

[SQUEAKING]

[RUSTLING]

[CLICKING]

**BERTHOLD
HORN:**

We'll start off by talking a little bit about a review of things we've done last time, perspective projection, and a little bit about things that are relevant to the homework problem, particularly the last two questions. And we'll go from perspective projection to motion. So in perspective projection we have a relationship between points in the environment and points in the image.

And we found that we had the simple relationship between the 3D world and the 2D world, as long as we pick a suitable coordinate system with the origin at the center of projection, and the z-axis along the optical axis, and so on. But then we're going to go from that to motion by differentiating that equation. And then we'll talk about motion of brightness patterns in the image itself.

So we had that, and then we had a vector version, which is only slightly more compact. So it's not, in this case, a great help. But we'll use both of them. We'll switch back and forth and use whatever is suitable for the problem that we're dealing with.

So let's start with motion. So that's a static setup. And we can easily relate points in the environment to points in the image.

Now, what if there's motion? What if that point in the environment moves? Obviously, there will be some motion in the image.

And if we can find out what that relationship is, then maybe in some circumstances, we can invert that so that we can measure the motion of brightness patterns in the image, as you do in the homework problem, and somehow use that to figure out what's happening out in the world. So we just differentiate. We take our perspective equation and we differentiate. So we get--

Now, on the right-hand side, we have a ratio. And so there are two parts to the derivative. And we just apply the rule for differentiation of a ratio.

And what I'm going to do a lot is reduce the intimidation level of the equation by substituting more easily digested symbols. So in this case, we've got all these derivatives. Well, what are they? They're really velocities.

So there's going to be a velocity in the image in the x direction, which I'll call u . And there's a velocity in the real world in the x direction, which I'll call U . And then we have a velocity in the real world in the z direction, which I'll call W .

So that looks a little bit less scary. And of course, I can do the same in the y direction, so apply the same idea over here. So that's the forward direction. That is, if you tell me what the motion vector is in 3D, here's the formula to tell me what the motion vector is in 2D. And of course, we might want to invert that, but for the moment, let's go with that.

Now, we can rewrite this in various forms. And since we'll use this a lot, let's do that. So one thing we can do is we can split up these terms into $1/z$, X/Z .

And why is that useful? Well, because we know that X/Z is little x over f . So we can rewrite this using image coordinates. And we end up with--

And one thing this allows us to do is to ask the question, where in the image is there no motion? So if we're looking at the movement of brightness patterns in the image, there may be some places where there isn't any. And those are of particular interest, first of all because we can find them using image processing techniques, and then because they tell us something about the environment or the motion.

So you can see, if we set this equal to 0, we find that's true at a point x_0 such that x_0 over f is big U over big W . And similarly for this one equal to 0. And so the point x_0, y_0 is that sought-after point where there is no image motion.

And we call it the "focus of expansion." And in the case of simple situation, like I'm approaching the wall, the focus of expansion is the point towards which I'm heading. And so we can express it directly in terms of the 3D motion vector.

And you'll notice that this is actually just a projection of the 3D motion vector into the image plane. So I take that same diagram over there, and now, instead of mapping the vector to the point P , I'm mapping the velocity. And if I do that, then that defines this point x_0, y_0 in the image where there is no motion.

And that's called the focus of expansion. That's a useful thing, because if you can find the focus of expansion, all you need to do is connect that point, x_0, y_0 to the origin, and you have a vector that tells you the direction of motion-- so a nice example of inverting that process.

Now, once we have the focus of expansion, we can rewrite those equations yet one more time. And so there are a couple of things there that are of interest. One of them is that the f 's cancel, so we can do that.

And then we actually can get an idea of what the image is going to look like. So let's suppose this is the image plane. Now we're looking at it straight on.

And suppose that this is our focus of expansion. And we're moving with respect to-- the environment is moving with respect to us or we're moving with respect to the environment. Einstein said that it's all relative. It doesn't matter. The only thing we can determine is the difference between the two motions.

Then, there will be no image motion here. And what about other places? Well, this tells you exactly what the pattern will be like. So if we go to the right such that $x_0 - x$ is positive, while we keep this one 0, what happens is U will become larger. The further we go away, the larger.

So we'll have a little vector here that tells us the image motion at that point, and another vector over here, which is perhaps larger. And then we can go to where this is negative, so the other side. Draw a vector here.

So I'm starting to draw a little vector diagram of the motion field, if you like, how things are expanding from that point. And I can do the same in the y direction. So suppose I keep x_0 equals x , but I change y_0 , so I'm going along this line.

I'm going to get vectors that point this way. And let me be adventurous and go off at right angles where x_0 minus x is the same as y_0 minus y . Well, that means that U and V are going to be the same.

That means I'm going to get a vector at 45 degrees, and opposite direction, like that. And I think you can see that we're going to fill in this whole diagram, this little vector diagram, with arrows that are pointing outwards from the focus of expansion. That that's what this equation says-- that all these vectors are radiating outward, and that's why it's called the focus of expansion.

So I'm moving towards that wall. And this is the place I'm actually going to hit. I may be going at an angle, but my whole image is looming. It's zooming outwards, which is a useful clue, a useful cue for measuring distance and velocity.

As you saw in the homework problem, there's the scale factor ambiguity that because of perspective projection, we can't actually tell absolute distances. And we have a similar phenomenon here. But what we can tell is this ratio.

And so this whole field depends critically on that ratio. So what is that? W over Z . And if I make that larger, then the vectors will be larger. That means I'm getting closer to the surface. I'm about to hit it. Conversely, if I make it smaller, that will be smaller.

If I make it negative, what happens? Well, all of these vectors are reversed. So it's a focus of compression. There's no such term, but we could call it that. It's just the inverse of the focus of expansion.

And that's what happens if I'm moving away from the surface. So I'm taking off from the lunar surface, and the image is compressing as I leave. So what is this?

Well, let's turn it on its head. What is Z over W ? And W , of course, is the rate of change of Z . So we've got Z over dZ/dT .

What are the units of that? So Z is in, say, meters. And dZ/dT is in meters per second. So the units of that ratio are--

STUDENT: Seconds.

BERTHOLD HORN: Seconds. And so can you think of what that quantity is?

STUDENT: [INAUDIBLE].

BERTHOLD HORN: Sorry?

STUDENT: Time to impact.

BERTHOLD HORN: Time to impact, great, so that's going to tell me how long it's going to take before I crash into this surface. And so that's a quantity that we're going to spend some time thinking about, both because it's important if you're say landing a NASA craft on Europa or if you're a fly landing on the ceiling. And also, it turns out to be relatively easy to compute compared to some of other things that we'll be looking at.

By the way, the landing on Europa is a JPL RFP. They wanted to have industry give them ideas of how they're going to do that, and how much it's going to cost. And we told them that this is the right way to do it.

And we never heard from them again. So I don't know. So if that craft crashes on Europa, don't blame me. I. Told them what to do.

So this is beginning to look kind of promising because we can actually see our way to inverting this imaging process and getting some useful information, just from the way the brightness pattern in the image changes. But I need to torture you a little bit more, and do this in vector form. And we'll see later why this is useful. Right now, it's not saving us a whole lot in terms of writing, going from the component form to the vector form, but we'll see that it does eventually.

So instead of differentiating the component versions, we're going to now differentiate the vector form. And of course, it's very similar. So what's the time derivative of a vector?

Well, it's just the vector where you've differentiated each of the components with respect to time. That's nothing very fancy. So we have that term again that corresponds to this first part here, but because it's a ratio, and because R can be changing, \dot{R} can be changing, we need to take account of the fact that it's a ratio.

And we get-- and so to make it look slightly less intimidating, we'll switch from Leibniz to Newtonian notation. Again, the underline always means that it's a vector.

And I'm splitting this up a bit. So the dot here denotes differentiation with respect to time. And so we have two terms, one of which is obvious scaling-- that there's a certain motion in 3D, and it's going to be magnified or demagnified by the ratio of these two distances.

So there's a big motion out here. It's a smaller motion in there by the ratio of those. But then there's this other term that corresponds to motion in depth that we have to take into account.

And I rewrote it this way because again, we can make use of the fact that we have this perspective projection equation. And so we can introduce the image coordinates. So focus of expansion, again, would be where \dot{r} is 0. And so that corresponds to z_0 , let me put the 1 over f on the other side.

That's easier. And so we basically get the same equation. I mean, if we just write out the components of that equation, we get what we had before.

Then just wanted to do one more little trick, which is in the appendix of the book, we have some useful results about all sorts of simple math that's needed, one of which has to do with vector equations and with manipulating cross products of cross products, that kind of thing, which I typically don't remember how to do. So I'll go to the appendix of the book. And the appendix is on the materials on the Stellar website.

So let me first transform this one more time. So we've got-- that's just rewriting this. And now, we use a result.

And the reason we do this is because it allows us to make some general statements about the flow. So cross product of cross products can be expressed as a difference of dot products. Now, you can match up these two lines, and you can see where we're going with this, which is--

So finally there, we have the image motion expressed in terms of world motion using this somewhat odd-looking expression. So why is this of interest? Well, the first thing you can say is that the result is perpendicular to z .

Because if you take the cross product of two vectors, the result is perpendicular to both of the vectors. So $\mathbf{r} \cdot \mathbf{z}$ equals 0. So is that surprising-- that the image motion is perpendicular to the z -axis?

So image motions in that plane-- z -axis is there. No, of course it's obvious. I mean, it couldn't be any other way. This is a good way to check the result. If we got anything else, there'd be a problem.

So in the image, we only have 2D. We have x, y . And we have as velocities u and v . We don't have a velocity in the z direction. That would be popping out of the image. So that's interesting.

And then something else we can look at is, what if the image motion is radially outward or inward-- that is, the motion is along the radius vector to the point in the scene? What do you expect will happen to the image motion in that case? So you have the baseball coming straight for you-- its image is doing what?

STUDENT: [INAUDIBLE].

BERTHOLD HORN: It's expanding, but it's not moving. So this is the case where we would expect image motion to be 0. And now we check this formula.

$\mathbf{R} \cdot \mathbf{R}$ and \mathbf{R} are parallel. When we take their cross product, the cross product's length is proportional to the sine of the angle between the two vectors. Where they're the same vector, then that angle is 0, and so the sine is 0.

And so this formula, while it took a little bit of work to get there, it's helpful for answering certain questions. And we can right away check things like this. If the thing is coming straight at you or you're moving directly towards it, there'll be no image motion.

Well, that's our focus of expansion. So if I'm moving towards the wall, that is the point where the motion vector lines up with the direction to that point. And so I get no image motion.

While we're here, we can look at some flow fields. So we already saw a flow field up there for a general motion towards some point. And we could look at just a couple of other cases.

So let's suppose that U, V and W could be anything, but let's make it-- so we're assuming now that two of those components are 0, so there's only motion in the world in the x direction. And then what do we expect in the image?

Well, from our formulas, we see that the only component of the image motion is little u . So if the world is moving in x , or I'm moving in x relative to the world, then the vector field of how brightness patterns move in the image looks like this. One thing to note, though, is that the length of those vectors is not the same-- that the length depends on z , or if you like, the inverse of z , and so on.

So if you look at all of these formulas, we got $\mathbf{R} \cdot \mathbf{Z}$, which is big Z . And look at those formulas up there. You've got $1/Z$.

So yes, this is a very simple vector field, but the vectors aren't all the same. So that kind of is unpleasant. At the same time, when something is unpleasant, you can sometimes take advantage of that. So in this case, if we have vector field like this, one thing we could try and do is recover depth, because these things are all inversely proportional to depth. And so again, an illustration of how once we understand the forward process, we can turn that around to try and solve the inverse problem.

U is 0, and let's try this one. Well, that's just turning everything 90 degrees. So that's not very interesting-- the same idea.

Now, I guess to exploit this idea of recovering depth from these fields, we would need to know Z . We need to know the velocity. So there's two things affecting what these pictures look like. One is the object, the shape, the distance, and the other one is the motion.

And if we know either one of the two, then we can calculate the other one. Of course, in practice, often we don't know either one, and we'd like to recover both. And then often, we end up with an ill-posed problem.

How about U is 0, V is 0, W is not equal to 0? Well, we already drew that diagram up there, kind of. If we look at the equations, we're just going to get F OE is-- so that's the simplest case of that picture, where the focus of expansion is right in the center of the image.

Well, this is starting to hint at some of the things we're going to pursue. We develop the equations for some transformation like this, between something in the world and something in the image, and then we try to invert it. And the forward one is always simple. There's the equation up there.

It's just plug in or whatever. The inverse isn't because often, there may be more than one solution, or there may be no solution, or there may be an infinite number of solutions, or it's ill-posed, in the sense that if you make some small measurement error, the answer is going to be perturbed by a large amount. So we'll have to be careful about that.

But talking of motion of image brightness patterns, let's talk about that. That's in the homework problem. So let's talk a little bit about that. So we'll ignore for the moment where these images come from.

What is an image? Well, an image is a 2D pattern of brightness values. So it's E as a function of x and y , if you like. And we already said something about that in terms of brightness being power per unit area. We'll make that more clear later on.

So his question was, what about color? And yes, we will be talking about color at some point, which is kind of, in a way, is this repeated three times. And a couple of reasons not to bring it in at this point.

One is, we'll do it more simply just using gray levels. And then color is peculiarly human-centric, in that the only reason that we use three numbers to represent color is because our particular color sensing system has three senses. But we'll talk about all of that later on.

For now, assume that we have taken RGB and turned it into brightness, which is, depending on who you read, some combination of the three components, mostly G. Because human vision is most sensitive to G, and judges absolute brightness based on some combination of R, G, and B.

Now, there are a couple of things I'm going to do repeatedly. So I want to kind of preemptively talk about them. One of them is I'm going to be switching back and forth between continuous and discrete.

So these days, with everything being digital, of course the images we get are discrete. They're quantized. They're quantized in space, and typically on a rectangular grid, which is kind of not the best grid to use, but that's what we use.

And then in brightness, so we also don't get continuous values for brightness. We get them quantized, often to as few as eight bits. But it turns out that a lot of what we do is easier to understand in the continuous domain.

So I've talked to you about E of x , y . That's my brightness pattern. For any x and y , it'll tell me what the brightness is, the power per unit area. But in practice, it's going to be more like I have an array of numbers with two indices, and dealing with E sub whatever.

Correspondingly, in the continuous domain, we'll often be taking integrals. For example, where there's some measurement, that's local at a pixel, but we want to extend it over the whole image. And we take either a single integral or double integral.

Well, of course in the discrete world, we'll be taking sums. But those are very straightforward transformations. But they are very useful, because on the one hand, we want to implement this as actual code, and on the other hand, we'd like to make it easier to develop the math. And it's almost inevitable that it's easier in the continuous world.

Then we get to derivatives. So of course in the continuous world, I can look at the x and y derivative of brightness. And the combination of these two is called the "brightness gradient," which is going to be very important in many of the things we do.

And in practice, I'm going to approximate those by making some sort of a first difference. And we'll see that that's one way of doing it. It's not a particularly good way of approximating the first derivative, but we'll talk about others.

And just in terms of writing, this is easier to write than that. So I know that's not a big argument, but we'll find that some of these things have a closed form solution in the continuous domain that you can easily obtain. And it's harder to do the same in the discrete domain. So I'll be switching back and forth.

With that in mind, let's look at our 1D image. So of course, images are typically 2D. There are 1D sensors, and they have some benefits.

One of them is that you can build a 1D sensor with a much larger number of pixels in one direction than you can a 2D sensor, although these days, 2D sensor could be 2K by 4K, some huge number. But linear sensors have been around with many thousands of pixels. And the disadvantage of a linear sensor is to get a real image, you need to scan it.

And this is done quite a lot in industry, where things are moving along on a conveyor belt. And you can use a relatively cheap, very high resolution, linear array sensor. And the conveyor belt motion provides the other dimension of the image. It's also used in satellite imaging, where very high-quality 1D sensors are used, and then the motion of the satellite provides the scanning to produce an image.

Anyway, suppose we have a 1D image, and suppose things move. So this is at time t . And this is at time t plus Δt . And think of your optical mouse. You're holding it down on the table.

It has a short focal length lens that's imaging the surface of the table. And if there's some texture to it, as there is on the wood over here, then that texture is mapped onto the image. And if you move the mouse, then the image of that surface will move.

And your job is to accurately estimate how fast it's moving. And so here's the picture before, two time steps. And so we're trying to figure out, how do I determine what happens?

So let's suppose we have a velocity of u . And so we've moved by that amount. So let me indicate that over here. So that's Δx .

Now, we're making some assumptions here that we'll get back to in a second. One of them is that the brightness doesn't change. And so if your optical mouse is looking at the surface of the table, then presumably between one frame and the next, the illumination isn't changing. It's using an LED with a constant current, and it's taking 2,000 frames a second, so it's unlikely to change.

The geometry is changing a little bit because you've moved, and now you're looking at the surface from a slightly different angle. But for most surfaces, the change in brightness due to that is very small. So right here, the way I've drawn these curves, I've assumed constant brightness. So let's get it right out front-- constant brightness assumption.

And we'll see that there are circumstances where that doesn't apply. But there are lots of cases where that is the case. For example, as I'm walking around the room, your shirt still looks maroonish red, and it doesn't really change.

And so we are used to most of the world being somewhat constant in color, constant in brightness, as well, even though we know that there are circumstances where that's not the case. For example, if I look at the reflection of the lights up there in my cell phone, they, of course, move relative to the cell phone as I move around. And we'll talk some more about that.

So let's enlarge a little piece of that. So we're blowing up this area. And why am I doing that?

Well, I'd like to use a linear approximation. So I'm assuming that this-- second assumption-- I'm assuming for the moment this curve is relatively smooth, such that if I look at a small enough part of it, I can approximate it as a line. And then there's a change in brightness, and there's some slope here.

And the slope relates the motion to the change in brightness. You can see where this is going. I'd like to use the change in brightness to determine the motion, because how else are we going to measure the motion in the image?

The pixels stay in the same place. All we've got is the brightness at a pixel is changing. And somehow, we have to invert that relationship.

So in terms of E of x and t , what is this slope? So this is in the x direction, and that's the rate of change of E with x . And so it's dE/dx slope.

And I'm going to write it that way, as a partial derivative. And might as well use the correct notation right away. We're going to need partial derivatives because we have both x and y and t . And so we need to be clear about which type of differentiation we're talking about.

So then that's the slope. So that means that ΔE must be the slope times Δx . And so that's going to be-- and I'm going to write this as E_x .

So again, in order to simplify notation, I will often use subscripts to denote partial derivatives. And E_x and E_y -- we use so much that it would be a pain to have to write them out in full every time. As we indicated, those are the components of the brightness gradient, which is used for all sorts of things, including edge detection.

So now, let me divide through by Δt . And this, I guess in the limit, as I take smaller and smaller time steps, is going to be that partial derivative. And if you like, I can rewrite one more time.

So you can see here why using the switch to the continuous domain is helpful. Because I can take the limits here as the Δt approaches 0, and get the partial derivative. Of course in practice, I'm getting frames at a fixed rate, and they have a certain interval.

I can't make that interval infinitely small. So I end up with a much messier expression. But the two are obviously related. So that's a result for 1D image, and that allows us to recover the motion.

So one thing I probably glossed over is that as we go from t to $t + \Delta t$, the brightness decreases. So this ΔE over here is actually in-- for a positive slope, the ΔE will be negative. So that may have gotten lost in here somewhere.

So remarkably, from a single pixel I can get the velocity in this 1D case. Now importantly, that's not true in the 2D case. And after all, that's the case we want to solve. So we're going to have to work a little bit harder.

And now a couple of things you'll see-- first of all, the faster things are moving, the larger E_t will be. Well, that's no surprise. If the brightness is changing slowly, then you speed things up, then the brightness will change more rapidly.

It's like taking your YouTube video and changing the playback rate. When you change the playback rate, you change all the velocities by the same factor. Then you will have more rapid changes of brightness-- intuitively obvious.

Another thing you'll notice is that there's a problem if E_x is 0. We can't do this if E_x is 0. So if we're, for example, at-- suppose we are here. Well, then if the thing moves, the brightness doesn't change or changes only a tiny bit.

And so from that, you can't really tell how far it moved. You pretty much have the same result if it moved that much or whatever. And just in terms of implementation, you're dividing by 0, so that's not going to be good.

And it's not just 0. It's when it's very small. Why is that? Well, because when it's very small, it probably means that you don't know it accurately, as a result of the fact that you obtain it by subtracting two pixel values.

So let's just again go over this. We're approximating this by saying it's $1/\Delta x \cdot E$ of x plus Δx . Well, that's fancy notation, but basically, there are two pixels, we take this value, and we subtract that value. So we read out the gray level there, read out the gray value there, subtract them. And that's how we estimate our brightness derivatives.

And similarly-- so what do we do with E_t ? Well, E_t is derived the same way, except that we have two frames. And we pick the same pixel out of the two frames, and we subtract their gray values. So it's just in a different direction that we're approximating the derivative.

And why is that important? Well, if there is a small brightness gradient, E_x is small, then those two values will be very similar. And since they are not known precisely, they have measurement error in them, we are now subtracting two quantities that are similar in size, so we get a smaller quantity.

And it's mostly noise at some point. So that tells us that this is something to avoid. We don't want to be-- and it makes sense. I mean, if the image is uniform and brightness, x is 0, you can move it and you can't tell that it moved.

I mean, you'd have to have some reference point like some texture on it to tell that it moved. And if you had texture on it, E_x wouldn't be 0. So I know I'm belaboring this over and over again, but it's very important, because it means that this type of measurement is A, very noisy, and B, not trustworthy unless certain image conditions are satisfied.

And then the next thing you can do is say, well, we can estimate velocity from a single pixel. But we've got lots of pixels. And that's part of what we're going to be using heavily, that the single pixel result is not great, but we've got a million or 10 million.

And so we can do things like least squares to improve the result dramatically. So it may be that the result from a single pixel is really flaky, but that's OK because we divide the error by 1,000 if we think about the statistics of having a million of these pixels. So I keep on saying they're noisy, and at the same time I'm saying we can do this. And the reason we can do it is because we've got lots of them.

So we talked about how to approximate how to switch back and forth between the continuous world and the discrete world. And now, I don't want to give away the solution to the homework problem. But the next step, I guess, would be to get a bunch of these and combine them.

So we might do something like this. If we have n pixels, then we might try something like this. So now, instead of just using two pixels, we use those two pixels to do the calculation, then we use these two pixels to do the calculation, then we use those two pixels to do the calculation, and so on.

And so we add them up. So we have a lot of noisy measurements, and we take the average. And magically, things improve.

And so just without going into hairy statistics or assumptions about probability distributions, roughly speaking, when you average n values, you reduce the standard deviation by the square root of n . And when you have these political surveys and they say they have a 5% margin of error, what they mean is that they talked to 400 people. Because the square root of 400 is 20. $1/20$ is 0.05-- so same principle.

So it doesn't improve linearly with n , unfortunately. That would be even better. But it does improve with the square root of n . So if we have a million pixel image, it improves by a factor of 1,000, which is pretty significant.

Now, if we just did this in the simple way, we'd still run into trouble. First of all, if E_x is really 0, you'd be dividing by 0. So you can't do that. I guess you could leave those pixels out.

And then if E_x is small, you still have an answer which is relatively bad. So there are places where the slope is large, where you get good information. And now, you're polluting it by adding in as an equal information from places where the slope is low.

And that's why in the homework problem, we talk about weighting. So don't just take an average and treat each of them equally. But multiply by some weight factor. And then, of course, what happens to the $1/n$? Well, you have to compensate for the fact that you've now switched to using weights.

As we mentioned, images are really 2D. So let's extend this analysis to 2D, and see what happens. Now, some of this may seem tedious and repetitive to some of you, but we're all on different wavelengths.

So I'm going to get the next result in several different ways. And different approaches may appeal to different people. So let me do it this way. So let's start by talking about the image volume.

So think about video, and we're stitching together the frames. So each of these cross-sections is a frame. And we end up with this three-dimensional thing which is brightness as a function of x , y , and t .

And typically in practice, we slice it across this way. But we could slice the volume any way you like. And in some cases, there are advantages to slicing it in different ways.

So that's part 1. And sometimes it's useful to visualize things this way. Then, we're going to be using partial derivatives.

So what are they? Well, it's just the derivatives in the axis direction here. So we'll be dealing with dE/dx , dE/dy , and dE/dt .

And just as we've been discussing, we approximate them by taking differences-- first differences of neighboring pixels in either the x , the y , or the t direction from frame to frame. So that's all that is. So some people find partial derivatives a little bit more scary than ordinary derivatives, but that's all it is-- we're just taking the derivatives in all three directions.

Whereas we could be doing something more complicated. So let's think about that. So suppose that this video is from me moving through an environment, and there's some object in the environment that I'm tracking.

And at time t_0 , it's there in the image. And then in the next frame, it's over here, and so on. So it follows some path in this three-dimensional volume. And I could do this for other points, and develop this whole set of curves. But let's just focus on one.

Now, one of the things I might want to do is see how this changes. So when I'm looking at different frames of an image, often what I want to do is not take the derivative at a pixel and see how changes to the next time frame, but I want to follow an object and see how it changes. And in many cases, I'm going to make the constant brightness assumption, that it's not changing in brightness.

And so I would like to express some constraint on the derivative along this curve. And so I could have a curve. How do I define that?

Well, one way is to give x and y as a function of t . So this green curve-- for each time t , I can give an x and a y . That's how I define that curve.

And then, what I'd like to do is look at, for example, the total derivative along that curve, keeping in mind that along that curve, not only is t changing, but x and y are changing in some defined way. So this is the total derivative. And in what we're going to be doing, we'll set that to 0 because we're assuming that this point maintains its brightness.

And then I can use the chain rule to split this up. So I'm going to get dx/dt times dE/dx plus-- so I can take this total derivative and express it in terms of partial derivatives. And again, that looks kind of intimidating, but I can easily rewrite this.

So that's equivalent to what we had for 1D motion. So this is showing a relationship between the brightness gradient and the image motion. And if I know the image motion, and I know the image, I can predict how it's going to change.

And it's very simple. I just use this formula. And what we're more interested in doing is we've got images, image sequences, and we want to find u and v .

So that's where we're going. Now over here, we could just solve for u , because we had a single equation and a single unknown. And that's another thing we're going to do a lot, is equation counting-- how many degrees of freedom? How many numbers are there that we don't know? And how many constraints do we have?

Over here, we have one constraint, that equation. We have one unknown, u -- perfect match. So we know that we are likely to get a finite number of solutions. And because it's linear, we get one solution.

But when you look at this, you'll see there's one equation, one constraint, but we've got two unknowns. We've got u and v . So if we're trying to recover the optical flow, which is that vector field that we're discussing, it's like we don't have enough information here to do that, which is dramatically different from the 1D case, where we had the match between the number of equations and the number of unknowns.

So what do we know? So are we lost at sea? Is it hopeless? So let's look at just what that constraint provides us.

So again, the overall objective here is we have a time-varying image, or in the discrete case, a sequence of image frames. And we're trying to recover the motion. And we find that assuming that things don't change in brightness, the images don't change as they move, we end up with that constraint equation.

And what does it tell us? Well, one way to think about it is to plot it in velocity space. So we're used to plotting images with x and y as coordinate axes, but actually for some purposes, it's useful to have a different kind of representation. And this is one.

And what it means is that any point in here is a particular velocity, that velocity. And for example, this is 0 velocity. It's not going anywhere.

Velocity space was used at one time in physics much more than it is now. And it's actually kind of neat. For example, if you look at planetary orbits, they're ellipses with the Sun at one focus. And ellipses are complicated.

If you plotted them in velocity space, they're circles, which was exploited in the early days before people could just feed equations into a computer and have them solved, where they had to reason about things geometrically. Anyway, velocity space is neat. And here we have a constraint in velocity space.

So this is saying that before we opened our eyes and looked at the image, we didn't know anything. The velocity could be anywhere in this plane. But now we have one constraint on it.

So what does that tell us? That must somehow limit the solution space. And yes, it's a linear equation in u and v , and so a linear equation in the 2D world corresponds to what?

STUDENT: [INAUDIBLE]

BERTHOLD HORN: Line, yes-- y equals mx plus C , for example. So it's going to be a line. And that's great, because it means that we haven't solved the problem, but we've come a long way.

So before, it could be anywhere. Now, it's going to be on the line. What we really like is to pin it down to a point. So we've come partway.

And so what line is it? Well, we can rewrite that equation like that. And rewrite it a bit more by normalizing this vector. So I'm turning this vector into a unit vector.

So first of all, again, a reminder that this is the brightness gradient. And the brightness gradient is very important for all sorts of things. If it's 0, nothing much is happening because the brightness is constant.

And more likely, it has a high value at a transition between-- if I'm looking at a transition between the wall and the blackboard, there's a big change in brightness. And therefore, the derivatives will be large. And therefore, the brightness gradient will be large.

And not only that, but the brightness gradient as a vector will be pointing perpendicular to the transition. Because E_x is very large here, E_y is 0. So this brightness gradient is very important.

And this is a unit vector, as you can easily tell by taking the sum of squares of the two components. And why am I doing this? Well, because I'm interested in the component of u and v in this direction, the direction specified by the unit vector.

And it's this constant. And so what this is saying is that u and v are perpendicular to some line. And this is what it looks like. You can check that.

I mean, basically, you need to find this point, and then calculate the distance from the origin. And it's going to be that. So not too interested right now in the details of that, other than that we've shown that it's a line, and that the line depends on the brightness gradient.

So it tells us a lot of things already. One of them is that when you make this localized measurement and you are under constraint, you don't have enough equations, you do know something. For example, in this diagram, I don't know what u and v is, but suppose that I give you a different coordinate system.

Locally, you're going to see linear patterns. You won't see that. If I keep on magnifying this, it's going to be more and more like a linear relationship. And so the argument is that if you look through an aperture, you cannot determine the motion, you can determine the motion in the direction of the brightness gradient. So that's what we did over here.

So in the 1D case, we were done here. In the 2D case, we're not done because we have a mismatch of constraints and unknowns. And so we need more constraints. So we need to know something else.

And sometimes, there's prior knowledge of the environment that helps you. Sometimes, you can look again at another time and get extra information. Now, let's suppose that we have the optical mouse problem, where the whole image is moving as one.

You're over a flat surface at a constant distance from the lens. And to a very high degree of approximation, the image is just moving as one. So we don't just have one pixel. We can do the same thing at two pixels.

That's at pixel 1. And that's at pixel 2. And our job is to recover u and v . Well, it's two linear equations.

We can write them that way, and then solve. Now, because we've got two constraints now, we have enough constraint to solve for the two unknowns. And it's just a linear equation, so it's very simple. We can actually write the answer out explicitly.

So it's very mechanical. I mean, we just invert this 2 by 2 matrix, and multiply the result by that vector, and we're done. And what is this?

Well, this is the determinant of that matrix. So for 2 by 2 and 3 by 3, we can explicitly write it down. Otherwise, we'll just use Gaussian elimination to solve the set of linear equations.

So in a way, we're done. We need two pixels. We can solve this, and now it's noisy, so we can improve the result by taking more than two pixels.

But before we do that, it's always important to check the edge conditions. This can fail. And it fails when the determinant is 0. So when does that happen?

Well, can that happen? Well, sure. Let's rewrite it one more time. So we have this nice linear method, that if we make these measurements at two pixels, we can solve for the motion.

But it won't work if this is the case. So what is that? Well, this tells you something about the brightness gradient. This is the direction of the brightness gradient, well, the tangent of it.

So what this is saying is that you have a problem if they're related. So this is where they don't have to be the same. I mean, obviously, if the brightness gradients are the same, you subtract those two quantities, you get 0, and things blow up. But they don't have to be the same. They just need to be proportional to each other.

We're just saying that the ratio of E_y to E_x is the same. And that makes sense, again, because that means that the brightness gradient is the same in the two places. That means that the constraint we get out of the equation is in the same direction. It's not providing new information.

Or go back to this diagram-- we now have two parallel lines. Well, if there's no measurement noise, they'll be on top of each other. And then you intersect them, and what do you get?

Well, the same line, so that's not very interesting. Or if you have noise, it'll be even worse because they won't even intersect. So this tells you a number of things.

One of them is this isn't going to work if the brightness gradient is the same everywhere. And that makes perfect sense. That's the aperture problem.

It also tells you that maybe you need to somehow weight contributions from different image regions based on this. Maybe you don't want to have contributions from a lot of image regions with the same brightness gradient because they're not really providing good constraint. They all doing the same thing.

So that tells you that rather than take this result for two pixels, and then apply it to other pairs of pixels, and just take the average, you'll want to weight it in some way. And that's kind of kludgy. I mean, that's trying to fix a problem in the simple way.

But we can do we can do this much better. So we have this equation that's supposedly true everywhere in the image. This is our magic equation, which relates image motion to brightness gradients.

And that's supposed to be 0. Now, if there's measurement noise, it won't be 0. Hopefully, it will be small.

If you plug in the wrong values of u and v , it won't be 0. So maybe one way of solving this problem is to search for u and v that makes that 0. Or if you can't make it 0, make it as small as possible. And so that motivates this approach.

So what's going on here? Well, the integrand is just this expression. And if everything was perfect and you have the correct values of u and v , the integrand would be 0, and you integrate it over the whole image, you get 0. And that's the smallest you can get because it's quadratic.

You can't get less than 0, so that's it. If you plug in the wrong values of u and v , well, it will not be 0. And so you can base a strategy for finding u and v on that.

Now, there are cases where this is going to fail. One is if, I don't know, you're without a light in a coal mine. If E is 0, all hope is lost, obviously.

If E_x and E_y are 0, all hope is lost because that means you're looking at a wall that's constant brightness. And you won't be able to see it moving because there's nothing on it to catch your attention and to track. If there was some texture on it, then E_x and E_y would not be 0 in some places.

And so this is a sort of approach we're going to use a lot. So we have this constraint, which would apply in an ideal case where we know the answer u and v . And we're going to take contributions of that.

And now, we can't just add them, because some of them might be positive and some might be negative. So it doesn't make sense to minimize the integral of them. We turn them into something that's always positive by squaring it, and so we're going to find these answers by minimizing that.

Now in this case, this is pretty straightforward. We now take that, and this is a calculus problem. So this is a function of u and v , and we're going to set the derivatives with respect to u and v equal to 0.

And in this case, we're lucky enough that the equations are linear. And we can solve it. And actually, the answer is very similar to this. We're going to end up with two equations and two unknowns, one equation from the derivative with respect to u of 0 and one with the derivative of v as 0.

And so two equations, two unknowns-- cool, but that can fail because they might not actually be different equations. They might be linearly dependent. So we'd want to worry about that.

And in this case, the failure mode is-- that turns out to be the determinant of that 2 by 2 matrix. So that corresponds to this thing over here. And in this case, that's when things fail.

So when can that happen? Well, certainly if E is 0 or E is a constant because then E_x and E_y are 0. Or maybe if just E_x is 0 everywhere because then this integral is 0 and that integral is 0.

So let's see-- so if E_x is 0 and E_y is not 0, what is that? Well, that means that I have one of those pictures that only varies in one direction. E_x is 0, meaning it's constant in the horizontal direction.

E_y is nonzero. It's varying in brightness in the vertical direction. And if I were to draw isophotes, they'd look like this.

So here E_x is 0. E_y is not 0. And we know that there's a problem. If I move in this direction, the image doesn't change, so can't solve.

So E_x equals 0 is a problem. Similarly, E_y equals 0 is a problem, just turned 90 degrees. So those are kind of obvious. I mean, we know those aren't going to work.

Is there anything else? Well, how about E_x equals E_y ? Well, if that's the case, then these two integrals are the same, and actually, that integral is the same, also. So we're going to get integral e_x squared all squared minus this thing all squared-- 0.

So that's bad. So what is that? Well, that's where isophotes are running at 45 degrees because the brightness gradient has the same x component as y component.

The brightness gradient is perpendicular to the isophote. And so $E_x = E_y$ means it's a 45-degree angle for the gradient. So the isophote is 45 degrees down, so that's the picture for that.

And of course, that is really the same thing as that, just rotated. And one of the things we're going to do is say that, well, something shouldn't depend on our choice of x and y and the image coordinate system. I mean, the answer would be different, but in the new coordinate system, it should mean the same thing.

So in this case, I shouldn't get some result that's particular to the x - or y -axis. It should just give me the same result if I rotate. So that's a 45-degree rotation.

I can get any rotation by doing something like that. Because then E_y over E_x is some constant, and that's the tangent of the angle of rotation or something like that. Plug that in here, and of course, you'll see that you get k squared times the integral over here. And the k squared appears over there. They cancel each other out.

So this is actually the most general case. All of these others are just special cases of that one so that summarizes all of the cases we found. They're all with isophotes at some angle, parallel lines.

And then the important question is, is that it? It turns out yes, that's it. And it's not easy to prove. You need to at least use the triangle inequality or something profound like that. So it's actually not that hard, but I think it's not really useful to go there.

So when can we do it? Well, as soon as there's curvature in the isophotes. Because if the isophote looks like this, if I move it tangent to the isophote, there are still changes. So the only problem is where the isophotes are all straight lines, parallel straight lines.

And in another way of thinking about it, if you have areas in the image where there's a "corner," something where the isophote makes a sharp turn-- so here are the brightness gradients. If there's some small area where the directions of the brightness gradients change a lot, that's good. Because what we're worried about is that the brightness gradients are all parallel to each other. That's the bad situation.

And so for many purposes, like image alignment and recognition, we look for places that are "interesting." And one interesting way is to look at the isophote curvature or equivalently, the rapid turning of the brightness gradients. Because in other areas of the image, there's less constraint.

If I have a part of the image that looks like this, on its own, it doesn't provide enough constraint. Now, if next to it is another area of the image where things look like this, if I have both of these, then that's fine. But if everything has the same gradient direction, that's not satisfactory.

So homework problem theoretically due on Thursday. And I know we haven't done a whole lot on this. So if you have problems with it, send me email, and I may say something that doesn't give away the answer but helps you. And we'll go from there.

What are we're going to do next is talk a little bit about this concept of noise gain. So the idea is that all this time in the back, we've been thinking about we only have pixels with eight bits, and they're noisy, and our calculations are going to be flaky unless this determinant is large, and so on. So there's a lot of attention to not just a formula for computing the answer, but making sure that the answer is actually meaningful, as opposed to you've taken two noisy numbers and divided them by each other, and the result is not very predictable.

So in noise gain, there's a very precise way of talking about it, which is saying if I make this much of a change in the image, what is the change in the result? So in this case, I'm trying to get the velocity of motion. How much is the velocity of motion going to change if I change the brightness gradient somewhere, I have the wrong measurement that's slightly off?

And of course, we want that to be not sensitive. We don't want it to be very sensitive to noise. And so we'll make that clear in terms of a one-dimensional transformation where we know the forward function, and we're trying to invert it, and we want to know when that inversion is sensible and when it's not. Oh, questions?