Computational Modeling of Gaze Behaviour in Real-World Settings

by

Karishma Singh

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Thesis advisor

Dr. Neil Bruce

Author

Karishma Singh

Computational Modeling of Gaze Behaviour in Real-World Settings

Abstract

Understanding Human gaze behavior is of value from basic science to a myriad of applications. Current technology places constraints on possible use cases in requiring that much of this analysis is restricted to a lab setting, with constraints imposed by limitations of the hardware setup required.

The objective of this thesis is to establish methods that allow for collection and analysis of gaze data in real-world settings, and allow for development of computational models of human gaze strategies that are more representative of real-world behaviour.

To achieve these objectives, the thesis presents methods for deriving rich models of the real world in the form of 3D models that provide a base representation that gaze direction may be measured against, and strategies for associating experimental eye-tracking data with these models derived from unconstrained real-world settings. The thesis also presents techniques that address qualitative and quantitative analysis, and visualization of gaze data associated with 3D models.

As a whole, the body of techniques presented provides a foundation for future research efforts in real-world gaze analysis, presenting many new opportunities for

experimental studies, and computational modeling efforts.

Contents

List of Figures

Acknowledgments

I would like to acknowledge the invaluable guidance and support of my supervisor Dr. Neil Bruce. This accomplishment would not have been possible without him. I am extremely grateful to have Dr. Jason Morrison and Dr. Yang Wang on my thesis committee.

I would like to thank my lab mates for their time, support and suggestions.

I am especially thankful to my wonderful family members and friends for their continuous motivation and support.

Finally, I would like to recognize the funding sources - The University of Manitoba, Faculty of Science for their continued financial support.

This thesis is dedicated to my parents
for their love, endless support
and encouragement.

Chapter 1

Introduction

In the real world, humans move both their head and eyes to position their gaze toward objects, or in general, targets of interest. A significant body of research has been devoted to understanding and predicting human gaze patterns based on data collected using eyetracking devices [7]. Human attention is important in almost every facet of our everyday lives, and also finds value in specific application domains. There is inherent value to better understanding the mechanisms and patterns associated with gaze behaviour on basic scientific merit, and in the value of this understanding in applications ranging from marketing and advertising, to architecture and landscape design, interface design and more generally in how humans interact with other people and their environment.

Overt visual attention [8] is a fundamental source of data derived from eye tracking measurements, which involves direct observation of eye movements that are made. This provides one natural, and readily observable avenue for measuring human allocation of attention. The study of how attention is allocated also has relevance across

different disciplines including at least Psychology, Medicine, Marketing, Engineering, Education and Computer Science. Knowing the position of gaze points and their dynamics provides an observable target or location of interest from which understanding can be inferred.

There exist a variety of commercial devices used to track human gaze points that are available. These devices vary according to a variety of parameters including durability, weight or portability, degree of precision and accuracy, and form-factor (glasses vs. desktop eye-trackers). Eye tracking glasses currently present the most portable tool for exploring eye tracking behaviour in real environments where participants can wear the glasses while simultaneously performing activities such as shopping in a grocery store. Such data is of value in analyzing human behaviour, including in the example given, where patterns are revealed about consumer behavior while making a decision about products available in a physical retail setting.

There are different classes of eye movements that include reflex based movements, pursuit (e.g. following motion), and fixations [9]. Fixations are typically characterized by sustained gaze within a localized spatial region than is sustained for some minimum period of time. A hallmark of fixations is also found in a preceding saccade, which involves a rapid eye movement to the point of fixation, marked by rotational acceleration of the eye. Visual information obtained during fixation is often a critical factor in determining points of interest for viewers, and availing different opportunities for application specific analysis. For example, the design of packaging or placement on a supermarket shelf may contribute to allocation of attention and [10] drive decision making strategies [11].

While portable head-worn eye trackers provide new possibilities, the extent of analysis that is possible using such devices remains rudimentary. If one has a highly controlled environment (e.g. a specific supermarket shelf), and data from a relatively stationary observer standing in front of this shelf, then qualitative analysis may be performed in examining topographical interest maps (heatmaps) in the 2D reference frame of the shelf, or considering peak regions of focus. There remains a significant gap between these scenarios (and even more basic lab settings) as compared with unconstrained viewing in physical real-world environments. The gap is due to limitations that come from physical or theoretical constraints on existing solutions to eye tracking measurement; Analysis is typically limited to a projection of the real world (e.g. a 2D image or set of images captured by the camera).

For all of the forementioned reasons, there is merit in considering how constraints on the study of gaze behaviour may be further relaxed through technical developments to allow for observation and modeling of behaviour in natural settings. This is especially important as evidence continues to emerge that gaze data collected in controlled or lab-based settings is a poor proxy for how humans actually behave in the real world [10].

1.1 Contributions

It is important to note that the central objective of the thesis is not the study of gaze behaviour itself, but rather developing a set of methods, techniques and algorithms that may significantly enrich the capacity to study human gaze behaviour in real-world settings and absent of current constraints that encumber what may

be measured or observed. This serves to enable further understanding of basic science in different fields of study that involve humans, or towards a variety of useful applications. The central contributions of this thesis are describe in what follows:

- We explore different solutions for achieving less constrained measurement of gaze behaviour in the real world;
- We examine methods in photogrammetry in detail paired with data from a portable eye tracker towards removing existing limitations on the study of gaze behaviour;
- We determine the best methods for building 3D models of the environment based on captured video or including additional sensors to provide context for studying gaze data;
- We propose different strategies for visualizing gaze data in 3D real-world settings; and
- We present quantitative metrics that may be applied to compare measured gaze data across different observers, across different experimental conditions, or derived from predictions of a computational model.

1.2 Thesis Organization

The remainder of the thesis is organized as follows. In Chapter 2, we briefly discuss relevant background to the topics addressed by the thesis. This is followed in Chapter 3 by coverage of existing related and relevant work that establishes current technology. Chapter 4 establishes the central methods and considerations required to address the problems identified in the list of contributions. Chapter 5 provides qualitative and

quantitative analysis in the form of techniques for visualization of gaze data in 3D real-world environments, and suitable metrics for comparing densities or gaze points on 3D models. Finally, Chapter 6 includes discussion of major contributions and provides a road map for future research efforts.

Chapter 2

Background

Eye movements provide an understanding of images in two ways. First, they highlight important regions or elements of the image by showing where the viewers attention is directed. Second, the patterns of saccades and fixations provide valuable information about the dynamics of the viewers gaze behavior. For example, the process of reading does not involve a smooth sweeping of eyes over the text, but instead a sequence of discrete fixations along the line of text [12].

Fixation involves maintaining a single location as the locus of gaze for an extended period of time. This may include involuntary eye movements. For example, the process of reading involves fixating eyes on successive locations across the screen [9], and sudden movement or distinct colours may automatically draw an observer's gaze [13].

Eye trackers used for recording gaze data [14; 15; 16] may be grouped into 3 types of categories: a) measuring the movement with a special lens or coil attached to the eye for very high precision (both spatial and temporal) measurements , b) optical tracking using a camera and without contact to the eye. This sometimes

involves using light in the near infrared wavelength as the pupil is easy to locate and segment automatically. c) placing electrodes around the eyes to measure the electric potentials. These trackers are used for research purposes in biology, psychology, HCI and a variety of application domains. The most popular method for eye tracking involves the use of video from a camera pointed at one or both eyes to extract the position of the eye in order to determine the direction of gaze.

The sections that follow outline prior work related to different facets of the proposed thesis. These include efforts towards 3D reconstruction, those that consider gaze within a 3D coordinate space, and predictive modeling efforts that are relevant to the methodology developed as part of the thesis respectively.

2.1 Capturing 3D Representations

There are a variety of methods for generating 3D representations of scenes from collections of still images or video. Some of the principal strategies include SLAM-based techniques and Photogrammetry. While Photogrammetry works with images for recovering the positions of surface points of a scene, SLAM methods track camera pose in a 3D environment based on drawing correspondences between discriminative features that are stable to affine transformation. While creation of a 3D model may seem removed from capturing gaze data, it provides a basic substrate for analyzing this data. That is, a 3D model allows for captured gaze data to be related to the environment, including offline for post-hoc analysis. Moreover, determination of depth based on stereo eye tracking is generally unreliable, and allowing a physical representation where eye rays can intersect a 3D model helps to address this challenge.

Photogrammetry

Baltsavias [17] discusses photogrammetry based on processing of images for 2D and 3D model reconstruction including for visualization, simulation, and other purposes. Large scale photogrammetry has been challenging in the past owing to the need to consider large numbers of potential matches between corresponding points across different viewpoints of an environment. As the number of images increases, the number of potential matches grows significantly. Modern hardware including GPU implementations of matching algorithms has lowered the computational burden that this poses. Photogrammetry [18] is a photography based approach that reconstructs objects in photographs by finding corresponding points in the real world across different images that may include overlapping objects or structure in the real world. With careful calibration, and good point matching, it is possible to derive a 3D model(see Fig. 2.1) consistent with the set of relative distances between features across multiple viewpoints.

Figure 2.1: Left: Collection of images extracted from a video of a real-world scene. Right: General view of application window in PhotoScan. The view of the model depends on the current processing stage which is point cloud in this case.

There is also a special case called stereophotogrammetry [19] which considers pairs of stereo images to establish these correspondences. The rays formed from the position of camera to the point on an object are intersected to form a triangle to determine the three-dimensional location of the point. With recent developments in computer vision, the need for stereo to achieve good depth estimation carries much less importance then it has traditionally [20].

Photogrammetry has various applications in geology, architecture [21], topographic mapping [22], archaeology [23], and in creating photorealistic environments for PC games.

Photogrammetry is an area that has been studied in some depth, and there exists a rich set of well established techniques for photogrammetry. The central elements for point matching and triangulation have been implemented many times over, and a variety of open source and commercial software applications implement these techniques along with subtle optimizations to improve final results. An image [24; 25] is formed from a projection of 3D geometry corresponding to the real-world. Objects are projected onto a 2D plane which results in loss of information about the depth of the object. However, a 3D point in the real world corresponding to a specific point or object in the image need be on the line of sight or a ray constructed from the initial camera location to the point on the object. Given two images, the location of a 3D point can be determined in world coordinates by the intersection of two projected rays from each of the respective viewpoints, a process called triangulation.

Given the rich body of existing work in photogrammetry, we don't aim to build a new solution from the ground up. With that said, it is reasonable to expect that for

Figure 2.2: The final stage in photogrammetry converts a 3D mesh into a textured 3D representation of a scene.

this particular problem, some approaches may be more effective than others. It is also the case that the type of data available (e.g. low resolution video, high-resolution SLR images, etc.) is relevant to the method chosen for 3D reconstruction. The outward facing camera on most eye tracking glasses is relatively low in resolution compared to modern cameras that have a portable, but not wearable form factor. An ambition of this section is therefore to determine the best approach, and set of methods subject to characteristics of the hardware. In addressing this, we examine a variety of different options for depth and 3D from imagery, including what concessions are made when lower resolution images are used.

The most challenging part of the process of converting multiple 2D images to a 3D model is determining the depth of the object in a given image, which is necessary for representing a point of object in 3D world coordinates. The depth determination involves comparing two images to find the position of matched elements which can be triangulated to find the 3D point. Finally a 3D mesh is created by calculating and combining the depth maps in considering a consensus across many potential matching points across views [26], which conform to the multiple views that form observations in the data.

Möller and Trumbore [27] have given an algorithm for determining very fast ray-triangle intersections. The method is advantageous as it does not require computing and storing triangle plane equations hence reducing memory storage costs.

Open source software for photogrammetry includes Visual SFM [3], CloudCompare [28], the Python photogrammetry toolbox [29] and several other implementations, all of which build point cloud data from a collection of images. One prominent

approach in the category of "Structure From Motion" is VisualSFM which attaches a GUI to the underlying algorithms for 3D reconstruction. This implementation also uses SiftGPU [1] and the CMVS/PMVS2 [30] tool. SiftGPU rapidly finds key-points 2.3 in different images and putative matches to estimate camera positions while CMVS/PMVS2 creates a point cloud from the matched images.

Some of the most popular commercial photogrammetry software applications are Autodesks 123D Catch [31] and Agisoft Photoscan. Agisoft Photoscan [32] is a professional photogrammetric tool used for processing digital images and creating professional quality 3D models that are used in various fields, in particular, for GIS applications, and for 3D reconstruction from UAV captured video [33]. It is based on latest multi-view 3D reconstruction algorithms and provides automated image alignment as well as 3D model reconstruction.

Given the long history of existing research in photogrammetry, as mentioned, it is not the ambition of the thesis to significantly advance methods in photogrammetry. However, given hardware typical of portable eye trackers, possibly supplemented by other sensors with higher quality optics, we endeavor to consider both the image characteristics and set of algorithms that give rise to high quality models and provide a workflow that has the smallest overhead beyond using only the base data captured by wearable 3D eye tracking glasses.

Structure from Motion

Structure from motion (SFM) [2] is a low-cost photogrammetric technique to estimate 3D structures from a series of overlapping, offset 2D images (generally a moving

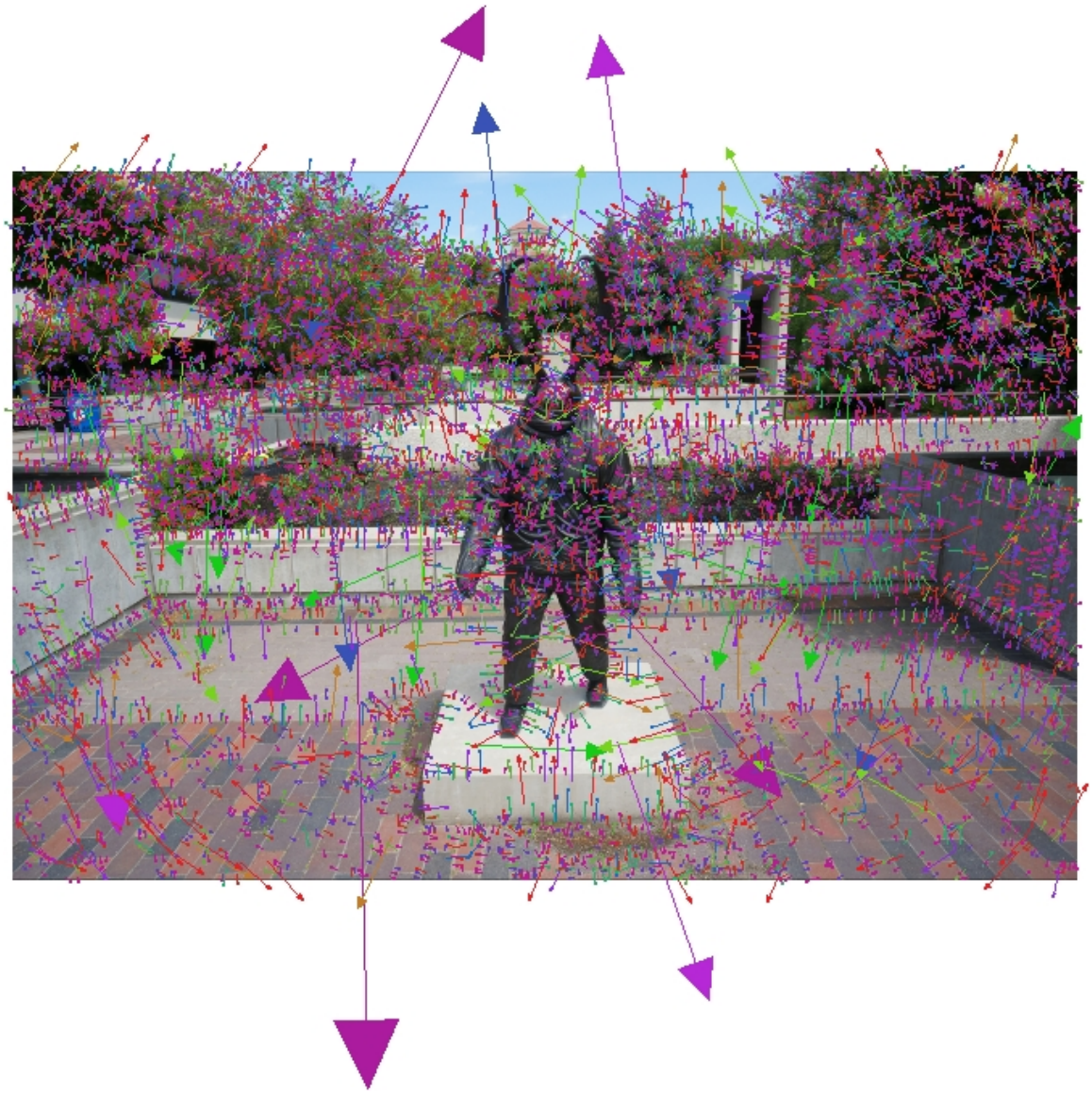


Figure 2.3: Keypoints on the object can be extracted by SIFT [1] for feature description.

camera). However, there is a difference between SFM and conventional photogrammetry, where the geometry of the scene is resolved without necessarily estimating known 3D positions from matching control points or 3D location and pose of the cameras. Instead, SFM reconstructs the camera pose and scene geometry simulta-

neously by using bundle adjustment, which automatically identifies and extracts the matching features in multiple overlapping images. These matching features in the Fig. 2.4 are tracked on an image to image basis thus enabling estimates of camera positions, and with tracking providing savings over estimates that involve views with no known relation. This approach is suitable for images that cover most of the full 3D structure of the environment viewed from varied positions or images, and captured by a moving sensor.

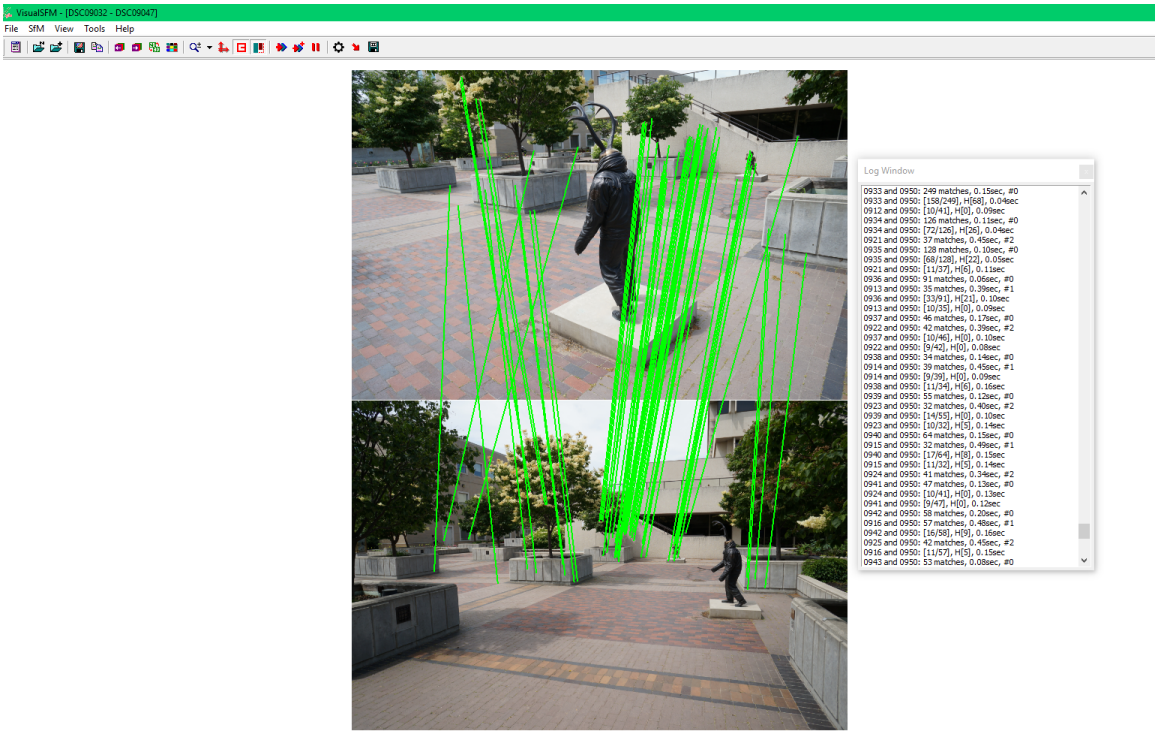


Figure 2.4: Structure-from-motion [2] performing feature detection and matching between different views of the images to estimate the camera positions.

The ground-control coordinates provide scale and orientation of camera positions, and this is often lacking in SFM based methods since the goal is directed more heavily towards reconstruction rather than pose estimation. Hence, most existing SFM based

methods generate dense point clouds in an image-space coordinate system aligned to a real-world coordinate system. We conducted experiments with the most popular freely available application VisualSFM [3] to process sequences of scene camera images and generate a corresponding point cloud. This GUI application runs very quickly as it is optimized for multicore parallelism for the most computationally expensive processes involved in reconstruction including feature matching, bundle adjustment, and feature detection. This package integrates various open-source applications including, SIFT on GPU(SiftGPU), Multicore Bundle Adjustment [34], CMVS [30] and PMVS2 [35]. Furukawas CMVS/PMVS2 toolchain is used by VisualSFM for dense reconstruction. The output produced by VisualSFM can work with many additional tools, including MeshRecon [36], SURE [37], CMP-MVS [38], and MVE [39]. In many cases, careful tweaks can result in improved meshing, which seems to be a significant difference between commercial solutions and open source implementations.

Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) is a computational problem which involves self-localization and mapping while navigating an (unknown) environment. This method is often applied in real-time, hence the navigator or the agent receives real-time estimates of the most recent state. Since this approach includes camera pose estimation and mapping of the environment it requires a reliable estimation of the scene geometry. 3D SLAM [40] has recently been one of the popular research topics because of its value to robotics, and in augmented and virtual reality applications which, require accurate pose estimates of a moving camera. It is also

a pre-requisite for performing many tasks in robotics including planning navigation along traced paths, localization, exploration and autonomous navigation. Various dense tracking and mapping methods have been developed including feature based methods. Many of these are limited to 3D reconstruction of a small environment, and typically an indoor environment (e.g. a part of a room). The reasons for this become clear from the discussion of recent SLAM methods that follow, in particular those that make use of depth sensing cameras.

Christian et al. [41] developed a method for camera tracking using RGB-D images. This method aims to minimize the photometric error by registering two consecutive RGB-D frames upon each other. To improve accuracy, features are also tracked over multiple frames from RGB-D camera. Previously, Bundle adjustment was used as one method for pose estimation and optimization [42], but these approaches lean more heavily on depth information and place less value on interest points or affine invariant features, causing loss of valuable information for some environments. They derived a probabilistic formulation to estimate the motion of a camera from RGB-D images derived from depth sensors (e.g. Microsoft Kinect). This work was extended in another paper [41] which presented a dense SLAM method for optimized trajectory estimation. This latter work provides a consistent 3D model(point cloud) and an entropy metric that forms the basis for loop closure and for drift-free pose estimates.

Jakob et.al [43]proposed a method called direct monocular SLAM which allows construction of large-scale maps of an unknown environment in addition to locally tracking camera motion. This algorithm uses a direct image alignment process which enables highly precise estimation of a pose. A pose graph is constructed, which

consists of keyframes and also computes 3D similarity transforms (rotation, scaling, translation) on vertices and edges respectively and is also relatively invariant to changes in scaling of observed geometry. Besides running on a CPU in real-time, this approach to SLAM can also run on a smartphone as if only odometry [44] is desired. As described in [43] this algorithm works on three main components: tracking of new frames, depth map estimation and map optimization. The tracking component estimates the pose of a rigid body with respect to the current keyframe considering the previous frame for initialization. To estimate a depth map, the current keyframe is replaced by the tracked frames. The map optimization component optimizes the global map using a pose graph optimization process [42] and loop closure is also performed for outdoor trajectories.

Erik et. al [45] propose a novel method of using a signed-distance function (SDF) for real-time camera tracking and 3D reconstruction of the environment from RGB-D images. This function minimizes error for depth images from an RGB-D sensor giving efficient camera pose estimation. They have used an iterative approach which involves estimating the camera pose based on the previous signed distance function and comparing this with the signed distance function from an new camera pose, and then updating. What makes this approach novel is that they optimized the camera pose directly on the signed distance function as compared to other approaches like KinectFusion [46].

It is important to note that many recent approaches have shifted to leaning on depth estimates, or depth cameras. This has no doubt produced improved capabilities. However, it is worth noting that almost any approach based on structured light

fails in outdoor environments. There is typically too much near IR sunlight for any approach to rely on structured light during the daytime. An alternative is to derive distances from time-of-flight, but this also presents limitations based on external lighting, and also on possible range. That is, content near to, or far from the camera is poorly represented.

2.1.1 Open Source Software

Autodesks 123D catch [47] is a great choice for building 3D models of simple objects from images but it has certain constraints that limit it's use. For example only a limited number of images (maximum 40-70 depending on size) may be used. Even though the software is freely available, the source code is not. This results in a black box nature where there is little to no control over the reconstruction process. When dealing with different sensors with varying properties, this is a consideration that is very limiting. There are several free open-source methods available to create 3D models using photogrammetry that produce better results than 123D catch, and also have more flexibility. These methods have been developed over decades of research and are capable of producing high quality representations of environments provided suitable source images are available. In all cases, this is achieved by examining the matching features in each image and performing 3D reconstruction based on overlapping features. Some methods perform the entire process from image alignment to textured model while others reconstruct a mesh from dense point clouds only, or do sparse structure from motion(SFM) 2.5. The selection of functionality present in these different implementations is why there are often cross-dependencies across

available solutions even if the core involves on particular open source implementation.

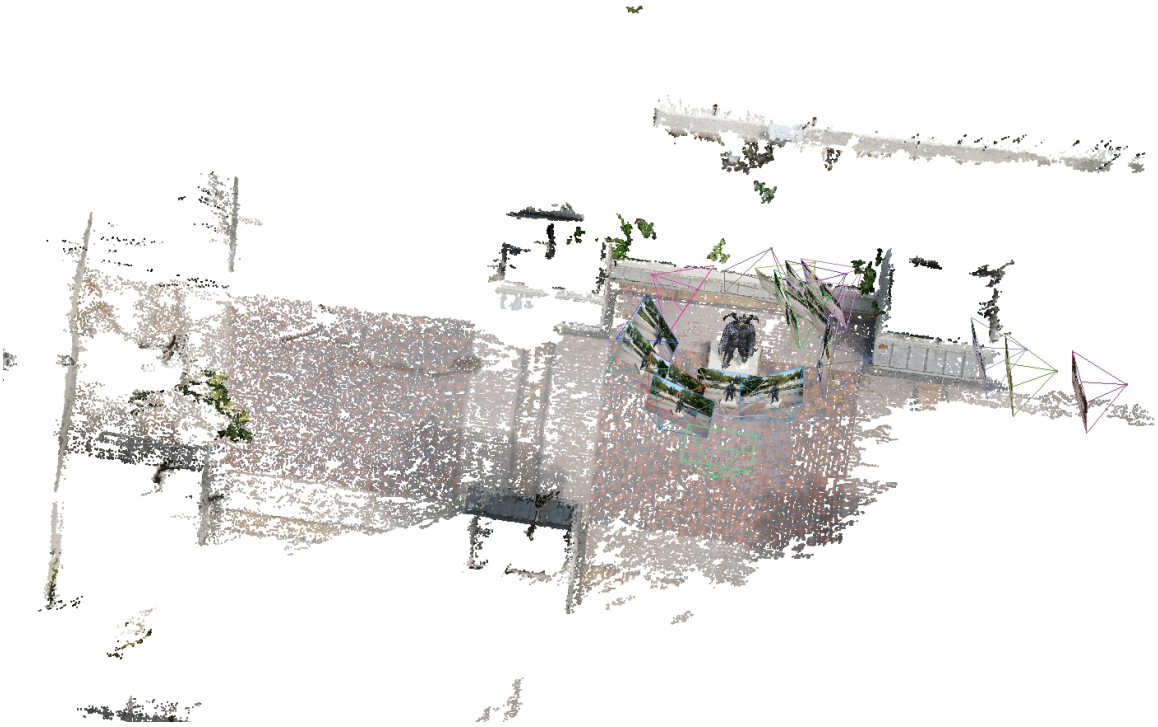


Figure 2.5: VisualSFM [3] uses multi-view stereo algorithms to reconstruct 3D structure of a scene visible in the images.

The Python Photogrammetry Toolbox consists of a set of python scripts which automate Bundler and CMVS/PMVS as well as providing a GUI. Bundler [48] relies on structure from motion to reconstruct 3D geometry of the scene as output. It successfully runs on unstructured image collections from the internet as well as structured image collections. Since Bundler produces a sparse point cloud, Furukawa introduced PMVS2 [35] for producing dense point clouds. PMVS is based on multiview stereophotogrammetry which functions similar to Bundler, taking a set of images and camera parameters as input and producing a 3D reconstructed model of the scene. It includes only rigid structure during reconstruction and ignores non-rigid structures.

The objective of multi-view stereo (MVS) [49] algorithms is to reconstruct a 3D model from multiple images of real objects or scenes captured from camera viewpoints. MVS is a term for a group of methods that solve the stereo correspondence problem when two images of the same scene are used [49] (even if not captured from a stereo camera). MVS algorithms are typically not recommended for a large number of images due to computational complexity. In such cases, CMVS [30] uses the output from an SFM solution (e.g. Bundler) and decomposes images into a set of clusters of feasible size. That is, lightweight sparse point-cloud and camera pose estimation is applied first. Following this, smaller subsets can be targeted with deeper analysis using PMVS to process each image cluster in parallel and independently followed by the union of all the reconstructed clusters to produce an overall consistent dense model.

CAD(Computer Aided Design) based tools [50] and passive image-based methods also exist. The overall approach of these methods is largely the same as those discussed, which each rely on collecting images from different viewpoints, providing or estimating camera parameters, and possibly reconstructing the 3D geometry of the real world scene. SFM algorithms are a bit different than MVS because the input is typically a series of images with some relation and estimation of camera pose is not a critical consideration.

Open Multi View Geometry (Open MVG) [51] is a library consisting of available code, that is highly readable which making it easy to use and adapt, and provides access to Multiple View Geometry problem solvers. It supports a range of methods from feature matching /detection in images to structure from motion, and provides a complete SFM pipeline. Open MVG is divided into various modules: libraries, sam-

ples, and software. Open MVG uses SIFT key points and their associated descriptors to compute matches between image pairs akin to many of the other methods.

As a whole, one can see a great deal of commonality across different methods. Ultimately though, the devil is in the details. This is revealed in our experimentation given that very different qualities are observed in results depending on methods applied, or depending on source data, even if these methods rely on many of the same common theoretical tenets.

2.1.2 Commercial Software

Photomodeler [52] is a commercial software offering providing accurate measurements of distances from photographs and for generating 3D models of a captured real-world scene. The concept of desktop close-range photogrammetry was introduced by this product and this has been one of the more prominent commercial offerings for this specific application. As this is a professional tool, it has been used in industries for performing critical measurement and modeling tasks. This tool is highly accurate, efficient, requires no expensive hardware and can provide measurements for both large or small scenes.

Another comparable product is Pix4D that transforms (possibly multispectral) images into high-quality 3D maps and models as shown in Fig. 2.6 using computer vision and photogrammetry algorithms. It is primarily designed for images from UAVs and produces results ranging from point clouds to terrain models, orthomosaics and textured models. Pix4D mapper Pro uses aerial imagery and produces a series of georeferenced 2D maps and 3D models. In addition to generating 3D textured

models and photorealistic texturing, it also allows for more rudimentary or domain specific representations including point clouds, contour lines for visualization of an areas topography, digital surface models, and other representations especially relevant to surveying.

Figure 2.6: High-quality 3D model created by Pix4D Mapper.

Autodesk ReCap [4] is a more recent product that follows 123D Catch, building this into a commercial product with more robust capabilities. This is provided in the form of both desktop software and cloud services to produce 3D models from captured real-world imagery and/or laser scans. This product appears to be relatively capable in producing models for large datasets, owing in part to cloud based computing

support.

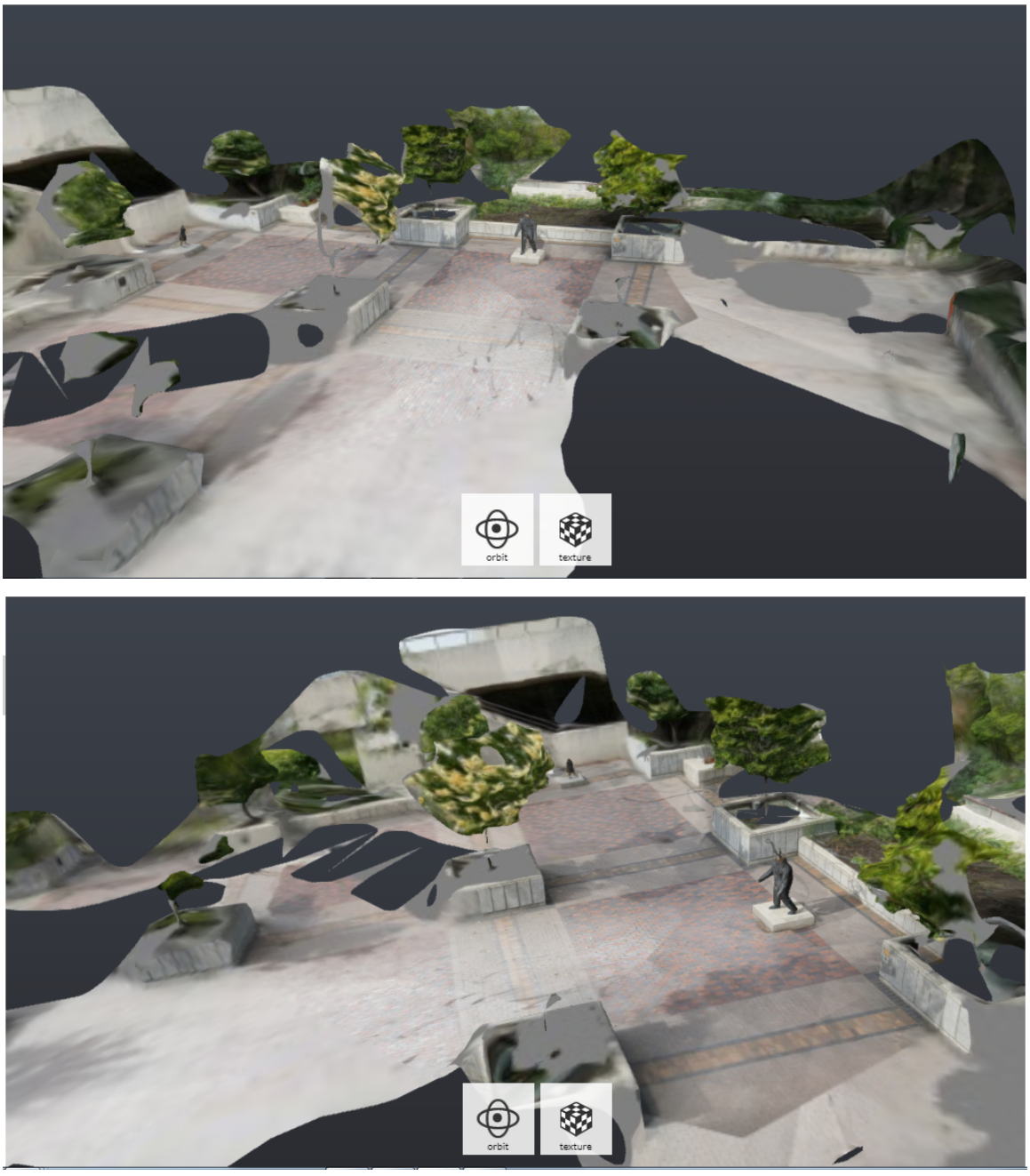
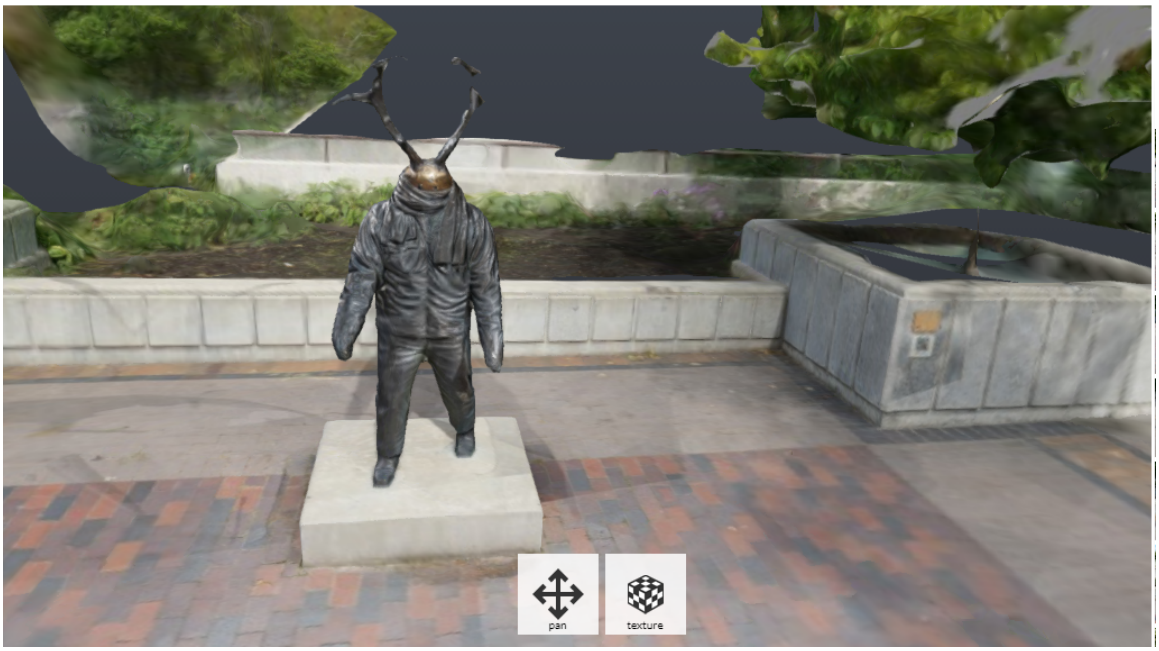


Figure 2.7: High-Resolution 3D models created with Autodesk ReCap's online application [4]

Besides providing Photo-to-3D cloud service this app allows building 3D photo-textured meshes, photo-based point clouds,orthographic views with elevation maps and model viewing options such as texture, shaded, and non-shaded as shown in Fig. 2.8a, 2.9a and 2.9b respectively.

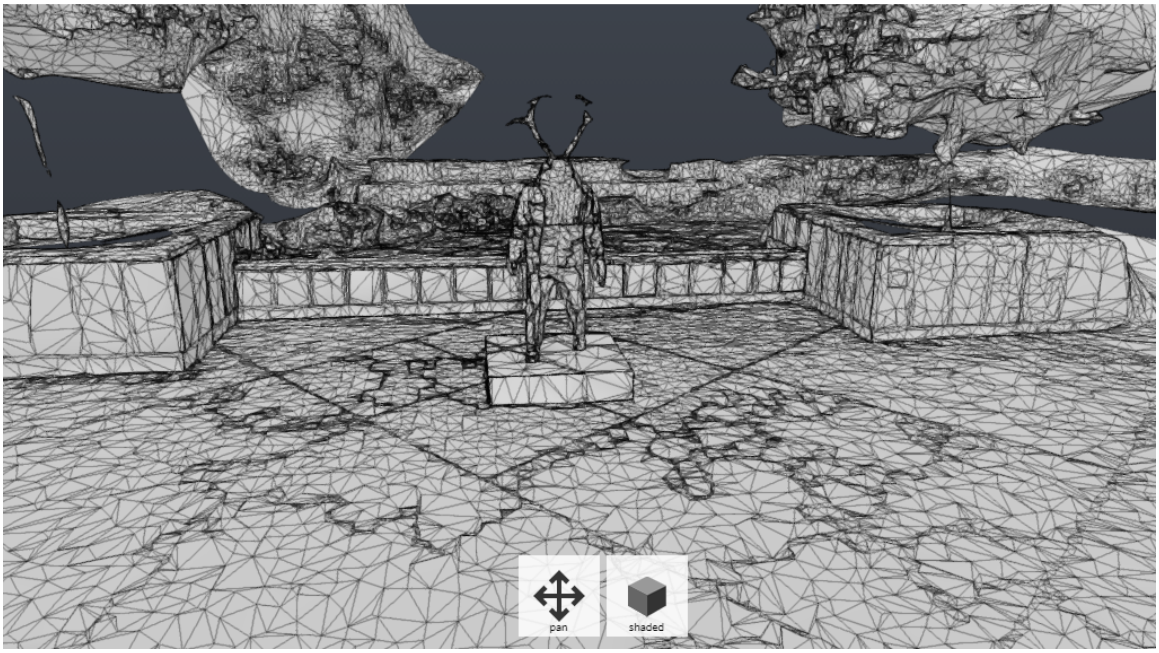


(a) 3D textured Mesh

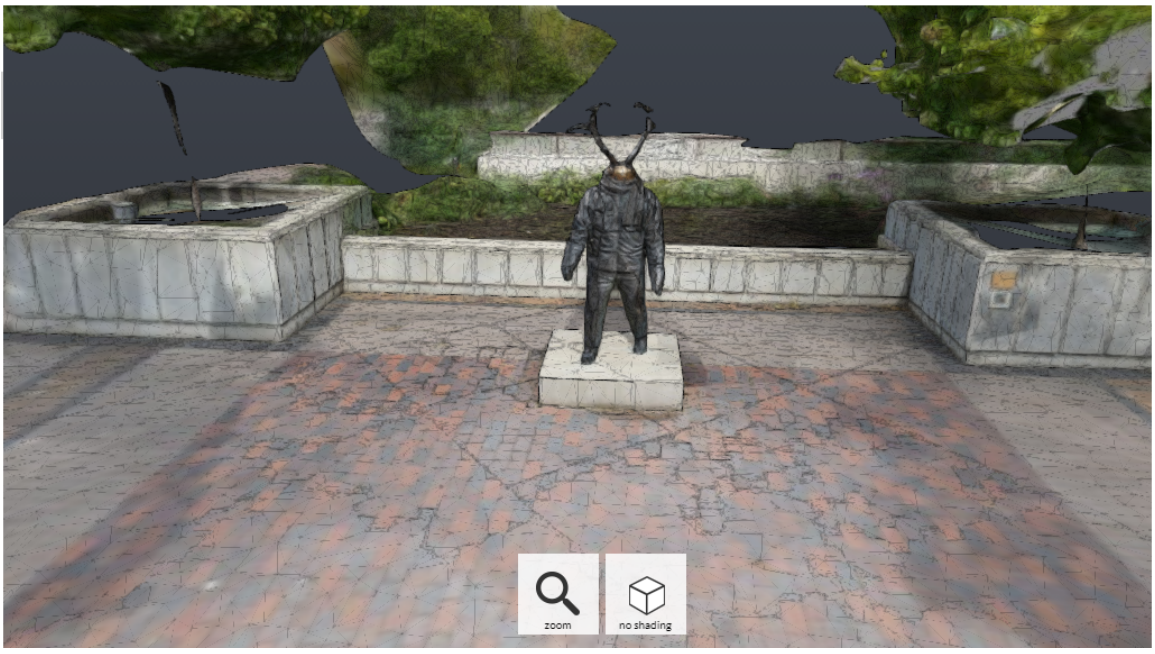
Autodesk Remake [53] was built to handle image-based 3D models and has been replaced by ReCap Photo, which has expanded photogrammetry capabilities to support drone and UAV workflows.

2.2 Human Gaze Behaviour

Human gaze behavior while freely viewing a scene provides significant information about context or an observer's intent. In identifying areas of interest in conjunction



(a) Shaded Mesh



(b) Non-shaded Mesh

with the content of the scene, one can better understand gaze behaviour or behaviour specific to a particular domain. Past research has examined these characteristics in detail through experiments designed to explore specific characteristics of fixations observed during scene viewing, and to understand implications interpreted in terms of neurobiology, psychology, social interaction, interaction with media, and in other domains. The following provides some background on human eye movements that serves to support understanding the methods and analysis that form the balance of this thesis.

2.2.1 Types of Eye Movements

The objective of eye tracking analysis is to analyze patterns of eye movements typically involving tasks such as free viewing of images, reading, driving etc. Eye tracking devices record these eye movements or the gaze points i.e., where a user is looking. There are different types of devices used for measurement, and each devices sample rates depend upon their sensor characteristics. In experimentation that appears in this thesis, SMI eye tracking glasses [54] are used, which allows recording of a users gaze direction in the reference frame of a forward facing scene camera with a sampling rate of 120 Hz in real-time. At a high level, the human eye primarily serves two main functions, tracking and fixation [55].

Fixations are the most common type of eye movement featured in eye tracking analysis. Fixations are derived from the raw temporal sampling of gaze direction recorded by the eye tracker. This forms an aggregation of point of gaze on the basis of a cluster of samples that reside within a specific localized area and span a time that

is typically between 200ms and 300ms [55]. There are a variety of criteria that can be used in this aggregation process, and most methods balance the characteristics of instrumentation with results from behavioural psychology. Fixation occurs when our eyes are relatively stable and target a particular location while scanning or reading to extract more detailed location specific information. This process is critical, as resolution of sampling is much higher at the center of gaze, and many tasks require fixation to be performed with any accuracy. Fixations are therefore a result of the processing of raw gaze data based on fixation detection algorithms. Details of fixations can be exported from the BeGaze analysis software as shown in Fig. 2.10. There are additional types of eye movements that are relevant to mention, however as with other work, this thesis is concerned exclusively with fixations.

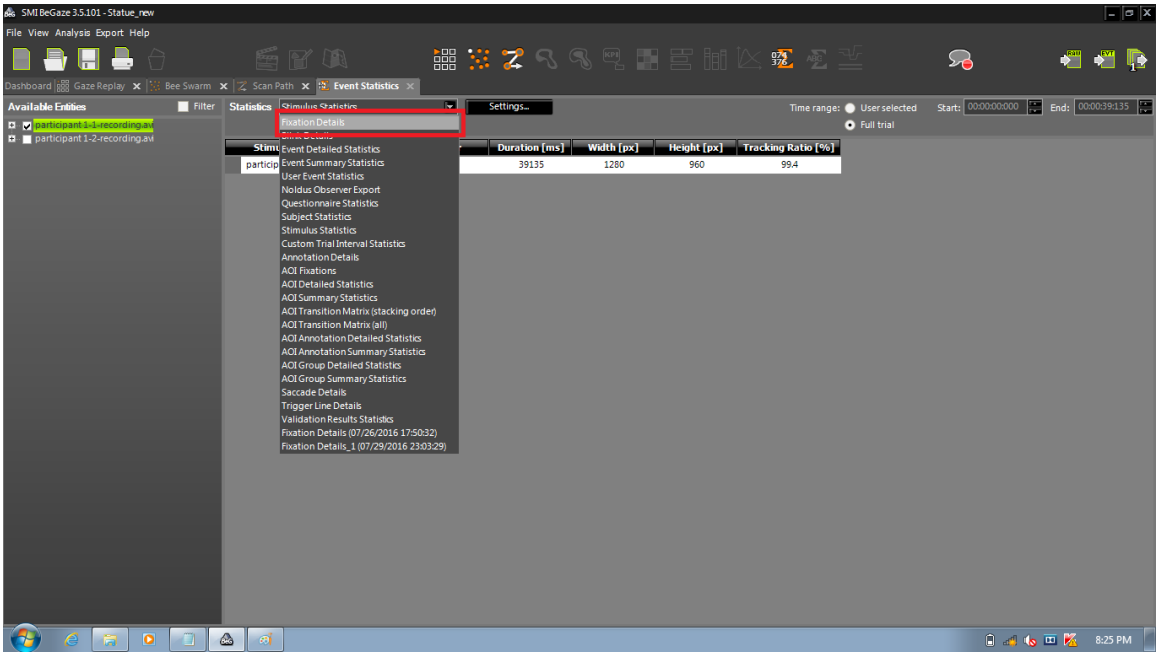


Figure 2.10: List of fixations exported from the BeGaze analysis software. [5]

Common statistics associated with the analysis of fixations include fixation dura-

tion (in milliseconds), number of fixations (i.e., fixation count) fixation position, and rotational acceleration of the eye preceding a fixation. There are different algorithms for identifying fixations. Salvucci and Goldberg [56] proposed algorithms based on velocity, dispersion-based algorithms, and area-based algorithms and classified algorithms in terms of how they utilize the retrieved visual information in eye tracking analysis.

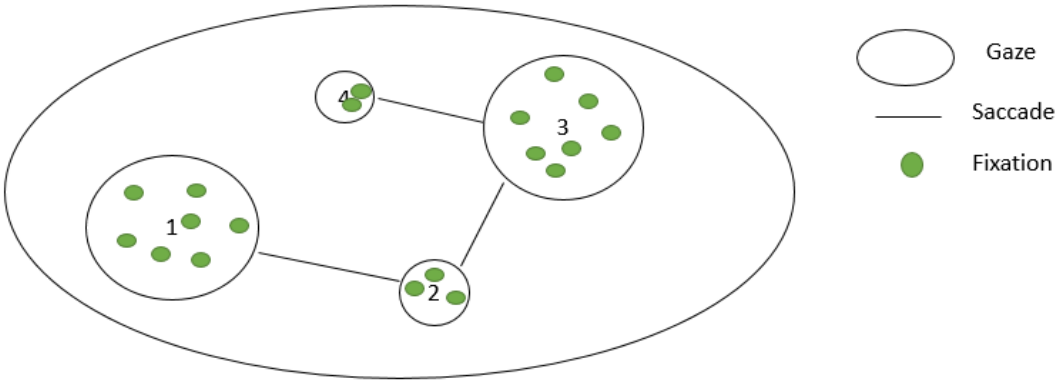


Figure 2.11: Types of eye movements observed while tracking any object.

There are six different types of eye movements that are commonly discussed [55]:

Saccadic Movements: These are ballistic movements since they have a fixed start and end point when triggered, and no correction is possible during the movement itself. Planning and execution of a single saccade is task dependent and takes about 150-200ms hence they are very rapid allowing the eyes to focus on different parts of a scene, image or data to retrieve visual information. The average span of a saccade is 20-40ms [57]. Common metrics for saccades are amplitude, velocity and duration.

Smooth Pursuit Movements: These eye movements track objects in motion. This is an important ability that allows for tracking of all kinds of moving objects in the real

world. This function is determined by the speed of the moving objects which means slower objects can be continuously tracked and are more easily analyzed visually. Very rapid motion often involves a combination of saccades and pursuit to 'keep up'. The key differences between smooth pursuit and saccadic movements are: 1. the former is continuous and requires feedback to track objects in motion, while the latter moves in an uneven, jumpy manner. 2. The degree of clarity, or information that may be discerned is higher for pure pursuit.

Vergence Movements: Saccades and fixations typically involve identical movement of each of the two eyes. In a situation where depth changes, the eyes need to either converge or diverge to keep objects in focus. In this case, the eyes may move in opposite directions inward to examine a closer object, or outward to observe an object farther away. These are slower than saccades and their main objective is to detect an objects distance, which helps the eyes to focus as well as intersect to the area of interest. This is termed a disconjugate [58] eye movement where eyes move in opposite directions as compared to saccades and smooth pursuit which are conjugate eye movements.

Vestibular Movements: These eye movements are influenced by the vestibular system. The interaction between vestibular and visual systems is referred as vestibulo-ocular reflex (VOR) [59]. This reflex function is used to stabilize our vision during head motion by causing eye movements in the direction opposite to head movement. This is especially the case where head movement changes relative to the direction of gravity. The objective of these movements is to retain the point of interest on the fovea during head and body movements. These are slower than saccades and faster

than smooth pursuit movements.

Optokinetic Movements: These are one of the gaze stabilization mechanisms which combine with vestibular movements to maintain a stable gaze during head rotation and translation while viewing. These eye movements are generated in the direction of large-field visual motion across retina. The purpose is to keep a constant alignment of fovea with the target despite of head and body movements. These might occur when looking out of a window while riding a rapid train or subway.

Physiological nystagmus: These eye movements are involuntary and detectable under special conditions where the eye aims to remain fixed.

Chapter 3

Related Work

Much of the more foundational work pertaining to this thesis is addressed in the preceding Chapter. This includes fundamental work relating to 3D reconstruction, and properties of gaze behaviour and fixation.

In contrast, this chapter deals specifically with prior efforts that have sought to achieve gaze tracking and analysis in real-world or 3D environments. This is a problem that has not been addressed to a great extent, in part owing to the fact that addressing the problem requires a relatively high degree of breadth including computer vision and image processing, specialized hardware, and understanding that crosses traditional disciplinary boundaries. Moreover, the parties that will benefit most from this work are mostly in realms of study that are relatively removed from Computer Science or Engineering. This implies a need to solve several aspects of the problem from the ground up. With that said, there are a variety of prior studies that have shown interest at the intersection of gaze in less constrained settings, or gaze with 3D data. These form the subject matter of this chapter.

As with Chapter 2, related work is divided into two critical aspects of addressing real-world eye tracking analysis. The first in establishing a representation that is reflective of the real world environment that avails itself to modeling and analysis; Individual images are unsuitable for such analysis as they fail to capture the structure of the environment or context surrounding the viewer. For this reason, one element of the proposed research involves establishing high-fidelity 3D models from capture devices (SLR, HD video) that allow for a 3D representation of the world to be built. A second challenging problem, is to relate raw eye tracking data, captured in a single camera head-centric reference frame, to the coordinate system of the eyes, and to point of gaze on the 3D model. In this section, we highlight work relevant to each of these elements.

3.1 Gaze in 3D

Yun et al. [60] make the observation that human behavior observed while viewing images contains signficant information about the content of images as well as human intent towards the images. They have conducted experiments to better understand the observers eye movements while viewing images and how an observer describes those images. There is a clear relationship between the visual target and the patterns of eye movements. If there is a complex scene, viewers usually prefer to fixate their eyes towards the center of the target so as to extract maximum information. Their work is unique in focusing primarily on establishing that eye movements including fixations and saccades provide an understanding of the scene by indicating relevant content. This speaks to the overall motivation of this thesis; to allow studies that tell

us something about behaviour to take place that have not been possible to date.

Paletta et al. [61] presented different environmental scenarios in work on estimating human visual attention, which was monitored precisely using eye tracking glasses. The eye tracking glasses provide human fixations with respect to the environment which are mapped directly to a 3D model. To reconstruct the 3D environment they consider approaches like SLAM (Simultaneous Localization And Mapping) [62], which involves constructing or updating a map of an unknown environment while simultaneously keeping track of the viewer's location within the map. Earlier work that employed a similar approach was based on single camera (Civera et al. [63]). The advent of Microsoft's Kinect, which directly determines the depth information per pixel, Paletta et al. produce a 3D model in also considering depth information to reconstruct a large scale dense environment. Their methodology produced a very small re-projection error enabling recovery of human visual attention semantics. This strategy is only suitable for indoor environments due to sensor limitations, and it is expected that much higher fidelity 3D models may be produced in leveraging photogrammetric methods based on the results we found.

3.2 Computational Modeling

In various domains including computer vision and psychology, efforts have been made to model human gaze behaviour. Saliency is one factor for which predictive models have been developed, and these methods aim to predict gaze direction during free viewing of an image. Another concept for representation and analysis of 3D meshes in graphics and geometric modeling is Mesh saliency [64] which is a measure

of regional importance for 3D meshes and for which Lee et al. [64] propose a method to compute mesh saliency. There has been relatively little work on predictive modeling of gaze patterns in 3D space, and therefore there is value in establishing a set of methods that allow for research to proceed in considering such models and more realistic behaviour that is associated with real-world viewing. In the context of methods developed for this thesis, we establish quantitative metrics that have the possibility of measuring how well a prediction (e.g. that from mesh saliency) is able to predict gaze patterns. Given that this prior work is primarily concerned with the viewpoint of 3D models in graphics, it is a poor proxy for human attention on an entire scene. With that said, this thesis includes metrics for measuring success of models in predicting gaze on a 3D model through methods analogous to those used for traditional 2D image based representations.

Chapter 4

Capture of 3D Gaze: Models and Methods

In this chapter, we present methods, results and analysis related to solving the problem of capturing and representing gaze data in 3 dimensions to support modeling and analysis efforts.

To explore gaze data in 3-dimensions, an SMI ETG2 Eye Tracker is used [54]. This includes video from a scene camera facing outward from the forehead, and two eye-facing cameras providing feeds for live analysis with the SMI Eye Tracker. The eye tracker has the form factor of a pair of glasses, and is typically worn while moving around or completing tasks. While an observer moves around a static scene, simultaneous and synchronized feeds of the scene (field of view) and eye movements are recorded. Recording is performed through proprietary software from SMI is called iView ETG. The iView server program records eye tracking data and displays current point of gaze on the computer screen in real-time. The sample rate of the video

captured by the ETG2 glasses is 30hz(screenshot). The two cameras situated on the rim of the glasses capture eye movements of the user which may then be related to a 2D plane parallel to the field of view.

Gaze data captured by the SMI eye tracking glasses is analyzed using SMI BeGaze [5] behavioral and gaze analysis software for eye tracking data. This software provides a variety of visualizations from gaze plots to attention maps as well as allowing data export in various forms, including export of captured video. The software allows export of two types of scene video with and without gaze cursor (represented by orange dots). The scene video with gaze cursor is included by selecting the "Fixation" option for the "Display:" property in the Gaze Replay module as shown in Fig. 4.1. For our experiment, we exported scene videos of individual participants with the gaze cursor overlay, raw gaze data and gaze event data to further analyze the positions of the gaze cursor based on custom scripts. It is worth mentioning that for many applications, the ETG2 with the orange dots overlayed is used directly in a qualitative manner to draw general conclusions about observer behaviour.

The (x, y) coordinates given for a gaze or fixation are calculated according to the parameters of the SMI event detection algorithm. As mentioned in the background, this is a process that involves a compromise between known device characteristics, and timing of saccades, dwell and raw gaze coordinates. The gaze samples collected during viewing are processed through event detection. These (x, y) coordinates are saved in the "Fixation Statistics" file under the "Event Statistics" module (Fig. 4.2). These statistics provide the list of fixations and timing, and therefore the timestamp may be used to associate fixations with particular frames of video for further analysis

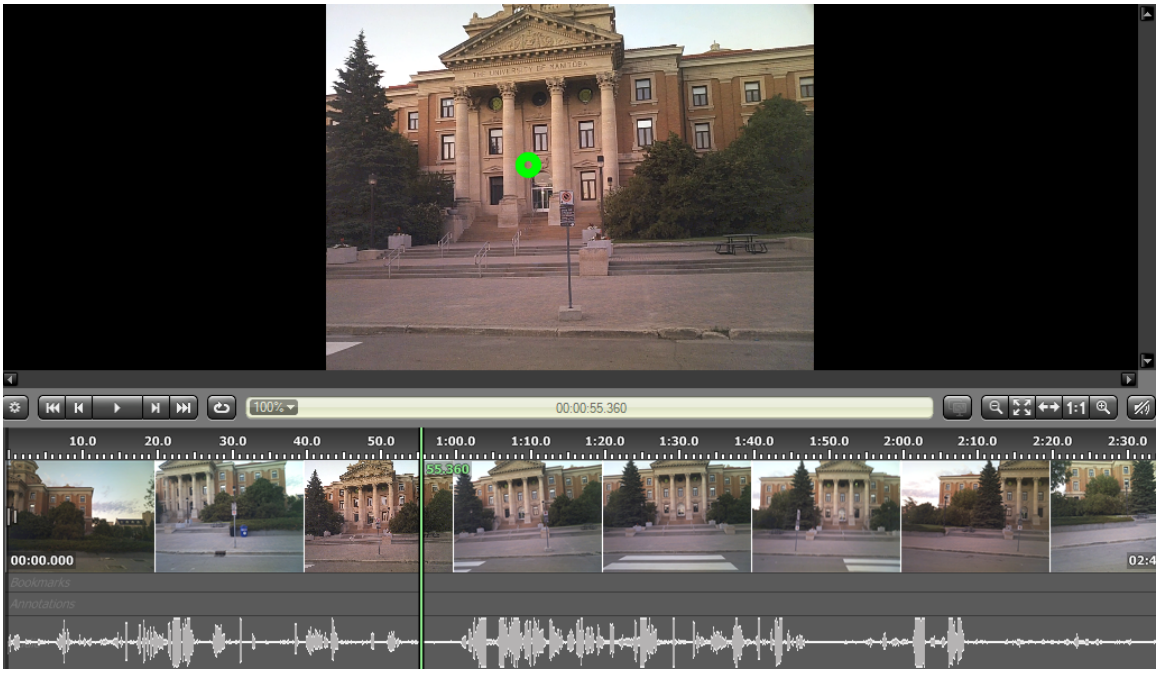


Figure 4.1: Scene video with the gaze cursor captured by the SMI eye tracking glasses.

offline and outside of the BeGaze software using custom scripts. After capturing data, a second step involves pre-processing of the recorded video sequence. In our case, this is accomplished using the Blender [65] 3D rendering software to extract frames from the video file.

4.1 Methods: 3D Model Building

Given extracted frames with fixations (or gaze points), and a corresponding dataset containing images of the scene, we seek to build a 3D model of the scene. Ultimately, the objective is to relate points of gaze to the 3D model, but for now we focus only on construction of the model. For 3D modelling several different strategies were considered through experimentation to determine what approach might be most successful

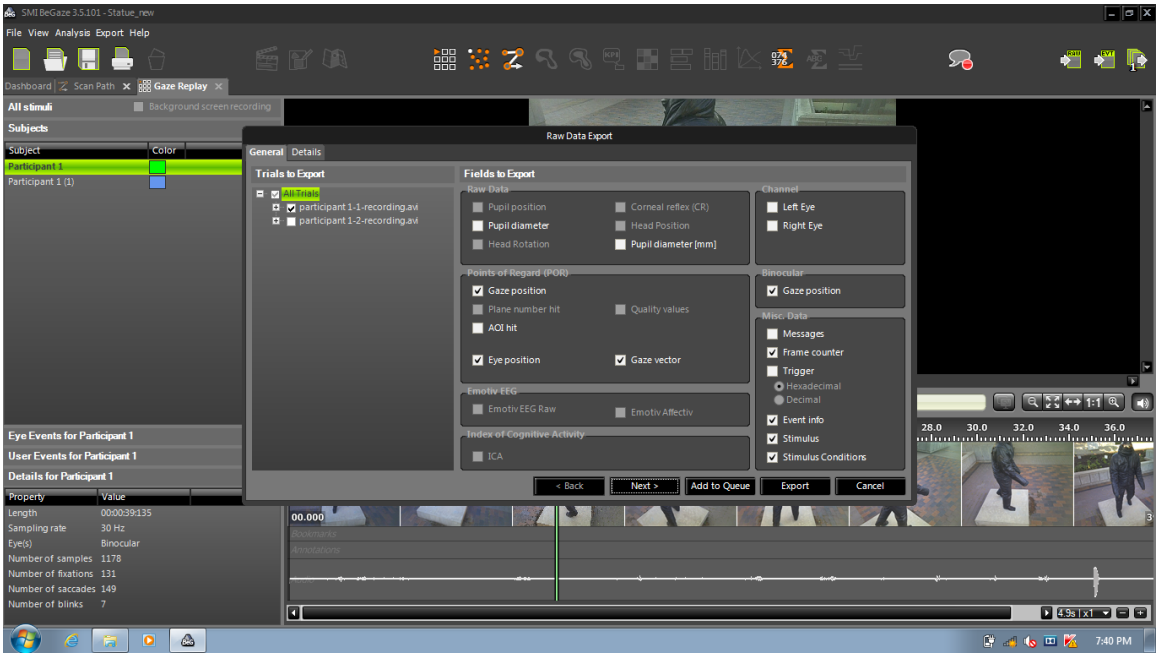


Figure 4.2: Raw Gaze data exported from the BeGaze software.

in producing high fidelity 3D scene models, or rather, to decide upon the most appropriate approach to addressing 3D reconstruction. Many attempts were made at building models based on only the scene camera images, and the scene camera images supplemented by higher quality images from a compact mirrorless Sony SLR. This includes a variety of the approaches discussed in Chapter 2, with particular emphasis on Agisoft PhotoScan, Visual SFM, and Autodesk ReCap 360. The 3D models constructed through photogrammetry were generally quite dense as compared to the ones from Visual SFM [3]. Ultimately through experimentation, it was determined that the reconstruction results from Agisoft PhotoScan were most promising. This software is based on the latest muti-view 3D reconstruction algorithms and is capable of producing effective results in an uncontrolled environment as well, making this a good choice for 3D model reconstruction. In particular, while the common

functions underlying different photogrammetric approaches are the same (matching SIFT points, sample consensus, etc.) there appear to be some optimizations that have been made that simply make the resulting mesh look better, or contain fewer artifacts than alternatives. The workflow shown in Fig. 4.3 of PhotoScan comprises of four main stages [32] Image alignment ; Dense point cloud generation ; Mesh Building ; Textured 3D Model.

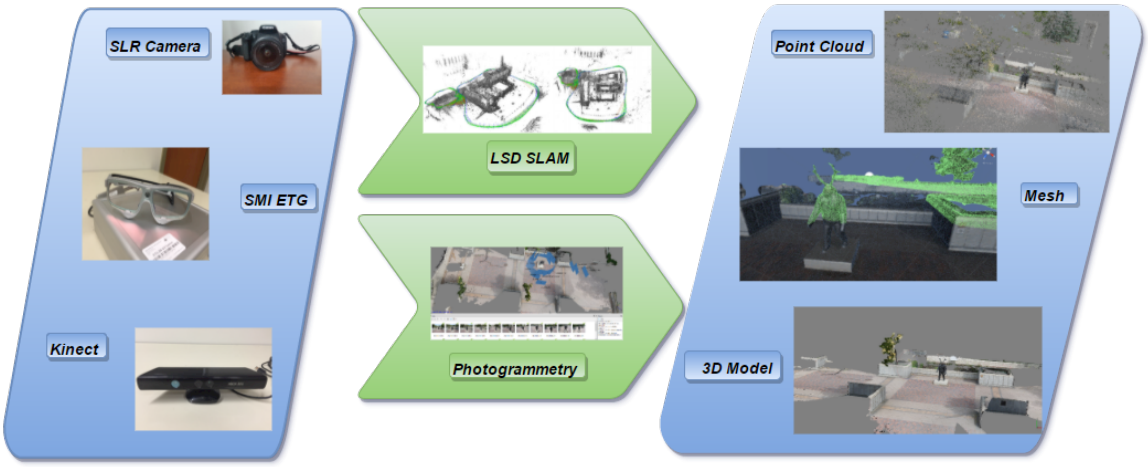


Figure 4.3: An overview of four main stages in PhotoScan to convert multiple 2D images into a 3D model.

4.1.1 Image Alignment

The dataset consisting of gazed at images (from BeGaze and Blender), are loaded into PhotoScan. Fig. 4.4 shows how the loaded images appear on the Workspace pane. Once the images are loaded, they are aligned. (through the Workflow menu). At this stage, one arrives at a sparse point cloud model based on estimated camera position and orientation for each image. The sparse point cloud along with the computed

camera positions is displayed in the console as seen in Fig. 4.4.

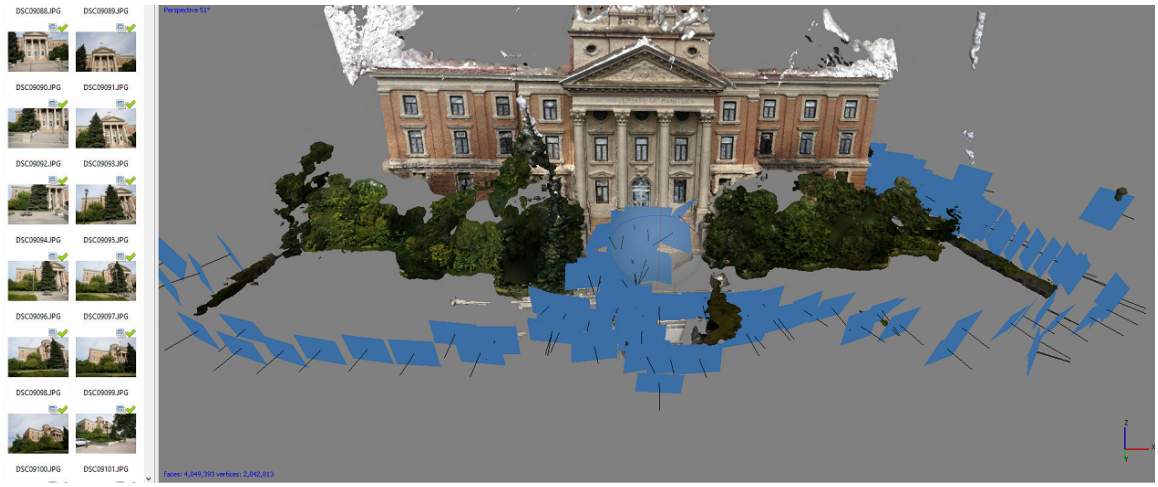


Figure 4.4: Sparse point cloud with estimated camera positions

It was observed that images of poor quality had the potential to be of significant detriment to the alignment process. For such cases automatic image quality estimation may be made based on the features [32]. Ultimately the goal is to preserve images that contain fixations to avoid losing any data (images with less than 50% of keypoints exceeding a minimum quality threshold are rejected). Adjustment of capture settings (e.g. level of light, resolution of video) can help to achieve better features for matching. One can also adjust parameters to ensure data retention and successful matching. This process is controlled by alignment parameters including thresholds for accuracy, pair preselection and setting limits on key points or tie points. In our experimentation, it was found that default settings for pair pre-selection, and a Key point limit of 40,000 and Tie point limit of 4000 [32] produced the best results.

4.1.2 Dense Point Cloud Generation

Following construction of the sparse point cloud in Fig. 4.5, this representation can be used with further processing to derive a dense point cloud as shown in Fig. 4.6. Although the specific relationship between these representations is obfuscated by not having access to source code, this looks to be very reminiscent (if not the same) as the relationship between CMVS and PMVS discussed in Chapter 2. Based on the estimated camera positions from the alignment process, depth information is available for each camera, and further refinement with confidence derived from the sparse cloud allows a single dense point cloud to be produced [66]. This process is controlled by parameters that reflect desired mesh quality and depth filtering nodes. Poor quality images may lead to outliers among points. Different thresholds for outlier rejection may be set ranging from mild to agressive rejection of outliers.

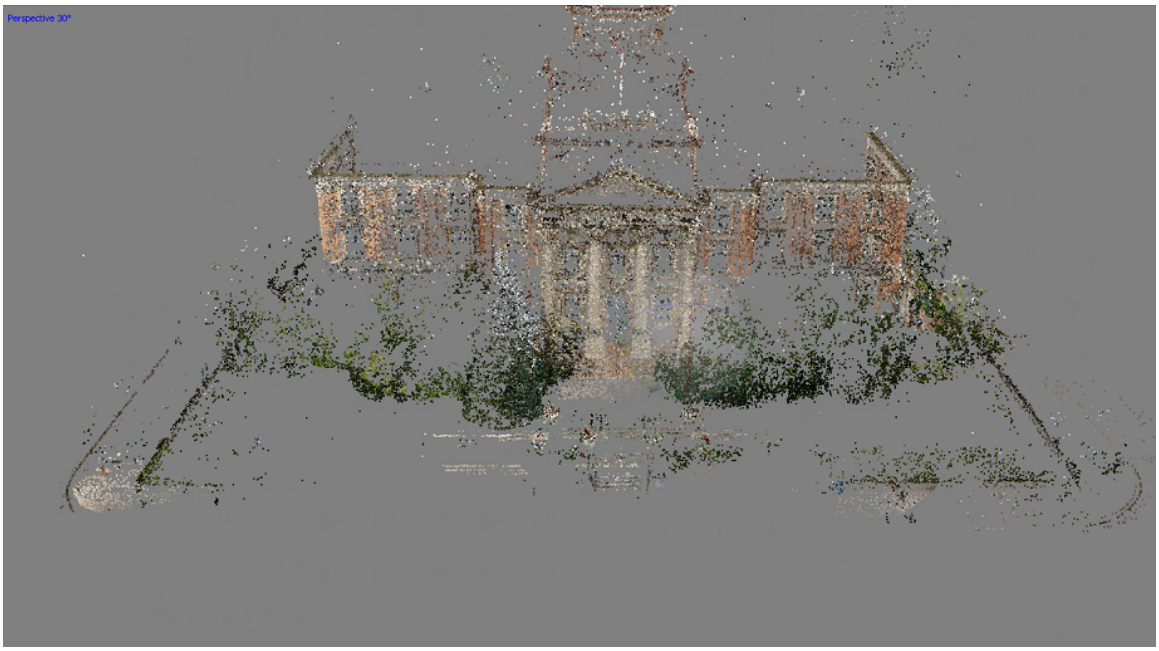


Figure 4.5: Point cloud formation

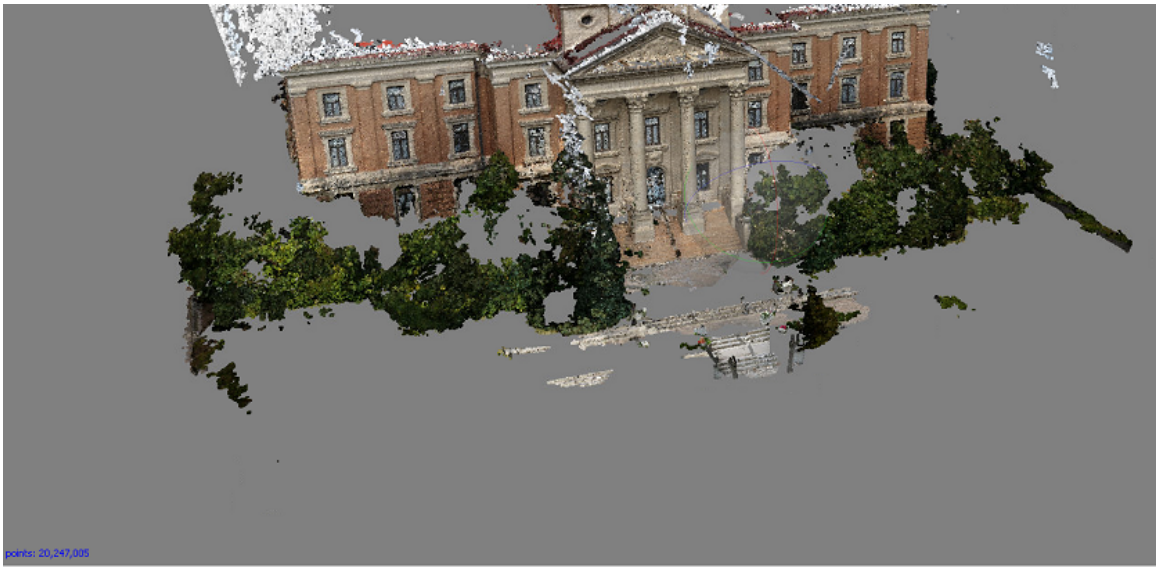


Figure 4.6: Further processing of the sparse point cloud produces a dense point cloud.

4.1.3 Building a Mesh

While a dense point cloud provides a perceptible representation of the scene, it has no surface. Ultimately having a model with a surface is critical to allowing gaze density to be assigned to the model, or for gaze vectors to intersect with the model. It is therefore desirable to convert the dense point cloud to a 3D polygonal mesh in Fig. 4.7. Reconstruction parameters include the surface type, desired polygon count, type of interpolation and point classes. Given sufficiently dense point clouds based on the volume of our data, we did not include interpolation since this has little effect on areas with dense point samples, and risks artifacts in areas of poor sampling. With a generated mesh certain corrections may be applied using mesh editing tools such as decimation, closing of holes, automatic filtering, removing detached components, smoothing, etc. This can allow unwanted portions of the model to be removed. Through experimentation, it was found that mesh decimation was useful, since the

number of polygons resulting from a dense point cloud far exceeds what is required to accurately represent the scene geometry. In fact, there is virtually no loss of precision in the geometry even in using a fraction of the original number of triangles (e.g. 10%). This also allows for any further processing on the mesh, including gaze related processing to be done in a manner that is not overly cumbersome in terms of computation required.

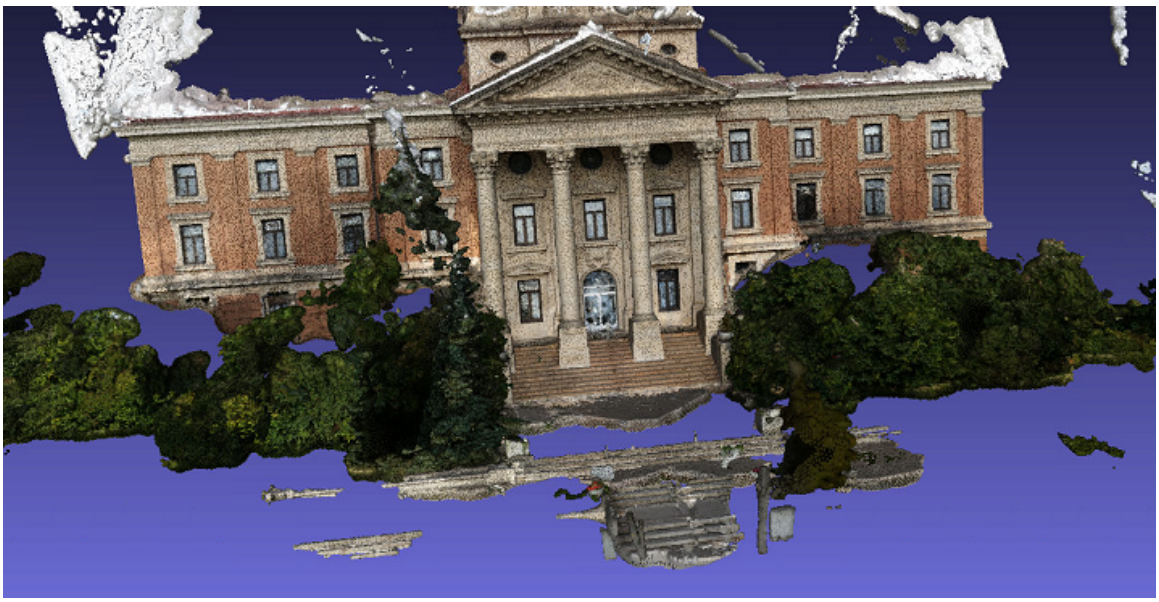


Figure 4.7: 3d Mesh formation from the dense point cloud

4.2 Methods: Gaze Capture

Given a high quality model produced by the scene camera (based on the steps discussed in the previous section), we hope to relate the gaze data captured to the 3D model in question. Estimated camera pose forms part of the reconstruction process, and is therefore retained and exported as shown in Fig. 4.8. In our workflow, we export

this in the bundler file format which allows for all data to be saved and retained if needed (including the sparse point cloud data). Orientation parameters are exported to a Chan file format in ASCII format with parameters saved in columns (one column per parameter, and one line per frame). The parameters saved from the reconstruction process to this file include: $framenumber$, $location_x$, $location_y$, $location_z$, $rotation_y$, $rotation_z$, $angle_y$. The rotation values are expressed in Euler angles and the last column i.e. $angle_y$ represents vertical field of view. The gaze dataset obtained from the scene video consists of frames with and without gaze. The notion of a "Fixation" in 3D space has much more complexity than for 2D screen space. This is true of the ETG2 Glasses, or any other head-mounted eye tracking system because much of what looks like a fixation is actually smooth pursuit, meaning the eyes are tracking a stationary object while the head moves. For this reason, we divided our dataset into two types:one with the original gaze positions exported from the scene video (this is essentially a head-centric representation) and the other consists of frames with gaze correction i.e. applying an algorithm to get the correct positions of gaze corresponding to where our eyes are looking at.

Given a number of image frames, and our 3D model, each frame has a corresponding camera position and rotation values which are used. In the context of 2D gaze analysis, often a heatmap or gaze density map is created by blurring discrete fixation points with a Gaussian blur that reflects the dropoff in acuity from the fovea moving to the periphery. We seek to accomplish the same effect, albeit in a 3D space. With that said, this is a much more complicated process in 3D.

To model this correctly, one has to effectively model the degree of sampling with

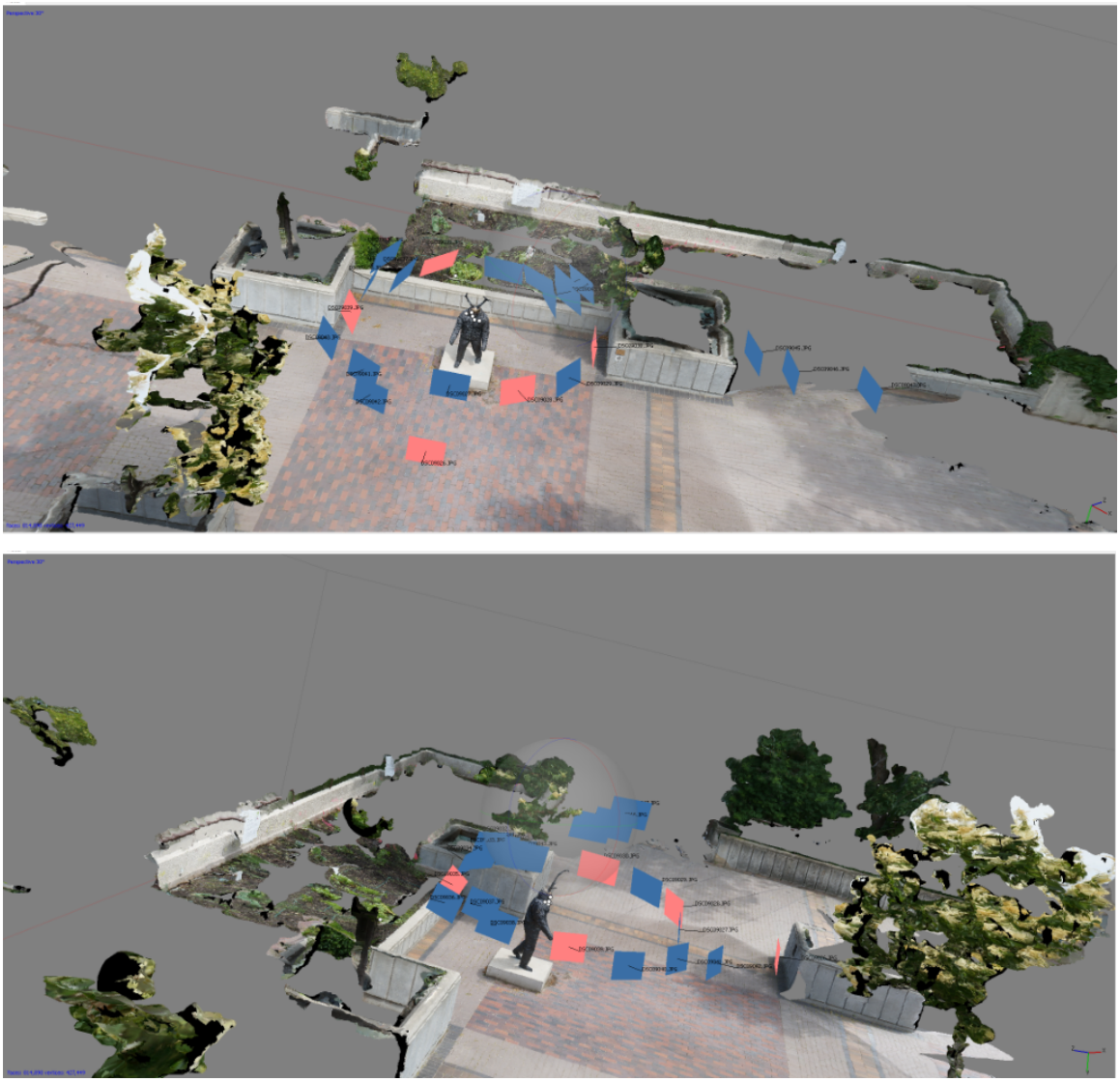


Figure 4.8: Different views of a 3D model with the aligned cameras.

a Gaussian drop-off [67] from a camera directed according to the viewing direction in a manner akin to ray casting. This provides the foundation for visualizing the gaze data. To achieve efficiency in implementing this representation, it is natural to use 3D modeling software that allows for custom scripting to be written, and we have used the Maya modeling software [6] to implement ray-casting from points of gaze onto

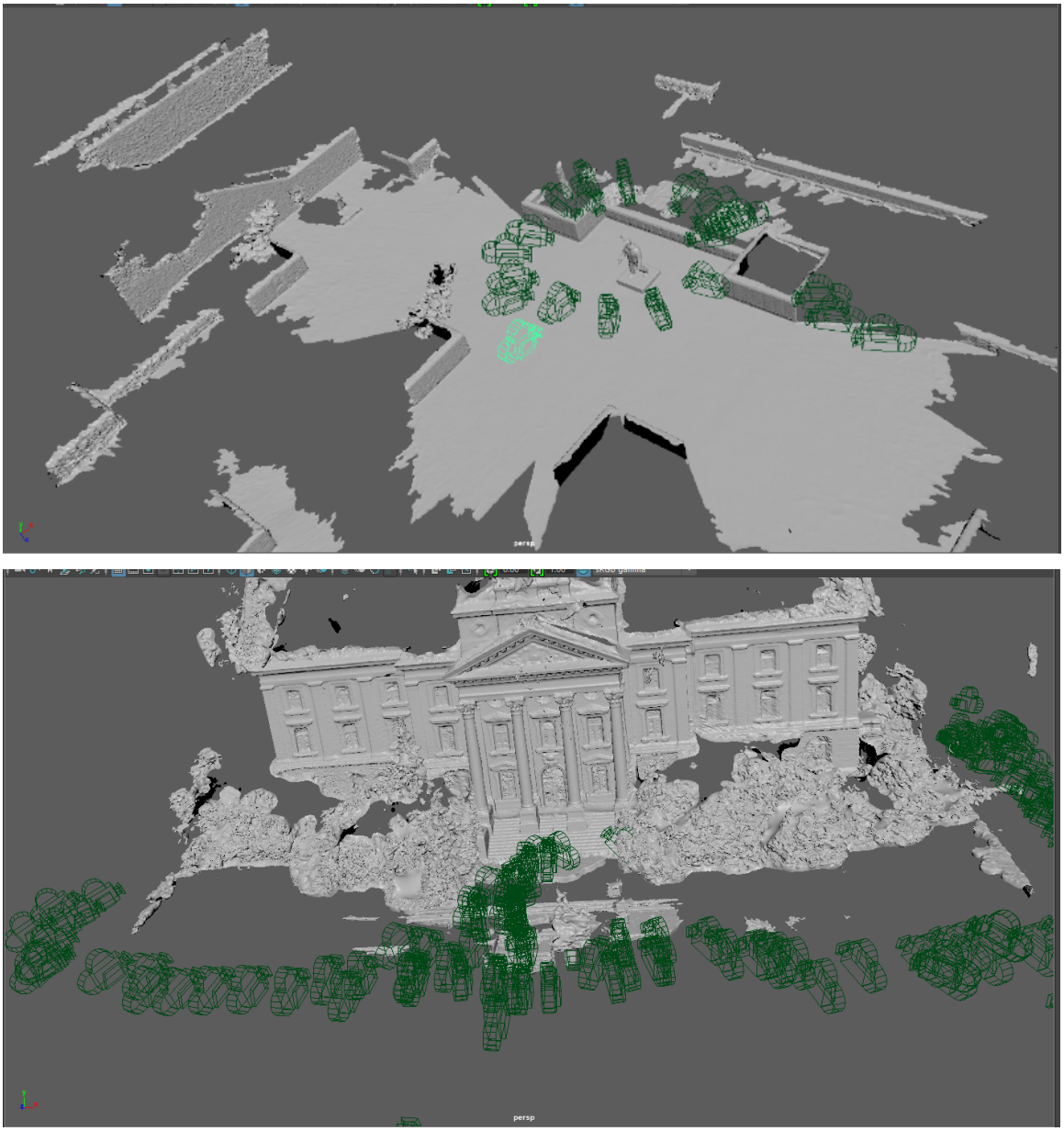


Figure 4.9: 3D models imported in Maya [6] to implement ray-casting.

the model that mimics the peripheral drop-off in visual acuity. Fig. 4.9 shows the imported models in Maya. The most popular software for rendering in the Computer Graphics industry are 3ds Max and Maya.

To create a visualization of the gaze data, we opted for ray tracing as shown in

Fig. 4.10 using a spotlight model and rendering the 3D scene using Mental Ray [68] (without Global Illumination since we don't want indirect lighting in our density). To use Mental Ray for Maya renderer involves preparatory setup (installing the mental ray plug-in, loading via the Plug-in-Manager: selecting loaded and autoload options for the "Mayatomr.mll plug-in"). The reason for applying the mental ray add-on for rendering is that this allows for masking of the spotlight. As mentioned, we don't want the density from viewpoint to reflect the drop-off in visual acuity from the centre of gaze. This can be accomplished by shining the spotlight through a mask aligned with the imaging plane of the camera that has maximum transparency at the centre, and becomes increasingly opaque moving peripherally. The dropoff in transparancy follows a 2D Gaussian distribution with a σ corresponding to 1 degree visual angle.

Maya includes more customization, which makes it a more powerful application for our purposes. MEL or Python are the scripting languages used in Maya to customize its core functionality as the commands and tools being used are written in these languages. MEL is considered user since all the interactions between users and Maya gets recorded in MEL, this makes it easy for the novice users to implement subroutines [69]. Another application we used for visualizing and processing three-dimensional mesh representations is MeshLab [70]. It is a free Windows, Linux and OS X application for simplifying, processing and converting large three-dimensional surface meshes to or from a various 3D model formats. Many standard methods for decimation, remeshing, or custom shaders can also be applied in MeshLab, if *post hoc* refinement of the model is desired.

4.3 Methods: Relating Gaze Data to Models

In the previous section, we alluded to some of the methods that are used to relate gaze data to captured 3D models. In this section we further elaborate on this relationship, while also including additional discussion of the rudiments of how this data may be visualized. Collections of data from an ensemble of observers that include factors such as fixation positions and duration are commonly collected and to investigate human gaze behavior. Visualization of gaze datasets helps in understanding and interpreting such data, and may be easier to interpret for trends than examining fixations, or sequences of fixations from single observers. In the domain of gaze analysis where two-dimensional viewing is concerned (e.g. viewing images), common representations include scan paths, heatmaps, attentional maps, etc. We aim to provide similar representations to relate our gaze data to the 3D model for qualitative and quantitative analysis. In implementing gaze visualization, the 3D model is imported into 3D rendering software, followed by running a python script that loads all relevant data. This places a spotlights at a camera position consistent with each view from the outward facing scene camera. The masked spotlights then project an amount of light that corresponds to the sampling acuity, and the ensemble of spotlights represents a density. The degree of light received by each triangle in the 3D model then corresponds to the degree of visual sampling by an observer or multiple observers. An example of this is shown in Fig. 4.10

Creation of a spotlight is applied to all camera positions and applied to all exported cameras as illustrated in Fig. 4.11 where cameras are replaced by spotlights.

4.3.1 Scene Camera Images (Head movements)

The base parameters of the spotlights are as follows: Cone Angle=160, Dropoff=0, Penumbra Angle=0, Intensity=0.05, and Decay Rate= Quadratic. This implies a viewing angle of the observer that covers 160 degrees visual angle. There is no dropoff of the spotlight, since this is provided by the transparency mask in the scene camera imaging plane. Intensity is adjusted to a level where the total number of cameras does not cause the intensity on the model to saturate. That is, if too high an intensity is chosen, the model can appear completely white, so the intensity needs to be adjusted in an inverse relationship with the number of fixations.

The entire process of spotlight generation and masking is produced by a python script in Maya. The script makes use of extant camera orientation values from the reconstruction stage (chan file), and creates spotlights using Maya commands while setting appropriate translate(x,y,z) and rotation(x,y,z) attributes of each spotlight to the position and rotation parameters of the cameras. Fig. 4.12 and Fig. 4.13 represent the rendered images of the 3D models. This gives an initial understanding of the gaze behavior when mapped on a model.

4.3.2 Algorithm for Gaze Correction

In the above samples, density is attributed in a head-centric reference frame. This is equivalent to an observer that is always looking perfectly straight ahead. However in reality, the eyes are pointing in a different direction independent of the position of the head. To get a true measure of acuity, placement of the spotlights needs to reflect a correction for the viewers direction of gaze. The eye tracking system involves two

cameras for tracking an observer's natural gaze behavior in real time. The SMI eye tracker [54] uses two cameras to track a viewers eyes from close proximity i.e. tracking the pupil and corneal reflection, and an additional high definition scene camera, records a sample of the real-world scene. The scene camera provides a horizontal viewing angle of 60 degrees from a 1/3" sensor with a 36mm lens. It offers a resolution of 1280x960 pixels at 60Hz. The scene camera of the SMI glasses provides the head centric view, while mapped gaze points(see Fig. 4.14) in the coordinate space of the scene's camera image were fetched in real-time from the BeGaze analysis software [5]. This implies that for gaze centric fixation density, spotlight pose needs to be adjusted to reflect the offset of the gaze in scene camera coordinates from the center of gaze. The angular extent of the scene which is imaged by the scene camera is termed the camera field of view or the camera angle of view [71]. The camera's FOV (field of view) depends on the lens of the camera as well as position of sensors and optics, and can be measured horizontally, vertically, or diagonally. This angle can be increased by using lenses with less than 2mm focal length but this can result in loss of accuracy and lowered resolution. Note that the human's field of view is larger than that of the camera. This is one place where a 3D representation can present an advantage. If a viewer is looking far to the left or right, the 3D model still receives the correct amount of light, consistent with the viewers field of view, even if they are looking at something outside of the frame of the scene camera.

The process for this correction is as follows: The displacement from the fixation is represented as Gx and Gy in a horizontal and vertical directions respectively. The viewing angle of the scene camera is 60 degrees horizontally hence 60/2=30 degrees

was the angle between the focal length of the scene camera represented as f and horizontal displacement Gx from the gaze point. Similarly, for y-axis the viewing angle of the scene camera was 46 degrees, hence, the angle between focal length and vertical displacement Gy was 23 degrees.

Equations for horizontal and vertical displacement:

$$G_x = (W - X), \qquad W = \frac{width of the scene camera image}{2} X = x coordinate of Gaze point \tag{4.1}$$

$$G_y = (H - Y), \qquad H = \frac{Height of the scene camera image}{2} Y = y coordinate of Gaze point \tag{4.2}$$

New Rotation Angles, R1 and R2

$$R1 = atand((Gx/W) * tan(30^\circ) \tag{4.3}$$

$$R2 = atand((Gy/H) * tand(23^\circ) \tag{4.4}$$

New Rotation Matrices (R_x and R_y) for each camera

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ \cos(d)(R2) & 0 & \sin(d)(R2) \\ 0 & \sin(d)(R1) & \cos(d)(R1) \end{bmatrix} \tag{4.5}$$

$$R_y = \begin{bmatrix} \cos(d)(R2) & 0 & \sin(d)(R2) \\ 0 & 1 & 0 \\ -\sin(d)(R2) & 0 & \cos(d)(R2) \end{bmatrix} \tag{4.6}$$

$$R(i) = R_x \times R_y, \qquad i = numberofcameras \tag{4.7}$$

The last step produces the correct rotation values(x,y,z) for each camera consistent with how the camera is oriented. The overall result of this is that all scene camera orientations are altered to reflect the direction an observer is looking, and the density conforms to an estimate of sampling from an observer.

4.3.3 Images with Gaze Correction(eye movements)

The following figures provide a comparison between images of a model when visualized in a head-centric reference frame(Fig. 4.15a and Fig. 4.16a) with the images in a gaze-centric reference frame(Fig. 4.15b and Fig. 4.16b). Although gaze is more evenly distributed on the target areas in the head centric case, with the gaze correction, we can clearly see the fixations are confined more to areas of interest in the scene. There is some similarity owing to center bias in gaze, and these results suggest that even a head mounted camera, absent of eye tracking might provide a reasonable proxy for point of gaze, provided a large enough sample size.

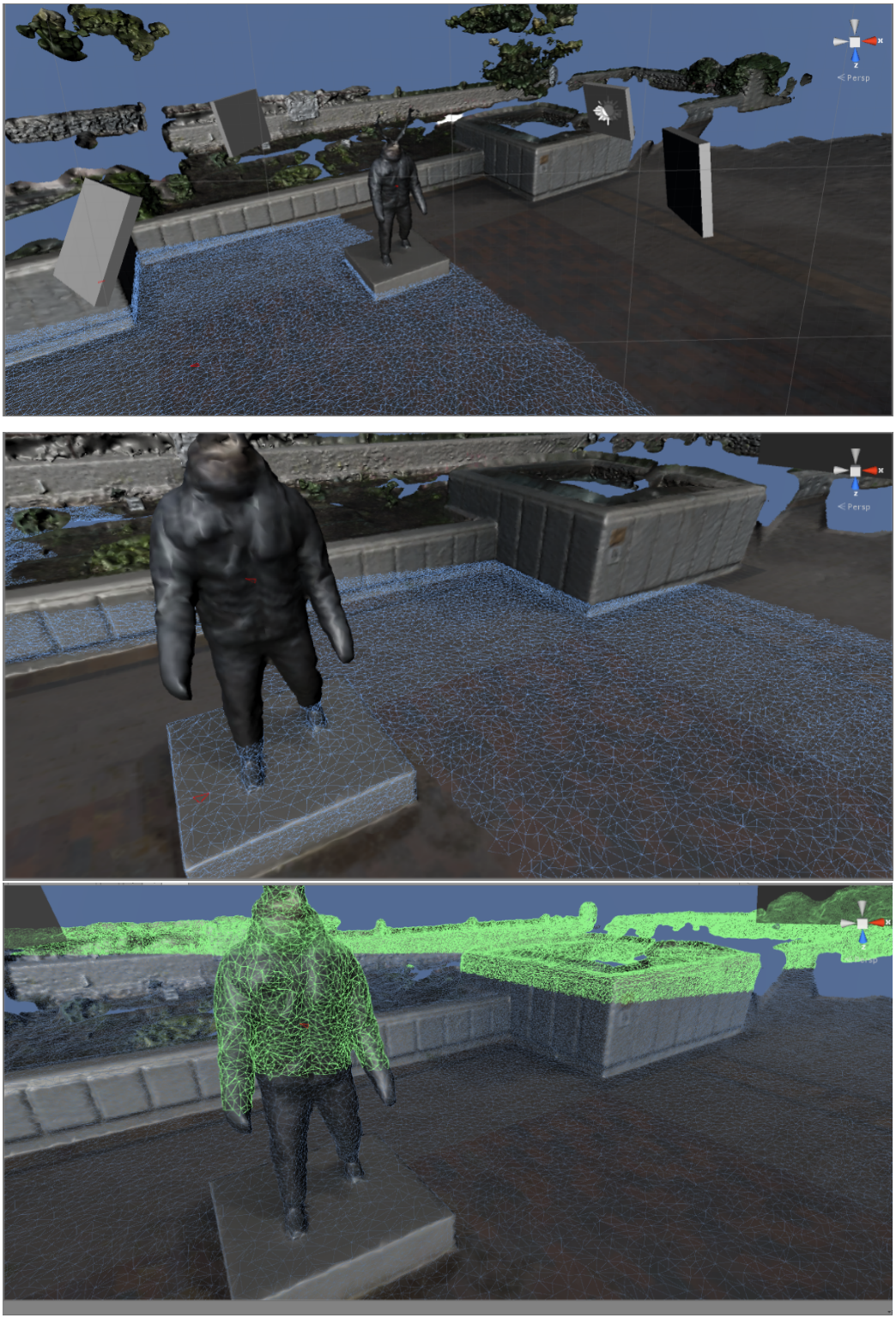


Figure 4.10: Ray Tracing to locate triangles in a 3D mesh which are illuminated in a spotlight model to visualize the gaze data. The degree of light received by each triangle in the 3D model corresponds to the degree of visual sampling by an observer.

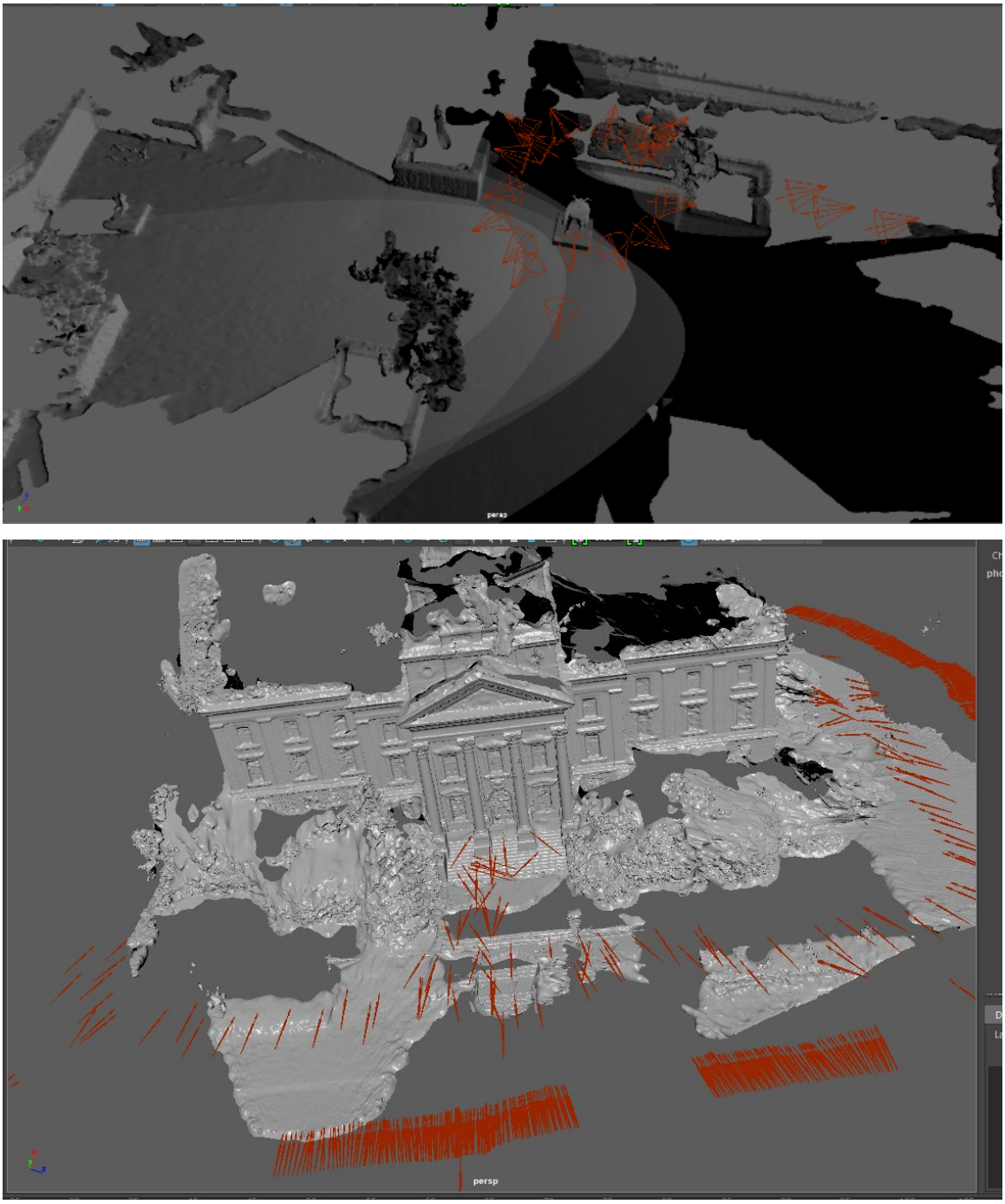


Figure 4.11: Models with cameras replaced by spotlights for visualizing the gaze points.

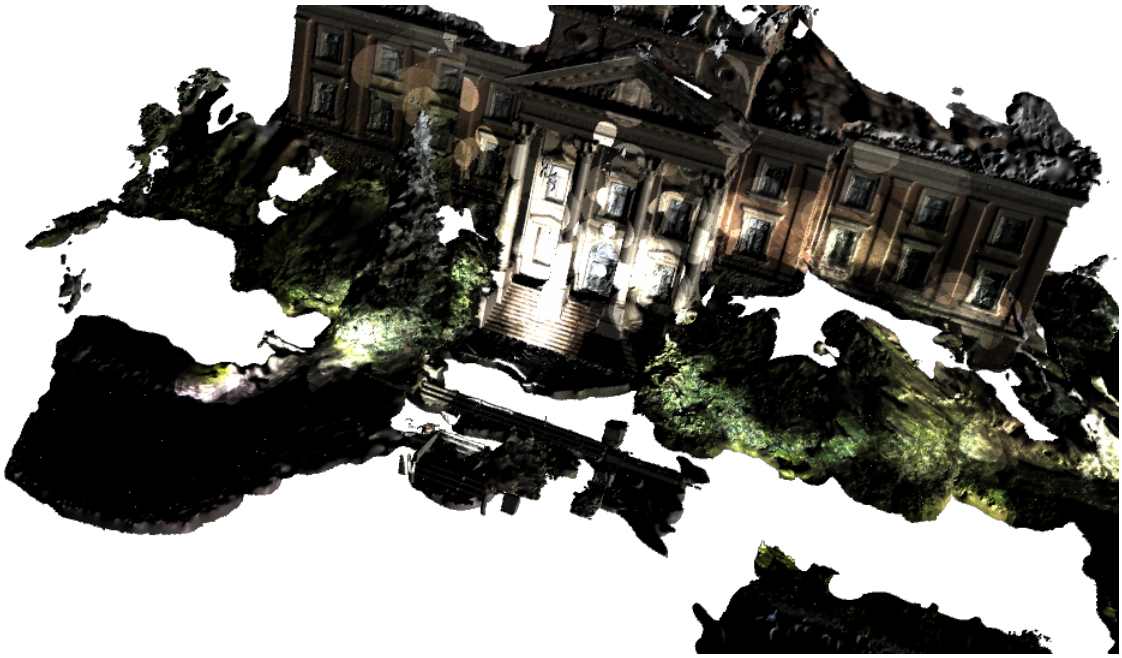


Figure 4.12: Head-centric fixations visualized on the 3D model of a building

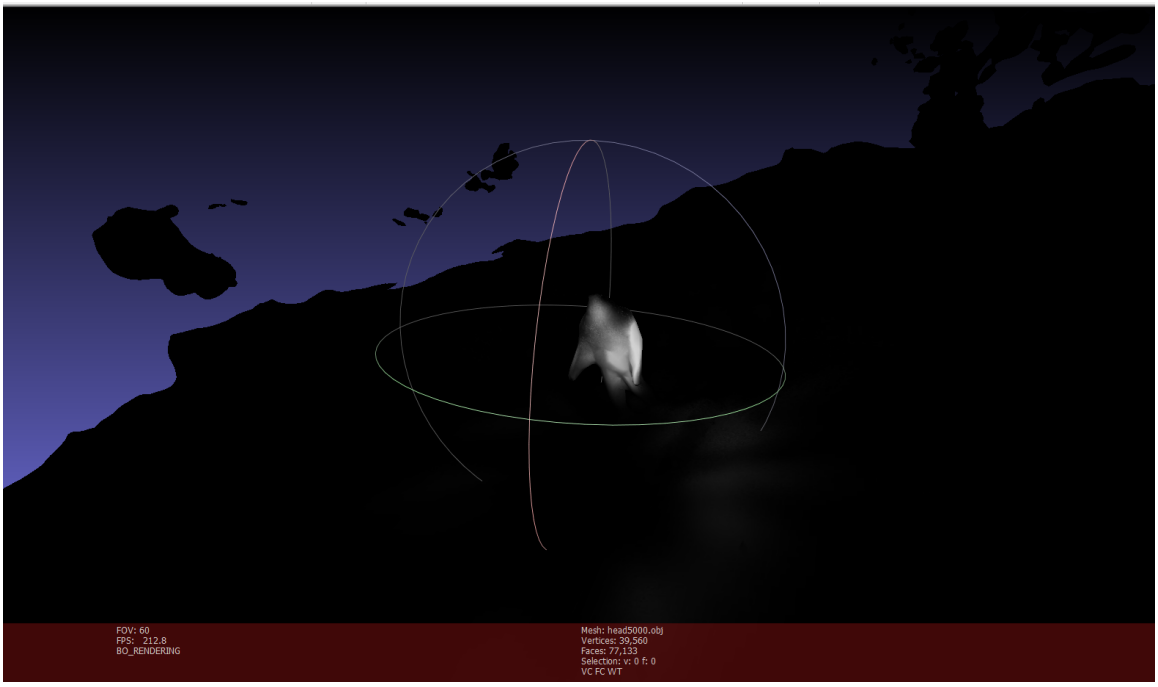


Figure 4.13: Another example of Head-centric fixations visualized on the 3D model of a statue.

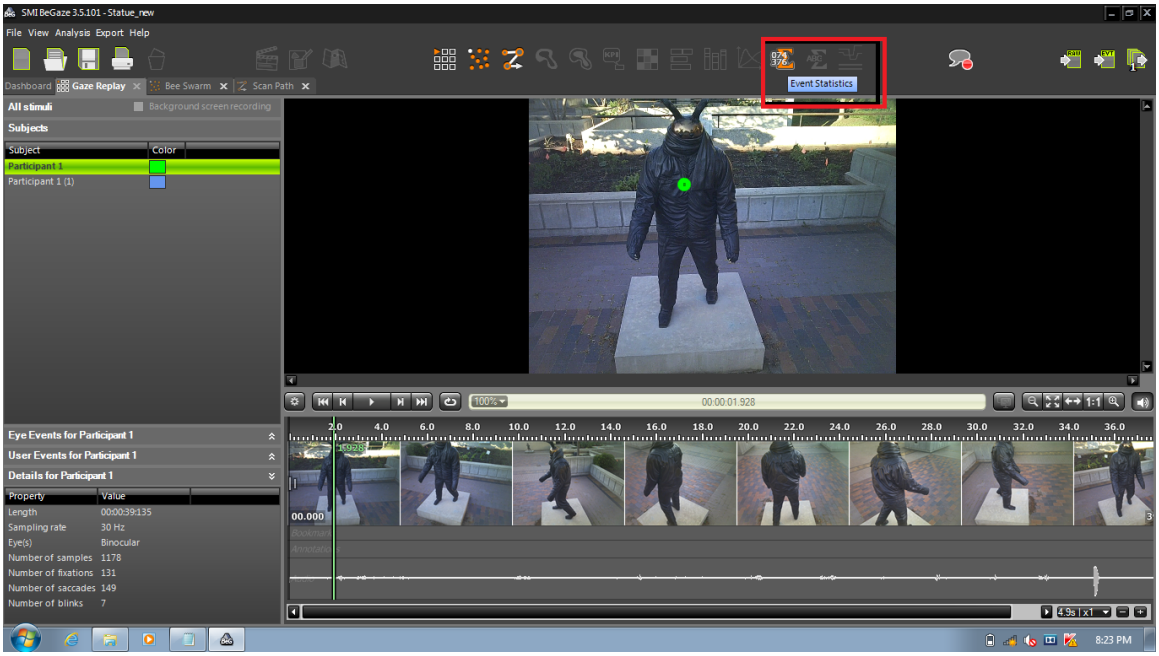


Figure 4.14: The mapped gaze points in the scene camera image are fetched using BeGaze analysis software [5]. Details of these gaze points are obtained by exporting the Event Statistics file from the software.

(a) The figure demonstrates a view when the model is visualized in a head-centric reference frame. The gaze is evenly distributed on the observed statue and the surrounding areas.

(b) The figure demonstrates a view when the model is visualized in a gaze-centric reference frame. With the gaze correction, the fixations seem to be confined more to the area of interest which is the statue in this case.

(a) The model of a building visualized in a head-centric reference frame with evenly distributed gaze.

(b) The model of a building visualized in a gaze-centric reference frame with the fixations on the areas of interest, which is the entrance in this case.

4.4 Conclusions

In this chapter we have presented details on how the methods that form the contribution of this thesis have been implemented. We have discussed different options for 3D reconstruction, and what methods and parameter choices ultimately proved most successful. We have also presented a workflow that goes from raw gaze data and scene camera imagery to rich 3D models with superimposed head-centric or eye-centric gaze density maps. This provides a foundation for more targeted experimental work or applications in the domain of studying real-world eye tracking.

Chapter 5

Data Visualization and Quantitative Analysis

5.1 Motivation

Data Visualization can help to understand the significance of analyzed data, or trends in data by presenting it in a visual context. This can play an important role in transforming complex data into a form where it may more readily inform on decision making, or drawing conclusions about behavioural trends.

Since much of the existing analysis of eye tracking data has happened in highly controlled environments (e.g. in a research lab) with rigid experimental setups and often having a seated viewer, methods are needed to address visualization and quantitative analysis of data in the form created by methods presented in the preceding chapter.

There is a rich body of existing work targeting visualization, and quantitative

analysis of gaze data when it is associated with a 2D representation such as an image, video, poster or supermarket shelf. Rather than create arbitrary or *ad hoc* representations for data analysis, a natural way to proceed is to examine how the existing set of prevailing methods can be transformed into a form that makes them amenable to analysis in the context of 3D models.

There is merit in considering how constraints on the study of gaze behavior may be further relaxed through technical developments to allow for observation and modeling of behavior in natural settings. The contents of this chapter establish how traditional methods for qualitative and quantitative analysis of gaze data can be transferred to considering 3D models.

5.2 Visualization

Following any data collection from an eye tracker, the data is typically analyzed using statistical methods and visualization techniques, which reveal characteristics of gaze data such as fixations, saccades, fixation densities (heatmaps). The visualization techniques can be categorized on the basis of the properties of gaze data and observer characteristics. These techniques help in exploring and analyzing different aspects of the recorded gaze data in a qualitative way. Blascheck et al. [72] discussed three classes of visualization techniques: point-based, AOI-based, and techniques using both. Statistical diagrams such as scatter plots, bar charts or line charts are commonly used techniques to visualize base statistics of gaze data set. This may include the degree of attention paid to a defined area of interest (AOI), or distributions of saccade distances between successive fixations.

Atkins et al. [73] used line charts to present x- and y- coordinates of recorded fixations. Bar charts were used by many authors to display a histogram of eye movement metrics such as fixation duration or saccadic amplitude. Brasel et al. [74] used a 3D bar chart to show the distribution of fixations in a TV viewing user experiment. Berg et al. [75] used scatter plots to compare saccadic amplitude and velocity between humans and macaques. Another type of statistical methods are box plots [76] and star plots [77] to analyze and display the statistical distribution of the properties of eye movements.

These all present useful characterizations of the base statistics or properties of eye movements that are especially relevant in experimental psychology where establishing significant differences across conditions or populations is the objective.

At a fundamental level, a powerful representation is the heatmap that represents the likelihood that a particular location in the image is fixated upon, or analogous to this, a representation of sampling density based on spatial smoothing of raw gaze data.

2D visualizations can represent both spatial and temporal data in this form, revealing focus of attention, and how it evolves over time. In the case of 3D data representative of real environments, it is natural to also consider representing this data in the form of sampling density or spatial heatmaps. However, in doing so there are different choices that may be made in the specifics of how to represent the focus of gaze in the form of a visualization. This includes both factors related to observations on how humans allocate attention, but also properties of the visualizations themselves (e.g. colormap properties).

In what follows, we describe how traditional methods for 2D gaze visualization may be used for 3D models, and also address factors related to both how data is interpreted, and how it is visualized or presented.

5.2.1 Methods and Examples

The following discusses two distinct approaches to visualization of heatmaps corresponding to 3D gaze data, and their interpretation in reference to psychological and biologically inspired factors in how to interpret the data. We also address the issue of how to display gaze densities in a way that is most readily interpreted by a human observer in paying careful consideration to characteristics of the colourmaps that are used. This is a factor that is typically given little consideration, but is important to qualitative impression produced by the models when interpreted by a human. It also follows that considerations relating to colourmaps are also relevant to 2D heatmaps, and are deserving of wider consideration by the research community.

Colourmaps and Perception

Many visualizations represent the density of gaze points or fixations as heatmaps. In a simple representation, one might choose to represent a high density of fixations using white, no fixations as black, and intermediate grades of gaze density as shades of gray. However, it is also common to use combinations of colors, which may vary from cold to hot, or follow the spectrum of visible light. One consideration that is not given a great deal of attention, is the relationship between how these colors are perceived, and the numerical measure of density. For example, the perceptual distance between

colours does not necessarily match the quantitative distance in density. This is a problem that has been recognized and addressed through carefully chosen colormaps [78]. However, despite the fact that more contrast may be readily observable from colour, it is not common practice to make sure that perceptual distances between colours adhere to quantitative differences in density. For this reason, 2-dimensional heatmaps would benefit from exercising more care in colourmaps chosen. Colourmaps in the context of our 3D models provide a useful way of characterizing the distribution of attention over models, and represent focal points in a scene. The visualizations presented in this thesis were based on free exploration of a few environments by a single observer. Note that the intent is not to understand behaviour, or conduct a behavioural study, but rather to provide a minimal set of data for demonstrative purposes of the methods presented in the thesis.

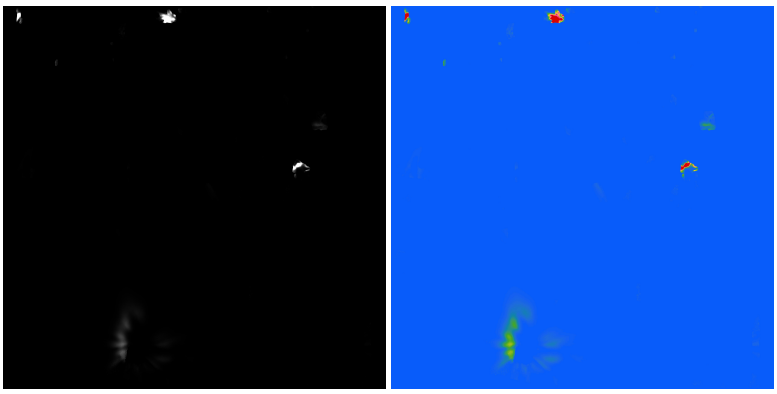


Figure 5.1: Left: The texture map from the original model representing fixations. Right: Perceptually uniform color map with the assigned colors to the fixation density depicting the distribution of attention.

In what follows, we present visualizations of gaze density (or attentional density)

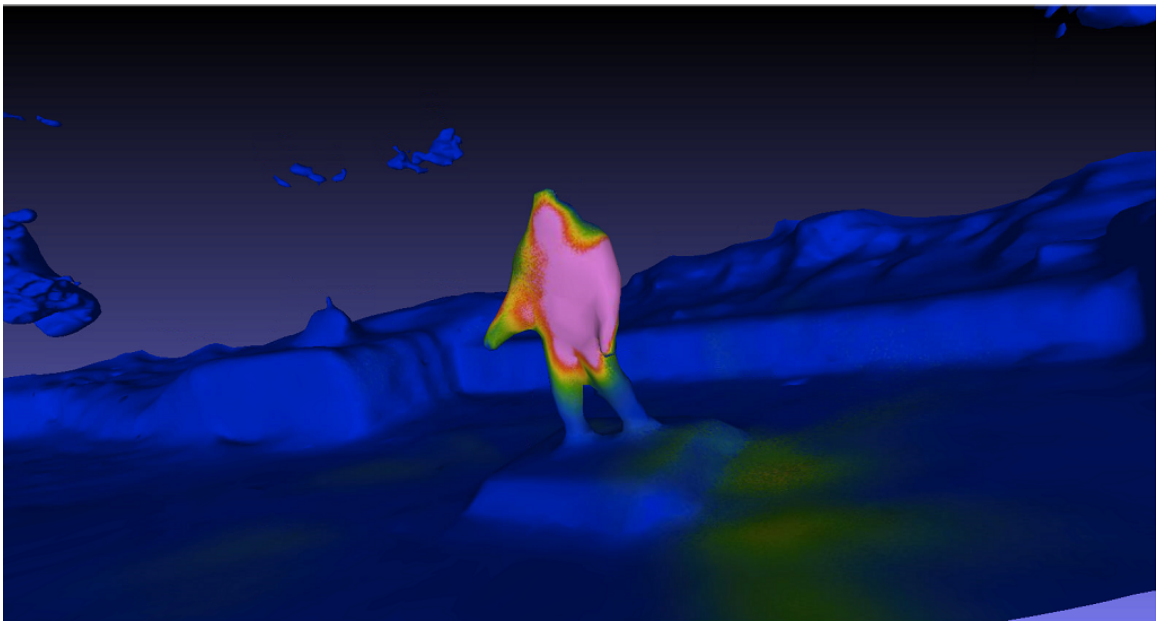


Figure 5.2a: Rainbow color map R1 applied as a texture to the 3D model

represented by coloured heatmaps that adhere to the direct mapping between perceptual and quantitative differences we have prescribed. These are based on Kovesi's colourmaps, which have a uniform perceptual contrast over their entire range [78]. This includes a function that offers different types of colormaps such as linear maps, diverging maps, cyclic maps, rainbow maps(Fig. 5.2a, Fig. 5.2b and Fig. 5.2c) and insoluminant maps. We use rainbow maps and apply these as a density derived texture as shown in Fig. 5.1 to the 3D models. Two different types of visualization are proposed, and the details and motivation for these visualizations are described in what follows.

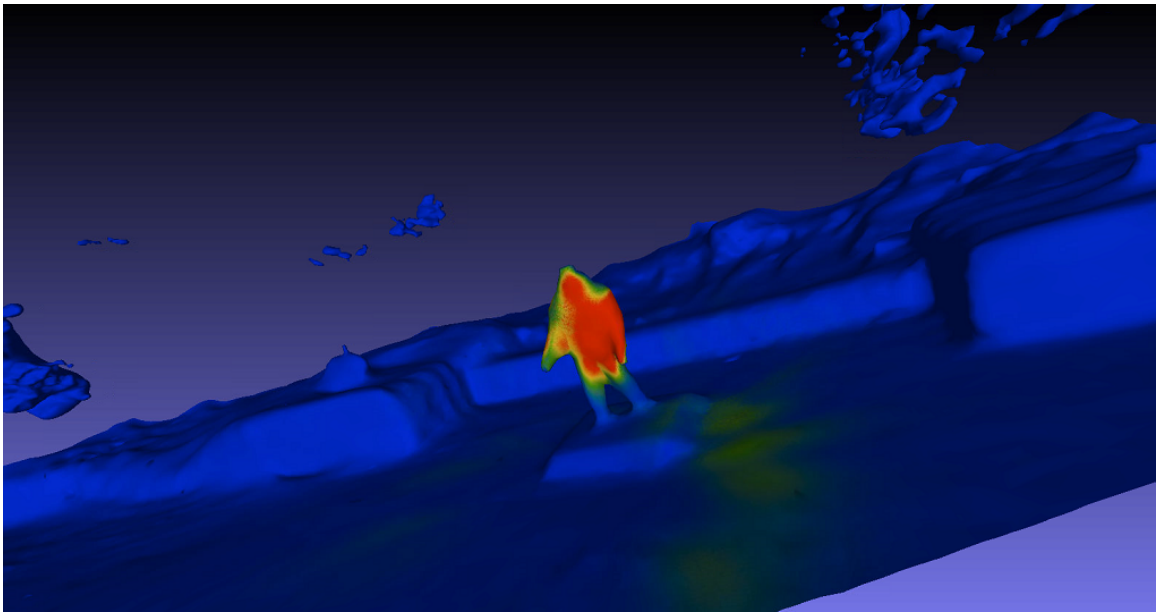


Figure 5.2b: Rainbow color map R2

Acuity Based Heatmaps

The essential elements for Acuity Based Heatmaps have been described in Chapter 4. These are derived from either a face-centric or gaze-centric spotlight directed through a mask(Fig. 5.3a and Fig. 5.3b) that varies in opacity so that the amount of light shone on the model corresponds to the density of gaze points. The PLY (Polygon file format) [79] format is used for storing graphical objects such as in the form of a polygonal mesh. The gaze density may be retained in the form of a texture, or values transferred directly to vertices themselves (a process known as baking). Gaze density is then visualized by assigning colours to the triangular mesh that represents fixation density according to the associated colourmap.

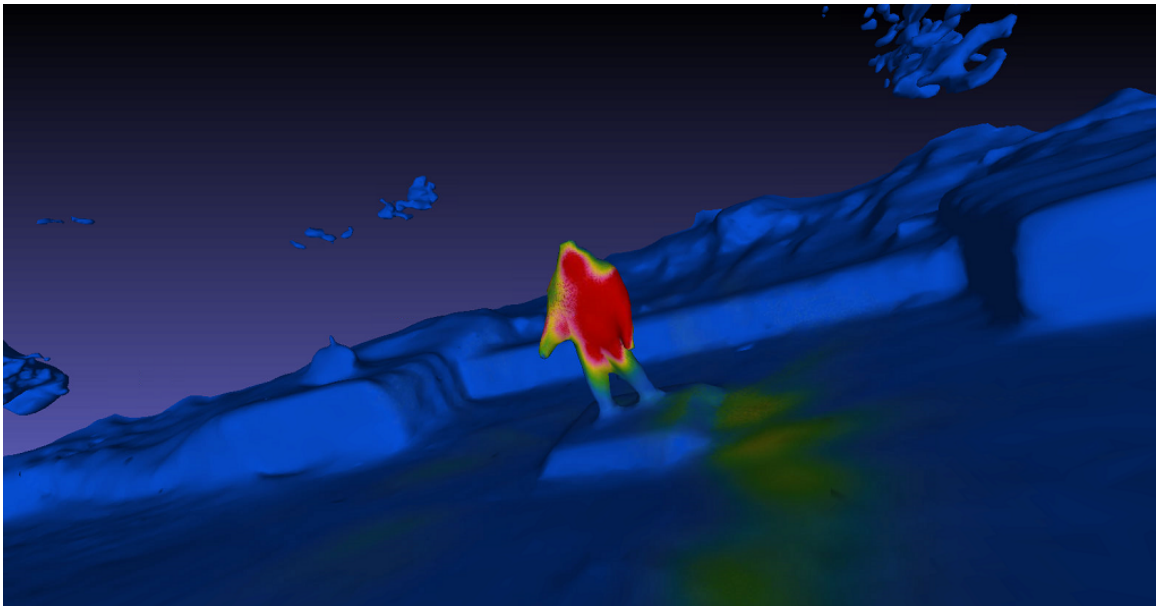
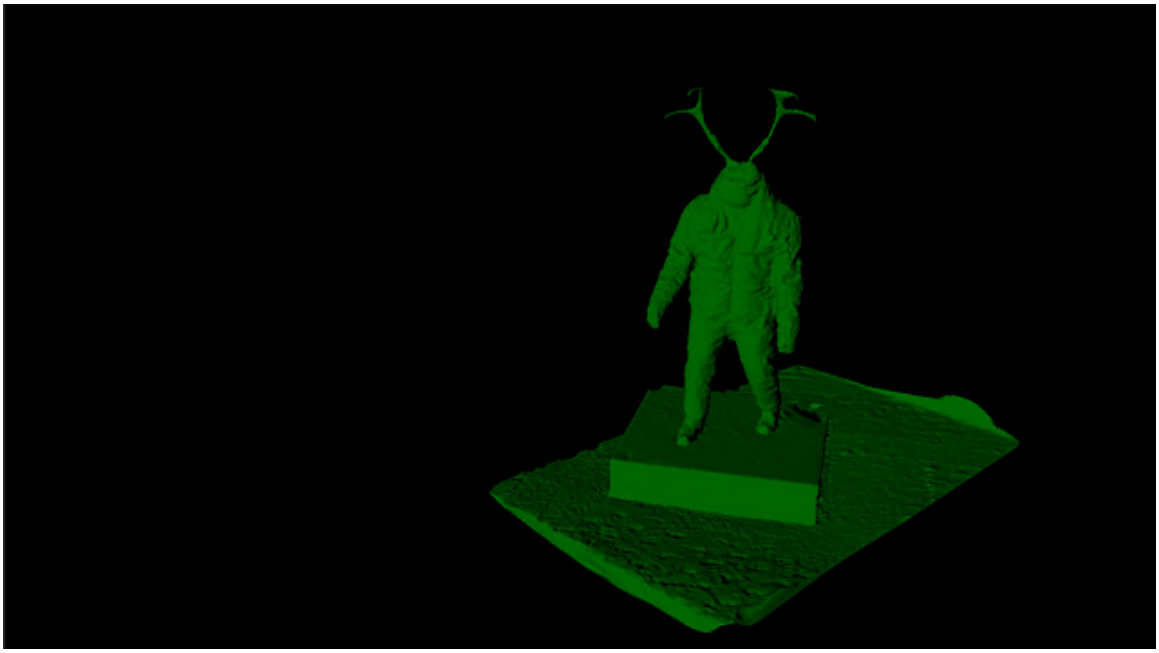


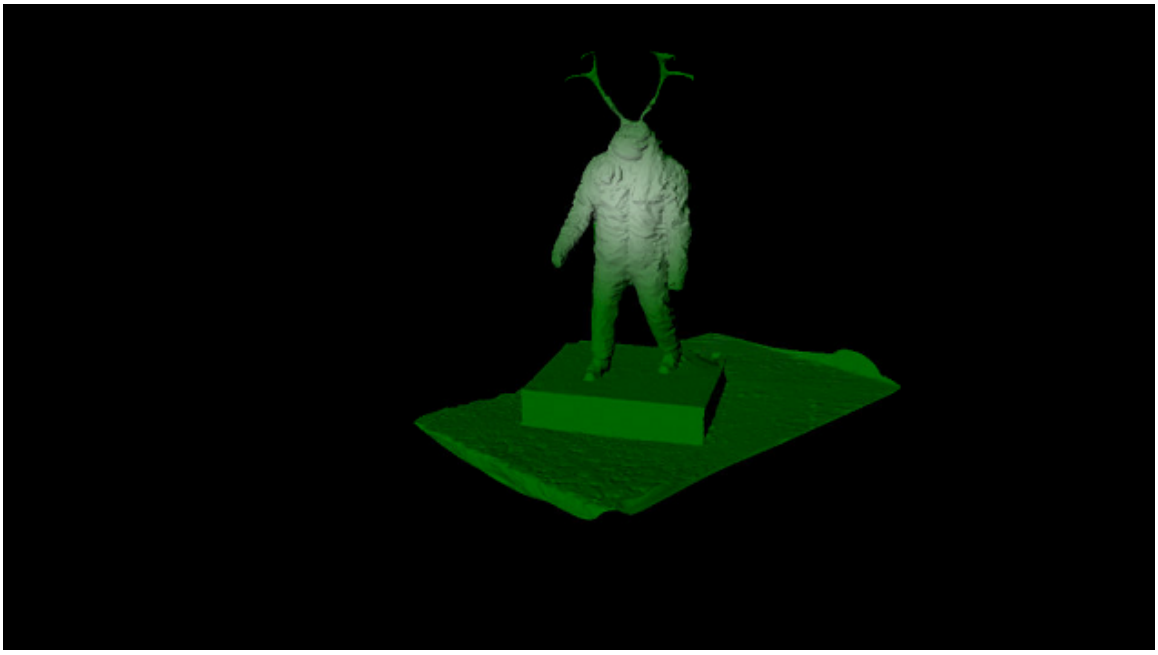
Figure 5.2c: Rainbow color map R3

Frame of Reference

Eye-head coordination is involved while locating gaze at the visual target. As mentioned, it is possible to correct for direction of gaze, or to consider sampling in head-centric coordinates. Although the intent at the onset of this work was to consider only gaze directed sampling, the head-centric densities are also quite interesting, and provide a useful reference for comparison. Eye and head movements usually take place in the same direction [80] but sometimes during visual search tasks such as reading the eyes might move in opposite direction of the head. There is also a known tendency to fixate in the centre of an image, or the visual field [81] which yields some similarity between the head-centric and gaze centric densities. This is another example of an observation that would not be possible to make in the context of traditional methods in eye tracking research.



(a) The textured 3D model viewed in a 3D modelling software. [6]



(b) Gaze-centric spotlight directed on the statue through a mask such that the the amount of light shone corresponds to the density of gaze points.

Surface Based Heatmaps

A alternative interpretation of the gaze data arises if one considers results that observe that visual attention seems to spread over the surface of objects. Given such an observation, it is natural to consider an alternative representation in which very small fixation targets(see Fig. 5.4) are first identified on the surface of objects, and then allowed to propagate or diffuse over the surface of the object. This can be done either by using surface based smoothing methods on the model itself, or in the case of complex geometry, but applying a small blur to a texture map that captures the loci of fixations. Given that local regions of the model are proximal on a gaze density texture map, blurring also has the effect of propagating gaze density over the surface, albeit in an approximate by much quicker fashion. Bojko [82] describes various types of attentional maps such as fixation count or fixation duration attention maps depending on the input data.

To represent visual attention for a 3D model we create surface-based attentional maps [83] as shown in Fig. 5.5. A surface-based attentional map may be thought of as a representation of where attention is directed, while strongly factoring in an assumption that attention propagates over 3D surfaces. This provides an example of one alternative representation that is distinct from an acuity based density that can be used to examine behaviour in a qualitative manner through visualization. This also demonstrates the significant benefit that the combination of a 3D model of the environment, and associated gaze data yields. There are few limits on the type of visualization that can be generated if a particular visualization is desired in considering a specific hypothesis.

5.3 Quantitative Analysis and Metrics

To ascertain maximum value from the methods presented in Chapter 4, it is sensible to also consider methods to quantify the relatedness of gaze patterns elicited by different observers, or ensembles of observers with different objectives, or for comparison with models that aim to predict gaze patterns for 3D models. Like the qualitative measures, there is also a rich set of prior work that addresses comparison of density maps, or positions of fixation points based on a 2D topographical representation. In what follows, we consider two specific measures, KL divergence and ROC Analysis, that are of particular importance to comparing densities or gaze points.

5.3.1 KL-Divergence

Various dissimilarity/similarity measures are important to solve many pattern recognition problems. These measures allow for comparison of two probability density functions. One metric is KL-divergence, also known as relative entropy [84]. KL-divergence is an asymmetric measure unlike most statistical measures of dispersion. Nevertheless, KL-divergence is a natural metric for comparison of normalized distributions. This metric involves information-gain [85], and captures KL-divergence between the predicted attention distribution, and a ground-truth distribution (or any other alternative attention distribution). To adapt this to 3D models, assessment is done on a per-triangle basis rather than a per-pixel basis, with all other elements remaining the same. A KL-divergence of 0 means that the two probability distributions being measured show similar behavior while a KL-divergence of 1 denotes very different behavior between the two distributions where the expected distribution

approaches 0. For the probability distributions P and Q, KL-divergence from Q to P is given by the expectated value of the logarithmic difference between P and Q, where the expectation is considered using probabilities from P. In this particular case, each of P and Q is a set of relative density values distributed across 2 mesh models with identical geometry that has a sum total of 1 across all triangles. This is effectively translating the base density into one where each triangle's value carries it's likelihood of being fixated by an observer. The equation for this measurement is as follows:

$$KL(Q\|P) = \sum_i Q(i) \log \frac{Q(i)}{P(i)} \tag{5.1}$$

In this case, Q and P refer to 2 densities, and i references each triangle in the two identical mesh models.

For the purpose of providing an example of applying this metric, we compare the original 3D model with gaze densities to the magnitude of model curvature where the curvatures were determined by different algorithms (colored according to gradient magnitudes in 3-space in figures). As mentioned earlier KL-divergence compares two probability distributions hence we considered our original model and the colorized model as the ground-truth distribution(P) and the predicted distribution(Q) respectively. For one such case, the KL-divergence from Q to P comes out to be 0.0051, which shows a surprising degree of agreement between local curvature and gazed at locations. This is perhaps not especially surprising in this case, owing to the fact that the statue is one of the few non-flat surfaces in the scene, and most fixations reside in this area.

5.3.2 ROC Analysis

Receiver Operating Characteristic (ROC) curves are a classic tool in signal detection theory [86]. The underlying premise of signal detection theory is that many detectors can have their level of sensitivity adjusted. This might imply that a weaker signal is able to be detected if a detector has its threshold adjusted downward. However, the greater the sensitivity, the greater the possibility that a detector will elicit a false detection (i.e. where no actual signal is present).

In the context of 2D ROC analysis in examining gaze, the pixel location of eye movements are examples where a detector should elicit a positive signal, while those without gaze points should elicit no signal. This formalism allows for a representation that is predictive of where eye movements may reside to be tested across the entire range of sensitivity. The extremes in this case are that the detector determines that every pixel is a fixated location, or that no pixel is a fixated location. For a continuous probability density over the image, intermediate thresholds may be chosen and treated as the detector. That is, for a given threshold, the continuous density map becomes binary, and it is possible to measure specifically which true signals (gazed at pixels) have a corresponding positive in the binary map. Choosing a large number of thresholds between the two extremes described yields an entire curve, and often the area under the curve is used as a summative measure of performance. The axes of this curve correspond to true positives (hits), and false positives (false alarms) respectively.

In considering how this representation might be adapted such that it is applicable to gazed at locations on 3D models, one can instead frame the problem with respect to

the mesh geometry rather than pixels. That is, a ground truth map can be generated by producing a small number of "on" triangles that correspond to the foveal region, or centre of gaze, and superimposed on the model. Given a continuous density map normalized to have the properties of a probability density function, one can select an arbitrary number of thresholds between the minimum and maximum values to produce an ROC curve for the 3D mesh model. Such a density map might be derived in a variety of ways. For example, a density might be derived from a number of observers, and ROC analysis used in a leave-one-out fashion to examine how well the ensemble of observers predicts each individuals viewing patterns. This may identify observers that display atypical behaviour, or reveal multiple strategies of viewing. One might also consider a density, and fixations from 2 different populations, and use ROC analysis to examine similarities or differences; this same type of analysis might also be performed for a single, or multiple groups across conditions (e.g. task directives). Finally, there is the opportunity to derive models that consider 3D geometry that form predictive computational models of gaze patterns. Given fixations from an observer, or several observers, the predicted density derived from the 3D model can also be tested. This gives a wide array of methods that allow for mobile data capture, and 3D models to be used in ways that enable new understanding in real-world scenarios. In the section that follows, we choose one relatively simple measurement in local surface curvature to demonstrate how a density might be generated that can be tested against fixated locations.

5.3.3 Examining Curvature

A 3D mesh is typically composed of vertices and faces of the triangles that form the geometry of the mesh. When a mesh is represented by many relatively small triangles, lighting models will produce a realistic representation of how the model is illuminated. For more coarse grained models, some degree of interpolation is necessary. Classic methods in graphics use lighting models that rely heavily on surface normals and smooth shading (Gouraud shading) can yield a good representation of a diffusely lit model [87] by storing a normal for each vertex of the mesh. The vertex itself does not have a normal, but rather the normal is an average of the normal of faces that include a given vertex. For smoother lighting, and especially specular lighting, interpolation of lighting across triangles is insufficient, and normals are also interpolated across triangles in determining how they are lit. While normals are orthogonal to a vector tangent to the face, normal interpolation is most important for parts of a model that vary substantially in direction. One can characterize this variation, and this is often framed in terms of local curvature on the surface of the model. Given that real scenes contain many flat areas (floors, walls), for demonstrative purposes, we aim to examine how measures of local curvature might be used in the context of the proposed metrics in their ability to predict points of gaze.

To achieve this, we generate representations of local curvature according to a variety of methods, and present this as colored models. In this case, the red, green and blue channels respectively represent the magnitude of curvature along the principal coordinate directions.

Robust Implicit Moving Least Squares(see Fig. 5.7): This method is based on a

robust objective function RIMLS (Robust Implicit Moving least squares [88], which preserves sharp features of the mesh by applying local kernel regression. Implicit Moving least squares(IMLS) based methods reconstruct only smooth surfaces. This family of methods combines with robust local kernel regression to give a robust IMLS estimate i.e. RIMLS for representing underlying surfaces. This approach is robust in the presence of noise, sparse sampling and outliers, and also improves sharp features and fine details irrespective of frequency.

Algebraic Point Set Surfaces(see Fig. 5.8): A second method for estimating local curvature is called Algebraic Point Set Surfaces (APSS) [89], which focuses on the local fit of algebraic spheres to the geometric data. Initially, Point Set Surfaces (PSS) were used that applied planar fits to point sets, but these methods were later replaced by the spherical fits. The sphere fitting MLS methods were considered to be better than planar MLS in terms of robustness while also providing a better approximation for mean curvature estimation. However, geometric sphere fits have several disadvantages related to the size of the center-radius representations considered while also requiring solution to a non-linear problem. A better alternative to this approach was to replace the geometric distance by an algebraic distance [90], which presents higher-quality results, especially for complex models.

Both APSS and RIMLS functions used to generate colored representations of curvature include the possibility of estimating other measures of curvature including Gaussian and principal curvatures (K1 and K2 [91]) besides mean curvature.

Curvature in Principal Directions: Several algorithms such as the Taubin approximation(Fig. 5.9), Principal Component Analysis(Fig. 5.10), Normal Cycles(Fig. 5.11)

and Pseudoinverse Quadratic Fitting(Fig. 5.12) can be applied to compute principal directions of curvature locally within a 3D model. Mean and Gaussian curvature are approximated based on Eulers formula, the maximum rate of surface bending and the maximum tangential direction of bending is determined by the two principal curvatures. These two principal curvatures or vectors are known as k1 and k2 which correspond to local shape of any point on a surface. The maximum bending rate of the surface and the tangent direction in which the bending occurs is described by k1 while k2 describes the rate of minimum bending of the surface along the tangent direction of minimum bending. Principal curvature is used as a precursor for computing Mean and Gaussian curvature over grouped point sets.

For mean curvature the focus is on local variation of the surface, and this metric allows for divergence of normals over the embedded surface. It is equal to the average of two principal curvatures of a surface where k1 and k2 are the maximal and minimal curvature respectively and their directions are orthogonal: $(H = (k1 + k2)/2)$ [92].

In the case of Gaussian curvature (also known as total curvature), one is interested in the product of the two principal curvatures $H = k1 * k2$ at a given point of a surface.

5.4 Conclusions

In this section we have presented a variety of techniques for qualitative and quantitative analysis of gaze data. This includes methods for transferring common analytical techniques that are used for gaze analysis on a flat display to allow applicability to a 3D model. We also highlight some important perceptual considerations in representing gaze density, and provide a few natural examples of visualizations, and other

examples that present the unique opportunities for visualization that the methods presented in the thesis allow. We also provide examples of observations that may be made from data visualization that could not be made based on existing methods in eye tracking research.

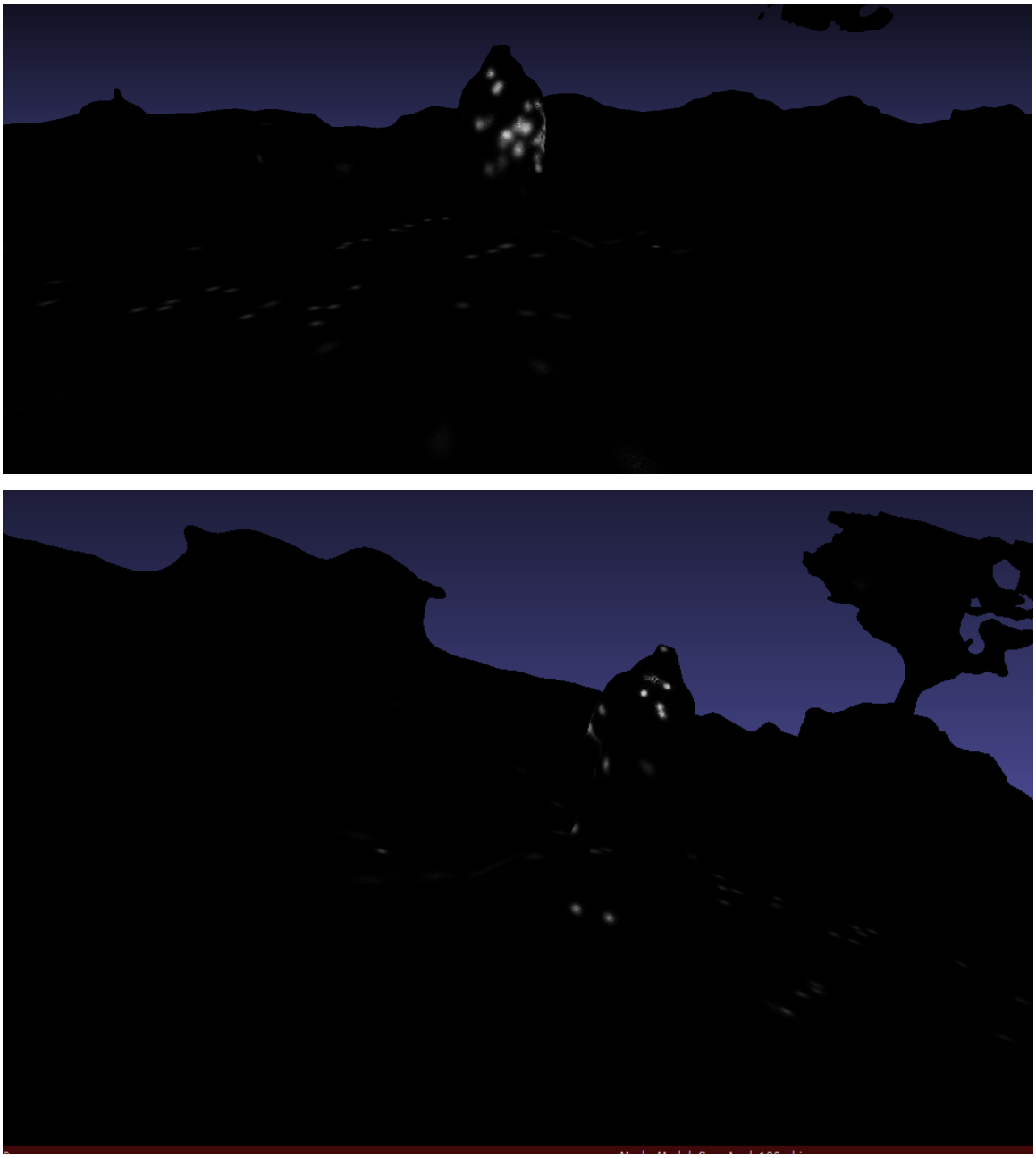


Figure 5.4: The rendered image of a model representing small fixation targets on the surface of the statue.

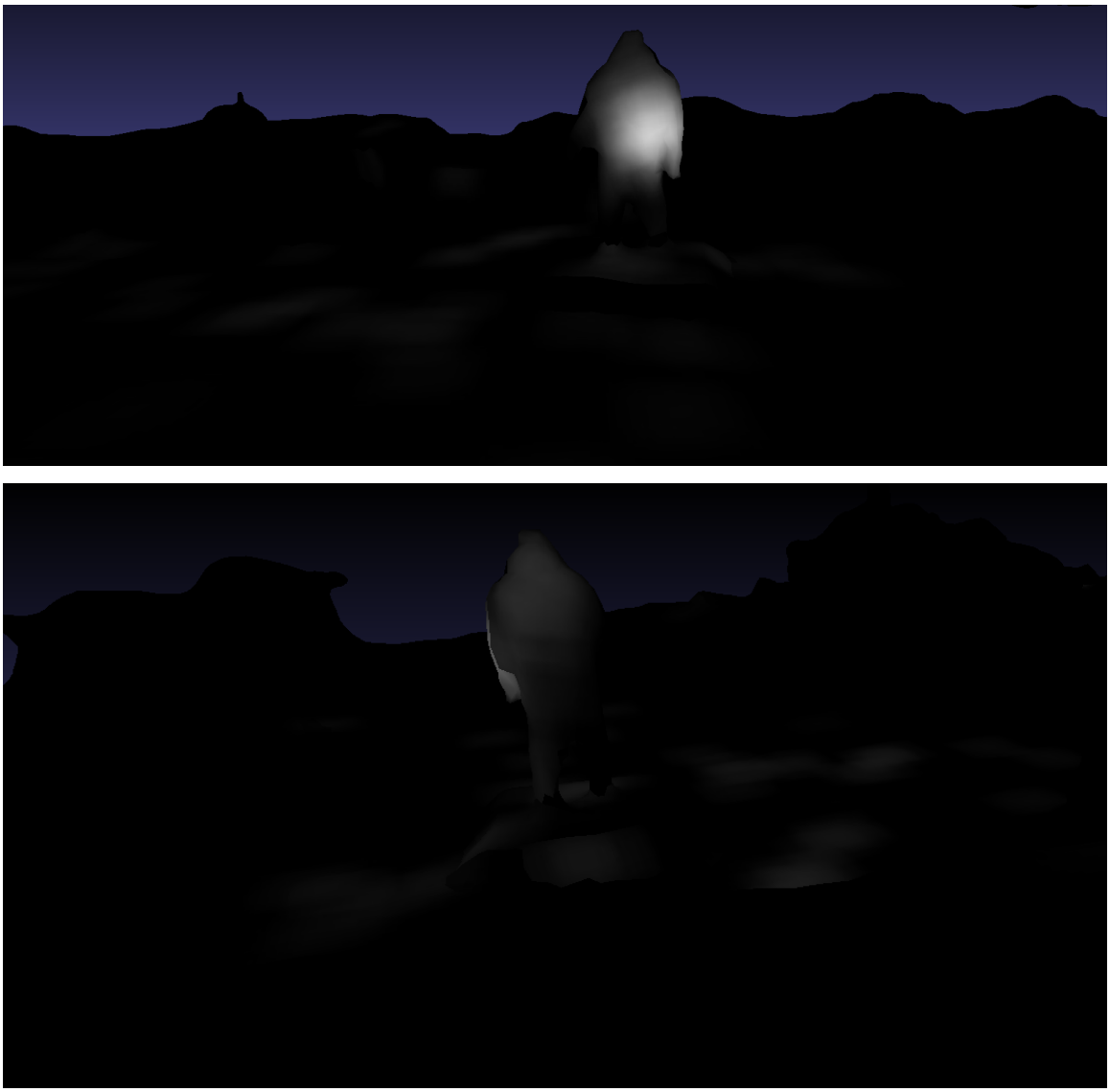


Figure 5.5: Surface-based attentional map with diffused fixations over the surface of the statue.

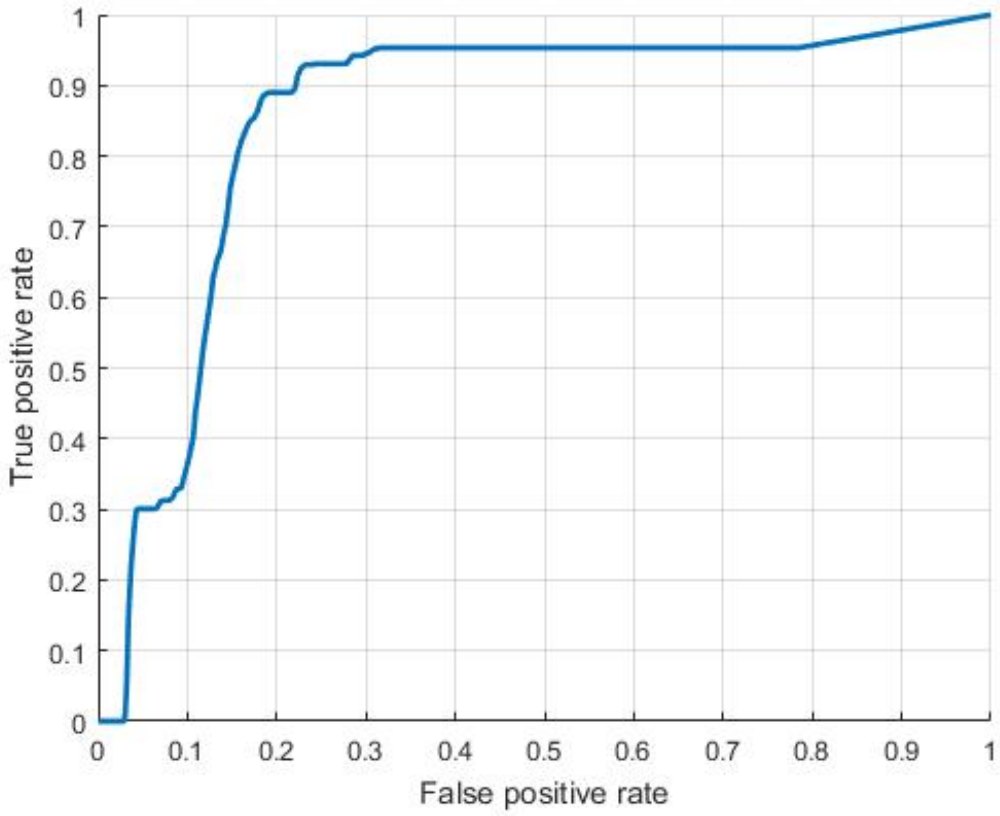


Figure 5.6: ROC curve for a predicted density map generated from a local surface curvature tested against fixations from an observer.

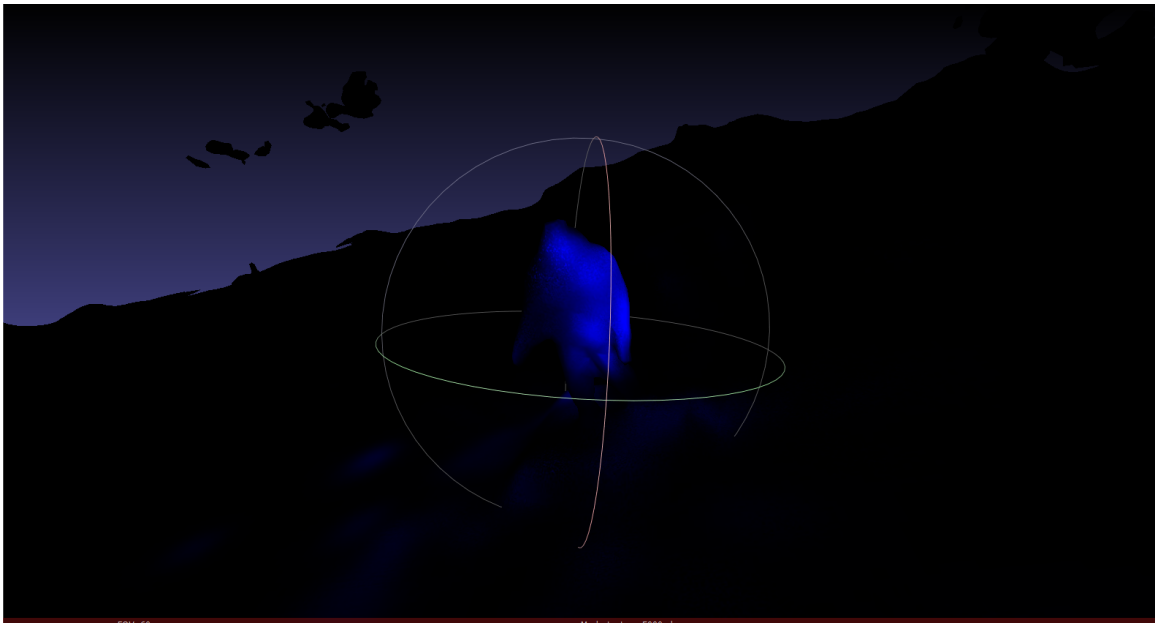


Figure 5.7: 3D representation of RIMLS curvature filter

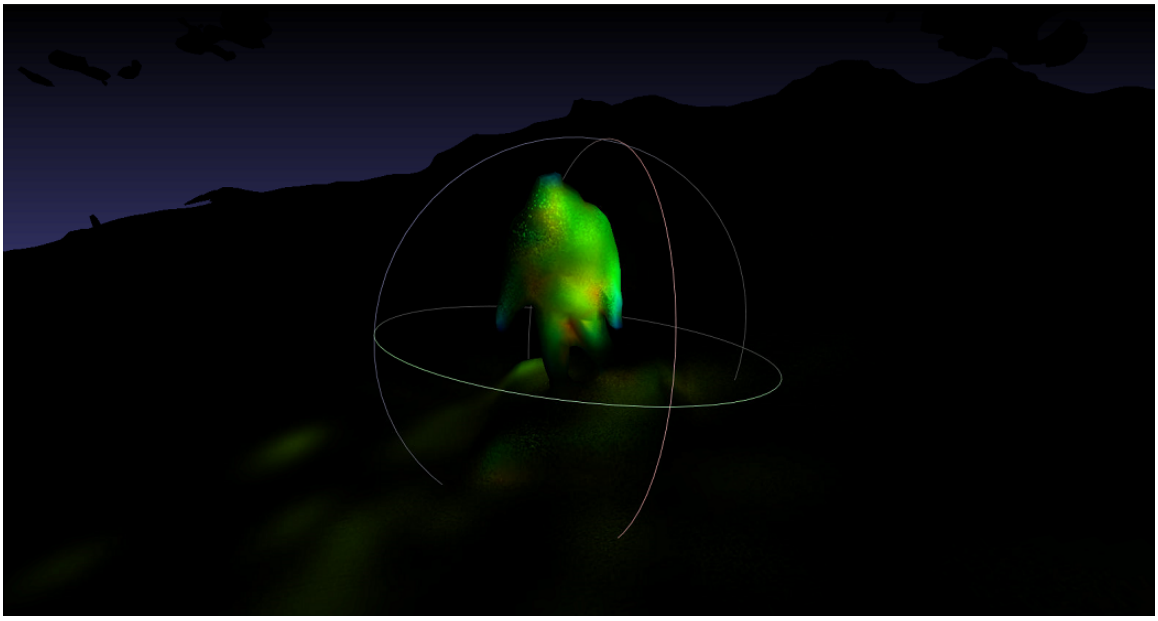


Figure 5.8: 3D representation of APSS curvature filter

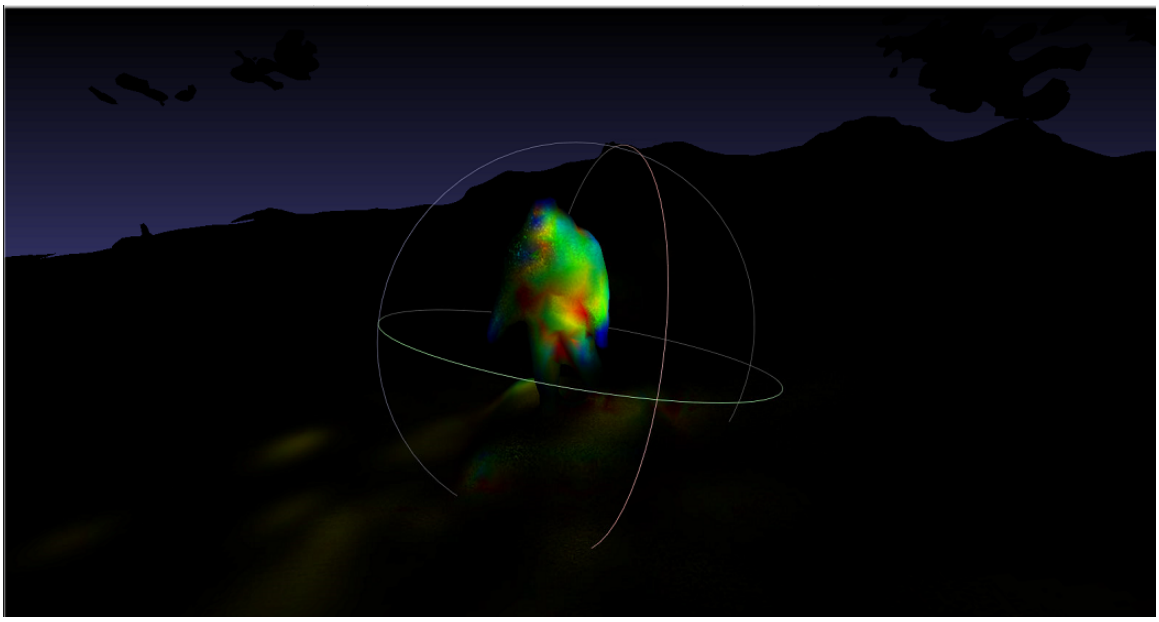


Figure 5.9: Principal directions of curvature computed with Taubin approximation

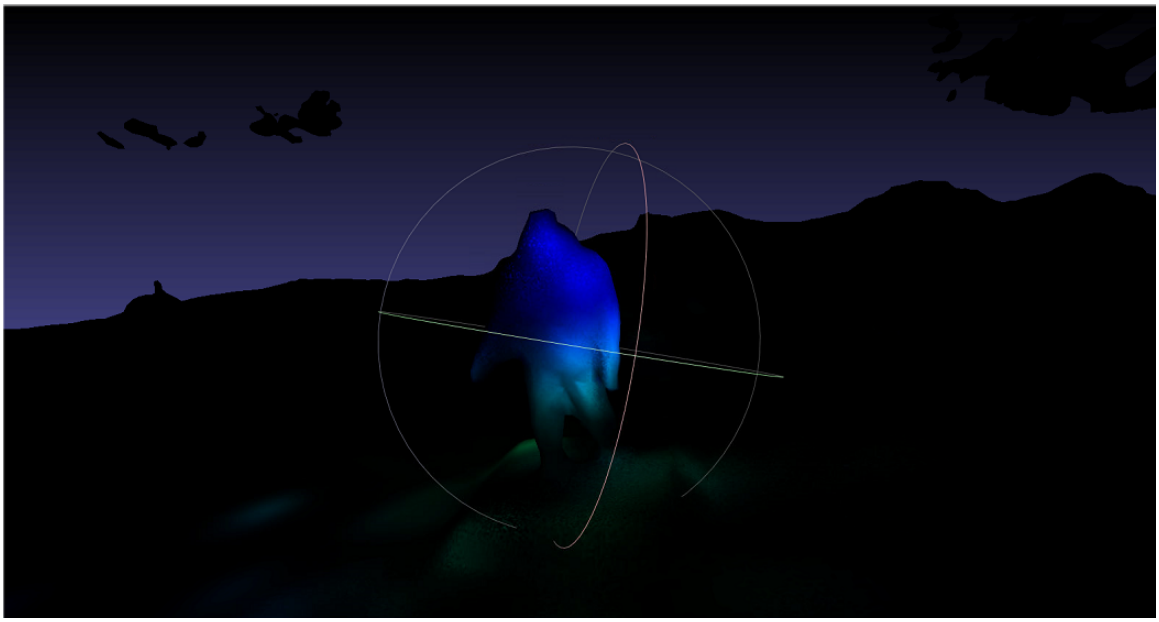


Figure 5.10: Principal directions of curvature computed with Principal Component Analysis

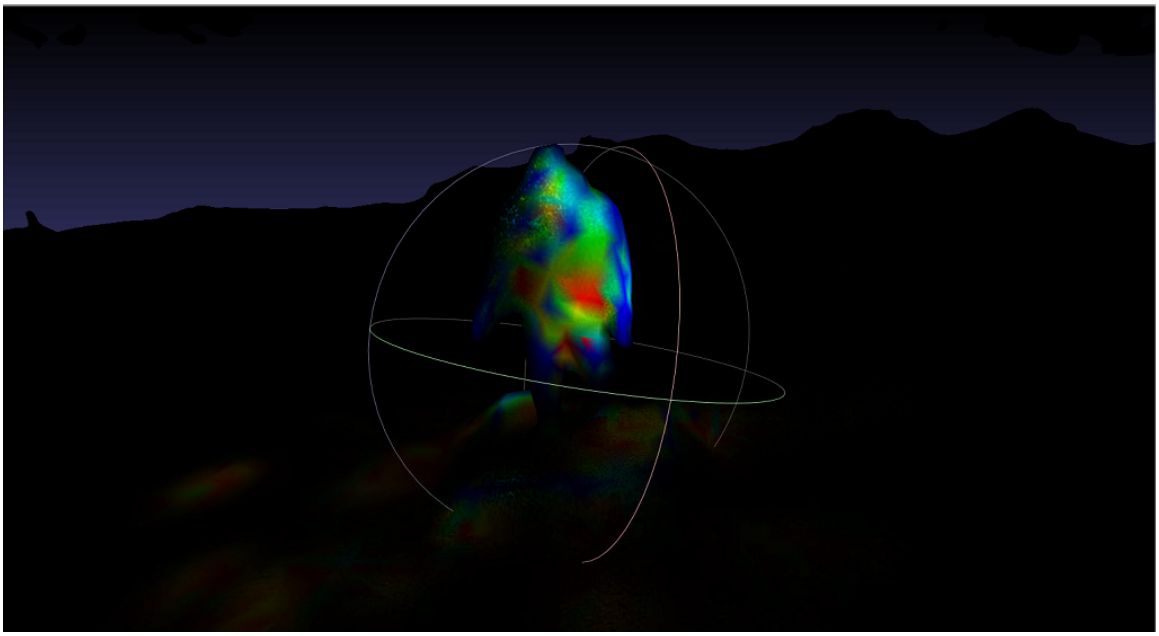


Figure 5.11: Principal directions of curvature computed with Normal Cycles method

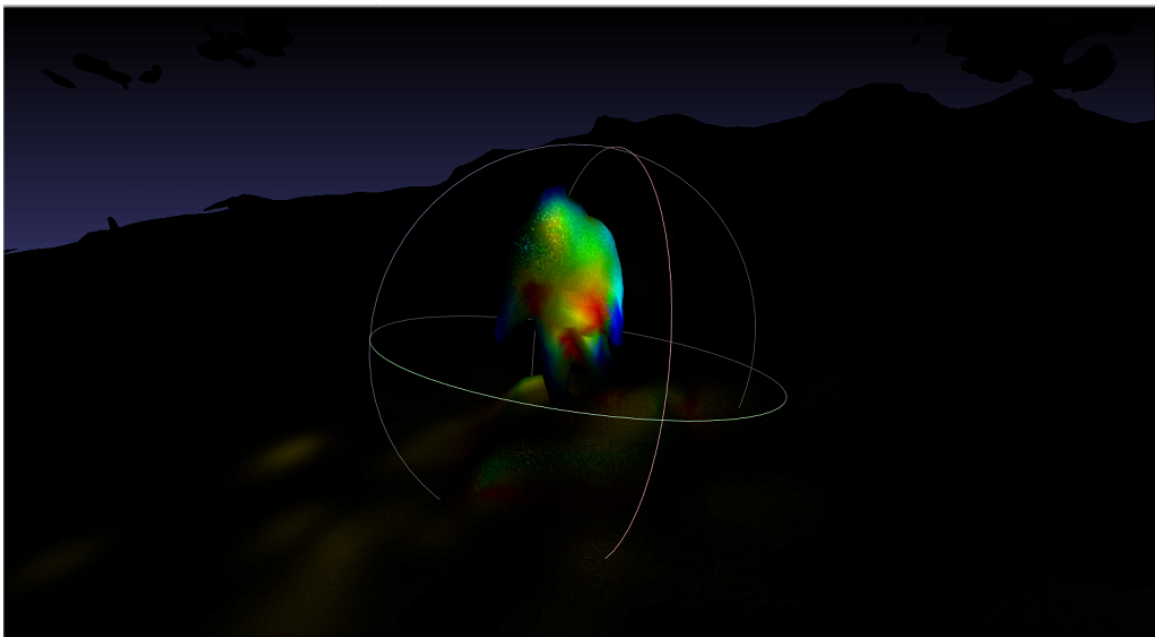


Figure 5.12: Principal directions of curvature computed with Pseudoinverse Quadric Fitting

Chapter 6

Conclusions and Future Work

In this thesis, we have considered how algorithms in computer vision, and recent advances in gaze tracking technology can be leveraged to allow for eye tracking in a relatively unconstrained manner. To date, the vast majority of eye tracking studies have been conducted in a lab environment, or within a highly constrained setting in the real world. The methods in this thesis sought to provide a complete end-to-end solution that allows for collection, and representation of gaze data in the real-world, and without traditional constraints.

To this end, we have performed a wide variety of experimentation using different methods in 3D reconstruction to determine how to best produce high quality 3D models for analysis. This is a critical consideration, since if any data collected is to reside in a 3-dimensional environment that may be unknown, one also requires a representation of this environment to relate any gaze data to. We have shown that it is possible to use existing sensors on modern eye tracking glasses, or in combination with SLR images to produce high-fidelity 3D models of environments.

We also consider the problem of how human gaze translates into a 3D representation. This involves two principal methods wherein either the acuity with which an environment is sampled is measured and recorded, or alternatively a surface-centric representation where the focal point of gaze is assumed to spread over the surface of a model. This has been achieved using a pipeline that leverages efficiencies of modern rendering hardware, and simulating sampling according to acuity by casting rays in the direction of the point of gaze. A masked spotlight model was used for this purpose, which allows the efficiency of programmable shaders to be exercised. This is a critical consideration since every point of gaze creates a light, and this imposes a high computational burden. For the surface based approach, we also demonstrate an approximate way of determining spread over surface in directly filtering more localized gaze maps represented as a texture.

Given that there are few examples of prior work that considers unconstrained gaze tracking in 3D, it is unclear how the data that is arrived at can be used, either for qualitative and quantitative analysis. Based on this motivation, we demonstrate how each of the most common strategies that apply to 2D eye tracking data may be transferred to three dimensions, and also discuss the implications of moving from pixel-centric to triangle-centric representations. This is accompanied by different strategies for visualizing the data, including both head-centric and gaze-centric renderings. One interesting observation made, is that head-centric renderings might provide a reasonable proxy to gaze-centric renderings when expensive proprietary lightweight eye trackers are unavailable. We also show how a modeling approach, based on local curvature might be applied to predicting gazed at points, and relate this to two metrics

that allow for comparison of two densities, or a density and binary fixation points respectively on 3D models.

As a whole, this work presents the first complete successful set of methodology for the capture of base data and models necessary for representing eye movement behaviour in real-world settings, while also illustrating how quantitative and qualitative analysis may be conducted on this data. This presents a solid foundation for future research across a variety of scientific disciplines, and suitable for numerous applications. This presents one of the most natural future prospects for the current work: quantifying and characterizing human gaze behavior in discipline, or scenario specific experimentation to learn new things about how humans behave.

The last decade has also seen a plethora of work on modeling gaze behaviour for still images, and predictive models of how humans view images on a screen. This however is a far cry from how humans behave in the real world. The work presented in this thesis sets the stage for development of models that appeal to both appearance, and geometry as represented by rich 3D models. In this regard, this work also establishes a window into a new paradigm of modeling human behaviour that may be more reflective of real behaviour, and more useful for this reason. As these methods are more broadly adopted by the community, and as extensive data sets are collected, this will present many new opportunities for those interested in computational modeling of gaze behaviour.

Bibliography

[1] C. Wu, "Siftgpu: A gpu implementation of scale invariant feature transform (sift)," 2007.

[2] M. Westoby, J. Brasington, N. Glasser, M. Hambrey, and J. Reynolds, "structure-from-motionphotogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology*, vol. 179, pp. 300–314, 2012.

[3] C. Wu *et al.*, "Visualsfm: A visual structure from motion system," 2011.

[4] J. REKITTKE and Y. NINSALAM, "Head in the point clouds–feet on the ground," in *15th annual International Conference on Information Technology in Landscape Architecture*. Wichmann, 2014, pp. 198–207.

[5] S.-S. Instruments, "Begaze 2.2 manual," *Germany: Teltow*, 2009.

[6] D. Derakhshani, *Introducing Autodesk Maya 2013*. John Wiley & Sons, 2012.

[7] V. Hugot, "Eye gaze analysis in human-human interactions," *Unpublished masters thesis, CSC, KTH, Sweden*, 2007.

[8] O. Le Meur, A. Ninassi, P. Le Callet, and D. Barba, "Overt visual attention for free-viewing and quality assessment tasks: Impact of the regions of interest on a video

quality metric," *Signal Processing: Image Communication*, vol. 25, no. 7, pp. 547–558, 2010.

[9] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel, "The role of fixational eye movements in visual perception," *Nature Reviews Neuroscience*, vol. 5, no. 3, pp. 229–240, 2004.

[10] E. Wästlund, P. Shams, M. Löfgren, L. Witell, and A. Gustafsson, "Consumer perception at point of purchase: Evaluating proposed package designs in an eye-tracking lab," *Journal of Business & Retail Management Research*, vol. 5, no. 1, pp. 41–50, 2010.

[11] J. L. Orquin and S. M. Loose, "Attention and choice: A review on eye movements in decision making," *Acta psychologica*, vol. 144, no. 1, pp. 190–206, 2013.

[12] L. A. Granka, T. Joachims, and G. Gay, "Eye-tracking analysis of user behavior in www search," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2004, pp. 478–479.

[13] S. Rahman and N. Bruce, "Saliency, scale and information: Towards a unifying theory," in *Advances in Neural Information Processing Systems*, 2015, pp. 2188–2196.

[14] D. A. Robinson, "A method of measuring eye movemnent using a scieral search coil in a magnetic field," *IEEE Transactions on bio-medical electronics*, vol. 10, no. 4, pp. 137–145, 1963.

[15] H. D. Crane and C. M. Steele, "Generation-v dual-purkinje-image eyetracker," *Applied Optics*, vol. 24, no. 4, pp. 527–537, 1985.

[16] T. Elbert, W. Lutzenberger, B. Rockstroh, and N. Birbaumer, "Removal of ocular

artifacts from the eega biophysical approach to the eog," *Electroencephalography and clinical neurophysiology*, vol. 60, no. 5, pp. 455–463, 1985.

[17] E. P. Baltsavias, "A comparison between photogrammetry and laser scanning," *ISPRS Journal of photogrammetry and Remote Sensing*, vol. 54, no. 2, pp. 83–94, 1999.

[18] J. Sužiedelytė-Visockienė, R. Bagdžiūnaitė, N. Malys, and V. Maliene, "Close-range photogrammetry enables documentation of environment-induced deformation of architectural heritage," *Environmental Engineering and Management Journal*, vol. 14, no. 6, pp. 1371–1381, 2015.

[19] T. Lundstrom, J. Baqersad, C. Niezrecki, and P. Avitabile, "Using high-speed stereophotogrammetry techniques to extract shape information from wind turbine/rotor operating data," in *Topics in Modal Analysis II, Volume 6*. Springer, 2012, pp. 269–275.

[20] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[21] M. V. Clark, *Walking through Rome: A Guide to Interesting Sites in the Eternal City*. iUniverse, 2013.

[22] P. D. A. Harvey, *The history of topographical maps: symbols, pictures and surveys*, 1980.

[23] J. Bourgeois and M. Meganck, *Aerial Photography and Archaeology 2003: A Century of Information; Papers Presented During the Conference Held at the Ghent University, December 10th-12th, 2003*. Academia press, 2005, vol. 4.

[24] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[25] T. Moons, L. Van Gool, and M. Vergauwen, *3D Reconstruction from multiple images, part 1: Principles*. Now Publishers Inc, 2009.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] T. Möller and B. Trumbore, "Fast, minimum storage ray/triangle intersection," in *ACM SIGGRAPH 2005 Courses*. ACM, 2005, p. 7.

[28] D. Girardeau-Montaut, "Cloudcompare-open source project," *OpenSource Project*, 2011.

[29] P. Moulon and A. Bezzi, "Python photogrammetry toolbox: a free solution for three-dimensional documentation," in *ArcheoFoss*, 2011, pp. 1–12.

[30] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *CVPR*, 2010.

[31] J. Biehler and B. Fane, *3D Printing with Autodesk: Create and Print 3D Objects with 123D, AutoCAD and Inventor*. Que Publishing, 2014.

[32] L. AgiSoft, "Agisoft photoscan," *Professional Edition*, 2014.

[33] F. Mancini, M. Dubbini, M. Gattelli, F. Stecchi, S. Fabbri, and G. Gabbianelli, "Using unmanned aerial vehicles (uav) for high-resolution reconstruction of topography: the structure from motion approach on coastal environments," *Remote Sensing*, vol. 5, no. 12, pp. 6880–6898, 2013.

[34] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE, 2011, pp. 3057–3064.

[35] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[36] Z. Kang and G. Medioni, "Fast dense 3d reconstruction using an adaptive multiscale discrete-continuous variational method," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on.* IEEE, 2014, pp. 53–60.

[37] M. Rothermel, K. Wenzel, D. Fritsch, and N. Haala, "Sure: Photogrammetric surface reconstruction from imagery," in *Proceedings LC3D Workshop, Berlin*, vol. 8, 2012.

[38] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz, "Occluding contours for multi-view stereo," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[39] S. Fuhrmann, F. Langguth, and M. Goesele, "Mve-a multi-view reconstruction environment." in *GCH*, 2014, pp. 11–18.

[40] A. J. Davison, Y. G. Cid, and N. Kita, "Real-time 3d slam with wide-angle vision," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 868–873, 2004.

[41] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* IEEE, 2013, pp. 2100–2106.

[42] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* IEEE, 2011, pp. 3607–3613.

[43] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision.* Springer, 2014, pp. 834–849.

[44] T. Schöps, J. Engel, and D. Cremers, "Semi-dense visual odometry for ar on a smartphone," in *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on.* IEEE, 2014, pp. 145–150.

[45] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-time camera tracking and 3d reconstruction using signed distance functions." in *Robotics: Science and Systems*, vol. 2, 2013.

[46] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on.* IEEE, 2011, pp. 127–136.

[47] J. Chandler and J. Fryer, "Autodesk 123d catch: how accurate is it," *Geomatics World*, vol. 2, no. 21, pp. 28–30, 2013.

[48] N. Snavely *et al.*, "Bundler: Structure from motion (sfm) for unordered image collections," *Available online: phototour. cs. washington. edu/bundler/(accessed on 12 July 2013)*, vol. 1, 2010.

[49] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Computer vision and*

pattern recognition, 2006 IEEE Computer Society Conference on, vol. 1. IEEE, 2006, pp. 519–528.

[50] M. Sarcar, K. M. Rao, and K. L. Narayan, *Computer aided design and manufacturing*. PHI Learning Pvt. Ltd., 2008.

[51] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "Openmvg: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2016, pp. 60–74.

[52] M. Dunn, "Recent developments in close range photogrammetry for mining and reclamation," *Proceedings America Society of Mining and Reclamation*, pp. 390–399, 2009.

[53] F. Bianconi, S. Catalucci, M. Filippucci, R. Marsili, M. Moretti, G. Rossi, and E. Speranzini, "Comparison between two non-contact techniques for art digitalization," in *Journal of Physics: Conference Series*, vol. 882, no. 1. IOP Publishing, 2017, p. 012005.

[54] S. Instruments, "Smi eye tracking glasses," 2012.

[55] K.-c. Chen and H. J. Choi, "Visual attention and eye movements," 2008.

[56] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*. ACM, 2000, pp. 71–78.

[57] K. Rayner, B. R. Foorman, C. A. Perfetti, D. Pesetsky, and M. S. Seidenberg, "How psychological science informs the teaching of reading," *Psychological science in the public interest*, vol. 2, no. 2, pp. 31–74, 2001.

[58] R. J. Leigh and D. S. Zee, *The neurology of eye movements*. Oxford University Press, USA, 2015, vol. 90.

[59] M. Dieterich and T. Brandt, "Vestibulo-ocular reflex." *Current opinion in neurology*, vol. 8, no. 1, pp. 83–88, 1995.

[60] K. Yun, Y. Peng, D. Samaras, G. J. Zelinsky, and T. L. Berg, "Studying relationships between human gaze, description, and computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 739–746.

[61] L. Paletta, K. Santner, G. Fritz, H. Mayer, and J. Schrammel, "3d attention: measurement of visual saliency using eye tracking glasses," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 199–204.

[62] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*. Springer, 2008, pp. 871–889.

[63] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular slam," *IEEE transactions on robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[64] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 659–666.

[65] R. Hess, *The essential Blender: guide to 3D creation with the open source suite Blender*. No Starch Press, 2007.

[66] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," in *Computer Graphics Forum*, vol. 36, no. 1. Wiley Online Library, 2017, pp. 301–329.

[67] J. Liu, "Simple technique for measurements of pulsed gaussian-beam spot sizes," *Optics letters*, vol. 7, no. 5, pp. 196–198, 1982.

[68] T. Driemeyer, *Rendering with mental ray® (mental ray® Handbooks)*. Springer-Verlag New York, Inc., Secaucus, NJ, 2005.

[69] D. J. Wheeler, "The use of sub-routines in programmes," in *Proceedings of the 1952 ACM National Meeting (Pittsburgh)*, ser. ACM '52. New York, NY, USA: ACM, 1952, pp. 235–236. [Online]. Available: http://doi.acm.org/10.1145/609784.609816

[70] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: an open-source mesh processing tool." in *Eurographics Italian Chapter Conference*, vol. 2008, 2008, pp. 129–136.

[71] N. Northey, "The angle of view of your lens," *The Camera; Columbia Photographic Society: Philadelphia, PA, USA*, 1916.

[72] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl, "Visualization of eye tracking data: A taxonomy and survey," in *Computer Graphics Forum*. Wiley Online Library, 2017.

[73] M. S. Atkins, X. Jiang, G. Tien, and B. Zheng, "Saccadic delays on targets while watching videos," in *Proceedings of the symposium on eye tracking research and applications*. ACM, 2012, pp. 405–408.

[74] S. A. Brasel and J. Gips, "Points of view: Where do we look when we watch tv?" *Perception*, vol. 37, no. 12, pp. 1890–1894, 2008.

[75] D. J. Berg, S. E. Boehnke, R. A. Marino, D. P. Munoz, and L. Itti, "Free viewing of

dynamic stimuli by humans and monkeys," *Journal of vision*, vol. 9, no. 5, pp. 19–19, 2009.

[76] A. J. Hornof and T. Halverson, "Cleaning up systematic error in eye-tracking data by using required fixation locations," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 592–604, 2002.

[77] J. H. Goldberg and J. I. Helfman, "Visual scanpath representation," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM, 2010, pp. 203–210.

[78] P. Kovesi, "Good colour maps: How to design them," *CoRR*, vol. abs/1509.03700, 2015. [Online]. Available: http://arxiv.org/abs/1509.03700

[79] P. Bourke, "Ply-polygon file format," *Dostupné z www: http://paulbourke.net/dataformats/ply*, 2009.

[80] E. Kowler, "Eye movements: The past 25years," *Vision research*, vol. 51, no. 13, pp. 1457–1483, 2011.

[81] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *Journal of vision*, vol. 7, no. 14, pp. 4–4, 2007.

[82] A. Bojko, *Eye tracking the user experience: A practical guide to research*. Rosenfeld Media, 2013.

[83] S. Stellmach, L. Nacke, and R. Dachselt, "3d attentional maps: aggregated gaze visualizations in three-dimensional virtual environments," in *Proceedings of the international conference on advanced visual interfaces*. ACM, 2010, pp. 345–348.

[84] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[85] A. Rényi *et al.*, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[86] J. A. Nevin, "Signal detection theory and operant behavior: A review of david m. green and john a. swets' signal detection theory and psychophysics." *Journal of the Experimental Analysis of Behavior*, vol. 12, no. 3, pp. 475–480, 1969.

[87] H. Gouraud, "Continuous shading of curved surfaces," *IEEE transactions on computers*, vol. 100, no. 6, pp. 623–629, 1971.

[88] S. Fleishman, D. Cohen-Or, and C. T. Silva, "Robust moving least-squares fitting with sharp features," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 544–552.

[89] G. Guennebaud and M. Gross, "Algebraic point set surfaces," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 23.

[90] V. Pratt, "Direct least-squares fitting of algebraic surfaces," in *ACM SIGGRAPH computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 145–152.

[91] R. Martin, "Estimation of principal curvatures from range data," *International journal of shape modeling*, vol. 4, no. 03n04, pp. 99–109, 1998.

[92] S. Rusinkiewicz, "Estimating curvatures and their derivatives on triangle meshes," in

3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on. IEEE, 2004, pp. 486–493.