MITOCW | mitmas_531f09_lec11_1.mp4

RAMESH RASKAR: Kind of a summary of a lot of the things we talked about. So if you remember in the beginning, he said, you can just start with a pit. And then it kind of develops with the lens. But from even here, you can go down two different parts, either compound eyes, where each sensor or set of sensors have their own optics, like a sort of straw, or same lens-- sorry, the same pixel, might get image from multiple lenses, like here, right? So that's superposition. So this is a position, and this is superposition.

And that concept of a position or superposition applies to all three types, shadows- or refraction- or reflection-based techniques. So we saw this last time, and we'll see how-- we already have some projects that are inspired by biological vision. You know, Matt is trying the chicken. And I think it's going to be--

[LAUGHTER]

It is going to be very popular. And I believe Santiago-- where's Santiago? Oh, yeah, his triangle-- the piston, kind of-- so some really great ideas. So I'm glad a lot of these concepts are coming together in the final projects.

So today, we'll talk about coded imaging. And the concept here is very simple, OK? So I'll start with this one, which is you have a taxi zipping very fast. And you want to kind of take a photo in such a way that you can recover the sharp detail afterwards in software. So it's a form of a co-design between how you capture the image and how you process the image.

In a typical film camera, or even it is digital camera, you take the picture, and that's basically the end of the story. And here, you're trying to do something clever about how the picture is taken. So of course, there are other opportunities of capturing this. You can either take a really short exposure photo. But that's going to be very dark. If you take a high ISO, you can recover some information, but still quite dark. Or you can just take a long-exposure photo by keeping the shutter open. But then you will get a blurry photo, which is well exposed, but a lot of the high-frequency details are lost.

And then if you try to apply some deblurring, you'll get a result that looks like this, which is kind of reasonable. You can see the number one on this, I guess, Thomas Train. But you get a lot of banding artifacts and a lot of repetition and noise here.

AUDIENCE: So what are those lenses?

RAMESH RASKAR: This lens? So when you try to recover this information, you start getting this banding artifacts. And we'll see it in the next slide, why that happens. So what's going on here is that if you have a sharp photo-- if you have a blurred photo, you can basically represent that as a sharp photo, where there is a convolution of the sharp photo with some kind of a convolution filter, OK? So if you look at-- where's my laser point?

If you look at this-- the tip of letter one here, it's been blurred by a certain amount of pixels in the horizontal direction. And if you keep the shutter open for even longer, it'll blur correspondingly longer. So you have basically a 1D convolution that's converting this image into this image. And of course, the goal usually is this is a photo that you capture and you would like to invert and get back this photo.

So one would say, OK, this converted with this gives me that. So just [INAUDIBLE] using the same filter and maybe you'll get that back. That doesn't work because something called division was 0. And the way to think about that is in the Fourier domain because convolution in the image domain, our primary domain, is multiplication in the Fourier-- just standard Fourier transform.

So if you take the Fourier transform of this and multiply that by the Fourier transform of this, you will get the Fourier transform of this, OK? So let's say we take this photo. Find the Fourier transform here. Multiply that by the Fourier transform of a box function, which is a sync. So what that means is that I'm going to take the lowest frequency, multiply by that value. I'm going to take the next frequency, multiply it by this value. Next frequency, multiply by this value, and so on, right? We're just going to multiply each of the frequencies in the image by the amplitude of the Fourier transform of this.

And you can already see that lower frequencies will be preserved, but higher frequencies will be highly attenuated. But there's also something strange happening. Even some of the lower frequencies are actually being set to 0, which means that in this photo, these frequencies are missing altogether. They have been suppressed. So it's not a traditional low pass filter. It's a low pass filter where some of the even lower frequencies are also being nullified, which means that if I tried to recover from this photo, this photo, there is no chance because I have already attenuated and have lost all those frequencies.

So the moment you take the photo, the damage is done. And there's nothing you can do to recover those frequencies because in the Fourier domain, all you have to do is take the Fourier transform of this and divide by the Fourier transform of this, which is this. And it will give you the [INAUDIBLE], OK? But the Fourier transform has some zeros, so you cannot divide those frequencies by 0 and recover an image.

So the culprit here is really this box function, which is equivalent to-- when you release the shutter, opening the-- release your shutter button-- opening the shutter and keeping it open for exposure duration and closing it. But that's the most natural thing to do. But apparently, it's not the most effective.

So what if you change that? What if you changed that? And instead of keeping the shutter open for the entire duration, you open and close it in a carefully chosen binary sequence. So for some time, the shutter was open, then shutter's closed. It's open for some time. Again, it's closed. Here, it's closed for quite some time, open for a short time. And so on.

So at the end, you still get just one photo. But now something magical has happened because first of all, if you look at this number one, you'll see that it's not the same as before. It has-- it seems to have these replicas. And the reason why this is better is you take the Fourier transform of this. It's actually flat, which means it's preserving all the frequencies in the image.

So we can be sure that, in this photo, all the spatial frequencies-- low frequencies, high frequencies-- they're all preserved. Of course, they're attenuated. It's not as high as-- it's not 1.0. It's reduced. Maybe it's 0.1 or so. So they're all attenuated, but there is still some hope to recover this photo back from this because, in the denominator, we will not have seen.

So of course, if you try to implement this mechanically, where you open the shutter and then mechanically try to close the shutter, that will be problematic. So what we did was we used an LCD-- actually, a ferroelectric LCD-- that becomes opaque and transparent. And in the old virtual-reality screens or even some of the games, you have these eyeglasses that flicker at 60 Hertz for time sequentially so that you can see the left eye versus, like-- because they're the same glasses.

And a traditional LCD, unfortunately, doesn't have a very high contrast. And Simon is discovering that one more time. But the ferroelectric LCDs have a contrast of 1,000 to 1. So when it's opaque, the amount of light that passes through, as compared to when it's transparent and the amount of light it passes through, the ratio is 1 of 2,000. So when you turn this ferroelectric LCD off, it's really, really opaque. Yeah.

AUDIENCE: Couldn't you just do a high-speed video or just a [INAUDIBLE] taking video and put out your frames?

RAMESH RASKAR: So the question was, why not just capture high-speed video and take all these frames, right, and then put them together? The problem is each of the frame will be extremely dark. So you are basically adding up a lot of noise. Every frame is dominated by noise.

AUDIENCE: Yeah, yeah.

RAMESH RASKAR: So when the shutter is transparent, that goes through. When the shutter is opaque, light doesn't go through. And that's your 1010 inquiry. So, again, the idea is very simple. Instead of keeping the shutter open for the entire duration and getting a well-exposed photo, the shutter is open for only half of the time.

AUDIENCE: There is an issues there. The support for the representation of the Fourier domain of that function that you describe there is infinite, right? So you actually truncate this in order to--

RAMESH RASKAR: It's not infinite because you still have some width.

AUDIENCE: Right, but you have infinite high frequencies there by the sharp conditions, right?

RAMESH RASKAR: Yeah, you can think-- I mean, you can think of this one goes to infinity. But there's hardly any energy left. So although it goes to infinity, there is not much energy left.

AUDIENCE: But when you get to invert the process then, that's why you're still not getting the perfect images to--

RAMESH RASKAR: In this case.

AUDIENCE: --in this case as well. You still lost some high frequency, right?

RAMESH RASKAR: So you haven't seen the results yet for this.

AUDIENCE: You show the text.

RAMESH RASKAR: Yes. So this is what it looks in this case. But it's a very controlled experiment in a laboratory. So you take the toy, and you move it in a very controlled way. And this is what you get in a traditional camera. And this is what you get in the flutter shutter. So these are real photos. And, yeah, you're right. I mean, you still get some noise. And actually, if you can pair this with ground truth, you'll see that it's OK, but it's not perfect.

AUDIENCE: Yeah, so let's say that you took the 0s from the [INAUDIBLE], right? And you just replaced it by something that is pretty close to 0, but not 0. And if you invert the process--

RAMESH RASKAR: From here, this is what you get.

AUDIENCE: That's-- OK--

RAMESH RASKAR: There's a deep learning of this.

AUDIENCE: OK.

RAMESH RASKAR: Yeah, and that's this loss of these frequencies also shows up as these artifacts at regular frequencies, at regular intervals. So, again, this one-- this doesn't go to infinity all the way. [INAUDIBLE]. It cuts off and corresponding to the width. The width of this post was very short than yesterday were very far away. OK? Yeah?

AUDIENCE: The filter is dependent from distance?

RAMESH RASKAR: The filters depend on multiple factors. So if your toy is moving or your taxi's moving really slow, then there is no need to-- in this case, the sequence was about 51-- actually, 52 vector long. So let's say your exposure time is about 104 milliseconds. It's open for two milliseconds. Here, it's open for four milliseconds, off for two milliseconds, four milliseconds, two. Maybe it's off for eight milliseconds, two, and so on.

AUDIENCE: Yes, but--

RAMESH RASKAR: But with a vector length of 52.

AUDIENCE: This filter is in time?

RAMESH RASKAR: In time.

AUDIENCE: And you think about filter from space?

RAMESH RASKAR: It corresponds automatically to filter in space.

AUDIENCE: Yeah, so [INAUDIBLE] it's dependent on distance, if you--

RAMESH RASKAR: Yeah, the speed, you mean.

AUDIENCE: --of faraway objects.

RAMESH RASKAR: Yes. So your actual blur in the image may not be exactly 52 pixels. It might be 10 pixels. It could be 100 pixels. So your 52 vector is going to stretch or shrink based on how fast the object is moving. And you're saying that it also depends on how far the object is in space because faster-moving objects. And you mostly have to think about image space motion because the speed in the real world and-- the distance are they get-- you divide to normalize by the distance. So you only have to worry about the image space distance. Yeah. Go ahead.

AUDIENCE: Could you get a similar effect if you had, like, instead of a hooded shutter, it could have flasher?

RAMESH RASKAR: Yeah, exactly. So if you're in a dark room, you can just-- if you're in a dark room, then you can just strobe the light, rather than opening and closing the shutter.

AUDIENCE: I think we might have a mobile demo of that scene.

RAMESH RASKAR: [LAUGHS] Well, I don't know how fast you can--

AUDIENCE: Well, the problem is you can't [INAUDIBLE].

RAMESH RASKAR: Yeah. So what are some-- let's look at some pictures, actually. So here is a demo. I think I've shown it to you before. This is on Broadway. This women try to figure out the car make and the license plate number. What's the license plate number?

AUDIENCE: 458.

[INTERPOSING VOICES]

AUDIENCE: 468.

AUDIENCE: Something.

RAMESH RASKAR: And the company?

AUDIENCE: [? One more time. ?]

RAMESH RASKAR: Yeah. So you get a reasonable result. But going back, what are the limitations of this method? Yes.

AUDIENCE: You need to know the motion or the direction of the motion.

RAMESH RASKAR: No, you need to know the point spread function, how the blur is created. If the car is moving from left to right versus right to left, you need to know that because the way your point spread function will be imposed on the scene will be different.

AUDIENCE: You still inspect the lighting.

RAMESH RASKAR: You just have the light-- very important, right? So this image is about half as bright as this one. What else?

AUDIENCE: I guess there should be a little less of an acceleration of-- all of them should be moving the same--

RAMESH RASKAR: Exactly. So whatever is moving has to move at a constant speed. If we did 100 milliseconds, it picks up speed, then your assumption that the 52-length vector will map to some stretched or shrunk version of 52 is not valid. Some parts will go faster and slower. What else?

AUDIENCE: [INAUDIBLE]

RAMESH RASKAR: Sorry?

AUDIENCE: If the object is moving in space, [INAUDIBLE] distance and then [INAUDIBLE].

RAMESH RASKAR: Yeah, so you-- so if it's moving in a perspective, for example, it's not so bad because you can rotate the image. And again, it'll become-- so that's not acceleration. That's still constant speed. It's acceleration in the measurement, but in the real world, it's still constant speed. So you can play with those tricks. You can either go to object space or you can come back to image space to make sure there is no acceleration. It's all linear.

AUDIENCE: So does this technique still work if you're moving in multiple directions at once over the duration?

RAMESH RASKAR: So if you have multiple cars, for example, and they're all independent, then it's fine because I can say this car is going this way. That car is going this way. As long as it's moving in a straight line at a constant speed, you're OK. But if the two cars overlap, what happens? Our model fails again. If two cars are partially overlapping during the exposure, it's possible, but it's more challenging because you don't know exactly how fast the two cars are moving. Yeah.

AUDIENCE: Sorry, might we need to know how fast the car is moving when you're setting up your shutter?

RAMESH RASKAR: No, when you're-- OK, so when you're setting up your shutter, if the car is moving really slow, and you don't expect it to blur by 52 pixels, and you expect it to blur by only 10 pixels, then using a 52 sequence is overkill. Maybe you should use a new sequence that's only about 10 long or 11 long, right? So it's just like--

AUDIENCE: OK, but that's just so you can get more light.

RAMESH RASKAR: No, that's so that it's most optimal for that setting, right? So it's like setting an exposure time. When I take a picture, the camera automatically decides what the exposure time should be. Similarly, you should look at the speed of how things are moving maybe with an ultrasound Doppler or whatever. And it says, things are not moving at all. So I should not use the flutter shutter at all. Until they're moving very slowly, maybe I should use a 10 long sequence. If things are moving a lot, maybe I should use a 52 sequence.

And to answer your other question, where you need to do is when we solve the system, we need to know how long the blur is, which is true in other cases as well. You need to know how much the blur is.

Another major disadvantage is let's say I want to take this bottle. And if I just rotate it and motion blur that, it will not work. For any point in the front that you're looking at it, it'll work. But the point that was in the back, that all of the 52 sequence-- maybe for the first 10, it was occluded. And the remaining 42, it was seen. You have to know exactly when that point became visible during that 52 window.

So in general, the technique works well when things are moving naturally. But if somebody wants to do this kind of an experiment, or move things behind an occluder and move out, those are very challenging scenarios.

AUDIENCE: Can you combine both horizontal and vertical [INAUDIBLE] masks?

RAMESH RASKAR: Vertical, horizontal is fine. You can-- it doesn't matter. It could be moving vertically. Basically, your point spread function-- the blur function will be vertical rather than horizontal.

AUDIENCE: Yeah, no, but if you have a combined motion, vertical and horizontal, you have to encode this with a mask?

RAMESH RASKAR: No, no, no. So let's say the two cars-- one is moving--

AUDIENCE: [INAUDIBLE] diagonally from their [? English ?] way, right?

RAMESH RASKAR: That's fine. As long as it's in any one direction, it's OK. So let me draw it.

AUDIENCE: But if you take a sharp turn, you pass through, or--

RAMESH RASKAR: Yeah, exactly. So you have to assume that the point-- so the basic assumption is that if you take any point in the scene, it's moving in a straight line, let's say. And if you have an object, and every point of that object moves in a straight line, OK. It doesn't matter which direction and what speed.

AUDIENCE: So this doesn't help at all with any image stabilization if somebody's holding the camera.

RAMESH RASKAR: It helps as well. So if you have-- let's say you have a camera shape. And I take a picture of an LED, and it creates some curve like that because that kind of shape. If I know that curve, maybe I can put a gyro. Then I can, again, figure that out. So the problem here really is the point spread function or the blurred function is very critical. And this is what we want to study about half of the class.

And the concept is very, very, very interesting because light is linear. So eventually, it's very linear. What happens to a point happens to the rest of the object. So if I have a car that's moving, and I tell you how exactly one point of the car is behaving in the image, I can tell you automatically how the rest of the car is behaving in the image because it's going to do-- all of it is going to have the same spread image.

So you can either-- for experiments, you can just put an LED on the car and see how that LED moves. And that tells you everything. And I'm sure you use this trick in other scenarios where you look at a very small impulse and see how the response is. There's also an impulse response. For those of you in audio, you might want to check and see how the room [INAUDIBLE].

And when you're trying to find a speed of a car, [INAUDIBLE], a very small impulse. And it answers and comes back. It does. The point spread function for your time of flight.

So that's the same concept here. You just want to call leading the world, take a picture, and see how it works. And this whole field of order dimension is basically engineering of the point spread function.

So if you take an ordinary camera, a film camera, and take a picture, you have no control over how light is spreading-- if something is moving or a focus has different color spectrum. And so an order dimension basically means you want to control how something is spreading on the image.

So we're going to engineer activity of the camera. So in this particular case, a point that was moving created a blur like this. And by engineering the time point spread function, it stops looking a bit like that. It's going to look like that, all right? It's going to look like fashion [? wise. ?] And then it just turns out that this one is easier to deal with than this one. So that's the basic concept, engineering or actively changing the point spread function.

So this is very counterintuitive because you would say, let me just build the best lens and the best exposure time. And so that kind of mimics the human eye. And once I have that, I have the best possible picture. But when it comes to actually extracting information from that scene, it turns out you need to strategically modify how the camera works so that all the information is somehow preserved.

Now the problem is, even after you are very careful and you have captured that image, it's still going to be somewhat garbled. It's going to be mixed in. But that's where the co-design comes in. So once you have this image, there is some hope, there is some computational technique, that will allow you to go from here to here.

And this is what kind of separates an animal eye from a computational eye because in most scenarios, an animal eye is just going to take the picture and try to make the best sense out of it. But a computational eye is going to apply a lot of processing to this and be able to recover that. As far as I know, animals don't have deconvolution circuitry or deep-learning circuitry. I can look at a blurry image and kind of figure out. I mean, this was a challenge for you, right?

Right. So we have pretty sophisticated eyes, but we're still not able to deep learn what this is. If you have some prior knowledge of how the Volkswagen logo looks like, maybe you can say, OK, maybe that was this. But on the other hand, if I give you this, you're immediately willing to believe that this photo is a blurred version of this photo.

And so kind of thinking about that is when you go from here to here, information is lost. When you go from here to here, we're trying to recover some information. So going from a sharp photo to a blurred photo is easy for us because we just have to lose some information or to imagine what it would look like if some of the information is removed from this image.

So the goal of coded imaging is to come up with clever mechanisms so that we can capture light but not just by converting photons into electrons, but actually modulating those photons, either blocking them or attenuating them or bending them, and so on. So that that's why a computational camera is doing the computation not just in Silicon but also in optics.

OK, so that was what we can do to preserve information in case of motion blur, right? And the circuit is very, very simple. You just take the hot shoe of the flash, and it triggers. When you lose the shutter, it triggers the circuit. And then you just cycle through the code that you care about.

What can we do for defocus blur that is for motion blur? What can you do for defocus blur? We, again, want to engineer the point spread function.

AUDIENCE: Spatial coding.

RAMESH RASKAR: Spatial coding. How would you apply spatial coding?

AUDIENCE: Coded aperture?

RAMESH RASKAR: Coded aperture. So this is coded exposure, coded aperture-- very easy. And all you're going to do is put some kind of a code in the aperture of the lens. And this is how, actually, it started in the days of-- in scientific imaging, especially in astronomy, coded apertures are very well known. And those of you attended Professor Han's lecture on Wednesday, that's what he talked about, coded apertures.

So I've been following this for a long, long time. And I thought, it must be useful for something in photography. And so I said, OK, let's try to put a coded aperture in the camera and see if we can deal with focus and so on. And that was back in 2004. And we tried it for six months, and it just didn't work. It was really frustrating-- really, really frustrating.

And then one fine day, I said, OK, if you can do this in space, I'm sure we can do this in time as well. And so we did this, and this worked right away, within a couple of weeks. So we went ahead and built this whole system. And that was just a graph paper. And then we said, OK, let's come back and think about this. What's going on? Why don't we get good results?

So it took almost two years to realize that to put this coded aperture in a camera, there are only a few places where you can put it to get good results. So out of that came this particular experiment. So I have a colleague, Jim Kobler, at MG Edge. And one day, he showed me-- this is his lens, by the way. He was telling me the story that he was fishing with his camera, and some creature came out of the water, some kind of an alligator. And he lost his balance, and the boat flipped upside down. Somehow, he managed to flip back in. And the alligator went away. But it completely damaged his camera that was with him, and it just wouldn't work.

So he just took out his lens, which is a standard Canon lens. And he said, let's open it all the way. So he ripped open all the damage. It had all the mud in it and so on. And then he just showed me this thing as is. And it was very fascinating because this is a standard film lens, which, of course, can also be used with a digital camera.

And this is a fixed focal length lens. It's 100-millimeter focal length lens. And when you focus with this, it works in very interesting ways. First of all, it doesn't have a single lens element. It has multiple lens elements. So when you change the focus, it has to do some really interesting things. It has to deal with chromatic aberration, geometric aberrations, such as radial distortion, and so on. So it has to move all these lenses with corresponding ratios, OK?

So I'll pass this around, and you'll see that there are these notches on this lens that are in a parabolic fashion. So when I wrote this, the internal lens-- the outermost lens and the innermost lens instruments are the same place. But all the inner lenses move with some particular ratio. It's amazing the way it's structured, right? So the multiple lenses are moving every time I move this. And they're moving because they're guided through these groups.

But there's one particular location that does not change in this lens, and that's the aperture. So we said, let's look at this aperture. And back then, it was still a reasonable-looking lens. So we went in our lab, and we cut open all the way. And you can start putting new apertures in this plane. So you can cut open that particular guy and start putting this aperture.

Now it turns out the center of production of this lens is very carefully designed by camera makers to be the same plane where you put your aperture. So when you change your f-stop and decrease it and increase it, it's all happening in the center of projection. Everybody knows central projection?

So when you think about a visual camera, you make this very simplistic assumption. That is a pinhole, and there's a sensor. And when you put a lens, we assume that the center of the lens is the central projection, that this always can be assumed to go to that point. When you have a bunch of lenses, like way over here, where is the center of protectionism? Is it here or here or here or here?

And of course, there is-- you can take a collection of these lenses and create one single center of projection for normal cameras. For professional lenses, that's not true. But for normal cameras, you have the central projection. But again, conceptually assume that all the rays are going through that point because you can replace this whole thing by one single lens in a [INAUDIBLE].

So finding that plane is actually a tricky problem. And in retrospect, it's very easy. If the lens makers are putting everything there, we should put a recorded aperture also in the same plane. So initially we said, oh, let's put it in the front. Let's put it in the back. We tried all those things. But that creates a blur that's not constant all over the image. And it has a lot of issues. But placing it over there, it turns out you get the same blur.

So what exactly happens if you take a picture of a point light, and everything is a sharp focus? Nothing changes, OK? If you have just an open aperture and take a picture of a point light, it looks like a disc. Now what's going to happen when you put this code, like the 7 by 7 mask, and take a out-of-focus picture? What will happen to the LED?

AUDIENCE: It's going to look like a code.

RAMESH RASKAR: It's going to look like the code, right? And why is that-- why is that happening? So let's think about [INAUDIBLE] focus. So we have our lens, right? And we have a point light. And we will put some code here. When it's in sharp focus, it doesn't really matter what the code is. Basically, you're talking about half the light, so the photo will be half a square. But other than that, it looks like an ordinary focus.

And that's why if you have some dust on your lens and so on, usually it doesn't matter unless you have the dust all the way on your front lens because the center operation's over here. So if the dust was over here, nothing will happen. The image will be slightly darker. But if the dust is all in the front, then you start seeing distance.

Anyway, so when it's in sharp focus, you just see the point. But let's say that your autofocus here. What will you see? You will see the same exact [INAUDIBLE]. So the [? ray ?] comes in. It's blocked. This ray goes in. It goes through. This ray comes in. It's blocked. This ray goes through, and so on.

So basically, you'll see the same [? boat. ?] If you put the sensor all the way here, you'll see the whole code. If you start moving away, the code will shrink. And eventually, when you put it here, we get another code. That's exactly what's happening here. When it's auto focus, we just see the code, all right?

By the way, this is the same idea behind another project, which is [INAUDIBLE]. So the idea came around at the same time of how to make this happen. OK.

AUDIENCE: Wouldn't the imaging of this code now still have to be blurred? Like, so is that basically multiple apertures that you're seeing?

RAMESH RASKAR: Yeah, so the photo here is nothing-- the photo here that you see is still blurred. It's just that it's blurred in a slightly different way-- strange. Here, it's blurred with that shape. Every point is blurred with that spread function. And you cannot see anything on the resolution chart. But here, if I just promote this guy-- no, that won't work because I'm in a different mode.

If I look at this picture, you will see that-- so this is a sharp photo. It's blurred with disc. And it's blurred with that function. You can already see that it seems to preserve slightly more information. But it's still-- you won't be able to with your naked eye. You'll not be able to figure out what underlying patterns are. But it turns out, after the blurring, you can.

All right, so then you can do these simple tricks, where the person you're interested in is out of focus. But then you can refocus digitally. So this is the input photo and the stock photos, all right? So the same exact trick, which is in case of motion, we created a point spread function that was engineered in one dimension. And here, we are engineering a point spread function that's two dimensional.

So here, we know that the Fourier transform of this 1D 52-length vector is broadband. It has energy at all the frequency. What can we say about this? It's fully transformed. What can we say? It's still 7 by 7. So its Fourier transform is also 7 by 7. When it's 52, its Fourier transform is 52 long.

AUDIENCE: It's more distributed instead of just all being near the center.

RAMESH RASKAR: So in 1D, this is what we saw, right? Its Fourier transform is flat. So there are 52 entries here, and almost all of them are the same. Now we're saying, think about the problem in 2D. And what's the Fourier transform of this?

So first, for this one, the Fourier transform is-- as we see, it's black. And then if you take that in 2D-- so how is the code? I'll give you a hint. If I just take a square aperture, a traditional one, and take a square transform, it will look-- the Fourier transform of this one looks something like this. [INAUDIBLE].

So Fourier transform of this one-- if I take the cross-section here, it's going to look the same. Same thing here for a square aperture. And now you're saying for this crossword-puzzle-shaped item, should be easy. It's going to look just like this one.

So a Fourier transform of 7 by 7 will have a peak in the middle. So the truly Fourier transform will have a peak in the middle. But the rest of the values will be constant. And that's the magic of a broadband code. So if we're placing a broadband code, certainly we have an opportunity to recover all the information.

So it seems very, very long winded, right? If all I wanted to do was create a photo from which I can deblur to get sharp photo, why do I need to think about all this theory, right? And the reason is, when we think about point spread function, it's just traditional signal processing. It's a convolution and so on, and it's much easier to think about convolution and deconvolution in frequency domain than in primal domain.

And in communication theory, everything is [INAUDIBLE]. We think about carrier frequencies of radio stations in frequencies. We say, my FM channel is at 99 megahertz, 100 megahertz, and so on. And we think about guard bands and audio bands and everything interested in frequency domain. And that's because it's signal processing. It's the same thing that's going on here. And convolution, deconvolution-- much easier to think in frequency domain.

Although all the analysis in the frequency domain, at the end, the solution is very easy-- just flutter the shutter or just put a coded aperture. Extremely simple solution to achieve that. So those are all good things about coded aperture. What are some bad things about coded aperture? What are some disadvantages here? It's very similar to the [INAUDIBLE].

AUDIENCE: Half the light.

RAMESH RASKAR: So half the light. Very good. And that's when you talk to people who build cameras, and you tell them, they say, no, no, no. That's not allowed. It doesn't cut the light. Yes.

AUDIENCE: Are the bokehs kind of uppity?

RAMESH RASKAR: The bokehs are-- it depends on your-- I mean, for your average consumer, I don't know whether this matters. But you're right. If you're looking at something that's-- we have bright lights in the scene. At a distance, take our false photo. They will all look like this.

AUDIENCE: Or you could put hearts in it, or, like--

AUDIENCE: Right, yeah, I was thinking maybe--

AUDIENCE: I mean, that's totally possibly.

[LAUGHTER]

RAMESH RASKAR: So an interesting art problem is how do you create-- how do you create a mask that visually looks aesthetic but is mathematically also invertible.

AUDIENCE: Yeah.

RAMESH RASKAR: Are there disadvantages? Or challenges? Not really disadvantage. Remember, in the motion case, we had to know how much the motion is. What do we need to know here?

AUDIENCE: We know how much the blur is.

RAMESH RASKAR: How much the blur is. And what is that function of? If anything, plane of focus is sharp. When it's out of plane of focus, it's blurred. But the size of the blur is dependent on what?

AUDIENCE: Belt.

RAMESH RASKAR: The belt. But not just depth-- depth from the plane of focus, right? So that's an extra parameter you would estimate somehow. Maybe you can use a rangefinder or something like that, or just a software. There are methods you can employ.

AUDIENCE: Don't you just try to assume something like we've got to see this contrast?

RAMESH RASKAR: Yeah.

AUDIENCE: Yeah.

RAMESH RASKAR: You could do that. It doesn't work that well. but you're right. That would be another way, too, to try this.

AUDIENCE: You can just maximize your hard edges in the image.

RAMESH RASKAR: Exactly. That's what you would do, like, in a light field, when we did the refocusing. That's the trick we used. We said, OK, let me try to refocus. I don't care about the depth. When it comes into sharp focus, my edges, that must be the right depth.

Unfortunately, it doesn't work out in this case. And we won't go into the detail, but the main reason is that, because it's coded aperture, no matter where you refocus, it still looks like it has very high frequencies. So that makes it challenging. Yes.

AUDIENCE: How did you come up with the pattern?

RAMESH RASKAR: Oh, exactly. So you need to find this 7-by-7 pattern or even the previous case, the 52 pattern. And you take a random sequence. Take a Fourier transform to see if it's flat. If it's not flat, you go to the next one.

AUDIENCE: Oh, so this is brute force? There's not, like, a pretty mathematical formula for this?

RAMESH RASKAR: So initial, that's what I did. I said, wow, it can't be that bad. 2 to the 50-- I mean, it's 52-element long. And I know some of them. I only want to take the ones in which about half of them are--

AUDIENCE: 1s.

RAMESH RASKAR: --1s and half of them are 0. So it can't be that bad. So I wrote a MATLAB script. And I said, by the time I come tomorrow morning, I'll find a really good code. And I came back next morning. Nothing had happened. I waited all day. It was still running. And it never came out of that. So 2 the 52 is pretty challenging.

AUDIENCE: Yes. Where's your [? canu ?] cluster? We need it.

RAMESH RASKAR: Yeah, so-- sorry?

AUDIENCE: Where's your [? canu ?] cluster? We need it.

RAMESH RASKAR: Exactly. But even if you use a cluster, it's still a pretty big number. So you can do some approximation. So you can start with some code and do a gradient descent and so on. Yeah.

AUDIENCE: Does the [? harder ?] [? mark ?] code or anything? Is that applicable here?

RAMESH RASKAR: Mhm. So actually, after we did these two projects, I attended Professor Han's lecture on computational imaging, which I highly recommend, by the way. It's terrific. And there are all these theories about how to create different codes for different applications. So [? harder ?] [? mark ?] code, which we learned about a few weeks ago or so-called broadband codes, they all have polynomial solutions and this and that.

There's no good solutions for 2D. But for 1D, there are some really good solutions to come up with that. And even for 2D, for certain dimensions, they call it one more 4 or three more 4 because prime numbers can be one more 4. Basically, when you divide by 4, the remainder can be 1 or 3. And there are certain sequences that are beautiful mathematical properties, of which sequences could have broadband properties and which may not. So it turns out you cannot-- there's a little bit of cheating going on here.

So you cannot really use the broadband code here either to give you the best result. You can call them broadband because their behavior is broadband. But the traditional code's called MURA code, M-U-R-A, Multiple Uniform Redundant Array. They invented not very long ago, maybe 20, 30 years ago. And they used in CDMA and many other astronomical-imaging applications. And they have similar properties of being-- if you take a circle to transform, it's broadband.

The problem is, in many of those example-- many of those applications, your convolution is actually circular. So you apply the filter, and then when you go off the edge, you apply the filter to the beginning of the signal. This particular filter is actually not circular, but it's linear. So when you apply the filter here, when you start applying the filter at the end of the image, you don't go back to the front of the image because, clearly, if I put an LED here, you get out of focus. If I put an LED here, you'll only get half of that. The rest of the half is just blocked. It's not going to magically appear over here. So that's the difference between linear convolution and circular convolution.

It turns out, for circular convolution, the match is very clean and beautiful and smoother course work. Or for linear convolution, there is no good mechanism. So we came up with our own code called RAT code, R-A-T, which is after three quarters. Oscar [INAUDIBLE].

AUDIENCE: So how did you find that code?

RAMESH RASKAR: By doing research.

AUDIENCE: Just doing research?

RAMESH RASKAR: Yeah.

AUDIENCE: OK.

RAMESH RASKAR: But it's not a brute-force search.

AUDIENCE: Yeah. It was an intelligent.

AUDIENCE: And if you included enough padding there, wouldn't you be able to use circular convolution?

RAMESH RASKAR: Yeah, I mean, circular convolution-- I mean the linear convolution is basically circular convolution with a lot of padding of 0s.

AUDIENCE: Yeah, because you said then the math would be easier, right?

RAMESH RASKAR: But then it's too large. I mean, finding a code that's 7 long or maybe 30 long is OK. Finding a code that's 1,000 long is nearly impossible.

AUDIENCE: So the difference between MULA and that is only on the edges? Or is it all over the picture?

RAMESH RASKAR: It's only the fact that one is linear convolution and one is circular convolution.

AUDIENCE: OK.

AUDIENCE: Yeah, and another thing is it's pretty amazing that [INAUDIBLE] because if you start just having very simple patterns on a square-- like, say, if you just draw this square and sat down this square, you get the free entrance form, and you have all--

RAMESH RASKAR: Yeah, all over the place. So, yeah, so it seems like can just choose a random sequence and get a similar property. But actually, it doesn't work. The chances of a random sequence doing the right thing for you is very, very low.

AUDIENCE: Instead of [INAUDIBLE].

[LAUGHTER]

AUDIENCE: Are astronomy people are already using--

RAMESH RASKAR: Mhm, yeah.

AUDIENCE: Or they were using this for [INAUDIBLE]?

RAMESH RASKAR: So in astronomy, you have circular convolution because they use either two mirror tiles and one sensor or one mirror tile and two sensors. So the whole circular convolution. So all right.

AUDIENCE: If you're tired of [INAUDIBLE] astronomically coded imagery.

RAMESH RASKAR: Repeat that.

AUDIENCE: If you're tiling the mask at aperture, but you are using single-tiled aperture--

RAMESH RASKAR: Right.

AUDIENCE: So if you're tiling that up--

RAMESH RASKAR: If you tile aperture, you'll get really horrible frequency response, unfortunately, because if you put two tiles, that means certain frequencies are lost.

AUDIENCE: [INAUDIBLE] impressive. It's saying that, if I understand this right, basically, by taking the DC coefficient, you're reconstructing almost everything. Is that--

RAMESH RASKAR: No, no, no, not DC coefficient because if you look here, all the high spatial-- I mean, the whole image is not one value.

AUDIENCE: Yeah, but look at that. That's the spectrum of your--

RAMESH RASKAR: Right. No, but there is a non-zero value at other frequencies.

AUDIENCE: Yeah, yeah, a few. But--

RAMESH RASKAR: No, no, that's very important.

AUDIENCE: Yeah, but by taking that, you could get a very good approximation.

RAMESH RASKAR: Yeah. But if-- to a naive consumer, this photo-- so look at this part, OK? This photo and this photo looks almost the same, right? And remember, in this photo, many of those frequencies are lost. And in this photo, those frequencies are not lost because all the frequencies are preserved.

But that's because our eyes are not very good at thinking about what the original image could be, given either this one or the previous one. So given this, I can challenge you that you're not able to predict that it has all this structure, right? From here, you cannot predict that you have the structure.

AUDIENCE: So how would you describe the mask as? Basically, you spread the energy in [INAUDIBLE] set of over many frequencies but very small coefficients. Is that--

RAMESH RASKAR: Exactly. It's about-- depending on the code, it's about 1/10 or 1/20 of the original power of that frequency. So you get significant attenuation. So the results are not perfect. If you look here, right, it's not it's not perfect results, whether it's here or here. Look at this. I wouldn't call it photography quality yet.

AUDIENCE: Yeah, no.

RAMESH RASKAR: But if you apply very simple-- but it is a raw resource. There is no medium filtering or smoothing or anything. It's just pure x equals b, x equals a backslash b.

AUDIENCE: Just the fact that the mask, I guess, gives you balance.

RAMESH RASKAR: Yeah, it's fun. What's amazing about coded imaging is that the math is elegant and beautiful and sometimes complicated, but the implementation is very easy. At the end, all I had to do is put this code or shutter it, and very easy to explain. My previous boss is to say, the best ideas are the ones that are easy to explain but difficult to conceive.

All right, so let's move on. OK, let me finish this one. So there's just one way of-- we only saw two ways of engineering the point spread function, one in motion and one in focus, right? But there are many others. We saw some of them over the course of the semester, where you can put, for example, a special filter in the lens so that you get blur that's independent of that.

AUDIENCE: Ramesh, let me ask you one more question.

RAMESH RASKAR: Yes, go ahead.

AUDIENCE: You had this binary mask, right?

RAMESH RASKAR: Mhm.

AUDIENCE: What if the mask was not quite? If you have some information by the board so that you could set up approximate [INAUDIBLE].

RAMESH RASKAR: Right.

AUDIENCE: So what would you have?

RAMESH RASKAR: So that's a very good question. So-- let's see. Let me get this out first. So if the function was-- if the function was continuous-- so in case of flutter shutter, we didn't have much of a choice. It's either opaque or transparent. It's one or the other.

AUDIENCE: Yeah, yeah.

RAMESH RASKAR: But in case of aperture, yes. It doesn't have to be opaque or transparent. It could be a continuous value. And initially, actually, I and my co-author, Amit Agrawal-- very smart guy-- we always had these arguments about maybe continuous is better. Maybe binary is better. And he continued to believe that continuous is better. But it turns out-- and we still don't agree with this, by the way. And nobody has written this down. It turns out that, for any continuous code, there is a corresponding binary code that will do an equally good job, so far.

And that's because in a binary code, you get to play with the phase function. I won't go to the detail. But because here, we are only showing you the amplitude of the Fourier transform but not the face. So you get that extra degree of freedom to play with. So if you play with the right phase, then it turns out you can always have a binary function. Mike?

AUDIENCE: Has anyone tried to combine the coded aperture and the coded [INAUDIBLE]?

RAMESH RASKAR: That's a great idea. People talk about it, but nobody has done it. It's just one of those things. It's just one of those things. It's like we are sick of it, so we don't want to do it. But I think it's worth trying. And because those are orthogonal motion blur. So here's a great thought experiment. So Mike's question was, there could be something that's moving, so it's motion blurred, but it's also out of focus. OK? Can you use both at the same time and record?

AUDIENCE: Yes.

AUDIENCE: What about the light width?

RAMESH RASKAR: Yeah, it's one fourth of the light, but let's not worry about that. OK.

[LAUGHTER]

Explain.

AUDIENCE: There are orthogonal technologies, basically.

RAMESH RASKAR: Exactly. So it's amazing because motion is time, and the focus is space. They're completely orthogonal. So you can play with it. It's very interesting.

AUDIENCE: But still, motion is being represented by space on the--

RAMESH RASKAR: Yeah, eventually, it's going to have a 2D projection.

AUDIENCE: Yeah.

RAMESH RASKAR: So that's very interesting. All right? So the point spread function, although I and my team were the first one to do that in a graphics vision domain, people have been trying to do that since mid '90s in imaging. And there was a very classic paper by Cathey and Dowski and others for so-called wavefront coding. And a lot of it is actually being used in cell phone cameras.

And what they do is they put this face mask between the object-- near the lens so that-- and we saw this in the beginning of the class-- so that the image does not come into sharp focus ever. Instead of that, it's like a set of straws. Imagine these are all straws that are coming in. And you just twist them. So the top one kind of goes at the top-- I'm sorry, at the bottom. The bottom one goes at the top. And when you think about the cross-section of all the straws, it's kind of cylindrical, when they all come together.

OK, I'm going to take all these straws, or maybe strings, if you want to think about it. And I'm going to twist them so that they remain cylindrical. So if I put my sensor here, if the image is out of focus by this width, if I put a sensor here, it's still out of focus but by the same width. So no matter where you are, the image is out of focus but by the same amount.

And you say, well, what's good about that? It's always out of focus. But turns out, the wavefront coding, as they call it, but you can think of this now we know what light field. So this just a unique light field of the scene. It turns out that from that, you can recover images. Like, so this is open aperture. I'm sorry, I don't have a picture. But we discussed it in the class, so I hope you remember that. I missed that picture.

We saw this right in the very first class, by the way. And the benefit of that, it turns out, is that it preserves the spatial frequencies, and it has the benefit that, no matter which steps you are at, you have the same defocus blur. So the disadvantage of coded aperture was that you need to know what the depth was to be able to deblur. But now, because it's independent of depth, you can just apply the same deconvolution and get back a sharper image.

So whether if I hold myself on camera, whether I'm here or here or at infinity, I get the same amount of blur. Same point spread function. And from that, I can deconvolute and get an extended depth of field that goes from very close to the lens to infinity. So OmniVision, which bought this company, cerium optics, which is named after Cathey, Dowski, and somebody-- those are the two professors at Colorado. And the last one, I forget.

That was just bought by OmniVision, which is a big cell phone-- I mean, big imaging company. Most of the business is cell phones. And they acquired the company and immediately laid off all the smart people who invented this. It's very sad Because that part is done. So they just wanted the technology. And it's in a lot of cameras.

There's another company called Tessera, which has a very similar solution. But what they do is-- this one, basically what it does-- and we discussed this, I think, in the beginning, the wavefront coding-- is they are simply placing an addition here so that this part of the lens will focus on an image here. This part of the lens will focus on this one. This one focuses here.

The top of the lens has a short focus lens. It focuses here. The second one focuses here. Third one focuses here. Fourth one focuses here. Fifth one focuses here. Here. And here. OK, so if you can imagine the main lens has a certain focal length. And we're just going to add a little bit of additional focal length, which is-- that's why you have focal length F1, F2, F10. And then [INAUDIBLE]. And this is the twist that I was talking about. This was [INAUDIBLE].

But within this region, the thickness will be a bonus. So you can either think of it as adding small matchsticks on top of the main lens-- or the way they do it is they actually put one single sheet that looks like that, an additional layer of support, a face mask. And a face mask basically means you are changing the face of incoming light.

And, as you know, if you have a piece of glass, and light is going through, it's going to slow down here and then again [INAUDIBLE]. That means you basically slowed down the light. And that's where the glass [INAUDIBLE]. If you have it at the top of the lens, like those two, it doesn't slow down that much. If you go to the middle of it, it slows down forever.

That's why, as we learned about at the beginning, if you have something very far away, this slows down a little bit. So those go over here. This goes over here. And everything just works out with operations.

But [INAUDIBLE] this extra piece of glass, you're saying, I'm going to speed up and slow down in a slightly different way [INAUDIBLE]. This is the Syrian optic solution or the [INAUDIBLE], which is actually bought as another company. [? Australia-- ?] forgetting the name. The solution is very similar. I'm sure they're fighting out in court right now.

Same solution. Instead of putting this particular guy, that's just going to add some extra glass, but mostly in a minor form. It's just [INAUDIBLE] on that one. So basically the same solution but creating different focal length for different [? partners. ?]

AUDIENCE: Yeah. Although you said, I mean, there's this portion there, where if you have another blur [INAUDIBLE], right?

RAMESH RASKAR: Right.

AUDIENCE: But what is being blurred? At each piece, or where is?

RAMESH RASKAR: Independent of the depth, you get the same blur.

AUDIENCE: Yeah, but see, some guys are focusing, say--

RAMESH RASKAR: At an angle? It doesn't really matter. It doesn't really matter because, just like in a traditional camera, even if the point is not on axis but off axis, you still get the same-- you'll still get a disc, right, which we saw in the--

AUDIENCE: Yeah, you get the deuce, but I think that the given picture, as you just move it back and forth, you're going to get a different color, as you-- I mean, a different amount of the mixture of--

RAMESH RASKAR: Different shape, you mean, or different color?

AUDIENCE: Different color.

RAMESH RASKAR: Not really.

AUDIENCE: Because look at that. If the guy was coming from the top, it's going to reach at some point.

RAMESH RASKAR: But they all have the same-- I mean, you're saying, because of chromatic aberration?

AUDIENCE: No, just because of the geometry, at least it seems to me.

RAMESH RASKAR: Using color or shape? Because just to be clear that we are not adding any color here. We're just adding one glass.

AUDIENCE: OK. OK, OK.

RAMESH RASKAR: We're just adding one glass. So we're painting the rays, but the colors are, for all practical purposes, that'd be the same.

AUDIENCE: Yeah, what I find is that maybe some points in the scene would be-- made sure even a piece of--

RAMESH RASKAR: Yeah, the effect is very low, though, remember. The effect is extremely low. So maybe you have a pixel and get blurred by 10 pixels or [INAUDIBLE]. It's not a global effect. So this picture, maybe-- this particular diagram is misleading because it seems like this point is going to go all the way. But this is very narrow. And the blur is only about 10 pixels, no matter where you [INAUDIBLE]. So maybe that was the matter.

So if you have a point of access, it's still going to create an image that's blurred 10 pixels. So this is, again, very counterintuitive, where you go to make the image intentionally blurred. It's just that it's blurred everywhere. And then we also saw this one very early on, where the point spread function-- typically when something goes in and out of focus, it looks like a point. And then when it goes out of focus, it looks like a disc. If it goes out of focus other ways, it still looks like a disc.

But this group at, again, at Colorado have-- when it's a sharp focus, you see two doors for similarity. And if you go in and out of focus, then the two dots [INAUDIBLE]. So they call it rotating point spread function.

AUDIENCE: Is it the same group that developed the [? framework? ?]

RAMESH RASKAR: It's not the same group, but same university and the same neighborhood.

AUDIENCE: What was the reasoning for developing the rotating point spread function?

RAMESH RASKAR: Doug's question is, what's the benefit of this?

AUDIENCE: Does [INAUDIBLE]?

[LAUGHTER]

RAMESH RASKAR: She would have used it by now.

AUDIENCE: Yeah.

RAMESH RASKAR: What's the benefit of the strange point spread function?

AUDIENCE: You know if you're out of focus in which direction.

AUDIENCE: Yeah.

RAMESH RASKAR: Yeah?

AUDIENCE: From the focal point.

RAMESH RASKAR: That's one.

AUDIENCE: And you know your--

RAMESH RASKAR: But do you know by how much?

AUDIENCE: Yeah, you know by how much because--

RAMESH RASKAR: Because it's an angle.

AUDIENCE: --rotation.

RAMESH RASKAR: So the goal here was no matter where you are, your point spread function is the same. The goal here is exactly opposite. If you go slightly out of focus, you get a very different point spread function. So this one they use in microscopy with fluorescent dye. So when you're looking with a microscope, depending on what the depth of your tagged particle is, the point spread function will look very different. So you can estimate the depth by looking at the orientation of those two dots. So that's very interesting.

AUDIENCE: But can't that guy keep going all the way in? At some point, you can't--

RAMESH RASKAR: No, it doesn't work. After some point, they'll stay the same.

AUDIENCE: OK.

RAMESH RASKAR: This is only in the [? sweet ?] region.

AUDIENCE: So have they been able to reconstruct three-dimensional neuronal structures, or--

RAMESH RASKAR: Yeah, that's why they're getting a lot of press. And they're doing some amazing work, [INAUDIBLE]. So they have a lot of collaborations, and now they're able to measure the z-dimension down to about 10 nanometers.

AUDIENCE: Wow.

RAMESH RASKAR: The xy still remains traditional microscope 1 micron, 1/2 micron. But the z-dimension is 10 nanometers. It's very new. They are still working on a lot of these concepts. OK?

So let's very briefly look at compressed sensing because it's something you should be familiar with. OK, so here's an idea that received a lot of publicity. It was even "The 10 Emerging Technologies" by a very reputable magazine. I hope you don't believe any of those things. It's a very cool idea, by the way. And as a scientist, I really like it. But when somebody like *Technology Review* or *Wired Magazine* says, Top 50, Top 10, of course, I wish I'm listed among them. But at the same time-- because, you know, it has good side effects.

Well, anyway, this single-pixel camera was listed as one of the big things in 2005 by*Technology Review,* which a magazine I really like, by the way. And the idea is, instead of taking one single photo, what you're going to do is-- let's say that's your scene. You're go to turn on-- you go to take a single photodetector and aim it at a set of micrometers. And in the simplest case, what you will do is you turn off all the micrometers, that light goes this way. And there's only one micrometer, like this one.

So a single photodetector-- this is like the dual photography we saw right at the beginning, where you can see that card. If I just turn on this one micrometer-- by the way, this is what's in your DLP projectors, the Texas Instruments Digital Light Processing Micrometer Displays. So it's very easily available. | just receive light from the scene for that one pixel. So this scene is being imaged on this little array.

And you just want to turn on this one pixel. And then the next picture, you're going to turn on the next pixel, and so on. And one at a time, if you go through this million pixels, you will get a million megapixel image, right? But of course, the light will be very little if you just turn on one pixel. So now, we'll do some [INAUDIBLE] multiplexing, which we saw a few classes ago, where you go to turn on over half of them, take one reading, turn some other random combination of half of them, and take a picture, and so on.

And, again, after-- now about half of them are contributing to the photodiodes. So the photodiode is very well exposed, and you can take a very short exposure reading. And, again, if you take million such readings, you can recover this picture. That's the concept.

AUDIENCE: Does it exponentially increase, the number of readings you have to take?

RAMESH RASKAR: No, just linearly. If you're on 2 megapixels, then you need to take 2 million [? pics. ?] All right? So the claim this group made at Rice University was that if I wanted a million-pixel image, I don't have to really take a million readings. I can do much fewer than a million readings. And the claim is that imagine if you had this photo as a JPEG. In a composite, it might take up only about tens of thousands of bytes. So let's say it takes up 10,000 bytes.

So if I can represent the image with 10,000 bytes, and I'm going to take a photo and compress it down to 10,000 bytes, I can't just directly measure only 10,000 values in the scene so that I save on everything, OK? So I can take this picture effectively with just 10,000 pixels but recreate a million-pixel image. And that's where the concept of compressive sensing or compressed imaging comes up.

You want to take something that is much higher resolution but recover it in a compressed way, where it's taking the picture with a hardware and compressing the software. You're going to compress it while sensing. So how does it look mathematically? So let's see. Let's see if there's an easy way to explain this in a shorter time.

So that's the trick we're going to do. We're going to take about half the pixel and measure the intensity and so on. So those are our measurements. So our unknown image is x. And we're going to take a lot of these projections. This is the [INAUDIBLE] matrix, for those of you familiar. And these are our measurements. So we're going to say, given these measurements, I'm going to recover my original image.

Now, when you think about a natural image, the claim is that if you just use DCT, some photo coefficients, then you can compress the image and represent them with very few bytes, only 10,000 bytes for a megapixel. So let's say your Fourier coefficients are here. And this is your image. That means that if I just put a Fourier transform here, then I can convert the coefficients into the image.

And the number of values required to represent an image are much fewer than the million values required here. So we have a million values here, but only about 10,000 values here. And the claim is that by using this understanding that my image can be represented in some transform basis-- in this case, Fourier basis-- using very few coefficients, can be exploited while I'm sensing.

OK, this is your optics. This is your map. See if I have it on slide 5. So that's the theory of compressive sensing, that, using some basis, I can transform the image and measure in [? your ?] measurements. And there are certain cases where it is really true. You have signals that can be compressed very easily.

A very classic example is in communication, where, if you are doing software radio, where you have a huge band of frequencies, and software radio-- instead of tuning it with electromagnetics, you just capture the whole signal. And then software, you can listen to any station. And the necklace theory says, if your band is, I don't know, 100 megahertz, then you must capture it with a signal that has a bandwidth of 100 milliamps.

But we know that in communication, not all bands are actually occupied. Many of the bands are empty. Only certain frequencies have a signal. So people have come up with very clever mechanisms, where they realize that you don't have to capture a 100-megahertz signal. Only some of them are actually on. I'm getting through the Fourier transform because in communication, that's natural.

And by doing that, they're able to sample this effect of a software radio with a detector that doesn't have to measure 100-megahertz-wide signal. It turns out for images, this doesn't work. And that's because there is no transform that allows you-- no linear transform that allows you to represent an image with very few coefficients. When you do JPEG, it does frequency transform. But after that, it does a lot of other things.

It says, perceptually, the higher frequencies are not as important, so I'm going to represent them with fewer quantization grids. Or certain values are too small. I'm just going to truncate them. So all this operation-- changing quantization bands, truncating, or thresholding, are all nonlinear operations. They are not linear operations. So it turns out there is no transform that allows you to represent an image with fewer coefficients. So in general, this scheme doesn't work.

But you will continue to see people who come to you and say, you know, I have this magical thing I just heard or compressive image something, and that will just solve a problem. There are certain images, like cartoons, that can be represented with very few samples because they have flat regions, sharp boundaries, and fluctuations. But a natural image, unfortunately, cannot be transformed that easily. And you can talk to Rohit, and he'll tell you all the details of the dangers of [? compositions. ?]

AUDIENCE: So the single-pixel camera is just a hypothesis but not--

RAMESH RASKAR: Yeah, but at the same time, it was the first one that kind of allowed people to visualize or kind of conceptualize in their mind what compressive sensing might do.

AUDIENCE: This idea is cool, but how feasible or how important it is to have a single sensor rather than having wide arrows sensing? So what this is achieving is basically allowing you to build a camera with a single sensor. But do we really want it just to do compressed sensing?

RAMESH RASKAR: From a scientific point of view, if somebody can build this and show that you can take fewer measurements and recover the image, that's a breakthrough. How do you use it? I agree with you that, in terms of practical implementation, maybe this is the best application, maybe it's not, and so on. But that's kind of a business reason.

AUDIENCE: Will this be faster than an array of sensors?

RAMESH RASKAR: Again, in terms of in practice, both of you are right. There are very few benefits. But if you just do compressive sensing, you realize it's a very, very active field.

AUDIENCE: So again, maybe a different type of sensing, but what are the features-- like, what are the people doing in computational photography for feature extraction in the same way that the brain processes certain features of linear [INAUDIBLE] to do a better compress sensing of context and imaging?

RAMESH RASKAR: You mean compressing an image or sensing with fewer samples?

AUDIENCE: Sensing with fewer samples.

RAMESH RASKAR: Yeah, so that all kind of gets clubbed into this concept of compressive sensing. If you think about a B1 and B2 and visual processing, there's a lot of work that has been done over the last 30 years. There's good work at CSAIL as well. But that's purely software.

AUDIENCE: Right.

RAMESH RASKAR: And maybe you're asking, can we use sensing mechanisms that are similar to our brain so that we don't--

AUDIENCE: You don't to do any software.

RAMESH RASKAR: Exactly. The secret of success for film, of film photography, is that if somebody had given you this problem before the invention of film, that there is a scene-- and I want to give you a sensation of the same scene-- time shifted or space shifted. There's so many ways you can solve that problem. You can start with a reproduction of a photo, or you can tap into retina. You can tap into V1, V2. You can interface that at any point in the pipeline for human vision.

But the simplest solution is to just create that photo on a passive surface and let the brain do that processing all over again. And so it's like a simple impedance match. If I can see the scene and understand it, I can just present that as is and let it go through. And this is how we have been treating photography all this time. It's a record of visual experience, which is great for humans, but it's not so great for computers because computers don't understand any of that.

And what you're saying is, what computers care about are all these high-level features. And that's why we're going back to the drawing board and saying, let's build cameras that are not mimicking human eye but actually extracting more information, like [? apertures ?] that we remove the flash camera, or additional information with light-field cameras or multi-spectral cameras and so on.

AUDIENCE: So what kind of-- so Brett was asking, why would you want to do [? precisely? ?] When do you have to reduce the number of measurements [INAUDIBLE]? And I think one of the problem [INAUDIBLE]. I don't know. The debate about whether it's really better or not is photography? [INAUDIBLE]

RAMESH RASKAR: Tomography, yeah.

AUDIENCE: Yeah, when you have to recover the sky, you want to take as few measurements as possible. So if you can reduce that [INAUDIBLE], That's one of the [? implications ?] of that.

RAMESH RASKAR: Right.

AUDIENCE: This one piece of camera-- really, it does that.

RAMESH RASKAR: But the benefit of tomography, which we studied in the last couple of lectures, is it's a very high-dimensional signal. And so usually, in a high-dimensional signal, there's lower sparsity. There are only a few places. If you think about taking a CAT scan of your body, there are only like four or five types of materials. There is muscle. There is blood. Whatever. There are only five or six things. It's like a cartoon.

AUDIENCE: Exactly. I find this is interesting because the test--

RAMESH RASKAR: It's a 3D cartoon.

AUDIENCE: But if you look at it, it looks just like a cartoon does-- some whites clothes, some black clothes, some--

RAMESH RASKAR: Exactly. And that's why compressive sensing works very well there.

AUDIENCE: So is compressive something used in anything commercially currently?

RAMESH RASKAR: A lot of people are getting grants.

AUDIENCE: Oh.

[LAUGHTER]

RAMESH RASKAR: Is that a commercial-enough reason?

AUDIENCE: No. But also--

RAMESH RASKAR: If you put those two words, your chances improve by 50%.

AUDIENCE: I was thinking of this interesting album that, I guess, extends to all you guys are talking about. So compressive sensing allows you to take less measurements. But the problem is you need to actually have more information about the scene before you take the measurement, which is another measurement.

RAMESH RASKAR: So actually, to clarify, the measurements are done in a non-adaptive manner. So you don't have to know anything about the scene to do this measurements. That's actually one power of--

AUDIENCE: I mean, if you want it to actually succeed--

RAMESH RASKAR: But when you reconstruct, you have to know something about the scene. You have to know in which transform basis is actually sparse. So is it sparse when you take a Fourier transform? Is it sparse when you take a [INAUDIBLE] transform? Is it sparse when you take gradients? Like in terms of cartoons, it's just gradients. So you have to know that when you do reconstruction.

But advantages are, at the time of capture, I just use this random basis or Fourier basis-- I mean, kind of a modified [INAUDIBLE] basis. I can just go ahead and spec sample it. And in software and reconstruction, I don't worry about some prior information about the scene, which is great.

AUDIENCE: Well, I think in the case where you're just taking a set style of captures, that you're limiting yourself in what kind of scenes will be compatible with that capture. So for example, if I just had a scene that's all white, then just one captured would be enough.

RAMESH RASKAR: Yeah, but that's because you know something about the scene.

AUDIENCE: Exactly. Exactly. So--

RAMESH RASKAR: But if you have this situation where you don't know anything about the scene, you'll just use the same exact procedure, for example.

AUDIENCE: Right, but you don't, then you lose the benefit of taking less pictures.

RAMESH RASKAR: No, the claim is that even if you don't know anything about the scene, you take very few measurements. All you know about the scene is that once you take its transform, some transform, it's very sparse. It can be represented in a complex place.

AUDIENCE: So I remember, actually, a mathematical mapping for this, where we're reducing dynamically the number of captures you have to take while you're capturing it.

RAMESH RASKAR: But that's adaptive measure-- adaptive measure.

AUDIENCE: Yes.

RAMESH RASKAR: Because once you take a picture, you say, let me see what I did not capture. So let me take the next one next. That's a very different problem.

AUDIENCE: Yeah. Well, anyways, the code for it's inside the dual photography thing.

RAMESH RASKAR: Yeah, somebody did dual photography with compressive sensing.

AUDIENCE: And it's adaptive.

RAMESH RASKAR: And it works very well because, again, it's a high-dimensional signal, 2D camera, 2D projector. It's four dimensional, but what you're trying to recover is two dimensional. So it works again. So tomography is the same. It's 4D capture for 3D representation.

OK, so I'm sorry we're not taking a break. Should we take a 30-second break before we move on to two very small topics, which is how to write a paper and wishlist for photography.