

---

## ENGINEERING TRIPPOS PART IIA

### GG2 CT reconstruction and visualisation

Final Report

**Name:** Michael Hutchinson, mjh252

**College:** Christ's College

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Reconstruction</b>	<b>1</b>
2.1	Initial reconstruction . . . . .	1
2.1.1	Scan calibration . . . . .	2
2.1.2	Hounsfield unit calibration . . . . .	2
2.2	Connection of the two scans . . . . .	3
2.3	P-FDK implementation . . . . .	3
2.4	Comparison of Parallel reconstruction to approximated P-FDK . . . . .	4
2.5	Investigations into filter types and $\omega_{max}$ . . . . .	4
<b>3</b>	<b>Visualisation</b>	<b>6</b>
3.1	Producing the 3D model . . . . .	6
3.2	Multiple segmentation . . . . .	6
<b>4</b>	<b>Animation and posing</b>	<b>7</b>
<b>5</b>	<b>3D printing</b>	<b>7</b>
<b>6</b>	<b>Discussion</b>	<b>7</b>
6.1	Advantages and limitations of CT methods . . . . .	7
6.2	Discussion of working in groups . . . . .	8
<b>7</b>	<b>Results</b>	<b>8</b>
<b>A</b>	<b>Reconstruction</b>	<b>9</b>
A.1	P-FDK code . . . . .	9
A.2	Investigations into filter types and $\omega_{max}$ . . . . .	10
<b>B</b>	<b>3D prints</b>	<b>10</b>

### Abstract

With the modern trend of using more and more CGI in films, and in particular the rise of using computer stand ins for actors, either to digitally change ages actors, perform stunts, or even bring them back from the dead to film a prequel, we decided to investigate using CT scanning technologies to create the 3D models required to produce these CGI effects. To do this we needed good, high definition CT reconstructions of the character, to extract a 3D model from this CT reconstruction, and finally to animate this 3D model. We also looked at 3D printing sections of clothing/armour for the action man figure that were form fitting, as proxies for tailored costumes and props that might be created for actors from scans of their bodies.

# 1 Introduction

The aim of this project was to investigate the viability of using CT scan data to generate 3D models for use in animation in films. The use of computer graphics as stand ins for people has become a common trend, replacing people for dangerous stunts, changing the age or appearance of actors, or animating characters for whom the actors are no longer with us. The process for acquiring these models is usually either done with a large number of photographs, or short range laser range-finding equipment. Both require specialist equipment, not easily available. CT detectors, while not simple machinery are widely available. Another advantage presented by CT data for this purpose is that we can see inside the person, and can get a picture of the underlying bone structure and musculature. This may help with animation of key, tricky to animate parts, such as the face or hands.

In order to evaluate this, we would need process the raw CT data to get high quality phantoms of the subject, from this, produce a high quality 3D model of the subject, and then take this and animate it. The work was split into the three sections as described and divided between the three group members. This division was not strictly stuck to, and we helped each other out where needed.

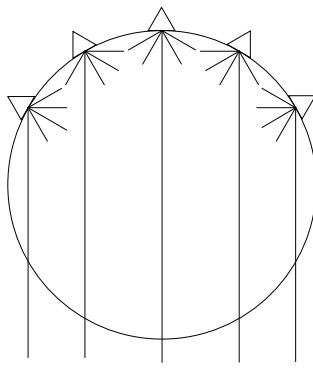
## 2 Reconstruction

This section of the report contains work done only by myself.

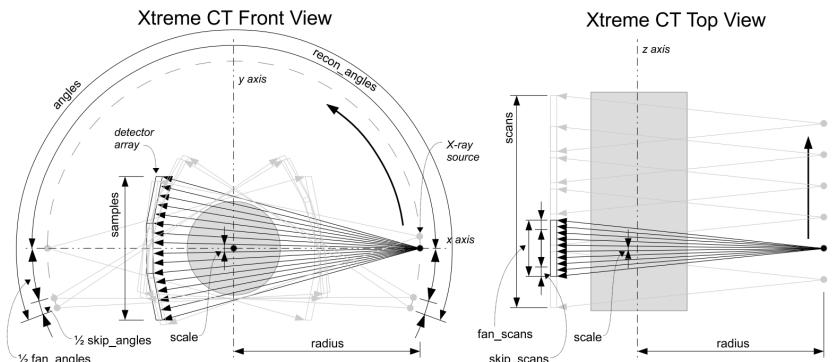
### 2.1 Initial reconstruction

The initial reconstruction of the xtreme ct data of the action man was done using the assumption that all the beams in the vertical plane are parallel. This assumption is reasonable as the detectors have small angles between relative to the source. This assumption is readdressed later on using the P-FDK method.

This reconstruction is very similar to the reconstruction process used in the previous part of the project and follows from Turbell 2001 [1] chapter 2.2 on filtered back propagation. The most important difference here is that the beams from the source in plane with the reconstructions are not parallel. One possible solution to this problem is to convert our fan beam sinogram into a parallel beam sinogram. Figure 1 shows how by considering beams from a number of different fans at different angles this is possible. This transformation involves a coordinate transform, dependant on the detector layout and interpolation between points on the sinogram. Turbell gives examples for this transform for both planar and circular detectors. The Xtreme CT machine has a more complicated 3 flat section geometry, seen in figure 2, take from the handout.



**Figure 1:** Conversion of fan beam sinogram to parallel sinogram



**Figure 2:** Layout of the Xtreme CT detector

The function `xtreme_fan_to_parallel` does this operation for the more complicated 3 straight section detector in the xtreme ct scanner. it computes:

$$p^P(\theta, t, s) = \text{xtreme\_fan\_to\_parallel}(p(\gamma, \beta, s))$$

$p(\gamma, \beta, s)$  is the sinogram with fan beams,  $\beta$  is the angle between the central beam and some reference,  $\gamma$  is the angle between the central beam and the beam for the sinogram sample,  $p^P(\theta, t, s)$  is the new sinogram with parallel beams,  $\theta$  is the angle between the central fan and some reference in the plane of rotation of the fan,  $t$  is the sample distance along the sinogram, and  $s$  the sample height of the sinogram.

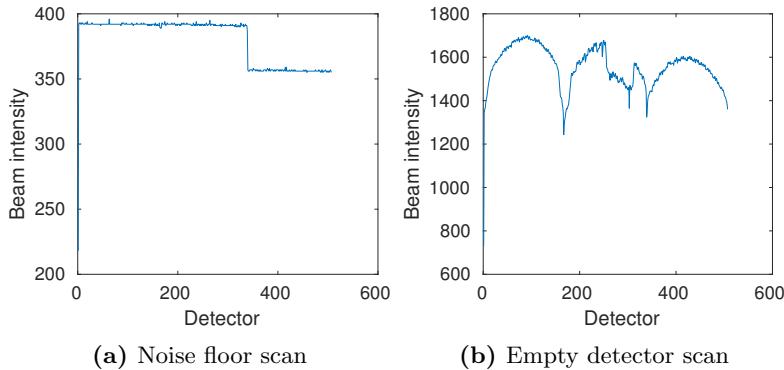
For a given slice, the reconstruction follows the procedure: Load the sinogram for the slice, along with the calibration data. Calibrate the scan using the calibration data. Convert the sinogram to parallel form. Filter the sinogram e.g. with a Ram-Lak filter. Perform back-projection.

### 2.1.1 Scan calibration

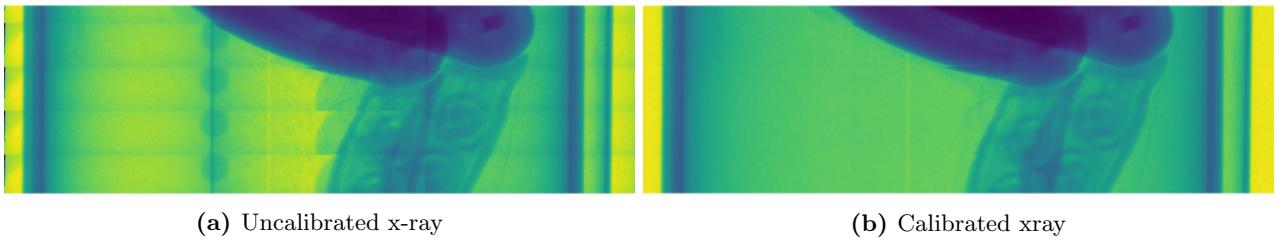
The calibration procedure for the sinograms from the xtreme scanner is similar to that before. The 2 pieces of calibration data we have are the noise floor and an empty scan of the detector. The noise floor tells us how much background radiation the detectors sees without any beams on. This is subtracted from both the actual scan and the empty scan to get the detector values without background radiation in them. The scan of the empty detector minus the noise floor gives us the total energy of the unattenuated beam,  $\sum_E I_{0,E}$  from equation 4 in the handout. Therefore:

$$\mu_{tot} = -\log \frac{I_{tot}}{\sum_E I_{0,E}} = -\log \frac{scan - scan\_min}{scan\_max - scan\_min}$$

Plots of example noise floors and empty scans can be seen in figure 3. The humped shape of the empty detector scans is due to the shape of the detector. Parts of these sections are further from the beam source, and so the fan has spread further, reducing the intensity of the beam, resulting in lower readings. This correction is very important, otherwise we would see bands of higher and lower attenuated material in the sinograms that do not exist. This is demonstrated in figure 4 which shows a re-sampled scan from a side elevation, much like an x-ray scan, with and without the calibration applied.

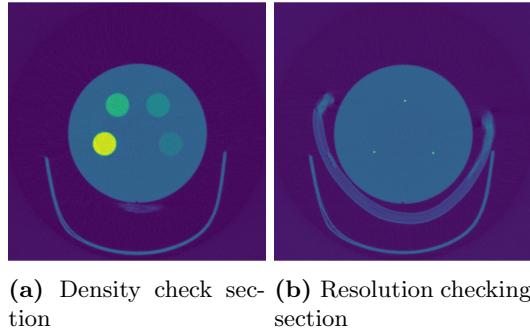


**Figure 3:** Example noise floor and empty detector scans



**Figure 4:** Calibration demonstration

### 2.1.2 Hounsfield unit calibration



**Figure 5:** Calibration object

The last part of this reconstruction is to calibrate the scans to Hounsfield units. The provided scan of a calibration unit was used to do this. The calibration block contains 4 rods of material of different densities, surrounded by perspex. There is also a resolution checking section to the object that contains 3 thin wires. The easiest way to calibrate the scanner is to use the attenuation value of the perspex holding the materials. Perspex has

a similar Houndsfield units value to water, and so is used as the zero point. The reconstructed values of the perspex were averaged over a large volume and this value used as the attenuation coefficient of water in equation 17 from the handout for calculating Hounsfield units. Figure 5 shows the calibration object in cross section.

3 main materials were found to make up action man. These were found by plotting a histogram of the Hounsfield for slices and looking at the results. Peaks were seen at 2000 and 2400, being the harder 2 plastics that made up his head and skeletal structure. The softer material covering his arms, shins and chest had a value of about 300. his clothes had a value very similar to that of air, or the noise in the detector, as the are mostly comprised of air in the fabric. It was very hard to distinguish these from the air.

## 2.2 Connection of the two scans

When being scanned, the action man figure did not fit fully into the scanning region of the xtreme ct machine. We therefore have 2 scans of action man and in order to produce a full reconstruction we must join these up. The 2 sets are neither aligned in rotation nor translation.

To align them, a function to rotate and translate slices, and a script to ovelay these were used. The correct slice position to join the 2 sets was found in 3D Slicer and the rotation/translation found by tuning them in matlab. Examples of the overlay functions output can be seen in figure 6.



(a) Poor phantom alignment (b) Good phantom alignment

**Figure 6:** Overlay examples

## 2.3 P-FDK implementation

In the previous section the assumption was made that along the axis rotation of the scanner, the beams of the fan are parallel to the planes of reconstruction. This is a good approximation as the angles between beams are not very large. If a shorter throw on the beams were used (this would increase the angle between beams), or if very precise scans are needed then this must be corrected for. An approach to doing this can be found in Turbell 2009 [1] section 3.3.1, the P-FDK method. In this we consider a third dimension to the sinogram,  $b$ , the z height of the sinogram. The equations for P-FDK vs Filtered back-projection to calculate a single (x,y,z) value are:

### Filtered back-projection

$$\begin{aligned} f(x, y, z) &= \int_0^{2\pi} \tilde{p}^P(\theta, t(\theta, x, y), s(z)) d\theta \\ t(\theta, x, y) &= x \sin(\theta) - y \cos(\theta) \\ s(z) &= z \end{aligned}$$

### P-FDK

$$\begin{aligned} f(x, y, z) &= \int_0^{2\pi} \tilde{p}^P(\theta, t(\theta, x, y), s(\theta, x, y, z)) d\theta \\ t(\theta, x, y) &= x \sin(\theta) - y \cos(\theta) \\ s(\theta, x, y, z) &= \text{a function dependant on the detector geometry} \end{aligned}$$

The main difference is the inclusion of a variable function  $s(\theta, x, y, z)$  in the integral. This samples from different height sinograms when computing  $f(x, y, z)$ , rather than just the sinogram at the height of the reconstruction plane. This is what accounts for the correction for angle of the fan beams. It should also be noted the in Turbell 2009, when calculating  $\tilde{p}^P(\theta, t, b)$ , which is the filtered parallel sinogram, there is included a pre-weight term, such that:

$$\tilde{p}^P(\theta, t, b) = \text{preweight}(t, s)p^P(\theta, t, s) * g^P(t)$$

This term accounts for the different path lengths of the beams. This was not included as it is very small and makes little difference in our case. For setups with wider fans it would be important to include this term.

A visual way of understanding the P-FDK algorithm is to look at the paths of the fans and the reconstruction layers. This can be seen in figure 8a. An analogous method to the back-projection method can be drawn from this.

In the back-projection method, to reconstruct a slice for a set of x and y coordinates, for each angle we "smear" the sinogram values at that angle back over the reconstruction image and add the contributions at

each point (interpolationg as we have a finite number of beams). In P-FDK, due to the spreading out of the beams, the planes in which we wish to reconstruct slices are not coincident and parallel to paths of the beams. In order find the correct values, we must consider more than one sinogram per angle to reconstruct the phantoms. If we consider a whole fan, we can reconstruct all the slices bounded within it. Similarly to before, but in 3 dimensions, we can "smear" the sinogram back along the path of the beam it took from source to detector, and then interpolate values for the slices we wish to reconstruct at the correct  $x$ ,  $y$  and  $z$  coordinates. Adding these up for each angle scanned we get the reconstructed phantoms.

To do this, we must find where for a given  $(x, y, z)$  in the reconstruction the beam passing through this point would lie on the sinogram. To do this 2 coordinate sets are defined.  $(x, y, z)$  is aligned with the reconstruction images, with  $z$  being vertical height,  $x, y$  are centred at the centre of the object.  $(t, v, s)$  defines a set of coordinates locked to the beams of the detector.  $s$  is vertical height, zeroed on the central fan, and coincident with  $z$ .  $t$  is perpendicular to the direction of the beams, and  $v$  the distance along the beam, away from the source centred on the centre of the object. These are seen in figure 7. Also considered in this diagram is the fact that the virtual sources in the parallel mapping lie on a circle. The distance therefore, from a source to the  $v = 0$  plane changes with the  $t$  coordinate. This distance is equal to  $\sqrt{R^2 - t^2}$ , where  $R$  is the radius of the fan's path.

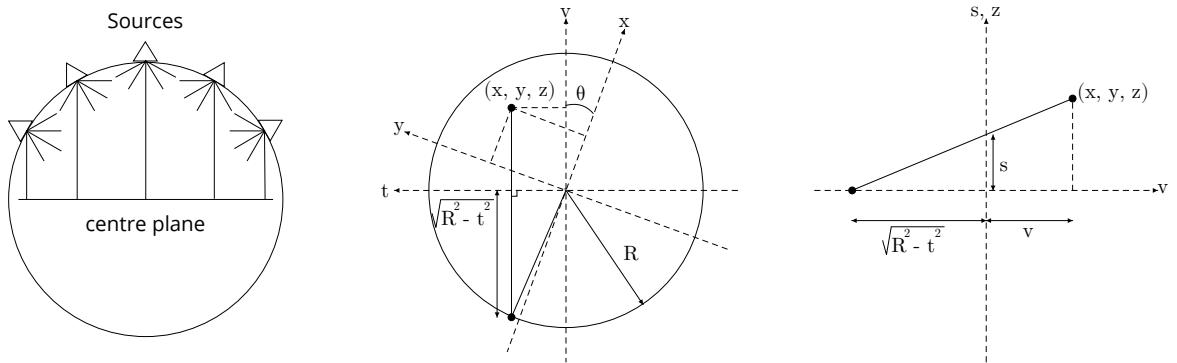
The transformation between systems is:

$$t(x, y, z, \theta) = y \cos(\theta) - x \sin(\theta), \quad v(x, y, z, \theta) = x \cos(\theta) + y \sin(\theta), \quad s = z$$

Using this information, we can for a given  $(x, y, z)$  find the point at which the beam connecting this point to the detector it came from crosses the  $v = 0$  plane. We know that at this plane, the beams from the sources to the detectors intersect the plane to form a regular grid, spacing  $h.\text{scale}$ . Therefore we can use the coordinate of the beam from  $(x, y, z)$  passing through the  $v = 0$  plane to find the point on the sinogram we should sample.

For point  $(x, y, z, \theta)$  we therefore sample from:

$$\tilde{p}^P(x, y, z, \theta) = \tilde{p}^P(\theta, t_{\text{plane intercept}}, s_{\text{plane intercept}}) = \tilde{p}^P(\theta, t, \frac{\sqrt{R^2 + t^2}}{\sqrt{R^2 + t^2} + v} z)$$



(a) Top view of the set up, showing the parallel beams passing through a grid at the origin  
(b) View of the coordinate systems along the  $z, s$  axes  
(c) View of the coordinate systems along the  $t$  axis

**Figure 7:** Coordinate systems for the beams and reconstruction

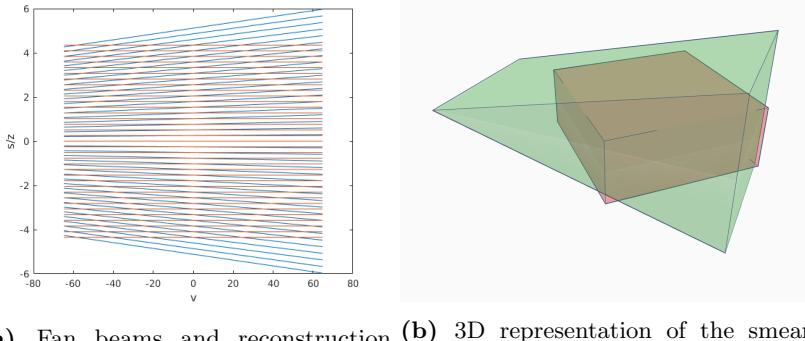
The code for computing this can be found in the appendix, listing 1. To reconstruct the full action man (1024 slices) it takes about 25 minutes with this algorithm.

## 2.4 Comparison of Parallel reconstruction to approximated P-FDK

Slices reconstructed from near the centre of fans should look relatively similar in both Parallel reconstruction and P-FDK as the beams from the fan are almost flat. At the edges of the fan, they should differ as the fan beam are angled. Figures 9 and 10 show this. In figure 11 we can see that in the parallel reconstruction, ridge artefacts occur due to the incorrect slice reconstruction at the edges of the fans. We can that these are removed using the P-FDK method.

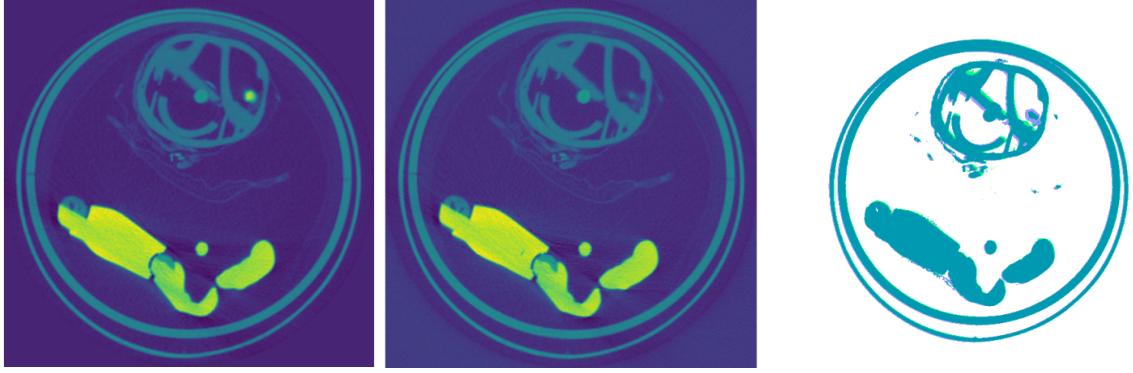
## 2.5 Investigations into filter types and $\omega_{max}$

Filters other than the Ram-Lak and raised cosine filters were also considered for the filtering of the sinogram data. Figure 15 (appendix) shows a comparison of a number of filters. Overall thy all to some extend provided



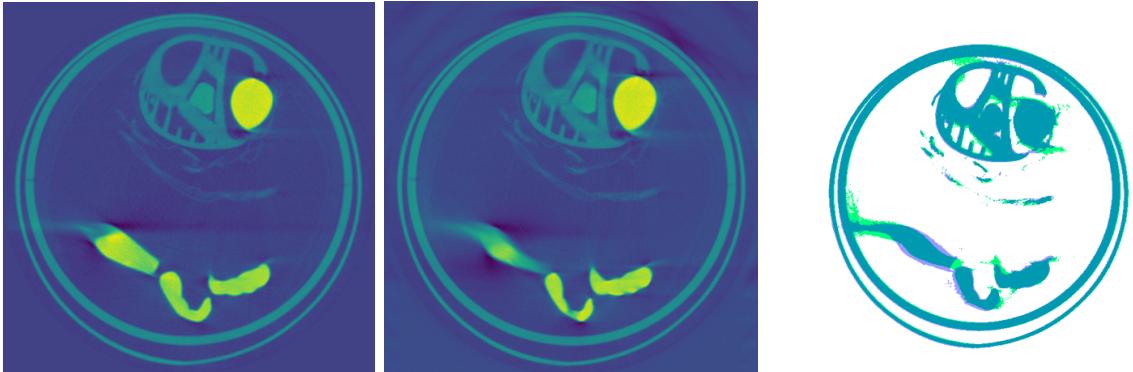
(a) Fan beams and reconstruction slices (b) 3D representation of the smear and reconstruction volumes for an angle of  $45^\circ$

**Figure 8:** Fan beams and reconstruction slices



(a) Middle of fan parallel reconstruction (b) Middle of fan P-FDK reconstruction (c) Overlay of (a) and (b)

**Figure 9:** Comparison of Parallel and P-FDK reconstructions at the edge of a fan

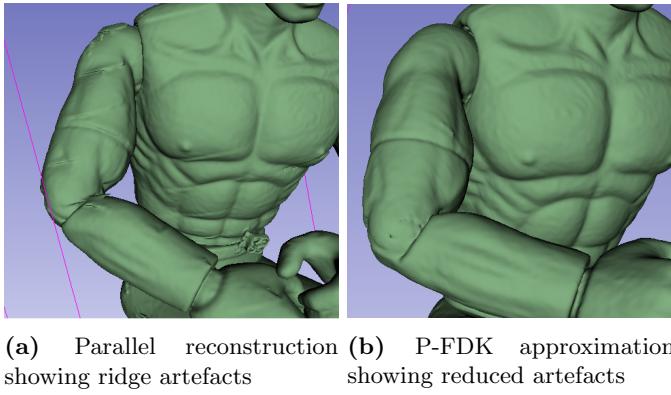


(a) Edge of fan parallel reconstruction (b) Edge of fan P-FDK reconstruction (c) Overlay of (a) and (b)

**Figure 10:** Comparison of Parallel and P-FDK reconstructions at the edge of a fan

noise reduction at the price of detail. For 3D modelling, these made little difference so none were used.

The maximum frequency allowed through the filtering process was also investigated. This corresponds to setting frequencies above a give  $\omega_{max}$  in the filter used to filter the data to 0 and adjusting any filter multipliers(e.g. raised cosine) to the new band of frequencies. This will clearly cut out high frequencies. The main result of this will be a smoothing effect. This could be advantageous if there is high frequency noise in the data, such as random noise on the detectors, as this will be smoothed out. A downside of this is that we will loose detail on sharp edges and small scale features. A series of different  $\omega_{max}$  were tried and a selection can be seen in figure 16 (Appendix). In order to loose minimum fine detail but still eliminate some noise a  $\omega_{max}$  of 0.45 of the maximum sampling frequency or greater was chosen.



**Figure 11:** Ridge artefact comparison

### 3 Visualisation

This section contains work done mostly by another team member, but I had some involvement with the segmentation process.

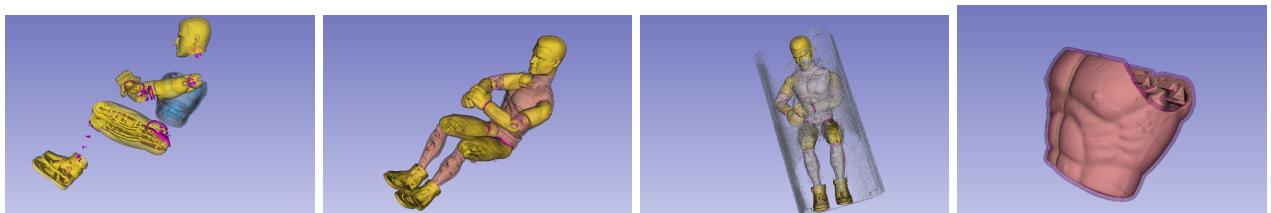
After the reconstructions had been made, the next step was to produce a 3D model of the action man from the dicom data. We looked at a number of tools for doing this. The main objective here was easy segmentation for the sections we wanted, easy clean up of model, as the clothes action man is wearing are difficult to remove in threshold segmentation, and easy export of STL files for printing and importing to modelling programs. For these reasons 3D Slicer was chosen. Another advantage of 3D slicer is the cross platform nature of it, so members of the team working on different OS's could all use the same program to prevent confusion.

#### 3.1 Producing the 3D model

In order to produce the 3D model we used the segmentation editor in 3D slicer. First we used the threshold settings to only select areas containing a certain range of dicom values. This selected all of the action man, but also the walls of the container and some parts of the clothes that action man was wearing that appear as small blobs. Using the scissors tool we remove the larger part of the outer cylinder, and the "Keep largest island" tool to remove most small bits of noise and the rest of the container. Next we wanted to fill in the insides of the body and limbs. We only need the outer surface to model, and adding the inside adds significant numbers of faces, which increases rendering complexity, and reduces responsiveness of programs for no benefit. To do this we inverted the segmentation, which gives us the selection of all the "air". We then removed all islands except the outside one and reverted the selection to get a model without any of the insides being hollow. Some manual clean up was required to separate the inside from the outside at this point as some parts of the walls of the action man had holes in. Finally small bits of noise connected to the model still were removed with the scissors tool. The results of this segmentation can be seen in figure 11.

#### 3.2 Multiple segmentation

We also segmented out tighter ranges of dicom values, letting us pick out the individual materials the the action man was made of, as well as his individual limbs. This might be more useful in more complex animation, or in medical purposes. This can be seen in figure 12 Having produced this model, we then exported it as an STL



(a) Dense plastic parts forming the structure of the action man  
 (b) Additional soft covering that makes up the chest and shins  
 (c) Clothes being partly picked out, along with a lot of noise  
 (d) A chest piece designed to fit perfectly to the action man

**Figure 12:** Comparison of multiple segmentation

file to be used in blender, a 3D modelling program.

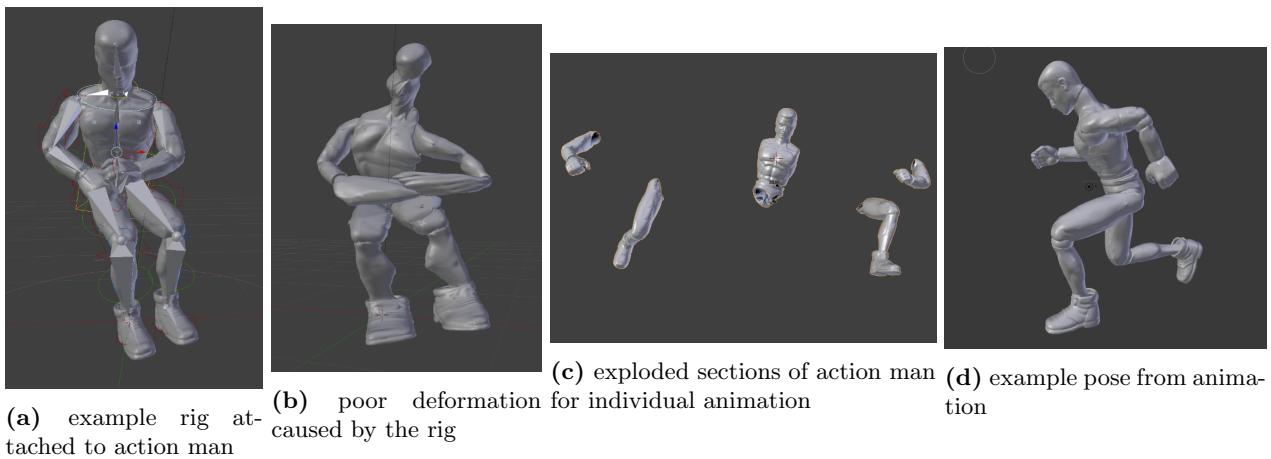
## 4 Animation and posing

This section contains work done by team member.

The software selected to do the 3D modelling in was blender. It is a free to use piece of software with a gentler learning curve than some others, and as we were new to modelling this was good.

2 main methods were attempted for animation. The first was full rigging of the model. This involves supplying a set of "bones" which will then attach to the object and deform the mesh as you move them. We attempted this method but ran in to serious difficulty with getting the the correct vertices to attach to the correct bones. Upon attempting to deform the mesh something like figure 13. The second method was to simply split apart the mesh into limb sections, and add sphere to replace the joints. We could then animate the parts using key frames to set the positions of parts at different times. This proved much easier, though slower than rigging to do the actual animation. With more time we believe we could get the rigging to work.

Before doing the animation on the model we applied a mesh decimation to the model. The original face count for the model was bout 3,000,000, extremely high. This caused significant slow down using programs to model this. Mesh decimation intelligently reduces the face count of the model and we found we could go as low as 300,000 faces with no perceptible difference. We could go down as far as 50,000 with only minimal detail loss as well.



**Figure 13:** comparison of animation methods

## 5 3D printing

We also 3D printed a number of sections of the action man in different poses from the 3D model and animation that we had. Also produced was a real version of a prop used in the animation, namely the parachute vest used in the short animation. Using this method, you could create actual real props for parts animated in CGI, smoothing the change between animated sections and section filmed in reality. Examples can be seen in figure 17 (appendix).

## 6 Discussion

### 6.1 Advantages and limitations of CT methods

There have been 2 main restrictions on getting good CT throughout the project. The first is the presence of beam hardening. This was investigated more in the first half of the project, but in short causes 'streaks' of lower attenuation areas to radiate from spots of higher attenuation. This can have significant effects on the reconstruction of the CT data, regardless of the number of views or angles the subject is observed from. While it is possible to see through the artefacts, it can mask details and will cause errors in extraction of 3D structures from the data, e.g. blood vessels from tissue. This is important as it forms a large part of medical diagnostics. A number of modern techniques are being developed to combat this. Dual energy CT scanners image the patient at 2 different energy x-ray spectrum, and use the difference in how they are attenuated to remove beam hardening. Photon counting detectors can measure the energy of photons hitting them, and using this we can see the energy spectrum of the beam hitting the detector. This also allows us to remove beam hardening.

Another limitation is the the number of samples and angles we take scans of. This is limited to prevent overdosing patients with radiation. Again investigated more in the previous report, a minimum number of angles is needed to prevent aliasing in the reconstruction and this is very important for clarity. The same

applies for samples, and with more samples we can also get better resolution in the reconstruction, important for looking at small details such as blood vessels or early stage cancers.

3D printing from medical data has huge implications and already has a number of highly useful applications. Examples included printing low cost prosthetics, custom fitted to people, possibly using CT reconstructions to take a model of people, tailor making replacement components such as joints, heart valves, cranium replacements, or ear cartilage which requires internal 3D models of that patient, and provide massive improvements over the previous state of the art for treating these problems.

## 6.2 Discussion of working in groups

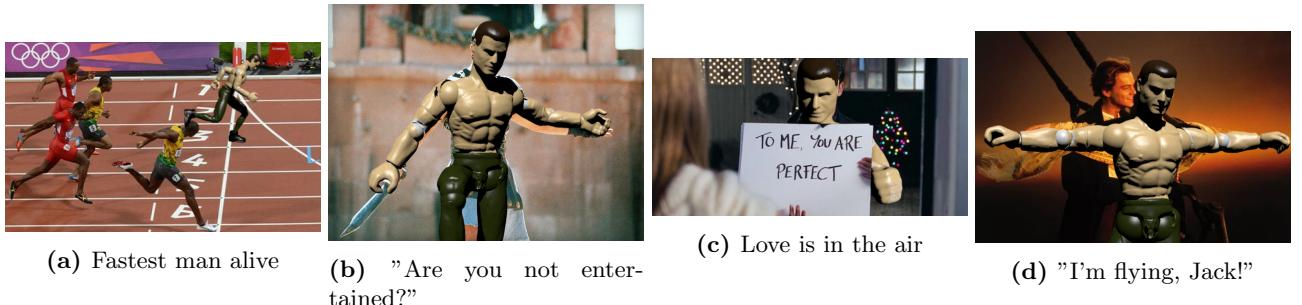
Over all our group functioned well together. In the first two weeks when we were working on very similar material we would each tackle the tasks separately to begin with and then discuss issues and difficulties together if we had them. This system worked well in letting everyone learn the material, but also ensuring that we could move through it swiftly. When writing the group report use of an online latex editor and complier allowed us to quickly compile a report as a group with all three working on it at the same time.

In the second two weeks we began working on separate tasks, although there was some overlap to balance workloads through time, and for interests sake. As much as possible we worked in the same physical space to allow us to discuss problems, and keep our ourselves informed on what the others were doing, allowing us to understand, if not be capable of, doing what the others were doing.

The downsides of working in a team were the added organisational issues, alleviated by very similar schedules, the co-storage of data and code, mitigated with a google drive and github, and the reduction in the individuals knowable about completing the tasks as a whole, mitigated as above. The biggest advantage is how much more the group could achieve in a two weeks, first by simply have three times the hours, and second by allowing individuals to task specialise, be it at computing reconstructions or learning how to 3D model from scratch in 2 weeks. Overall in this group project, the advantages of being a team heavily outweighed the disadvantages and allowed us to complete a project we were proud of in such a short space of time.

## 7 Results

After creating a system to allow us to animate the action man model we had produced we, create a short animation demonstrating the viability of this technique. This can be found [here](#) and [here](#). We aslo edited the model into a number of famous film stills, in figure 14. This demonstrates the viability of this technique for acquiring 3D models for animation. An advantage over the standard methods using photographs and LIDAR are the inclusion of bone and muscle structure in the scans that might lead to better animation of the subject, especially in intricate parts such as the face. The main disadvantage is the radiation dose the subject receives, and the potential loss of some surface quality due to the limits on the radiation dose.



**Figure 14:** Edited stills from films using our model

## References

- [1] Henrik Turbell. *Cone-beam reconstruction using filtered backprojection*. PhD thesis, 2001.

## A Reconstruction

### A.1 P-FDK code

```
1 function [recon] = xtreme_reconstruct_p_fdk(h, slice_stop, alpha)
2 run_time = tic;
3 % compute relevant factors
4 scale          = h.scale;
5 radius         = h.radius * h.scale;
6 samples        = h.samples;
7 scans          = h.scans;
8 fan_scans     = h.fan_scans;
9 fan_recon_scans = h.fan_scans - 2*h.skip_scans;
10 recon_angles  = h.recon_angles;
11 width_2       = (h.samples-1)/2 * h.scale;
12 height_fan_origin_2 = (h.fan_scans-1)/2 * h.scale;
13 height_layer_origin_2 = (h.fan_scans -2 * h.skip_scans - 1)/2 * h.scale;
14 % produce grid spaces for sample and transforming
15 [TT, SS] = ndgrid(linspace(-width_2, width_2, samples), ...
16                    linspace(-height_fan_origin_2, height_fan_origin_2, fan_scans));
17 [xx, yy, zz] = ndgrid(linspace(-width_2, width_2, h.samples), ...
18                    linspace(-width_2, width_2, h.samples), ...
19                    linspace(-height_layer_origin_2, height_layer_origin_2, fan_recon_scans));
20 % empty block to populate
21 recon = zeros(samples, samples, (ceil(slice_stop/fan_scans)*(fan_recon_scans)));
22 total_fans = ceil(slice_stop/fan_scans);
23
24 for fan=1:min(total_fans, floor(scans/fan_scans))
25     fprintf('=====\\nNew Fan: %d\\n\\n', fan)
26     fan_time = tic;
27 % start position og the fan in the scans
28     fan_pos = (fan-1) * fan_scans + 1;
29 % stop if end reached
30     if(fan_pos < slice_stop)
31         % load and process relevant sinograms for this fan
32         recon_block = zeros(samples, samples, fan_recon_scans);
33         data_block = zeros(recon_angles, samples, fan_scans);
34         for g=1:(fan_scans)
35             if(fan_pos + g <= scans)
36                 [slice, slice_min, slice_max] = xtreme_get_rsq_slice(h, fan_pos+g);
37                 slice = xtreme_calibrate(slice, slice_max, slice_min);
38                 slice = xtreme_fan_to_parallel(h, slice);
39                 slice = ramp_filter(slice, scale, alpha);
40                 data_block(:, :, g) = slice;
41             end
42         end
43
44         angle_time = tic;
45         for a=1:h.recon_angles
46             % transofrm coordinates for this angle
47             p = a*pi/recon_angles;
48             tt = xx*cos(p) - yy*sin(p);
49             vv = yy*cos(p) + xx * sin(p);
50             % calculae distance from source to v=0 plane
51             dist_to_source = sqrt(radius.^2 -tt.^2);
52             % compute s coordinate of v=0 plane crossing for given points
53             zz_scale_factors = (vv + dist_to_source)./dist_to_source;
54             ss = zz./zz_scale_factors;
55             % interpolate values
56             sinogram_function = griddedInterpolant(TT, SS, squeeze(data_block(a, :, :)), 'linear', 'none');
57             Y = sinogram_function(tt, ss);
58             Y(isnan(Y)) = 0;
59             % add to the reconstruction image
60             recon_block = recon_block + Y*(pi/h.recon_angles);
61             if(mod(a, 10) == 0)
62                 fprintf(1, 'Angle %d \\t\\tTime Remaining %.2f \\n', a, (toc(angle_time)/a) * ((total_fans-fan)
63             end
64         end
65         fprintf('Fan %d took %02g seconds\\n\\n', fan, toc(fan_time))
66         % calibrate block
67         recon_block = xtreme_hu(recon_block);
68         % insert into the reconstruction
```

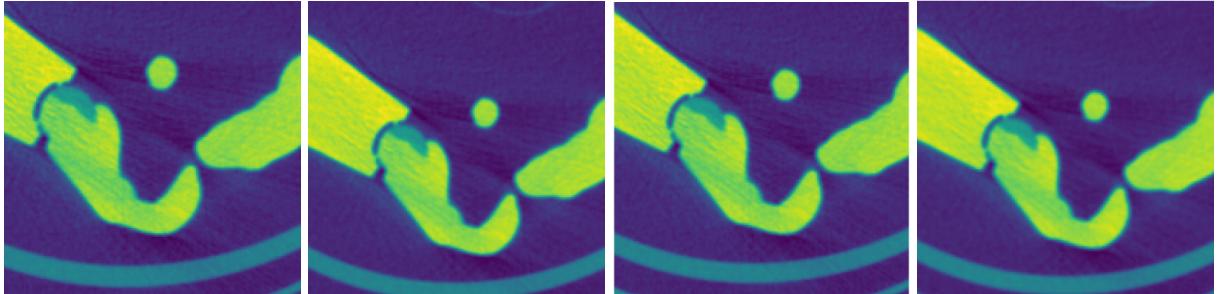
```

69     recon(:,:,((fan-1)*(h.fan_scans-2*h.skip_scans)+1):((fan)*(fan_recon_scans))) = recon_block;
70
71 end
72 fprintf('Reconstruction took %04g seconds to do %d fans and %d slices\n', toc(runtime), floor(h.scans/h.fan_
73 end

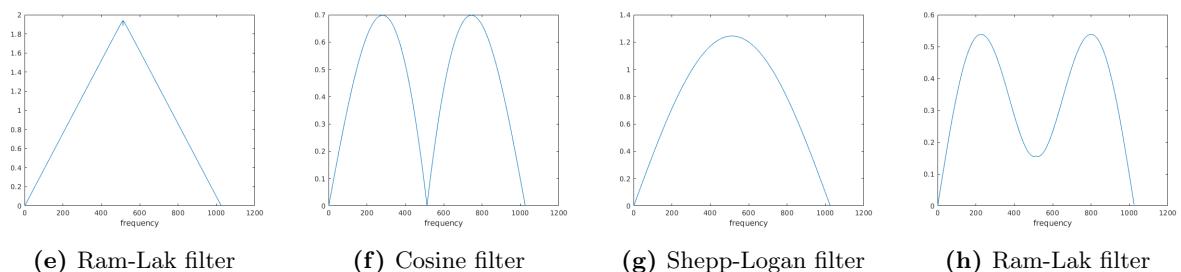
```

**Listing 1:** Function to calibrate sinograms

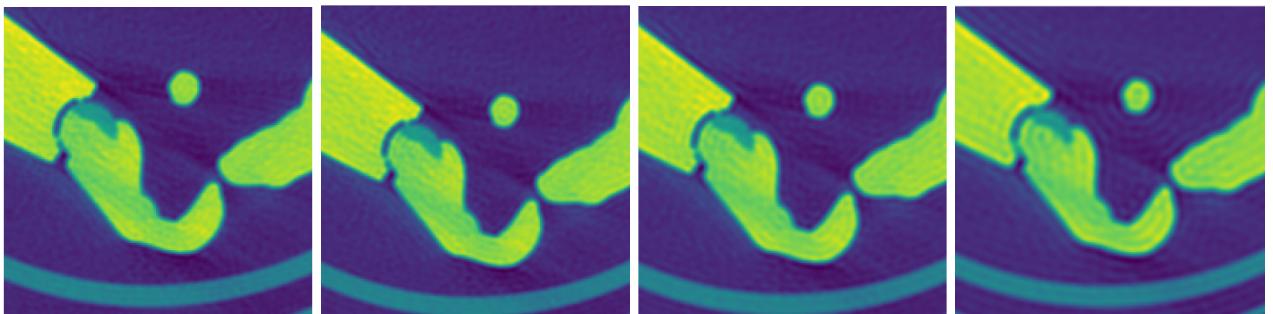
## A.2 Investigations into filter types and $\omega_{max}$



(a) Ram-Lak filtered reconstruction    (b) Cosine filtered reconstruction    (c) Shepp-Logan filtered reconstruction    (d) Hamming filtered reconstruction



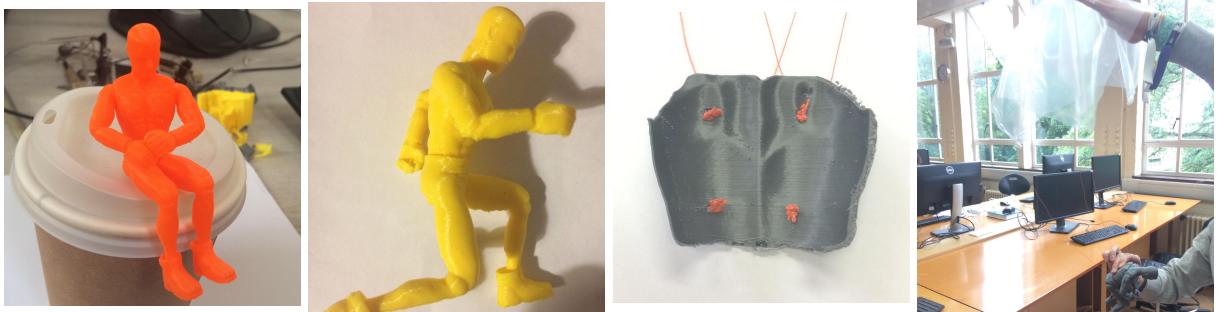
**Figure 15:** Comparison of filters on reconstruction



(a)  $\omega_{max} = 0.5$  of maximum    (b)  $\omega_{max} = 0.45$  of maximum    (c)  $\omega_{max} = 0.4$  of maximum    (d)  $\omega_{max} = 0.3$  of maximum

**Figure 16:** Comparison of  $\omega_{max}$  values

## B 3D prints



(a) Action man in a sitting pose    (b) Action man in a running pose    (c) Parachute pack from the animation printed out    (d) Parachute pack attached to action man

**Figure 17:** Example 3D prints