[slide 1]

Hi, I’m Katrina and I’m going to tell you a bit about my project.

[slide 2]

The purpose of my project is to find a way of representing an environment that is efficient to calculate, store and update. Being able to do this is very important for applications such as robotic mapping and 3D modelling.

A general representation of an environment is called a scene map.

[slide 3] ~ 20s

The most obvious way to implement something link this is a point cloud. However there are two major problems with doing this. The first is that the number of points will blow up very quickly, and thus take up a lot of storage space. The second is that it does not give a way of interpolating between points, which is a problem if, for example, you are interested in what the environment is like in a specific area in which there is not a point.

[slide 4] ~43s

This is why my project uses a depth map. That is, a function that given a direction will return the depth of the environment in that direction from a specific point. In this case, the depth map returns the depth from O to any point P in the environment.

[slide 5] ~57

Again, there are some relatively obvious ways to implement something like this. Returning to the idea of point clouds, we could simply form a triangular mesh where the vertices are the points. This solves the interpolation issue, however, the difficulty of updating remains.

Consider what happens when we observe a new point that is not radially aligned with an existing vertex.

[slide 6] ~1:20

We could simply add another vertex for that point. However this has the same issue of blowing up the number of vertices required.

[slide 7] ~1:26

Alternatively, we could work out some way to move the existing vertices in order to include the new point. This necessitates changing the triangles around the one which the point lies in, as each vertex is part of more than one triangle.

[slide 8] ~1:40

So now that you know why more complicated methods may be necessary, I can start explaining what I’ve actually been working on. That is, basis functions. You may have heard the term basis in linear algebra. This is a similar concept, in that if you have a set of basis functions for some space, then a linear combination of these basis functions can represent any element in that space. However, the difference lies in the fact that this linear combination can be an infinite sum. An example of basis functions being used that you are probably all familiar with is Fourier Transforms. In this case, the basis functions are sinusoidal waves with various translations and dilations.

There are many different possible bases for a space; choosing one that is well-suited to what you are trying to represent is important as it will allow for efficient representations. This project will investigate two bases of the function space L\_2(S^2), that is square-integrable functions on the sphere. These bases are the spherical harmonics and spherical Haar wavelets.

[slide 9] ~2:41

I’m not going to go into the technical details of what exactly spherical harmonics are in this talk, but I will explain some of the basics.

The spherical harmonics are only a basis for L\_2(S^2) when the degrees of the spherical harmonics approach infinity. Using a basis representation where the number of basis functions goes to infinity is not possible for this application, so my work uses a truncated basis. This means that representations of the environment are not necessarily exact. However this will be the case with any basis for L\_2(S^2) as it is an infinite-dimensional space.

Ok, now let me explain a bit more about what’s on this slide. Spherical harmonics have a degree, which in this slide is called “l”. On the left, the first few degrees of the spherical harmonics are shown. Note the symmetrical nature of the shapes.

The right shows a 3D spherical harmonic representation of a model. These models do a good job of capturing the overall shape, but lack the texture of the original model.

Together, these diagrams shows some of the advantages and weaknesses of spherical harmonics. In terms of advantages, they are smooth and relatively simple to understand and implement.

The main disadvantages are that they are not localised and that they are sensitive to symmetry. These can be benefits if the environment being modelled is very regular and symmetrical, however this is usually not the case.

[slide 10] ~3:00

To combat these disadvantages, I investigated spherical Haar wavelets. However, before I can go into what those are I need to explain some of their predecessors.

Wavelets on the real line were introduced as a way of localising Fourier transforms. The idea of using translations and dilations of a single function remains the same, however, there are two major differences from the Fourier transform. The first, is that the function is not periodic. The second is there is not a standard choice for what this function should be.

Wavelets have equivalents of many of the concepts from Fourier analysis, such as fast wavelet transforms for discrete signals. They also have a few new concepts. A major one is the idea of introducing scaling functions.

A basis could be formed just using wavelet functions or scaling functions, however, combining them gives the resulting basis some nice properties. For example, making it easier to do the afore-mentioned fast wavelet transform. Note that combining two bases into one like this is only valid because they are a basis for an infinite dimensional space.

Ok, now that you know what wavelets in general are, I can talk about Haar wavelets specifically. The main idea of a Haar wavelet is that they divide their domain using a nested set of partitionings. This diagram shows one level of a Haar wavelet on the real line. Note that the scaling function is constant over the support of the wavelet function, whereas the wavelet function transitions from a value of 1 to a value of -1 half way through. The wavelet function one level down would be positive between 0 and 1/4, and negative between ¼ and ½. That is, it has the same shape as this wavelet but its support is halved.

[slide 11]

This idea can then be generalised to other domains, such as on the square

[slide 12] ~4:43

And on the sphere. Note that there is not actually a standardised way of defining Haar wavelets on the sphere. The one I’m using in this project is given by Lessig and Fiume in the paper cited here <point>, I call these spherical Haar wavelets throughout this talk for brevity, as they are the only type that I implemented.

These wavelets are defined with respect to spherical triangles that lie on the unit sphere. Each triangle then has 4 child triangles.

One of the major negatives of this set of basis functions is that the locations of the vertices of the spherical triangles need to be chosen to meet specific criteria, which I will not go into here.

[slide 13] ~5:18

Ok, so I’ve talked about the basis functions that I am using to create the models, now I need to go into how I update them. One complication with the updating is that for the application I am looking at, the measurements are taken from a camera that changes its pose (that is, its position and orientation) between batches of measurements. The model is defined in a space fixed frame, thus measurements from the camera fixed frame will not be in alignment with the model.

There are two ways to rectify this. One is to transform the model to align it with the measurements, which corresponds to doing the update in the camera fixed frame. The other is to transform the measurements so they are aligned with the model. This corresponds to doing the update in the space fixed frame.

[slide 14] ~5:58

If we go with the second method, there is an issue of the model point that was originally being moved no longer being in radial alignment with the point it was being moved to, as shown here. When I say radially aligned, I mean that a straight line which passes through both points can also pass through the origin. Not having radially aligned points is an issue because we have a depth map, so we can only change the value of what it gives us for a certain direction.

This picture shows the issue, note that it only occurs when the camera is translated. Basically, we took a measurement that told us to move the purple point to the black point, which is fine to do in the camera fixed frame. However, when we move that measurement back to the space fixed frame, the points are no longer radially aligned. Thus we have two main choices. We can move the wrong point on the model to the right measurement location (i.e. move the orange point to the black point). Alternatively, we can move the right point on the model to the wrong point in the environment. (i.e. move the purple point to the blue point).

In terms of what I actually did in my project, I chose to move the wrong point on the model to the right measurement location. However, given more time it would be interesting to investigate which method, or combination of methods, would be best.

[slide 15] ~7:10

Ok, so here’s a summary of my main results. I found very little difference between the two update methods for spherical harmonics. The major performance difference is that the moved measurements method is much faster.

For the spherical Haar wavelets, I didn’t have many results when making these slides. However, since then I have found that the moved model method converges faster than the moved measurements method. They also take a similar amount of computational time.

For both bases, the moved models methods could only be used for rotated measurements, whereas the moved measurements methods could also be used with translations.

One aspect of major importance for all of the methods was choosing a good step size for the gradient descent that was used to do the updates.

Another key point for the spherical Haar wavelets specifically is that the choice of data structure for representing them and their coefficients is very important. One of the benefits of wavelets is that they can be very efficient in representing signals in terms of number of coefficients used. This is achieved by simply zeroing coefficients that are small, which works due to the localised nature of wavelets. However, in order for this result in improvements in terms of space and computational time efficiency, the data structure used to store them needs to be able to take advantage of the sparse nature of the data.

I also have some models created with spherical harmonics and spherical Haar wavelets on this slide <point>. As you can see, the spherical harmonics are better at representing a cylinder, whereas wavelets are more efficient for representing a sphere with a single outwards spike.

[slide 16] ~8:43 (can skip)

Here’s my sum-squares error graph for spherical harmonics. Unfortunately I didn’t have these results for spherical Haar wavelets yet when I was putting together my slides. As you can see, the moved model method and moved measurements method are basically identical.

[slide 17] ~9:00

So, to summarise. My main work was on implementing ways to create a base model with spherical harmonics basis functions and a spherical Haar wavelet basis and how to update said models. In particular, how to update with measurements taken with respect to a frame that is not aligned with the frame the model is defined in.

In terms of future work, an obvious immediate step would be to model different shapes to find which ones each basis is suited for. Another avenue to investigate is which way to implement the moved measurements method and how that interacts with the shape being modelled.

It would also be good to look at different data structures to try to find an optimal one for representing wavelet bases and coefficients.

As a longer term goal, it would also be interesting to investigate different spherical wavelet bases.

~9:46