slide1:

Welcome back. We can see today’s topic is CT Reconstruction.

In this session, we’ll build on that foundation and map raw X-ray measurements into cross-sectional images. By the end, you’ll have a clear roadmap of the reconstruction pipeline and where classic methods, such as filtered back-projection and modern iterative ideas, fit into an online, real-world CT workflow.

slide2:

Let’s take a look at the roadmap: we’re at the CT Reconstruction milestone. Keep one core idea in mind from the physics background—the roles of the X-ray source, the detector, and how a measurement is formed. Each reading can be interpreted as a line integral of the attenuation coefficient along a straight ray.

With that model in place, we now focus on algorithms that invert those line integrals to recover cross-sectional images. If you’d like more depth on the geometry or notation, the companion reading for this module provides it

slide3:

Let’s take a look at a few habits that make this module easier.

First, recommended reading: Ge’s book and the widely used “Green Book.” Together, they give both intuition and practical formulas. Next, focus on the logic flow in the video lessons and slides—how each idea leads to the next. Aim to capture the key concepts rather than memorize isolated facts.

Build skills by working on problems from homework and textbook questions; problem-solving is where concepts become durable.Adopt a simple routine: preview → learn → review → repeat. Preview the slide headlines, watch the lesson, then review by summarizing in your own words and redoing one or two problems.

For foundations, give extra attention to Fourier analysis and the sampling theorem—these are the grammar of modern imaging. They will pay off when we study CT filtered back-projection and, later, MRI pulse sequences and other modalities. If questions come up, post them in the course Q&A so others can benefit too.

slide4:

Here we observe the plan for CT reconstruction. We will connect data space—the set of X-ray line integrals—to image space—the distribution of attenuation, usually written as mu of x, y. Think of this as a duality of information: each view in the sinogram encodes a different slice of information about the same object.

We’ll compare two families of methods:

Algebraic approach.We model the scanner with linear equations: A times x equals b.This viewpoint helps us reason about solution uniqueness—for example, when A has full column rank—and about data independence and sufficiency, meaning we need enough non-redundant views and detector samples to determine x stably. Regularization can be added to handle noise or limited angles.

Analytic approach.

This relies on Fourier analysis. By the Fourier Slice Theorem, the one-dimensional Fourier transform of a projection at angle theta equals a radial slice through the two-dimensional Fourier transform of the object at the same angle.

From this principle, we derive filtered back-projection (FBP): first apply a frequency-domain ramp filter—magnitude of omega—to each projection, then smear, or back-project, the filtered data across the image grid. FBP is fast and remains the standard for many clinical workflows.

slide5:

Let’s take a moment to underline the information. Like in physics, we have a particle property and a wave property. For image reconstruction, we can make an analogy. We can view the underlying image in two ways.

First, we view the image as a collection of pixels if the image is two-dimensional, or as pixels for three-dimensional images—a collection of particles—here meaning picture elements; the pixels, the pixels (for 3-D, these are often called voxels).

The other way is complementary: we view the image as a superposition of waves. For a two-dimensional image, for example, think of the image as a summation of many waves propagating along different orientations. For a given orientation, the wave can be at different frequencies, and you have amplitude, frequency, and phase. If all these parameters are set correctly and you add all these waves together, you obtain the image. This is the basic, high-level idea—the particle and wave perspectives of the image and of image reconstruction.

slide6:

Let me explain a little further. Take an arbitrary picture—say, a portrait of our former president. You can decompose the picture into many, many small elements. For each element, it’s very simple: just a homogeneous pixel, a small square. Add these together.

The trick is really the amplitude: you need to make sure all these amplitudes are modulated nicely. Then, when you put them together, you have the perception of a picture or a natural scene, whatever. This is one way to represent a picture.

slide7:

Another way, as mentioned, is Fourier analysis. In a two-dimensional picture, you perform a Fourier analysis and obtain a Fourier spectrum.

Pick any point: the frequency is proportional to the distance from the origin to that point in the Fourier domain. In this example, you see the frequency, and you also have an amplitude proportional to the value at this particular point. That gives a particular wave propagating along that direction.

Fourier analysis uses all kinds of wave components: the DC component, horizontal waves, vertical waves, low frequency, high frequency, and waves that propagate in arbitrary directions—all these kinds of waves. The trick is the setting of the parameters. Once you find the coefficients, you add the Fourier components together, and then you can recover the image—in this case, Albert Einstein.

So there are two ways, and based on these two perspectives, we can have two approaches.

slide8:

We introduced two families of methods—algebraic and analytic. Let’s begin with the algebraic approach and set up the measurement model it relies on.

slide9:

The measurement can be written in line-integral form or in ray-sum form. Let me explain.

First, we have an incoming intensity, N i, and an output intensity, N o. The output is related to the input through linear attenuation. For a single element along the ray, the attenuation is mu times delta x, where delta x is the pixel size.

With multiple material components along the path, the incoming intensity is attenuated by the first pixel; the output of the first pixel becomes the input to the second pixel, and so on. Adding these effects together gives the standard relationship. This is a quick review of the earlier module.

The attenuated output equals the input multiplied by an exponential factor whose exponent is the negative sum of the little mu k values, each weighted by delta x. This is our imaging model or data model.

What is known and what is unknown?Known: N i, N o, and the step size delta x that we choose.Unknown: the set mu k, for k equals 1, 2, 3, …, n.

Normalize by taking a ratio and then a natural logarithm:

“Natural log of N i divided by N o equals the sum over k of mu k times delta x.”

This is a linear equation in the unknowns mu k, with delta x acting as a weight. Because N i and N o are measured, the right-hand side is known after the log. For each X-ray path, we obtain one linear equation. With many paths, we obtain a system of linear equations, often written compactly as A times x equals b. In this algebraic view, the image is treated as pixels, and each measurement provides a linear constraint on those pixel values.

If we take the limit as delta x becomes very small, the sum becomes an integral:

“The integral along the ray of mu of x, d x, equals the natural log of N i over N o.”

So, in the discrete domain we speak of a ray-sum—a summation—and in the continuous domain we speak of a line integral—an integral along the X-ray path. Either way, X-ray measurements provide ray-sums or line integrals of the attenuation field. With this model in mind, we can now discuss why the resulting linear system can have a unique solution, at least heuristically.

slide10:

This is what I call the onion-peeling idea.

Think of a picture and decompose it into a collection of pixels. The pixel does not have to be rectangular; here, I use triangular pixels. I perform the decomposition into many, many triangular pixels and then resolve the unknowns layer by layer. Start with the outermost unknowns, represented as a right triangle. This is a heuristic idea, so follow the logic.

In the limiting case, imagine the object support where an X-ray touches one molecule, an extremely small element. Take this as an example: you have a small material element. You know the incoming intensity, you know the attenuated intensity, and because the data are partitioned, you know the total path length. From these three knowns, you can resolve the only unknown, the attenuation coefficient mu—call it mu-zero or mu-one—for that pixel. By assumption, this pixel is homogeneous. With this simple argument, you now know mu for this right triangle. By the same argument, all the right-layer pixels can be directly measured with a prior flow of X-rays.

Once that is done, move to the next layer, colored green. Use a green X-ray. You know the incoming intensity and the attenuated intensity. You already know mu for the right element, and you also know mu for the current neighbor. Because you know these mu values, you can compute how the incoming intensity is attenuated and thus determine the incoming flux into the green pixel. You also know the attenuated intensity leaving this location. That attenuated intensity is the quantity just out of the green pixel but further attenuated by the right pixel. Using Beer’s law in reverse, you back out the intensity before that extra attenuation. Now you know the value here, you know the value there, and you know the total length, so you can resolve mu for this particular green triangular pixel. Likewise, each of the green elements is resolved.

Next, move to the light-blue ray. At this point, the green and the right layers are known; only this one remaining element is unknown. The same heuristic applies.

This argument ties the relations without complicated mathematics. Heuristically, you can peel the image layer by layer and resolve the underlying image algebraically. When you solve the linear system equation pixel-wise, you are essentially doing exactly this: layer by layer. In parallel-beam geometry, you can send all possible rays and reorder them so you effectively view the object from the outermost layer inward, resolving all unknowns layer by layer. This is the key argument.

slide11:

So, mathematically, we can state a data sufficiency condition for two-dimensional image reconstruction. You have a cross-section, and you arbitrarily draw lines—these are the X-ray paths. For any such line, we say you can find at least one source position.

What does that mean? It means that along this line, an X-ray source sends rays in this direction, so the line integral, or ray-sum, is measured. It doesn’t mean you have every piece of information you might want. It means: if a line intersects the cross-section, then we have data on that line. That is the maximum amount of data you could have, and, under this simple understanding, that amount is sufficient.

This is the way I understand the subject and how I try to explain it to students. I like to give you a picture and some visual, geometrical ideas. If you feel confused, you can watch my lesson. Right now, I think there are about half a million viewers, a lot of likes, so I really hope your watch gives me a like—that's good.

Some other heuristics are very important but not easy. For example, when you compute the coefficient, think of it as an inner product—a high-dimensional vector projected onto a basis function. That is a very important heuristic, but I doubt all of you fully understand it. If you don't, please review. I hope, by the end of this month, to upload a newer version that will be much better than the rough draft, so please read it. Even if I wouldn't test it again, you need to understand Fourier analysis so that you understand CT and MRI much better. Anyway, this is the idea of the data sufficiency condition.

slide12:

Okay. With X-rays, we can measure parallel-beam projections as shown here. Along each X-ray, each datum gives you one linear system equation. If you have a parallel beam at projection angle theta, and you have, say, one hundred rays or five hundred rays, then you have one hundred or five hundred linear system equations.

One view is not enough, because you cannot resolve superimposed structures—you see two things on top of each other, and you do not know which one is on top and which one is beneath. So you need to keep changing theta.

The original function f of x, y is converted to a new two-dimensional function p of theta, t. Theta is the angle; t is the coordinate. Using X-ray measurements, you perform a physical or mathematical transform from f of x, y to p of theta, t.

For example, if you have a bright, small disk, it will trace a sinusoidal curve. That is why we call the data-domain representation a sinogram. We also call it a Radon transform—Radon was a mathematician many years ago. So this is a projection, one view; the sinogram puts all the views together.

slide13:

Computed tomography was previously called computer-aided tomography, C-A-T—so we usually put a cute cat here. C-T is nothing but the inverse process. Once you have sinogram data, p of theta, t, the question is: given the measurements, how do you invert the process?

What is the underlying image that explains this data? Once you have the data, you reconstruct the image. The inverse process goes from data to image. X-ray measurement goes from image to data. The tomographic algorithm goes the other way, from data to image, the underlying image.

And how do you do it? The picture I gave you shows you can use an impending method, so you have a heuristic feeling.

slide14:

Now, let me give you a numerical example.

In practice, the image can be five hundred twelve by five hundred twelve, but for teaching purposes, I'll use a two by two image. This is a simple case, but the essential idea is already there.

You have pixel values one, two, three, four—very simple—but you don't know these; this is just something I set up. What you are allowed to probe is with X-rays. I told you X-rays cannot pinpoint a single pixel. If you had a magic pen that could read out a point one pixel like a photograph, you would not need tomographic reconstruction—that would be too simple.

With X-ray measurements, we can get some information. Suppose you shoot X-rays this way. As I explained, an X-ray measurement gives you a ray-sum. From the earlier slide: if you do not send X-rays through these two pixels, you will not be able to say what mu one and mu two are. But you do know the sum of these two pixels: mu one plus mu two equals seven. From this, you still do not know mu one and mu two, but you know the sum is seven. Likewise, take a vertical ray: you get mu two plus mu four equals four.

With X-ray measurements, you can write a number of equations. Solve the equations and you get the unknowns—that's the idea. Normally, for an n by n image, you have n by n unknowns. Here, for two by two, you have four unknowns, so you need to shoot four rays and get four measurements: four linear equations, four unknowns. It looks perfect, right? Not that simple.

Look at this: if you shoot these two rays, the first two equations added together give a right-hand side of ten, and the last two equations added together are also ten. So if you subtract one equation from the sum, you get the remaining one. It's a little tricky, but the point is that these four equations are not totally independent. From three of them you can derive the last one.

The trick is that "number of equations equals number of unknowns" only works under the condition that each equation provides new information—they must be independent. If they are not independent, you do not have enough equations to solve the problem.

So you should shoot a ray along the diagonal direction. Then you may get "this one plus this one plus this one plus this one," and you will have enough independent equations to solve uniquely.

Anyway, remember: "number of equations equals number of unknowns" is under the assumption that all equations are independent. Otherwise, you do not have enough equations.

slide15:

Here, I'll still use two horizontal rays and two vertical rays to show how we solve linear system equations using iterative algorithms through trial and error.

Why explain an iterative algorithm? For a simple case like this, you can solve it directly using an analytic method. But when the number of unknowns is huge—like millions or billions of equations—the direct method will not work efficiently. You may not have enough computer memory, and other issues may arise. So you use an iterative algorithm to solve the situation. Also, the iterative algorithm I'm about to explain lets you impose prior knowledge, like non-negativity or smoothness. These topics are beyond the scope of this lecture.

Let me give you the basic idea—how you can do trial and error to solve a system of linear equations. As I told you, this is your underlying image. You have four measurements—two horizontal and two vertical—and you try to solve this system. The starting point: whenever you use an iterative algorithm, you need a starting point. Here, because I know nothing about the image, I choose a neutral guess and assume nothing in the field of view, so every pixel is zero. This is my starting point, or guess zero. It is a natural and unbiased starting point.

First, if this guess were correct, then the vertical integrals must be zero and zero. Based on the assumption, I get these two estimated values: zero, zero. We call this a predicted or synthetic projection. It's not that you must accept zero, zero just because I say so—you can challenge it. You can say, "If everything is zero, the vertical integrals must be zero, zero." But the physical measurement says six and four. It is not zero, zero. How do we explain the contradiction?

We compare the measurement with the prediction. We see errors six and four—here you measured six but predicted zero, so the error is six; there you measured four but predicted zero, so the error is four. This positive error indicates my initial guess underestimates the pixel values.

Along this ray, the real measurement is four, but I predicted zero. Clearly, there must be something in these two pixels. Their values should add to six along the first vertical ray. I do not know whether to give more to the first pixel or the second pixel, so to be fair, I evenly divide the error: six becomes three and three. I put the error back. After this redistribution, the two values add to six, so that vertical error is removed. Likewise, for the second vertical ray, four becomes two and two. After this, I am vertically consistent: along the first column, three plus three equals six, and along the second column, two plus two equals four, matching the measurements.

Next, let's double-check the horizontal integrals. Now the row sums are five and five. The measured horizontal sums are seven and three. Comparing again, the errors are plus two and minus two. A positive error means I still underestimate that row; a negative error means the true sum is less than my current estimate.

I redistribute the errors: the plus two is decomposed into one and one and added back to the two pixels in the first row; the minus two is decomposed into minus one and minus one and subtracted from the two pixels in the second row. In this case, we are lucky—after this second iteration, we obtain the correct result. Once we reach this state, the vertical and horizontal data are both perfectly explained. We are done—this illustrates the idea.

In real situations, it is never this simple. You need many iterations and many unknowns go back and forth, and a well-designed iterative algorithm will ensure that after many iterations the solution converges. Sometimes the iterative process gives an oscillating solution, and you need regularization. But again, for this undergraduate-level course, notice the basic idea.

slide16:

So, in summary, the algebraic approach goes in the following steps.

First, convert the data into line integrals to form a system of linear equations.This is the first step.

Second, solve the system of linear equations to reconstruct the underlying image, like the iterative process I showed you.

If needed—although I didn't explain the details—you regularize the reconstruction with prior knowledge. For example, you know the CT attenuation coefficient, mu, determines attenuated X-ray intensity and cannot be negative. So, during the iterative process, if the current solution gives a negative value, you know it cannot be negative; you force the negative to zero. That is how prior knowledge regularizes the image reconstruction.

Then you iteratively refine the intermediate image, or current guess, one cycle at a time until the outcome is satisfactory.

How do you know the outcome is satisfactory? One way is visual—the image makes sense given your prior knowledge. Another way is data consistency—based on your current image, you predict the projection data; if your prediction compares well with the measurement—very close—you say, okay, it is good enough. As far as data fitness is concerned, we are doing a good job.

So, that is the idea of the first algebraic approach. The first one is not very hard, but at least you are sure that we can do it.

slide17:

Next, the analytic approach involves the Fourier slice theorem—that's the key point—so we need Fourier analysis.

If you already understand Fourier analysis very well, you will have a good time here. If you are still confused about Fourier analysis, you may struggle a little, and I recommend a review. This is not an easy topic, but it is very cool. For the Fourier analysis, analytic approach, let me give you a heuristic explanation.

slide18:

From this point of view, you shoot X-rays, and you are not trying to get a result directly. This is a different perspective. You shoot parallel-beam X-rays, going this way and that way, and then you get line-integral measurements. What do such measurements do? They act like a probing wave.

In this case, think of the underlying image not as Albert Einstein or a cross-section of your chest. Think of the image as a summation of many waves. Because of Fourier analysis, you can always do this—you reconstruct a bunch of waves.

For example, consider waves propagating horizontally, like the green wave. For any horizontal wave, let me make some comments so you understand why Fourier analysis works nicely. For such a horizontal wave, for all X-ray projections along oblique directions—that is, any orientation that is not vertical—we can say one thing: if you do the vertical line integral, the wave has a positive cycle and a negative cycle that cancel out. The wave just keeps oscillating. So if the projection orientation is horizontal or makes an oblique angle—any degree except vertical—all these ray-sums give you zero.

Getting zero means you get no information about the underlying horizontal wave—except in one direction: the vertical direction. Along the vertical projection, the positive and negative cycles do not cancel, so only the vertical projection gives you critical information about a horizontal wave. Other directions cancel out.

From the vertical projection, you obtain information about waves propagating horizontally. And Fourier analysis says the image decomposes into many waves along different orientations. So, to get horizontal waves, you use vertical projections. For waves at one-degree orientation, you need projections at ninety degrees plus one degree—the orthogonal direction.

If you want to resolve all the waves, you need projection angles spanning zero to one-hundred-eighty degrees. This is the heuristic: you need orientations that probe each wave family. With the horizontal wave superimposed this way, you take the vertical projection; then, if you do Fourier analysis along the horizontal direction, you should recover all the wave information along that horizontal direction. Let me explain this a little better.

slide19:

In the so-called Fourier slice theorem, it basically says this: you have the parallel-beam projection. So p of theta—this angle is theta. You get all these projections, and each projection carries wave information along its direction, the X-ray direction.

The information carried by this projection profile is waves along the p-axis. This p-axis makes an angle theta. It’s the same idea as before: a vertical projection carries wave information along the horizontal direction. So a projection at angle theta carries wave information along the p-axis, making angle theta.

If you perform a one-dimensional Fourier analysis, you get the Fourier spectrum. This green profile is the Fourier spectrum of the projection profile. The Fourier spectrum lies along the rho-axis, making the same angle theta. A point on this rho-axis gives a wave whose frequency is proportional to the distance from that point to the system origin. It is a wave propagating along the theta direction, orthogonal to the X-ray beam direction. Along this rho-axis, you have many points; these points represent unique two-dimensional waves propagating along this direction, making an angle theta.

This is heuristic. You need a one-dimensional Fourier transform to recover a radial line profile in Fourier space. To reconstruct a two-dimensional image, f of x, y, you need all the Fourier information, so the angle theta needs to go from zero to one hundred eighty degrees. When the rho-axis orientation changes from theta equals zero, then one, two, three, up to one hundred eighty degrees, the whole Fourier space is swept by the rho-axis. That means with one projection, you only measure information along one line. But if you change theta from zero to one hundred eighty degrees, all the data points in the Fourier space have been measured—you have all the information. Then you can perform a two-dimensional inverse Fourier transform.

This is a geometrical perspective on how to reconstruct the image using Fourier analysis, or from a wave-analysis perspective. Just these few things—two slides. I hope you understand these two slides. Then we will take a minute's rest. I will show you my mathematics.

Step by step, you will understand this geometrical wave argument a little better. OK. I think the algebraic perspective may be easier for you. The wave analysis—if you feel confused, feel free to ask me. Think about it. This is a very elegant way to solve the problem. The Fourier slice theorem is an important theorem for tomographic reconstruction.

slide20:

I uploaded a chapter by Clark in a CT book. It’s a very good one. When I was a fourth-year graduate student, I did a summer job with my supervisor. That summer, I read through his book and implemented the cone-beam reconstruction algorithm. In my view, this is still the best engineering book to explain CT principles.

The green textbook explains back-projection, but not as clearly as Clark's book. To give a better, deeper explanation, we will use Clark's Chapter 3.

Now the coordinate system is x, y. Draw a line making an angle theta; call this line the t-axis. The projection is defined by the perpendicular distance from a point to the t-axis. The line through the origin at that distance is t. It can be expressed asx cosine theta plus y sine theta equals t.

The unit directional vector along the t-axis has components cosine theta, sine theta. They didn't draw it, but you can think of cosine theta and sine theta as the two components of the unit vector along the t-direction.

Take an arbitrary point x, y. Regarding the point as a vector from the origin. Project this vector onto the t-axis. The resulting distance is t. This is really an inner product: one vector with components x, y, and the unit vector with components cosine theta, sine theta. The inner product equals t. As long as t is the same, all points that satisfy this equation lie along the same straight line at a distance t.

You then do a line integral along that line at an angle theta and distance t. The projection p of theta, t, has two variables, but for a given theta, it is a one-dimensional function of t.

This line integral can also be written as a double integral. You integrate the underlying image over the whole plane, and you place a delta function along that line so that only the values on the line contribute. In other words, the double integral with the delta function is nothing but the line integral along that direction. This gives a convenient two-dimensional notation for the projection.

slide21:

Then we can write the standard Fourier transform for the underlying function f of x, y. Equation seven is the two-dimensional Fourier transform—straightforward. Let me write the one-dimensional Fourier transform—just the definition—equation eight.

I have the one-dimensional function, which is the projection profile p of theta, t. It has one variable. Then you perform the Fourier transform, and you get the one-dimensional Fourier transform with frequency variable w. Here, the two-dimensional frequency variables are u and v. These two are just definitions of the two-dimensional and one-dimensional Fourier transforms.

Now let's go a little further. Consider the simplest example, the simplest case, and we try to compute the Fourier analysis. Fourier analysis is a two-dimensional function of u, v, but we set v equal to zero. That means we only consider Fourier coefficients along the u-axis—that's just one line through the two-dimensional Fourier spectrum. Given v equals zero, this becomes a one-dimensional function. So equation seven becomes equation nine. With v equal to zero, you get this part.

Rearrange this a little bit, because now the kernel, the exponential function, does not depend on y. So the integral with respect to y can be grouped into the inner integral. You get this part. And you see that the expression in the bracket is nothing but a vertical integral. Because you have a two-dimensional function, you just do the line integral along the y-axis. So this part in the bracket is nothing but a vertical integral.

slide22:

So this is nothing but a vertical integral. A vertical integral is the projection profile when theta equals zero. So that is a vertical integral—you get this.

Now, in two-dimensional Fourier analysis, F of u, zero—F of u, zero—the vertical integral is p at theta equals zero of x. Put this part into the bracket, and then you have this expression.

So this is two-dimensional Fourier analysis along the u-axis when v equals zero—this is the u-axis. And here is the vertical integral. The vertical integral is a function of x in the x–y plane. And this part is one-dimensional Fourier analysis with respect to the variable x. So this is the one-dimensional Fourier transform. You can write it as the one-dimensional Fourier transform when theta equals zero, and the variable is u.

This is the one-dimensional Fourier transform in the general case for an arbitrary angle theta. So this is a special case of what I called the Fourier slice theorem. If you have a vertical projection profile, and you perform one-dimensional Fourier analysis, you get a one-dimensional Fourier spectrum—you got a one-dimensional Fourier spectrum. It is nothing but the projection profile in the two-dimensional Fourier-transform space along the u-axis. So this is a special case of the Fourier slice theorem.

slide23:

When theta equals zero, you get the two-dimensional Fourier spectrum profile as a one-dimensional profile along the u-axis. How do you get this special-case profile along the single line u? Set theta to zero. You obtain vertical integrals. Vertical integrals give you a one-dimensional signal. You perform a one-dimensional Fourier transform, and you get the spectrum. So you do the physical measurement to get this profile; you perform the one-dimensional Fourier transform; you get the projection profile along this line. In the case I just showed, this line corresponds to theta equals zero, along the u-axis.

In this way, you recover Fourier-space information along only one line. With an X-ray measurement, you recover Fourier-domain information. Ideally, you want all the Fourier information recovered. Then you perform the two-dimensional inverse Fourier transform and reconstruct the underlying image f of x, y. The Fourier slice theorem is not limited to theta equals zero. The theorem claims that for arbitrary theta, this holds true. For an arbitrary theta, you have an arbitrary one-dimensional projection profile. You perform the one-dimensional Fourier transform and recover the profile along the corresponding line. As explained earlier, if you keep theta changing from zero to one hundred eighty degrees, the line sweeps the whole Fourier space, and you recover all the information. That is the idea.

Now the heuristic: if you have the vertical projection property proved as such, you can immediately understand that the general case must be true. Why? Because the two-dimensional Fourier transform has a rotation property. If you rotate the object by thirty degrees, the two-dimensional Fourier spectrum is also rotated by thirty degrees. Since the angle theta is arbitrary due to the rotation property, you can select the angle as you set up the system. So if you ask, “I have this projection profile; I perform a one-dimensional Fourier transform; will I get the same type of profile along the corresponding line?” The answer is yes. I can select my x-axis along that direction and my y-axis perpendicular to it. In this rotated x–y coordinate, I can use exactly what I explained. Therefore, the Fourier slice theorem must be true. You can understand this immediately from the rotation property of the two-dimensional Fourier transform.

We can also do it mathematically. We introduce a rotated coordinate system t, s. For u, v, we introduce t, s; and actually this t, s is better placed in the x–y space. So we move this to the x–y plane.

slide24:

You can link the t, s coordinates to x, y through a straightforward coordinate transformation. Then you can go through the mathematical derivations to show the Fourier slice theorem in the general case. The theta angle is the same in the u, v domain and in the x, y domain.

With this coordinate transform, p of theta, t, for angle theta as a function of t, can be expressed as a vertical projection in the t, s coordinate system. Here s is the vertical direction in the t, s system.

Next, perform the one-dimensional Fourier transform. The exponential factor is e to the power minus j two pi w t. Then do the same trick as before: insert the definition of the projection profile for angle theta into the bracket. You get the resulting expression, pretty much like the simplest case we did.

slide25:

Then you change the transform back from the t, s system to the x, y coordinate system.

You can express it this way. Rearranging a little, it can be expressed as a two-dimensional Fourier transform, capital F. The u component is w cosine theta, and the v component is w sine theta. Because in the two-dimensional Fourier transform you have x u plus y v, the factor w is redistributed back, giving this relationship.

S theta of w equals F of open parenthesis w cosine theta, w sine theta close parenthesis.

This is nothing but the general form of the Fourier slice theorem. This is the mathematical derivation. You can review it yourself and check Chapter 3 if needed.

slide26:

Let me visualize what we mentioned.

You have the underlying image f of x and y, and using X-ray measurements, you get the projection profile p of theta and t. For a given theta, you perform a one-dimensional Fourier transform with respect to t. You get a line profile, making the angle theta in the two-dimensional Fourier space.

But theta can be changed as you wish from zero to pi, that is, zero to one hundred eighty degrees. So you get a line for theta equals zero, a line for theta equals fifteen degrees, a line for theta equals eighty degrees, and a line for theta equals one hundred seventy degrees, for example. When theta changes from zero to one hundred eighty degrees, all these radial lines will fully cover the Fourier space u, v. All the values are measured this way.

When you know f of u, v completely, you perform the inverse Fourier transform and recover f of x, y. So the analytic process here is completed using the Fourier transform, particularly in the form of the Fourier slice theorem. This is a Fourier imaging example: we explain measured data in the Fourier space. We try to fill the Fourier space completely, and then we can perform image reconstruction.

This is just a little more specialized idea of how you use the Fourier transform to do CT reconstruction.

Okay.

So let me go a step deeper. This is the general idea. First, I explained the general idea, then I explained the Fourier slice theorem, a little more specifically.

slide27:

Now let’s go even deeper and give a specific algorithm called filtered backprojection. This is nothing but the inverse Fourier transform, not in the rectangular coordinate system, but in the polar coordinate system, because we keep changing theta. So we should represent the inverse Fourier transform in polar coordinates, so that what you measure in polar coordinates fits directly into the formula.

Let me go through the mathematical steps. This is the inverse two-dimensional Fourier transform. If you know the two-dimensional Fourier spectrum, capital F of u, v, you perform the inverse transform, and you get f of x, y.

My motivation is to use polar coordinates. In polar coordinates, you have radial lines in the Fourier space. That radial variable is W—this is really the rho I showed you before—and the polar angle is theta. Sou equals W times cosine theta, and v equals W times sine theta. That is the polar-to-rectangular coordinate transformation.

For d u d v, the small differential area element, in polar coordinates, you need W dW d theta. The small area element d u d v becomes W dW d theta in polar form. Here, W is the radius. A small angle d theta gives an arc length W d theta. A small radial increment dW gives the thickness. Multiply them together and you get the small area element. This is just your calculus: W dW d theta.

slide28:

So you put the inverse Fourier transform in terms of the polar coordinate system. In polar coordinates, the angle theta goes a full circle, and the radius W goes from zero at the origin all the way to infinity. That covers the full space.

Then we do a little trick. We decompose the full circle into two half-scans: from zero to pi, and from pi to two pi. For the second part, I write theta plus pi instead of just theta. That is why the interval from pi to two pi becomes zero to pi after the change of variables. So you get this form.

For parallel-beam geometry, the angular range of zero to one hundred eighty degrees is enough. If you scan from zero to three hundred sixty degrees, you simply double the information; you really need only half of it.

Mathematically, we handle this by changing the capital F of W, theta, plus one hundred eighty degrees back to a function of theta. Use the identities cosine of theta plus pi equals minus cosine theta, and sine of theta plus pi equals minus sine theta.

slide29:

Taking all these trivial transformations, here is the property we use and can easily verify in Fourier analysis: when the angle shifts by one hundred eighty degrees, you keep the same angle but reverse the sign of w. With this, you only need the angular range from zero to pi.

Next, change W to minus W. Then the radial limit from zero to infinity becomes zero to minus infinity. Swap the limits, and the inner integral runs from minus infinity to plus infinity. Put everything together and you have:

“f at x, y equals the integral from theta equals zero to pi, integral from w equals minus infinity to infinity, S theta of w times absolute value of w times e to the j two pi w t d w, d theta; with t equals x cosine theta plus y sine theta.”

Where t equals x cosine theta plus y sine theta.

All these are just mathematical details—if you get lost, review and you will be able to follow. The key point is that we end up with this formula. This formula is what we call filtered backprojection. If you got lost somewhere, it is not critical; you can review. But trust me for now—through these steps, you get this result. This is filtered backprojection.

Why “filtered”? Because here you see S of w—the one-dimensional Fourier spectrum along the radial line—and it is multiplied by the absolute value of w. If you performed the inverse Fourier transform without the absolute value of w, you would go back to the original projection profile, the one-dimensional signal you measured. But with the additional factor absolute value of w, the original Fourier spectrum is modified.

When w is small—near the origin—low-frequency components are weighted small. When w is large—at high frequency—the weighting is large. So S of w times absolute value of w is a high-pass filtering: the high-frequency components are elevated in proportion to absolute value of w. After this spectral modification, you perform the inverse Fourier transform and go back to the projection domain. Because of the high-frequency enhancement, this inverse transform is no longer the original projection profile. Instead, it is a modified, high-pass-filtered projection profile. It is not p theta of t anymore; it is q theta of t, the filtered projection profile. That is why we call it filtered.

What do I mean by backprojection?

You filter first—this one-dimensional filtration—and then you put the filtered values back into the field of view. According to the argument t equals x cosine theta plus y sine theta, for any x, y you retrieve the value q theta of t from your filtered projection profile and accumulate it back over angles. That step is the backprojection process. “Filtration” may be clear; “backprojection” can feel a little confusing, but it simply means smearing the filtered profile back across the image along the corresponding rays.

slide30:

So here we have this picture. The backprojection is visualized. This is the projection profile after filtration, so it is a filtered projection profile. This filtered projection profile, Q theta i of t, is really smeared back over the field of view. Think of this projection profile as being smeared back from this particular direction. You have many filtered projection profiles; you sweep all of them and smear them back over the field of view. Add them together to get the result.

For a given x, y, what is the contribution from a given Q theta? That is the question I want to explain. Suppose x, y is here. For this x, y—the point in image space—how much contribution do we get from a particular filtered projection profile, here Q theta i? You do this inner product: you get t. Spoken clearly: t equals x cosine theta i plus y sine theta i. That t is the distance between this line and the central line, both perpendicular to this direction, making an angle theta i. For any point along this line, the inner product gives the same t. So, as long as the pixel is on this line at distance t, you retrieve this value from the filtered projection profile. For all points on this line, t is the same. You compute t, you get the value, and you put that value back for any x, y on this line. That value is the same here, here, and here—all along the line. If that value is one, then along that line every point gets one. This is what I call smearing back.

That was for this theta, theta i. For another angle, you smear back in a different way. The process is linear—the integral means summation. For each theta, you retrieve the value from Q theta, then add it back to the particular point x, y. For another angle you do the same. You can view it like this: from this filtered projection profile, you get the value here; along a different direction, another projection profile gives you another value through that same point x, y. So, heuristically, from the filtered projection domain, any profile is smeared back uniformly, its value covering all points along its line. All projection angles are added in the same way, and then you have the image value recovered. Think about that. I will show reconstructing examples in a few minutes, so you can get a better idea. OKay.

Now, several slides with green buttons are about the reconstruction filter. Here, the reconstruction filter—a high-pass filter—is the absolute value of w. This filter does not have an inverse Fourier transform by itself, because the absolute value of w is not integrable. However, we can introduce a band-limited assumption: for a given projection profile, you have a maximum frequency, a maximum bandwidth W.

So the filtration really needs to be done with this truncated filter. In the original Fourier space, the ideal is the absolute value of omega extending to infinity. We assume a maximum bandwidth capital W; outside that, nothing is meaningful for the data. With this window, the spectrum becomes integrable. You truncate the otherwise divergent high-pass filter, then perform the inverse transform with respect to this truncated version, H of omega. You compute the inverse Fourier transform as the spatial-domain counterpart of the high-pass filter. You do the computation, and you get this part.

slide31:

Now we assume capital W is the maximum bandwidth. Because of that, we can use the sampling theorem. The projection profile can be expressed through the sampling kernel in terms of discrete sampling points P theta at k tau—where tau is the sampling step.

Likewise, the filtering kernel, under the assumption of limited bandwidth capital W, can be expressed in terms of sampled values. So you get these two equations.

slide32:

So the continuous-domain expression can be expressed in terms of discrete data points. You get the filtered projection profile in terms of sampled data points. This is the band-limited high-pass filtering.

You can read more if you are interested, but anyway, these are practical implementation details about high-pass filtering for filtered backprojection. What I have explained here is only the two-dimensional case.

slide33:

And in the three-dimensional case, you have an extended, higher-dimensional version of the Fourier slice theorem. The mathematics is a little more complicated, but the essential idea is still the same.

You have a 3-D object. You view it as a superposition of 3-D waves—not 2-D waves—with all kinds of orientations and frequencies. When you form a projection from one direction and superimpose contributions along that direction, you only get information for the wave components whose directional vector and frequency components lie in the plane orthogonal to the projection direction.

In other words, the 2-D Fourier transform of a 2-D projection gives you a central plane through the 3-D Fourier transform of the object, oriented perpendicular to the projection rays. This is really just the extension of the 2-D Fourier slice theorem.

I don't want to confuse you too much—again, you see this nice green button.

slide34:

So, analytic approach: we convert X-ray data into Fourier space—equivalently, we could work in Radon space—but here I explain the Fourier-space processing. We invert the Fourier-space transform according to a closed-form formula, like the one I derived. Filtered backprojection is an analytic formula; you do not need iterative reconstruction.

In the reconstruction process, some filtering or prior processing steps may be used. An iterative algorithm is very useful when the data are not complete—say, some views are missing, projections are blocked by metal, or there are other imperfections in data acquisition. Analytic pros: you don't need an iterative process; you have a formula—that's nice. But it assumes low noise and complete data.

So the two approaches—iterative and analytic—have their strengths and weaknesses. Nowadays, iterative methods are often more popular when we need low-dose reconstruction or a very short acquisition time. And filtered backprojection, as I explained here, is just the mathematical formulas plus the geometrical pictures I mentioned.

slide35:

Now I'm going to show you some numerical examples at the end so you have a better understanding.

These examples let you see how what we learned works on real cases. After that, we'll take a quick look at a few clinical images.

slide36:

So the real example is a simple cross-section with a small bright disk.

You do one-dimensional projection data acquisition at an angle theta. You get this profile, and you see a peak due to the bright disk. This is at an angle theta. Theta can be changed from zero degrees to pi. As theta changes, the peak traces half of a sinusoidal curve in the sinogram. For this given view, the projection profile is here.

We have this example, but at different elevations—that is, at different theta angles—you have different data. So, if you do backprojection, what will happen? I explained a lot about backprojection: backprojection is the filtered projection profile smeared back over the field of view. Let me show you what happens. First, let's not bother with any filtration. This is the projection profile, and I backproject it.

slide37:

So this is the projection profile, something like this. You do backprojection—you smear the profile back along the original X-ray path over the field of view. The higher value here is put back, so along this line, the value is placed at every pixel on the line. Here, the value is a little low, so along this line, every point is a little dark. You find the pixels on that line, take the value, and smear it back.

This is from one direction. You do the same from every direction. The smear-back results are added together, because the integral with respect to d-theta is just a summation. You smear back one projection—you get a picture like this. You take another projection—you smear back the other way. You add the two together. Many projections add many contributions together.

slide38:

From one hundred eighty degrees, you add everything together. You get something not too much different from the truth, but it looks very blurred. You still see a region that is a little high, because this part of the projection is high. When you backproject along the original X-ray paths, that high value is placed along those lines.

However, there are artifacts. In the smearing-back process, a high value here gets spread to places that are not actually high in the object. Backprojection alone smears the high values around. So, simply collecting projections and doing backprojection is not enough to recover an accurate image. That is why we really need to filter the projections.

slide39:

So you have a projection profile. You don’t just smear it back. First, apply a high-pass filter with kernel H. After filtering, you get a high-pass-filtered projection profile, called P-prime of theta, t.

Then you perform backprojection. That’s filtered backprojection: filter each projection with H, and backproject every filtered projection profile.

slide40:

This is the sinogram. The filtering is one-dimensional along the horizontal direction, so a high-pass filter emphasizes edges.

Here is the original sinogram; here is the filtered sinogram. Once it is filtered, we use these data for backprojection.

slide41:

Then you can recover the original image. This looks very close to the original. If you have many angles, with very small angular increments, and the detector spacing is very small, the result becomes closer and closer to the true image.

This is just a numerical example. The next few slides show a MATLAB implementation—also your homework—so you can read through and try it yourself.

slide42:

MATLAB has toolbox commands that implement projection, filtering, and filtered backprojection.

The keyword is radon. You supply an image and theta. Theta can be a single value, which gives one projection, or an array, which gives many projections.

slide43:

The inverse transformation is done with iradon. The projection beams are laid out line by line, and the default line spacing is one pixel—one unit apart.

slide44:

The setup looks like this: parallel-beam geometry with sensors on one side, a source on the other, and the system rotating by the angle theta to collect projections.

slide45:

When you do projection, they basically decompose a single pixel into a two-by-two matrix, then shoot rays through all these centers into the projection domain.

If a ray hits the center of a detector bin, that detector takes the full value. If it hits the boundary between two detector bins, the value is divided equally. If it hits at an arbitrary location, linear interpolation is used.

slide46:

Here's a sample Radon code you can run yourself. It generates an image with a white block, then produces the projection data.

After that you apply the filtration. The red arrows simply point to figures that appear on the next slide.

slide47:

You get this projection profile—the sinogram.

slide48:

Next, use the inverse Radon code, which essentially performs filtered backprojection. You can also choose linear interpolation, and if you set the filter to "none," that means no filtering—just simple backprojection.

slide49:

On the top row images, you can see the difference: the unfiltered backprojection is blurry, while the filtered backprojection looks much closer to the original image. This is the key to reconstructing CT images.

The bottom row shows some clinical examples. It feels like magic. Review the material; if you didn't do the preview and you don't review, you won't catch all the tricks. But if you read carefully, you'll understand the secret: the X-ray machine sends beams you never see, yet your internal structure is clearly revealed.

You can resolve features down to about one-third of a millimeter, with every detail laid out by the algorithms I just explained. It's an amazing achievement—inner vision with X-rays.

slide50:

For homework, review what we did with the block, and do the same steps using an ellipse.

Work through the MATLAB code. Some variable names may not be obvious from the slides, but if you open MATLAB Help or search online, everything is straightforward. Spend some time, get familiar, and see how filtered backprojection works for you.

That's all for today—thank you.