MITOCW | mitmas_531f09_lec06_2.mp4

PROFESSOR: Yeah, and again, just go slow, because we just talked about this [INAUDIBLE].

MATT HIRSCH: So this is a project that-- this presentation actually came from SIGGRAPH this August. I'm going to skip around a little bit.

PROFESSOR: You might want to say just a couple of sentences about-- Mike had a question about, how did class projects start in this class, and how did they evolve? And I said you'd be a good example of how it started. So just say a couple of sentences about how you started--

MATT HIRSCH: Shall I start with that?

PROFESSOR: Yeah.

MATT HIRSCH: OK. So I guess I had taken Ramesh's class in the spring term before your first class here and had been really interested in camera work. And so Ramesh and I had been thinking about projects after that class. And over the summer, I kind of tried to develop some ideas that might lead to a thesis. And so I guess coming into this class, when I took it, I had the intention to hopefully develop some project that I did in here into some thesis work. And I guess that is a good place to start, maybe, if you're interested in really developing a project beyond the scope of the class, is to have some other good motivation to do that, like a thesis.

But yeah, then when I did the final project for this class, I started thinking about some of the work that Ramesh had presented and working with a post-do-- or a [INAUDIBLE] who's soon to be graduating, who has done some work with Ramesh back at [INAUDIBLE], and really got excited about it and so ended up developing that into my thesis. And that's where this project came from. So--

PROFESSOR: It's a great example. As you'll see, it's a great example of beautiful theory, beautiful implementation, and impactful implementation. So those three [INAUDIBLE] that we have-- novelty, execution, and impact, are [INAUDIBLE]. It's surprising he didn't win the best [INAUDIBLE] award., even by a popular vote [INAUDIBLE]. That's the perfect award.

MATT HIRSCH: Well, we had some really cool-- have you shown them the projects from last--

PROFESSOR: Yeah, just very briefly. But [INAUDIBLE], we should talk over them more.

MATT HIRSCH: Yeah, so I guess I'll just get started with this. The goal here is to think about new ways of interacting with a thin-screen device. And imagine if your screen could basically not only support on-screen touch interaction, but this type of off-screen hover gesture interaction that we think about here at the Media Lab in a lot of contexts. And so you see that brief example where I'm able to lift up my hand and interact in free space right in front of the screen. And here's another example where I'm manipulating an object, and I select it by touching the screen in traditional mode, but can pull my hand away and rotate it around like Luke Skywalker.

PROFESSOR: Use the laser pointer.

MATT HIRSCH: OK. So this is kind of inspired by three emerging areas in HCI and camera research. One of them is this new class of light-sensitive display. You can mention if you've covered some of this stuff, and I'll skip it.

PROFESSOR: Not how it works, so this is perfect.

MATT HIRSCH: You all know how LCDs function. They're basically a matrix of transistors. There's a couple of companies that are taking that transistor matrix and embedding a single extra transistor that's optically sensitive into the matrix so that the net result is the entire LCD is a large-area optical sensor. And they're using these for touch interactions. So that was one inspiration.

And then the second, of course, is depth cameras, which you may have covered a little bit. There are a couple of different techniques, but the upshot is that these cameras not only produce an RGB image of a scene, but a map of the scene where, for each pixel, you have a measurement of the distance from the camera to some object.

And the third, I guess, is this sort of ubiquitous multitouch display, which has been popularized by Jeff Han here and the CNN wall, and of course the iPhone and a lot of other consumer electronics devices.

PROFESSOR: [INAUDIBLE] other people, that makes a good use right now.

MATT HIRSCH: So we're kind of inspired by the ability to so easily interact or so intuitively interact with information on the screen and thinking about, can we take that one step further? So what if you could combine all of these things, basically, and be able to, because it's an LCD, build it all into a thin package, but also, because it's a depth camera, be able to track hands out in front of the screen? So that's where we started.

The benefit of that, of course, would be that you can bring this depth sensibility to all sorts of consumer electronics type of devices that don't have it or it wouldn't even be possible to think about today, like an iPhone or a laptop. So to give you a brief overview of the results-- I think Ramesh may have covered some of this. This is basically a light field, and you've probably seen--

PROFESSOR: We just covered today.

MATT HIRSCH: Yeah.

PROFESSOR: [INAUDIBLE]

MATT HIRSCH: So this is what we were able to capture. And there, you see a synthetically refocused image. You guys, I think you just did your--

PROFESSOR: That's our synthetic aperture.

MATT HIRSCH: --synthetic aperture project. So this is one application of that, where we're basically taking the set of images captured by the light field, synthetically refocusing them, and extracting a depth. So just to give you some of the-- I guess to think about how to adapt one of these optical touchscreens, you know, it works a lot like a document scanner, where you have this array of pixels without lenses.

An object that's touching that layer of pixels can have a sharp image made because you have a one-to-one correspondence between a point in the scene and a point on your sensor, just by virtue of their being so close together. But of course, when you pull your hand away from that sensor, you no longer have that one-to-one correspondence. Rays can travel from this object to many different pixels, and so you get a blur.

So I guess our approach, then, is to think about a way we can basically bring that one-to-one correspondence back without using any kind of lenses. And so what we do is separate the sensor by a small margin from the display and then display one of these types of masks that Ramesh was just describing to you guys. So in this case, what we're considering is using the LCD as both a display device for the user to see the images, like a typical LCD screen, and also as a device to create one of these masks that Ramesh was describing in order to encode the scene in a way that we can decode in software.

So here's an idea of what that device might look like. You have your LCD screen here, and some distance behind it, you have a sensor layer. And out here, you have objects. And you can actually decode-- the vision is that in this thin device, you could then decode this object, process the imagery, and then re-display it on the screen, maybe in a modified way, or in the case that I'm describing, to interact with the screen gesturally.

And so this is a kind of interesting device. And I like to think about the pinhole analogy, because it's very intuitive. Just looking at that mask, it's not quite clear to me what you get just from an intuitive sense.

But the pinhole makes a very easy case to think about. If you imagine tiling those pinholes all across the aperture or all across the LCD, you get, basically, many tiny cameras covering the screen. And each of those cameras has a slightly different perspective of the scene. And putting those together, that's basically a light field that you're capturing right there.

And the interesting thing in thinking about this is when we-- this is not a normal type of camera that you're used to using to capture pictures of birthday parties or whatever. It's going to produce a pretty strange-looking image because, if you think about how you would image an object out in front of the device, first of all, it produces an orthographic image. There's no perspective here.

Let's say this is one of my pinholes, right? If I want to image something that's off to the left of my device, what I do is I take maybe the pixel on my sensor that's over here for each of these tiled pinhole cameras. And that ray basically goes out into the scene and is projected out in a straight line. So there's no perspective. It's all parallel rays.

And the other interesting thing to think about is, you can see that the resolution actually does decrease. Imagine this blue is the size of my pixel. I can project that pixel out into the scene to see what I'm measuring, and you can see that as an object gets further away, my pixel, relative to the size of the object, is increasing. So just a brief tangent there.

But I'll just show you a couple more ideas of how this might go. You might think about being able to navigate spaces by just moving your hand in free space. And because it's a-- think-- because it's optically sensitive, you can think about-- where is it?

You can think about doing a demo like this, where I'm actually taking a real-world flashlight and projecting it into a virtual scene. So I'm taking real light and mapping the light field that I capture into a virtual world. This is an interesting mixed reality.

PROFESSOR: Is it the end?

MATT HIRSCH: Yeah, it seems a little-- you can see there's a hand here holding a flashlight. And that's actually shining light into this virtual world. So I guess maybe I'm going to skip over this a little bit. But you can think-- there are lots of ways to accomplish this, or accomplish parts of what I'm describing, that don't involve using the method that I'm describing. But if you look at the entire package-- putting it in a thin device, being able to capture both touch and gesture-- I think it becomes a pretty compelling idea.

So I think Ramesh has described a little bit about light fields before. I'm just going to cover the basic ingredients that I've found very helpful to understand the theory behind this. And if I'm covering something that you've already seen, you can stop me. But in a light field-- let's just imagine the 2D case-- when you have a ray, the basic idea is you want to parameterize this ray. You want to describe a set of rays in a new space.

So I have a ray that has some intersection with a sensor plane. And it intersects with an angle. So if I just plot the point where it intersects and the angle at which it intersects in this new space, this is a light field. And you can see, if I have a whole set of rays, that creates a sort of line in the light field space over there.

And then if I have-- oh, well, this is actually important. If I-- oh, that's weird. I imported this from PowerPoint, so not all of these-- OK, well, it's important to note that one of these is what a real sensor measures. If you take a real sensor in the real world and just expose it to light without any lens or anything in front of it, it integrates rays from all directions. And so what you're actually measuring in the light field space is one of these lines.

So this is what a sensor array might look like. Now, if I think about the frequency domain picture of this space, I'll have some light field here that I'm taking projections through, basically. And that light field has some spectrum in the frequency domain. And as I mentioned, these lines are projections through the light field.

So there's something called the Fourier slice theorem, which basically says if I take a projection through a function in this domain, I'm actually taking a slice through my spectrum in the frequency domain. So I guess the important thought trap here is that if I am actually measuring this with my real-world sensor, in the frequency domain, I'm actually measuring only what's along this axis. So this is the only thing accessible to me with a real-world sensor. And of course, what I actually have over there is that whole spectrum.

And so the question is, how can I access that data? So the next important thing to keep in mind when trying to understand this is the skew property of the light field, which basically says, if I'm going to plot a light field, I can either look at it from this perspective or from this perspective. So I can say, as my ray travels through free space, I'm going to just plot the light field that it creates over on the right side there. And you can see a kind of interesting effect where, as I add rays, a straight line from one position in space becomes a skewed line in another position.

And then, as you may recall from physics or signal processing theory, if I perform a convolution function between any array of delta functions and some other arbitrary function, I get a tiled version of my arbitrary function. So I'll tie this all together, I promise. Just try to keep these things in mind as we go.

And then I guess-- I don't know if Ramesh has used the term spatial heterodyning, but I think it's a kind of cool historical note that heterodyning is a word that comes from old radio broadcasts. And when we say-- it was really a technique that multiplied a voice signal by a high-frequency signal in order to transmit it. It'd basically shift that voice up into a radio spectrum that could be broadcast. And this is really what we're doing when we send a ray through a mass. We're actually multiplying that ray by some frequency pattern. Spatial frequency, in this case, instead of time, but similar principle.

And so, to bring this all together now, imagine I create a mask that has a transform that looks like this. It's a series of delta functions. And because I'm multiplying it in the primal domain, in the frequency domain, I'm convolving it. So I'm convolving my mask spectrum with the light field spectrum that I want to measure. And so what you get is a tiled version of that light field spectrum.

And remember that shift property. I've offset my mask from my sensor a little bit. So the mask-- well, the light field that I've created is actually going to be skewed by the time it reaches my sensor.

And so now you see the really cool, insightful part of all of this is that if you look at what's on the fx axis down there, you can see, along this axis, I get different portions of the spectrum. So I've now created a way to measure pieces of that spectrum on my real-world sensor. So I can just rearrange those things and reconstruct a good portion of my light fields back to me.

So the important thing to keep in mind, or one important thing to keep in mind, which I guess goes towards some of the discussion you guys were having before I began speaking, is that you really have to make sure these light field spectral copies are band limited so that they won't interfere with each other. And I guess that speaks to Ramesh's point from earlier.

So in terms of building this actual prototype, it's all fine to do this in theory, but we want to actually do it in practice and see if what we're doing on paper really makes sense. So what we really want is, remember, an LCD separated from a large-area sensor by a small margin. But that's a really difficult thing to get. These things will be out there in the consumer market in the near future, but they're not right now.

So what we actually ended up doing was taking a couple of cameras and simply imaging a diffuser. Much like the movie screen shows you a slice of the light passing through a certain space, a diffuser will just show us optically what we would like to measure electronically in this plane. So here are the actual cameras that we used, and here's one prototype. I have, actually, a slightly newer one now, but the LCD was sitting in this screen, and this is a diffuser.

And then I'll just quickly run through the software pipeline that we wrote. I won't spend too much time dwelling on this, but the basic idea is we want a time multiplex between displaying an image for the user to see and illuminating that image from behind, and then switching to the mask mode where we don't want to illuminate it because we don't want to interfere with our measurement. And we want to actually capture the data that is being modulated by that mask. And so I'll just play a couple of videos of the data from different portions of that pipeline. So here, you see the actual [INAUDIBLE] code that we display on the screen.

PROFESSOR: [INAUDIBLE] this one doesn't look like cosine masks. It's a real binary mask.

MATT HIRSCH: Yeah, this is the binary mask that Ramesh was mentioning. And it actually turns out-- if you recall from my theory description, the only thing I mentioned about the mask was that it has a transform that's a series of deltas. So it turns out, if you tile any code, you can get a transform that ends up being a series of delta functions. Those functions will have different weights, but it's kind of the Fourier series effect, right? And--

PROFESSOR: If you tile anything, you'll get [? deltas. ?]

MATT HIRSCH: Yeah. And I guess the reason that this mask was chosen was that it actually is sort of optimal in terms of light efficiency. So this mask allows 50% of the light to pass through, which is pretty remarkable considering we're reconstructing an image without a lens. Just for comparison, the pinhole allows something like 1% to 2%, depending on the size of the pinhole you used. And I think the cosine mask had 18%?

PROFESSOR: 18%. And there was a question out there about difference with a pinhole and-- in this sense. That's between 1% and 50%.

MATT HIRSCH: Yeah. So the data that our sensor captures, and if you were sitting behind the screen, this is what you would see, basically, if a hand is moving around here and touching the screen, hovering over it. And you can see the high-frequency noise, or it looks like noise, that the mask creates. And from that, we can decode this light field.

PROFESSOR: [INAUDIBLE] were 20 pixels, you think? [INAUDIBLE]

MATT HIRSCH: Yeah, so it's a 20 by 20 light field angularly, and then each little tile there is about 100 by 80 pixels. So you can see many views of that hand moving around, basically. And then from that light field, as I mentioned, for each frame, we get this stack of images, or we can refocus that light field at a number of depth.

PROFESSOR: It's doing the refocusing, which is your first part of assignment, in real time from this 400 images, like 20 by 20?

MATT HIRSCH: Yeah.

PROFESSOR: Yeah.

MATT HIRSCH: Yeah. And then once we have a refocused image, for each-- basically, we have a whole stack of images. We traverse that image and use a method called depth from focus where we basically look at the contrast in each of those images at each pixel. And from that, we get a depth map. And that's the basic ingredient into all of those interactions down there.

PROFESSOR: So that's for extract credit, the depth from focusing your [? SIM. ?]

MATT HIRSCH: So I had some videos in there, but they don't work in there.

PROFESSOR: Any questions?

MATT HIRSCH: So yeah, I think that's probably good [INAUDIBLE], right?

PROFESSOR: Yeah, that's what I thought. Yeah. Yes?

AUDIENCE: Are you doing this in real time? Because [INAUDIBLE] complication [INAUDIBLE].

MATT HIRSCH: Well, no, luckily, computers are very fast. And actually, there's a couple of even free Fourier transform libraries. One is called FFTW. And it optimizes itself to your hardware, and it can run very fast.

AUDIENCE: Was this made by-- was it in MATLAB, or--

MATT HIRSCH: No, no, this was written in C. But it--

PROFESSOR: It's running in real time. You can interact with it in real time.

MATT HIRSCH: Close to real time.

AUDIENCE: Six frames in a second. [INAUDIBLE].

MATT HIRSCH: So the demo runs at about 20 frames per second.

PROFESSOR: It's not 60 frames per second.

MATT HIRSCH: Yeah, we're hoping to improve that.

AUDIENCE: How about latency from hand movement to actual movement?

MATT HIRSCH: It's one frame or so.

AUDIENCE: We're on a different system.

MATT HIRSCH: Yeah, computers are very fast. The key is really to pick a Fourier transform that can be broken down into small prime factors, because you can implement that very quickly. But as long as you do that, it'll run real fast. Kevin?

AUDIENCE: What's the slowest part of [INAUDIBLE]?

MATT HIRSCH: The slowest parts are the parts that I had to write. I mean, there are a lot tools. Like, OpenCV is a great tool for working with graphics in real time. There are Fourier transform libraries.

Things that are slow on a modern computer are memory accesses. And so I end up having big sets of data that I have to remap. For example, you can measure a 2D data from the camera, right? But then you have to work with 4D data, and you have to remap it in a way that your Fourier transform library can understand it. And so that remapping is actually one of the longest steps. It's just simply reshuffling things.

And I guess one of the most challenging practical problems here is literally just synchronizing everything. While computers are very fast, they're also not very reliable in terms of timing. So things can just happen whenever they happen. And especially with rendering things on a video card and trying to understand exactly when they're going to show up on a monitor, there are a lot of different and variable delays in that that are difficult to account for. So that's something I'm still working on.

PROFESSOR: That's the kind of final project we want to see.

AUDIENCE: [INAUDIBLE] question. Why is the diffuser a key part of this device?

MATT HIRSCH: The diffuser is just our stand-in sensor.

PROFESSOR: Yeah. I mean, this is the key part, right? The camera of the future will not look like anything like cameras today. Your LCD screen, a 15-inch screen, is actually your camera in the future. It's just that right now we don't have it.

AUDIENCE: So you're using this as your sensor and then actually imaging the thing and then working from the image that you get.

PROFESSOR: It's a shortcut for now.

AUDIENCE: OK, I see. Yeah.

PROFESSOR: But in the future, the whole thing will be a camera. So if somebody wants to take this concept further, by the way, what will you do when the camera is 15 inch wide, but only when you touch it you get an image? It's like a document scan. You take anything away from it, you just get a blurred thing. So if anyone wants to think about that further, I'll be very interested--

MATT HIRSCH: So they're selling these devices now that-- there's a laptop on sale in Japan that has a trackpad that's made from one of these optical LCDs. So this is a very near-term technology. In a couple of years, it'll be everywhere. So it's something cool to think about using.

AUDIENCE: I have another question. What's the range of this device? Like, if I were to soften [INAUDIBLE].

MATT HIRSCH: Yeah, that's a good question. This is a very big parameter space. There are a lot of variables to change. So the prototype that we built, we optimized for about 50 centimeters in front of the screen. But you can think about basically changing that separation between the screen and the sensor and changing the pixel size of the screen and sensor. And all of those variables will allow you to control the range of depth that you can measure.

PROFESSOR: And before we do it on a device like this and just do however and so on. That [INAUDIBLE] small form factor.

MATT HIRSCH: Yeah, we just need to buy one of those laptops.

PROFESSOR: Yeah.

MATT HIRSCH: Yeah.

PROFESSOR: Then the parameters will be very different, because then you don't expect to interact [INAUDIBLE]. So just to give some context for the class, of course, you did not do this whole thing in real time. You looked at only the capture, is that right?

MATT HIRSCH: Yeah, that's right. And in fact, for the class, I started without an LCD because there are a lot of challenging hardware issues with getting an LCD to work like that. So what we started with was simply a printed mask. And there's a great resource right here in Cambridge called PageWorks, who can print very high-resolution masks. I think that's what you use for your [INAUDIBLE].

PROFESSOR: So he did a static version first, for the class, and did a proof of concept [INAUDIBLE] theory and the static prototype. And then, the two months after that, towards the SIGGRAPH deadline, he did all these things. Unfortunately, the paper was not accepted despite all the great work and all the results. That tells you how high the bar is.

MATT HIRSCH: Luckily, we got to SIGGRAPH agent. So [INAUDIBLE] a school in Louisiana.

AUDIENCE: Then you've got to buy the laptop, right?

MATT HIRSCH: That's right.

PROFESSOR: But it was cool enough that when he presented it at SIGGRAPH in New Orleans, he won the second best paper award [INAUDIBLE].

MATT HIRSCH: It was a poster.

PROFESSOR: [INAUDIBLE]. So now he's a [INAUDIBLE].

MATT HIRSCH: I don't know what that means, but--

PROFESSOR: Eventually, [INAUDIBLE]. Cool, any other questions for Matt? So this is an example of how you think about image formation and image processing in higher dimensions. It opens up a completely new space.

Don't think of camera as something that takes the 3D world and maps to the 2D image and all you can do is fiddle around with pixels. There's a lot more going on if you start thinking about the whole packet. So let's take a short break, and we'll come back and talk about other HCI applications of cameras in general, not just [INAUDIBLE] cameras.