Object Reconstruction Using Shadows (Computed Tomography)

Tejas Penugonda , Harshul Chand , Jiaqi Chen , Ivan Wong

Department of Physics, Imperial College London

OBJECTIVES

- Explore the mathematics behind computer tomography using visible light a safer alternative to X-Rays.
- To demonstrate that an object's shape can be reconstructed using only its shadows.
- To automate the data collection process.
- Successfully create a digital model suitable for 3D printing.
- To develop a method for capturing shadows from multiple angles.

INTRODUCTION

3D reconstruction can be hard to visualize but it can be broken down simply. By placing a 3D object between a light and a screen— we can generate a shadow with its edge in the form of a closed loop. If you trace lines from each point of that curve back to the source, you create a cone-like surface that defines a volume enclosing the object. Rotating the setup and repeating this process from different angles gives multiple volumes. Their intersection then gradually reveals the true shape of the object. Simple.

VISUALIZING THE 2D CASE

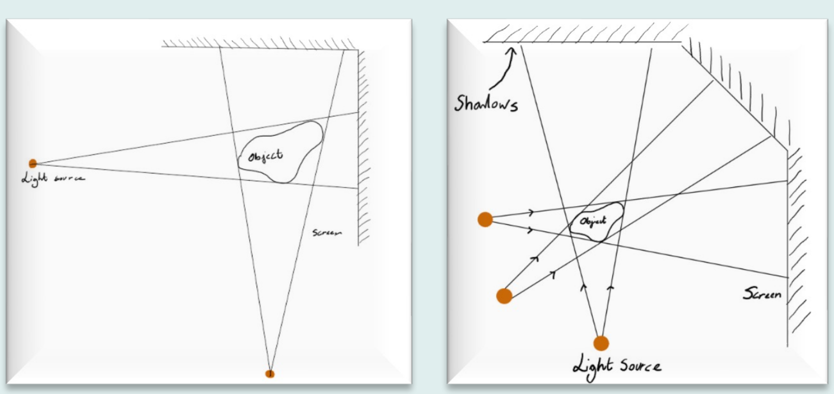


Fig1. It is then possible to see that the greater the number of angles taken the closer the intersecting area will be to the full volume. This is extendable to 3D.

2D EXAMPLE

Drop a dimension and consider an irregular 2D object. Now, is there a way to determine its outline using only a light source and a screen? Place the object in front of a light source and observe the shadow it casts on a screen. This shadow will appear as a straight line.

A crucial property of the shadow is that if you trace lines from the edges of the shadow back to the light source, these lines will always fully enclose the object that cast the shadow. Since the object lies within each pair of traced lines, anything outside them can't be part of the object. Repeating this from different angles and intersecting all regions reveals where the object must be.

ANALOGY

Think of it like starting with a massive block of wood and trying to carve out an object. Each new volume you generate allows you to "cut away" everything outside of it. With each new angle, more parts of the block that cannot contain the object are removed.

Repeating this process again and again, we continue shaving away the excess, and the remaining shape gets closer and closer to the original object we're trying to recover.

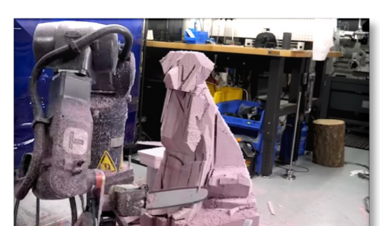


Fig 3. Sort of like the dog the youtuber "Stuff Made here" made.

DATA COLLECTION

To capture many images (in our case, 800), we used a stepper motor synchronized with the frame rate of our camera.

The stepper motor moves in small, discrete steps (which can be assumed to be instantaneous) and then waits for a short period—this delay is denoted by ***d***, and we can set its value. Another adjustable parameter on the stepper motor is the total number of steps per revolution.

In our setup, we used approximately 3200 micro-steps per full rotation—so fine that, with the instruments we had, the motion appeared completely smooth and continuous.

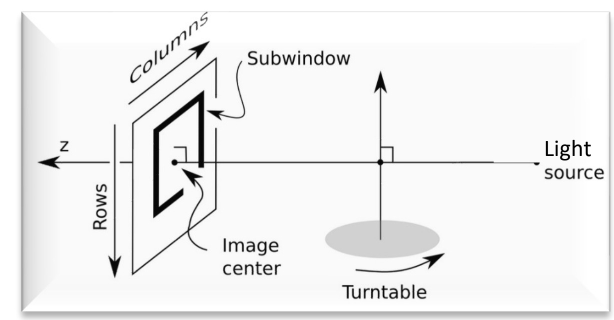


Fig 3 . Rotating both the light source and the screen was highly impractical for our setup, so instead, we chose to rotate the object itself. The object was placed on a turntable, which was positioned between a light source and a screen.

Since videos are recorded by rapidly taking images, we can then record 1 complete rotation of the turn table, and we would then equally spaced apart images! For example, Imagine a camera recording at 60 fps. If we program the motor to complete one full rotation in 2 seconds, the camera will capture 120 images at evenly spaced angles.

TECHNICAL DETAILS

- We calculated the delay d so that the motor completes one full revolution while the camera captures exactly 800 frames using the following formula described in **Fig 4.**
- Although we assumed that each step takes zero time, each micro-step takes about 50 microseconds—which is still so short that no frames are missed.
- The turntable doesn't spin continuously in a strict sense, but with 3200 micro-steps per revolution, the motion appears extremely smooth.

$$d = \frac{k}{f \cdot 3200}$$

Fig 4. k is the number of desired frames (800), **f** is the frame rate of the camera (in fps)

DATA PROCESSING

Method

- The method we use to extract information from shadows is called FDK, after Feldkamp, Davis, and Kress. It has three steps: weighting, filtering, and back projection. The last step reconstructs the object, but the first two prepare the images for processing.

Goal

- Our goal is to explore the math behind CT scanning and understand scanning errors. One problem is that objects far from the rotation axis change their distance from the light as they spin, making their size appear to grow and shrink. This means overlapping volumes might not always mean the object is truly there.

Solution

- To fix this, we divide the whole volume (like the original wood) into small cubes. Each cube gets a score every time it appears in the reconstruction. Cubes near the rotation center get higher scores because they suffer less distortion, so their position is more reliable.
- Filtering reduces noise in the data, making images clearer—like special effects in movies.
- We repeat these steps for many angles, add the scores, then remove cubes with low scores. This leaves an approximate shape of the object.

FDK PIPELINE

Weighting → **Filtering** → **Back Projection**

- **Weighting:** Assign weights proportional to cos(**θ**) to each point.
- **Filtering:** Convolve the weighted images with the filter function, W(x).
- **Back Projection:** Find the sum of all weights and reject all values lower than a threshold

STEP 1: WEIGHTS

A brief description of the set-up is also displayed in Fig. 6.

The weight is calculated using the following formula shown below.

$$W(u,v) = \frac{D}{\sqrt{D^2+u^2+v^2}} = \cos\theta$$

STEP 2: FILTERING

Filtering is done using the following equation:

$$H(|\omega|) = |\omega|\, W(|\omega|/\omega_c)$$

We multiply the Fourier transform of the projection function (the weighted shadow image) with a filter function defined as:

$$P_w \xrightarrow{\text{FFT}} \mathcal{F}\{P_w\}(\omega)\ \cdot H(|\omega|)$$

$$\xrightarrow{\text{iFFT}} q(\beta,u,v).$$

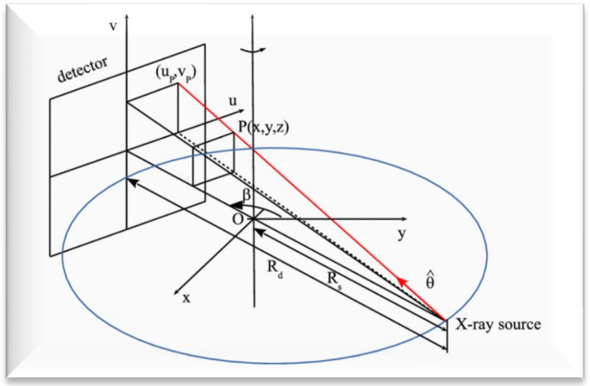
Window	$W(x)$
none (Ram-Lak)	1
Shepp–Logan	$\dfrac{\sin(\pi x/2)}{\pi x/2}$
Cosine	$\cos(\pi x/2)$
Hamming	$0.54 + 0.46\cos(\pi x)$
Hann	$0.5 + 0.5\cos(\pi x)$
Parzen	$(1-x)^3$

Fig 5. Common ramp×window filters. Multiplying by a tapered window damps noise and truncation spikes at the cost of resolution.

STEP 3: BACK PROJECTION

- This step rebuilds the 3D image from all the filtered X-ray projections.
- For every angle of the rotating X-ray source, the data is projected back into the image space.
- Each voxel in the volume is updated based on how it maps onto the detector at that angle.
- A correction factor is applied to account for the widening of the cone-shaped beam.
- By combining all these contributions, the internal structure of the object is reconstructed.

$$f(\mathbf{x}) = \frac{1}{2\pi}\int_0^{2\pi} \frac{D^2}{L(\beta,\mathbf{x})^2}\, q(\beta, u(\beta,\mathbf{x}), v(\beta,\mathbf{x}))\, d\beta$$



[2]

Fig 6: Cone-beam CT setup (phone-CT). A point source follows a circular trajectory of radius Rs = R = DSO, emitting a diverging cone of rays through the object onto a flat detector. The detector plane remains orthogonal to the source–object line for every view **β**.

ASSUMPTIONS OF FDK

- 1. Circular orbit. Complete 360◦ (or 180 deg+ fan, satisfying Tuy's Sufficiency Condition) coverage; no axial translation.
- 2. Planar, centered detector. Tilt or offset produces "crescent" artefacts.
- 3. Small cone angle. Central slice is exact; accuracy decays with height.
- 4. Uniform angular sampling. N**β** ≳ 400–1600 avoids view aliasing (Δ**β** ≲ du/D).

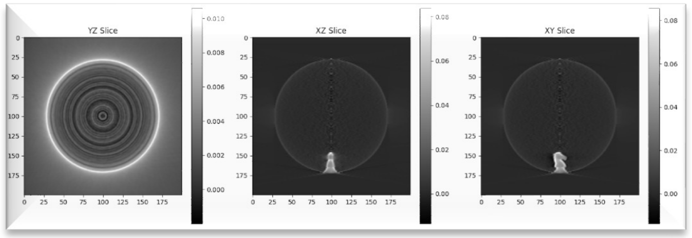
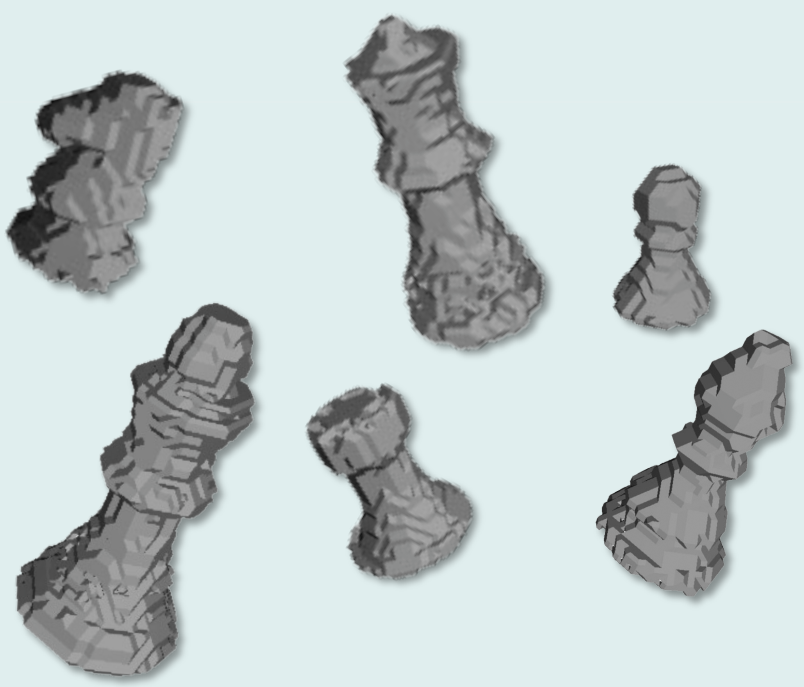


Fig 7. Images achieved after weighing and filtering (Shepp-Logan)

RESULTS

To handle common scanning challenges—like off-axis distortion and noise—we carefully refined our method. We introduced a scoring system for small volume elements (voxels), with extra weight given to those near the rotation axis to account for warping. We also applied filtering to reduce noise and sharpen features, while being mindful of artifacts like truncation walls caused by edge padding.

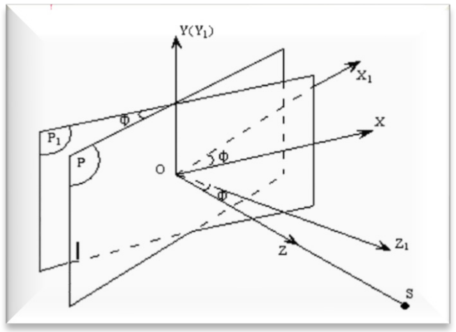
By combining data from multiple angles, applying mathematical processing, and managing reconstruction artifacts, we successfully generated a detailed and accurate 3D approximation of the chess piece. Along the way, we not only explored the technical process behind CT imaging but also gained insight into the mathematical foundations and real-world limitations of digital reconstruction.



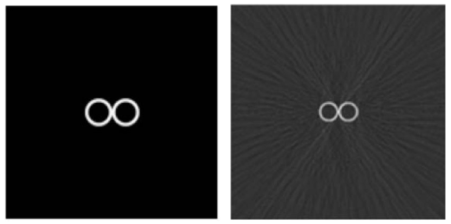
OFF-AXIS ERRORS

One key problem with this method is that objects far from the rotation axis change their distance from the light source as they spin. This causes their apparent size to grow and shrink periodically.

Because of this, even if volumes overlap in the images, we can't be completely sure if the object is really there, or if the overlap is just because it was closer to the light at some times.



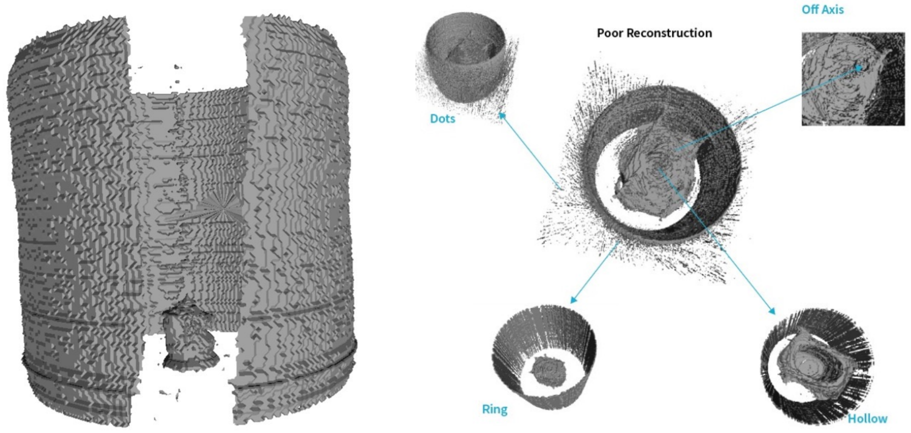
[3]



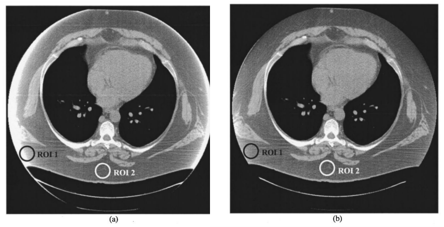
[3]

As shown in a different study, a detector rotation of π/4π/4 around the Y-axis introduces off-axis errors, resulting in a smaller and blurred projection due to geometric distortion.

ARTEFACT GALLERY



WHY DO WE GET WHITE STREAKS?



[4]

- After applying a filter, bright white streaks often appear at the edges of the images.
- This happens because many filters add padding (extra zero values) at the edges.
- As more filtered images are captured and back-projected, the padded values stack up.
- The result is a visible cylindrical "wall" around the reconstructed object — an artifact caused by the filter.
- A sharp step in the signal becomes a spike in the reconstruction because filtering in CT acts like taking a derivative. Specifically, multiplying the Fourier transform by **|ω|** (ramp) turns a step into:

$$h(u) = \mathcal{F}^{-1}\{\operatorname{sgn}\omega\} = -\frac{1}{\pi u}$$

- This creates strong edge artifacts known as "truncation walls."
  Filters like Shepp–Logan (0.6), Hamming (0.5), and Parzen (0.3) reduce the spike by softening the ramp, balancing sharpness with artifact suppression.
  A cosine taper fade can also be applied at the edges to further reduce these artifacts smoothly.