BERTHOLD HORN: So welcome to Machine Vision 6.801, 6.866. And I'm not sure how we got so lucky, but we have the classroom that's the furthest from my office. So I guess I'm going to get a lot of exercise. And I think we're going to have a lot of stragglers coming in.

What's there to know? Just about everything's on the website. So I can probably eliminate a lot of the administrivia. Please make sure that you're actually registered for the course on the website and take a look at the assignments.

And hopefully you've either had a chance to look at chapters 1 and 2, or you're about to. That's the assignment for this week. And there's a homework problem. And you're probably saying, God, I just arrived here. How can there be a homework problem?

Well, I'm sorry. But the term is getting shorter and shorter. And if I work backwards from when the faculty rules say the last assignment can be due, we have to start now.

Now the good news in return is there's no final. So, yes, there is a homework problem starting right away, but there's no final. And there's a homework problem only every second week, so it's not a huge burden.

And there are some take-home quizzes. So two of the times where you'd normally have a homework are going to be glorified homeworks that count more than the others. And they are called quizzes. So total, I think, there are five homework problems and two quizzes.

Collaboration-- collaboration's OK on the homework problems, but please make a note of who you worked with. It's not OK on the take-home quizzes.

6.866-- so those of you in 6.866, the difference is that there's a term project. So you will be implementing some machine vision method, preferably one that we cover in the course. And there'll be a proposal due about a month from now-- I'll let as we go along-- telling me what you're planning to do.

And preference is going to be given to dynamic problems rather than single image static analysis, image motion, that kind of thing. And if there's enough interest, we'll have a session on how to do this on an Android phone.

And I'm a little reluctant to do that because some of you don't have an Android phone. And I have some loaners. But you know what it's like-- these darn things go out of fashion in two years. And so all of the interesting new stuff having to do with a camera on Android is not in some of the box full of old smartphones I have.

But that is an option. So one of the ways of doing your term project is to do an Android studio project. And to help you with that, we have a canned ready-made project that you can modify rather than starting from scratch.

OK, what else? Grades-- so for 6.801, it's a split-- half for your homework problems and half for your take-home quizzes. So clearly, the take-home quizzes count more. For 6.866, it's split three ways-- a third for take-home homework problems, a third for quizzes, and a third for the project.

And again, collaboration on the projects I actually favor, because there's just a finite length of time in the term. You've got other courses to deal with. Oftentimes, people end up postponing it near the end. So if you're working with someone else, that can often encourage you to start early and also make sure that you're making some progress.

Textbook-- there's no textbook, as you saw. If you have *Robot Vision,* that could be useful. We're not going to cover all of *Robot Vision,* we cover maybe a third to a half. And quite a lot of the material we cover is referenced through papers, which we will put up on the Stellar website.

So in fact, if you look at the website, you'll see there's a lot of material. And don't be scared. I mean, a lot of that is just for your reference. Like, if you're working on your project, then you need to know how to do-- I don't know-- SIFT, then it's there. So you're not expected to read all of that.

So it's the *Robot Vision* book. It should be on the website. If it is not on the materials, so when you get to the Stellar website, there's a tab-- there's two tabs. And the second one is-- I forget what, but that's the one where all the good stuff is.

And then when you get to that page, one of the windows says, Material. And unfortunately, it only shows you a little bit of it. You have to click on it to see all the materials. So it should be there. And we'll be doing this with some of the other chapters and some of the papers, as I mentioned.

OK, also, of course, there are errors in the textbook. And so the errata for the textbook are online. So if you have the book, you could go through and red mark all of the bad spots.

So reading, read chapters 1 and 2. Don't worry about all the reference material. You won't be reading all of it.

So what are we doing today? Well, mostly I need to tell you enough so you can do the homework problem. That's one function.

And the other one is to give you an idea of what the course is about. And these two things kind of conflict. So I'll try and do both.

In terms of the course, I am supposed to tell you what the objectives are. So I made up something. Learn how to recover information about environment from the images.

And so we're going to take this inverse graphics view where there's a 3D world out there, we get 2D images, and we're trying to interpret what's happening in the world. Vision is an amazing sense because it's non-contact and it provides so much information.

But it's in a kind of coded form, because we're not getting all the information that's possible. We don't get 3D, for example. So that's the topic that we're going to discuss. And hopefully, you will then understand image formation and understand how to reverse that to try and get a description of the environment from the images.

Outcomes-- well, you'll understand what's now called physics-based machine vision. So the approach we're going to take is pretty much-- they're light rays, they bounce off surfaces, they form an image. And that's physics-- rays, lenses, power per unit area, that kind of stuff.

And from that, we can write down equations. We can see how much energy gets into this pixel in the camera based on the object out there. How it's illuminated, how it reflects light, and so on.

And from the equations, we then try to invert this. So the equations depend on parameters we're interested in, like speed, time until we run into a wall, the type of surface cover, and so on. So that's physics-based machine vision.

And it's the preparation for more advanced machine vision courses. So there's some basic material that everyone should know about how images are formed. That's going to be useful for other courses.

And if you're going into learning approaches, one of the advantages of taking this course is it'll teach you how to extract useful features. So you can learn with raw data, like just the gray levels at every pixel. And that's not a particularly good approach. It's much better if you can already extract information, like texture, distance, shape, size, and so on. And do the more advanced work on that.

And, well, also, one of the things some people enjoy is to see real applications of some interesting but relatively simple math and physics. It's like, sometimes we forget about this when we're so immersed in programming in Java or something. But there's a lot of math we learned, and sometimes resent the learning because, like, why am I learning this.

Well, it's neat to find out that it's actually really useful. And so that brings me to the next topic, which is that, yes, there will be math, but nothing sophisticated. It's engineering math-- calculus, that kind of thing, derivatives, vectors, matrices, maybe a little bit of linear algebra, maybe some ordinary differential equation, that kind of stuff, nothing too advanced, no number theory or anything like that. And there'll be some geometry and a little bit of linear system.

So you saw the prerequisite was 6.003. And that's because we'll talk a little bit about convolution when we talk about image formation. But we're not going to go very deep into any of that. First of all, of course, it's covered in 6.003 now, since they changed the material to include images. And then we have other things to worry about.

So that's what the course is about. I should also tell you what it's not. So it's not image processing. So what's the difference? Well, image processing is where you take an image, you do something to it, and you have a new image, perhaps improved in some way, enhanced edges, reduce the noise, smooth things out, or whatever.

And that's that provides useful tools for some of the things we're doing. But that's not the focus of the course. There are courses that do that. I mean, 6.003 does some of it already. 6.344 or 6.341, they used to be 6.342. So there's a slew of image processing courses that tell you how to program your DSP to do some transformation on an image.

And that's not what we're doing. This is not about pattern recognition. So I think of pattern recognition as you give me an image and I'll tell you whether it's a poodle or a cat. We're not going to be doing that.

And, of course, there are some courses on that touch on that in Course 9, particularly with respect to human vision and how you might implement those capabilities in hardware. And of course, machine learning is into that.

And that brings me to machine learning. This is not a machine learning course. And there are 6.036, 6.869, 6.862, 6.867, et cetera, et cetera. So there are plenty of machine learning courses. And we don't have to touch on that here.

And also, I want to show how far you can get just understanding the physics of the situation and modeling it without any black box that you feed examples into. In other words, we're going to be very interested in so-called direct computations, where there's some simple computation that you perform all over the image and it gives you some result, like, OK, my optical mouse is moving to the right by 0.1 centimeter, or something like that.

It's also not about computational imaging. And what is that about? So computational imaging is where image formation is not through a physical apparatus, but through computing. So it sounds obvious.

Well, we have lenses. Lenses are incredible. Lenses are analog computers that take light rays that come in and reprogram them to go in different directions to form an image. And they've been around a few hundred years.

And we don't really appreciate them, because they do it at the speed of light. I mean, if you try to do that in a digital computer, it would be very, very hard. And we perfected them to where I just saw an ad for a camera that had a 125-to-1 zoom ratio. I mean, if the people that started using lenses like Galileo and people in the Netherlands, they'd be just amazed at what we can do with lenses.

So we have this physical apparatus that will do this kind of computation, but there are certain cases where we can't use that. So for example, in computed tomography, we're shooting X-rays through a body, we get an image, but it's hard to interpret. I mean, you can sometimes see tissue with very high contrast, like bones will stand out.

But if you want the 3D picture of what's inside, you have to take lots of these pictures and combine them computationally. We don't have a physical apparatus like an X-ray lens mirror gadget interferometer that final result is the image. Here, the final result is computed.

Even more, so an MRI-- we have a big magnet with a gradient field, we have little magnets, that modulate it. We have RF, some signal comes out, it gets processed. And ta-da, we have an image of a cross-section of the body. So that's computational imaging. And we won't be doing that.

There is a course, 6.870, which is not offered this term, but it goes into that. And we're also not going to say much about human vision. Again, Course 9 will do that.

Now in the interest of getting far enough to do the homework problem, I was going to not do a slideshow. But I think it's just traditional to do a slide. Show so I will try and get this to work. It's not always successful because my computer has some interface problems. But let's see what we can do.

OK, so let's talk about machine vision and some of the examples you'll see in this set of slides. Not all of it will be clear with my brief introduction. But we'll go back to this later on in the term.

So what are the sorts of things we might be interested in doing? Well, one is to recover image motion. And you can imagine various applications in, say, autonomous vehicles and what have you.

Another thing we might want to do is estimate surface shape. As we said, we don't get 3D from our cameras-- well, not most cameras. And if we do get 3D, then it's usually not very great quality.

But we know that humans find it pretty straightforward to see three-dimensional shapes that are depicted in photos, and photos are flat. So where's the 3D come from? So that's something we'll look at.

Then there are really simple questions, like, forgot my optical mouse. How do optical mice work? Well, it's a motion vision problem. It's a very simple motion vision problem, but it's a good place to start talking about motion vision.

So as I mentioned, we will take a physics-based approach to the problem. And we'll do things like recover observer motion from time varying images. Again, we can think of autonomous cars.

We can recover the time to collision from a monocular image sequence. That's interesting because think that to get depth we might use two cameras and binocular vision, like we have two eyes and a certain baseline and we can triangulate and figure out how far things are away. And so it's kind of surprising that it's relatively straightforward to figure out the time to contact, which is the ratio of the speed to the distance.

So if I've got 10 meters to that wall and I'm going 10 meters per second, I'll hit it in a second. So I need to do two things. I need to estimate the distance and I need to estimate the speed. And both of these are machine vision problems that we can attack.

And it turns out that there's a very direct method that doesn't involve any higher level reasoning that gives us that ratio. And it's very useful. And it's also suggestive of biological mechanisms, because animals use time to contact for various purposes, like not running into each other.

Flies, pretty small nervous system, use time to contact to land. So they know what to do when they get close enough to the surface.

And so it's interesting that we can have some idea about how a biological system might do that. Contour maps from aerial photographs-- that's how old maps are made these days. And we'll talk about some industrial machine vision work.

And that's partly because actually those machines those systems really have to work very, very well, not like 99% of the time. And so they actually pooh-pooh some of the things as academics talk about, because they're just not ready for that kind of environment. And they've come up with some very good methods of their own. And so it'll be interesting to talk about that.

So at a higher level, we want to develop a description of the environment just based on images. After we've done some preliminary work and put together some methods, we'll use them to solve what was at one point thought to be an important problem, which is picking an object out of a pile of objects.

So in manufacturing, often parts are palletized or arranged. Resistors come on a tape. And so by the time they get to the machine that's supposed to insert them in the circuit board, you know its orientation. And so it makes it very simple to build advanced automation systems.

But when you look at humans building things, there's a box of this and there's a box of that and there's a box of these other types of parts. And they're all jumbled. And they don't lie in a fixed orientation so that you can just grab them using fixed robotic motions.

And so we will put together some machine vision methods that allow us to find out where a part is and how to control the manipulator to pick it up. We'll talk a lot about ill-posed problems. So according to Hadamard, ill-posed problems are problems that either do not have a solution, have an infinite number of solutions, or, from our point of view, most importantly, have solutions that depend sensitively on the initial conditions.

So if you have a machine vision method that, say, determines the position and orientation of your camera, and it works with perfect measurements, that's great. But in the real world, there are always small errors in measurements.

Sometimes you're lucky to get things accurate to within a pixel. And what you want is not to have a method where a small change in the measurement is going to produce a huge error in the result. And unfortunately, the field has quite a few of those. And we'll discuss some of them.

A very famous one is the so-called eight-point algorithm, which works beautifully on perfect data, like your double precision numbers. And even if you put in a small amount of error, it gives you absurd results. And yet many papers have been published on it.

OK. We can recover surface shape from monocular images. Let's look at that a little bit. So what do you see there? Think about what that could be. So if you don't know what it is, do you see it as a flat surface? Let's start there.

So no, you don't see it as a flat surface. So that's where I was really going with this. I promise you this scheme is perfectly flat. There's no trickery here. But you are able to perceive some three-dimensional shape, even though you're unfamiliar with this surface, with this picture.

And it happens to be gravel braids in a river north of Denali in Alaska in winter, covered in snow, and so on. But the important thing is that we can all agree that there's some groove here. And there's a downward slope on this side, and so on.

So that shows that even though images provide only two-dimensional information directly, we can infer a three-dimensional information. And that's one of the things we're going to explore.

So how is it that even though the image is flat, we see a three-dimensional shape. And of course, it's very common and very important. You look at a picture of some politician in the newspaper, well, the paper is flat, but you can see that face as some sort of shape in 3D, probably not with very precise metric precision. But you can recognize that person based not just on whether they have a mustache or they're wearing earrings or something. But you have some idea of what the shape of their nose is and so on.

So here, for example, is Richard Feynman's nose. And on the right is an algorithm exploring it to determine its shape. So you can see that, even though presumably he washed his face and it's pretty much uniform in properties all over, where it's curved down it's darker. Where it's facing the light source, which, in this case, is near the camera, it's bright. And so you have some idea of slope, that the brightness is somehow related to slope.

What makes it interesting is that while slope is not a simple thing, it's not one number-- it's two, right, because we can have a slope in x and we can have a slope in y. But we only get one constraint. We only get one brightness measurement.

So that's the kind of problem we're going to be faced with all the time where we're counting constraints versus unknowns. How much information do we need to solve for these variables? And how sensitive is it going to be to errors in those measurements, as we mentioned?

And there's a contour map of these nose. And I mean, once you've got the 3D shape, you can do all sorts of things. You can put it in a 3D printer and give it to him as a birthday present and whatnot. And he has a somewhat later result where we're looking at an image of a hemisphere-- well, actually an oblate ellipsoid. And we're asked to recover its shape.

And these are iterations of an algorithm that works on a grid and finally achieves the correct shape. And we'll talk about the interesting intermediate cases where there's ridges where the solution is not satisfied. And the isolated points that are conical. And it's interesting in this case to look at just how the solution evolves.

So here's a overall picture of a machine vision in context. So first we have a scene, where world out there. And the illumination of that scene is important. That's why that's shown, although it's shown with dotted marks because we're not putting that much emphasis on it.

There's an imaging device, typically with a lens or mirrors or something. And we get an image. And then the job of the machine vision system is to build a description. And when it becomes interesting is when you then use that description to go back and do something in that world.

And so in my view, some of the more interesting things are robotics applications where the proof of the pudding is when you actually go out and the robot grabs something and it's grabbing it the correct way. That's one way you can know. That's one constraint on your machine vision program. If your machine vision program is not working, that probably won't happen.

So in many other cases, if the final output is a description of the environment, who's to say whether it's correct. It depends on the application. I mean, if it's there for purposes of writing a poem about the environment, that's one thing. If its purpose is to assemble an engine, then it's this type of situation where we have some feedback. If it works, then probably the machine vision part work correctly.

Here's the time to contact problem that I was talking about. And as you can imagine, of course, as you move towards the surface, the image seems to expand. And that's the cue.

But how do you measure that expansion? Because all you've got are these gray levels, this array of numbers. How do you measure that? And how do you do it accurately and fast?

And also we've noted that somehow there are interesting aspects like one camera-- don't need two. The other one is that for many of the things we do, we need to know things about the camera, like the focal length. And we need to know where the optical axis strikes the image plane.

So we've got this array of pixels. But where's the center? Well, you can just divide the number of columns and the number of rows by 2. But that's totally arbitrary.

What you really want to know is, if you put the axis through the lens, where does it hit that image plane. And of course the manufacturer typically tries to make that be exactly the center of your image sensor. But it's always going to be a little bit off. And in fact, in many cases, they don't particularly care.

Because if my camera puts the center of the image 100 pixels to the right I probably won't notice in normal use. If I'm going to post on Facebook, it doesn't really make any difference. If I'm going to use it in industrial machine vision, it does make a difference. And so that kind of calibration is something we'll talk about as well.

And what's interesting is that in this particular case, we don't even need that. We don't even need to know the focal length, which seems really strange. Because if you have a longer focal length, that means the image is going to be expanded. So it would seem that would affect this process.

But what's interesting is that at the same time as the image is expanded, the image motion is expanded. And so the ratio of the two is maintained. So from that point of view, it's a very interesting problem. Because unlike many others, we don't need that information.

So here's an example of approaching this truck. And over here's a plot-- time, horizontal. And vertical is the computed time to contact. The red curve is the computed. And the barely visible green dotted line is the true value.

In the process, by the way, we expose another concept, which is the focus of expansion. So as we approach this truck, you'll notice that we end up on the door, which is not the center of the first image. So we're actually moving at an angle. We're not moving straight along the optical axis of the camera, but we're moving at an angle.

And the focus of expansion is very important, because it tells us in 3D what the motion vector is. So in addition to finding the time to contact, we want to find the focus of the expansion.

And there's another one. This one was done using time lapse, moving the car a little bit every time. And, well, I'm not very good at moving things exactly 10 millimeters. So it's a bit more noisy than the previous one.

So, yeah, we'll be talking a little bit about coordinate systems and transformations between coordinate systems. For example, in the case of the robot applications, we want to have a transformation between a coordinate system that's native to the camera. When you get the robot, it has kinematics programmed into it so that you can tell it in x, y, z where to go, and in angle how to orient the gripper.

But that's in terms of its defined coordinate system, which is probably the origin's in the base where it's bolted in the ground. Whereas your camera up here, it probably likes a coordinate system where its center of projection is the origin. So we'll have to talk about those kinds of things.

And I won't go into that. We'll talk about this later. So I mentioned analog computing. And now we just automatically-- everything is digital. But there are some things that are kind of tedious. If you have to process 10 million pixels and do complicated things with them, since a digital computing isn't getting any faster, that can be a problem.

OK. So you can use parallelism. So there's still an interest in analog. And so here, this is the output of a chip that we built to find the focus of expansion. And it's basically instantaneous, unlike the digital calculation.

And the plot is a little hard to see. But let's see, the circle, they're determined by two different algorithms. And you can see that there's some error. But overall, the cross-- the x and the old are sort of on top of each other.

This was a fun project because to have a chip fabricated is expensive. And so you can't afford to screw up too many times. And of course, with an algorithm this complicated, what's the chance you'll get it right the first time?

So the student finally reached the point where OPA wouldn't pay for any more fabs. And the last problem was there was a large current to the substrate, which caused it to get warm. And of course, once it gets hot, it doesn't work anymore. So he'd come in every morning with a cooler full of ice cubes and a little aquarium pump and cooled his focus of expansion chip to make sure that it wouldn't overheat.

So we talked a little bit about projection and motion. Let's talk about brightness. So as you'll see, you can split down the middle what we'll have to say about image formation.

So the first half is the one that's covered in physics projection. It answers the question where so what is the relationship between points in the environment and points in the image? Well, raised-- you connect it with a straight line through the center of projection, and you're pretty much done. That's called perspective projection. And we'll talk about that.

But then the other half of the question is, how bright. What is the gray level at a point in color terms, RGB values, at a point? And so that's less often addressed in some other courses. And we'll spend some time on that.

And obviously, we'll need to do that if we're going to solve that shape from shading problem, for example. So what is this? So we've got three pictures here taken from pretty much the same camera orientation and position of downtown Montreal. And obviously if you go to a particular pixel in the three images, they're going to have different values. Of course, the lighting has changed.

So what this illustrates right away is that illumination plays an important role. And obviously we'd like to be insensitive to that. And in fact, if you showed anyone one of these three pictures separately, they'd say, oh, yeah, OK, that's plus Sainte Ville Marie. And they wouldn't even think about the fact that the gray levels are totally different, because we automatically accommodate that difference.

So we'll be looking at diagrams like this where we have a light source shown as the sun, and an image device shown as an eye, and a tiny piece of the surface. And the three angles that control the reflection. And so what we see from that direction is a function of where that light comes from, what type of a material it is, and how it's oriented.

And we'll particularly focus on that orientation question. Because if we can figure out what the surface orientation is at lots of points, we can try and reconstruct the surface. And there's that business of counting constraints, again, because what's the surface orientation? It's two variables. Because you can tilt it in x and you can tilt it y. That's the crude way to see why that is.

And what are we getting? We're getting one brightness measurement. So we kind of it's not clear you can do it. It might be under constraint.

And the image you get of an object depends on its orientation. And the way I've shown it here is to show the same object basically in many different orientations. And not only does it outline change, but you can see the brightness within the outline depends a lot on that as well.

And things depend a lot on the surface reflecting properties. So on the left, we have a matte surface-- white matte paint out of a spray can. And on the right we have a metallic surface. And so even though it's the same shape, we have a very different appearance. And so we'll have to take that into account and try and understand how do you describe that. What equation or what terminology shall we use for that?

So we'll jump ahead here to one approach to this question, which is, suppose we lived in a solar system with three suns that have different colors. This is what we get there's a cube. And it would make things very easy, right, because there's a relationship between the color and the orientation.

So if I have that particular type of blue out there, I know that the surface is oriented in that particular way. So that would make the problem very easy. And so that leads us to an idea of how to solve this problem.

So as I mentioned, there's this so-called bin of parts problems, which we were foolish enough to believe what the mechanical engineers wrote in their annual report. So what they said was, here are the 10 most important problem to solve in mechanical engineering. And this was, I forget, number 2-- how to pick parts when they're not palletized, when they're not perfectly arranged.

And so here the task is to take one after another of these rings off the pile of rings. And of course, if they were just lying on the surface, it would be easy, because there are only that many stable positions. Well, for this object only two. And so it would be pretty straightforward.

But since they can lie on top of each other, they can take on any orientation in space. And also, they obscure each other. And also shadows of one fall on the other. So it gets more interesting.

And you can see that it took many experiments to get this right. So these objects got a little bit hammered. So you have to be insensitive to the noise due to that.

And we need a calibration. So we need to know the relationship between surface orientation and what we get in the image. And so how best to calibrate?

Well, you want an object of known shape. And nothing better than a sphere for that. It's very cheap. You just go to the store and buy one. You don't have to manufacture a paraboloid or something.

And this may be a little odd picture, but now this is looking up into the ceiling. So in the ceiling, there are three sets of fluorescent lights. And in this case, they're all three turned on.

But in the experiment, they're used one at a time. So we have three different illuminating conditions. And we get a constraint at each pixel out of each one. So ta-da-- we have enough constraints.

We've got the three constraints at every pixel. We need two for surface orientation. And we have an extra one. Well, the extra one allows us to cope with albedo, changes in reflectance. So we can actually recover both the surface orientation and the reflectance of the surface, if we do this with three lights.

So here's our calibration object illuminated by one of those lights. And now we repeat it with the other two. And just for human consumption, we can combine the results into an RGB picture. So this is actually three separate pictures. And we've used them as the red, green, and blue planes of a color picture.

And you can see that different surface orientations produce different colors. Meaning, different results under the three illuminating conditions. And so conversely, if I have the three images, I can go to a pixel, read off the three values, and figure out what the orientation is.

And you might see a few things. One of them is that there are certain areas where the color is not changing very rapidly. Well, that's bad, right. Because that means that if there's some small error in your measurement, you can't be sure exactly where you are.

And the other area's where the color is changing pretty dramatically. And that's great because any tiny change in surface orientation will have an effect. And so one of the things we'll talk about is that kind of noise gain, that sensitivity to measurement error.

Why worry about it? Well, images are noisy. So first of all, one of the images-- you're looking at the 8-bit images. There's one part in 256. That's really crude quantization.

And you can't even trust the bottom one or two bits of those. If you're lucky and you get raw images out of a fancy DSLR, you might have 10 bits or 12.

Another way to look at it is that a pixel is small. How big is a pixel in a typical camera? So we can figure it out. So the chip is a few millimeters by a few millimeters. And we got a few thousand by a few thousand columns and rows.

So it's a few microns. And they're huge trade-offs. Like the one in you in your phone has smaller pixels. The one in a DSLR has larger pixels. But in any case, they're tiny.

Now imagine light bouncing around the room. A little bit of that light goes through the lens. And a tiny, tiny part of that gets onto that one pixel. So the number of photons that actually hit a pixel is relatively small. It's like a million or less.

And so that means that now we have to worry about statistics of counting. As you can imagine, if you have 10 photons, is it nine? Is it 10? Is it 11? That's a huge error.

So if you're a million, it's already better. It's like one in the 1,000. But so the number of photons that can go into a single pixel is small. But not only is there a little light coming in, but actually the pixel itself can't store that much. The photons are converted to electrons. Each pixel is like a tiny capacitor that can take a certain charge before it's full. So anyway, images are noisy. So we have to be cognizant of that.

So that was the calibration. Now we go to the real object. And again, different surface orientations produce different colors.

From that, we can construct this so-called needle diagram. So imagine that we divide the surface up into little patches. And at each point, we erect the surface normal. And then these tiny little-- may be hard to see-- but they're tiny, little bluish spikes that are the projections of those surface normals. So in some areas, like here, they're pretty much pointing straight out at you.

So here you're you looking perpendicular onto the surface. Whereas over here, the surface is curving down and you're looking sideways. So that's a description of the surface and we could use that to reconstruct the shape. But if we're doing recognition and finding out orientation, we might do something else.

So here, you see it's actually slightly more complicated, because you've got shadows. And it's harder to see, but there's also interflection. That is, with these white objects, light bounces off each of them in a mat way, goes everywhere. And it spills onto the other surfaces. So it's not quite as simple as I explained.

So what do we do with our surface normals? Well, we want a compact convenient description of shape. And for this purpose, one such description is something called an extended Gaussian image, which we'll discuss in class where you take all of those needles and you throw them out onto a sphere.

And so for example, for this object, we have a flat surface at the top. All of those patches of that surface have the same orientation. So they're going to contribute that big pile of dots at the North Pole. So just cut that short. It's a representation in 3D that's very convenient if we need to know the orientation of the object, because if we rotate this object, that representation just rotates. You can think of many other representations that don't have that property.

OK, so here it is. You could imagine that it wasn't easy to get the sponsor of the project to pay for these parts here. I think they were concerned they were not for experimental purposes.

So this is a single camera system, so there's no depth. So the way this works is that you do all this image processing. You figure out which object to pick up and how it's oriented. And then you reach down with a hand until a beam is interrupted, then you know the depth.

So here the beam is interrupted. And now the robot backs up. And here it orients the hand for grasping. And then it comes back and grasps that object, and so on.

And I show this because another calibration I left out was what I previously mentioned-- the relationship between the robot coordinate system and the vision system coordinate system. And one way of dealing with that is to have a robot carry around something that's easy to see and accurately locatable.

This is something called a surveyor's mark, because surveyors have used that trick for a very long time. It's easy to process the image. And you can find the location of the intersection of these two lines very accurately with sub-pixel accuracy.

So you move that around in the workspace and then fit the transformation to it. And then you can use that to-- OK, back to more serious stuff.

So that should give you a taste of the kind of thing that we'll be doing. And what I'm going to do now is work towards what you need for the homework problem. So first, are there any questions about what you saw? I mean, a lot of that's going to get filled in as we go through the term.

So I mentioned this idea of inverse graphics. So if we have a world model, we can make an image. People who are into graphics will hate me saying that. But that's the easy part. That's the forward problem. It's well-defined.

And the interesting part is, how do you do it well? How do you do it fast? How do you do it when the scene has only changed slightly and you don't want to have to recompute everything and so on.

But what we're trying to do is invert that process. So we take the image. And we're trying to learn something about the world. Now we can't actually reconstruct the world. We typically don't end up with a 3D printer doing that.

Usually, this ends as a kind of description. It might be a shape or identity of some object or its orientation in space, whatever is required for the task that we have. It might be some industrial assembly task, or it might be reading the print on a pharmaceutical bottle to make sure that it's readable, and so on. But that's the loop. And that's why we like to talk about it as inverse graphics.

Now to do that, we need to understand the image formation. And that sounds pretty straightforward, but it has two parts, both of which we'll explore in detail as we go along.

Then with inverse problems, like here we're trying to invert that, we often find that they're ill-posed. And as I mentioned, that means that they don't have a solution, have an infinite number of solutions, or have solutions that depend sensitively on the data.

And that doesn't mean it's hopeless, but it does mean that we need methods that can deal with that. And often we'll end up with some optimization method. And in this course, the optimization method of choice is least squares.

Why is that? Well, the fancy probability people will tell you that this is not a robust method. If you have outliers, it won't work very well. And that's great.

But in many practical cases, least squares is easy to implement and leads to a closed form solution. Wherever we can get a closed form solution, we're happy, because we don't have iteration. We don't have the chance of getting stuck in a local minimum or something. So we'll be doing a lot of least squares.

But we have to be aware of-- I already mentioned-- noise gain. So not only do we want to have a method for solving the problem, but we'd like to be able to say how robust it is. If my image measurements are off by 1%, does that mean that the answers are completely meaningless? Or does it mean that they're just off by 1%. So that kind of thing.

Diving right in, we're going to address this problem. And it's straightforward. And we'll start off with something called the pinhole model.

Now we know that real cameras use lenses, or in some cases mirrors. Why pinholes? Well, that's because the projection in the camera with a lens is the same-- it's trying to be exactly the same as a pinhole camera.

By the way, there's a great example of a pinhole camera in Santa Monica. It's a camera obscura. You walk into this small building that's completely windowless. It's dark inside. And there's a single hole in the wall.

And on the other side on the other wall painted white, you see an inverted image of the world. And you see people walking by and so on. So that's a nice example of a pinhole camera.

So here's a box to keep the light out. And then we have a hole in it. And on the opposite side of the box, we see projected a view of the world.

So let's just try and figure out what that prediction is. So there's a point in the world, uppercase P. And there's a little p point in the image plane.

So the back of the box is going to be our image plane. And our retina is not flat. We're just going to deal with flat image sensors because all the semiconductor sensors are flat. And if it's not flat, we can transform. But we'll just work with that.

So what we want to know is what's the relationship between these two. And so this is a 3D picture. And now let me draw a 2D picture.

OK, so we're going to call this f. And f is alluding to focal length. Although in this case, there's no lens. There's no focal length. But we'll just call that distance f.

And we'll call this distance little x. And we'll call this distance big X, and this distance big Z. So in the real world, we have a big X, big Y, big Z. And in the image plane, we have little x. And we're going to have little y and f.

And, well, there's similar triangles. So we can immediately write. OK And although this isn't completely kosher, I can do the same thing in the y plane.

So I can draw the same diagram, just slice the world in a different way and I get the companion equation. And that's it. That's perspective projection.

Now why is it so simple? Well, it's because we picked a particular coordinate system. So we didn't just have an arbitrary coordinate system in the world. We picked a camera-centric coordinate system. And that's made the equation just about trivial.

So what did we do? Well, this point here is called the center of projection. And we put that at the origin. We just made that 0, 0, 0 in the coordinate system. And so this is also COP. And then he has the image plane, IP, Image Plane.

OK, so we did two things. The one was we put the origin at the center of projection. And the other one is we lined up the axes with the optical axes.

So what's the optical axis? Well, in a lens, a lens has a cylindrical symmetry. So it has the cylinder has an axis. But there's no lens here.

But what we can do is we can look at where a perpendicular dropped from the center of projection onto the image plane strikes the image plane. So we've used that as a reference. And so that's going to be our optical axis. It's the perpendicular from the center of projection onto the image plane. And we line up the z-axis with that. That's going to be our z-axis.

So it's a very special coordinate system, but it makes the whole thing very easy. And then if we do have a different coordinate system on our robot or whatever, we just need to deal with the transformation between this special camera-centric coordinate system and that coordinate system.

Now one of the things that's very convenient-- well, not only are they going to make me walk across campus, but I'm going to get upper body strength as well. This is great.

OK, so what we do is we flip the image plane forward. So the image on your retina is upside down. And in many cases, that's inconvenience. So what we can do is we can just pretend that the world actually looks like this.

That's pretty much the same diagram. We've just flipped 180 degrees what was behind the camera and in front. And it makes the equations even more obvious. The ratio of this to that is the ratio of this to that.

Now that sounds straightforward and somewhat boring. But it has a number of implications. The first one is it's non-linear. So we know that things are linear, our math becomes easier and so on.

But here we're dividing by z. So on the one hand, that's an inconvenience, because, like you take derivatives and stuff or the ratio. That's not so nice.

But on the other hand, it gives us some hope. Because if the result depends on z, we can turn that on its head and say, oh, maybe then we can find z. So we can get an advantage out of what seems like a disadvantage.

And then the next thing is-- we won't to do it today, but we'll be doing it soon-- is to talk about motion. So what happens? Well, we just differentiate that equation with respect to time.

And what will that give us? Right now, we have a relationship between points in 3D and points in the image. And when we differentiate, we can get a relationship between motion in 3D and motion in the image.

And why is that interesting? Well, it means that if I can measure motion in the image, which I can, I can try and guess what the motion is in 3D.

Now the relationship is not that simple. For example, if the motion in 3D is straight towards me, the baseball bat is going to hit me in the head, then the motion in the image is very, very small.

So you'll have to take into account that transformation. But I do want to know that the relationship between motion in 3D and motion in 2D. And I get it just by differentiating that.

Then, I want to introduce several things that we use a lot in the course. The next one is vectors. So we're in 3D. Why am I talking about components? I should be just using vectors.

So first of all, notation. In publications, in engineering publication, not math publication, vectors are usually denoted with bold letters. And so if you look at *Robot Vision* or some paper on the subject, you'll see vectors in bold.

Now I can't do both on the blackboard. And so we use underline. And actually, there was a time where you didn't type set your own papers-- just a second. But somebody at the publisher type set your paper. So how did you tell them to type set in bold? You underlined it.

I mean, the camera actually works the way that works up there in most cases. Some of them will have mirrors to fold the optical path. This is like a conceptual convenience, just to make it easier, not to have to.

I mean, maybe some people don't have a problem with minus signs. But to me, it's confusing having that one upside down. So I prefer to do it this way. But the actual apparatus that works that way.

So the other bit of notation that we need is a hat for unit vector, because we'll be dealing with unit vectors quite a bit. For example, you saw that we talked about the surface orientation on that donut in terms of unit vectors. It's a direction. So we use a hat on top of a vector.

And so let's turn that into vector notation. Well, I love this. So I claim that this is basically the same as that up there, right. Because if you go component by component, the first component is little x over f is big X over big Z.

The second component is letter y over f is big Y over big Z. And the third component is f over f is z over z. So that doesn't do anything to us. So that's the equivalent.

And now I can just define a vector r. So this is little r. Now I've got a mixed notation, right, because I've got a big Z in here. Well, that's the third component of big R vector. So I just dot product with unit vector z.

So let me write that out in full. So that's x, y, z, transpose dot. So the unit vector in the z direction along the optical axis is just 0, 0, 1. And so I finally have the equivalent of the equations up there in component form. I have it here in vector form. So that's perspective projection in vector form.

Now usually at this point, you say, look, how easy it got by using vector notation. Well, it isn't really any easier looking. This is one of those rare cases where it didn't buy you a whole lot in terms of number of symbols you have to write down, and so on.

Nevertheless, the compactness of that notation comes out when we start manipulating it. If you have to carry around all these individual components all the time, that can get pretty tedious. Whereas if you use the vector, it's more interesting.

And as I've mentioned, one of the things we're going to do soon is differentiate that with respect to time. And then on the left, we'll have image motion. And on the right, we'll have real world motion. And the equation we get will give the relationship between the two.

So this may sound a little bit haphazard and chopped up, the way we're doing today. And that's only because I want to cover stuff in chapter 1 and 2 and the material you need for the homework problem. So rather than pursue perspective projection, well we're going to jump to brightness in a second.

But first, let me say something else, which is that I'm thinking of these vectors as column vectors. And that's arbitrary because we can establish a relationship between skinny matrices and vectors, either way. I can think of x, y, and z stacked up vertically above each other as a 3-by-1 matrix. Or I can write them horizontally, x, y, z, and it's a 1-by-3 matrix.

And just for consistency, I'm always going to think of them as column vectors. And that's why sometimes I need a transpose. And that's what the symbol T is for.

So I didn't say it here, but we can now go back to this. So if I write it this way, it's a row vector. But actually all my vectors are supposed to be column vectors. So it's stuck in the transpose. So another bit of notation. So all pretty straightforward, though.

OK, let's talk about brightness. So brightness depends on a bunch of different things. It depends on illumination, and in a linear way. In that you throw more illumination on an object, it's going to be brighter.

And there are few laws that are really, really linear. This is linear over many, many, many orders of magnitude. I mean, when does it stop being linear? Well, when you put so much energy on the surface that you're melting it. You have to actually have enough energy to fry it.

And it's a little bit like Ohm's law, which is also one of those remarkable things that for some materials is linear over many, many orders of magnitude. Anyway, but this depends on the illumination. And then it depends on how the surface reflects light. And so we'll have to talk about that.

Now, obviously, there's a difference in terms of amount. So my laptop reflects relatively little light. Whereas, my shirt reflects more light.

Anyone want to guess what the percentage of incident solar radiation that the moon reflects? It's a trick / Do you happen to know?

It sort of looks white in the sky. So it's got to be 90% or something? It's 11%. It's as black as coal. And so why don't you know that? Well, because you have no comparison.

Now if I went up there with a sheet of white paper and held it next to the moon, you'd say oh, yeah, God, it's really dark. But no one does that, it just hangs up there and you have no reference.

So this business about brightness is tricky. You got to be careful about that. And by the way, why is it as dark as coal? It's because of solar wind impinging on the surface.

And you also probably know that where they were, quote, "recent" craters like only in the last few million years. They have bright streaks. That's because where the underlying material is exposed and the sun hasn't yet done its work on them. Anyway, so brightness depends on reflectance.

How about distance? If I have a light bulb, it's less intense when I go further away. So there's an inverse square law. So does that apply to image formation.

In the more normal sense, if I walk away from that wall, if I stand on this side of the room, that wall is only a quarter as bright as when I stand over here, right. Do you believe that? I can sell you a bridge in Brooklyn, then.

No, of course, it's the same politeness. And you know that. So what's going on? Why is it not follow the inverse square law?

Well, the reason is that at the same time as I'm getting closer to, the area that's imaged on one of my receptors is larger on the wall. Or if you want to think of it in terms of the little light bulb, so the LED, imagine that the wall is covered with lots of LEDs. And each of them does, in fact, follow the 1 over r squared law.

But if you think about how many LEDs are imaged on one of my pixels, that goes as a square of the distance. So the two exactly cancel out. And so in fact, we can ask that one out. And so what else does it depend on? Well, it doesn't depend on the distance itself, but it depends on the rate of change of distance or orientation.

And not in a terribly simple way, but we can start with a simple example. So here's a surface element, some little patch of a surface. And here's a light source. And what we find is that it is foreshortening. That is the power that hits the surface. Per unit area is less.

So I can measure the power in this plane, so many watts per square meter, which in the case of the sun is about a kilowatt per square meter. But obviously that same energy is spread out over a larger area. This length is bigger than that length. And so the illumination of this surface is less.

And how much? Well, we can express it in terms of this angle, which is the incident angle, theta of i. And that is the same angle as that angle, I think. And there's a co-signed relationship between this red length and this length.

So we find out that, in this case, the illumination on the surface varies as the cosine of the angle. And this is something that we'll see again and again.

Now it doesn't necessarily mean that its brightness, the amount of light it reflects, goes as the cosine of the incident angle. That is the simplest case. And so here's an example where we could use an image brightness measurement to learn something about the surface, because we can look at different parts of the surface. And they'll have different brightnesses, depending on this angle.

Now, does it tell us the orientation of every little facet of the object? Some people are shaking their heads. No, right, because, again, it's one measurement two unknowns. Why are there two unknowns?

Well, one way to see it is to think of a surface normal, a vector that's perpendicular to the surface. And the way I can talk about the orientation of the surface is just to tell you what that unit normal is.

So how many degrees of freedom are there? How many numbers? Well, I need three numbers to define a vector. So it sounds like three, except I have a constraint-- three components.

This isn't just any old vector-- this is a unit vector. So x squared plus y squared plus z squared equals to 1. So I have one constraint. So actually surface orientation has two degrees of freedom.

And since this is such an important point, let's look at it another way. So another way of specifying surface orientation is to take this unit normal and put its tail at the center of a unit sphere and see where it hits the sphere.

And so every surface orientation then corresponds to a point on the sphere. And I can talk about points on the sphere using various ways, but one is latitude and longitude. And that's two variables.

So that tells us, again, in another way that a unit normal has two degrees of freedom. And if I want to pin it down, I better have two constraints. So that's the way that was going.

And that makes it interesting. I mean, if we could just say, OK, I'll measure the brightness and it's 0.6 and the orientation is such and such, the course would be over. It'd be pretty boring. But it isn't. It's not easy. We need more constraint. And we'll see different ways of solving this problem.

One of them you saw in the slides was a brute force one saying, well, we just get more constrained. We illuminated with a different light source. We get a different constraint because the other light source would have a different angle. And then I can solve at every point.

So from an industrial implementation point of view, that's great. You can do that. You can either use multiple light sources, put on at different times, or you can use colored light sources, and so on.

But suppose you were interested in, how come people can do this. They don't play tricks with light sources. And they don't live in a world with three suns of different colors. Then we'll have to do something more sophisticated. And we'll study that.

How are we doing? Are we getting there? OK, so the foreshortening comes up in two places. The one is here where we're talking about incident life. But actually foreshortening also plays a role in the other direction.

Whoa, high friction blackboard. Also, hard to erase.

So it's really the same geometry, except now the rays are going in the other direction. and like so. And I have a foreshortening on the receiving end as well.

So in a real emerging situation in 3D, we'll see both of these phenomena. There's the foreshortening that affects the incident of illumination as up there. And then there's this effect.

And for example, I can illustrate to you right away the stupidity of some textbooks. So some textbooks say that there's a type of surface called inversion which emits energy equally in all directions. That's what they literally say.

Well, if that's true, then that energy is imaged in a certain area that changes as I change the tilt of the surface. And as I tilt the surface more and more and more, that image area becomes smaller and smaller. But it's receiving the same power, supposedly, according to these guys.

And what does that mean? That means you're going to fry your retina right at the occluding boundary, because all that energy is now focusing on a tiny, tiny area. So this is an important idea. And it comes in when we talk about the reflectance of surfaces. And we need to be aware of it.

So now something I want to end up on is, we're solving a tough problem. With 3D, we only got 2D images. So maybe we're lucky and we have several.

But you've got a function of three variables that's got so much more flexibility than a few functions of two variables. So why does this work at all?

Well, the reason it works is that we are not living in a world of colored Jell-O. So we're living in a very special visual world. So if I'm looking at some person back there, the ray coming from the surface of his skin to my pupil is not interrupted, and it's a straight line.

Why? Well, because we're going through air. And air is a refractive index, almost exactly 1. And at least it doesn't vary from that position to here.

And there's nothing in between. There's no smoking allowed in this room, so it can't be absorbed. And that's very unusual.

And the other thing is that person has a solid surface. I'm not looking into some semi-translucent complicated thing. So they're straight line rays and there's a solid surface. Therefore, there's a 2D-to-2D correspondence. The surface of that person-- sorry, I keep on looking at the same person. He's getting embarrassed.

But 2D, we can talk about points on the cheek of this person using two vectors, u and v. And that's mapped in some curvilinear way into the 2D in my image.

And that's one reason why this works. It's not really 3D to 2D. It's a curvilinear 2D to 2D. And what's the contrast?

Well, suppose I fill the room up with Jell-O. And then somebody goes in with the hypodermic, injects colored dye all over the show. And then I come in the door and I'm not allowed to move around. I can just stand at the door and I can look in the room.

Can I figure out the distribution of color dye? No. Because in every direction, everything is superimposed from the back of the room to the front.

And so you can't disentangle it from one view. Can you do it? Yeah, if you have lots of views. And that's tomography.

So we're in an interesting world. Tomography in a way is more complicated, but it's also in a way much simpler. The math is very simple.

And we have a world where there's a match of dimensions. But the equations are complicated. So it's not so easy to do that inversion.

I think we need to stop. OK, any questions? So about the homework problem, you should be able to do at least the first three or five questions, probably the fourth. And then on Tuesday, we'll cover what you need to do the last one.