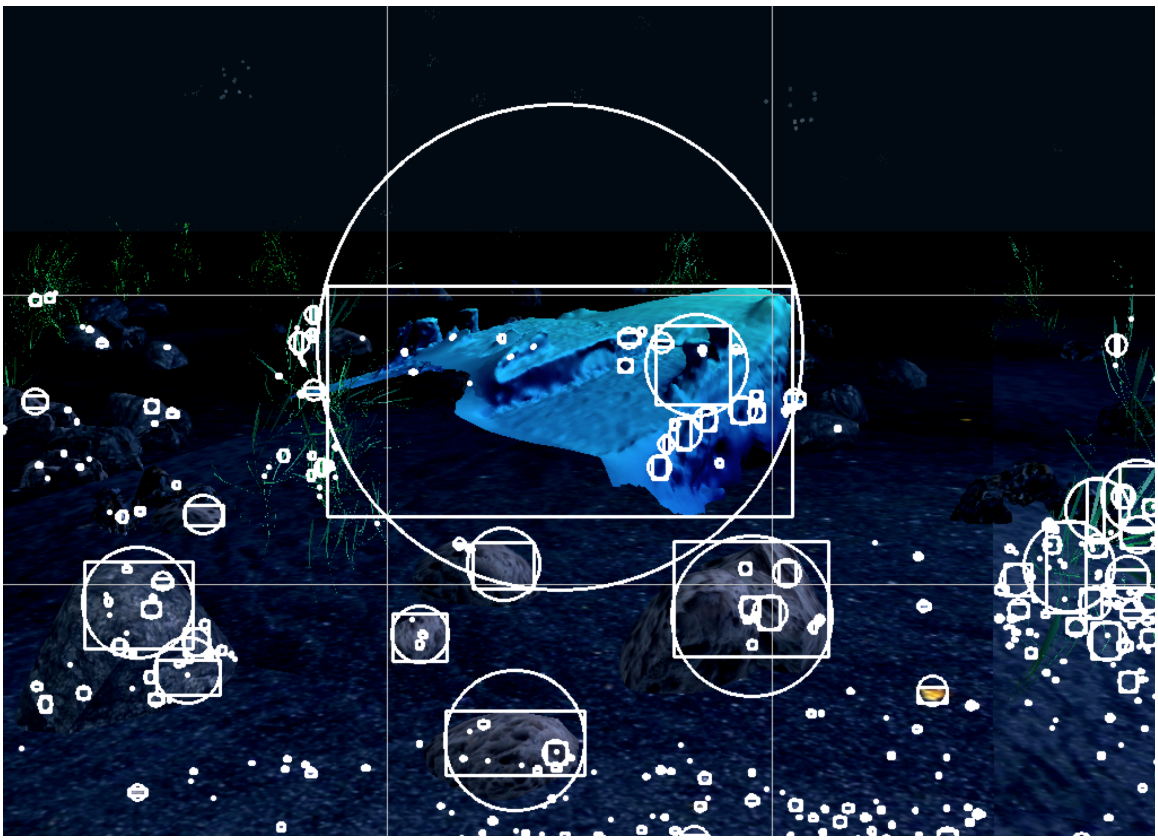


Figure 4.3: OpenCV methodology to draw bounding rectangles and circles around detected contours in an image.

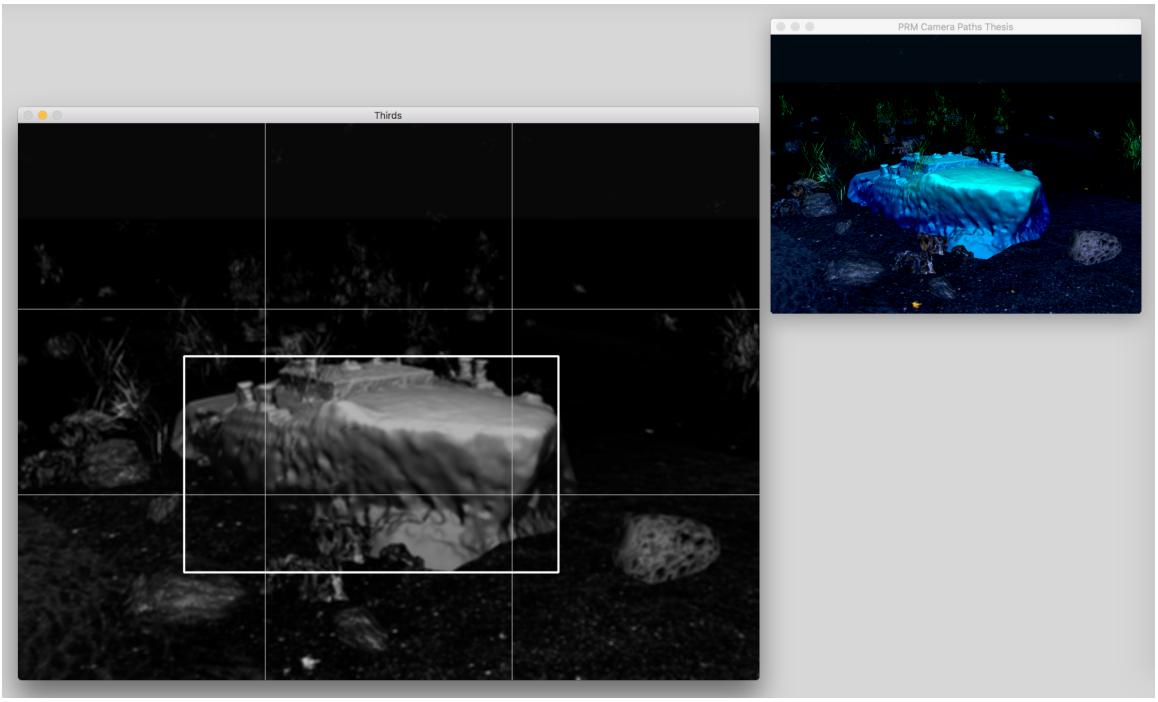


Figure 4.4: Rule of thirds detection. On the right the OpenGL scene is rendered and on the left, OpenCV is used to detect the largest shape in the image and compared to the thirds lines of the image.

calculation means a smaller weight is better because it indicates the wreck is closer to the top thirds line of the image.

We chose to measure the rule of thirds this way based on the results of our user study discussed in Section 6.1. Our use of the largest bounding rectangle comes from the observation that if the shipwreck is shown in the image then it will create the largest bounding box. If the shipwreck is not visible from the selected node's viewpoint then a rock or other scene feature will create the largest bounding box and be used in the rule of thirds calculation. A limitation of our method as discussed in Section 5.3 is that we can only have one main model in our scene for our detection algorithm to work well. Caustics were turned off for this section of our algorithm to increase accuracy of the bounding box detection.

4.6.2.2 Model Normals Detection

From a geometric perspective, we want camera views that show us as much of the model as possible. The node weight calculation detects how much of a given object is pointing towards the camera via use of the model's normals. Specifically, a vertex shader is used to transform the normals into the camera's eye space. When these normals are then used to color the rasterized fragments, a normal that is pointing primarily in the direction of the camera will be colored blue. OpenCV is used to detect the proportion of pixels that have a blue component above a certain threshold via the inRange() function. This can be seen in Figure 4.5. The portion of pixels that have a high blue component compared to the rest of the pixels is used as the weight of this node. The algorithm prefers more model normals facing the camera, meaning that higher weights are better weights. In this part of the algorithm, only the shipwreck is rendered because increasing the normals facing the camera from the rock models or seafloor will not increase the aesthetic quality of a generated path.

4.6.2.3 Thirds and Normals Combination

The rule of thirds and normal detection both have their own shortcomings so we combine them to try to get the best path possible. The rule of thirds is optimal for creating smooth paths but can view the wreck from really bad angles at times. A super low viewpoint, almost from the seafloor can create a path following the rule of thirds just as well as one from a higher, more scenic view. The normal detection does a much better job with this, but can create very jumpy paths that leap from one extreme angle to another. We had to find a way to combine the thirds scores that are better if they are lower and normals scores where a higher score is better. The thirds score cannot be higher than 1 since it is the absolute value of 1/3 subtracted from a number between 0 and 1. This knowledge is used to generate a score where a

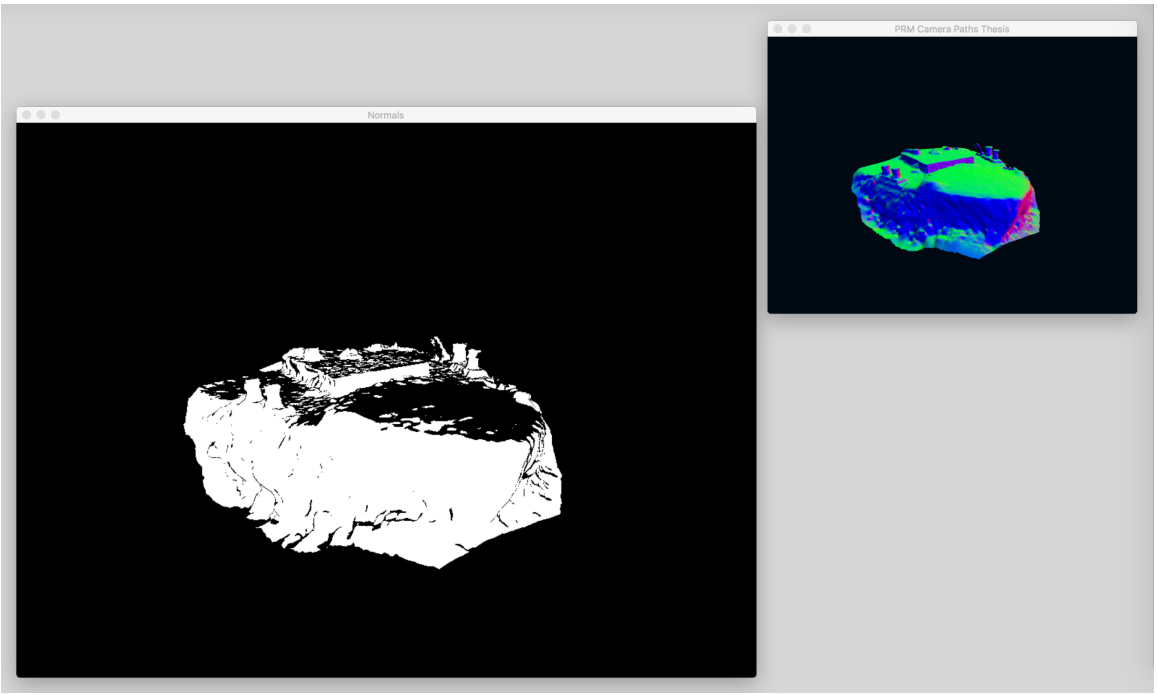


Figure 4.5: Object normals detection. Shipwreck is rendered from a specified node viewpoint with object normals facing the camera in the upper right window, OpenCV detects normals facing the camera in the left window.

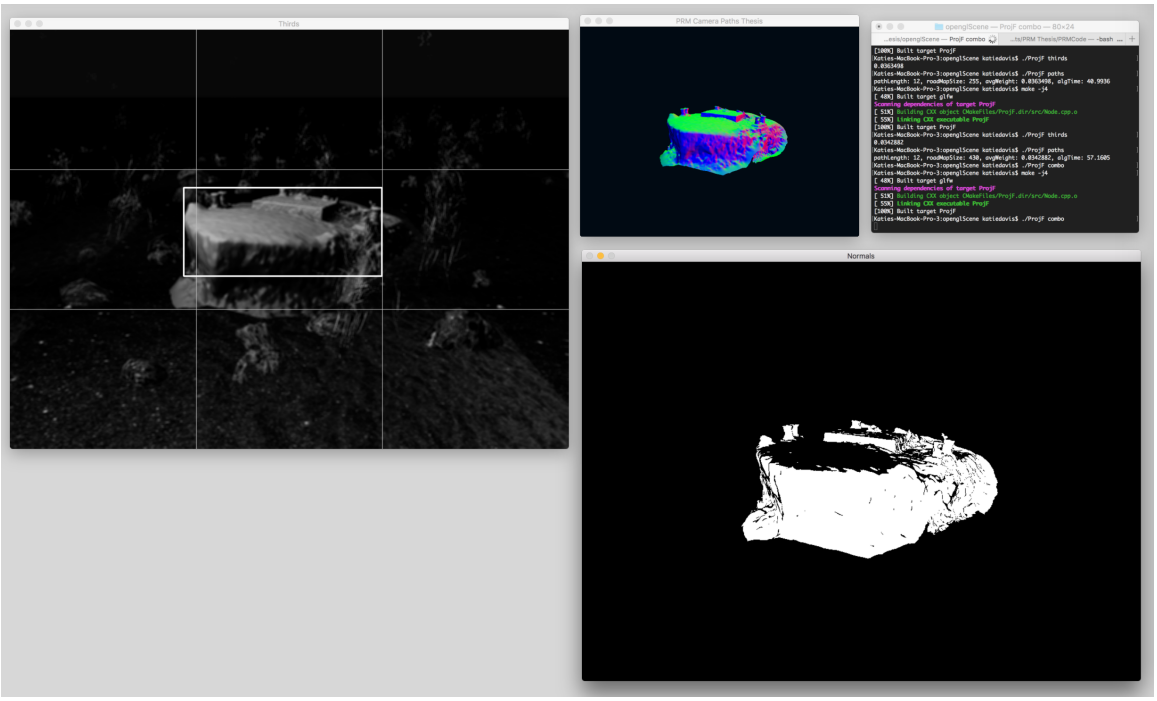


Figure 4.6: Combination node weight calculation. Rule of thirds and model normals are detected for each viewpoint and the combined weights are set as the node weight.

larger number indicates a better score by subtracting the calculated thirds value from 1. The normals score is then added to this rule of thirds calculation result to create a higher score that combines the two methods. An example of a node viewpoint being evaluated with both methods is shown in Figure 4.6. This combination method was noticeably slower because it required rendering the scene, converting the image to OpenCV format, and running OpenCV algorithms twice for each node viewpoint.

4.6.3 Node Weight & Best Weight List

For all node weight strategies, a threshold for "high" weight nodes has to be chosen. These selected nodes were put on the highWeightNode list that was used during the node selection process. For the rule of thirds, nodes with weight 0.04 or lower were added to the best weight list while nodes with weight 0.25 or higher were considered good for normals scores. Since these methods were combined by subtracting the

thirds score from 1 to create positive values, a good combination score was defined as (1 - 0.04) + 0.25 = 1.21.

4.7 Storing Complete Paths

Each time a new node is added to the roadmap, the algorithm checks if that node completes a path of the desired path length. If a full length path is found then the roadmap is traversed from the newest node back up to the root node. This path must then be reversed so it is in the correct order that starts from the root node. Position and direction vectors of each node in the path are then written out to a file so the path can be re-created later. Statistics about the path such as path length, roadmap size, average node weight, and time to generate path are also written to the file.

4.8 Path Playback from File

Through command line parameter specifications, the OpenGL code can be used to playback a path from a file instead of generating a path. Each node is read in from the file and stored in an array. A full list of camera positions and directions is generated by creating a Catmull-Rom spline between the nodes. This playback feature was used to create the videos shown in the user study discussed in Section 6.2. A spline path interpolating between nodes in a generated path can be seen in pink in Figure 4.7.

4.9 Virtual World Description

The sample virtual world used in this path generation system has many complex elements in it. Rocks, seaweed, and bubbles are randomly placed throughout the scene. There are 500 rocks strewn throughout the scene from a combination of four differently shaped rock models and three rock textures. One hundred instances of a

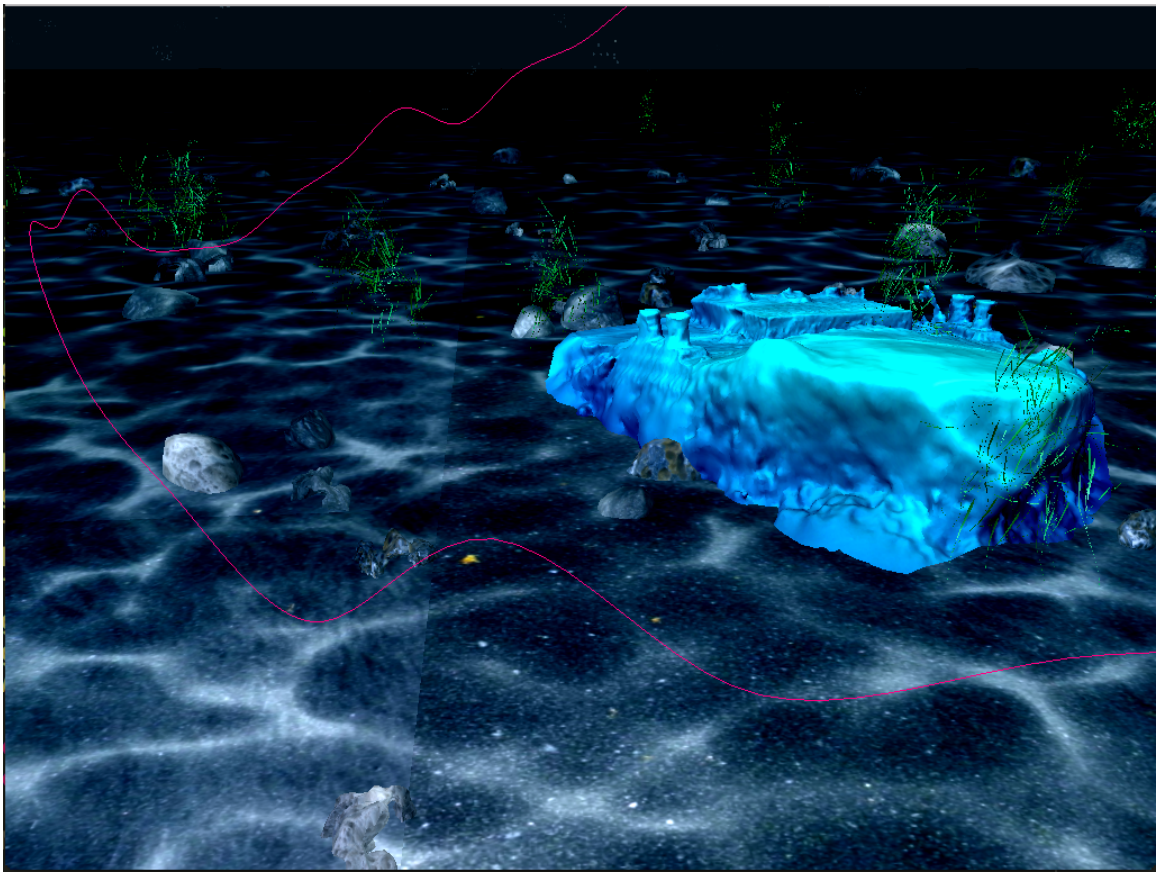


Figure 4.7: Shipwreck scene with generated spline path shown in pink.

seaweed model are placed in the scene. A sand texture is tiled onto the seafloor and caustics play over the entire scene. There are 31 different caustics textures that create a complete loop. Every time something in the scene is drawn, the current caustics texture is drawn on top of it. Different caustics textures are interpolated between to make smooth, constantly changing caustics. Bubbles float up at various intervals throughout random locations in the scene. The seaweed slowly waves back and forth simulating a gentle ocean current flowing through the scene. An example of all these elements in the scene can be seen in Figure 4.8.

The digital shipwreck model was created from an actual shipwreck in Malta, the X127, as described in Sections 2.3 and 2.4. The geometry of the wreck is loaded into the program from a .obj model file. The shipwreck texture is loaded from a .jpg file and applied to the model through texture coordinates. Since the X127 wreck lays on a diagonal, the deeper part of the wreck sits in water that is too murky for the AUV to take high quality video footage. Therefore, only the bow and front portion of the wreck are shown in the model. Based on historical drawings, we estimate that the front quarter of the ship is shown in our digital model. Close up views of the wreck are shown in Figure 4.9

4.10 Implementation Details

Implementation of this work was done in C++ using the Open Graphics Library (OpenGL) application programming interface and the Open Computer Vision (OpenCV) library. It was run on a 2015 MacBook Pro running macOS Sierra 10.12.3 with a 2.9 GHz Intel Core i5 processor. Average run times and performance details are given in Section 5.1.

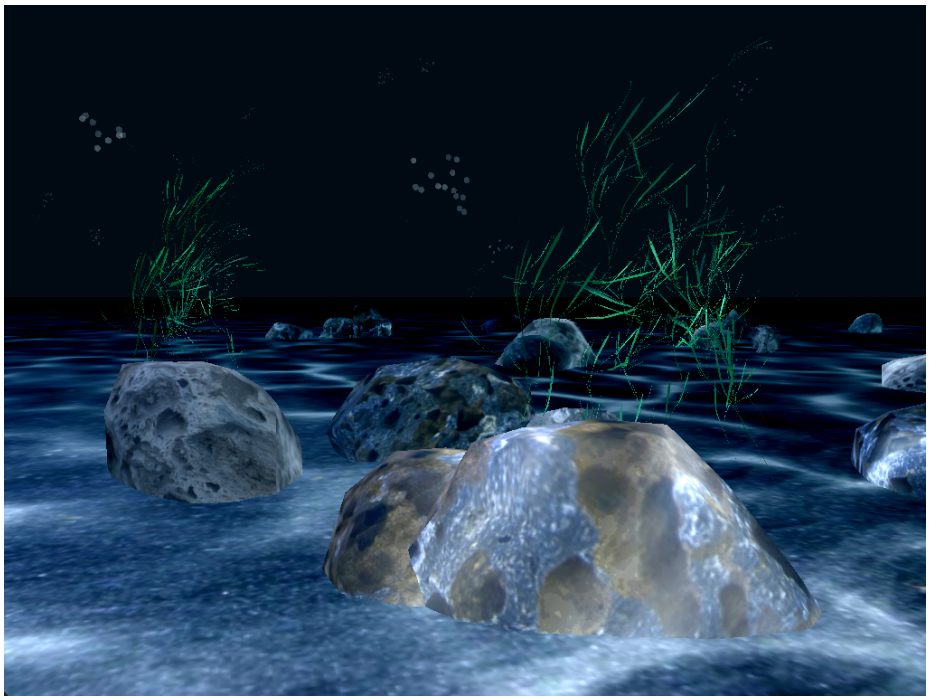
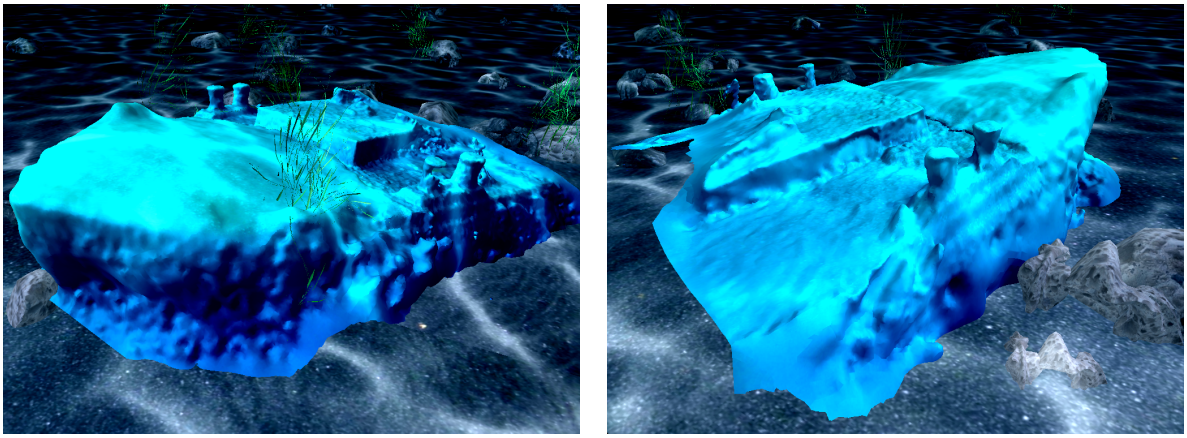


Figure 4.8: Rocks, seaweed, sand, bubbles, and caustics displayed in OpenGL scene.



(a) (b)

Figure 4.9: Close up views of the X127 shipwreck model.

4.11 Summary

This chapter gave specifics of the camera path generation system provided in this work. A probabilistic roadmap algorithm adds nodes to a tree until a path of sufficient length has been generated. A node to expand from is selected, then new nodes are generated given precise parameters. Virtual camera viewpoints, specified in new nodes, are evaluated on cinematographic and geometric principles, i.e., the rule of thirds and model normals, and assigned a weight. These weights are used in the node selection process, generating aesthetically pleasing camera paths.

Chapter 5

RESULTS

This thesis presents methodology to automatically generate visually pleasing virtual camera paths. It uses a PRM algorithm employing cinematographic and geometric principles for viewpoint evaluation. During development, one of our challenges was generating consistently good paths. Some of our paths cut off half the shipwreck, went under the seafloor, or were so bumpy they made you sick. However, a lot of the paths were smooth and presented interesting viewpoints of the shipwreck. In order to perfect these paths, we had to introduce more and more parameters and tune them properly. These parameters include:

- Path length
- Initial height
- Initial pitch
- Height delta
- Pitch delta
- Path radius
- Elliptical factor of path
- Variation in radius
- Number of spline iterations between nodes (affects speed of camera)
- Blue threshold for counting a normal as pointing towards the camera
- Blur and threshold value when detecting contours for the rule of thirds

- "High weight" list threshold for thirds, normals, and combination
- Frequency of selecting from high weight list versus random selection
- Threshold when selecting a random node to expand from for thirds, normals, and combination
- Delta when changing the threshold for selecting a random node
- Number of iterations used to find the best root node

With so many different parameters to tweak, it was impossible to test all of them so we choose only a few to change for our user study in Chapter 6. The other parameters we set to our best judgement of a "good" value and kept that value constant through all testing. One study discussed in Section 6.2.1, keeps all parameters the same and varies the method used for calculating node weights. It compares the rule of thirds, model normals, combination, a completely flat path, and a completely random path. The second study discussed in Section 6.2.2 uses a combination of thirds and normals to calculate node weights and instead varies parameters about the path. The parameters used are summarized in Table 6.3 and include path length, radius, deltas, elliptical factor of path, and more. The camera viewpoints in the best generated path from both studies can be seen in Figure 5.1.

5.1 Performance

All motion planning algorithms can be time and memory consuming to run, with the PRM being no exception. One way to generate a smooth camera path is to put so many nodes on a circular path that each viewpoint is extremely close together. The camera will flow smoothly through node transitions due to the short distance between viewpoints. We attempted this in our initial implementation, generating

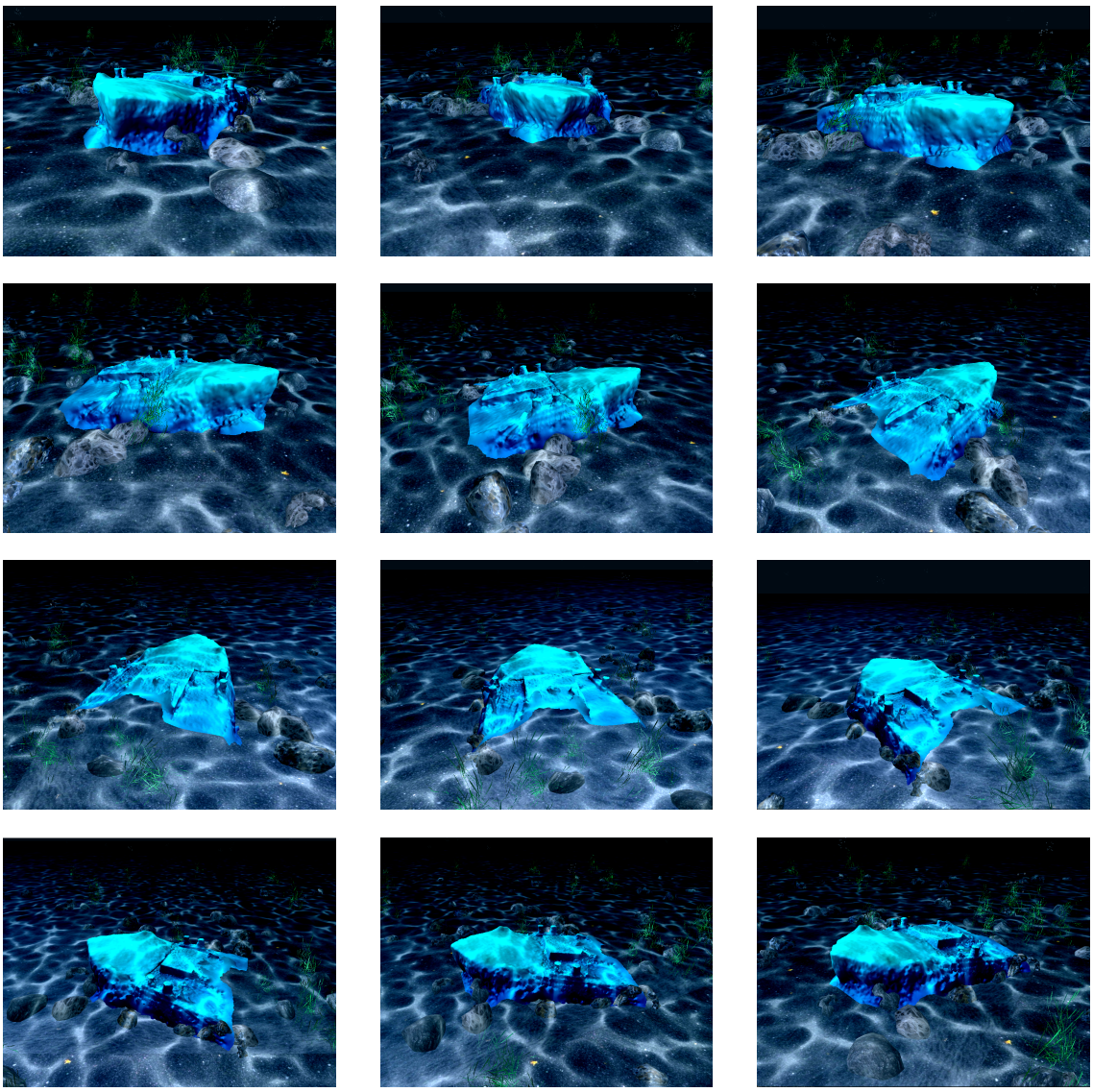


Figure 5.1: Camera viewpoints used in the winning path in the user studies. A spline interpolates between these views to create a 15 second fly-through of the scene.

paths of length 40 in about 10 seconds with full roadmaps containing over 20,000,000 nodes. The time it took to generate longer paths went up exponentially, taking 30 seconds to generate a path of length 50 and a path of length 60 could not be generated in five minutes. We experimented with many different optimizations such as:

- Limiting the number of children each node could have.
- Spatially hashing the configuration space and limiting the number of nodes in each spatial range.
- Selecting a random point in the configuration space and expanding off the nearest node to it.

An alternative way to get smooth paths is to generate splines between fewer viewpoints. We decided to use splines in our final implementation because we could use fewer nodes in a path but get smoother transitions. Fewer nodes in a path meant lower time and memory costs, allowing us to generate and compare a larger number of paths. The final algorithm uses around 12 nodes to generate a path. This decision is justified by the assumption that a movie director creating an overview scene like our circular fly-through would only place a few key viewpoints and let the camera operator move smoothly between these views. The evaluation type, path length, average node weight, roadmap size, and time to generate the path for the paths evaluated in our user study, excluding the flat and random ones, are summarized in Table 5.1. Typical roadmap sizes vary between a couple hundred nodes and a thousand nodes depending on the path length. The time to generate a path is typically between 10 seconds and one minute.

Table 5.1: Summary of best paths from each evaluation scheme. Average Weight is the average weight of all nodes in final path. Roadmap Size is the number of nodes in the entire roadmap and Time to Generate is time spent generating the roadmap in seconds.

Evaluation Type	Path Length	Average Weight	Roadmap Size	Time to Generate (s)
Thirds	12	0.0131	1134	38.5
Normals	12	0.1681	202	6.4
Combination	12	1.0467	663	36.4
Combination	12	0.99642	243	19.5
Combination	45	0.9858	1080	48.8
Combination	10	0.993663	56	12.4
Combination	12	0.917036	238	19.4

5.2 Path Deltas

Height deltas between nodes for each path in the user studies were calculated. Table 5.2 summarizes the minimum, maximum, and average height delta of each path. Excluding the flat path which has 0 delta, the smallest minimum delta was path 2C with a 0.021 delta and the largest maximum delta was 3.892 for path 1A. The average deltas were all relatively close with the smallest average being 0.489 for path 2B and the largest average of 1.600 for path 1A.

5.3 Limitations

Our fly-through generation system has a number of limitations that will be improved in future iterations of this project. First, our system only works on scenes with one central model because our rule of thirds detection algorithm only looks at the single largest shape. Similarly, our rule of thirds algorithm always detects something so if the shipwreck is out of view it might select a rock for use in the calculation. Our

Table 5.2: Height deltas between nodes from paths in the user study.

Video	Path Type	Min Delta	Max Delta	Average Delta
1A	Thirds	0.158	3.892	1.600
1B	Random	0.071	2.266	0.961
1C	Combination	0.349	2.151	0.993
1D	Flat	0.0	0.0	0.0
1E	Normals	0.099	2.118	0.901
2A	More variation	0.196	2.302	1.040
2B	Many nodes	0.030	0.996	0.489
2C	Circular, small variation	0.021	1.284	0.697
2D	Elliptical	0.030	1.863	0.764
2E	Random	0.307	2.277	1.079

general algorithms could work for many scenes, but most all of the parameters listed above would need to be adjusted for a different scene.

Chapter 6

VALIDATION

We performed three separate surveys, hoping to gain information about different aspects of this virtual camera path generation project. The first was an early stage study using still images hoping to see people's preference with regards to the rule of thirds. The second two surveys were performed after all coding was completed, and aimed to validate participant's preferences about generated camera paths. One of these surveys focused on different viewpoint evaluation methods, and the other focused on types of paths. These studies were distributed via Google Forms and sanctioned by the Cal Poly Human Subjects Committee. Participants in all surveys consisted primarily of Cal Poly students but included other family and friends.

Ten total camera paths were generated by our system for use in these studies. Random and flat paths were used as control paths. The other paths were chosen algorithmically to avoid selection bias. Thirty potential paths were generated for each video in the studies and the path with the best weight given the current evaluation criteria was used.

6.1 Rule of Thirds Evaluation

In January 2017, we collected a survey focused on preferences of images from a scene of the X127 shipwreck early in the development cycle. We had 30 participants in the survey. Our goal was to see if people truly liked images that followed the rule of thirds as described in Chapter 2. The full survey questionnaire is shown in Appendix A. Two sets of three similar images were given. One set of images displayed the starboard (right) side of the wreck, viewed from a forward, top perspective. The second set

Table 6.1: Shipwreck images survey results. Survey was designed to find preference with regards to rule of thirds. Two sets of images were ranked 1 (most preferred) through 3 (least preferred) by 30 participants. Frequency of each preference ranking for each image is shown in columns 2 through 4 and average score is shown in column 5. Image 2 and Image B were the most preferred images with the lowest average score.

Preference	1 (Most)	2 (Middle)	3 (Least)	Average
Image 1	6	11	13	2.233
Image 2	22	6	2	1.333
Image 3	2	13	15	2.433
Image A	9	14	7	1.933
Image B	18	11	1	1.433
Image C	3	5	22	2.633

of images showed the port (left) side of the wreck from a closer viewpoint, looking slightly down on the wreck. Each set of images had the ship aligned vertically in the middle of the scene, near the top of the scene, and near the bottom of the scene. The images were framed with the shipwreck close enough to the camera that the top of the wreck formed a pseudo horizon in the scene. In the scene, the seafloor is covered with sand, rocks, seaweed, and caustics. The background of the scene is completely black with bubbles floating up occasionally from random locations on the seafloor. Survey participants were asked to rank each set of images by preference with 1 being most preferred and 3 being least preferred. The frequency of counts and average of all participant rankings is shown in Table 6.1. We found that people preferred the second image from each set where the shipwreck created a horizon line at the top thirds line of the image. The winning images with the best average scores of 1.333 and 1.433 respectively can be seen in Figures 6.1 and 6.2. We used these findings to implement the rule of thirds detection employed in our path generation algorithm.

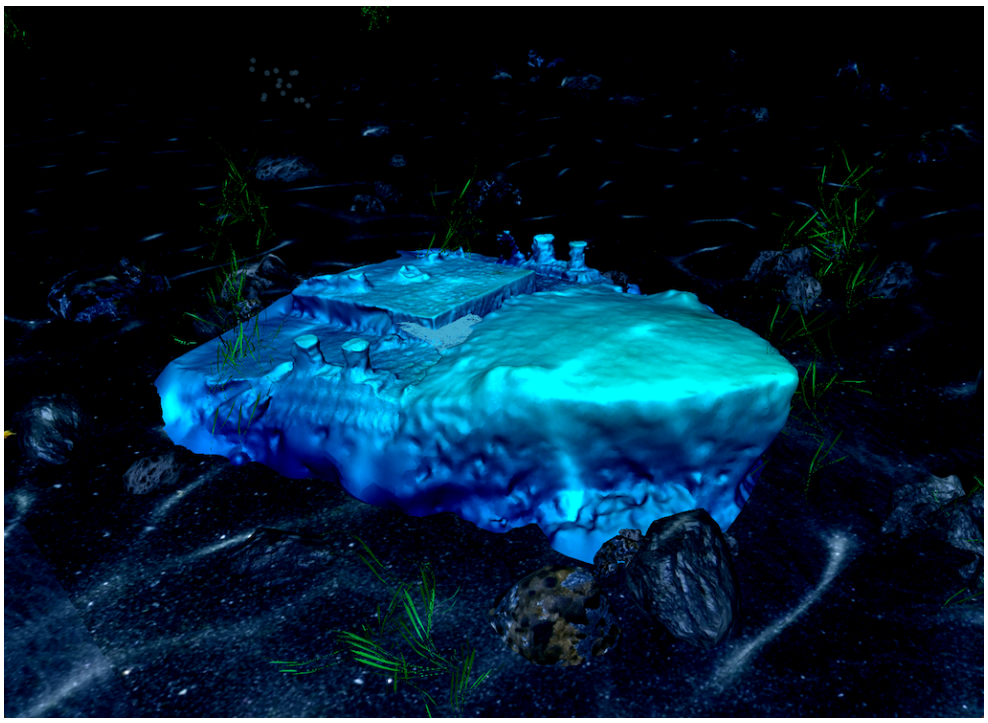


Figure 6.1: Winning image from set 1 of rule of thirds study, Image 2.

Figure 6.2: Winning image from set 2 of rule of thirds study, Image B.

6.2 Camera Path Videos Evaluation

In March 2017, we performed a second set of user studies focused on validating the camera paths our program outputted. This study was reviewed and accepted by the Cal Poly Institutional Review Board. Study participants were primarily collected through Cal Poly Facebook pages. Participants completed an informed consent form before participating and were kept anonymous. Some participants completed only one survey and some participated in both. In each survey, participants were asked to rank five 15 second videos in order of preference. They were given an opportunity to provide comments after each video and final comments at the end. Full surveys are available in Appendix B and survey videos can be watched at `https://www.youtube.com/channel/UCUH0rkZ8iDLe9LjA2n8DPCA`.

6.2.1 Viewpoint Evaluation Methods Study

This study focused on evaluating camera paths that were generated via different evaluation methods. For this survey, path parameters were kept constant and only node evaluation criteria were changed. All units are based on this specific OpenGL scene. Discussed in Section 4.5, 1 meter in the real world is about 1.6 units in our virtual world. All paths in this survey had length 12 and were generated with a radius of 26 that was allowed to vary by plus or minus 1. The paths had a slightly elliptical path that was 0.75 width in the z component. The initial height varied between 4 and 12 and the initial pitch angle varied between 45° and 0° below the horizon line. The max height delta possible was 2.5 and the max pitch change was 7.5°. The thresholds of what defined a "good" node discussed in Section 4.4, were kept constant throughout all path generation for both surveys. The viewpoint evaluation methods used to produce each video are as follows:

Table 6.2: Viewpoint evaluation method study results. 1 is highest or most preferred ranking and 5 is worst or least preferred. Columns 2 through 6 show the number of participants that gave each video that ranking and the last column shows the average score for each video. Video A was the highest ranked with an average score of 2.14.

Preference	1	2	3	4	5	Average
Video A (Thirds)	13	10	8	2	2	2.14
Video B (Random)	0	0	2	11	22	4.57
Video C (Combination)	10	5	9	8	3	2.69
Video D (Flat)	7	11	10	5	2	2.54
Video E (Normals)	5	9	6	9	6	3.06

- Video D was a completely flat path with no variation in height.
- Video B displayed a path that was randomly generated with the above parameters. Each camera viewpoint was not evaluated at all, random nodes were selected to expand off of until the path was long enough.
- Videos A, E, and C used the rule of thirds, model normals, and a combination of both respectively to evaluate potential camera viewpoints.

To avoid bias, 30 paths were generated of each type and the path with the best average node score was chosen for the study with a Python script. Results of the study are displayed in Table 6.2. Thirty-five people participated in the study.

6.2.1.1 Ranking Results

The results show that people preferred the path generated with only the rule of thirds. The second highest path was flat, followed closely by the combination path. Normals came in a solid fourth with random at a strong last place. The results were

surprisingly spread out, with all paths except random having at least two rankings for every possible position.

6.2.1.2 Video Comments

Quite a few people left comments, averaging 12.8 comments per video and 8 "last thoughts" comments at the end. Comments for video C were the most positive including "Nice and smooth", "fav", and "Camera moves a little far away from the ship, but movement feels natural for underwater". There were some conflicting comments such as "I like how this one feels slower which is nice" and "This feels like the camera is moving faster than both A and B and thus is more dynamic". The paths were all set to take the exact same amount of time to travel, so no videos were actually faster than any others.

Even though video A scored the highest, it has some less positive comments like "The beginning and middle make sense. From a cinematic perspective, it seems like an establishing shot of a thing (this looks like the from of a sunken ship). It gets a little strange towards the end, bobbing up and down", "Smooth with good in and out movement. The angle change was a little abrupt towards the end", and "huh".

Video B was the randomly generated one, ranking lowest, and the shipwreck gets partially cut off for a section of the video. This was noted in many comments like "Too much of the wreck is cut out of the frame during the path", "Not great as it lost sight of the object", "Camera wasn't looking at ship the whole time", and "Gets cut off:(". The camera viewpoint used in this path can be seen in Figure 6.3.

The flat path earning the second highest score, video D, received a variety of comments including "Fast and boring", "Feels a little boring/flat but good", "This is a great looking, low, and dramatic shot. The subject stays in frame the entire time and no significantly noticeable bobbing makes it feel out of place", and "Good

consistent view of the object. Didn't present the object at different angles, but the consistency was nice".

Video E, generated using normals and ranking fourth also received varying comments such as "The bobbing in this makes the shot awkward. The subject (sunken ship) stays in frame, but drifts between more of the center (where it probably should be) to more of the top (suggesting that something below the ship in the frame is the subject for a moment) and back (which then just seems strange)" and "Like this the best; differing camera height and pitch makes the reel feel more 'alive' with diverse perspectives".

6.2.1.3 Additional Comments

Overall, people seemed to have positive feelings towards this project. In the additional comments section someone wrote "This is neat. I can see some applications of this in the field of architecture and design. Building 3D models and auto-generating appropriate cinematic views for presentation of the model can save designers a lot of time". Another comment said "I generally preferred the animations with smooth paths, but the shaky-path animations had a "hand cam" quality that I might have enjoyed more if the subject was more appropriate to that style (e.g. an animation of a subject doing something active, like biking or snowboarding)". The last comment left stated "Really cool work. Would love to hear about how you came up with the algorithm". Since one of our viewpoint evaluation techniques, the rule of thirds, won out over flat, with our combination technique as a close third place, and random was the worst by far, we conclude that users prefer our path generation techniques over no evaluation technique as in random and flat.

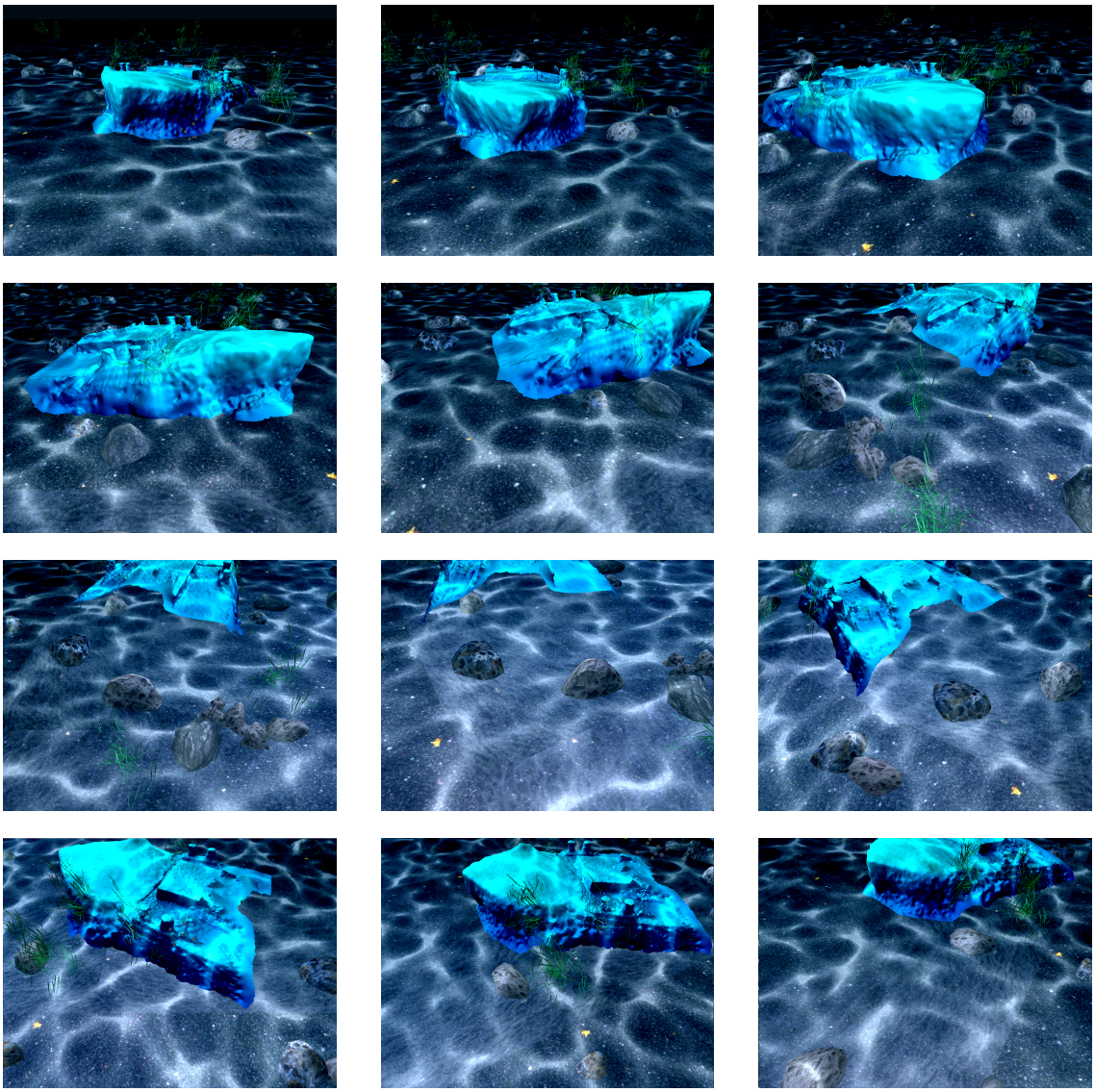


Figure 6.3: Camera viewpoints used in the random path from viewpoint evaluation method study, video B. Shipwreck model gets cut off in many of the views.

6.2.1.4 Path Selection Note

It is worth noting that the "best" path chosen by the Python script from the 30 generated videos is not necessarily the path we think looks best. Just because the path has the best score from the evaluation technique, does not mean that it is necessarily smooth and shows the wreck from interesting angles. As mentioned in the comments for video A, some of the paths jump unnecessarily up and down. We could eliminate this by choosing smaller height and pitch deltas but then we would limit the possible angles we could see the wreck from. One of our future work goals is to add a measure of jumpiness to paths. The more a path jumps up and down, the lower score it would receive.

6.2.2 Path Types Study

This study kept the viewpoint evaluation method constant, using a combination of rule of thirds and model normals. The variable being studied here was path type. Videos A through E respectively displayed paths with large variation, many nodes, small variation and perfectly circular, largely elliptical, and random. The relevant parameters used to generate each path are summarized in Table 6.3. Excluding random, 30 paths of each type were generated and the best scoring path of each type was used in the study. Since the random path used no evaluation criteria, one random path was generated and used in the study. A summary of the study results can be seen in Table 6.4. This study had 32 participants total.

6.2.2.1 Ranking Results

This study had much clearer results than the viewpoint evaluation methods study, with almost one entire point between the average score for each video, except the

Table 6.3: Parameters used for each video in the path types study. Video A was a path with large variations, B contained many nodes, C was perfectly circular with small variation, D was extremely elliptical, and E was random.

Video	Nodes in Path	Radius	Radius Variation	Elliptical	Height Delta	Pitch Delta
A	12	27	+-3	x, z	2.5	7.5°
B	45	30	+-1	x, 0.8z	1.0	5°
C	10	23	+-0	x, z	1.5	5°
D	12	26	+-1.5	1.3x, 0.8z	2.0	5°
E	12	27	+-1	x, 0.8z	2.5	7.5°

last two places. Video A, with the largest allowable deltas and radius variations, was the clear winner with 18 of 32 people ranking it number 1 and 12 people ranking it number 2. The camera viewpoints used in this path can be seen in Figure 5.1 in Chapter 5. The perfectly circular path, video C, came in second and video D, with the highly elliptical path came in third. Fourth and fifth place were very close but random, video E, came in fourth while the path with many nodes, video B, came in last.

6.2.2.2 Video Comments

Even though fewer people participated in this study than the last one, there were more comments with an average of 17.4 comments per video. The winning video, video A, received the most positive comments such as "The camera angle was perfect because the blue object was in view the entire time. It was also slow and steady which was easy to look at and gave ample time to view everything within frame", "A good path with nice camera height that shows some details of the topside of the ship, with a nice ending in stable, bottom-up shot of the bow", and "Good balance of seeing the rocks, full landscape, and ship".

Table 6.4: Results from the path types video study. Columns 2 through 6 show the number of participants that put each preference for each video with 1 being most preferred and 5 being least preferred. The last column shows the average score each video received. The winning video was Video A with an average score of 1.50

Preference	1	2	3	4	5	Average
Video A (More variation)	18	12	2	0	0	1.50
Video B (Many nodes)	1	0	2	12	17	4.375
Video C (Circular, small variation)	9	13	9	0	1	2.09
Video D (Elliptical)	3	6	18	4	1	2.81
Video E (Random)	1	1	1	16	13	4.22

Participants noted that they liked video C second best with comments such as "Second best, but I think we don't get to see the top enough", "Nicely controlled and sweeping camera movement. Subject stays in frame. Fewer dramatic moments than A", and "A nice path, but more diversity in camera height as well as a bit overall camera height may reveal more detail on the topside of the ship".

The elliptical path in video D got a variety of comments such as "Path radius seems unnecessarily large", "elliptical orbit is a bit confusing", "I liked the zoom in at the beginning, but don't think it should have zoomed back out after the fly-by", and "Longer range shots are nice".

The random path, video E, received many comments about the ship being cut off including "Centering seems totally off, annoying to not be able to see the object", "Motion was smooth, but it didn't focus on the actual object much", "Subject (shipwreck) seems to be off-camera for too long", and "This one makes me think I should be looking at the sand, yet there's nothing there".

The last place video with a large number of nodes, video B, received the most comments, all about the path being too bumpy. Since the path consists of 45 dif-

ferent viewpoints with a possible radius variation of 1, a possible height delta of 1.5 and pitch delta of 5° this path truly is bumpy. A sample of these comments are "vertical and z motion very distracting", "Way too much jerky dollying movement (forward-backward)", "super shaky", "Too much zooming in and out, kinda giving me a headache", and "The camera was really wobbly and extremely nauseating to watch. I was immediately put off and didn't want to watch the rest despite the video only being 14 seconds".

6.2.2.3 Additional Comments

In the additional comments/feedback section at the end, there was again positive sentiment towards these videos. Comments included "Video A seems to be a "safe" choice for something like an establishing shot, however, I liked D's sweeping, smooth motion more, though it may be better suited to a dramatic establishment shot", "Camera A looked really nice. I'd love to see this on an arbitrary mesh", and "Once again, I prefer the most stable and consistent path".

Some participants noted that they liked the scene with comments like "Love the water and the ship! Would love if the ship were less reflective to see any holes or indents more easily. But that's just a personal preference", "beautiful renders, although the sea floor was a bit tile-y. I wish I could've seen the subject from above", and "The lighting in this scene is really good".

We can conclude that users strongly preferred our generated paths over random ones but did not like paths with a lot of undulation. They preferred paths that kept the shipwreck in focus the whole time but showed it from a variety of angles. Mentioned in the future work, Section 7.1, we would like to add a measure of bounciness to calculating our path weight since users preferred large deltas so they could see the

model from many angles without the camera jumping up-and-down or in-and-out too much.

Chapter 7

CONCLUSION

7.1 Future Work

This thesis introduces a novel fly-though generation system based on cinematographic and geometric principles shown to be preferred by users over flat and random paths. However, it has a number of limitations and areas for improvement. Future work includes:

- Implementation of fly-through path generation system with motion planning algorithms other than the PRM.
- Comparison between Catmull-Rom splines and other types of splines such as B-splines for interpolating between nodes.
- Adding more cinematographic principles to path generation rules such as leading lines and visual balance.
- Add measure of jumpiness to path weight calculation and have jumpier paths receive worse scores as this was a main complaint in the user study.
- Test the fly-through generation system with other scenes and different models.
- Include the kinematic constraints of an AUV in the node generation process so the paths could actually be used by a robot to take footage of shipwrecks.
- Expand PRM algorithm to work in all six degrees of freedom to generate more interesting paths.
- Keep the OpenCV image data on the GPU for faster computation, rather than transferring it to the CPU.

7.2 Conclusion

This work automatically generates camera fly-throughs of single model scenes. This is important because hand generating fly-throughs is a laborious, time-consuming task. The example scene used is a underwater scene focused on a 3D re-creation of the X127 shipwreck located in Malta. The model was created through use of an autonomous underwater vehicle and photogrammetry. A probabilistic roadmap algorithm is used to generate camera paths. The roadmap uses a list of "high weight" nodes to generate more ideal paths. The node viewpoints are evaluated on cinematographic and geometric principles, characterized by the rule of thirds and model normals.

Three user studies were preformed to validate different aspects of this system. An initial user study was done validating the rule of thirds as used in our scene. Two studies were done with videos created from our final path generation system using different viewpoint evaulation criteria for one study and different path types for the other. In the first of these two studies, we found that users preferred the paths generated using the rule of thirds over flat and random ones. They strongly preferred all viewpoint evaluation methods used by our system– the rule of thirds, model normals, and a combination of these, over randomly generated paths. The last study also found that users preferred paths generated by our system over random ones except paths with too much bumpiness that made users feel nauseous. They liked paths with few nodes but large deltas so the shipwreck was visible from a variety of angles without too much camera undulation.

BIBLIOGRAPHY

[1] 10 top photography composition rules. http://www.photographymad.com/pages/view/10-top-photography-composition-rules.

[2] Cal Poly Github. http://www.github.com/CalPoly.

[3] Panning best practices. http://www.red.com/learn/red-101/camera-panning-speed.

[4] Tutorial 12: Quaternions. http://www.xojo3d.com/tut012.php.

[5] Dive sites: X127 Lighter "Coralita" (Manoel Island), 2015. http://www.divesystemsmalta.com/about-malta/dive-sites-x127/.

[6] The X127 water lighter (Coralita), 2016. http://www.dawndiving.com/the-x127-water-lighter-coralita/.

[7] D. B. Christianson, S. E. Anderson, and L. wei He. Declarative camera control for automatic cinematography. *American Association for Artificial Intelligence*, Dec. 1996.

[8] C. Clark. Harvey Mudd College E160 – lecture 13 autonomous robot navigation, June 2016. http://www.hmc.edu/lair/E160/E160-Lecture14-MotionPlanning.pdf.

[9] L. K. Dale and N. M. Amato. Probabilistic roadmaps-putting it all together. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 1940–1947 vol.2, 5 2001.

[10] T. Gambin. A phoenician shipwreck off Gozo, Malta. *The Journal of the Archaeological Society, Malta*, 2010.

[11] D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 4420–4426 vol.3, Sept 2003.

[12] A. Huamn. Creating bounding boxes and circles for contours. `http://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/bounding_rects_circles/bounding_rects_circles.html#bounding-rects-circles`.

[13] N. Joubert, J. L. E, D. B. Goldman, F. Berthouzoz, M. Roberts, J. A. Landay, and P. Hanrahan. Towards a Drone Cinematographer: Guiding Quadrotor Cameras using Visual Composition Principles. *ArXiv e-prints*, Oct. 2016.

[14] T.-Y. Li and C.-C. Cheng. *Real-Time Camera Planning for Navigation in Virtual Environments*, pages 118–129. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[15] T.-Y. Li and Y.-C. Shie. An incremental learning approach to motion planning with roadmap management. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 4, pages 3411–3416 vol.4, May 2002.

[16] L. Liu, R. Chen, L. Wolf, and D. Cohen-Or. Optimizing photo composition. *Computer Graphics Forum*, 29(2):469–478, 2010.

[17] L. Mai, H. Le, Y. Niu, and F. Liu. Rule of thirds detection from photograph. In *Proceedings of the 2011 IEEE International Symposium on Multimedia*, ISM '11, pages 91–96, Washington, DC, USA, 2011. IEEE Computer Society.

[18] M. Rantanen. *Improving Probabilistic Roadmap Methods for Fast Motion Planning*. PhD thesis, School of Information Sciences, University of Tampere, Aug. 2014.

[19] V. Setlur, S. Takagi, R. Raskar, M. Gleicher, and B. Gooch. Automatic image retargeting. In *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia*, MUM '05, pages 59–68, New York, NY, USA, 2005. ACM.

[20] K. Sijmons. Introduction on photogrammetry. `http://drm.cenn.org/Trainings/Generation%20of%20geodatabases%20using%20ARCGIS%20and%20ERDAS/Lectures/Introduction%20on%20Photogrammetry.pdf`.

[21] V. K. Viswanathan, Z. Lobo, J. Lupanow, S. M. T. S. von Fock, Z. Wood, T. Gambin, and C. Clark. Auv motion-planning for photogrammetric reconstruction of marine archaeological sites. In *Proceedings of the 2017 IEEE International Conference on Robotics and Automation*, May 2017.

[22] S. M. T. S. von Fock, S. Bilich, K. Davis, V. K. Viswanthan, Z. Lobo, J. Lupanow, C. Clark, T. Gambin, and Z. Wood. Pipeline for reconstruction and visualization of underwater archaeology sites using photogrammetry. In *Proceedings of the 2017 ISCA International Conference on Computers and Their Applications*, Mar. 2017.

APPENDICES

Appendix A

RULE OF THIRDS IMAGE EVALUATION

Thesis Survey- Shipwreck Images

Please rank each set of images in order of preference. There is no right or wrong answer- this is about what looks best to you!

* Required

1. **Set 1** *
   Mark only one oval per row.

	1 (most preferred)	2	3 (least preferred)
Image 1			
Image 2			
Image 3			

Image 1

Image 2

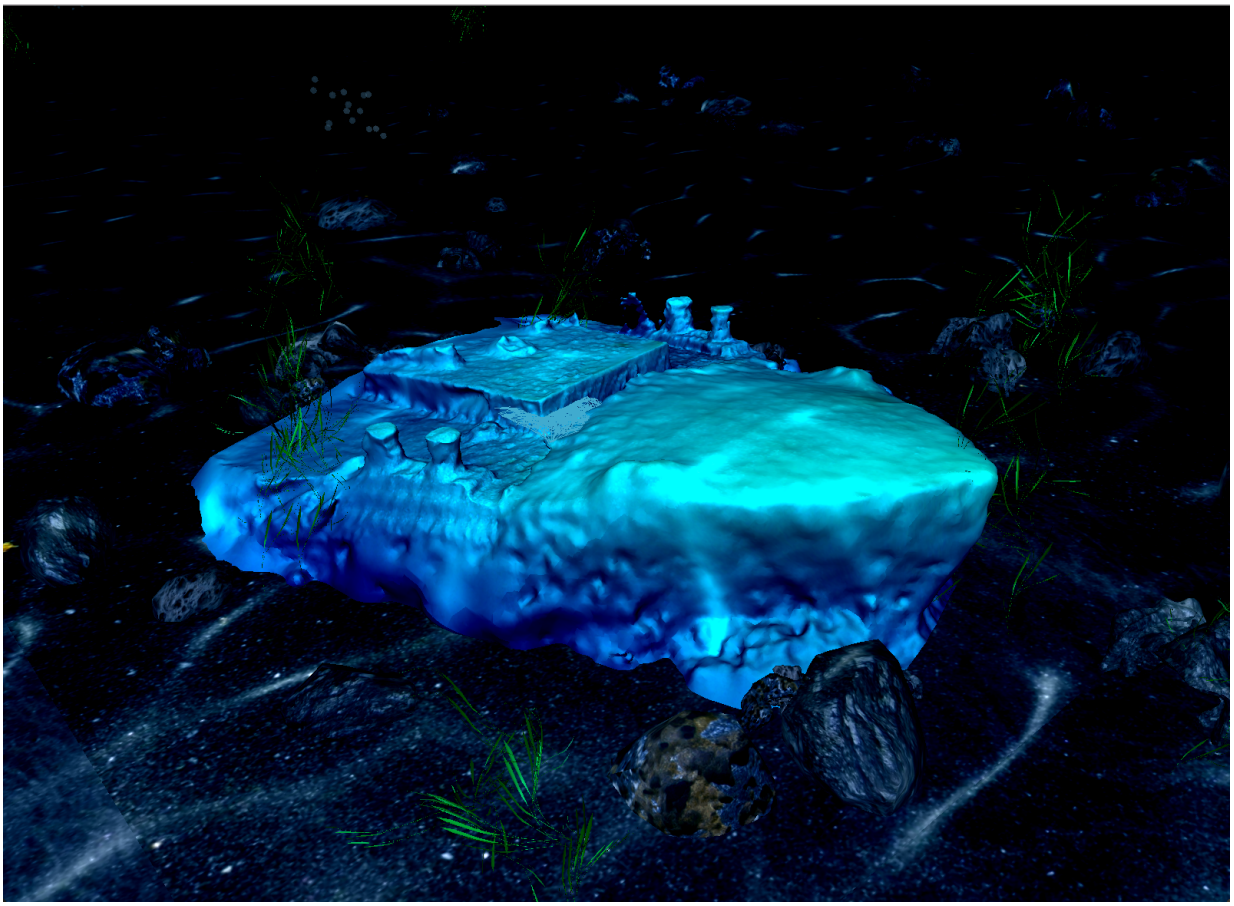
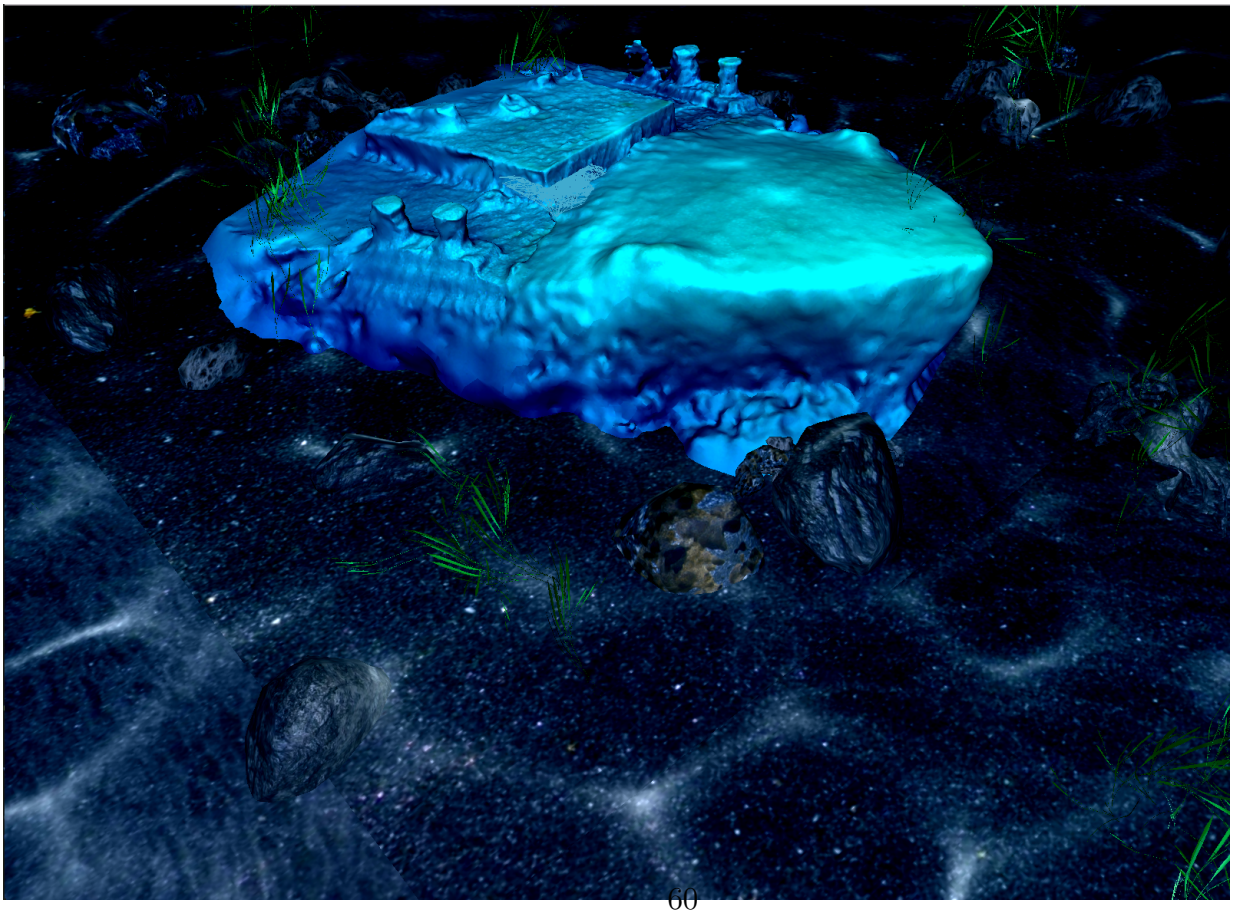


Image 3



2. **Set 2** *

Mark only one oval per row.

	1 (most preferred)	2	3 (least preferred)
Image A			
Image B			
Image C			

Image A

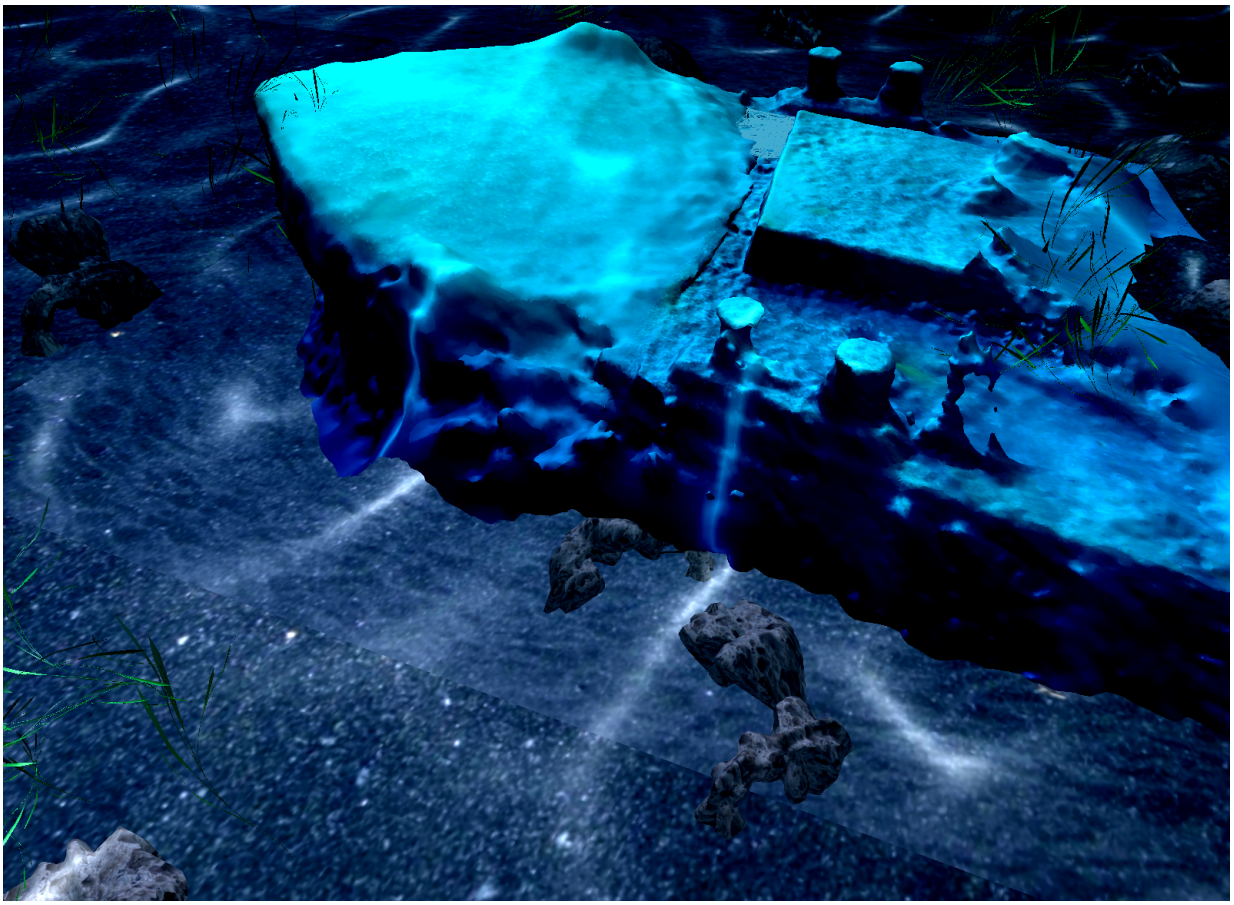


Image B

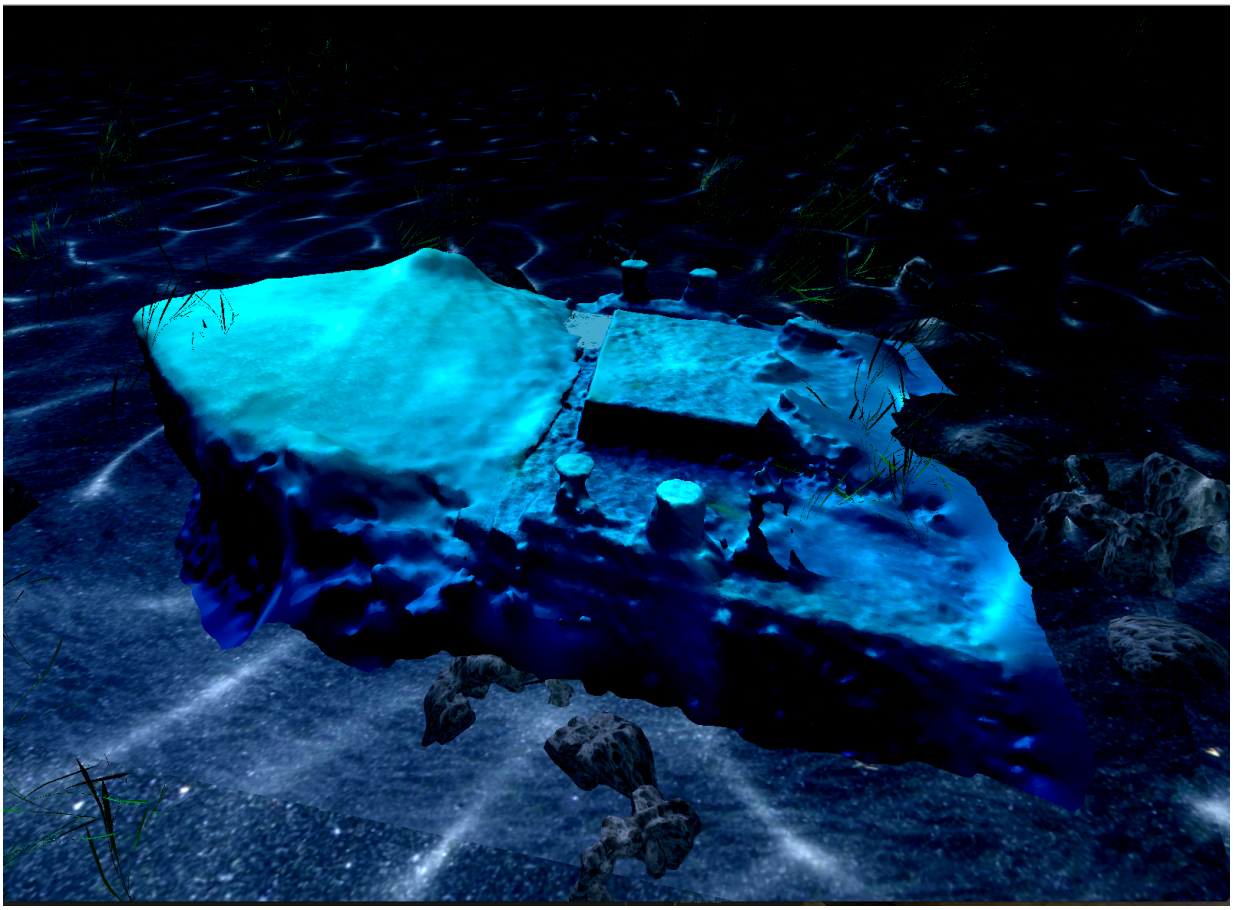
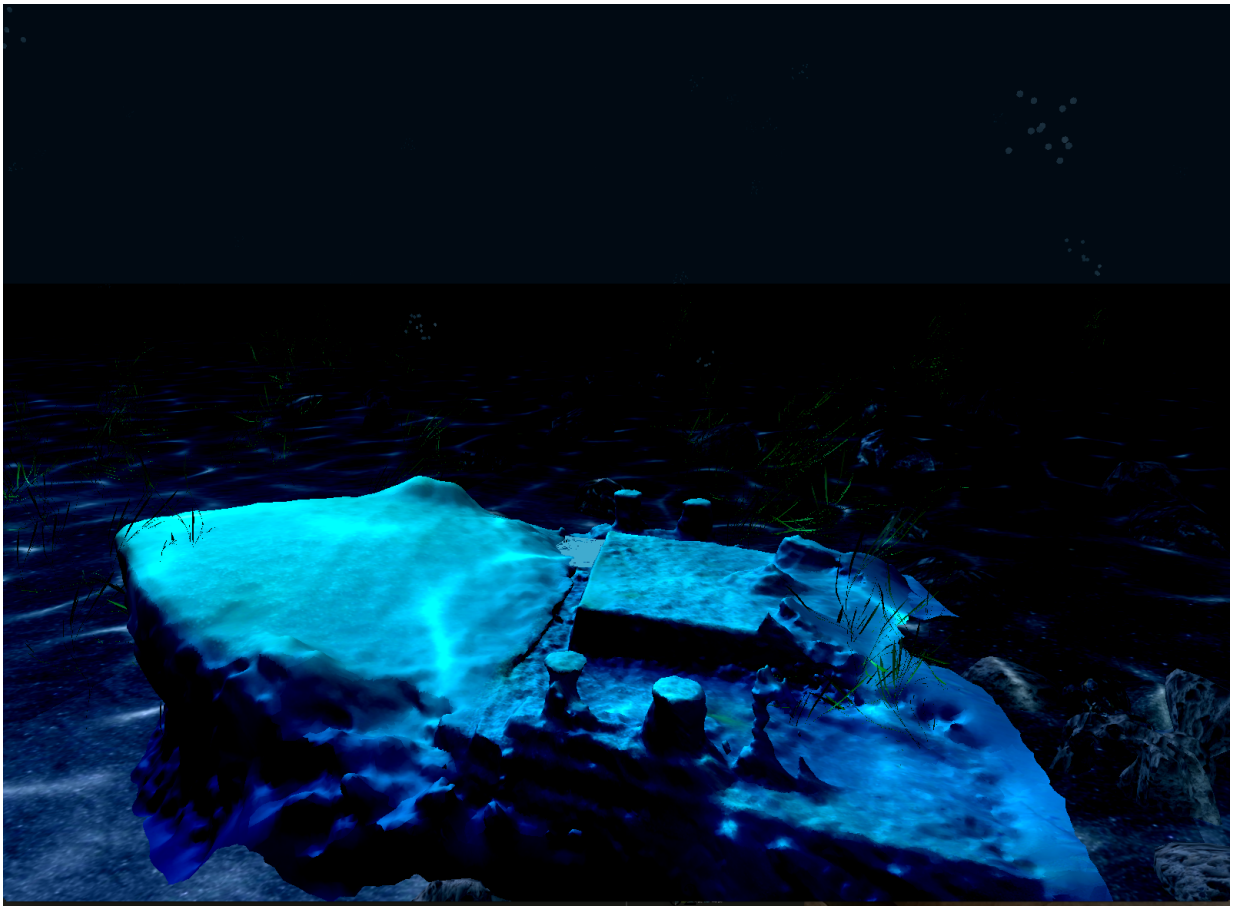


Image C



Appendix B

GENERATED PATH VIDEO EVALUATION SURVEYS

Automatically Generated Virtual Camera Paths: Evaluation Methods

INFORMED CONSENT TO PARTICIPATE IN A RESEARCH PROJECT, “PROBABILISTIC ROADMAPS FOR VIRTUAL CAMERA PATHING WITH CINEMATOGRAPHIC PRINCIPLES”

A research project on using probabilistic roadmaps for generating virtual camera paths with cinematographic principles is being conducted by graduate student Katie Davis and Dr. Zoë Wood in the Department of Computer Science at Cal Poly, San Luis Obispo. The purpose of the study is to validate a thesis focused on computationally creating animations.

You are being asked to take part in this study by completing the following questionnaire. You will be asked to watch a few short videos of less than a minute in length and order them by preference. Your participation will take approximately five to ten minutes. Please be aware that you are not required to participate in this research, you may omit any items that you prefer not to answer, and you may discontinue your participation at any time without penalty.

There are no risks anticipated with participation in this study. Your responses will be provided anonymously to protect your privacy. Potential benefits associated with the study include furthering the understanding of what makes good camera paths.

If you have questions regarding this study or would like to be informed of the results when the study is completed, please feel free to contact Katie Davis at kdavis22@calpoly.edu or Zoë Wood at zwood@calpoly.edu. If you have concerns regarding the manner in which the study is conducted, you may contact Dr. Michael Black, Chair of the Cal Poly Institutional Review Board, at (805) 756-2894, mblack@calpoly.edu, or Dr. Dean Wendt, Dean of Research, at (805) 756-1508, dwendt@calpoly.edu.

If you agree to voluntarily participate in this research project as described, please indicate your agreement by completing and submitting the following questionnaire. Please print a copy of this consent form now for your reference, and thank you for your participation in this research.

* Required

1. **Participation Agreement** *

   Mark only one oval.

   - ◯ Yes, I volunteer
   - ◯ No, I do not volunteer *Stop filling out this form.*

Skip to question 2.

Survey

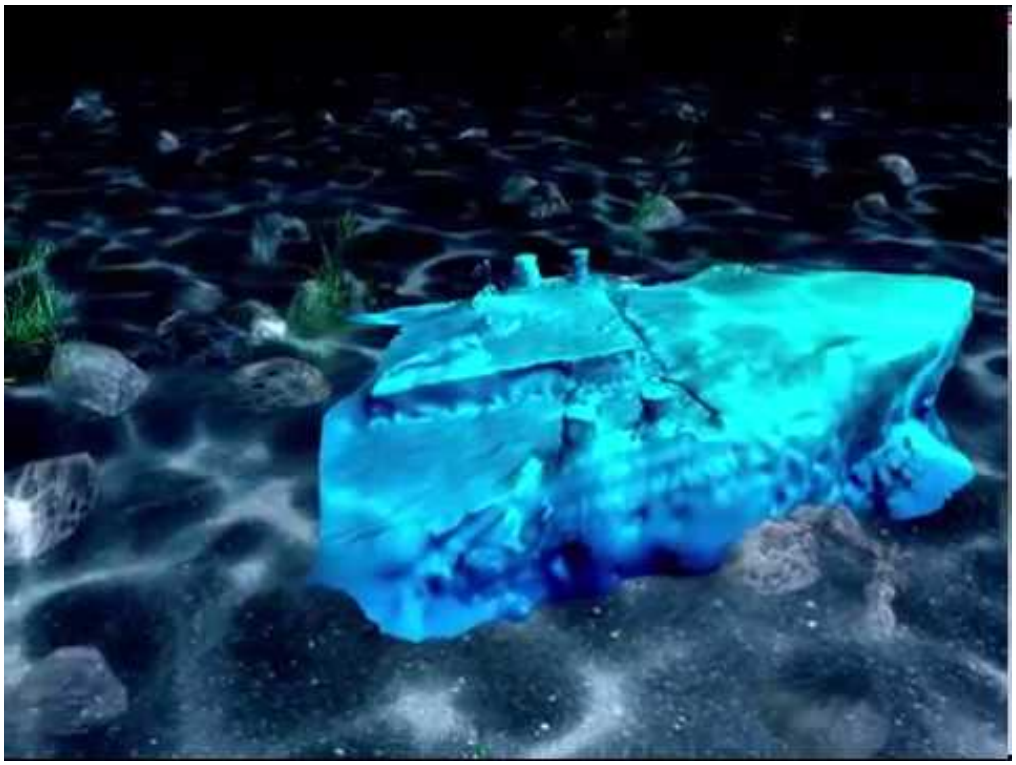
Evaluation of Cal Poly Masters Thesis about using probabilistic roadmap algorithms to generate virtual camera paths with cinematographic principles. This survey focuses on different methods of evaluating camera viewpoints. Videos have no sound. Replay button is in bottom left of every video if you need to watch them multiple times. If you wish to opt out at any point please close this page.

2. **Please rank the following videos by preference** *

   Mark only one oval per row.

	1 (most preferred)	2	3	4	5 (least preferred)
Video A	◯	◯	◯	◯	◯
Video B	◯	◯	◯	◯	◯
Video C	◯	◯	◯	◯	◯
Video D	◯	◯	◯	◯	◯
Video E	◯	◯	◯	◯	◯

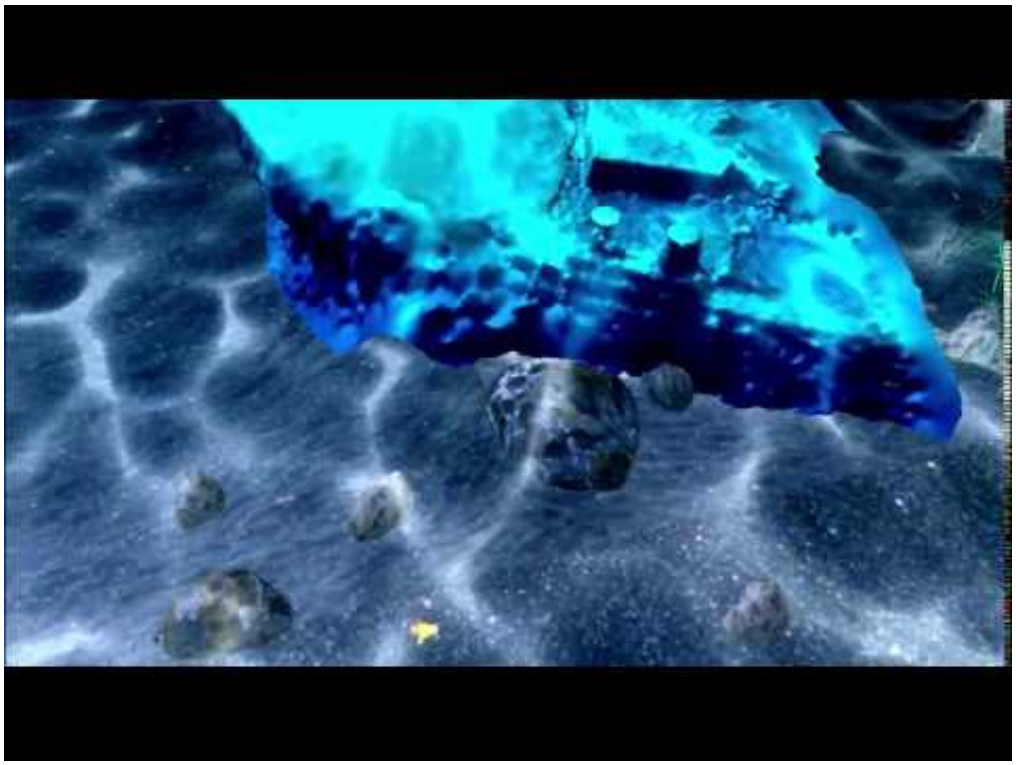
Video A



http://youtube.com/watch?v=_RenQhdZLkA

3. **Video A Comments**

Video B



http://youtube.com/watch?v=EjK1VGKeDG0

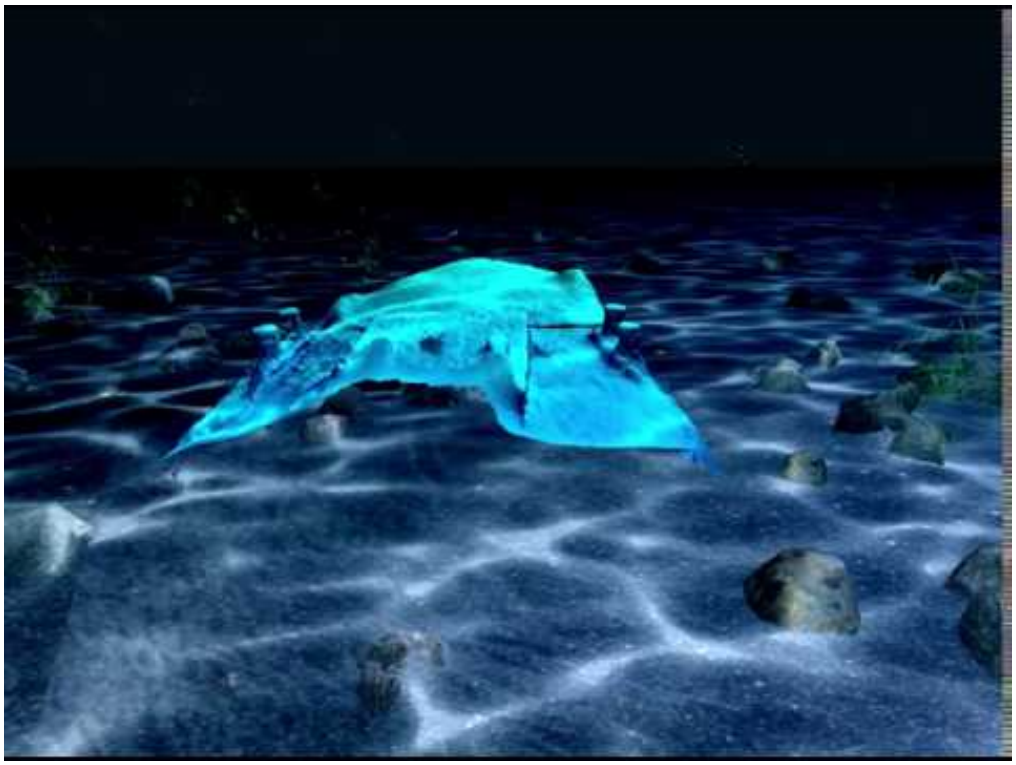
4. **Video B Comments**

Video C

http://youtube.com/watch?v=cezK0zCdiZ4

5. **Video C Comments**

Video D



http://youtube.com/watch?v=1S0SJQS7VR8

6. **Video D Comments**

Video E

http://youtube.com/watch?v=5aovCnCIWRU

7. **Video E Comments**

Last Thoughts

8. **Additional comments or feedback**

The survey of generated paths focusing on different path types is identical to the above survey with different videos. These videos can be found on this youtube channel `https://www.youtube.com/channel/UCUH0rkZ8iDLe9LjA2n8DPCA`. Links to specific videos are as follows.

Video A: `http://youtube.com/watch?v=fpLR5iENRBw`

Video B: `http://youtube.com/watch?v=sw8GIwJTm94`

Video C: `http://youtube.com/watch?v=Dwv5gBDGOqQ`

Video D: `http://youtube.com/watch?v=JW6LxsfddO4`

Video E: `http://youtube.com/watch?v=KWo2JnwJ5qg`