Summer Undergraduate Research Opportunities

Emile Okada

University of Cambridge

July, 2016

1 Week 1

1.1 Reading

I started the week reading Chapter 1 section 2 of Charles L. Epstein's "Introduction to the Mathematics of Medical Imaging". It covered how to reconstuct a 2D convex object from the shadows of an object. If $h(\theta)$ is the shadow function as described in the book (essentially the distance of the support line in direction $(-\sin(\theta), \cos(\theta))$ from the origin), then the convex hull can be parameterized by

$$(x(\theta), y(\theta)) = h(\theta) \cdot (\cos(\theta), \sin(\theta)) + h'(\theta) \cdot (-\sin(\theta), \cos(\theta)). \tag{1}$$

We can extend this idea to 3D by considering slices of the object. Fix some vector **v** and then consider the collection of planes perpendicular to **v**. In each of the planes one can use the 2D method to construct a 2D convex hull of the intersection of the object with the plane. Stringing all these 2D slices together then gives a rough reconstruction of the 3D object from its shadows.

I also spent some time reading up on the TV transform and scale spaces, to get a rough idea of which project I'd like to do. I ended up going with the tomography project, but spend roughly 1.5 days doing reading for the other project.

1.1.1 Remaining tasks

The above method only allows for 'slicewise convex' reconstructions. While better than convex, this doesn't ustilize all the data available from the shadows e.g. see the rabit ears below. We know the two ears are two separate 'blobs' and we can tell this from the shadow, but this information is nonetheless lost in the reconstruction. I will try to find ways on how to improve on this (so that e.g. if we scan a human we can see two legs rather than one large blob).

1.2 Coding

On Thursday I started coding. To have some examples to work with I used the Wolfram Mathematica ExampleData function to obtain a 3D model of a rabit.

Figure 1: 3D model of a rabit from Mathematica ExampleData.

I then took its orthogonal projection and binarized the image to obtain shadows of the rabit from 10 different angles.

Figure 2: Shadow of rabit with $\theta = 0$.   Figure 3: Shadow of rabit with $\theta = \pi/2$.

I then finished writing the code for reconstructing the object from its shadows in python (see page 17 or my github) on Friday.

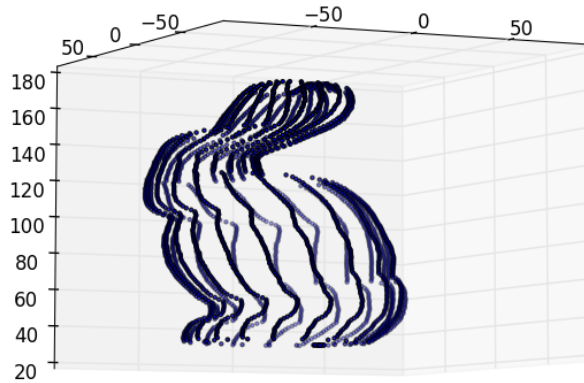Figure 4: Reconstruction of the rabit using the script on page 17.

1.2.1 Challenges

I had some problems originally with taking the derivative of the shadow function since the input was a bit noisy. To fix this problem I convolved the data with a discrete gaussian (binomial distribution) to smooth out the noise. This worked well with the example data. However, it remains to see whether it will work well enough for the actual real life pictures which will probably be a lot more noisy.

1.2.2 Remaining tasks

There are currently a lot more data points than I need. At the moment I'm calculating the shadow function fow each row of the image matrix. This results in a lot of points which causes the rendering to take some time. I can probably get away with a lot fewer points in the z direction. I also plan on adding a polygonal mesh so it looks a bit nicer. Lastly, I should probably also do some

smoothing in the z direction to fix the sudden cutoffs that tend to occur when the variations in the 3D object are smaller than the 'resolution' in the z-direction.
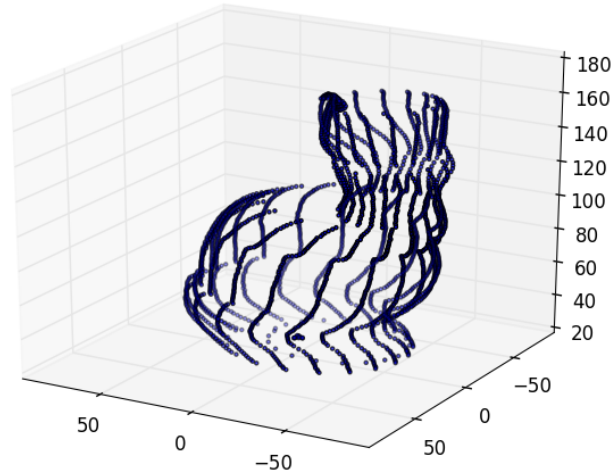

Figure 5: Cutoff of rabits ears. Can probably be fixed by smoothing the data in the z direction.

1.3 Building

I haven't begun building yet, but I now have a rough idea of the the set should look like. At the moment I'm thinking of creating a simple turn table (rotating platform) on which a person stands. I'll then place some bright lights in front for them and a big white sheet of paper behind on which to shine their shadow. A camera is then placed behind the paper to capture the shadows.

1.3.1 Remaining tasks

I'll create a proper design this week and make a basic prototype on a smaller scale hopefully by next week.

2 Week 2

2.1 Reading

This week I had a chat with Martin and Matthias on how to resolve objects like rabit ears (i.e. disjoint unions of convex objects). They suggested a 'back projection' method that is a generalization (or in some sense a generalization of the 'dual') of the method from week 1. The methods can be thought of as follows: imagine standing inside a circle with walls around. Every, say, $\pi/8$ radians there is a light source that creates a shadow diagonally opposite. Now extend each shadow out of the wall to the opposite side, to create 'cylinders' with cross sections with the shape of the shadow. We then intersect these cylinders to get the reconstruction. In some sense this method utilizes all the information available and can distinguish between legs, rabit ears and other disjoint unions

of convex blobs. We also don't need to differentiate anything with this method which is always nice.

2.1.1   Remaining tasks

Read up on TV smoothing for 3D surfaces to make smoother reconstructions.

2.2   Coding

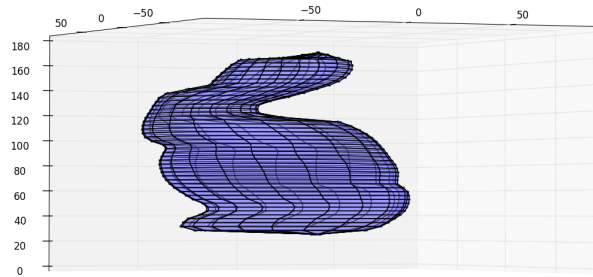Before I implemented the new method I added a polygonal mesh to the original method.

Figure 6: Original reconstruction with polygonal mesh added.

I also implemented the new method. It works by reducing it to the 2D version of the problem and doing the reconstruction by slices like the previous method. It is slightly more computationally expensive (the 'cyclinders' are represented in arrays and the intersection is done by pointwise multiplication). However, there results are a lot better.
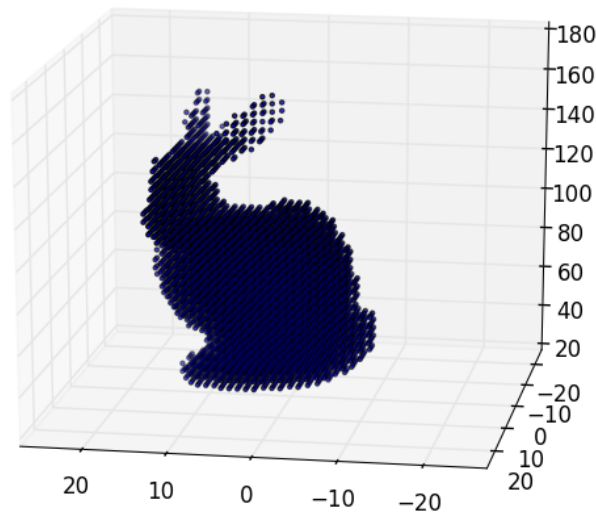
Figure 7: Reconstruciton of rabit with two disjoint ears.

I also implemented a method that only displays the boundary of the reconstructions by taking the morphological perimeter of each slice.
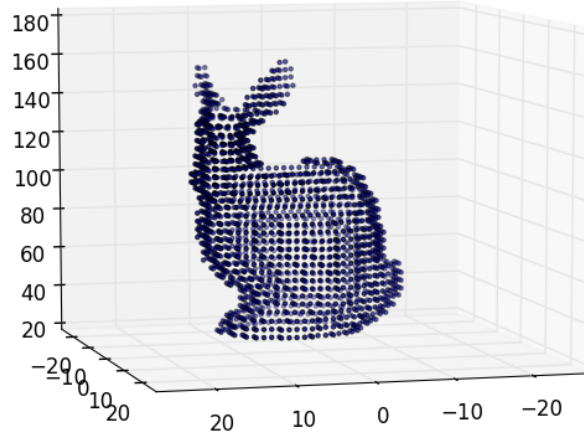

Figure 8: Boundary of rabit reconstruciton.

2.2.1   Remaining tasks

It would be good to add a mesh to this method as well and maybe get some lighting to make it look more realistic. Also need to implement TV method, but will probably wait to do that until I have something that works first.

2.3   Building

I've done some reseach on turntables for humans. I've found quite a few DIY turntables that are cheap and relatively straightforward to make (see this, this and this). I also looked for some ready made ones, but the ones I found were either too expensive (âĆň650) or made for mannequins so can't take the full load of a human. It seems like the best option would be to make my own. I'll contact the engineering department to ask if there is any way I could use some of their tools (possibly under the supervision of an engineering friend) to make the turntable myself. I've also looked at lighting and it seems quite cheap so shouldn't be a problem. Hopefully I'll start ordering the equipment this week.

3   Week 3

3.1   Reading

This week I read up on ROF denoising and the TV transform. In particular I read the Chambolle & Pock paper. I then extended the ROF denoising algorithm to 3D 'images' in the obvious way.

3.2   Coding

In preparation for the 3D version of the TV denoising algorithm I implemented the 2D version as outlined in the paper.
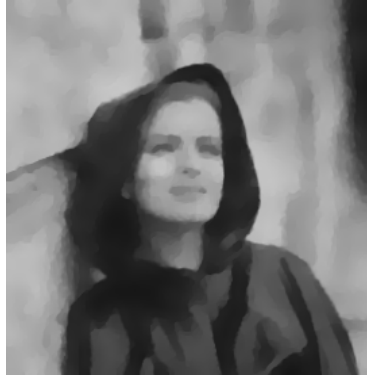

Figure 9: Look, less noise! (with very few iterations since python is slow...)

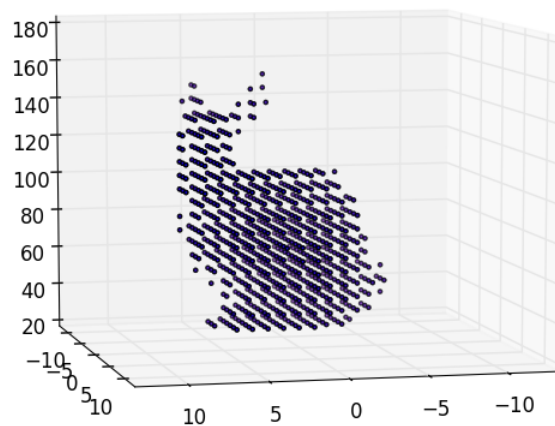I then implemented the 3D version. Here is a sample output:

Figure 10: Denoised rabit. It doesn't look very different however since the input data didn't have much noise to start with. This is good!

3.2.1   Remaining tasks

I should probably make the rabit look solid rather than a cloud of dots. I should also test the algorithm on some noisy data as well. Maybe I can add in some facny lighting and fun statistics about water content and whatnot (for the purpose of the science fair). I think I'll also reimplement the code in Matlab since python is super slow.

3.3   Building

I set up a meeting with the dyson center in the engineering department this week. We might be able to use their spaces to construct the scanner. We discussed how to make the turntable in particular. One of the designs we looked at however seemed somewhat difficult to make and would probably require us to hire the workshop to make it. I'll thus look more into the simpler looking turntable.

3.3.1   Remaining tasks

I need to make some accurate drawings for the turntable. I'll need exact measurements etc. and also I need to find the parts I need. Then I can make a cost estimate and then meet with the people in the engineering department again to discuss feasiblity.

4   Week 4

My parents came to visit over the weekend + Monday and left on Tuesday so didn't get as much done as I'd like. I also spent Thursday + Friday preparing the presentation for the group meeting.

4.1   Building

On Tuesday I visited the Dyson Center to ask about the designs and using their space. One of the designs for the turntable that I had found was probably a bit beyond my technical skills so I decided to go for the simpler design. While there they kindly explained the key parts of the design and what a bearing was. They also said they had to double check that it was ok that we used space in the Dyson Center since we were from the maths department. Hopefully they will get back to us soon.

I've also looked more closely at what I need to do buildingwise and started looking up parts. A rough upper bound on the cost of the turn table is £100. This includes the cost of plywood, wood glue, bearings, motor, and the computer components (bluetooth component and arduino nano to control the motor remotely). Note that this is an upper bound and can probably be brought down maybe £30-£40.

The shadow screen will probably be quite a bit cheaper. A 2m PVC pipe costs roughly £3. Trow in the cost of some joints and the fabric (this is the most expensive part, maybe £25, but I'll look for some cheaper alternatives) and the total can be roughly bounded above by £50.

The last remaining part is the light. I noticed that the light from a projector is very strong and might be able to do the trick. If not, I found some outdoor lights that cost £8, so I just need something to mount them on.

4.2   Remaining tasks

This week I need to start ordering the parts. Then hopefully I can start building something next week.

5 Week 5

This week I was helping out with a maths summer school in the mornings (9am-11am) since I'm mostly just waiting for parts to arrive.

5.1 Reading

This week Martin explained to me an alternate way to perform the smoothing on the 3D reconstruction.

The idea is to solve

$$\hat{u} = \operatorname*{argmin}_{u}\{\mathrm{TV}(u) \text{ s.t. } SR(u) = 0, u \in [0,1]\} \tag{2}$$

where R is the discrete Radon transform and S is a subsampling operator which depends on the input image (it ensures that the Radon tranform of u stays 0 at the same places as for the inital images).

To solve this problem we can use the Chambolle Pock framework by letting

$$F(y_1, y_2) = \chi_{\cdot=0}(y_1) + \|\|y_2\|_2\|_1$$
$$G(x) = \chi_{[0,1]}(x)$$
$$K = (SR, \nabla)^T$$

5.1.1 Remaining tasks

I'll need to actually implement this. Doesn't seem like too much work, but will wait till I've actually made the 3D-scanning device first.

Anothering that I need to do is find a robust way to generate a polygonal mesh of the reconstructed object. I can't seem to find an easy way to do this.

5.2 Building

Over the past week I begun ordering all the parts we need. The main difficulty has been in obtaining furniture grade PVC fittings in the UK. The DYI market is large in the US, but quite small in the UK. The only seemingly decent supplier in the UK however is currently not in operation due to a wedding of some sorts (?) (it is a very small company). I we've thus resorted to ordering the fittings from Germany, and the pipes from the UK.

Otherwise, we now have the fabric, wood glue, Lazy Susan bearings, extension cords, webcam, motor and printer.

It remains to order some of the electronic circuit components, plywood and possibly a usb extension cable.

I also went to the Dyson center to get some advice. Here are scans of the sketches that Dr. Roebuck made while explaining how to make the electric ciruit and where to get plywood.
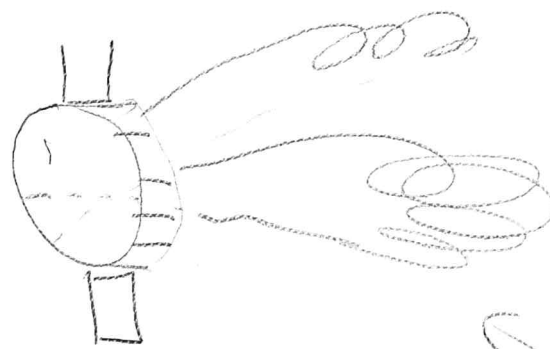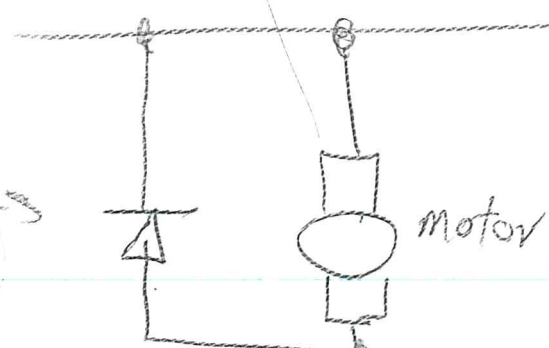
Brushed DC motor ← tractional ✓

worm

$I \propto torque$

modern is brushless ✗

12v high current supply

freewheel high current diode →

motor

Parallel port digital output line

DO?

BUZ11

up to about 10A

acting as a controlled switch

gnd

0v for high current supply

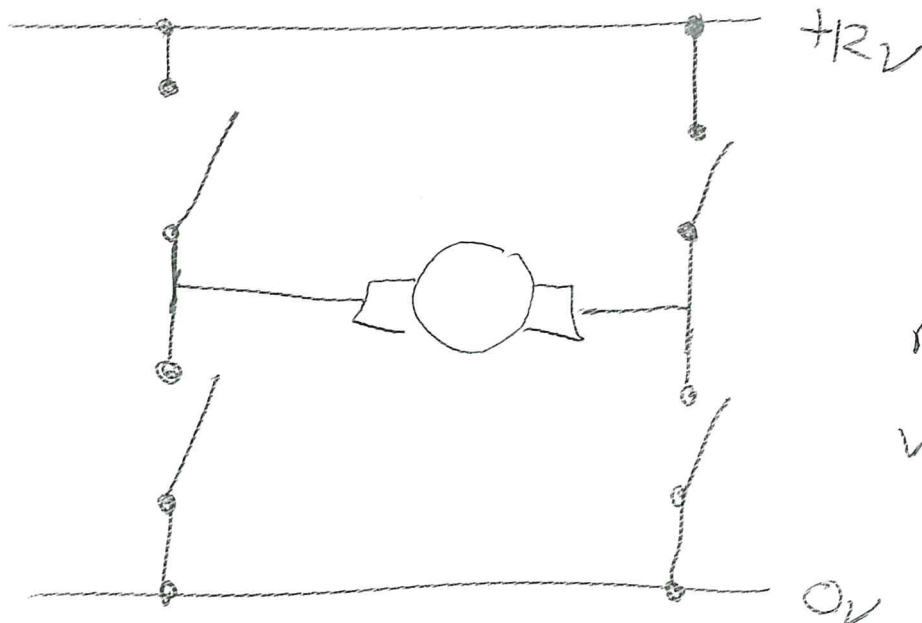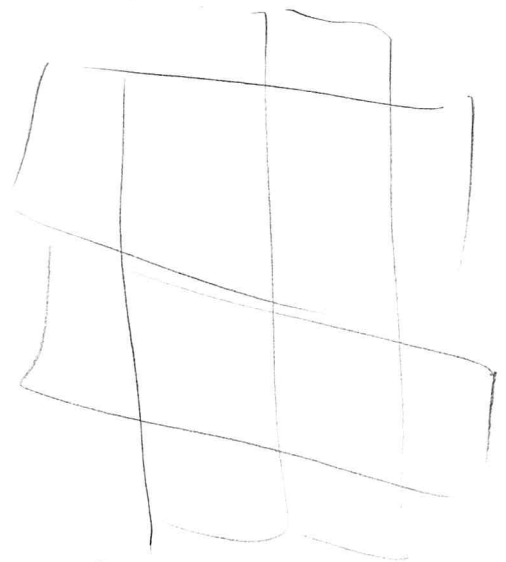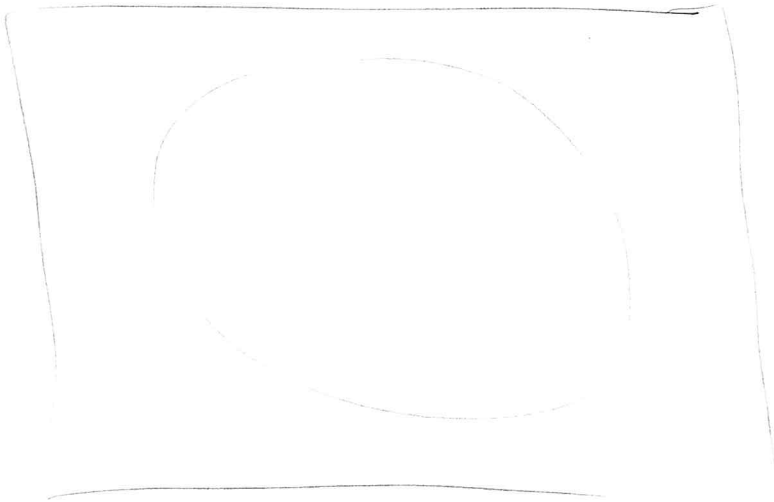~~https://~~ ~~hs~~ hobarts.com

http://hpclaser.co.uk

450mm wide max, 9mm thickness max

12mm laser ply

600x900 or 600x600

www.technologysupplies.co.uk

+12v

0v

Microswitch (with roller)

rswww.com

www.farnell.co.uk

Parallel port - direct to mosfet

6   Week 6

6.1   Reading

This week I've read up a bit on electronic circuits. After talking to Dr. Roebuck, I got some advice on what sort of components I'd need to control the motor using a computer. As such I've read up on MOSFETs, parallel ports and microswitches. The parallel port allows me to connect the circuit to the computer, while the MOSFET acts as a switch to turn the motor on an off. Lastly, the microswitch can help keep track of how many degrees the turntable has tunred by counting the numer of teeth that have passed by.

6.1.1   Remaining tasks

I'll need to find a way to undo the warping caused by the wide angle webcam.

6.2   Coding

After talking to Matthias, it turns out matlab has an 'isosurface' function which allows me to create a surface around the reconstructed object and add lighting and shadows. As such I've been migrating my python code to matlab. The other reason I'm doing this is because matlab seems to be quite a bit quicker with dealing with large array's of the type I'm working with. I've migrated rougly half of the code now and intend to finish by the end of this week.

6.3   Building

The PVC pipes have now arrived and we're only waiting for the fittings from Germany now. It seems like the muslin cloth for my backdrop is taking longer than anticipated to arrive so I might have to send an email to enquire. The plywood has also been ordered so I should be able to start cutting it by midweek. The motor is also here now, so I only need to order the remaining electronic components. All that remains is dealing with the lighting. I've emailed a friend who is an engineer at a lighting company in Cambridge to ask for his advice. He had previously mentioned that we might be able to inherit some spare lighting or at the very least get some tips on what kind of strength we need to get a decent shadow in a standardly illuminated room.

6.3.1   Changes

I've made some modifications to the design of the shadow screen. With the new wide angled webcam, I can place the webcam roughly 1m away from the screen. As such I intend to adjoin an additional structure to the screen to ensure that the webcam stays at a fixed position relative to the screen. This way I can also hopefully correct for the distortions caused by the wide angle lens since I know precisely the distance between the webcam and the shadow screen.
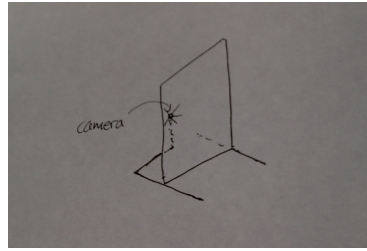
Figure 11: Mounting of the webcam.

6.3.2   Remaining tasks

Other than obviously assembling the setup, I need to determine a proper power supply. The engineering department has kindly allowed me to borrow one of their variable power supplies so that I can determine what kind of current I need. Thus once the turntable is functional and tested I can buy a fixed current power supply.

7   Week 7

7.1   Coding

This week I finished rewriting all the code in Matlab. I also added some new functionality. The program now creates a surface mesh and adds lighting to the reconstruction.
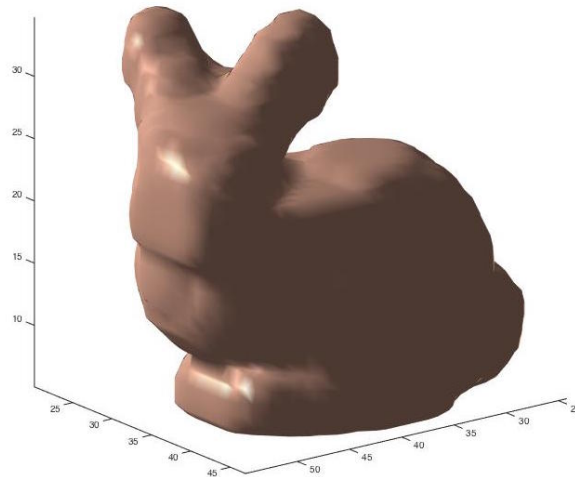

Figure 12: Reconstructed rabit with surface mesh and lighting.

A higher resolution can be obtainted by lowering the 'xy_scaling' parameter in the 'recon' class, but the tradeoff is that it becomes quite a bit slower.

7.1.1   Remaining tasks

Create an interface which people at the science fair can use. In other words, a start button to rotate the person and then an interface which allows them to view the object from different angles.

7.2   Building

The frame of the shadow screen was completed this week. The pvc pipes were cut into two 2m pieces, two 1.5m pieces, one 1m piece, one 0.75m piece and four 0.5m pieces. Assembly takes roughly 1min so is very convenient. The finished frame looks like
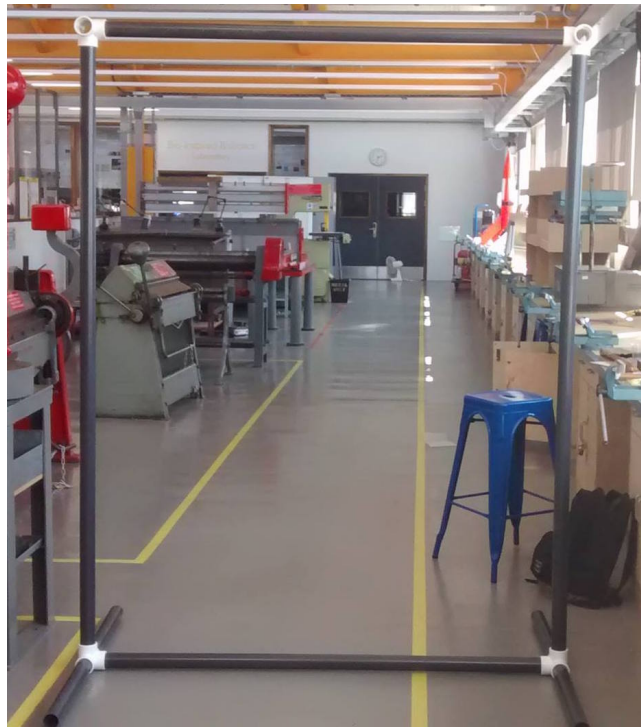

Figure 13: Frame of shadow screen.

The fabric also arrived and came in one 2m by 1.5m peice. To attach the fabric to the frame I decided to sew on elastic cloth as 'handles'.

One early problem that I found was that the fabric was letting too much light in. While the shadow was nice and clear on the side that was illuminated, from the back it got washed out by the weaker ambient lighting.

Figure 14: The shadow from the front view is nice and clear...

Figure 15: ... however the current fabric is too transparent so the camera picks up the ambient background lighting as well as the actual object.

To try to fix the problem I tried folding the fabric to see if increasing the thickness helped improve clarity.

Figure 16: No folds lets in too much light...

Figure 17: ... two layers seems to block a lot of the ambient background lighting

Figure 18: ... three layers blocks out too much of the intended lighting as well.

Two folds seems to give the optimal clarity so I've purchased another 2mx1.5m peice of fabric. Another problem that has turned up is that it is difficult to get the shadow of the feet to show up on the shadow screen. It seems like to fix this we'd need another light source at a lower height, but then we'd get two shadows which isn't really a very good idea. I think at this point it seems like a wiser idea to restrict the reconstruction from the waist up (ideally we'd have a large raised platform and a parabolic light reflector, but time, skill and monetary constraints render this infeasible).

In other news, there's been progress on the turntable. The cogs have been cut out. It only remains to 3D print the smaller cog, to lubricate the bearings, get the motor working and then put together the circuit.

Figure 19: The turntable cog.

8   Code

8.1   3D reconstruction from shadows

```
classdef recon
    properties
        im_dir;
        files;
        height;
        width;
        mid;
        vertical_spacing = 5;
        vertical_resolution;
        shadow_bands;
        xy_scaling = 5;
        scaled_width;
        scaled_mid;
        teeth = 20;
        increments;
    end
    methods
        function obj = recon(im_directory)
            obj.im_dir = im_directory;
            obj.files = get_files(im_directory);
            [obj.height, obj.width] = ...
                size(rgb2gray(imread([im_directory,obj.files{1,1}])));
            obj.mid = round(obj.width/2);

            obj.scaled_width = round(obj.width/obj.xy_scaling);
            obj.scaled_mid = round(obj.scaled_width/2);
            obj.vertical_resolution = ...
                floor(obj.height/obj.vertical_spacing);

            [obj.increments, ~] = size(obj.files);
            obj.shadow_bands = cell(obj.increments,obj.vertical_resolution);

            for i = 1:obj.increments
                img = double(rgb2gray(...
                    imread([im_directory,obj.files{i,1}])))/255;
                img = round(img);
                for j = 1:obj.vertical_resolution
                    obj.shadow_bands{i,j} = ...
                        read_bands(obj,img(j*obj.vertical_spacing,:));
                end
            end

        end
        function theta = increment2theta(obj,i)
            theta = 2*pi*obj.files{i,2}/obj.teeth;
        end
```

```matlab
function bands = read_bands(obj,row)
    d = diff(row);
    left_edges = find(~(d+1))-obj.mid;
    right_edges = find(~(d-1))+1-obj.mid;

    if length(left_edges)>length(right_edges)
        right_edges = [right_edges,length(row)];
    elseif length(left_edges)<length(right_edges)
        left_edges = [1, left_edges];
    end
    if length(left_edges) == length(right_edges) ...
            && ~isempty(left_edges)
        bands = [left_edges; right_edges]';
    else
        bands = [];
    end
end
function is_in = in_band(obj,theta,m,n,bands)
    d = -(m-obj.scaled_mid)*sin(theta)+...
        (n-obj.scaled_mid)*cos(theta);
    [h,~] = size(bands);
    for i = 1:h
        if bands(i,1) <= obj.xy_scaling*d && ...
                obj.xy_scaling*d <= bands(i,2)
            is_in = true;
            return
        end
    end
    is_in = false;
end
function mat = intersected_area(obj, z_coord)
    mat = ones(obj.scaled_width,obj.scaled_width);
    for i = 1:obj.scaled_width
        for j = 1:obj.scaled_width
            for k = 1:obj.increments
                if ~obj.in_band(obj.increment2theta(k),...
                        i,j,obj.shadow_bands{k,z_coord})
                    mat(i,j) = 0;
                    break;
                end
            end
        end
    end
end
function vol = volume(obj)
    vol = zeros(obj.scaled_width,obj.scaled_width);
    for i = 1:obj.vertical_resolution
        vol(:,:,i) = obj.intersected_area(...
            obj.vertical_resolution-i+1);
    end
```

```
        end
    end
end
```