12-2014

Computing Approximate Solutions to the Art Gallery Problem and Watchman Route Problem by Means of Photon Mapping

Bruce Andrew Johnson
University of Tennessee - Knoxville, bjohnson3141@gmail.com

Recommended Citation

Johnson, Bruce Andrew, "Computing Approximate Solutions to the Art Gallery Problem and Watchman Route Problem by Means of Photon Mapping. " PhD diss., University of Tennessee, 2014.
https://trace.tennessee.edu/utk_graddiss/3141

To the Graduate Council:

I am submitting herewith a dissertation written by Bruce Andrew Johnson entitled "Computing Approximate Solutions to the Art Gallery Problem and Watchman Route Problem by Means of Photon Mapping." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Hairong Qi, Major Professor

We have read this dissertation and recommend its acceptance:

Lynne Parker, Lou Gross, Seddik Djouadi

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Computing Approximate Solutions to the Art Gallery Problem and Watchman Route Problem by Means of Photon Mapping

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Bruce Andrew Johnson

December 2014

DEDICATION

I dedicate this work to my wife Natthida Nonprasert Johnson. Love you, tee rak!

I dedicate this work to my good friends James Calloway, Bob Lowe, Rob Schwalb and Erik Sledd for being interesting, funny and inspirational.

I dedicate this work to my MTSU philosophy professor Dr. Ron Bombardi. Your Philosophy of Science course shaped my thinking in many profound ways.

I offer a special dedication to my dearly departed uncle Ben Moore. You were a great guy who stood up for me when others didn't.

ACKNOWLEDGEMENTS

I would like to thank Dr. Hairong Qi for serving as my advisor, and Dr. Lou Gross, Dr. Lynn Parker and Dr. Seddik Djouadi for serving on my committee.

I would like to thank Dr. Quyen Hyuhn of NSWC PCD for his generous support and encouragement.

I would like to thank Dr. Tory Cobb and Dr. Jason Isaacs of NSWC PCD for many helpful and productive discussions.

This work was funded by the Naval Surface Warfare Center Panama City Division's In-House Laboratory Independent Research (ILIR) program.

ABSTRACT

Wireless sensor networks (WSNs) can be partitioned into component sensor nodes (SNs) who are meant to sense information arriving from multiple spectra in their environment. Determining where to place SNs such that the amount of information gained is maximized while the number of SNs used to gain that information is minimized is an instance of solving the art gallery problem (AGP). In order to provide approximate solutions to the AGP, we present the Sensor Placement Optimization via Queries (SPOQ) algorithm that uses level sets populated by queries to a photon map in order to find observation points that sense as many photons as possible. Since we are using photon mapping as our means of modeling how information is conveyed, SPOQ can then take into account static or dynamic environmental conditions and can use exploratory or precomputed sensing.

Unmanned vehicles can be designated more generally as UxVs where “x” indicates the environment they are expected to operate – either in the air, on the ground, underwater or on the water’s surface. Determining how to plan an optimal route by a single UxV or multiple UxVs operating in their environment such that the amount of information gained is maximized while the cost of gaining that information is minimized is an instance of solving the watchman route problem (WRP). In order to provide approximate solutions to the WRP, we present the Photon-mapping-Informed active-Contour Route Designator (PICRD) algorithm. PICRD heuristically solves the WRP by utilizing SPOQ’s approximately optimal AGP-solving vertices and connecting them with the high visibility vertices provided by a photon-mapping-informed Chan-Vese or *k*-means segmentation mesh using a shortest-route path-finding algorithm. Since we are using photon-mapping as our foundation for determining sensor coverage by the PICRD algorithm, we can then take into account the behavior of photons as they propagate through the various environmental conditions that might be encountered by a single or multiple UxVs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

Quantifying the ability to see (or *visibility*) arises in many scientific fields such as rendering, visualization, surveillance, and navigation. One of the fundamental research problems confronting designers of wireless sensor networks (WSNs) and unmanned vehicles (UVs) concerns the attempt to assure that every portion of the WSN's or UV’s operational environment is sensed. Providing knowledge of an environment by a WSN or UV enhances the ability of those operating the WSN or UV to more fully understand what events are taking place in that environment and the subsequent actions that should be taken in response to those events. Lack of information (i.e. inadequate sensing) could conceal vital information that would compromise the ability of the WSN or UV operator to make an intelligent assessment of their environment. Therefore, the importance of quantifying and then optimizing visibility cannot be understated.

1.1 Motivation behind Solving the Art Gallery Problem

Wireless sensor networks applied to 3D spaces are gaining in importance for WSN researchers seeking to understand how to perform sensing in (typically) non-terrestrial environments. Applications are being created that assume that the WSN is deployed in the air or underwater where the constituent sensor nodes may occupy different heights or depths respectively. For example, small aerial wireless sensor nodes have been developed that are meant to perform fire monitoring [1]; underwater sensor nodes have been developed for the purpose of performing pollution monitoring [2]; another possible terrestrial 3D WSN application example would be the placement of sensor nodes at varying heights in a forest in order to monitor and prevent the spread of forest fires [3].

Quantifying the ability of a WSN to sense the environment it is deployed in is known as the *coverage problem* [4]. This problem seeks to answer the question: “How well do the sensors monitor the space they occupy?” This question can be expressed as *k-coverage* [4] which asks “Given an integer k and a sensed region R, can it be assured that each point in R is monitored by k sensors?”

The coverage problem has been considered by the mathematics community in the subject area known as *visibility optimization* [5]. One such visibility optimization problem is the art gallery problem (AGP) that derives its name from the hypothetical situation wherein an art gallery owner seeks to have every painting covered with as few sensors as possible. The AGP is similar to k-coverage except that it seeks to use the smallest k sensors necessary to assure that every point in R is covered by at least one sensor.

Let D be a set of points which comprise a surface and let it be a member of the compact subset Ω of $\Re^d$ where $\Re^d$ is a d-dimensional subset of the real numbers. The AGP is posited as the following question:

- What is the minimum number of sensors necessary to assure that every point in Ω is sensed [5]?

The hidden assumption behind the formulation of the AGP is that all points in R that fall within the sensing range of the sensor are equally capable of being monitored from a given observation point. While this level of mathematical abstraction of an ideal environment and its sensor is useful, it does not take into account mitigating factors that can disrupt the ability of a sensor to obtain optimal coverage. Furthermore, works concerned with the AGP [5], [6] and [7] presume that the monitored space is *known* prior to the placement of the sensors – an assumption that cannot hold valid in changing environments. This leads us to pose the following questions:

1. How does the WSN researcher determine how to solve the AGP in an online manner in 3D environments that have not been explored yet or have been altered after the passage of a unit of time?
2. How does a WSN researcher determine how to solve the AGP in 3D environments that are less than ideal (such as those prone to fog, dust, rain and so forth)?
3. How does a WSN researcher determine how to solve the AGP in 3D environments wherein the ability of a given sensor to sense its surroundings changes with time (such as a monitored space transitioning from day to nighttime)?

Consideration of these questions is becoming more important as higher-quality simulations are demanded by WSN researchers. Enhancing the realism of the expected environmental circumstances in which the WSN will operate can save deployment and infrastructure costs when attempting to understand the WSN's operational outcome. Furthermore, the ability of the researcher to optimize the placement of a WSN under realistic scenarios leads to greater quality of service and efficiency. For example, if a certain sensor node is useful only during the day due to its location but becomes useless at night, it can conserve its battery life by waiting to activate during the day in order to best cover its space to be monitored.

In order to be able to address these questions, we examine the nature of the photon. The photon is the fundamental quantum constituent of electromagnetic (EM) radiation. Since sensors are meant to detect some member of the EM spectrum, the photon may also be regarded as being responsible for transferring information from a point in the monitored space to the sensor that detects them. Such a notion of photons being the enablers of sensing is intuitively and scientifically confirmed by the fact that our eyes' ability to see is a consequence of visible-light-photons reflecting off of surfaces and triggering the biochemical reactions necessary to allow sight [8].

When a sensor node is regarded as being responsible for detecting the energy transmitted by the photons that reflect off of a surface in an environment, the AGP can be recast by asking the following question:

- What is the minimum number of sensors necessary to assure that every photon available to convey information about Ω is sensed?

Computing approximate solutions to this formulation of the AGP relies upon the creation of a model describing how a set of photons will be propagated through a 3D virtual environment. Having this photon propagation model allows us to address the three questions posed above in the following manner:

- The first question can be addressed by containing the result of applying a 3D photon propagation model in a low-memory data structure that returns the number of photons sensed in a particular area quickly. Such a data structure enables fast exploration of an unknown or changing environment in order to determine the best location to place a sensor so that it may sense as many photons as possible.
- The second question can be addressed by an understanding of how the 3D photon propagation model gets altered by the less-than-ideal environmental circumstances the photons encounter. The best location to place a sensor may then take into consideration the photon's behavior after it has been affected by these less-than-ideal circumstances.

- The third question can be addressed by altering the number of photons used in the 3D photon propagation model. A photon-rich environment simulates a daytime circumstance whereas a photon-poor environment simulates a nighttime circumstance.

Modeling the behavior of photons is a well-established area of study in the graphics community and one such algorithm that produces a photon model and data structure that meets our requirements is *photon mapping* [9]. The photon mapping algorithm provides us with the following:

- The presumption that virtual photons are propagated in a 3D virtual environment,
- The use of a balanced *k*-d tree data structure [10] (known as a *photon map*) requiring O(log *p*) query time and O(*p*) memory to store the results propagating *p* photons,
- The ability to take into account how participating media affects the behavior of photons that propagate within the 3D virtual environment, and
- The ability to alter the number of photons that will propagate through the virtual environment.

1.2 Motivation behind Solving the Watchman Route Problem

Unmanned vehicles are gaining wider use and applicability as their operating environments continue to expand. These unmanned vehicles can be designated more generally as UxVs where "*x*" indicates the environment they are expected to operate – either in the <u>A</u>ir, on the <u>G</u>round, <u>U</u>nderwater or on the water's <u>S</u>urface. In order to maximize the UxV's utility within these operating environments, it is necessary to understand how to plan an optimal route within the context of the UxV's operating environment such that the amount of information gained by the

UxV is maximized and the cost of gaining that information is minimized. Furthermore, when simulating the operation of the UxV in a 3D virtual environment, it is necessary to have a set of possible waypoints that may be selected or discarded when portions of the 3D virtual environment become forbidden or impassable.

Determining an optimal route which maximizes the amount of information obtained while minimizing the consumption of resources is a matter of solving a visibility optimization problem known as the *watchman route problem* (WRP). The algorithm we will present in this dissertation is meant to provide a set of candidate waypoints in a 3D simulated environment to allow for greater adaptation in choosing a route to solve the WRP while maintaining cognizance of environmental variability.

The WRP derives its name from a hypothetical situation wherein an art gallery owner uses a watchman to monitor every painting in the gallery by using the minimum distance route through the gallery for the watchman to walk. The WRP is an optimization problem that seeks to find the minimum distance for an observer to trace through some environment such that every point in that environment is monitored. The WRP is considered to be an NP-hard problem.

Let D be a set of points which comprise a surface and let it be a member of the compact subset Ω of $\Re^d$ where $\Re^d$ is a d-dimensional subset of the real numbers. The WRP is posited as the following question:

- What is the shortest UxV route available such that the maximum number of points found in Ω are sensed [5]?

The typical assumption made about the environment the UxV is supposed to operate within while solving the WRP is that it is both unchanging and a vacuum. These assumptions do not hold up in the real world where various events which may prevent a UxV from obtaining the information required to solve the WRP. The questions that were asked and the answers provided by the photon mapping algorithm in Section 1.1 with regard to the AGP may be applied with equal relevance to the WRP. Thus, in our UxV path-planning context, the WRP is posed as the following question:

- What is the shortest UxV route available such that the maximum number of points in the environment are sensed?

A photon map, being ultimately a 3D query-able *k*-d tree, may be queried at some point in space in the virtual environment to yield a subset of the photons contained within it. The number of photons returned by the photon map at a point in space represents the amount of illumination-energy present at that point and may be regarded as its *photon volume*. (Conversely, if a point is not illuminated, it may be regarded as being *shadowed*.)

In order to incorporate photon mapping into our watchman's route designator, we use the Chan-Vese segmentation algorithm [11] on a set of candidate waypoints that have associated with them a photon volume. Having a measurement of a point's photon volume provides us with the ability to create a gradient distinguishing between those points in a 3D virtual environment which are shadowed from those that are illuminated. Establishing this sort of distinction is a prerequisite to being able to perform our photon-mapping-informed Chan-Vese segmentation method. When we say that we are using the Chan-Vese segmentation method, we are producing a 3D mesh within a 3D virtual environment, minimized with respect to the Mumford-Shah [12] energy functional, such that those vertices comprising the mesh may serve as UxV-path waypoints providing a greater photon volume than those waypoints which are not elements of the mesh. Furthermore, using photon-mapping as a means of informing the evolution of the Chan-Vese segmentation algorithm's segmentation boundary allows for the accommodation of dynamic scenes which may also be affected by information-attenuating events per the four benefits provided by the photon mapping algorithm explained in the previous section.

In order to assure that we will select the vertices with the highest photon volume, we choose initializing waypoints for our Chan-Vese segmentation algorithm that solve the art gallery problem (AGP). The AGP is closely related to the WRP in the sense that it is an NP-hard visibility optimization problem concerned with selecting the minimum number of points in space that offer complete visibility. The difference between the WRP and AGP is that the AGP is concerned only with selecting fixed points in space but it is not concerned with finding the shortest path connecting them. As we will show, making the effort to solve the AGP in order to generate the initial vertices for the Chan-Vese segmentation algorithm provides us with the maximum probability of segmentation occurring and also assures us that the vertices comprising the segmentation mesh have greater photon volume than those vertices not within the mesh.

Once this segmentation mesh is created, a conventional shortest-path-finding algorithm can then be applied to connect the vertices contained within the segmentation mesh to solve the AGP. These connected vertices lead to the production of a shortest path which senses more photons than a path that does not utilize the vertices provided by the mesh. The WRP is then solved immediately in a heuristic manner since the WRP route will be designated by choosing the shortest route from elements comprising a minimum-energy mesh encompassing the (approximately optimal) AGP-solving vertices. Furthermore, since our approximately optimal AGP-solver (described in Chapter 4) does not require foreknowledge of the 3D environment it is placed within, we can then (randomly) discard or add vertices prior to generating approximate solutions to the AGP (and subsequently the WRP) in order to simulate forbidden zones or moving obstacles within our virtual environment.

1.3 Dissertation Outline

This dissertation is organized thusly: Chapter 2 provides a literature review of works concerning the AGP, Chan-Vese segmentation algorithm and the WRP. Chapter 3 provides an in-depth discussion regarding the photon mapping algorithm and how it may be used to measure coverage. Chapter 4 provides a description of our approximately optimal AGP-solver. Chapter 5 provides a discussion of our version of the Chan-Vese segmentation algorithm. Chapter 6 shows how the efforts described in Chapters 5 and 6 produce our WPR-solver. Chapter 7 discusses what future work we expect to emanate from this research.

1.4 Contributions from This Work

The contributions arriving from this work are three-fold:

1. We show how the use of the photon mapping algorithm, in combination with a modified prevailing approximately optimal AGP-solving algorithm, allows us to create an online approximate solution to the AGP that answers the three AGP-related questions given in Section 1.1.
2. We have produced a 3D Chan-Vese segmentation method informed by the (multispectral) photon mapping algorithm and our approximately optimal AGP-solver that produces a mesh leading us to answer the questions given in Section 1.2.
3. We have established an approximately optimal WRP-solver utilizing the Chan-Vese algorithm mentioned above to find the shortest path through a virtual environment while maximizing the number of photons sensed and taking into account dynamic, less-than-perfect environments.

Figure 1.1 below summarizes how these contributions link together.

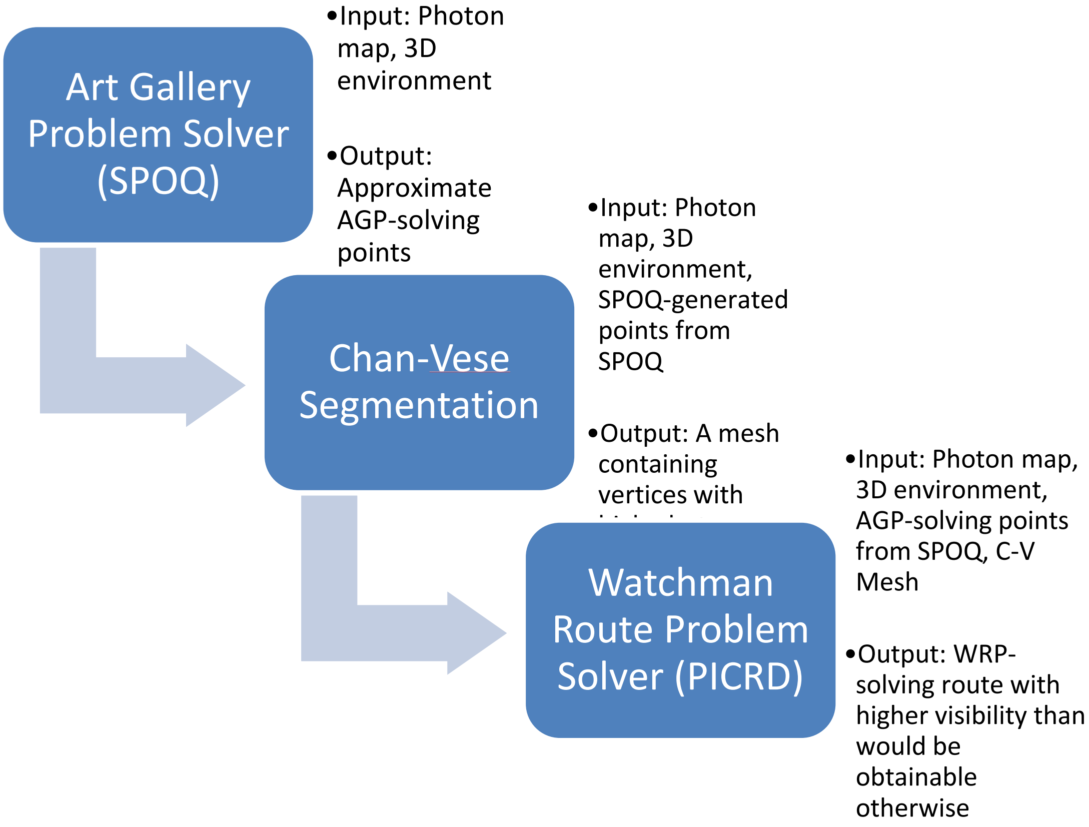

Art Gallery Problem Solver (SPOQ)
•Input: Photon map, 3D environment
•Output: Approximate AGP-solving points
Chan-Vese Segmentation
•Input: Photon map, 3D environment, SPOQ-generated points from SPOQ
•Output: A mesh containing vertices with
Watchman Route Problem Solver (PICRD)
•Input: Photon map, 3D environment, AGP-solving points from SPOQ, C-V Mesh
•Output: WRP-solving route with higher visibility than would be obtainable otherwise

Figure 1.1. Depiction of how the contributions described in this dissertation link together.

CHAPTER 2

LITERATURE REVIEW

2.1 Work Related to the Art Gallery Problem

Consideration of how to solve the coverage problem has been the subject of much research. These efforts can be generally divided into two categories. The first category is dedicated to producing an algorithm meant to establish and maintaining coverage and the second category is dedicated to a theoretical analysis of coverage performance.

Algorithms have been developed to maintain sensor coverage while using the minimum number of sensors [13] and [14]. These efforts presume that the sensing range is a circle and can sense something within their sensing range with certainty. Other efforts to maintain sensing coverage make the same circular sensing range assumption but have a probabilistic sensing ability [15] and [16]. Theoretical studies regarding how to solve the *k*-coverage problem have been formulated for different sensor deployment strategies [17]. These studies presume that the sensing range is circular and can sense whatever is within their sensing range with certainty. Another study [18] continues to presume that sensor has a disc-shaped sensing range, but inject some probability-based uncertainty in the sensor's ability to provide coverage.

Most articles published about obtaining sensor coverage presume that the virtual environment to be sensed is 2D. There have been a few works [19] and [20] published which presume that the environment to be sensed is 3D. These works presume that the sensor is surrounded by some space-filling volume which represents the limits of their omnidirectional sensor's range. Their aim is to place the sensors in the 3D virtual environment in such a way that fills up as much space as possible while using as few sensors as possible.

A key feature of the previous efforts is the construction of a *sensor grid* where the virtual sensors will be placed. In the 2D and 3D cases, the sensor grid is created by placing a tessellated set of points or a space filling volume filled with points, respectively, on the environment to be sensed such that a sensor positioned at point x is displaced from its neighboring by some distance. In both the 2D and 3D cases, there exists the possibility of having an object (whose dimensions are defined by *surfaces*) occupying the environment to be sensed such that no sensor grid point may be placed inside the boundaries of these objects.

In the mathematics community, sensor coverage is given the name *visibility coverage* [5]. Whatever is covered within the sensing range of a sensor is regarded as being *visible*. The sensor is regarded as an *observer* and the position the observer resides is an *observer station*. The sensor grid mentioned above is thus called an *observer grid*. Determining what is visible by an observer is accomplished by tracing a ray from the observer station until it intersects with an object. The point of intersection on the object is thus regarded as being visible and is thus *observed*.

Let O be a set of points that comprise a surface and let it be a member of the compact subset Ω of $\Re^d$ where $\Re^d$ is a d-dimensional subset of the real numbers. The set O is an *occluder* residing in *environment* Ω. The set $X = \Omega \setminus O$ is the set of points in which an omnidirectional observer may be stationed and $x_0 \in X$ is some observer station. An observer grid $G \subseteq X$ comprises of a sequence of points $g_0, g_1, \ldots g_n$ wherein element g_i is separated by its neighboring elements by some distance such that the distance does not exceed the boundaries of G. This distance assumption ensures that at least one observer station is found within the confines of G.

The AGP is posited by the mathematics community as the following question:

- What is the minimal number of observers necessary to ensure that the maximum number of points in Ω are observed?

In [5], Tsai et al formulated a method for achieving visibility in a 3D virtual environment without considering optimal observer placement. Building upon this work, Cheng and Tsai [7] solved the AGP by dismissing global optimality and considered only local maxima as a suitable solution. This is due to the fact that the AGP is known to be non-convex and thus gradient ascent methods are incapable of determining globally optimized visibility coverage [6]. Goroshin, et al [6] expanded upon the work of Cheng and Tsai by providing a greedy iterative algorithm within the level set framework in order to solve the AGP.

The efforts made by Cheng, Tsai, et al and Goroshin et al have centered on using the so-called "fast visibility sweeping" algorithm as a means of performing "implicit ray tracing" [6] in order to provide the information necessary for providing approximate solutions to the AGP in both 2D and 3D environments.

Given environment Ω, a *known* environment allows the observers to have already sensed Ω prior to computing approximate solutions to the AGP. An *unknown* environment means that observers cannot have sensed Ω prior to computing approximate solutions to the AGP and further implies that they will have to explore *X* or *G* in order to solve the AGP.

The level set methods advanced in [5, 6, 7] are useful in a known environment. The computational cost of the AGP-related algorithms presented in [5, 6, 7] is $O(l{\cdot}m)$ steps where l is the number of sensors and m is the number of observation points in the virtual environment. Furthermore, in order to perform the visibility calculation in [6], the visibility of every single observer point must be precomputed and stored in memory at a cost of $O(m^{2d})$ memory where m

is the number of elements contained in the environment in consideration and *d* is the dimension of the environment.

2.2 Work Related to the Chan-Vese Segmentation Algorithm

The level set method of segmentation is a means of modeling how a segmenting curve's boundary will propagate with time without the need to worry about the movement of an individual point on that segmenting curve's boundary. The initial condition for the level set method of image segmentation is a closed curve introduced at some region of interest to be segmented within the image. The basic idea is to advance the segmenting curve's boundary as a so-called *active contour* or *snake* [21] where the boundary is expected to eventually wrap around the perimeter of an area of interest in the image to be segmented. The level set method of image segmentation is particularly useful when one desires to segment an image with complex topology or faces the possibility that the evolving curve's boundary will split during the segmentation [22].

Chan and Vese [11] introduced a method for advancing the segmenting curve's boundary that utilized the well-posed case of the Mumford-Shah functional [12] that is minimized when the segmenting curve's boundary reaches the perimeter of the area of interest to be segmented within the image. Hence, their energy function, or "fitting term", established a search criterion for the segmenting boundary's advancement and halting.

The topic of segmentation of 3D volumes without using the Chan-Vese method has been considered in [23], [24] and [25]. In [23] they perform segmentation using multiple discretization equations for solving the curvature term. In [24] they perform segmentation without the need to solve a PDE. In [25] they use multiple "smoothing terms" in order to control the evolution of the segmenting boundary.

2.3 Work Related to Path Planning and the Watchman Route Problem

UxV path planning is a broad field of study. It may be roughly divided into two categories: quantitative navigation and topological navigation [26]. Topological navigation is concerned strictly with finding a particular path from a starting point to an end goal. Topological navigation is different from quantitative navigation in the sense that it is not concerned with a quantitative measure of optimality in order to achieve a goal [27]. Quantitative and topological path planning depend on creating a representation of the space the UxV is expected to navigate within, accommodating dynamic environments and computational tractability.

The representation of the environment in which the UxV is expected to navigate is its *configuration space* [26]. A configuration space may be three dimensional on the *x*-, *y*- and *z*-axes. Three further dimensions may be added if the UxV can exhibit yaw, pitch or roll in the ϕ, θ, γ angles, respectively. A graph data structure can be used to represent the configuration space. The most popular configuration space graph representation is a two-dimensional set of tessellated grid points [26]. However, the representation could be that of a Voronoi graph or octree [28]. Quantitative path planning algorithms usually divide the path into a set of sub-goals - known as *waypoints* - which are fixed points found in the space the UxV is expected to navigate within. These waypoints are then sequentially navigated to until all waypoint sub-goals have been accomplished in a manner that is optimal.

The optimality of the path obtained is a matter of measurement by some metric. This metric may be the shortest amount of time taken, smallest energy consumption, the shortest distance traveled or safety [27]. One possible means of obtaining a shortest-path-length metric may be assigning a numerical weight to an edge connecting two vertices in the configuration space's representative graph. The path-planner would then find the optimal path when it designates a route with the smallest total edge weight [29]. A possible WRP solver would obtain optimality if it found the smallest total edge weight that allows it to traverse between its path waypoints while maximizing how much of the environment is sensed while taking that path.

The use of PDE-based level set and fast marching techniques for shortest path planning have been discussed in [30] and [31]. In [30] and [31] Kimmel and Sethian developed a means of creating a set of evolving curves for the purpose of determining the shortest path length necessary to reach a goal. Given a starting point x_o and destination point x_d, fast marching methods were used to create a level set curve originating at x_o that rippled iteratively outward until a level set curve intersected point x_d. A path was then traced from x_d back to x_o such that each path waypoint intersected orthogonally with a previous smaller-perimeter iteration of the evolving curve. Each waypoint is separated by a minimum arc length. The advantage of their fast marching method was the ability of the evolving front to "flow around" obstacles and thus provide an easy rejection of paths that might be blocked.

Consideration of how to solve the WRP using PDE-based variational techniques has been considered previously in [6], [5] and [7]. In [5], [6] and [7] the authors approached the WRP by representing the WRP objectives as energy functionals and the watchman's route was represented as an implicit curve based upon the use of level sets. The authors in [5] evolved curves representing the watchman's route according to gradient descent on the Euler-Lagrange equation of a visibility functional. In [6] Goroshin et al. devised a visibility functional that was locally maximized by using gradient ascent on the level set curve. In [7] the authors presented a 3D variational path-planner that had applicability to visibility calculation (although this was not explicitly calculated). This involved adapting a preexisting level set framework for accommodating movement of curves in 3D space. Note that in a manner similar to the AGP, the solution space for the WRP is non-convex and is only guaranteed to attain maximum visibility coverage within the configuration space while attaining a local minimum path length.

Consideration of how to provide a solution to the WRP in a 3D virtual environment *without* using PDE to inform path- planning has been the subject of research previously by [32], [33], and [34]. The methods considered previously relied upon computing approximate solutions to the AGP and then using a traveling salesman problem (TSP) solver to connect the subsequent approximately optimal AGP-solving waypoints within the environment. The work of [32] applied their WRP-solution to the inspection of a ship's hull. The work of [33] further refined the efforts made in [32] by offering a shorter, smoother WRP solution which took into account differential environmental changes in the TSP route edges. The efforts made by [34] built upon the work of [35] but with the application of a *k*-d tree to make fast approximately optimal AGP-solving sensor coverage queries on the way towards obtaining an approximate solution to the WRP using a TSP solver.

Dynamic environments can experience effects which may alter the ability of the UxV to maneuver in their environment. In [27] Lolla et al. constructed a fast marching method similar in concept to that found in [31] which provides a shortest path length solution to Unmanned Underwater Vehicle (UUV) path planning. Their path-planner is distinguished by the fact that it takes into account environmental effects arriving from simulated variable ocean current flow in a 2D configuration space.

It is important to note that the previously-considered 3D WRP-related algorithms assume a static configuration space that is known prior to their algorithm's execution and are thus not meant to accommodate dynamic environments. We will show that since our approximately optimal AGP-solving algorithm does not assume a known environment prior to deployment, we may then take into account dynamic environments. Furthermore, we will show that a naïvely-generated shortest route is *not* unique. It is possible to create a short route that provides better visibility coverage by first utilizing a segmentation algorithm to generate shortest-path waypoints with greater visibility. Having this ability to alter a watchman's route using an online approximately optimal AGP-solving method enhances UxV autonomy by promoting adaptability. We will provide scenarios in our Results section that demonstrate how to alter the watchman's route (which are ultimately applicable to UxV routes) under changing environmental circumstances.

It is also important to note that each AGP and WRP solution assumes that the configuration space in which the solution is determined remains static and is not subject to varying environmental conditions. Our goal is to combine the ability to represent environmental attenuating effects on visibility by means of photon mapping with AGP and WRP solutions in order to provide more physically-accurate results.

2.4 Chapter Summary

The AGP and WRP are important problems in the fields of surveillance and UxV path-planning. Previous AGP solutions did not use a structured photon transport model to measure how the sensors would behave in different environmental conditions. Previous WRP solutions did not treat path development as a segmentation problem. We will show how the inclusion of photon-mapping and segmentation algorithms lead to more robust solutions to the AGP and WRP.

CHAPTER 3

PHOTON MAPPING

This work proposes a new coverage formulation that utilizes the modeling of photons through the photon mapping algorithm. We show how this algorithm provides us with the means to answer the three questions posed in Section 1.1. Photon mapping is a global illumination algorithm that is typically associated with graphics rendering software and is known for its ability to provide high-quality physics-accurate representations of a given 3D virtual environment [9]. Photon mapping is a two-pass algorithm comprised of constructing a photon map in the first pass and performing rendering in the second.

3.1 Constructing the Photon Map

The first pass is composed of three actions: emitting photons, propagating photons and then storing photons [9]. The first action entails allocating *p* virtual photons and then launching them into the 3D virtual environment from an arbitrarily-located photon source. The second action entails tracing the photons' movements - or *bounces* - through the 3D virtual environment until each photon has either been absorbed, has left the confines of the 3D virtual environment or has made a predefined number of bounces. The third action entails storing the photon's power, final point of intersection, and direction of entry with a surface in the 3D environment in a photon map (whose data structure is that of a balanced *k*-d tree) [10].

A virtual photon's initial direction is randomly selected based upon whether the photon source is spherical, rectangular or directional. Each photon is assumed to be of equal intensity and has a fragment of the total energy contained in the emitter. This energy for an individual photon is given by the equation $e_{\lambda} = (h \cdot c)/\lambda$ where h is Planck’s constant, c is the speed of light in a vacuum and λ is the photon’s wavelength. Note that a photon to be emitted by a photon source need not be confined to the visible light spectrum; it may in fact represent any member of the EM spectrum since a given photon's energy is a function of its wavelength. Usually, the power variable in a virtual photon's data structure is a triple representing the power scale in the red, green and blue wavelengths [9]. However, multiple wavelengths could be used [36] and [37].

After a photon propagates through the 3D environment and then impinges upon a surface, the photon may be absorbed, reflected or transmitted depending upon the material properties of the surface it strikes. The determination as to whether a photon is absorbed, transmitted or reflected is accomplished by means of the *Russian roulette* Monte Carlo technique [9] and [38]. If the photon is absorbed by a surface, no further action will be taken. If the photon is reflected, it will either be reflected by a specular-reflective surface (such as polished metal) or a diffusely-reflective surface (such as a wooden desk). When a photon strikes a specular-reflective surface, the reflected direction is determined by the mirror's reflection angle [39]. Should the photon strike a diffusely-reflective surface, the photon is stored in the photon map and then reflected but in a direction determined by the *bidirectional reflectance distribution function* (BRDF) [40]. If the photon is transmitted, the photon is not stored in the photon map and Snell's law is used to determine its outgoing direction. These concepts are illustrated in Figure 3.1.

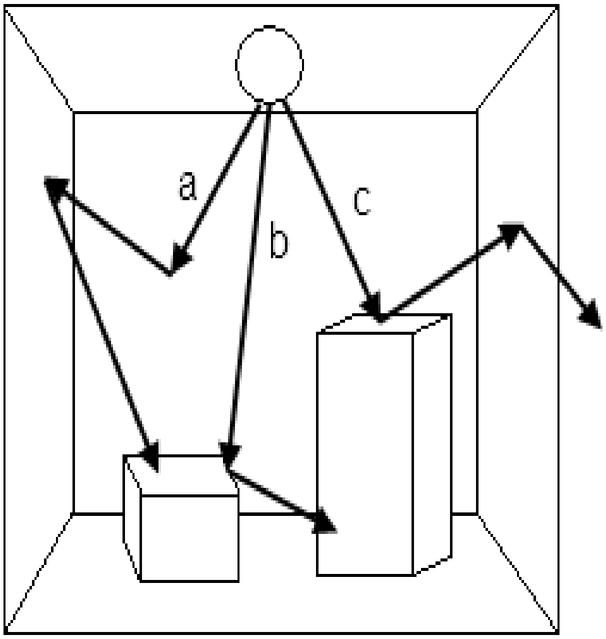


Figure 3.1: Here is a stylized version of a photon map in a simple box scene. Photon **a** is diffusely reflected until it is absorbed, photon **b** is specularly reflected and photon **c** is diffusely reflected until it leaves the environment. Note the presence of the photon source in the upper center of the environment.

The Russian Roulette Monte Carlo technique was first introduced in the context of particle physics and is used to eliminate unimportant photons. If a photon is absorbed, for example, then no further action need be taken and thus computational resources can be expended only on those photons that are either transmitted or reflected and thus contribute to the observer's understanding of the scene monitored. Let $\zeta \in [0, 1]$ be a uniformly distributed random variable. The probabilities for a photon to be absorbed, reflected or transmitted is decided thusly:

$$\begin{array}{rcc} \zeta \epsilon \, [0, d] & \rightarrow & \text{diffuse reflection} \\ \zeta \epsilon \, (d, s + d] & \rightarrow & \text{specular reflection} \\ \zeta \epsilon \, (s + d, s + d + t] & \rightarrow & \text{transmission} \\ \zeta \epsilon \, (s + d + t, 1] & \rightarrow & \text{absorption} \end{array}$$

where *d*, *s* are the material’s diffuse and specular reflection coeffients, respectively and *t* is its transmission coeffient. The accommodation of photons that lie outside the bandwidth of visible light may be accomplished by adjusting the probabilities of absorption, reflection and transmission.

Multiple photon maps may be used to store photons that manifest different types of behavior [9]. One such photon map is the *caustic photon map* which stores photons that are purely transmissive before hitting a diffuse surface. For the purposes of this work, we will use only one photon map known as the *global photon map*.

3.1.1 Participating Media

When we want to take into account the effects of participating media such as fog, rain, smoke and so forth, Jensen advocates the use of a *volume photon map* which stores the photons' interaction with the media [9]. Adjustments to the propagation of the photons through the media are made based upon the media's density, whether it is hetero- or homogeneous and whether or not the photon's scattering by the media is anisotropic. The photons which escape interaction with the participating media are stored in a global photon map per the rules explained above. The concept of participating media as it applies to photon mapping is illustrated in Figure 3.2.

3.2 Using the Photon Map

The photons contained in the photon map have conventionally been used in the second pass of the photon mapping algorithm by a ray-tracer in order to provide a rendering of a 3D virtual environment. Rendering is achieved by determining the total amount of energy radiated from every surface in the virtual environment which has been intersected by a ray cast by a ray-tracer at some observation point. A query to the photon map yields those photons that contribute to the radiant energy - expressed as a color - found at that point of intersection. A query takes the shape of a sphere (or *query-sphere*) of radius *r* centered at point x_o (or *query-point*). For our purposes, the query-sphere and the query-point are analogous to a sensor node's (or UxV's) sensing range and location, respectively. Since the photon map utilized by the photon mapping algorithm is guaranteed to be balanced, the computational cost of getting the photons that populate the query sphere is O(log *p*) where *p* is the total number of photons launched in the virtual environment. The memory requirement is O(*p*).The query-point and point of intersection on the radiating surface are traditionally regarded as equivalent. But we claim that we are not limited by this presupposition.

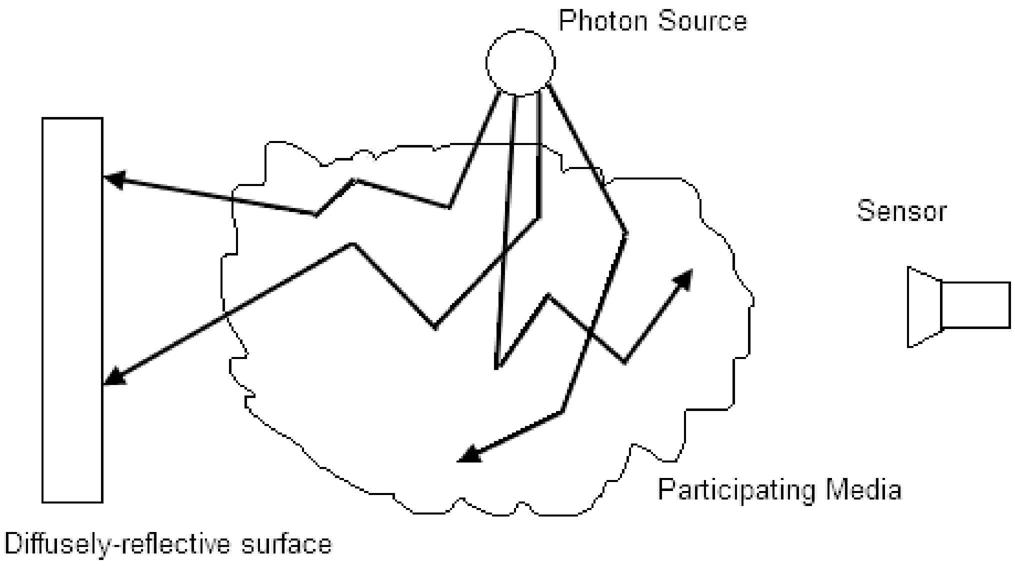


Figure 3.2: Photon scattering in the participating media in the center of the figure is stored in a volume photon map whereas those photons which strike the diffusely-reflective surface on the left are stored in the global photon map. Photons in both maps can be perceived by the sensor on the right.

The query-sphere can also be regarded as a means of gathering the ambient energy and therefore information emanating from surfaces in the environment to be sensed. If a photon stored within the photon map is located a distance which is less than or equal to *r*, that virtual photon is returned by the photon map. In Section 3.2 we will discuss how we decide if the returned photon is indeed visible to the sensor. By making a sensor’s sensing range to be the equivalent in dimension of a query-sphere, we presume that sensors are omnidirectional. This presumption limits the types of sensors that we can model. We will discuss how to overcome this limitation in Chapter 5, Section 5.1

3.3 Using the Photon Map to Measure Coverage

The photon map stores the photon's power, final point of intersection on a surface and direction of entry. In our application, the stored photons are modified in their data structure in that they store the surface's normal vector translated to point of the photon's intersection. We will show how this variable aids in the determination of the coverage of a monitored space.

A query-point may be placed near a thin wall and the volume of query-sphere may contain visible-light-photons from both sides of the thin wall which are invisible to the query point but may still falsely contribute to the visible surface’s covered photons. This concept of having both visible and invisible photons in the query sphere is illustrated in Figure 3.3.

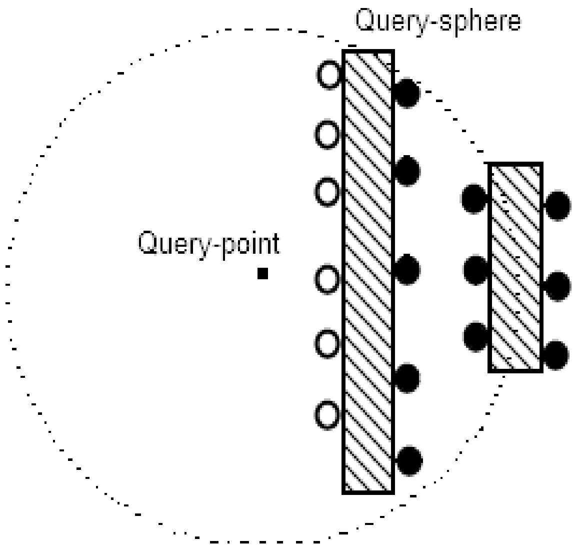


Figure 3.3: The query sphere can encompass photons on surfaces that are visible (white) and shadowed (black).

Distinguishing between those photons that are responsible for contributing to the sensor’s ability to sense and those that don't is a well-established problem in photon mapping and has been subject to consideration previously [9, 41]. We use the name *photon culling* to describe the act of testing and removing those photons found within the query-sphere which do not contribute to sensor coverage. Those photons which are occluded by an intervening obstacle are said to be *shadowed*. The idea of distinguishing between photons which contribute to the radiant energy of a surface and those that don't was first mentioned in [42] with the introduction of so-called *shadow photons* who reside on surfaces that are oriented away from the direct illumination of the photon source. This was done as a means of establishing where shadows will lie when the scene is rendered from a prescribed observation point. Another attempt at establishing whether a photon is relevant was done in [43] with the introduction of importance maps containing important photons or “importons” which concentrate photons in those areas which are most likely to be visible to an observer at a prescribed observation point. Since it cannot be known *a priori* where our sensors will be placed in order to optimally obtain the most information, we cannot know in advance what photons will be shadowed and what photons will be important. We must focus our attention strictly to culling those shadowed photons which are within the volume of the query-sphere. That is to say, we must focus on keeping those photons which contribute to the ability of the sensor to cover the area monitored at that sensors observation point.

A means of culling non-contributing photons is to perform a dot product test on the normal vector emanating from point of intersection of the photon and the query-point. If the dot product performed on the query-point and the normal emanating from the point upon which the photon resides is greater than 0, then the query-point and the point of intersection are connected to one another and thus visible to one another. This technique for photon culling is similar to what was proposed by Jensen in [9].

The dot product test has a computation time of O(q) and is thus fast but it cannot decide if there is an intervening polygon between the query-point and tested photon. The dot product test can yield photons that have a positive visibility dot product but are in fact occluded. This concept is illustrated in Figure 3.4.

The photons for which the dot product test has been applied are thus called invisible (have a negative dot product test), falsely-visible (have a positive dot product test but are occluded by a thin

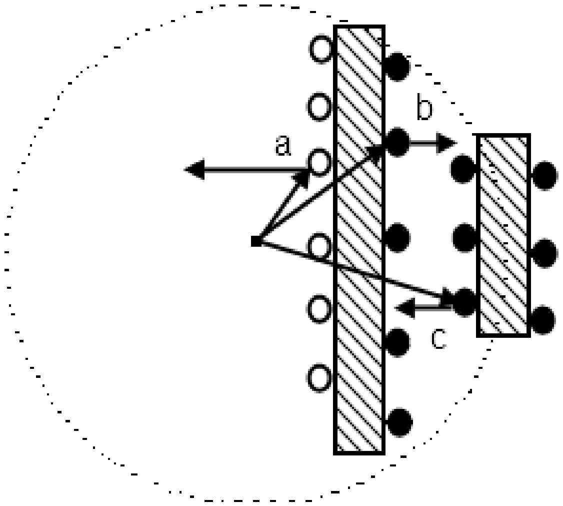


Figure 3.4: The dot product test can determine if a photon is visible (a) or invisible (b) but allows for false positives (c).

wall) or truly-visible (have a positive dot product test and have no intervening obstacle between them and the query-point). Those photons which may be either invisible or visible are regarded as potentially-visible. The dot product test can be avoided if the surfaces in the 3D environment are known to be convex and of sufficient thickness such that no query sphere's volume will also accommodate potentially non-contributing photons.

When no such assumption can be made, a means of distinguishing between falsely- and truly-visible photons from the set of potentially-visible photons is the hidden point removal (HPR) algorithm [44]. This algorithm has been applied to point clouds on object surfaces as a means of distinguishing between those surface points which are visible. In our case, this algorithm can be applied to the “cloud” of photons surrounding the query-point to cull useless photons.

Given a query-sphere of radius r centered at the query-point x_o and containing the photons $Q = \{p_i \,|1 \leq i \leq q\}$ sampling a surface S, the goal of HPR is to determine whether photon p_i is visible from point x_o. The algorithm consists of first performing spherical inversion on the points in Q and then performing convex hull construction on the inverted points. Those points which reside on the convex hull are considered visible. The algorithm does not assume that the topology of the surface upon which the photons reside is known nor does it assume the availability of the normal vectors emanating from the photons. The HPR algorithm has a computational complexity of $\mathrm{O}(q \cdot \log q)$.

Since the point cloud necessary for HPR's operation is the result of performing a query on a balanced k-d tree, the computational cost of getting the photons that populate Q is $\mathrm{O}(\log p)$ where p is the total number of photons launched in the virtual environment. Hence, a photon map query augmented by the HPR photon culling algorithm has a computational complexity of $\mathrm{O}(q \cdot \log q \cdot \log p)$ where $q < p$. However, if HPR is not used, the dot product test can be substituted for a computational cost of $\mathrm{O}(q \cdot \log p)$. Of course if the surfaces are thick enough to accommodate the query sphere without the possibility of including falsely-visible photons, all photon culling tests can be ignored resulting in a computational cost of $\mathrm{O}(\log p)$.

3.4 Chapter Summary

Photon mapping is a well-established algorithm that is typically associated with graphics rendering software and is known for its ability to provide high quality, physically-accurate representations of a given virtual environment while also considering the effects of participating media [9]. Photon mapping is a two-pass algorithm comprised of constructing a photon map in the first pass and performing rendering in the second.

The first pass entails sending photons from a photon source out into a 3D virtual environment and then storing the photons in a photon map. When a photon impinges upon a surface, the photon's point of intersection, color and exiting direction are stored in memory in the photon map. After intersecting with a surface, the photon's probability of reflecting, absorbing or transmitting/refracting is provided by the material's properties and is chosen using the *Russian roulette* Monte Carlo method [38]. If the photon is absorbed, no further action is performed. If reflected, a new direction for the photon is determined using the bidirectional reflectance distribution function [40]. If the photon is transmitted, Snell’s Law is used as a means of determining its exiting direction. Note that as the photon map is being constructed, the effects of the participating media that the photons must traverse through may also be accounted for [9]. Those photons which escape interaction with the participating media are stored in a *global photon map* whereas those photons which interact with the participating media are stored in a *volume photon map*. Furthermore, the type of photon considered by the photon map construction pass need not be confined to the visible light portion of the EM spectrum and may, in fact, represent any member of the EM spectrum [36, 37]. Accommodating photons that lie outside the visible light portion of the EM spectrum is a matter of adjusting the probabilities of interaction with the materials comprising the models within the virtual environment.

The photons contained in the photon map have conventionally been used in the second pass of the photon mapping algorithm by a ray-tracer in order to provide a rendering of the 3D virtual environment. Photon mapping provides a means of evaluating what can be sensed at a particular juncture in space by using the queries that would have been used to enhance ray tracing for the purpose of determining what can be sensed in the form of a photon count or photon volume.

CHAPTER 4

A NEW APPROACH TO COMPUTING APPROXIMATE SOLUTIONS TO THE ART GALLERY PROBLEM

By using the photon mapping algorithm, we are now enabled with the ability to provide approximate solutions to the AGP while taking into consideration the three questions we have posed in Section 1.1. The photons are all assumed to be propagating in a 3D environment. The photon map's data structure has information that may be accessed quickly, and whose memory requirements are scalable. By using a volumetric photon map, we may take into account participating media and by controlling the number of photons launched into a virtual environment we can simulate a photon-rich or photon-poor environment. For the purposes of our work, we are not advocating the utilization of the photon map algorithm in the traditional manner. What we are proposing is the use of the second pass of the photon mapping algorithm as a means of providing an assessment of the ability to cover a 3D surface, not as a means of helping to render that surface.

4.1 Establishing Conditions for Our Approximate AGP-solver

Our approximately optimal AGP-solver relies upon the creation of a three-dimensional grid of points surrounding the virtual environment's 3D models. Successful generation of approximately optimal AGP-solving observation posts depends upon two things:

1. The construction of a *sensor-position* grid *G*, and
2. The correct determination of the number of photons (or photon volume) capable of being sensed by the sensor occupying a grid point.

Once these conditions are met, we may then provide an approximate solution to the AGP in a 3D virtual environment.

4.1.1 Construction of a sensor-position grid

Let O be a set of points which comprise a surface and let it be a member of the compact subset $\Omega \subset \Re^3$. The set O is an *occluder* residing in *environment* Ω. The set $X = \Omega \backslash O$ is the set of points in which a (possibly) omnidirectional sensor node may be stationed and $x_\text{o} \in X$ is some sensor station. A sensor-position grid $G \subset X$ is comprised of a sequence of points $g_0, g_1, \ldots g_n$ wherein element g_i is separated by its neighboring elements by some distance such that the distance does not exceed the boundaries of G. (This distance assumption ensures that at least one sensor station is found within the confines of G.)

4.1.2 Establishing visibility for a sensor at an observation point

Understanding which photons are responsible for contributing to a sensor's ability to perceive their surroundings and those that do not is a well-established problem in photon mapping and has been discussed previously in Chapter 3.3.

4.2 Combining Concepts

Given $x \in O$ representing a point on the occluder's surface that has been illuminated by the photon mapping algorithm and a sensor residing at point $g_o \in G$ in the virtual environment, a photon map query supplemented with a photon culling method described in Chapter 3.3 produces the following function $f : R \rightarrow D$ such that range $R = \{(x, g_o) \mid x, g_o \in \Re^3 \text{ and } g_o \neq x\}$ and domain $D = \{\Re^+$ if x has a photon visible to g_o; 0 if a photon has been completely absorbed at x; $\Re^-$ if x has a photon not visible to $g_o\}$ where $\Re^+$ and $\Re^-$ are the sets of positive and negative real numbers, respectively.

Having this function available leads to the production of the following level set function [6]:

$$\Phi(x; g_o) = \min_{z \in L(x, g_o)} \Psi(z) \qquad (1)$$

where Φ is non-positive at x when x is invisible to the sensor at g_{o} and positive otherwise, $L(x, g_o)$ is a line segment connecting the sensor at the grid point x_{o} to the point x and $\Psi > 0$ when z is an unobstructed line segment connecting x and g_{o}. This function permits the distinction between the surfaces which are and are not visible to a sensor. In our terminology, the function distinguishes between those photons (and the surfaces they illuminate) which are (in)visible to a sensor.

Given the grid points, g_0, g_1, … g_n and the set of photon-illuminated points y_0, y_1 … y_p residing on the surfaces of the occluders O, the establishment of visibility is a consequence of determining the amount of non-occluded space encompassed within the viewing area sensed by n sensors. This concept is expressed in the following equation inspired by [6]

$$V(g_0, g_1, \dots g_n) = \sum_{k=0}^{p} H\left(\sum_{i=0}^{n} H\big(\Phi(y_k; g_i)\big)\right) \qquad (2)$$

where H(•) is the one-dimensional Heaviside step function, with H(0) = 0. The entire possible visible volume is

$$\sum_{k=0}^{p} H\big(\Psi(y_k)\big) \qquad (3)$$

The normalized visible volume, V_{norm}, describes the fraction of coverage proved by the sensors and is provided by the following equation

$$V_{norm} = \frac{\sum_{k=0}^{p}\left(\sum_{i=0}^{n} H\left(\Phi(y_k; x_i)\right)\right)}{\sum_{k=0}^{p} H\left(\Psi(y_k)\right)} \tag{4}$$

and $0 \leq V_{norm} \leq 1$.

The method for establishing sensor placement in order to achieve maximum coverage is provided by the following equation

$$\underset{g_j \in G}{argmax} \sum_{k=0}^{p} H\left(H\left(\Phi(y_k; g_j)\right) - H\left(\sum_{\substack{i=0 \\ i \neq j}}^{n} H\left(\Phi(y_k; g_i)\right) \right) \right) \tag{5}$$

The first term gives the visible region provided by the *j*th sensor. The second term is the visible volume of photons provided by all other sensors.

The equation provides an exhaustive search of the optimal location for one sensor at a time. The process is repeated for each sensor in a manner similar to simulated annealing. According to [6] “the optimal sensor positions may not be unique; therefore an optimal position is chosen at random in order to avoid limit cycles. Observers are updated in random order for the same reason.”

The pseudocode for the Sensor Placement Optimization via Queries (SPOQ) algorithm responsible for computing approximate solutions to the AGP in a *known* environment Ω is inspired by [6] and is given below.

Algorithm 1 SPOQ

```
1. Construct a photon map
2. for $g_i \in G$ do
3. Store the $p$ photons observed by $g_i$
4.  end for
5. Initialize $V_{norm}^{k}$ using Equation (4)
6. Initialize $V_{norm}^{k-1} = 0$
7. while $V_{norm}^{k} > V_{norm}^{k-1}$ do
8.        for all $n$ sensors do
9.            choose a sensor $j$ at random without replacement
10.           find a globally optimal solution to Eq. (5) for sensor $j$
11.       end for
12.       $V_{norm}^{k-1} = V_{norm}^{k}$
13.     Recompute $V_{norm}^{k}$ using new sensor positions
14. end while
```

What distinguishes this version of SPOQ from [6] is that at each grid point $g_i \in G$, the number of photons obtained at that grid point are stored prior to executing the while loop. In an *unexplored* environment, then steps 1 – 4 would be omitted and step 10 would entail making a query to the photon map rather than doing a look up of what had been stored previously at grid point g_i.

We use the following algorithm detailing the steps taken in step 10 of SPOQ:

Algorithm 2 SPOQ: Photon query step

```
1. Set sensor $j$'s visibility to 0
2. Get the current visibility volume for all sensors other than $j$
3. for all $g_i \in G$ do
4.     Search the photon map to get photons who contribute to the visibility of sensor $j$ using photon culling
5.     Set the visible volume equal to the summation of all previously seen visible volumes
6. end for
7. Select maximum visible volume
```

Recall that distinguishing between those photons that contribute to visibility and those that do not is a matter of culling those photons which do not contribute to visibility via the dot-product test or some other means described in Section 3.3.

The significance of this effort is that it allows for the realization of an *online* approximately optimal AGP-solver. This property is achieved by the fact that once a sensor-position grid or photon map has been altered, fast photon map queries allow for SPOQ to adapt the selection of approximately optimal AGP-solving points in reaction to the grid's alteration. As we shall see in Chapter 6, having this online ability to provide approximately optimal solutions to the AGP can enhance UxV autonomy by promoting route-adaptability.

4.2.1 Algorithm Analysis

By using queries to the photon mapping algorithm's photon map as input to SPOQ, it then becomes possible to solve the AGP problems while taking into account participating media in complex 3D virtual environments illuminated by different types of EM radiation. The computation and memory requirements depend on the number of photon sources used and whether the environment is known or unknown. When dealing with an unknown environment, for each of the *l* sensors who will occupy the *m* observation points, queries to the photon map with photon culling have a computational cost of $O(q \cdot \log q \cdot \log p)$ where *p* is the number of photons launched into the virtual environment, *q* is the number of photons returned by a query and $q < p$. The visibility optimization in an unknown environment requires $O((lm) \cdot q \cdot \log q \cdot \log p)$ operations whereas a known environment requires $O(lm)$ operations where $l << m$. The difference stems from the fact that in an unknown environment a new photon map query with photon culling must be performed at each sensor position in order to update the information obtained as the sensor explores the 3D virtual environment whereas in a known environment the photon map query result may be precomputed at each observation point and then utilized by each respective sensor. The memory requirement is $O(rp)$ for *r* photon sources where *r* is is an integer greater than or equal to 1 for an unknown environment and $O(lmq)$ for a known environment.

4.2.2 Addressing the Three Questions Posed In Section 1.1

When addressing the questions posed in Section 1.1, the prospect of providing coverage in a 3D environment arrives immediately as a result of the very nature of the use of photon mapping. Every photon used to solve the AGP is presumed to propagate in a 3D environment.

The first question is addressed by the fact that our SPOQ method is capable of handling either known or unknown environments. The second question is addressed by SPOQ by introducing the concept of a volumetric photon map; we can simulate the behavior of participating media [9, 45]. The third question is addressed by SPOQ by considering that in photon-rich or photon-poor environments such as that encountered during day or night time respectively; SPOQ can determine where to put sensors in order to capture as much information as possible. All gradations of light availability can be modeled by simply varying the number of photons and photon sources used in the simulated environment.

4.2.3 Steps Taken to Solve the AGP Using SPOQ

The SPOQ method may be used by following the steps given below:

1. Create the virtual environment's model
2. Construct the sensor-position grid.
3. Apply photon mapping to the virtual environment.
4. Apply SPOQ
5. Collect results.
6. Alter the grid or model in some way
7. Repeat steps 3 through 6.

Step 1 entails using some modeling program. This program may be something similar to Google's Sketchup [46] which offers many free well-crafted models through its 3D Warehouse [47]. Constructing the sensor-position grid in step 3 is a matter of determining the dimensions of the sensor-position grid and then populating it with points separated by some regular predetermined distance such that they do not exceed the dimensions of the sensor-position grid.

Applying photon mapping in step 4 is a matter of deciding how many photons will be used. This choice can be used to simulate a photon rich (daytime) or photon poor (nighttime) environment. This step can also be a point where a volumetric photon map may be introduced.

Application of SPOQ requires the user to set the dimensions of the sensor node's sensing range (or the radius of the query-sphere). After application of SPOQ, the results are collected and the user now knows the positions that *k* sensors must occupy in order to sense as many photons as possible given their sensing range and environmental circumstances.

4.3 Problem Statement

To demonstrate the efficacy of our algorithm, we applied it to the following scenarios:

1. Using 1188 observer grid points and 2500, 5000 and 7500 photons cast on the Cityscape model; using 15000, 20000 and 25000 photons cast on the Canyon and Arena models with 990 and 2520 grid points, respectively.
2. Using the same respective number of photons cast with a 6897-points observer grid for the Cityscape model; using a 1152- and 4550-point observer grid for the Canyon and Arena models, respectively.
3. Repeating the first scenario but using randomly-generated forbidden zones constituting 10%, 15% and 20% of the grid points available, and
4. Repeating the first scenario but using a set of forbidden zones representing a moving obstacle constituting 10% of the grid points available moving gradually over 3 time steps.

The first scenario may be regarded as a means of testing SPOQ with gradually increasing “brightness” in terms of photons launched, the second scenario represents the scalability of the SPOQ algorithm when applied to larger sensor-position grids and the last two scenarios represent an environment with increasing difficulty placing sensors. In each scenario, our goal is to show that we can sense as many photons as possible. In each case, we applied our algorithm to 2, 3, 4 and 5 sensors. In all cases, the radius of the query sphere was set at the length of the arbitrary model depicted. Furthermore, in all cases, SPOQ was applied to an initially *unknown* environment.

4.4 Results

The first step mentioned above is shown below in Figure 4.1 with the city, Canyon and Arena models rendered using Sketchup. The construction of the accompanying sensor-position grid, application of photon mapping and the results of applying SPOQ for the three models are depicted respectively in Figures 4.2, 4.3, and 4.4.

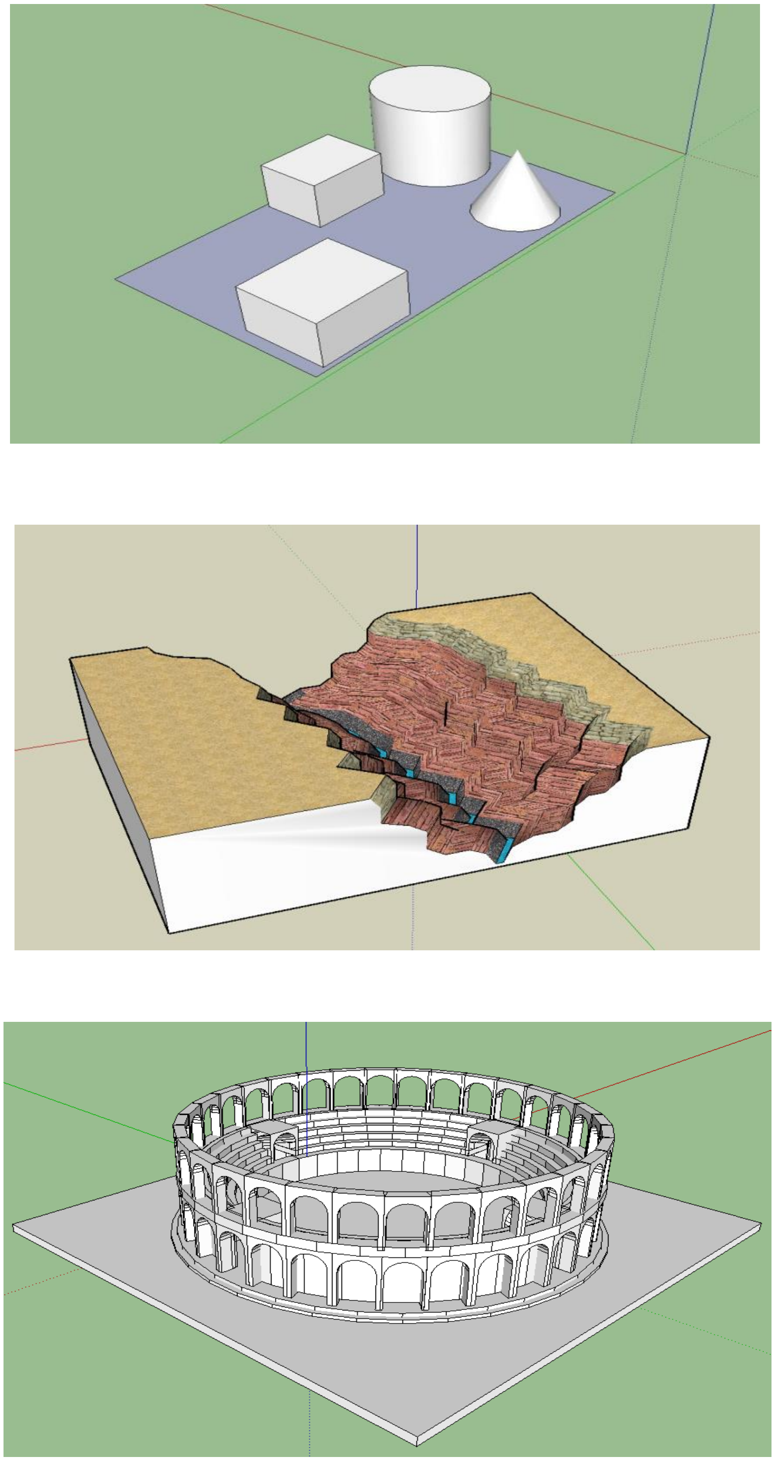


Figure 4.1: This sequence of images depicts three 3D models rendered using Sketchup.

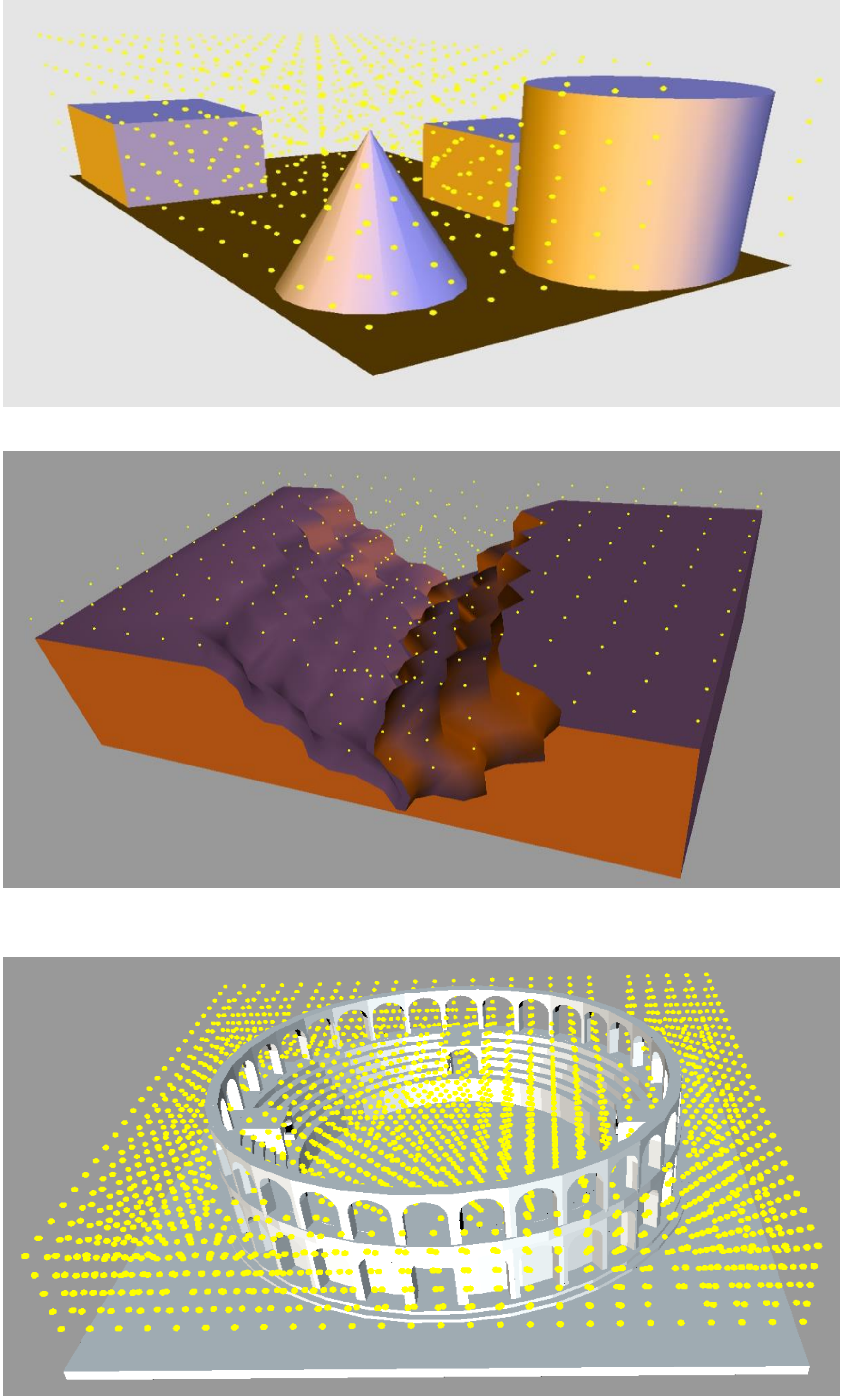


Figure 4.2: The yellow spheres represent the result of constructing a sensor grid to our three models

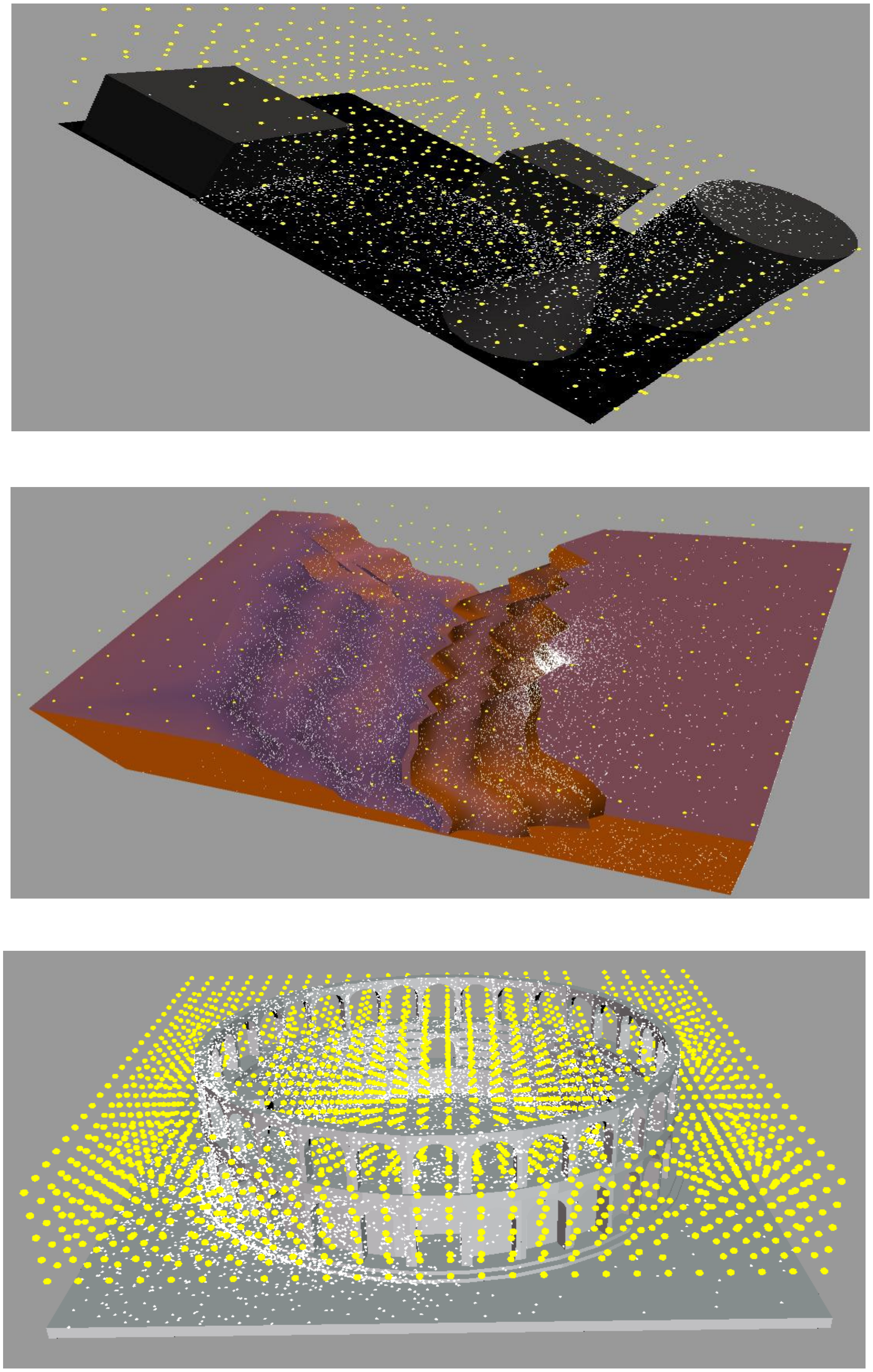


Figure 4.3: Here we see the result (white dots) of applying of photon mapping to our three models.

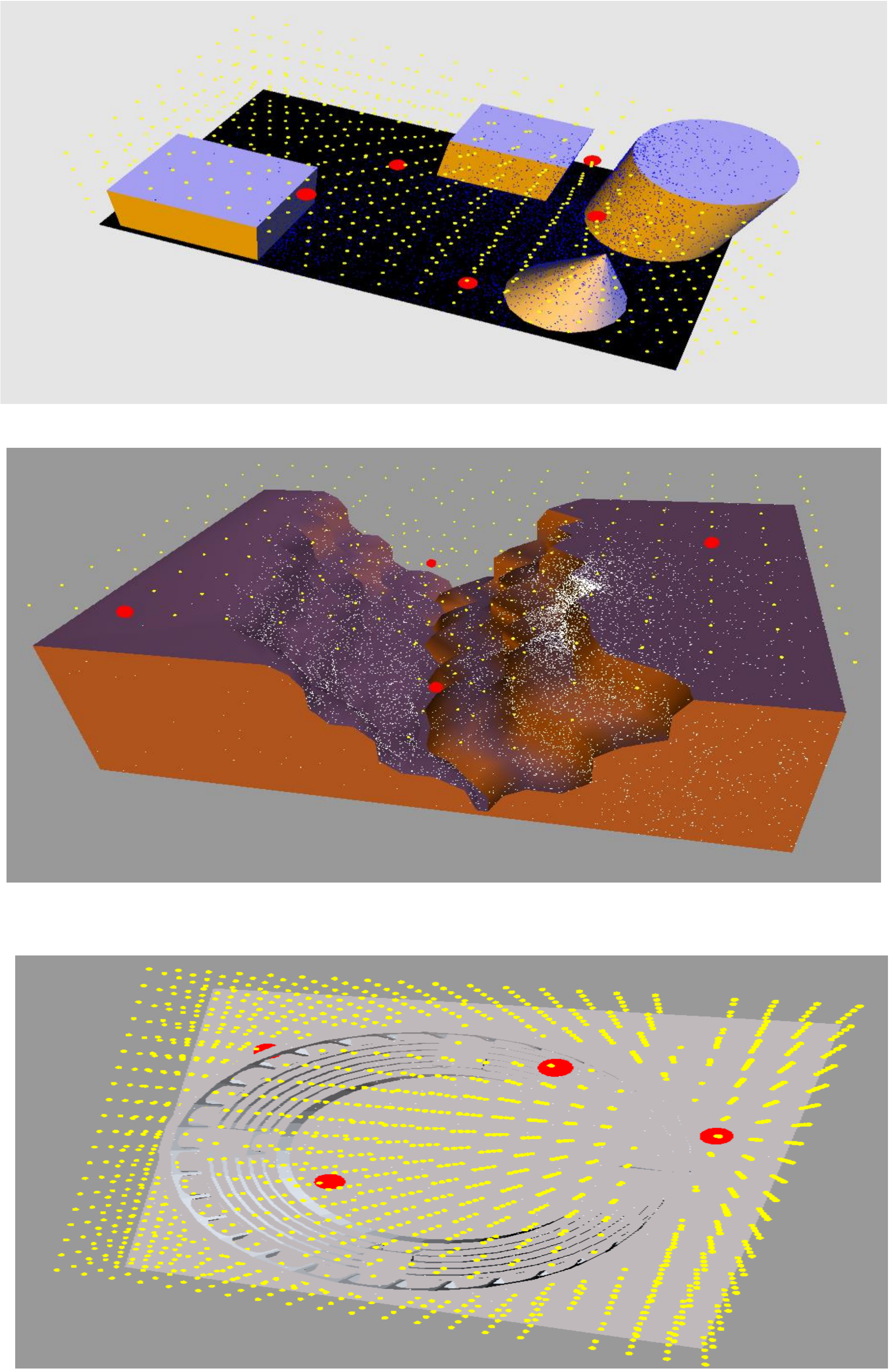


Figure 4.4: This sequence of images depicts the result of applying SPOQ to our three models. These points sense the maximum number of unique photons available.

The following tables represent the coverage obtained by the SPOQ simulation algorithm on the models depicted in the figures above using successively greater numbers of photons. The successively greater number of photons represents an environment that transitions from a photon poor or "dark" environment to a photon rich or "bright" environment. The environment was unknown prior to using SPOQ.

Table 4.1: Coverage Obtained Under Circumstances Described by Scenario 1

Observers	% Photons Covered			Time (seconds)		
Low Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	89.4	88.7	84.7	0.47	0.46	0.82
3	92.9	90.7	87.4	0.33	0.55	0.93
4	94.5	92.1	88.3	0.44	0.71	0.97
5	93.4	92.4	89.1	0.73	0.88	1.1
Medium Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	90.2	89.7	85.3	0.7	0.61	3.5
3	93.3	91.8	87.1	0.9	0.67	3.4
4	93.5	92.1	87.9	1.1	1.2	4.2
5	93.9	92.0	89.9	1.4	0.98	6.9
High Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	89.6	91.9	85.4	0.89	0.9	1.0
3	93.6	92.0	86.6	1.3	1.3	2.4
4	93.2	92.2	87.9	1.2	1.8	3.5
5	94.1	93.1	86.2	1.5	5.2	6.4

Table 4.2: Coverage Obtained Under Circumstances Described by Scenario 2

Observers	% Photons Covered			Time (seconds)		
Low Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	90.7	89.6	86.1	2.1	1.2	2.8
3	94.2	91.3	87.5	3.1	1.9	3.6
4	93.6	92.1	87.9	6.8	3.3	4.8
5	94.2	92.3	88.3	8.4	4.2	7.5
Medium Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	89.6	91.2	85.6	4.3	1.5	4.8
3	93.6	92.1	87.4	6.1	1.8	5.6
4	93.5	92.0	88.0	8.1	2.8	5.9
5	94.1	92.3	88.9	6.8	3.1	7.2
High Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	90.2	91.4	85.6	6.1	1.6	4.1
3	93.7	92.1	87.1	8.7	2.2	5.4
4	95.1	92.3	88.2	9.1	3.3	7.2
5	95.2	92.5	89.0	9.7	6.7	7.6

Table 4.3: Coverage Obtained Under Circumstances Described by Scenario 3

Observers	% Photons Covered			Time (seconds)		
Low Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	89.6	89.7	85.7	0.8	0.51	0.91
3	92.9	90.4	86.4	0.86	0.54	1.1
4	93.4	91.6	88.1	1.6	0.63	1.3
5	93.6	92.2	88.7	1.9	0.91	1.4
Medium Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	89.7	88.4	86.1	0.56	0.71	3.4
3	92.7	91.5	86.2	0.83	0.77	4.0
4	93.6	92.1	88.1	1.6	1.2	4.2
5	93.1	92.6	89.9	1.4	1.3	7.2
High Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	89.7	91.4	85.6	0.7	1.1	1.6
3	93.4	92.5	86.8	1.0	1.5	3.5
4	93.2	92.7	87.6	1.2	4.9	4.2
5	92.1	93.1	87.9	1.3	6.2	5.9

Table 4.4: Coverage Obtained Under Circumstances Described by Scenario 4

Observers	% Photons Covered			Time (seconds)		
Low Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	88.9	87.4	85.6	0.8	0.50	1.2
3	91.4	91.6	87.6	0.92	0.64	1.6
4	93.6	92.4	88.2	1.1	0.74	2.3
5	94.0	93.1	88.5	1.4	0.88	3.5
Medium Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	89.5	89.4	86.1	0.76	0.76	3.4
3	92.4	90.5	86.5	1.1	0.82	4.5
4	93.7	91.1	88.1	1.3	1.6	5.3
5	92.2	92.2	89.7	1.4	2.1	7.6
High Photon Count						
	City	Canyon	Arena	City	Canyon	Arena
2	90.7	91.1	86.3	0.91	2.1	2.3
3	91.4	92.6	86.6	1.1	2.6	3.7
4	92.5	92.8	87.5	1.3	5.9	5.2
5	92.0	93.3	88.9	1.5	7.1	6.6

4.5 Summary and Conclusions

Determining optimal WSN coverage is an instance of the art gallery problem (AGP) which asks "What is the minimum number of sensors necessary to assure that every point in the monitored space is sensed?" We recast the AGP by asking the question "What is the minimum number of sensors necessary to assure that every photon available to convey information about the monitored space is sensed?" When the AGP is posited in this manner, we have a means of determining WSN coverage optimization that uses the modeled behavior of photons to be sensed. Modeling the propagation of a set of virtual photons through a virtual environment may be accomplished by the photon mapping algorithm. This algorithm produces a queryable data structure called a photon map containing the virtual photons' contribution to the sensibility of the virtual environment’s surfaces. In order to accomplish the goal of computing approximate solutions to the AGP in three dimensions, we present the Sensor Placement Optimization via Queries (SPOQ) algorithm that uses level sets populated by queries to a photon map in order to find observation points that sense as many photons as possible. Since we are using photon mapping as our means of modeling photons, SPOQ can take into account static or dynamic environmental conditions and can use exploratory or precomputed sensing. The SPOQ method is computationally efficient, requiring less memory than other approximately optimal AGP solutions.

We have introduced a new means of computing approximately optimal solutions to the AGP that relies upon the modeling of a physical phenomenon. The introduction of photon mapping – which ultimately relies on stochastic means of determining the behavior of photons as the conveyors of information about an environment in order to solve the AGP – means that we cannot absolutely guarantee that every point lying on a 3D surface will be illuminated. Hence, we state that our solution is approximately optimal.

An objection to our technique for achieving optimal visibility coverage is that by placing sensors at locations which are illuminated by photons, we are neglecting those areas which are not so illuminated. This objection is countered by simply using additional photon sources or by attaching a photon emitter to the photon sensor. The use of such a sensor/emitter sensor node configuration would necessarily entail longer photon map construction times and larger memory consumption, but would not affect the order of operation of the execution of SPOQ.

Another objection to our method is that we have not stated *exactly* how many photons should be launched by a photon source in order to illuminate an environment. The question of exactly how many photons to use by a photon source is in fact an open topic of research that was mentioned by Jensen [9]. More recently, the topic was considered in [48] wherein the authors discussed a *progressive photon mapping* method. Their method starts with the launching of *p* photons and then increases the number of photons launched in the virtual environment until the correct radiance estimate for a surface is achieved. Understanding the optimal number of photons to launch from a photon source *initially* is a problem that is still under investigation.

Another objection might be that we have assumed the use of an omnidirectional sensor. While such sensors do exist [49], a more realistic sensor would be a non-omnidirectional sensor which receives photons from a viewing frustum. This limitation is overcome by the fact that if we are given a query-sphere, we can carve it into a viewing frustum or any other shape. Once the query-sphere has been reshaped into a *query-volume*, we can make the query-volume directional by only allowing the inclusion of those photons that have arrived from a particular direction into the query-volume. For example, if we carve a query-sphere into a viewing frustum, we can make the viewing frustum only perceive those photons that have arrived from the front of the viewing frustum and exclude the rest. The SPOQ algorithm can then be applied as mentioned in Chapter 4, Section 3.3 but we may limit its use to only the photons perceived at the viewing-frustum's opening.

While we acknowledge these objections to the use of SPOQ, we believe that the advantages of SPOQ overwhelm these objections. We believe that SPOQ will prove useful for future research endeavors within the WSN community.

CHAPTER 5

PHOTOM-MAPPING INFORMED MULTISPECTRAL CHAN-VESE SEGMENTATION

5.1 Motivation for Using the Chan-Vese Method of Segmentation

As was discussed in Section 2.3, the variational watchman routes we presented were founded on minimizing an energy functional. In a similar manner, the motivation for using the Chan-Vese method of segmentation is to minimize the Mumford-Shah energy functional [12]. This functional – given below in section 5.3 - has typically been associated with image segmentation. Its purpose is to provide a minimum length curve in 2D space that defines the border of the region of interest in the image to be segmented. The key feature of the Chan-Vese method of segmentation is its ability to utilize the Mumford-Shah energy functional to evolve an initial segmentation curve such that it makes a distinction between pixels in an image [11]. The Chan-Vese segmentation algorithm can thus be regarded as a search algorithm meant to establish such a distinction. “Energy minimization” for the Chan-Vese segmentation algorithm occurs when a smooth curve conforms to the boundary of the object to be segmented [11].

In order to understand and enable greater surveillance capabilities (for candidate waypoints that may solve the WRP) in arbitrary environments, it is necessary to take into account the influence of sensing spectra extending beyond the visible range as well as provide paths for the UxV responsible for performing surveillance to follow. Having a UxV capable of sensing its surroundings while it traverses to fixed points of visitation (or *waypoints*) leads us to ask the following questions:

1. How do we decide what areas are illuminated and which are not in order to best determine what is able to be sensed by a UxV?
2. How do we inform the UxV's waypoint path planner such that it will traverse through those areas which are more (or less) illuminated (and thus provide an approximately optimal solution to the WRP)?

In order to answer these questions we will use photon mapping. Recall that the number of photons obtainable by a UxV at a point in space may be regarded as that point's *photon volume*. Conversely, if a point is not well-illuminated, it may be regarded as being *shadowed*. By using the photon mapping algorithm, we are enabled with the ability to distinguish between areas with greater and lesser photon volume in a 3D environment while taking into consideration the factors we considered in Chapter 1 namely: UxV-sensor attenuation due to environmental effects, multiple EM spectra, and UxV-sensing which has been altered after the passage of time.

When we say that we are performing segmentation, we mean that we are producing a connected 3D mesh within a 3D virtual environment such that those vertices comprising the mesh may serve as UxV waypoints affording a greater photon volume within the UxV's sensing range than waypoints who are not elements of the mesh. This segmentation mesh allows us to answer the above questions in the following manner:

1. The UxV's sensing capabilities can be determined by a measurement of the photon volume available within the UxV's sensing range at a particular vertex on the mesh and
2. Approximately optimal WRP-solving UxV routes can be confined to those vertices on the segmentation mesh that provide the greatest (or least) photon volume.

5.2 Utilizing the Chan-Vese Method of Segmentation

Our segmentation method relies upon the creation of a connected three-dimensional grid of points (or vertices) surrounding the virtual environment's 3D models such that these vertices sense more photons than would be sensed by elements not in the mesh. These points are waypoints contained in a *UxV-position grid* in which a UxV may move to or remain stationed. Successful segmentation results in the creation of a mesh encompassing those UxV-position points that possess greater photon volume within the UxV's sensing range than those who do not. Successful segmentation depends upon three things:

1. The construction of a UxV-position grid,
2. The correct determination of the number of photons (or photon volume) capable of being sensed by the UxV occupying a grid point, and
3. Initialization of the Chan-Vese segmentation algorithm.

Once these three conditions are met, we may then segment the 3D virtual environment to obtain our desired results.

5.2.1 Construction of a UxV-position grid

Understanding how the UxV-position grid *G* is constructed is similar to how the sensor observation grid was constructed in Chapter 4.2 In our application; we assume an equal geodesic distance of one separating the grid points as well as full connectivity among the observer stations in *G*.

5.2.2 Determining Which Photons are Perceptible

Understanding which photons are responsible for contributing to a UxV's ability to perceive their surroundings and those that do not has been considered previously in Chapter 3.3.

5.2.3 Establishing Initial Conditions for Chan-Vese Segmentation

The establishment of the initial segmentation curve for the Chan-Vese segmentation method is of critical importance to the outcome of the segmentation. This topic has been considered previously in [50] and [51]. The essence of the initialization problem is expressed by the fact that if a local minimum is found at the beginning of the Chan-Vese segmentation method's execution, then the segmenting curve will not advance.

We propose a new means of establishing the Chan-Vese segmentation algorithm's initial condition. Since our goal is to construct a mesh that maximizes the photon volume sensed at the vertices comprising the mesh, our initial condition will be a curve comprised of connected waypoints that have a high sensed photon volume. To establish the waypoints comprising this initial segmentation curve, we will solve the AGP using the SPOQ algorithm outlined in Chapter 4.3.

Recall that the AGP is posited by the mathematics community as the following question:

- What is the minimal number of observers necessary to ensure that the maximum number of points in environment Ω are observed?

In our case, we want to find the minimal number of observers necessary to ensure that the maximum number of *uniquely*-observable photon volume is obtained at a particular observation point. Having those UxV-position stations that provide maximum uniquely-observed photon volume allows us to establish an initial segmentation curve that provides the highest probability of performing segmentation.

Given an environment Ω populated with a set of UxV-position grid points, *G*, illuminated by *p* photons, the SPOQ algorithm produces *j* grid points $g_0 \ldots g_j \subseteq G$ which have the greatest uniquely-observed photon volume sensed by a UxV stationed at a grid point $g \in g_0 \ldots g_j$. The initial segmentation curve is created by connecting the grid points comprising g_0 ... g_j by some means (such as Dijkstra's path-finding algorithm [52]) such that they form a closed curve. Since the Chan-Vese segmentation algorithm relies upon region competition of energy gradients as a means of evolving the segmentation curve [11], the establishment of an initialization curve comprised of UxV-position grid points that are (highly) distinguished from neighboring grid point thus leads to a non-zero influence on the segmentation curve's evolution. Hence, by taking the effort to solve the art gallery problem, we can ensure a high likelihood of segmentation occurring.

5.3 Combining Concepts

Given environment Ω containing UxV-position grid points G, let the function $V{:}G\rightarrow\{0\}\cup\Re^{+}$ represent the photon volume obtained by some means at an element of G. Our goal is to find a minimum to the functional $F(\phi)$ below provided by Mumford and Shah [12] for some function $\phi{:}G\rightarrow\Re$.

$$F(\phi) = \mu\left(\sum_{i=1}^{n}\left|\nabla H(\phi_i)\right|\right)^{b} + \nu\sum_{i=1}^{n}H(\phi_i) + \lambda_1\sum_{i=1}^{n}|V-c_1|^2 H(\phi_i) + \lambda_2\sum_{i=1}^{n}|V-c_2|^2\left(1-H(\phi_i)\right) \qquad (6)$$

where μ, ν, λ_1, λ_2 and b are arbitrarily defined parameters and H is the Heaviside function

$$H_\varepsilon(x) = \frac{1}{2}\left(1+\frac{2}{\pi}\tan^{-1}\left(\frac{x}{\varepsilon}\right)\right)$$

As the signed distance function ϕ - or segmentation mesh - evolves, the values assumed for an input $g \in G$ are given below

$$\begin{cases} \phi > 0, & if\ g \text{ is within the mesh} \\ \phi = 0, & if\ g \text{ is on the perimeter of the mesh} \\ \phi < 0, & if\ g \text{ is outside the mesh} \end{cases}$$

The Mumford-Shah functional [12] used by Chan and Vese [11] is obtained by setting μ = 0.5, ν = 0, λ_1 = 1, λ_2 = 1and b = 1. (In our application, we do something similar.) The values of c_1 and c_2 are the region averages of V in the respective regions where $\phi \geq 0$ and $\phi < 0$ and are given below.

$$c_1 = \frac{\sum_{i=1}^{n} V \cdot H(\phi_i)}{\sum_{i=1}^{n} H(\phi_i)}$$

$$c_2 = \frac{\sum_{i=1}^{n} V \cdot (1 - H(\phi_i))}{\sum_{i=1}^{n}(1 - H(\phi_i))}$$

In [11] Chan and Vese use Euler-Lagrange equations and the gradient descent method to produce the following discretized partial differential equation for the level set function ϕ to minimize $F(\phi)$ at a particular time step t.

$$\phi_t = \tilde{\delta}_h(\phi)[\mu \cdot \kappa - \nu - \lambda_1(V - c_1)^2 + \lambda_2(V - c_2)^2] \qquad (7)$$

where ϕ_t is the level set representation of the evolving segmentation curve at some time step t, κ represents the segmentation curve's curvature and $\tilde{\delta}_h$ is the discrete delta function which ensures curve smoothness. In Chan and Vese's original paper, the value of $\tilde{\delta}_h$ was $\varepsilon/(\pi(\varepsilon^2 + \phi^2))$ where ε is a (small) positive constant. In our application, the value of $\tilde{\delta}_h$ is calculated by means of the algorithm provided by Smereka in [53].

We will now show how we discretize and solve the above PDE numerically in three dimensions. Let $\phi_{i,j,k}^{n}$ denote the value of the evolving curve at grid point i,j,k at iterative step n. We use the following notation given by [54] for the spatial finite differences. We presume equal (i.e. geodesic) spacing h = 1 between grid points.

$$\Delta_+^x \phi_{i,j,k}^n = \phi_{i+1,j,k}^n - \phi_{i,j,k}^n$$

$$\Delta_-^x \phi_{i,j,k}^n = \phi_{i,j,k}^n - \phi_{i-1,j,k}^n$$

$$\Delta_+^y \phi_{i,j,k}^n = \phi_{i,j+1,k}^n - \phi_{i,j,k}^n$$

$$\Delta_-^y \phi_{i,j,k}^n = \phi_{i,j,k}^n - \phi_{i,j-1,k}^n$$

$$\Delta_+^z \phi_{i,j,k}^n = \phi_{i,j,k+1}^n - \phi_{i,j,k}^n$$

$$\Delta_-^z \phi_{i,j,k}^n = \phi_{i,j,k}^n - \phi_{i,j,k-1}^n$$

In order to simplify the notation used in our equations for expressing the solution to the discretized version of equation (3), we use the following finite central differences which are inspired by [25]:

$$\Delta^x \phi_{i+1/2,j,k}^n = \phi_{i+1,j+1,k}^n + \phi_{i,j+1,k}^n - \phi_{i,j-1,k}^n - \phi_{i+1,j-1,k}^n$$

$$\Delta^x \phi_{i,j,k+1/2}^n = \phi_{i-1,j+1,k}^n + \phi_{i,j+1,k}^n - \phi_{i,j-1,k}^n - \phi_{i-1,j-1,k}^n$$

$$\Delta^y \phi_{i+1/2,j,k}^n = \phi_{i+1,j+1,k}^n + \phi_{i+1,j,k}^n - \phi_{i-1,j+1,k}^n - \phi_{i-1,j,k}^n$$

$$\Delta^y \phi_{i,j,k+1/2}^n = \phi_{i,j-1,k+1}^n + \phi_{i,j,k+1}^n - \phi_{i,j-1,k-1}^n - \phi_{i,j,k-1}^n$$

$$\Delta^z \phi_{i+1/2,j,k}^n = \phi_{i+1,j,k+1}^n + \phi_{i+1,j,k}^n - \phi_{i-1,j,k}^n - \phi_{i-1,j,k+1}^n$$

$$\Delta^z \phi_{i,j,k+1/2}^n = \phi_{i,j+1,k+1}^n + \phi_{i,j+1,k}^n - \phi_{i,j-1,k}^n - \phi_{i,j-1,k+1}^n$$

The central difference formulations for$\Delta^x \phi_{i-1/2,j,k}^n$ $\Delta^x \phi_{i,j,k-1/2}^n$, $\Delta^y \phi_{i-1/2,j,k}^n$, $\Delta^y \phi_{i,j,k-1/2}^n$, $\Delta^z \phi_{i-1/2,j,k}^n$ and $\Delta^z \phi_{i,j,k-1/2}^n$ follow a pattern similar to that shown above.

Let m_x, m_y and m_z be the number of points of grid G in the x, y and z directions, respectively. The von Neumann boundary conditions apply namely:

$$\phi^n_{0,j,k} = \phi^n_{1,j,k}, \phi^n_{m_x,j,k} = \phi^n_{m_x-1,j,k}$$

$$\phi^n_{i,0,k} = \phi^n_{i,1,k}, \phi^n_{i,m_y,k} = \phi^n_{i,m_y-1,k}$$

$$\phi^n_{i,j,0} = \phi^n_{i,j,1}, \phi^n_{i,j,m_z} = \phi^n_{i,j,m_z-1}.$$

We use the following variables in order to streamline our notation of the iterative, numerical solution to ϕ^{n+1}:

$$C_1 = \left(\left(\Delta^x_+ \phi^n_{i,j,k}\right)^2 + \frac{\left(\Delta^x \phi^n_{i+1/2,j,k}\right)^2}{4} + \frac{\left(\Delta^x \phi^n_{i,j,k+1/2}\right)^2}{4}\right)^{-\frac{1}{2}}$$

$$C_2 = \left(\left(\Delta^x_- \phi^n_{i,j,k}\right)^2 + \frac{\left(\Delta^x \phi^n_{i,j+1/2,k}\right)^2}{4} + \frac{\left(\Delta^x \phi^n_{i,j,k+1/2}\right)^2}{4}\right)^{-\frac{1}{2}}$$

The variables C_3, C_4, C_5 and C_6 follow a similar pattern to that shown above in the sense that the first element in the summation is$\left(\Delta^y_+ \phi^n_{i,j,k}\right)^2$,$\left(\Delta^y_- \phi^n_{i,j,k}\right)^2$, $\left(\Delta^z_+ \phi^n_{i,j,k}\right)^2$, and $\left(\Delta^z_- \phi^n_{i,j,k}\right)^2$respectively. The second and third elements in the summation conform to a similar pattern as that shown in C_1 and C_2.

The method advocated in [54] provides us with the following iterative solution to ϕ^{n+1}:

$$\phi^{n+1}_{i,j,k} = F_1\phi^{n+1}_{i+1,j,k} + F_2\phi^{n+1}_{i-1,j,k} + F_3\phi^{n+1}_{i,j+1,k} + F_4\phi^{n+1}_{i,j-1,k} + F_5\phi^{n+1}_{i,j,k+1} + F_6\phi^{n+1}_{i,j,k-1} + Fw_{i,j,k} \tag{8}$$

where

$$F_l = \frac{\Delta t \tilde{\delta}_h \left(\phi^n_{i,j,k}\right) \mu(p \cdot L(\phi^n)^{p-1}) C_l}{h \, + \, \Delta t \tilde{\delta}_h \left(\phi^n_{i,j,k}\right) \mu(p \cdot L(\phi^n)^{p-1}) \left(\sum_{g\,=\,1}^{6} C_g\right)}$$

for *l* = 1, 2, 3, 4, 5, 6,

$$F = \frac{h}{h \, + \, \Delta t \tilde{\delta}_h \left(\phi^n_{i,j,k}\right) \mu(p \cdot L(\phi^n)^{p-1}) \left(\sum_{g\,=\,1}^{6} C_g\right)}$$

and the error factor $w_{i,j,k}$ is

$$w_{i,j,k} = \phi^n_{i,j,k} - \Delta t \tilde{\delta}_h \left(\phi^n_{i,j,k}\right) \left(\nu \lambda_1 \left(V_{i,j,k} - c_1(\phi^n)\right)^2 - \lambda_2 \left(V_{i,j,k} - c_2(\phi^n)\right)^2 \right)$$

The variable $L(\phi^n)$ is the surface area of the segmentation mesh and the formula for its calculation is:

$$L(\phi^n) = \sum_{i=1}^{n} \tilde{\delta}_h(\phi^n_i) \left|\nabla \phi^n_i\right| .$$

The delta function $\tilde{\delta}_h$ serves to smooth the segmentation mesh as it evolves. For details regarding its calculation, please refer to [53].

5.3.1 Steps Taken by the Chan-Vese Algorithm

The steps for our photon mapping informed active contour Chan-Vese algorithm are given below:

1. Construct a 3D model of the virtual environment

2. Construct the UxV-position grid *G*

3. Apply the photon mapping algorithm to the virtual environment

4. Obtain the photon volume at every grid point $g_i \in G$

5. Utilize SPOQ in the manner described in Chapter 4, Section 4.3 in order to establish the initial condition $\phi_{i,j,k}^{0}$to the Chan-Vese segmentation algorithm.

6. Apply the iterative solution to $\phi_{i,j,k}^{n+1}$ given by Equation 8 using the initial conditions established in Step 5.

5.3.2 Algorithm Analysis

The construction of the UxV-position grid in step 2 has a O(*n*) complexity where *n* is the number of grid points in the virtual environment. The application of the photon mapping algorithm in step 3 has a O(*p*) complexity where *p* is the total number of photons launched in the environment [9]. Obtaining the photon volume at every grid point prescribed in step 4, depending upon the method used as described in Chapter 3 Section 3.2, can have a $O(n \cdot \log p)$, $O(n \cdot q \cdot \log p)$, or $O(n \cdot q \cdot \log q \cdot \log p)$ complexity where *q* is the number of photons returned by a photon query and $q < p$. The application of SPOQ wherein the photon volume is calculated for every grid point prior to its operation has a time complexity of $O(l \cdot n)$ where *l* is the number of way points that obtain maximum visibility and $l << n$. The iterative Chan-Vese algorithm we have described has a complexity of O(*n*) [55].

5.4 Problem Statement

To demonstrate the efficacy of our algorithm, we applied it to the following scenarios:

1. Using 1188 observer grid points and 5000 photons cast on the Cityscape model; using 25000 photons cast on Canyon and Arena models with 990 and 2520 grid points, respectively.
2. Using 7500 photons cast with a 1188-points observer grid for the Cityscape model; using 25000 photons cast on a 990 and 2520-point observer grid for the Canyon and Arena models, respectively.
3. Repeating the first scenario but using 6872 grid points for the Cityscape, 2520 grid points for the Canyon and 4550 grid points for the Arena model, and
4. Repeating the second scenario but using 6872 grid points for the Cityscape, 2520 grid points for the Canyon and 4550 grid points for the Arena model.

The first scenario may be regarded as a "baseline", the second scenario represents a low photon count (i.e. dim) environment and the last two scenarios demonstrate the scalability of the algorithm to more complicated grids.

In each case, the Chan-Vese algorithm was compared to a *k*-means classifier arriving from the ITK toolkit [56] that clusters around "light" and "dark" grid points. The elements that *k*-means operates on consists of an array of size $|G|$ wherein the *i*th element corresponds to element $g_i \in G$ and that element's value is g_i's photon volume. The initial mean value of the cluster of "light" grid points arrives from the photon volume of those points used as the initial condition to the Chan-Vese algorithm. The resulting segmentation mesh provides a set of bright grid elements where the most photons may be sensed. This segmentation algorithm mesh is compared with our Chan-Vese algorithm in the Results section.

5.5 Results

Figure 5.1 starts where the results visualized in Chapter 4.4 ended. Here we are showing the approximately optimal AGP-solving points after they have been connected in the Cityscape model. Figure 5.2 shows the output of the Chan-Vese segmentation algorithm operating in the circumstances described by scenario 1 and applied to our Cityscape model.

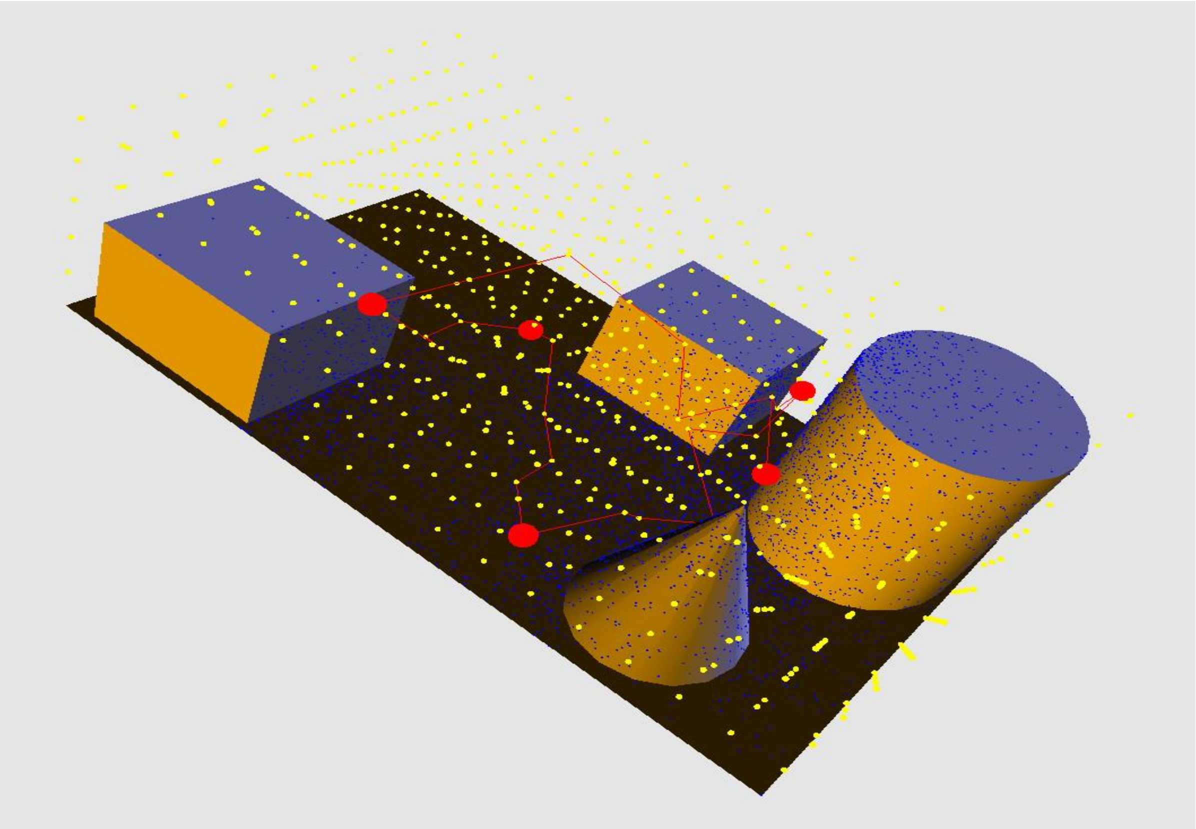


Figure 5.1: The red lines indicate the utilization of Dijkstra's algorithm to establish initial conditions for our Chan-Vese algorithm.

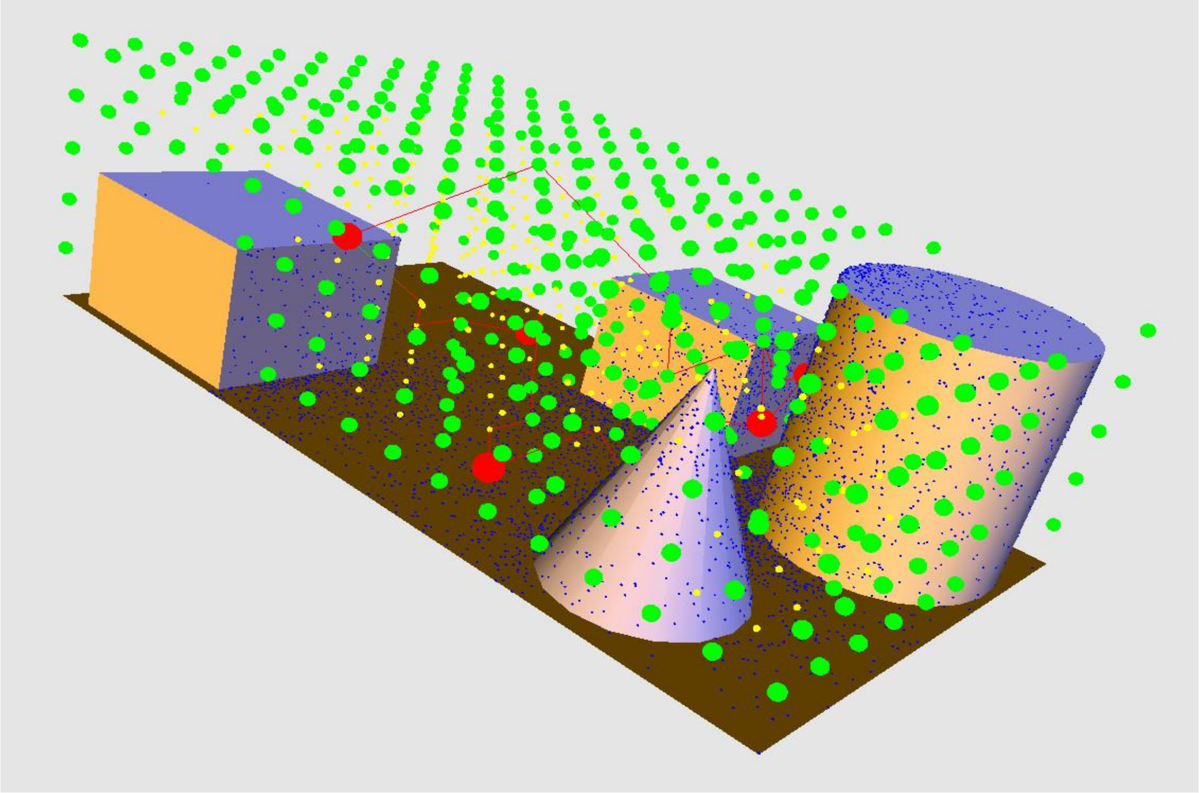
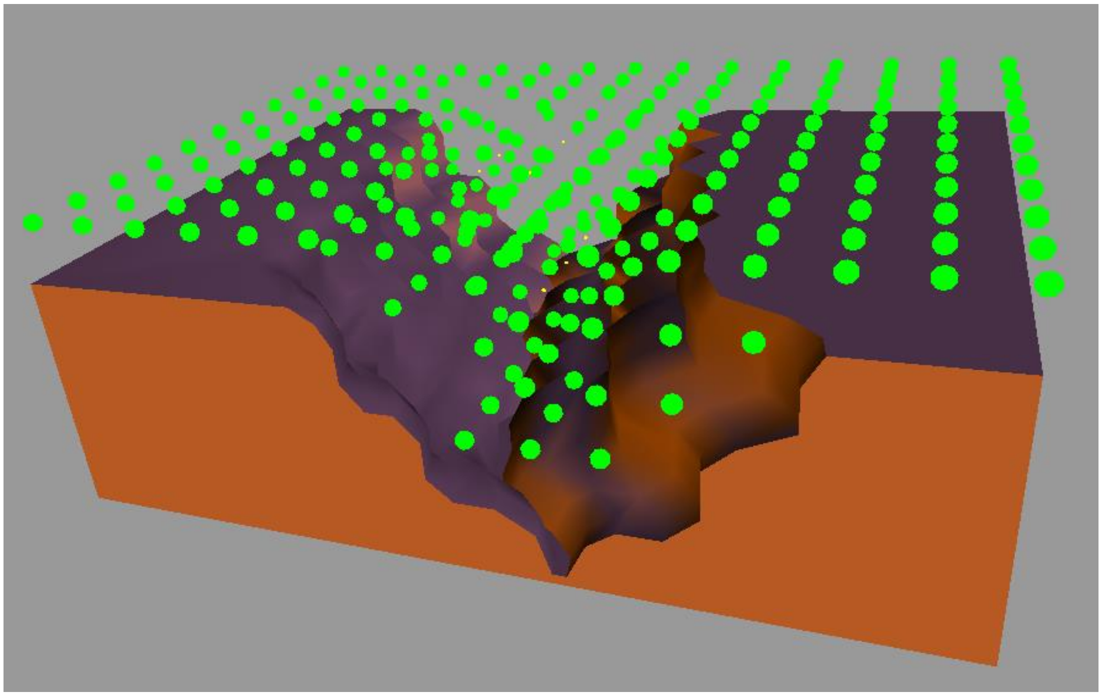


Figure 5.2: The result of our Chan-Vese algorithm as applied to the city scape model. The green vertices are in the segmentation mesh, yellow vertices are not in the mesh and red vertices solve the AGP.

Now that we see how to initiate and apply the Chan-Vese segmentation algorithm, Figure 5.3 depicts the results of applying the Chan-Vese segmentation algorithm to our Canyon and Arena models.



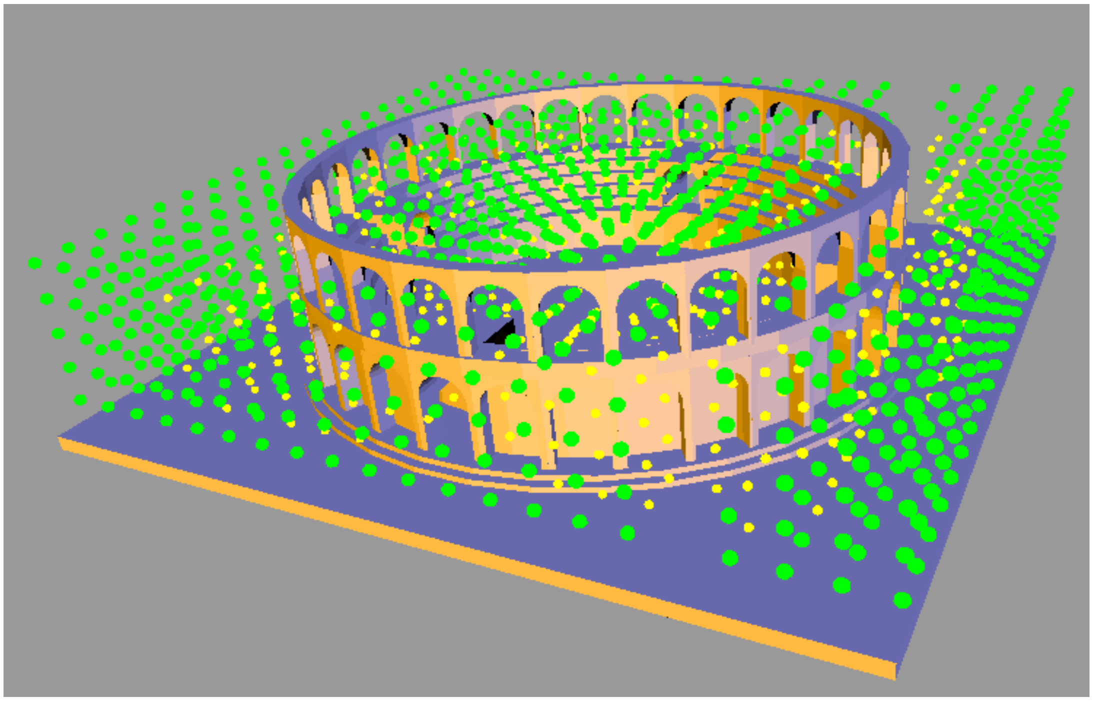


Figure 5.3: The result of our Chan-Vese algorithm as applied to the canyon and arena models. The green vertices are in the segmentation mesh, yellow vertices are not in the mesh.

Table 5.1: Coverage Obtained as the Chan-Vese Method Progresses

Number of Chan-Vese Segmentation Applications	Chan-Vese Mesh Photon Volume Coverage	Non-Chan-Vese Photon Volume Mesh Coverage
0	42817	467584
3	221429	249598
6	267712	203315

Table 5.1 above demonstrates the photon volume obtained at different steps of the application of our algorithm to the Cityscape model. It indicates coverage obtained by vertices included and not included in the Chan-Vese segmentation mesh.

The following tables show the performance of our algorithm as applied to the scenarios described above while using the Cityscape model. Table 5.2 shows the respective algorithms' ability to perceive photons and timing. Table 5.3 shows the respective segmentation meshes' "connectivity" which is the cardinality of the largest set of connected vertices provided by the `connected_components()` algorithm implemented in the boost graph library [57]. The connectivity number represents the number of candidate waypoints that may be traversed by a UxV in the respective segmentation meshes. Hence, a higher connectivity count is better as it indicates that a greater number of paths may be chosen by a UxV to realize multiple mission scenarios.

Table 5.2 indicates that while *k*-means is faster and produces a greater sensed photon volume per mesh vertex, the total number of photons available to be observed is similar to that provided by the Chan-Vese segmentation algorithm. (Note that an observation point within a mesh may have overlapping surveillance with a neighboring point.) The strength of the Chan-Vese algorithm is demonstrated in Table 5.3 as it produces a larger set of candidate way points that may be traversed by a UxV thus allowing for a greater number of possible mission applications while maintaining a high degree of surveillance capability as the UxV traverses these waypoints.

Table 5.2: Comparison Between Photon Mapping Informed Chan-Vese and *k*-means in Terms of Time and Photons per Vertex

Segmentation Method	Photon Volume per Vertex	Percentage of all available photons viewed	Timing (ms)	Scenario
Chan-Vese	1221.5	92.3	60	1
k-means	3770.3	91.4	1	
Chan-Vese	1170.7	91	50	2
k -means	3690.4	89	1	
Chan-Vese	1661.1	96	47	3
k -means	3655.1	94.5	4	
Chan-Vese	1658.1	95.3	45	4
k -means	3652.3	94.3	3	

Table 5.3: Comparison Between Photon Mapping Informed Chan-Vese and *k*-means in Terms of Connectivity.

Segmentation Method	Connectivity	Percentage Improvement of Chan-Vese over *k* -Means	Scenario
Chan-Vese	435	65	1
k-means	151		
Chan-Vese	433	65	2
k-means	151		
Chan-Vese	3090	40	3
k-means	1846		
Chan-Vese	3084	40	4
k-means	1844		

5.6 Summary and Conclusion

Understanding how to provide better surveillance in areas not viewable by visible light can arrive by modeling a virtual environment illuminated by photons in the non-visible spectrum and providing the UxVs populating these environments with the tools to maximize their sensing capabilities. In order to enhance UxV sensing ability as well as enable UxV route path-planning, we propose a 3D segmentation algorithm based upon the Chan-Vese method to create a connected 3D mesh within the UxVs' photon-mapping-illuminated virtual environment. The resulting segmentation mesh's vertices contain more photons to be sensed by a UxV traversing the mesh than could have been sensed if the UxV had traveled elsewhere. The connectedness of the segmentation mesh gives the UxV uninterrupted travel through these highly-illuminated areas and allows for a variety of mission-planning scenarios. The initialization problem inherent to the Chan-Vese segmentation algorithm is overcome in a novel way by using output from SPOQ to produce an initial segmentation curve comprised of vertices which are highly distinguished from their neighbors. The results of our segmentation algorithm enables a UxV to focus its attention on areas in the 3D environment that maximize the (non-)visible spectrum photons obtainable by their sensors or conversely explore areas have not been well-illuminated. The significance of this effort is that it enables the realization applications answering the questions asked in the Introduction that solve the WRP in multi-spectral 3D environments which may then be affected by information-attenuating events such as simulated fog, rain, and so forth.

We see that the *k*-means algorithm confines the UxV movement to a small subset of the UxV-position grid. However, the *k*-means algorithm can also serve as a method of performing pre-segmentation of a UxV-position grid so as to identify those portions of the UxV-position grid that possess greater and lesser amounts of multi-spectral photon volume. This pre-segmented mesh can then allow for multi-UxV path planning as the Chan-Vese segmentation algorithm can then be applied to the pre-segmented UxV-position sub-grids. This resulting set of candidate waypoints can then be assigned that sub-grid's UxV. The concept of pre-segmentation by *k*-means in order to inform Chan-Vese arrives from [58].

CHAPTER 6

A NEW APPROACH TO COMPUTING APPROXIMATE SOLUTIONS TO THE WATCHMAN ROUTE PROBLEM

6.1 Utilizing the Chan-Vese Segmentation Mesh to Solve the WRP

Once this segmentation mesh described in Chapter 5 is created, a conventional shortest-path-finding algorithm can then be applied to connect the vertices contained within the segmentation mesh to approximately solve the AGP. These connected vertices lead to the production of a shortest path which senses more photons than a path that does not utilize the vertices provided by the mesh. The WRP is then solved immediately in a heuristic manner since the WRP route will be designated by choosing the shortest route from elements comprising a minimum-energy mesh encompassing the SPOQ-provided vertices. Furthermore, since SPOQ does not require foreknowledge of the 3D environment it is placed within, we can then (randomly) discard or add vertices prior to providing approximately optimal solutions to the AGP (and subsequent approximately optimal WRP-solving routes) in order to simulate forbidden zones or moving obstacles within our virtual environment.

The name we give to our heuristic WRP-solving algorithm is the Photon-mapping-Informed active-Contour Route Designator or PICRD. The name derives from the fact that we are using a photon-mapping-informed PDE-based active contour segmentation algorithm to allow us to designate a shortest-path route through the high-visibility vertices comprising the 3D mesh produced within our 3D virtual environment.

The contribution this dissertation makes as a result of using our PICRD algorithm is to enable UxVs to concentrate their attention on areas in the 3D virtual environment that maximize the number of visibility-conveying multispectral photons obtainable by their sensors while minimizing the size of the route taken and also accounting for varying environmental circumstances. Since we are using photon mapping as our foundation for PICRD, we can take into account the behavior of photons as they propagate through the various environmental conditions that may be encountered by a UxV. Since we are using a segmentation algorithm's determination of what waypoints have approximately maximum visibility, we may then guide the creation of the approximately optimal WRP-solving route to provide maximum coverage above that provided by a conventional shortest path finding algorithm. Since we are using the online SPOQ algorithm, we may allow PICRD to alter the watchman's route after the passage of a unit of time in response to dynamic simulated environmental circumstances.

6.2 The PICRD Algorithm

Once our photon-mapping-informed Chan-Vese algorithm has been applied to G to produce the resulting segmentation mesh G_r, a route may then be chosen from the resultant visibility mesh by means of some graph search algorithm. The waypoints to be connected will include the SPOQ-produced UxV-observer points established by the SPOQ algorithm. The use of these approximately optimal AGP-solving waypoints ensures that the maximum number of uniquely-sensed photons will be observed. The shortest path found through the visibility mesh G_r will then also maximize the number of photons covered and thus solve the WRP.

Combining the steps considered previously, the complete PICRD algorithm is given below.

Algorithm 3 PICRD

```
1. Construct a 3D model of the virtual environment
2. Construct the UxV-observer grid $G$
3. Apply the photon mapping algorithm to the virtual environment
4. Obtain the photon volume at every grid point $g_i \in G$
5. Solve the AGP for grid $G$ using the SPOQ algorithm
6. Apply the 3D photon-mapping-informed Chan-Vese segmentation algorithm
7. Connect the SPOQ-provided UxV-observer points established in Step 5 using shortest path-
   finding algorithm while using the segmentation-mesh vertices produced by Step 6 as
   waypoints.
8. Alter the environment according to some criteria.
9. Repeat steps 2 – 8 for time steps $t_0 \ldots t_n$.
```

6.2.1 Algorithm Analysis

The construction of the UxV-observer grid in step 2 has a O(n) complexity where n is the number of grid points in the virtual environment. The application of the photon mapping algorithm in step 3 has a O(p) complexity where p is the total number of photons launched in the environment [9]. Obtaining the photon volume at every grid point prescribed in step 4, depending upon the method used as described in Section III B, can have either a O($n \cdot \log p$) or O($n \cdot q \cdot \log p$) computational complexity where q is the number of photons returned by a photon query and $q < p$. The application of SPOQ wherein the photon volume is calculated for every grid point prior to its operation has a time complexity of O($l \cdot n$) where l is the number of waypoints that obtain maximum visibility and $l << n$. The iterative Chan-Vese algorithm has an

O(n) complexity [55]. The application of Dijkstra's algorithm [52] has a time complexity of O($|E| + |V|\log|V|$) [59] where E and V are the number of edges and vertices in the segmentation mesh G_r, respectively.

6.3 Problem Statement

6.3.1 Initial Assumptions

When performing our experiments listed below, we make the following assumptions:

1. The photons are confined to the visible light spectrum.
2. There is no participating medium for the photons to interact with.
3. The shortest-path-finding algorithm used to connect our SPOQ-provided points is a greedy version of Dijkstra's algorithm [52].
4. The sensors used are omnidirectional with a range of one third the length of the model on the z-axis.
5. Only 4 observers are used for SPOQ.

6.3.2 Scenarios Considered

We will apply PICRD to the following scenarios.

1. Using 1188 observer grid points and 2500, 5000 and 7500 photons cast on the Cityscape model; using 15000, 20000 and 25000 photons cast on Canyon and Arena models with 990 and 2520 grid points, respectively.
2. Using the same respective number of photons cast with a 6897-points observer grid for the Cityscape model; using a 1152 and 4550-point observer grid for the Canyon and Arena models, respectively.
3. Repeating the first scenario but using randomly-generated forbidden zones constituting 10%, 15% and 20% of the grid points available, and

4. Repeating the first scenario but using a set of forbidden zones representing a moving obstacle constituting 10% of the grid points available moving gradually over 3 time steps.

The first scenario may be regarded as a means of testing PICRD with gradually increasing “brightness” in terms of photons launched, the second scenario represents the scalability of the PICRD algorithm when applied to larger grids and the last two scenarios represent an environment with increasing difficulty in maneuvering. In each scenario, our goal is to show that we can sense a greater number of photons using the PICRD algorithm than would be possible if we were to use a naïve Dijkstra’s algorithm with no additional information.

As we have considered in Chapter 5, in each scenario the PICRD algorithm’s Chan-Vese segmentation algorithm was substituted with a *k*-means classifier arriving from the ITK toolkit [56]. The input to the *k*-means classifier consists of an array called G_a of size $|G|$ whose *i*th element’s value is g_i’s photon volume where $g_i \in G$. The initial mean (l) for the cluster of “light” grid points arrives from calculating the average photon volume of those grid points used to initialize the Chan-Vese algorithm. All grid points in G_a whose photon volume is greater than or equal to l are placed in the “light” cluster whereas all other grid points are placed in the “dark” cluster. The resulting segmentation mesh provides a set of bright grid elements where the most photons may be sensed and may serve as candidate waypoints for our WRP path-planner.

6.4 Results

Figures 6.1 and 6.2 depict scenarios that start where we ended in Chapter 5, namely we depict the results of comparing PICRD with Dijkstra’s algorithm. The PICRD route is designated in red whereas the Dijkstra route is designated in blue. Green vertices designate PICRD-usable waypoints whereas yellow vertices are not.

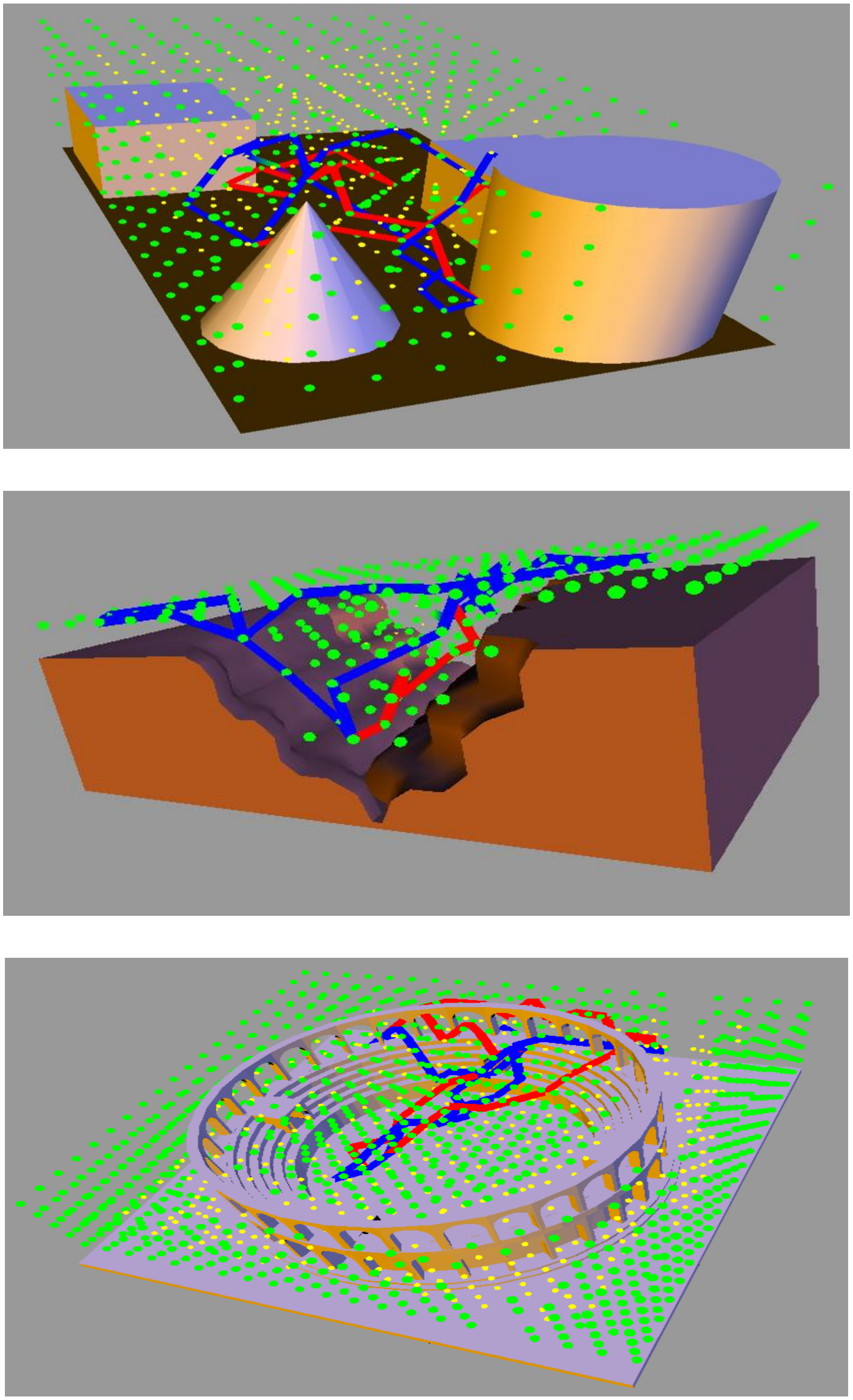


Figure 6.1: The PICRD algorithm (red route) compared to the naïve Dijkstra algorithm (blue route) in Scenario 1.

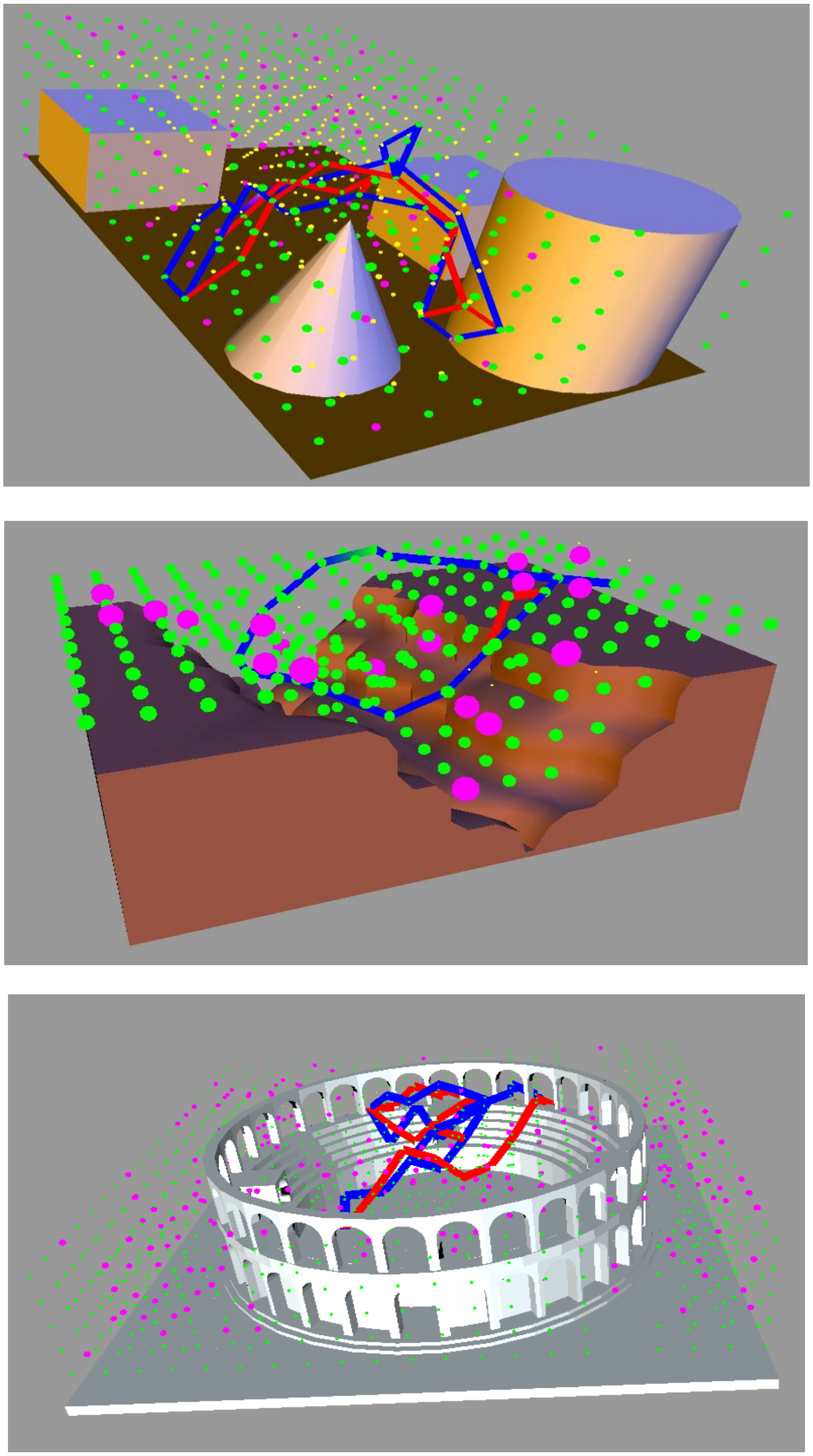


Figure 6.2: The PICRD algorithm in comparison to the naïve Dijkstra algorithm in Scenario 3 with 10% of the grid points removed. Note that the forbidden zones are designated in pink.

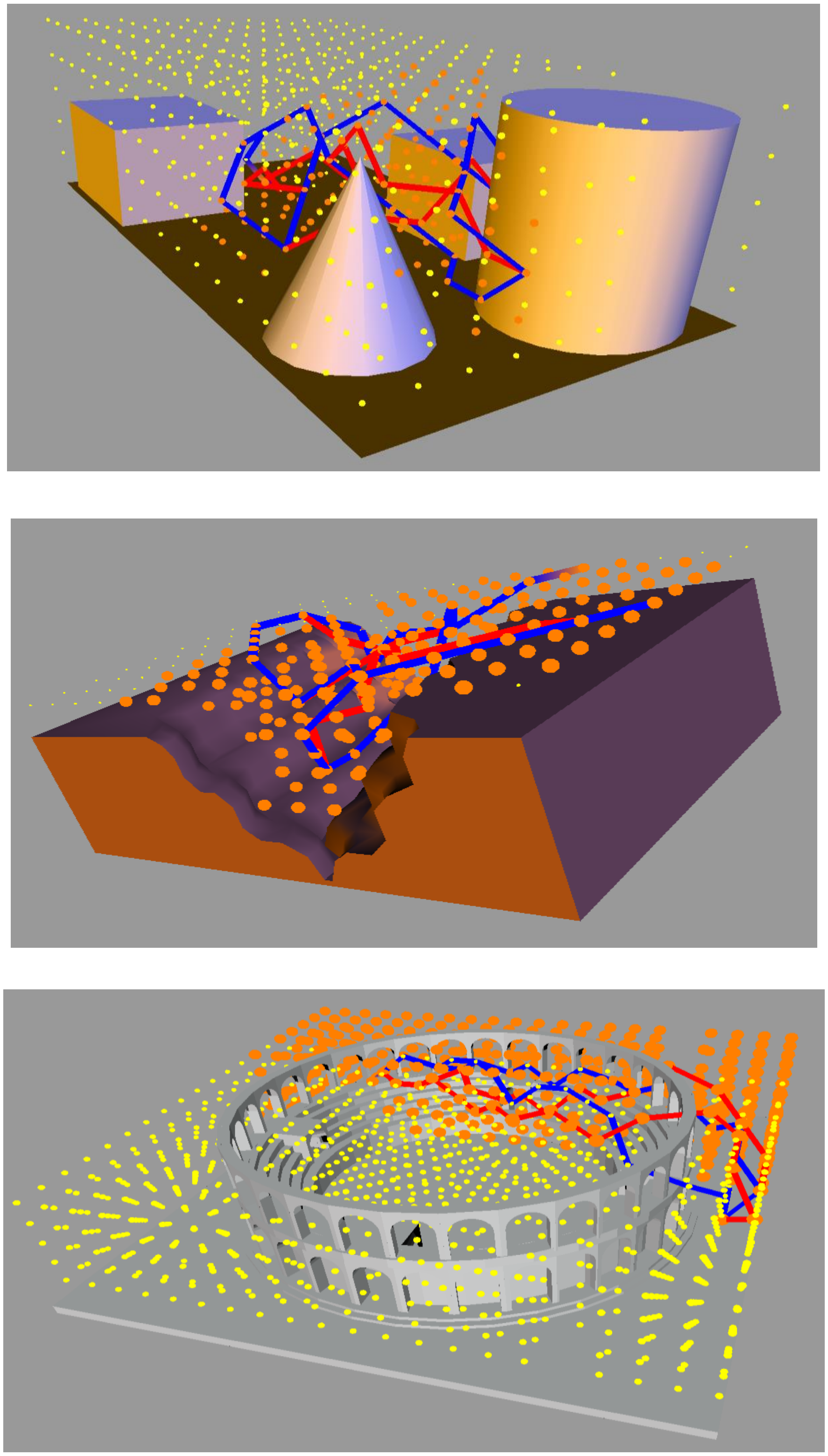


Figure 6.3: The *k*-means-moderated PICRD algorithm (red route) compared to the naïve Dijkstra algorithm (blue route) operating in circumstances described by Scenario 1.

Figure 6.3 above depicts Scenario 1 showing the result of path planning using the *k*-means algorithm rather than the Chan-Vese algorithm. Note that this mesh – depicted using orange vertices - is much smaller and closely confined than the Chan-Vese segmentation mesh. (As we shall see, the smaller mesh size diminishes the ability to support route-adaptability in changing environmental circumstances.)

The tables below show the results of applying PICRD to scenarios 1, 2, 3 and 4. Note that the column indicating a percentage of uniquely sensed photons represents the coverage obtained as a percentage of the total number of photons available to be sensed. The subsequent column measures coverage in terms of the photon volume sensed as the shortest route is traversed. The path distance is the length of the edges connecting the vertices together. Recall that we assign a distance of one to each edge in the segmentation mesh. Since we are finding approximately optimal solutions to the WRP, short path distance routes with a high photon volume are desired.

Table 6.1: WRP Performance under Circumstances Described by Scenario 1

Path Finding Algorithm	% of Uniquely-Sensed Photons	Photon Volume Sensed	Path Distance	Time Required (ms)
Cityscape Model				
Photons Launched: 2500				
PICRD	93.8	19832	29	61.7
Dijkstra	93.9	15311	29	31.3
k-means	92.7	18532	28	32.3
Photons Launched: 5000				
PICRD	91.2	45853	20	60.4
Dijkstra	90.6	38586	20	30.7
k-means	90.7	46323	20	31.7
Photons Launched: 7500				
PICRD	90.4	64640	26	60.2
Dijkstra	90.2	52454	26	29.5
k-means	90.3	60292	26	30.5
Canyon Model				
Photons Launched: 15000				
PICRD	93.9	278004	32	48.7
Dijkstra	93.8	264678	32	28.4
k-means	93.6	277524	32	29.5
Photons Launched: 20000				
PICRD	93.7	412732	33	52.1
Dijkstra	93.6	398002	33	29.1
k-means	94.0	427321	33	30.2
Photons Launched: 25000				
PICRD	93.5	519675	32	48.8
Dijkstra	93.1	516858	32	27.7
k-means	93.2	517625	32	29.4
Arena Model				
Photons Launched: 15000				
PICRD	89.4	166920	31	85.6
Dijkstra	90.1	151482	30	43.1
k-means	89.7	164723	30	44.5
Photons Launched: 20000				
PICRD	90.4	190186	34	88.9
Dijkstra	90.1	183697	32	45.6
k-means	90.0	191293	32	46.1
Photons Launched: 25000				
PICRD	88.1	247220	32	87.1
Dijkstra	89.3	243347	29	40.1
k-means	89.5	247832	29	42.3

Table 6.2: WRP Performance under Circumstances Described by Scenario 2

Path Finding Algorithm	% of Uniquely-Sensed Photons	Photon Volume Sensed	Path Distance	Time Required (ms)
Cityscape Model				
Photons Launched: 2500				
PICRD	93.4	29478	42	577.4
Dijkstra	93.2	25186	42	346.3
k-means	93.2	27325	38	350.4
Photons Launched: 5000				
PICRD	94.6	118567	61	561.6
Dijkstra	94.7	104510	60	341.9
k-means	94.5	115872	55	337.3
Photons Launched: 7500				
PICRD	93.3	156021	57	574.9
Dijkstra	91.3	142719	57	342.2
k-means	92.8	152826	50	338.5
Canyon Model				
Photons Launched: 15000				
PICRD	92.7	254946	34	58.7
Dijkstra	93.0	252437	34	38.9
k-means	93.4	255214	33	41.6
Photons Launched: 20000				
PICRD	93.2	445592	34	62.1
Dijkstra	92.8	430497	34	39.1
k-means	93.0	454612	33	42.2
Photons Launched: 25000				
PICRD	92.1	555002	36	48.8
Dijkstra	92.4	552050	36	41.5
k-means	91.9	553781	35	44.4
Arena Model				
Photons Launched: 15000				
PICRD	90.9	177581	43	785.6
Dijkstra	91.1	144848	42	441.6
k-means	90.2	167249	42	445.9
Photons Launched: 20000				
PICRD	91.3	418070	47	769.8
Dijkstra	90.1	355320	45	447.0
k-means	91.5	387256	45	454.3
Photons Launched: 25000				
PICRD	90.1	361571	47	787.1
Dijkstra	90.3	301959	46	444.1
k-means	90.4	345715	46	439.3

Table 6.3: WRP Performance under Circumstances Described by Scenario 3

Path Finding Algorithm	% of Uniquely-Sensed Photons	Photon Volume Sensed	Path Distance	Time Required (ms)
Cityscape Model				
% of Observer Grid Forbidden: 10				
PICRD	93.5	57784	25	60.7
Dijkstra	94.1	48695	25	29.5
k-means	N/A	N/A	N/A	N/A
% of Observer Grid Forbidden: 15				
PICRD	93.1	54446	17	61.2
Dijkstra	93.7	45113	16	32.1
k-means	N/A	N/A	N/A	N/A
% of Observer Grid Forbidden: 20				
PICRD	92.6	53278	18	61.1
Dijkstra	91.2	44788	18	30.4
k-means	N/A	N/A	N/A	N/A
Canyon Model				
% of Observer Grid Forbidden: 10				
PICRD	92.9	526404	34	46.1
Dijkstra	93.1	515378	33	27.2
k-means	92.6	526513	32	29.2
% of Observer Grid Forbidden: 15				
PICRD	92.6	512262	35	56.1
Dijkstra	92.4	498452	35	30.3
k-means	93.2	526211	35	32.5
% of Observer Grid Forbidden: 20				
PICRD	93.2	517675	32	54.8
Dijkstra	93.1	514858	32	32.5
k-means	92.5	516513	32	35.1
Arena Model				
% of Observer Grid Forbidden: 10				
PICRD	89.4	245654	32	87.6
Dijkstra	90.1	236495	31	45.3
k-means	N/A	N/A	N/A	N/A
% of Observer Grid Forbidden: 15				
PICRD	90.4	242192	33	89.2
Dijkstra	90.1	238622	33	46.6
k-means	N/A	N/A	N/A	N/A
% of Observer Grid Forbidden: 20				
PICRD	88.1	232361	31	88.1
Dijkstra	89.3	231252	30	42.1
k-means	N/A	N/A	N/A	N/A

Table 6.4: WRP Performance under Circumstances Described by Scenario 4

Path Finding Algorithm	% of Uniquely-Sensed Photons	Photon Volume Sensed	Path Distance	Time Required (ms)
Cityscape Model				
Time Step 0				
PICRD	92.3	55438	24	60.8
Dijkstra	93.4	49251	24	29.7
k-means	N/A	N/A	N/A	N/A
Time Step 1				
PICRD	93.3	56892	26	62.2
Dijkstra	92.2	48598	26	30.9
k-means	N/A	N/A	N/A	N/A
Time Step 2				
PICRD	94.2	56287	24	59.9
Dijkstra	93.7	47912	24	32.0
k-means	N/A	N/A	N/A	N/A
Canyon Model				
Time Step 0				
PICRD	91.7	527378	34	45.1
Dijkstra	92.1	516587	33	23.6
k-means	91.5	527724	32	26.4
Time Step 1				
PICRD	92.6	526532	34	45.3
Dijkstra	92.7	511384	33	28.2
k-means	92.1	525527	32	30.2
Time Step 2				
PICRD	92.9	526485	33	47.1
Dijkstra	93.1	518178	33	28.2
k-means	92.6	526149	32	31.2
Arena Model				
Time Step 0				
PICRD	90.4	245654	32	88.4
Dijkstra	90.1	234495	31	46.1
k-means	N/A	N/A	N/A	N/A
Time Step 1				
PICRD	89.6	245654	32	87.0
Dijkstra	89.4	234495	31	44.2
k-means	N/A	N/A	N/A	N/A
Time Step 2				
PICRD	90.3	245654	31	86.8
Dijkstra	90.2	234495	31	46.3
k-means	N/A	N/A	N/A	N/A

To explain this difference in performance between the *k*-means-segmented PICRD and Chan-Vese-segmented PICRD consider Table 6.5 which shows the Cityscape model's respective segmentation meshes' "connectivity". This connectivity value is the cardinality of the largest set of connected vertices provided by the `connected_components()` algorithm implemented in the boost graph library [57]. The connectivity number represents the number of candidate waypoints that may be traversed by a UxV in the respective segmentation meshes. Hence, a higher connectivity count is better as it indicates that a greater number of paths may be chosen by a UxV in order to adapt to different scenarios. The strength of the Chan-Vese algorithm - demonstrated in Table 6.5 - arrives from the fact that it produces a larger set of high-photon-volume candidate way points that may be traversed by a UxV. This allows for a greater number of possible mission applications while maintaining a high degree of surveillance capability as the UxV traverses these waypoints. Similar results can apply to the Canyon and Arena models' segmentation meshes.

Table 6.5: Comparison Between Photon Mapping Informed Chan-Vese and *k*-means in Terms of Connectivity for Scenarios 3 and 4.

Segmentation Method	Connectivity	Percentage of Vertices Removed from Consideration	Scenario
Chan-Vese	392	10	3
k-means	136		
Chan-Vese	370	15	3
k-means	128		
Chan-Vese	348	20	3
k-means	121		
Chan-Vese	392	10	4
k-means	136		

6.5 Summary and Conclusion

We have presented an algorithm providing a heuristic solution to the WRP within a 3D virtual environment populated by simulated unmanned vehicles UxVs. The contributions made by our algorithm are three-fold. First, we utilized photon mapping as our means of representing the information sensed by a UV. Second, we use the photon map to generate an online approximate solution to the closely-related NP-hard art gallery problem (AGP). Third, we use a 3D Chan-Vese segmentation algorithm initialized by our SPOQ to produce a candidate set of path-planning waypoints. The use of photon mapping by our online AGP solver allows us to simulate and adapt UV operation to accommodate variable, less-than-ideal environmental circumstances. The use of our 3D Chan-Vese segmentation algorithm creates a set of candidate waypoints that yield greater visibility coverage when computing approximately optimal solutions to the WRP than would be obtainable otherwise. Our efforts yield a set of shortest routes that provide greater sensed information than shortest routes chosen using a conventional approximately optimal WRP-solving technique while retaining route-adaptability under variable environmental conditions.

The possible objections leveled against SPOQ can be equally leveled against PICRD. It could also be argued that performing the Chan-Vese segmentation algorithm as a means of informing the development of the watchman's route is unnecessary since a shortest-path-finding algorithm connecting the SPOQ-provided waypoints ensures that the maximum number of uniquely-sensed photons will be detected by the watchman's route. Note however that due to the non-convex nature of the WRP, this route does not ensure maximum visibility coverage as it is being travelled. Performing the Chan-Vese segmentation algorithm gives us a set of candidate waypoints for our shortest-path-finding algorithm with a higher photon volume. The selection of candidate waypoints from within G_r by our shortest-path-finding algorithm assures us that as we traverse between SPOQ-provided waypoints, we will cover a greater total photon volume and thus ensure better coverage along the watchman’s route.

CHAPTER 7

FUTURE RESEARCH

7.1 Future Research on Computing Approximate Solutions to the Art Gallery Problem

This work represents the first step in the development of a means of obtaining optimal sensor coverage in multiple types of 3D environments. There are three areas of development that we want to pursue, namely

1. the accommodation of multiple heterogeneous sensing ranges,
2. the accommodation of sensors which use sound rather than EM spectra, and
3. the accommodation of real time updates to the photon map in order to represent a changing environment.

The first effort may be accomplished by the use of an area of study in mathematics known as *sphere packing* [60]. This problem seeks to find an arrangement of spheres - that may have unequal radii - such that the spheres fill as much of the volume as possible. In the context of SPOQ, this sphere arrangement represents an observer grid containing sensors with heterogeneous sensing ranges. The application of SPOQ would proceed in the manner mentioned above except for the fact that it would be highly unlikely that a particular sensor occupying a given observation point would have a sensing range equal to that of its neighbor.

The second effort has been considered before by the introduction of *phonon mapping* [61]. The photon mapping algorithm's construction of the photon map need not be confined strictly to the EM spectrum. It has been shown that by extending the concept of the photon to be any discrete packet of information-bearing energy propagating through a medium, the behavior of sound traversing through air can also be modeled. This packet of sound energy is known as a *phonon* rather than a photon. While the phonon's operational characteristics in air have been considered previously, the transmission of a phonon through water has not. We expect that it would entail the use of the equivalent of a volumetric photon map except that it would be applied to water rather than air. Providing coverage for maritime environments using phonons is an "over the horizon" area of research that has not been fully developed.

The third effort may be accomplished by means of performing parallel computation on a photon map in order to update it in real time. This topic has been subject to consideration previously [62, 63]. Essentially, what these efforts are aiming to achieve is the rapid updating to a *k*-d tree in response to the movement of objects within the virtual environment. Each photon can operate independently of every other photon and can be assigned its own thread on an *n*-core machine. Once the *k*-d tree is updated on some core in an *n*-core machine, SPOQ can be used on that particular core. Given an *n*-core processor, SPOQ could provide the optimal sensor placement of a dynamic scene operating *n* time steps into the future. It should be noted that while parallelizing the photon map algorithm is well established, parallelizing the phonon mapping algorithm has not been done.

7.2 Future Research Regarding the Photon Mapping Informed Multispectral Chan Vese Segmentation Algorithm

The *k*-means algorithm confines the UxV movement to a small subset of the UxV-position grid. However, the *k*-means algorithm can also serve as a method of performing pre-segmentation of a UxV-position grid so as to identify those portions of the UxV-position grid that possess greater and lesser amounts of multi-spectral photon volume. This pre-segmented mesh can then allow for multi-UxV path planning as the Chan-Vese segmentation algorithm can then be applied to the pre-segmented UxV-position sub-grids. This resulting set of candidate waypoints can then be assigned that sub-grid's UxV. The concept of pre-segmentation by *k*-means in order to inform Chan-Vese arrives from [58].

Because we have utilized the Chan-Vese segmentation method as the foundation of PICRD, we may now take advantage of the research that has been done to improve this segmentation algorithm. There are two possible avenues of improvement that we may take: the SPOQ algorithm or the Chan-Vese segmentation method. Improvements to the SPOQ algorithm have been covered in section 5.2. There are three possible improvements we can make to the Chan-Vese algorithm: parallelizing its operation and creating a convex search space.

As we discussed above in Section 7.2, SPOQ lends itself to parallelization. The parallelization of the Chan-Vese component of PICRD has been considered before in [23]. Their main idea is to partition the 2D image or 3D volume into non-overlapping regions and applying the Chan-Vese segmentation algorithm independently to the sub-regions. When a segmentation curve arrives upon the boundaries of the sub-region, the curve is exchanged with its neighboring curve.

As we have mentioned before, the Mumford-Shah energy functional is non-convex. However, in [64], Brown et al. introduced a means of modifying the Chan-Vese algorithm such that it avoids local minima. Recall that constants c_1 and c_2 were set to fixed values prescribed by Chan and Vese in [11]. The values of c_1 and c_2 describe the average pixel intensity inside and outside the evolving contour, respectively. It was shown in [65] by Chan et al that global optimality of the segmentation result can be achieved by fixing the values of c_1 and c_2. In [64], Brown et al. showed that by obeying certain conditions, the values of c_1 and c_2 can be treated as functions v_1 and v_2 and can then be amenable to calculation in order to find global minimizers. They demonstrated that their technique approximates the true global minimizers to within 0.01–1.8%. As a consequence of their efforts, we feel that modifying our photon-mapping informed version of the Chan-Vese segmentation algorithm in order to incorporate these new results can make the calculation of visibility even more accurate.

7.3 Future Research on Computing Approximate Solutions to the Watchman Route Problem

There has been a research push towards removing the human operator from a vehicle completely and introducing vehicular autonomy. Three such pushes have been the Defense Advanced Research Projects Agency's (DARPA) Grand Challenge [66], Urban Challenge [67] and the Robotics Challenge [68]. The first two Challenges were dedicated to addressing how an autonomous wheeled vehicle could complete a course in a desert or urban environment, respectively. The last such Challenge was dedicated to addressing how an autonomous legged-robot could complete a course consisting of tasks dedicated to rescuing humans in an emergency scenario.

With regard to the Grand and Urban Challenges, a single autonomous robot was responsible for navigating through the terrain. As we can see, the WRP emanates from the AGP which means that sensor networks can collaborate with UxVs in order to assist in surveillance and completion of a particular task. Furthermore, as we suggested in Section 7.2, we can partition an area to be observed into subsections using *k*-means that allows us dedicate resources to areas that have greater and lesser illumination.

While the future research efforts related to the development of PICRD mirror that discussed in Sections 7.1 and 7.2, a further research effort would entail adding UxV dynamics to the model such that the path traced by the simulated UxV is not merely traversing to a set of points, but is in fact following points in accordance to the capacities of the vehicle being simulated. Such capacities include fuel efficiency, aerodynamics, fluid mechanics, friction, and etcetera. The important thing to note is that a framework now exists for computing approximate solutions to the WRP. The inclusion of more sophisticated vehicular dynamics models would only serve to improve the usefulness of PICRD.

7.4 Future Research Regarding Machine Learning

Regarding human intelligence, the famed physicist Michio Kaku said “We run simulations into the future. Animals apparently don’t do this. They don’t plan to hibernate. They don’t plan the next day’s agenda. They have no conception of tomorrow to the best of our ability. But that’s what our brain does. Our brain is a prediction machine.” [69]. Prediction of the outcome of a given set of circumstances is the most fundamental component of intelligence [70]. The ability to predict outcomes can be narrowed in scope to the field of scene understanding. Having the ability to understand a scene implies the ability to predict a scene’s behavior. Such understanding leads to better and more efficient surveillance due to greater adaptability in executing a mission.

What we propose for future research is the utilization of recent developments in 3D scene understanding in combination with the SPOQ and PICRD algorithms presented here to provide a means of providing optimal surveillance of entities residing in a changing 3D scene. We ask the question: “Given the video capture of a scene, how do we direct static sensors or UxVs to best surveil an item of interest (IOI) based upon their predicted behavior?” In order to answer this question, we must address the following challenges, namely

1. How do we create a 3D representation of that IOI?
2. How do we predict the behavior of the 3D IOI within the scene?

3. How do we integrate the knowledge gained from answering questions 1 and 2 such that we can solve the AGP/WRP?

In order to address challenge posed by question 1, we look to the results produced by Zia, Stark and Schindler in their paper “Are Cars Just 3D Boxes? – Jointly Estimating the 3D Shape of Multiple Objects” [71]. In this paper, the authors matched a 3D model class (an SUV in their particular experimental case) to a 3D bounding box generated within a 2D image. In order to address the challenge posed by question 2, we look to the results generated by Walker, Gupta and Herbert in their paper “Patch to the Future: Unsupervised Visual Prediction” [72]. In this paper, the authors utilized a reward function on textures that are to be traversed by the IOI within the video frame and subsequently generated a set of high probability routes to be followed by the IOI.

In [71], the authors present an algorithm that requires a binary occlusion determination for each vertex in the model as well as a 3D ground plane for 3D model placement. Note that the occlusion data required by their algorithm may serve as input to a *k*-d tree. Furthermore, once the scene’s 3D object class for the IOI is isolated and a ground-plane is known, an observation-grid may then be constructed. With the construction of the *k*-d tree and observation grid, the minimum requirements for the operation of SPOQ and PICRD are fulfilled.

Once we have met the minimum requirements to use SPOQ/PICRD for a single frame of video, we want to then be able to predict future behavior and thus answer question 3. Such AGP/WRP-amenable prediction may be accomplished by using the algorithm in [72]. This paper’s algorithm operates over a sequence of video frames (which may arrive from frames used in [71]) in order to predict the IOI’s future behavior. Given such a prediction, we may then infer the best sensor placement using SPOQ at a frame generated at time t_0. We may then generate a

PICRD-provided route that links frame t_0 together with similarly SPOQ-moderated frames at times $t_1 \ldots t_n$. Due to the fast queries provided by the *k*-d tree used by SPOQ, SPOQ-provided results may be computed in parallel for video frames captured at times $t_0 \ldots t_n$.

Using the algorithm provided in [71] has some limitations. The limitations considered in [71] are the fact that they limited themselves to only one model class and they must be sure that their bounding-box generator can reliably isolate the IOI. These limitations can be addressed by using a greater number of models and by a more robust classifier (such as that provided by [73]).

A limitation of [72] (but not considered in their paper) is the fact that weather events were not included in the determination of what routes are traversable. Such weather events would of course limit the number of routes that may be taken by the IOI. However once these events are considered, and their effect on occlusion is determined, SPOQ has the ability to model their effect upon computing approximate solutions to the AGP.

7.5 Summary and Conclusion

There are many avenues of improvement available. The work presented in this dissertation can be regarded as an initial step towards creating a more robust and versatile AGP and WRP solver. We look forward to enhancing our algorithms' capabilities in the manner described above.

7.6 Publications

1. B. Johnson, H. Qi and J. Isaacs "Optimizing Coverage of Three-Dimensional Wireless Sensor Networks by Means of Photon Mapping" *Proceedings of the Winter Simulation Conference* (*WSC* '13), Washington, DC Dec 8-11.
2. B. Johnson, H. Qi, J. Isaacs, A photon-mapping informed Chan-Vese segmentation algorithm to enable multispectral sensing and path-planning in 3D virtual environments, *Proceedings of the Computer Vision and Pattern Recognition* (*CVPR*) *Workshop on Perception Beyond the Visible Spectrum* (*PBVS* '14), Columbus, OH, Jun. 23, 2014.
3. B. Johnson, H. Qi, J. Isaacs, A Comparative Study of Methods to Solve the Watchman Route Problem in a Photon Mapping-Illuminated 3D Virtual Environment, *Proceedings of the Applied Imagery and Pattern Recognition Workshop* (*AIPR* '14)), Washington, D. C. Oct. 14-16, 2014.
4. B. Johnson, H. Qi, J. Isaacs, Computing a heuristic solution to the watchman route problem by means of photon mapping within a 3D virtual environment testbed. *Proceedings of the Knowledge, Skill, and Behavior Transfer in Autonomous Robots Symposium* (*KSBT '14*), Arlington, VA, Nov. 13-15, 2014.
5. B. Johnson, H. Qi, J. Isaacs, Optimizing coverage of three-dimensional wireless sensor networks by means of photon mapping, *IEEE Sensors Journal*
6. B. Johnson, H. Qi, J. Isaacs, Computing a heuristic solution to the watchman route problem by means of photon mapping, *IEEE Transactions on Robotics and Automation*

REFERENCES

[1] P. Aveek, Z. Sun, F. Mokaya, and P. Zhang, SensorFly: Controlled-mobile sensing platform for indoor emergency response applications. *Proceedings of the 10th International Conference on Information Processing in Sensor Networks*, (*ISPN* '11), Chicago, IL, April 12-14, 2011.

[2] A. Khan, A., L. Jenkins, Undersea wireless sensor network for ocean pollution prevention. *Proceedings of Communication Systems Software and Middleware and Workshops*, (*COMSWARE* 2008), Bangalore, India, Jan. 6-10, 2008.

[3] J. Lloret, M. Garcia, D. Bri, S. Sendra, A wireless sensor network deployment for rural and forest fire detection and verification, *Sensors*, 9(11): 8722 – 8747, 2009.

[4] C-F. Huang and Y-C. Tseng, The coverage problem in a wireless sensor network, *Mobile Networks and Applications*, 10(4): 519 – 528, 2005.

[5] Y-H. Tsai, L-T. Cheng, S. Osher, P. Burchard, and G. Sapiro, Visibility and its dynamics in a PDE based implicit framework, *Journal of Computational Physics*. 199(1): 260 – 290, 2004.

[6] R. Goroshin, Q. Huynh, and H. M. Zhou, Approximate solutions to several visibility optimization problems, *Communications in Mathematical Sciences* 9(2): 535 – 550, 2011.

[7] L-T. Cheng and Y-H. Tsai, Visibility optimization using variational approaches, *Communications in Mathematical Sciences* 3(3): 425 – 451, 2005.

[8] D. L. Schacter, *Psychology, 2nd Edition*, Worth Publishers, New York, NY, 2011.

[9] H. W. Jensen, *Realistic Image Synthesis Using Photon Mapping*, A. K. Peters, New York, NY, 2001.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms 3rd Edition*. MIT Press, Cambridge, MA, 2009.

[11] T. Chan and L. Vese, Active contours without edges. *IEEE Transactions on Image Processing* 10(2): 266 – 277, 2001.

[12] D. Mumford and J. Shah, Optimal approximation by piecewise smooth functions and associated variational problems, *Communications in Pure and Applied Math.* 42(5): 577 – 685, 1989.

[13] G. Xing, R. Rui, B. Liu, J. Wang, X. Jia, C-W. Yi. Data fusion improves the coverage of wireless sensor networks. *Proceedings of the 15th International Conference on Mobile Computing and Networking* (*Mobicom* ’09), Beijing China, Sept 20-25, 2009.

[14] T. Yan, T. He, and J. A. Stankovic. Differentiated surveillance for sensor networks. *Proceedings of* 1*st International Conference on Embedded Networked Sensor Systems* (*SenSys* ’03). Los Angeles, CA, Nov 5 - 7, 2003.

[15] N. Ahmed, S. S. Kanhere and S. Jha. Probabilistic coverage in wireless sensor networks. *Proceedings of Conference on Local Computer Networks*. Sydney, NSW, Nov. 17, 2005.

[16] S. Ren, Q. Li, H. Wang, X. Chen and X. Zhang, Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks, *IEEE Transactions of Parallel Distributed Systems*, 18(3): 334-350, 2007.

[17] W. Wang, V. Srinivasan, C. Kee-Chaing, Coverage in hybrid mobile sensor networks, *IEEE Transactions on Mobile Computing*, 7(11): 1374 – 1387, 2008.

[18] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, 1(1): 36-72, 2005.

[19] S. M. Nazrul Alam, and Z. J. Haas, Coverage and connectivity in three-dimensional networks. *Proceedings of 12th International Conference on Mobile Computing and Networking* (*MobiCom* ‘06), Los Angeles, CA, Sept. 24-29, 2006.

[20] S. M. Nazrul Alam and Z. J. Haas, Coverage and connectivity in three-dimensional underwater sensor networks, *Wireless Communication and Mobile Computing Journal*, 8(8): 995 - 1009, 2008.

[21] M. Kass, A. Witkin and D. Terzopoulos. Snakes: active contour models, *International Journal of Computer Vision*, 1(4): 321 - 331, 1988.

[22] J. A. Sethian, *Tracking Interfaces with Level Sets*, American Scientist, May-June, 1997.

[23] O. Rousseau and Y. Bourgault, An iterative active contour algorithm applied to heart segmentation, *Proceedings of the SIAM Conference on Imaging Science.* San Diego, CA, 7-9 July, 2008.

[24] D. Jianyuan and H. Chongyang, 3D fast level set image segmentation based on Chan-Vese model, *3rd International Conference on Bioinformatics and Biomedical Engineering*. Beijing, China, June 11-13 2009.

[25] J. Zhang, K. Chen and B. Yu, A 3D multi-grid algorithm for the Chan-Vese model of variational image segmentation, *International Journal of Computer Mathematics* 89(2): 160-189, 2012.

[26] R. R. Murphy, *Introduction to AI Robotics.* MIT Press, Cambridge, MA, 2000.

[27] T. Lolla, M. P. Ueckermann, K. Yigit, P. J. Haley, and P. F. J. Lermusiaux, Path planning in time dependent flow fields using level set methods, *IEEE International Conference on Robotics and Automation*, Saint Paul, Minnesota, May 14-18, 2012.

[28] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics.* 1*st Edition*, Cambridge University Press, Cambridge UK, 2000.

[29] P. K. C. Wang, Optimal path planning based on visibility, *Journal of Optimization Theory Applications*, 117(1): 157–181, 2003.

[30] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, UK, 1999.

[31] R. Kimmel and J. A. Sethian, Fast marching methods for robotic navigation with constraints, Technical Report, Univ. of California, Berkeley, May, 1996.

[32] B. Englot and F. Hover, Sampling-based coverage path planning for inspection of complex structures, *Proceedings of the International Conference on Automated Planning and Scheduling,* (*ICAPS* 2012), Sao Paulo, Brazil, June 25-29, 2012.

[33] G. Papadopoulos, H. Kurniawati, and N.M. Patrikalakis. Asymptotically optimal inspection planning using systems with differential constraints, *Proceedings of the International Conference on Robotics and Automation*, (*ICRA* 2013), Karlsruhe, Germany, May 6-10, 2013.

[34] P. Janousek, and J. Faigl. Speeding up coverage queries in 3D multi-goal path planning. *Proceedings of the International Conference on Robotics and Automation* (*ICRA* 2013) Karlsruhe, Germany, May 6-10, 2013.

[35] J. Faigl, Approximate solutions of the multiple watchman routes problem with restricted visibility range, *IEEE Transactions on Neural Networks*, 21(10): 1668-1679, 2010.

[36] A. Goodenough, R. Raqueno, M. Bellandi, S. Brown, and J. A. Schott, Flexible hyperspectral simulation tool for complex littoral environments, *Proceedings of SPIE* 6204, *Photonics for Port and Harbor Security II*, May 12, 2006.

[37] A. Boudet, M. Paulin, P. Pitot, D. Pratmarty, Low memory spectral photon mapping, (*WSCG* ’04), Plzen, Czech Republic, Feb 2 – 6, 2004.

[38] J. Avro, and D. B. Kirk, Particle transport and image synthesis in computer graphics, *Proceedings of Siggraph* '90, 24(4): 63 – 66, 1990.

[39] T-T. Yu, J. Lowther, C-K. Shene, Photon mapping made easy. *Proceedings of the* 36*th SIGSE Technical Symposium on Computer Science Education* (*SIGCSE ’05*), St. Louis, MO, Feb. 23–27, 2005.

[40] X. He, K. Torrance, F. Sillon, and D. A. Greenberg, Comprehensive physical model for light reflection, *Proceedings of 18th Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH ’91*), 175-186: New York, NY, 1991.

[41] I. Wald, Photorealistic *Rendering using the Photon Map*, Master’s Thesis, University of Kaiserslautern, Sept. 1999.

[42] H. W. Jensen, N. J. Christiansen, Efficiently rendering shadows using the photon map *Proceedings of Compugraphics* '95, December, Alvor, Portugal, 1995.

[43] I. Peter and G. Pietrek, Importance driven construction of photon maps. *Proceedings of the Eurographics Workshop*, Vienna, Austria, Jun 29 – Jul 1, 1998.

[44] S. Katz, A. Tal, and R. Basri, Direct visibility of point sets. *Proceedings of SIGGRAPH* ’07, San Diego, CA, Aug. 5-9, 2007.

[45] R. Fedkiw, J. Stam, and H. W. Jensen, Visual simulation of smoke. *Proceedings of SIGGRAPH* 2001, Los Angeles, CA, Aug 12-17, 2001.

[46] http://www.sketchup.com

[47] http://sketchup.google.com/3dwarehouse

[48] T. Hachisuka, S. Ogaki and H. W. Jensen, Progressive photon mapping, *Proceedings of SIGGRAPH Asia, 2008*. 27(5), New York, NY, 2008.

[49] Y. Yagi, Real-time omnidirectional image sensor for mobile robot navigation, *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, Vancouver, Canada Oct 27-30, 2002.

[50] B. Johnson and Y. Pan, A sensitivity analysis of initial conditions for the Mumford-Shah based level set method of image segmentation, *Proceedings of ACMSE* 2007, Winston-Salem, NC, March 23-24, 2007.

[51] Y. Pan, J. D. Birdwell, S. M. Djouadi, Efficient bottom up image segmentation using region competition and the Mumford-Shah model for color and textured images, *Proceedings of the Eighth IEEE International Symposium on Multimedia* (*ISM* ’06), San Diego, CA, Dec. 11-13, 2006.

[52] E. W. Djikstra, A note on two problems in connexion with graphs, *Numerische Mathematik* 1(1): 269-271, 1959.

[53] P. Smereka, The numerical approximation of a delta function with application to level set methods, *Journal of Computational Physics*, 211(1), pp. 77-90, 2006.

[54] G. Aubert, and L. Vese, A variational method in image recovery, *SIAM Journal of Numerical Analysis*, 34(5): 1948-1979, 1997.

[55] Songsong Li and Qingpu Zhang, Fast image segmentation based on efficient implementation of the Chan-Vese model with discrete gray level sets, *Mathematical Problems in Engineering*, vol. 2013, Article ID 508543, 2013.

[56] http://www.itk.org

[57] http://www.boost.org

[58] S. Gao, J. Yang and Y. Yan, A local modified Chan-Vese model for segmenting inhomogeneous multiphase images, *International Journal of Imaging Systems and Technology* 2(2): 103-113, June 2012.

[59] M. L. Fredman and R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3): 596 – 615, 1987.

[60] T. C. Hales, The sphere packing problem, *Journal of Computational and Applied Mathematics* 44(1): 41 – 76, 1992.

[61] B. De Greve, T. Willems, T. De Muer, and D. Botteldooren, The effect of diffuse reflections on noise maps using phonon mapping. *Proceedings of the 13th International Congress on Sound and Vibration* (*ICSV* 13), Vienna, Austria, Jul. 2 – 6, 2006.

[62] M. McGuire and D. Luebke, Hardware-accelerated global illumination by image space photon mapping, *Proceedings of the 2009 ACM SIGGRAPH*, New Orleans, LA, Aug. 1, 2009.

[63] K. Zhou, Q. Hou, R. Wang, and B. Guo, Real-Time KD-tree construction on graphics hardware, *ACM Transactions on Graphics.* 27(5): Article 126, 2008

[64] E. S. Brown, T. Chan, and X. Bresson, Completely convex formulation of the Chan-Vese image segmentation model, *International Journal of Computer Vision* 98(1): 103–121, 2012.

[65] T. F. Chan, S. Esedoglu, and M. Nikolova, Algorithms for finding global minimizers of image segmentation and denoising models, *SIAM Journal on Applied Mathematics*, 66(5): 1632 – 1648, 2006.

[66] http://en.wikipedia.org/wiki/DARPA_Grand_Challenge

[67] http://en.wikipedia.org/wiki/DARPA_Urban_Challenge

[68] http://en.wikipedia.org/wiki/DARPA_Robotics_Challenge

[69] M. Kaku, *The Future of the Mind: The Scientific Quest to Understand, Enhance, and Empower the Mind*, Double Day, New York, NY, 2014.

[70] J. Hawkins and S. Blakeslee, *On Intelligence*, Times Books, New York, 2004.

[71] M. Z. Zia, M. Stark and K. Schindler, Are cars just 3D boxes? – jointly estimating the 3D shape of multiple objects, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR* '14), Columbus, OH, June 24 -27, 2014.

[72] J. Walker, A. Gupta and M. Herbert, Patch to the future: unsupervised visual prediction, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR* '14), Columbus, OH, June 24 -27, 2014.

[73] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, Scalable object detection using deep neural networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR* '14), Columbus, OH, June 24 -27, 2014.

VITA

Bruce Johnson received a B.S. degree in chemistry from Middle Tennessee State University in 1997 an M.S. in computer science in 2005 from the University of Tennessee, Knoxville. He finished his PhD in electrical engineering from the University of Tennessee, Knoxville in 2014. Since 2009 he has been working as a research scientist at the Naval Surface Warfare Center in Panama City. His research interests include wireless network sensor simulation, sensor coverage and numerical analysis.