Can You Reconstruct Objects Based on Their Shadows? If Not Why Not?

Casey Sinclair

March 2026

1 Introduction

Take a look at the 3 puzzles shown below. Are you able to determine the 3D objects by their shadows for each puzzle with 100% certainty?

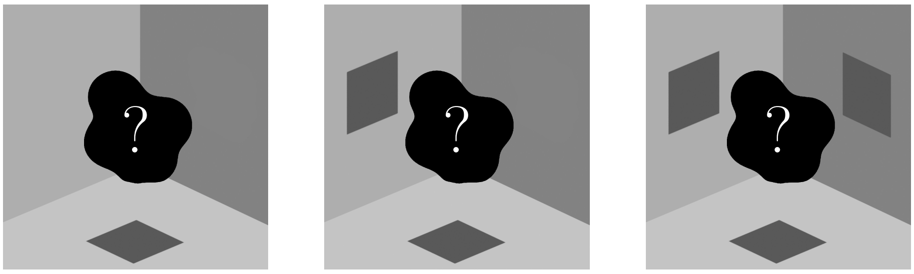
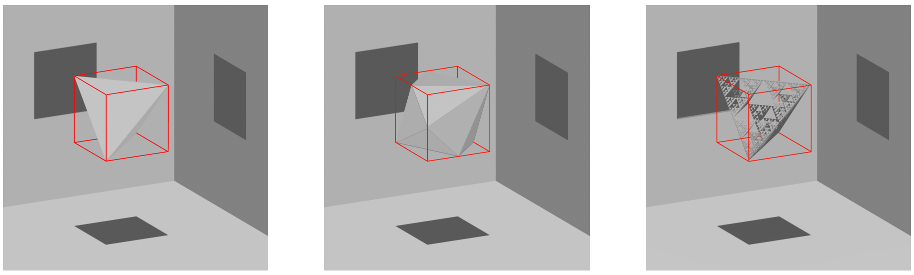


Figure 1: 3D Shadow Puzzle

It's clear that we can't for the first 2 puzzles. A 3D object has three degrees of spatial freedom, so one or two shadows leave at least one dimension unaccounted for. The puzzle becomes more interesting when there are 3 shadows. Because surely the unknown object must be a **cube**, right? Well... That's not the only thing it can be. It can also be a Tetrahedron or Hexagonal Bipyramid or even Sierpinski Tetrahedron.



Tetrahedron | Hexagonal Bipyramid | Sierpinski Tetrahedron

Figure 2: Solution of The 3D Shadow Puzzle

This begs the question: **What if we had all the shadows?** Would we be able to construct the object with 100% certainty? In more mathematical words: **Is an object uniquely defined by its shadows?**

2 But What is a Shadow?

2.1 Definitions

When tackling complicated problems in mathematics, it is often fruitful to simplify. Let's step down to 2 dimensions.

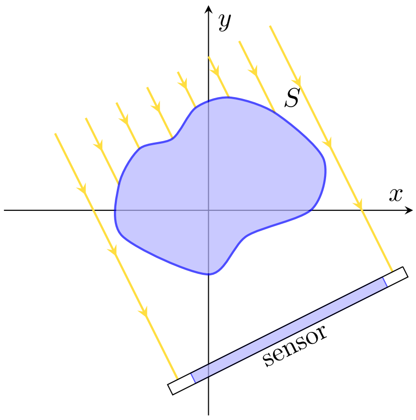


Figure 3: 2D Shadow Diagram

In 3 dimensions shadows are 2 dimensional, because they are projections of a 3 dimensional object onto a plane. So shadows in 2 dimensions should be 1 dimensional, because it would be a projection of a 2 dimensional object onto the line. To capture shadows from all angles you could imagine a shadow sensor and a light source rotating around the origin capturing the shadow data continuously. We can model the shape being measured as a set $S \subset \mathbb{R}^2$. To project any point $(x, y) \in S$ onto the sensor, we take its dot product with a unit vector T pointing in the direction of the sensor. The key insight is that the dot product with a unit vector tells you how far a point "reaches" in that direction, which is exactly what a shadow is. The natural choice for T is $(\cos(\phi), \sin(\phi))$, where $\phi \in [0, \pi)$. The set of all points in the shadow at a certain angle is called the shadow projection set.

Definition 1 (Shadow Projection Set)**.** Let $S \subset \mathbb{R}^2$ be a shape and let $\phi \in [0, \pi)$. The *shadow projection set* $\mathcal{P}_S(\phi) \subset \mathbb{R}$ is defined as:

$$\mathcal{P}_S(\phi) = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix} \middle| (x, y) \in S \right\} = \left\{ x\cos(\phi) + y\sin(\phi) \;\middle|\; (x, y) \in S \right\}$$

Although $\mathcal{P}_S(\phi)$ seems like an overly formal way to describe something simple, please bear with me, the utility of this definition will become clear shortly.

2.2 "Sine"s of Light

Before we go any further, we need a way to visualize the shadow projection set across all angles. Let's do this by graphing $\mathcal{P}_S(\phi)$ on the y axis and ϕ on the x axis, where white is the points in the set being graphed and black is otherwise. To build intuition, consider 2 examples:

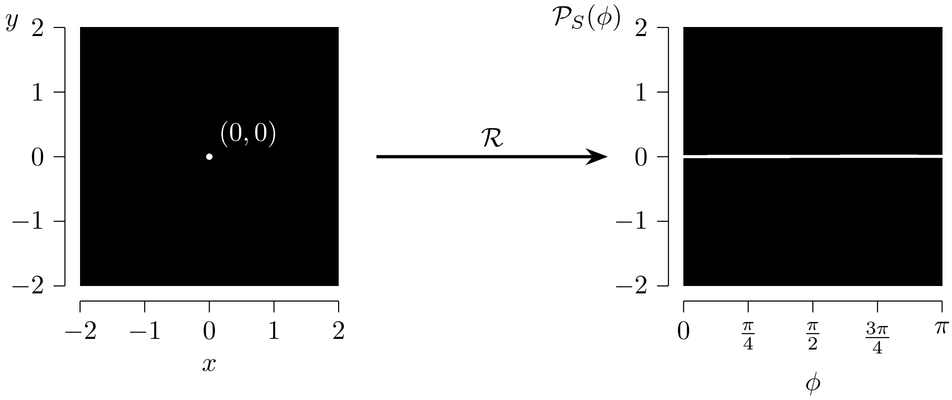


Figure 4: Sinogram of a Point at $(0, 0)$

A point at the origin has shadow $\mathcal{P}_{\{(0,0)\}}(\phi) = \{0\}$, as it always remains on the center of the sensor.

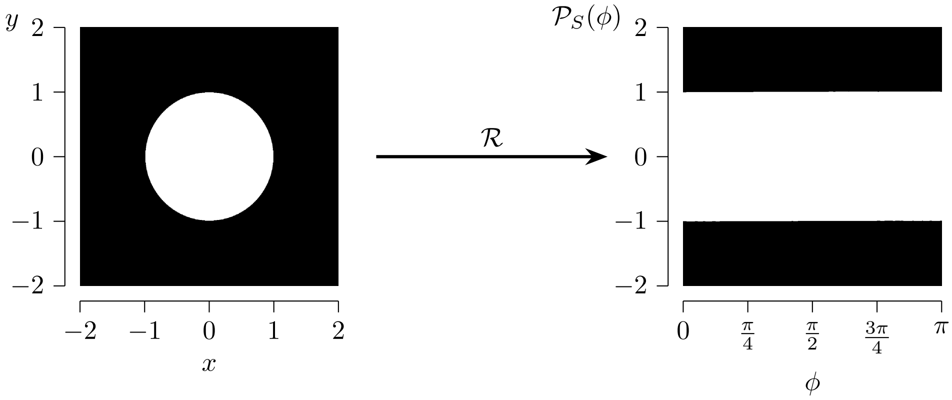


Figure 5: Sinogram of a Unit Disk

Similarly the shadow of a unit disk centered at the origin has shadow $\mathcal{P}_{\{(x,y)|x^2+y^2\leq 1\}}(\phi) = [-1, 1]$, as the unit disk is rotationally symmetric and therefore always falls on the same interval on the sensor. Now shift the point away from the origin to some (x, y). It would have shadow $\mathcal{P}_{\{(x,y)\}}(\phi) = \{x\cos(\phi) + y\sin(\phi)\}$:

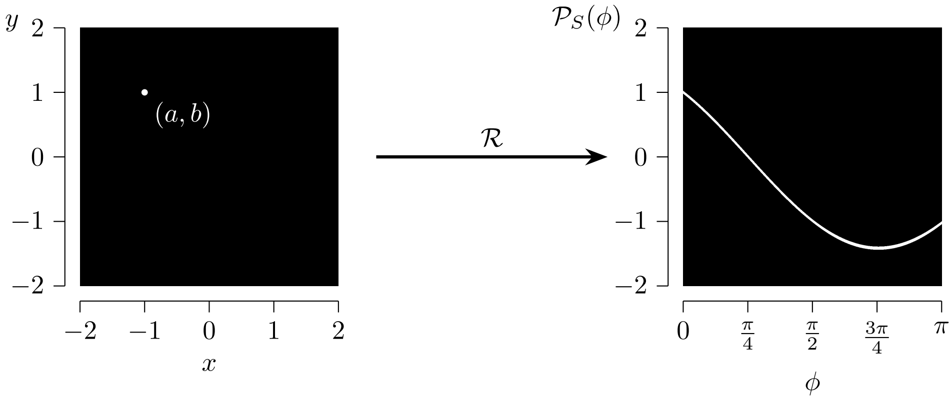


Figure 6: Sinogram of a Point at (a, b)

The shadow oscillates as ϕ varies, because the further an object is from the origin, the more its shadow meanders on the sensor. This kind of visualization is called a **sinogram**, and it will become our main tool in visualizing projections.

3 From Shadows to Shapes

Let's figure out how to reverse the process or in other words to back-project it. Luckily, we can think algorithmically about the shadow projection set, like so:

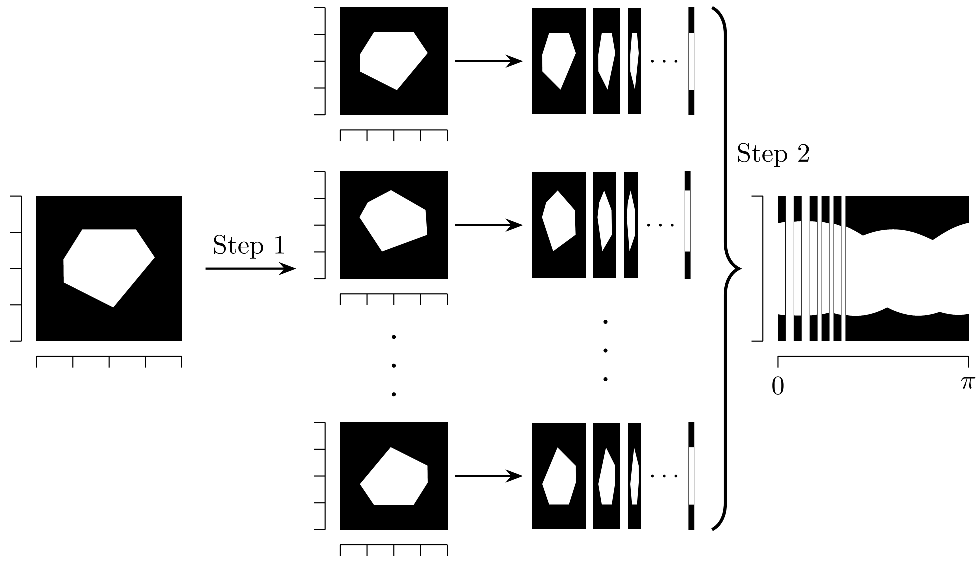


Figure 7: Shadow Projection Algorithm

Finding a back-projection formula would only entail going backwards on our algorithm. First we chop the sinogram into strips of individual angles. At each angle our algorithm is the squishing the object down to a lower dimension relative to the sensor. So let's do the exact opposite; Let's smear the shadow data across the $\mathbb{R}^2$ plane.

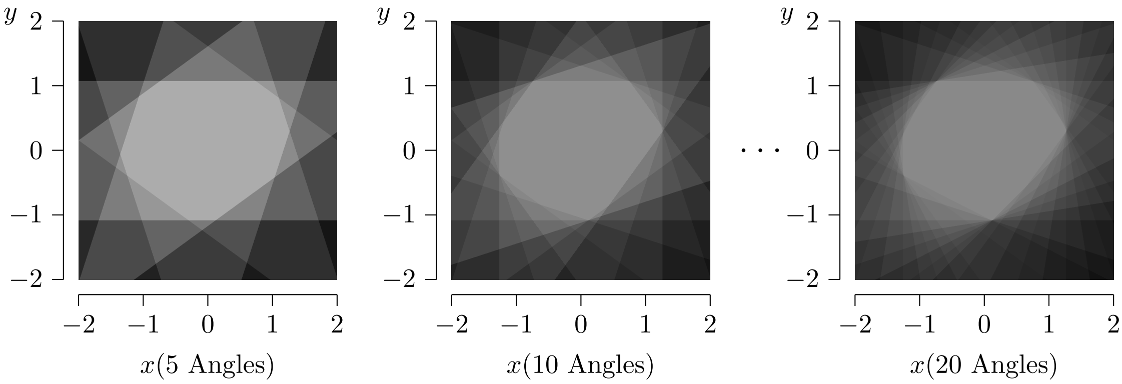


Figure 8: The Smear Set with More and More Angles

Even from here we can see our image is already half way to reconstruction. Mathematically the act of smearing a shadow can be defined as taking a certain point $p \in \mathcal{P}_S(\phi)$ for a fixed ϕ and creating a line on p with angle $\frac{\pi}{2} + \phi$, because the line is perpendicular to the sensor.

In order to do this more formally: Let us first define a line in the $\mathbb{R}^2$ plane.

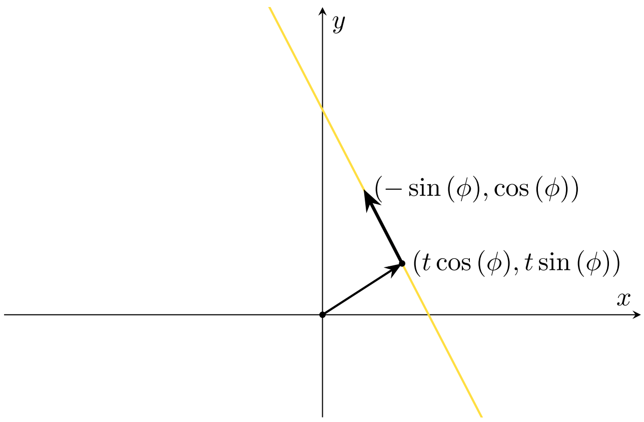


Figure 9: Graph of a Parametric Line

Definition 2 (Parametric Definition of a Line)**.** For $t, \theta \in \mathbb{R}$, the line $\ell_{t,\theta} : \mathbb{R} \to \mathbb{R}^2$ is the unique line passing through the point $(t\cos\theta, t\sin\theta)$ and perpendicular to the radial vector $(\cos(\theta), \sin(\theta))$, parametrized by:

$$\begin{aligned} \ell_{t,\theta}(s) &= \langle t\cos(\theta),\ t\sin(\theta)\rangle + s\,\langle -\sin(\theta),\ \cos(\theta)\rangle, \quad s \in \mathbb{R} \\ &= \langle\, t\cos(\theta) - s\sin(\theta),\ \ t\sin(\theta) + s\cos(\theta)\,\rangle \end{aligned}$$

In our context we can imagine first embedding $\mathcal{P}_S(\phi)$ on the x-axis in the $\mathbb{R}^2$ plane. Then we'll rotate all points $p \in \mathcal{P}_S(\phi)$ by ϕ mapping $(p, 0) \longrightarrow (p\cos(\phi), p\sin(\phi))$. This rotated point will be the point that our line passes through.

Definition 3 (Smear Set)**.** Let $S \subseteq \mathbb{R}^2$, and let $\phi \in \mathbb{R}$. The *smear set* $\mathcal{C}_{\mathcal{P}_S(\phi)}$ is the subset of $\mathbb{R}^2$ obtained by replacing each projected point $p \in \mathcal{P}_S(\phi)$ with the line $\ell_{p,\phi}$ (as in Definition 2):

$$\mathcal{C}_{\mathcal{P}_S(\phi)} = \bigcup_{p \in \mathcal{P}_S(\phi)} \ell_{p,\phi} = \left\{ \langle p\cos\phi - s\sin\phi,\ p\sin\phi + s\cos\phi \rangle \ \middle|\ p \in \mathcal{P}_S(\phi),\ s \in \mathbb{R} \right\}$$

Finally we'll reverse the final step by: Let's take all the angles and put them together by intersecting them, so that a point belongs to our reconstructed object if it lies in every smear set simultaneously:

Definition 4 (Shadow Back-Projection)**.** Let $S \subseteq \mathbb{R}^2$. The *shadow back-projection* of S is:

$$\mathcal{B}(\mathcal{P}_S) = \bigcap_{\phi \in [0,\pi)} \mathcal{C}_{\mathcal{P}_S(\phi)}$$

And using definition 4 this is our final back-projection:

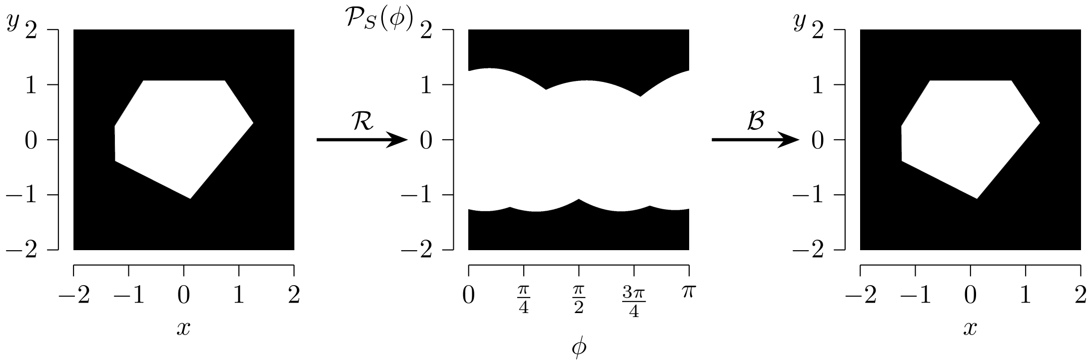


Figure 10: Final Reconstruction of a Shape S

As we can see it's dead on accurate and we've successfully recreated the original image... right?

4 The Problem Lurking in the Shadows

4.1 The Problem

Well... I lied. Some of the astute among you might have noticed that the derivation of the back-projection was a bit hand-wavy. And you'd be right to think that! To illustrate the problem we have, consider the star:

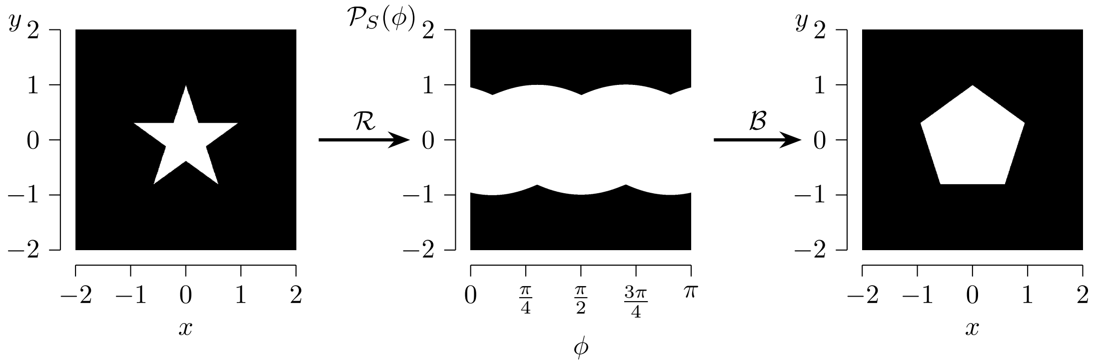


Figure 11: Failed Reconstruction of a Star

As you can see the back-projection of the star turns into a pentagon. So what went wrong? Consider 3 points A, B, and C on the concave dips of the star.

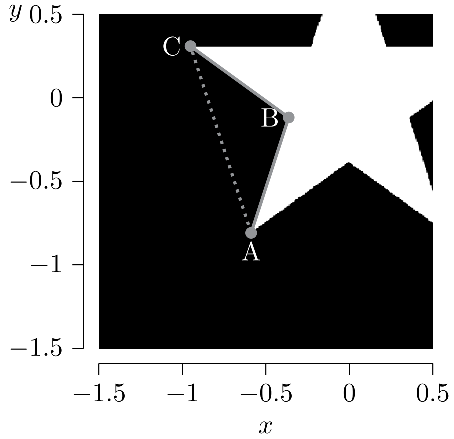


Figure 12: Graph of a Zoomed Star

The line segment AC lies outside the star. But our shadow projection can't distinguish between the shadow cast by ABC and AC. The projection can't see the dip, in other words it loses **depth** information. We didn't encounter this with the previous example because all the shapes there were convex, which means there won't ever be a line between 2 points in the same shape that go out of that shape.

So in fact it is **NOT** possible generally to construct an object based on its shadows. However if the shape is convex it will always be possible:

Theorem 1 (Weak Shadow Reconstruction)**.** Let $S \subseteq \mathbb{R}^2$ be a non-empty, compact, convex set. Then the shadow back-projection recovers S exactly:

$$\mathcal{B}(\mathcal{P}_S) = S$$

4.2 What the Shadow Projection actually does

To keep it short: I don't know (Kind of)! For a connected shape S it can be proven that it creates the smallest convex shape containing that shape $\text{conv}(S)$. However for a non-connected shape $S = S_1 \cup S_2 \cup \dots S_n$, where S_n are separate connected shapes, I conjecture that it creates some kind of shape S', where $\bigcup \text{conv}(S_n) \subseteq S' \subseteq \text{conv}(S)$.

5 Let There Be Depth!

So is that it? Should we be satisfied with the fact that Shadow Reconstruction is not always possible? I say no! If what we are missing is depth then let us find a way to let shadows communicate depth. And for that we turn to physics.

5.1 The Physics

Ever notice how light still bleeds through a piece of paper? Stack more sheets and less light gets through. For certain "semi-transparent" materials, the light intensity passing through an object carries information about its thickness. This gives shadows a way to encode **depth**!

Figure 13: How Light Bleeds Through Semi-Transparent Paper [1]

This relationship is described by Beer's Law:

$$I_1 = I_0 e^{-\mu \cdot \Delta x} \tag{1}$$

Where I_1 is the outgoing light intensity, $I_0$is the incoming light intensity,Δx is the thickness, and μ is the "attenuation constant", which governs how transparent the object is. An attenuation of 0 would be 100% transparent and an attenuation of ∞ would be 0% transparent or completely opaque. As a trick, we can treat μ as a function $\mu(x, y)$ defined on the whole of $\mathbb{R}^2$, where points outside the object simply have $\mu = 0$.

Let's take a look at a random shape S:

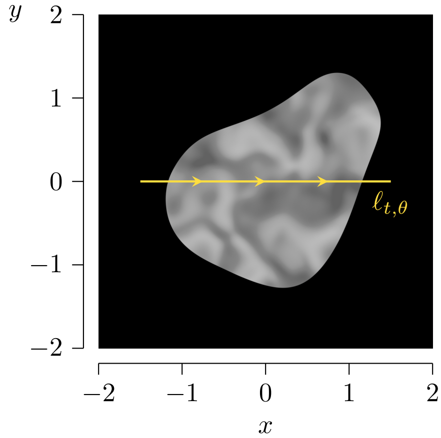


Figure 14: A Shape with Random Attenuation

This figure shows the value of the attenuation as a scale of black and white values on the graph where black is 0 attenuation and white is max (μ). Let's focus on 1 light ray $\ell_{t,\theta}$. By rearranging Beer's law:

$$-\ln\left(\frac{I_1}{I_0}\right) = \mu(x,y)\cdot\Delta s \tag{2}$$

The left hand side is measurable directly by the sensor. However $\mu(x,y)$ isn't a nice constant function. So we must approximate it as constant μ's over segments of Δs:

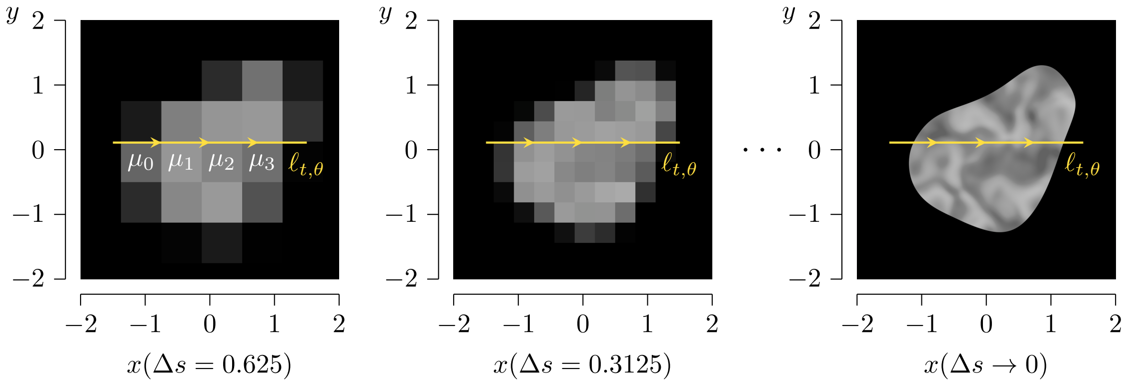


Figure 15: Approximation of The Attenuation of a Shape

As this approximation gets finer or mathematically as $\Delta s \to 0$, we can express beer's law as such:

$$-\ln\left(\frac{I_1}{I_0}\right) = \int_{\ell_{t,\theta}} \mu(x,y)\ ds \tag{3}$$

And thus we have created a way to project objects **without losing** its attenuation data/ **depth information**!

5.2 The Radon Transform

In order to make this definition more useful mathematically we can take some of the physics out of it and define a transform that does this kind of projection with any function called the Radon Transform.

Definition 5 (The Radon Transform)**.** Let $f : \mathbb{R}^2 \to \mathbb{R}$ be an integrable function. The *Radon transform* $\mathcal{R}_{f(x,y)} : \mathbb{R} \times [0, \pi) \to \mathbb{R}$ is defined by:

$$\mathcal{R}_{f(x,y)}(t, \theta) \;=\; \int_{\ell_{t,\theta}} f\, ds \;=\; \int_{-\infty}^{\infty} f\big(t\cos\theta - s\sin\theta,\;\; t\sin\theta + s\cos\theta\big)\, ds$$

where $\ell_{t,\theta}$ is the line from Definition 2 and s is its arc-length parameter.

Returning to our sinogram, let's revisit the pentagon and the star. Recall that their shadow projection sets were identical, and that's why the back-projection couldn't tell them apart. Now we could visualize $\mathcal{R}_{f(x,y)}(t, \theta)$ using sinograms, where white represents $\max\left(\mathcal{R}_{f(x,y)}(t, \theta)\right)$, black represents 0, and gray are all the values in between:

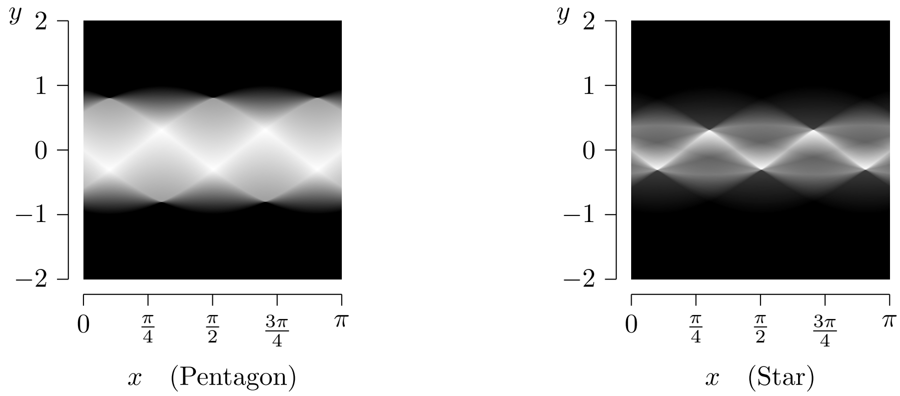


Figure 16: Comparison of a Sinogram of a Pentagon and a Star

Here we see a stark difference. The pentagon's sinogram has noticeably more white, which intuitively is because it's "thicker" or "chunkier" than the star, so its light rays pass through more material at every angle.

5.3 Creating an Inverse...

This task might seem daunting. However we've built a back-projection before! So let's use the same outline here. We'll first smear our projection at every angle on $\mathbb{R}^2$. But this time we don't take intersections. Instead, for each point (x, y) we average the Radon values of every line passing through it across all angles.

This is the back-projection!

Definition 6 (Back-Projection Operator)**.** Let $f : \mathbb{R}^2 \to \mathbb{R}$ be integrable. The *back-projection* $\mathcal{B}(\mathcal{R}_{f(x,y)}(t, \theta)) : \mathbb{R}^2 \to \mathbb{R}$ is defined for each $(x, y) \in \mathbb{R}^2$ by:

$$\mathcal{B}\big(\mathcal{R}_{f(x,y)}(t, \theta)\big)(x, y) \;=\; \frac{1}{\pi} \int_0^{\pi} \mathcal{R}_{f(x,y)}\big(x\cos\theta + y\sin\theta,\; \theta\big)\, d\theta$$

Let's return to the star:

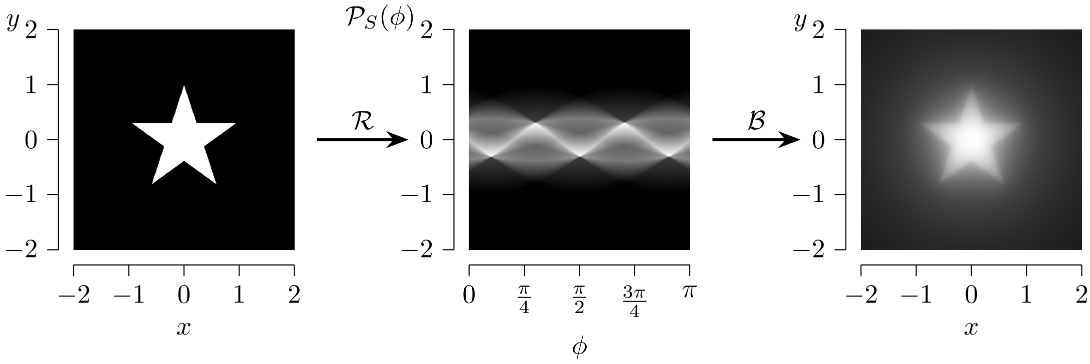


Figure 17: Unfiltered Back-Projection of a Star

And... The resulting images came out blurry. Why...?

5.4 The Light at The End of The Tunnel

Notice the boundary between the inside and outside of our shape. In terms of $\mu(x, y)$ this is a sudden jump in value or a sharp discontinuity. When we took an average across all angles, instead of reinforcing the sharp edge, we smeared it outward in all directions.

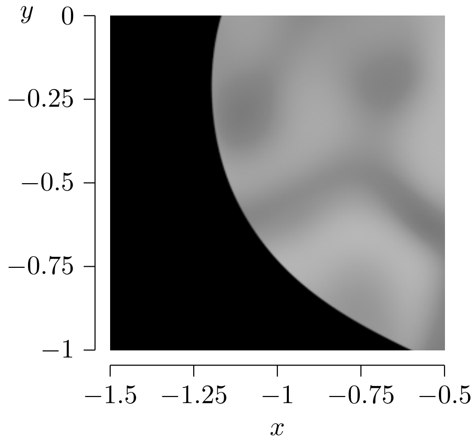


Figure 18: Graph of a Zoomed Shape with Random Attenuation

The key insight here lies in the use of Fourier transforms. Although I won't formally derive or go too deep into detail about the transform here, as it deserves a whole essay on its own. However I want you to get a gut understanding of why the use of this tool is necessary. The 1D Fourier transform acts on 1D functions. It and its inverse are defined to be:

$$\mathcal{F}[f(x)](\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-2\pi i \omega t}\, dt$$

$$\mathcal{F}^{-1}[F(\omega)](x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(t) e^{itx}\, dt$$

While the 2D Fourier transform acts on 2D functions, where it and its inverse are defined to be:

$$\mathcal{F}_2[f(x,y)](X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i(xX+yY)}\, dx\, dy$$
$$\mathcal{F}_2^{-1}[F(X,Y)](x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(X,Y) e^{i(xX+yY)}\, dX\, dY$$

The Fourier transform essentially acts like a portal into the frequency spectrum of function. Consider for example the function $f(x) = \sin(3x)$, we know that it has a frequency of 3. That is reflected in a spike in the Fourier transform like such:

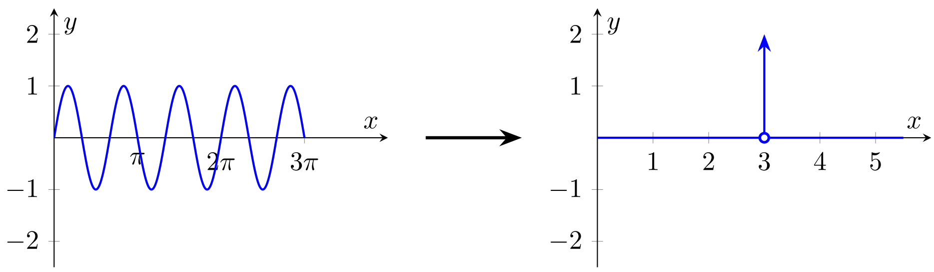


Figure 19: Fourier Transform of a Sine Wave

Consider more complicated functions such as the step function you could imagine that we perfectly approximate them using sines like this $\frac{4}{\pi}\sum_{n=0}^{N}\frac{\sin((2n+1)x)}{2n+1}$. The question becomes which frequencies are most influential in creating that sudden jump?

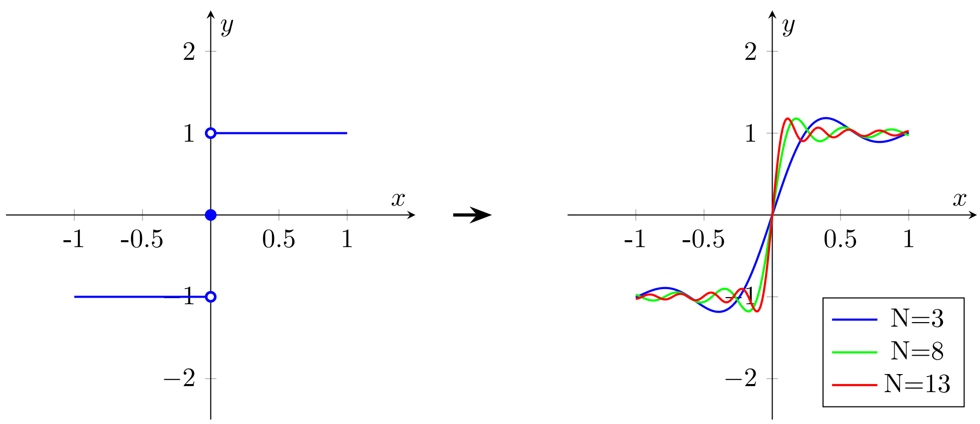


Figure 20: Fourier Series of The Step Function

Just by the example above we can see that it's the higher frequencies. So if we can increase the magnitude of those frequencies we might get a sharper image. Mathematically this is done by multiplying $\mathcal{F}_1[f]$ a Ram-Lak Filter $|\,\omega\,|$:

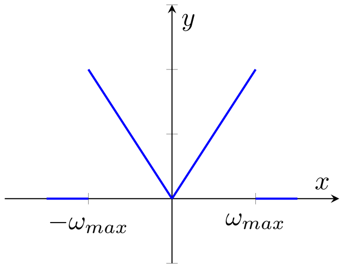


Figure 21: The Ram-Lak Filter

So to build up what we call the filtered back-projection, we must first chop up our sinogram at each angle and take the 1D Fourier transform of each of those strips. Then we'll amplify the sharp edges using Ram-Lak's filter. Next, we'll stitch it back together into a 2D image and using the inverse 2D Fourier Transform we'll get out of the function's frequency spectrum. Finally we'll back project that image once more and we'll get our final filtered image or $\mathcal{B}(\mathcal{F}_2^{-1}(|\omega|\mathcal{F}(\mathcal{R}_S)))$:

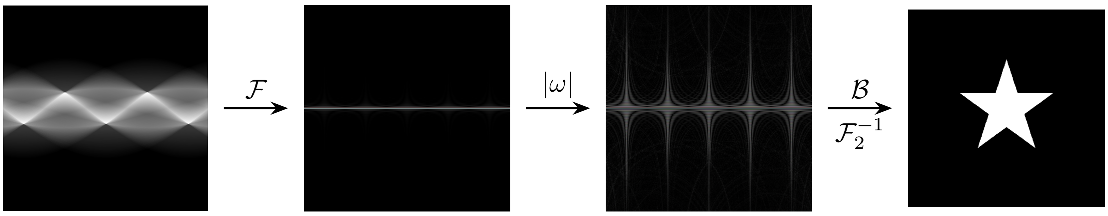


Figure 22: Filtered Back-Projection of a Star

6 X-rays and CT-scans??? (Conclusion)

We started with a simple puzzle: Can you reconstruct an object based on its shadows? As it turns out the answer depends entirely on what kind of shadows you have! Opaque shadows lose depth and can only reconstruct convex shapes. While "semi-transparent" shadows encode depth and have enough information to be accurately reconstructed (with a few tricks).

Moreover this kind of mathematics actually has real world applications! Most biological tissue has a measurable attenuation coefficient at X-ray frequencies, making the human body effectively semi-transparent to X-rays. The Radon Transform lays the foundation for a field in mathematics called **tomography**.

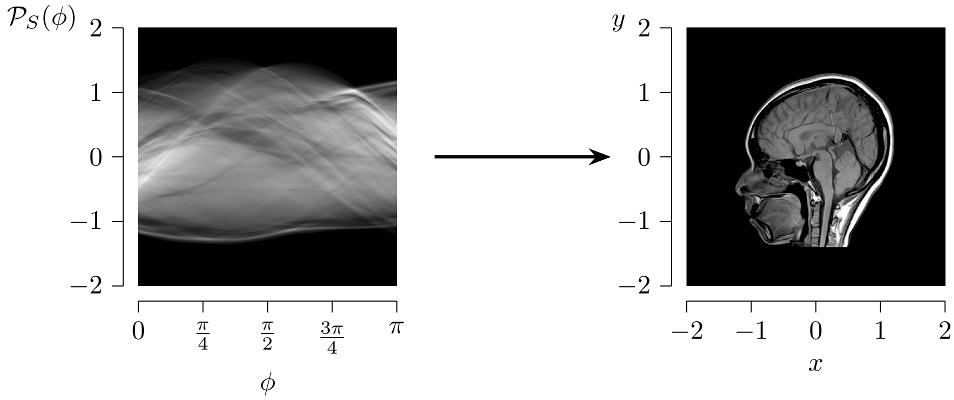
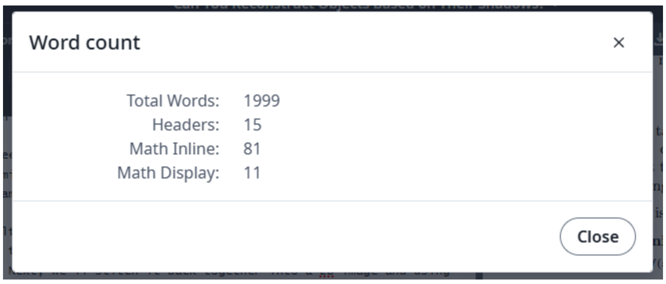


Figure 23: Sinogram of a Head [2]

I'll leave you with this: We began our journey playing with shadows on a wall and ended on the mathematics that's used every day in a hospital. You'll never know where a curious question might lead you. So be curious, play with mathematics, and most importantly have fun!

Final Note : And that ladies and gentlemen, is exactly 1999 words! Kind of poetic don't you think?



References

Sources

Feeman, T. (2015). *The Mathematics of Medical Imaging.* Springer

Images

[1] https://pixabay.com/photos/flowers-lamp-japanese-paper-mino-2710381/

[2] https://radiopaedia.org/cases/normal-mri-brain-3