[SQUEAKING]

[RUSTLING]

[CLICKING]

BERTHOLD HORN: Let's start with some announcements. We've had three homework problems and we're ready for the first take home quiz. The rules, again, are that it'll count twice as much as the homework problem and there's no collaboration, unlike homework problems. So that's next week.

And then those of you in 6866, there's a proposal due on the 22nd and that's meant to be short and that's where you tell me what you'll be doing for the term project. And so the idea of the term project is to take some machine vision problem, preferably something we discussed in class, and implement it in some way, your choice. It could be, I don't know, Windows, MATLAB, Android, whatever.

It could also be something more theoretical. It could be some mathematical solution to some machine vision problem. Most people opt to do some kind of implementation and it's very flexible. I mean, if you find that you can use OpenCV to implement part of what you want to do, go ahead and do that.

But of course, then your contribution has to be on top of that-- not just using OpenCV, but doing something useful with it. And if you have problems coming up with ideas, send me email. What will happen with the proposal is I will take a look at them and if I can point you to sources that might be helpful in implementing that project, then I will do so. OK.

And we're just about to finish our discussion of how to extract stuff from image brightness and in particular, shape from shading. And it's a little bit abstract, a little bit mathematical, and we'll soon have a big change of pace when we start talking about industrial machine vision. And of course, we can't cover everything, and we'll take a different approach to covering it. So rather than use published papers or textbooks, we'll look at patents.

And part of the reason for that is that in our world, you publish papers. That's what you get credit for. In their world, you don't publish. That's what you get credit for. So when you do see what they're doing, it's in the patents where they're trying to cover themselves, protect themselves from somebody else using that same idea. So that'll be a big change of pace.

And in the process we'll learn a little bit about patent and patent language, since that's an important topic if you're an entrepreneur involved in a startup or something of that sort. So obviously that's going to be a little different from partial differential equations and some of you may be looking forward to that. But let's finish with the partial differential equations.

So where are we? Oh, OK. Well, for example, you could implement time to contact on your Android phone and in that case, you'd use Android Studio and I could supply you with a dummy project that's just a shell so that you don't have to write all of those files for Android Studio. So that's an example.

You could implement some of the subpixel methods that we'll talk about for edge detection and use whatever, MATLAB, whatever's convenient. So that's an example of a project. Another example, more theoretical, would be we've talked about shape from shading in the context of particular types of reflectance maps like Hapke and you could implement-- you could work out the details for a different type of reflectance map. There would be a more abstract mathematical project.

And I think what I'll do is I'll pull together some of those and maybe say something about them on the Stellar website. OK, so the first part of the term we were focusing mostly on image projection, perspective projection equation, and derivatives, motion, motion in the world, motion in the image, and then we switched to one thing that we can do with image brightness measurements, which is the other half of the image formation.

And in particular, we're looking at shape from shading. And so far we've solved the problem for a very particular case, which is the Hapke model of surface reflectance. And at the core of all of it is the image irradiance equation, which basically says that brightness at some point in the image is the reflectance map corresponding to that surface orientation.

So here we're focusing on the dependence of brightness on surface orientation. And as we mentioned, it depends on illumination, it depends on surface material-- that's where the reflectance map comes into it-- and it depends on the geometry, in particular, surface orientation. And it's a local thing, so that's good. It means that the brightness measurement at a particular point in the image typically depends on what's happening at the corresponding point on the object.

And the reflectance map was our way of summarizing the detailed reflecting properties which, in turn, atomically are reflected in the BRDF. So we had the biodirectional reflectance distribution function, but we pretty much built up on that to we have a somewhat easier to manage reflectance map. So the BRDF depends on four parameters. The reflectance map only depends on two, but it's built on top of the BRDF.

OK. So we looked at a particular case which is that of reflecting properties of the moon, the mare of the moon, and the other rocky planets and in that case, we found that we could solve this problem in a particular direction. So in the case of the moon, if we take the ecliptic plane, then that's the direction along which we can actually determine the surface orientation.

So we can integrate out in certain directions and we can't integrate at all in the direction of right angles and I mean, typically, we'd be looking at some small patch, but this gives us an idea of what directions we can perform this in and what not. OK. And what we ended up with is a set of equations that take us from point to point on the surface and first of all, x and y are varying in this way-- and I have to apologize.

I think maybe the last time I ended up with those reversed. I had the first one being qs and Ps, and this corresponds to that angle of rotation where we rotated one coordinate system to the other. And then what's the change in height? Well, according to chain rule, which looks intimidating, but it's just P times the derivative up here.

So we got-- well, just P times Ps or Ps plus qs q. OK, so this is the rule we can use to take a small step. We take a small step in the image based on that and that corresponds to a small step in height. And I forgot to mention it here, but we're now assuming orthographic projection.

So that's a point that's important which is that once we switch to dealing with brightness, we switch to orthographic projection. Why? Well, because it makes everything much easier. All of this can be done with perspective projection and it originally was, but the math gets messy.

And so the thing to remember is that pretending we have a telecentric lens-- we talked about telecentric lenses-- and using a orthographic projection is it corresponds to being very far away, as we are when we're looking at the moon and the light source is very far away, and we have a very small visual angle. And it simplifies things. It's just like Lambertian, you know? It's not that these methods are restricted to Lambertian-- well, in fact, we're talking about non-Lambertian here. It's just that you can do some interesting things if you make that assumption and then generalize from there.

OK, so that's the rule. And basically we have three ordinary differential equations that we're going to solve numerically, and we don't need very sophisticated methods like eighth order Runge-Kutta or something. We're just going to do forward Euler or in other words, if you have the slope and you have a step size, you just multiply the slope by the step size to see how much higher you've come.

So that that's the method we're going to use. And of course, it's not terribly accurate, but if we make the step small enough, it's good enough. And the measurements we're dealing with are themselves noisy, so it doesn't make sense to apply a method that's good to 12 decimal places when we start off with things based on image brightness measurements. OK.

Now, how do we employ this? Well, we need to tie it to brightness. So somehow brightness has to feed into this equation. Well, for the Hapke type surface, we have this kind of dependence-- or actually it was cosine theta i over cosine. And in our arrangement for orthographic projection, it's very convenient to have the viewer up along the z-axis. OK.

And then we can plug in terms of p and q. So that's in terms of angles and in terms of unit normals, but we would like to express it in terms of p and q. Then you remember that n dot s came out to be this thing. And let's see. We're going to take the square root of that and then we're going to divide it by n dot b, where v is the same as z, so that's n dot z.

So that's 1 over that and that conveniently cancels out so we'll end up with that. So this is our r of p and q and by the assumption here about the image irradiance equation, that's e of x and y. So we can write e of x and y is this thing. And you can see the term we're looking for is right in here, Ps P plus qs q, so we need to square the whole thing, get rid of the square root, and then multiply through by rs and subtract the 1. So we end up with--

OK. So for this particular surface, there's a direct relationship between the quantity we can measure, e, and the thing we need to continue the solution. Right? So what's rs? Well, rs is just dependent on the source position. It's a constant. It just avoids having to write that all the time.

So we just take the brightness, we square it, we multiply it by rs, we subtract 1, and there's the derivative in the z direction and that's it. We just march along from one point to the next, adjusting x, y, and z as we go. And at each point we-- do we know the surface orientation? Well, a bit of it. We know the slope in that direction. I mean, that's what we're exploiting. But as we indicated, we don't know anything about the slope in the other direction.

So no, we don't know the surface orientation based on this and we need something else to do that. Now since each of these profiles is going to be independent to actually get z as a function of x and y, a real description of the surface, we need to somehow have an initial condition for these differential equations. So x and y, well, we pick some point in the image to start, but what about z?

Well, under our assumption of this image formation model, there is no dependence on z. The dependence is on the slope of z, right? So actually if we moved this object in the z direction, its image wouldn't change. Well, under perspective projection it would change in size, but we are not dealing with perspective projection, we're dealing with orthographic projection and so its size doesn't change, so there's an ambiguity.

So for each of those curves that we're computing we need an initial condition, so actually, we need an initial curve. And so in 3D, how do we do that? Well, here's a way of defining a curve. We have some parameter that varies along the curve. Could be arc length or some arbitrary parameter, eta, and for each eta we give a position in space x, y, and z and that's a curve.

And so let's assume we have that initial curve, so some sort of curve like this. And then we can start at any point on that curve and integrate out those equations numerically, and there's our surface. And as we mentioned, we can actually go in both directions from the initial curve.

And so we end up with z of x,y or actually z of eta and psi because the way we've parameterized it is one parameter goes along the curve, the other parameter goes along the initial curve. So it's a surface in 3D and it takes two parameters to parameterize that. So that's pretty straightforward, I hope, that in that particular case, we have some very special properties.

One of them is that we can locally determine the slope in a particular direction and that means, of course, we can go in that direction and build up a curve. And that's not going to be true in the general case, so what do we do about the general case? So we'll still start off with image irradiance equation, which says that the brightness at a particular point in the image is dependent on the surface orientation at the corresponding point on the object. And we'll try and follow this model here.

So suppose we had some particular point, x, y, z, and then we take a small step. And in the image, let's suppose the step size is delta x, delta y, and for the moment we won't say which direction we're going, we'll just leave that unknown. And to construct the solution, what we need to know is, what's z? How is z going to change?

And so of course, we have that relationship. The change in z is dz dx times delta x plus dz dy times delta y and so we can calculate the change in height if we know p and q. And suppose we know p and q, then we're at a new point on the surface and we can repeat. We take a small step in x and y and now over here we kept on going in a certain direction.

In this case, we may need to choose a direction in some particular way, but we need to know p and q. OK, well, we can assume that we start off not only knowing x, y, and z, but also the surface orientation. So we could have x, y, and z and p and q, but then how do we update every step we need to update p and q? So here we have updates rules for x, y, and z. For x and y, we're the ones controlling the step and then z, the change in z, is given by this equation.

So well, we can use the same chain rule trick. We can say that delta p is p sub x delta x plus p sub y delta y and delta q is q sub x delta x plus q sub y delta y so that we can update p and q as we go along. So we're not only updating x, y, and z, but we're also updating p and q. So this is kind of interesting. Before, we had a curb in space. We were tracing out x, y, and z as we construct a solution.

Now we've got more because at every point we know p and q, which means we know the surface orientation. So what we're really constructing now is a strip. Well, not very elegant . And this is called a characteristic strip characteristic of that differential equation. And that means that we're carrying a long surface orientation so if I wanted to, I could erect surface normals as I go along. So that's obviously more information than just a curve.

So that's what we'll be doing. We'll update not just x, y, and z, but p and q, which we didn't have to do over here because of the particular properties of the Hapke type model. OK, but how do we do this? Well, in order to update we need to know Px, Py, qx, and qy. And we can write this another way in matrix form.

So there are two linear equations, two unknowns, we can write it with a 2 by 2 matrix. And so what are these r, rs, and t? So r is p sub x, which is really the second derivative of z. s is p sub y, which is 2 sub x, which is this one. So the quantities we need in order to use this update rule are the second partial derivatives of height, and those are interesting because they correspond to curvature.

So the first derivatives have to do with surface orientation and the second derivatives have to do with how quickly the orientation is changing and that, of course, is curvature. And for a 3D surface, curvature is a little bit more complicated than it is for, say, a curve in the plane, and you need three numbers to describe it. So for a curve in the plane, you can just give the radius of curvature or the inverse of that, which is called curvature-- just one number.

But for 3D surface it's a little bit more complicated, and you need this whole matrix of second order derivatives and it's called the Hessian matrix. And in here I assume that the order of differentiation doesn't matter, that z of x, y is z of y, x, and that will be true for some reasonable surface-- won't specify the exact conditions for that. And of course, you can construct pathological things that don't satisfy that, but those are mathematical curiosities rather than real surfaces that we'll meet in machine vision.

OK, so that's the curvature matrix. And so if we know the steps and we know the matrix, we can calculate the change in p and q and we can continue the solution. Well, that means that we should add rs and t to our menagerie of variables. So we're going to carry along x, y, and z, p and q, rs and t. Yeah, we can do that. Now do we update rs and t second derivatives? Well, we use the third derivatives.

So I think you can see where this is going. This is kind of continuing ad nauseum using higher and higher order derivatives and so that's probably not going to work. In fact, we end up with more unknowns. Here we've got-- before we didn't know p and q, two unknowns. Now we don't know rs and t, three unknowns. So it's not going in a good direction. But what's neat is that we haven't yet even used our image irradiance equation. We haven't looked at the image.

So far we're just playing with derivatives, so that's obviously a flaw in our reasoning here. We're only looking at the derivatives of z, we're not using image brightness measurements at all, so that doesn't make any sense. So let's see what we can do with the image irradiance equation and in particular, we're often interested in the brightness gradient so let's look at the brightness gradient. So which way do I want to write this?

Again, by the chain rule, we'll get the derivative r with respect to p times dp dx plus r with respect to q times dq dx. And of course, these are the very quantities that we've run into over here. So this is what we call r, this is s, that's s, and that's t. So that's an interesting analogy with this. We can write this in matrix vector form.

That's the same matrix. So that matrix is important and it makes sense. OK? If you have a surface with constant surface orientation, the image will be constant brightness in this model where brightness depends only on surface orientation. If we are looking for a gradient, we're looking for changes in brightness and those are only going to happen if there's changes in surface orientation. Changes in surface orientation correspond to curvature, right?

So at one place I'm going downhill and then it's flat and then it's going uphill. Second derivative is non-zero, and exactly the second derivative matrix controls that. That's the precise statement of how that all works. Well, if we look at these two things juxtaposed, there's that common matrix h.

Now, if we could somehow figure out what h was, we could just plug it in here and we'd be golden. We can implement this method because we take a small step in the image delta x, delta y, multiply by this matrix h, and out comes the small change in p and q So we can have updated rules for x, y, z, and p and we're done. So I guess the question is, how do you solve this thing for h?

So what have we got? So this we can get from the image, the brightness gradient. So that's available. And then this we get from our reflectance model, so this is from the reflectance map. Assuming we know p and q-- and we said we're carrying along x, y, z, p and q, and so if we have a model of how the surface reflects light, we have a reflectance map, we can just take the gradient in the reflectance map, which is r sub p r sub q.

So we have this vector and we have that vector. Can we solve for h? So one of the things we use a lot is equation counting and constraint counting, unknown counting. So what have we got? Well, these are two equations, two linear equations. So two equations, how many unknowns? Three, right? We got rs and t.

So no, we can't do that. That's too bad. We had a very good thing going there, because here we can get this from the image, we can get that from the reflectance map, and if we could solve for h we could plug it in here and we'd be done. OK, so here's the whole trick of the method, which is that because h of p is in both of these equations, we can make some progress.

We won't be able to solve for h, but that's not really our aim. Our aim is to get an increment p and q. The only reason we want h is because this is our formula for computing the change in p and q. Well, maybe we can't solve for h, but maybe if we pick delta x delta y in a nice way we can use this formula. Right?

And so how would we pick it? Well, we'd want to pattern match these two things, right? So can you see what's going to happen? What's the direction that we're going to go? What delta x and delta y would you use so that we can actually compute delta p delta q using that formula? So pattern match.

OK, how about this equals that? Right? So let's try. And just so we can control the step size, let's multiply it by some small quantity. So that's it. I mean, before Hapke we had a direction that we had to pick and remember, we couldn't compute the profile in any other direction. The direction was given.

But what was special about Hapke was that direction was the same everywhere, so it was built into the whole solution. Well, now maybe the direction is going to change as we explore the surface. OK. So what happens if we try this? Well, then we plug that into that equation and we get this and we get--

So again, there's a particular direction that we can make progress in and that's this direction. And if we go in that direction, we can figure out how to change p and q and that's it. We're done. So if we want to summarize all of this, we have-- right? That's just from here. And then we've got-- let's leave out dz for the moment. dp, d-- and of course, we're interested in z so we need to write that one as well.

So what we've got is five ordinary differential equations and they're particularly simple first order equations. And so we explore the surface along these curves and actually along these strips, and those are the equations that generate that strip. So it's very simple. As we go along, we have the image brightness, we look at the brightness gradient, and that's going to tell us how to update p and q.

Since we're carrying along p and q, we know where we are in the reflectance map so we can compute r sub p, r sub q. That tells us the step to take, the update in x and y. And then, well, there's also this output rule, so to speak, which tells us how much the height is changing. And that's just based on p delta x plus q delta y, same old formula we've used all along.

I separate this equation from the rest because the other is our dynamic system where the first two feed into the second two and the second two feed into the first two. So if you're thinking control theory and stability and stuff like that, that's the interesting part, that they're these two systems that are feeding into each other.

And it's kind of weird, but what happens is that in the image space and the gradient space, we have this way of going in gradient directions. And so let's plot the isophotes, just some random isophotes, in those two spaces. So I don't know, this one here could be Lambertian. I don't know, this is some-- whatever that is. These are the isophotes in the image.

And what this is saying if we're at a particular point-- suppose we're here in x and y and we're also carrying along p and q, so we're also somewhere in-- let's suppose we're here. Then the step we take is based on the gradient, which is perpendicular to the isolines. And so the step we take here though, weirdly, is dependent on the gradient there.

So this is the actual step we take and then the step we take in p and q is strangely dependent on the gradient there. So we actually take a step in that direction. So it's a little weird. You're not going uphill. You're not just doing gradient ascent or gradient descent, but you're going in the gradient in the other diagram.

Anyway, this makes it clear how to implement that. You just have to have these two things, the image e of x and y and the reflectance map r of p and q. And once you plunk down somewhere in there, you just follow this rule and it'll trace out a curve in the image.

And indirectly, it'll trace out a curve in 3D and actually, it'll trace out a whole strip because all along we know the surface orientation. And that's a little different from the Hapke case where we don't know p and q as we go along. We only know one component in a certain direction. OK.

So we've reduced our image irradiance equation to those simple ordinary differential coupled-- ordinary differential equations. And this is a partial differential equation. Why is that? Well, because p is dz dx and q is dz dy and we're just making things look less intimidating by using these abbreviations p and q, but this is really a first order nonlinear partial differential equation.

And in physics you run into loads of partial differential equations, but they're generally second order. Those are the ones of interest, heat flow, wave propagation. So they're typically second order and they're typically linear and here we got something unusual. We have first order, which you think should be simpler, and it's a nonlinear.

And so if it wasn't for that, I wouldn't have to explain all of this because you would have learned this in physics. But physics does second order linear PVEs and not first order nonlinear PVEs, so we've just come up with a method for solving those and that's what we need to do in shape from shading since the brightness depends on the first derivative.

OK. Now this is general for any r of p and q. Let's just take a look at it for some particular surface properties that we've been studying. So one of them, of course, is Hapke, and that's a special case we solved up there. But let's just see how the general case reduces if we assume this for the reflectance map. OK, so here.

So that's a reflectance map for Hapke. So what do we need? We need r sub t, so we differentiate this with respect to p. As square roots, are we going to get 1/2 divided by the square root? Let's take out the 1 over square root of rs first. It's just a constant. And then the rest is going to be 1 over-- right? Because we have something raised to the 1/2 power so you differentiate that, you get 1/2 times that thing to the minus 1/2 power.

And then we have to differentiate what's this term inside with respect to p and we get Ps. And of course, r sub q is very similar. And then the other one we need is p r sub p plus q r sub q, that's just going to be the same thing.

Now these three share this multiplier and that multiplier, as we mentioned last time, is really just controlling how fast you go along the curve as you solve it. So you could change that. And I mean, it would change numerical stability and how accurate the solution is, but in the infinitesimal case it wouldn't change the solution.

So actually, we could remove these three terms as long as we do it on all three equations. And then we have Rp is proportional to Ps and Rq is proportional to qs. And so our update rules-- the update rule for x is just Ps. the update rule for y is qs, and that's what we had had up there. So the general case reduces down to this pretty easily, particularly since, I guess, this is, let's see, rs e squared minus 1, as we showed up there somewhere.

OK, so that's good. The general case reduces correctly to that special case that we solved first. Let's look at some other case. So we said that in the scanning electron microscope, we had dependence of slope. Right? Remember that unless you do something strange to your microscope, it's rotationally symmetric in imaging.

And so if you look at a reflectance map for that instrument the brightness is only dependent on the slope, the magnitude of the gradient, not the gradient direction. So OK, so you know, what's if? Well, that depends on the instrument and also the material of the object. And so you need to calibrate that. But let's leave it general. Let's just leave it as if. OK, then to use this method we need r sub p and r sub q.

We differentiate with respect to p and we get that. And differentiate it with respect to q, we get that. And so this is going to tell us our update for x and this is going to tell us update for y and we also need pr p plus qr q, which is going to be this constant times-- and this is an update for the height.

So we can certainly apply this method to scanning electron microscope images. And again, there's this constant multiplier here. We'll talk about this some more, but-- oh, not the whole thing-- but that only affects how fast we move along the solution, so we could actually simplify things by getting rid of that. And then the equations are very simple.

What is it telling us? It's telling us that the direction we're going is the gradient. We're going uphill, so the gradient is the direction of steepest descent. So if I'm standing on the mountain-- you know what the gradient is, so that's where we're going. All downhill. As we said, we can reverse the direction. We can make delta chi be negative and go in minus p minus q direction. So it's very simple.

And then here's the rule that tells us how much we're updating z. So scanning electron microscope is a little bit simpler than a Lambertian, but rather than solve this one separately after doing Hapke, we just went to the general case in general. OK.

And you can do the same for Lambertian. Unfortunately, it gets messy because the Lambertian has that square root and 1 plus p squared plus q squared. But of course, you can do it. OK, so a couple of things. One to remember is that we're dealing with a solution that generates characteristic strips. So we're not just exploring the surface along curves, but along the curve we also know the surface orientation.

And then another related concept is that of a base characteristic. OK? So the characteristic strip has x, y, z, p, and q along the strip and the base characteristic is just-- let's see-- the projection into the image plane. And to some extent, that's of interest to us because that's what we. We have the image, we're trying to explore it, and for one thing, we want to make sure that we cover much of the image with these curves.

And of course, the ultimate goal is the surface in 3D, but we're also quite interested in what happens in the image plane that we're actually covering that. OK. So what might this look, like these basic characteristics? So here's our image and for the moment, we're assuming we have some sort of initial curve and then these base characteristics go up from there and maybe the other direction.

So as I mentioned, one reason you might be interested in these basic characteristics is because you want to make sure that you're exploring as much as possible of the image and not leaving out some areas. And also you might say, well, this is no man's land. I should really be interpolating another one in here. And in some other areas, conversely the base characteristics might get close together and you might say, well, that's unreasonable.

I should really have pretty much the same height there, so just drop one of them or merge them, take their average. So in terms of implementing this, you'd be looking at these basic characteristics and interpolating and removing as required. Now another issue is this sounds very sequential, which is unpleasant from the point of view of implementation because it could take a long time to do this, and also unpleasant from the point of view of biological interpretation.

But it turns out that the solutions along these curves are independent. I mean, each of them satisfies a state of differential equations and the only way they interact is that, well, they all sprout from the initial curve. So actually you could have a process running along each of these curves, so it is parallelizable.

And that's actually kind of implied by what I said a minute ago because if you're going to interpolate new characteristics, it's best that you do it as you grow and say, oh, wait, these two are getting too far apart so let me interpolate a new one there or if they get too close together, let me merge them. So it's not full parallelism. It's not like you can do something at every pixel at the same time, but it's a significant improvement over complete serial computation.

So it's like a wavefront that's propagating outward. So if we have some initial conditions, you can imagine that as the solutions progress we could keep them moving at more or less the same speed and then look at neighboring ones in order to improve and interpolate and what have you.

So that means that they ought to move at similar speeds, so that gets us to this question of speed. And I mean, in terms of the numerical solution of these equations, it's just the step size, you know? What step size? Well, clearly if the step size is a hundredth of a pixel, that's overkill. That's not going to work very well because the brightness doesn't change much in the hundredth of a pixel.

Conversely if the step size is a hundred pixels, that's probably completely wrong because you're missing all of the in-between brightness variations, so you'd like to have a reasonable step size. And so let's look at what we can do in terms of controlling the step size. And as I mentioned, all we need to do, really, is multiply all of our equations by the same quantity and all it does is it changes the increment. And so let's look at some simple cases. So constant step size in z.

So that's interesting because that means that you're stepping from contour to contour. Think of a contour map. If we implement that, then these would be contours of constant height on the surface and what we're doing is all of these solutions, as they grow, are going from the contour at a thousand meters to the contour at 990 meters to the contour at 980 meters and so on. And so that's an interesting and useful way of controlling the step size. And what do we need to do that?

Well, we've got pRp plus qRq in the equation for z, which just disappeared, And so we just divide by that. Why is that? Well, because then dz d-- let's call it side dash-- is one. Right? So we had this equation over here, dz d psi is pRp plus qRq.

Now if we just divide by that then the rate of change is-- the derivative is one and that means that we have constant increments in the z direction. Of course, we have to divide all of the other equations by the same factor. So that's an easy to visualize change that has potential benefits.

We just multiply all of the-- divide all of the equations by that, and then we're stepping from contour to contour as we explore the surface and that makes it a little clearer how we're exploring the surface. OK. But we can pick something else. For example, we were talking about steps in terms of pixels. So secondly, we can look at constant step size in image.

So we want delta x squared plus delta y squared to be a constant, and those are proportional to Rp and Rq. So we divide by-- I shouldn't have wiped out these equations yet because it's handy to have them at this point. dx, d-- So that assures that square root of delta x squared plus delta y squared is going to be constant if you make it 1. Oh.

OK, so that's another way where instead of moving in constant increments in height, the intervals in the image are fixed in size. And well, a couple of issues with that. One of them is that those curves may run at different rates so that one of the curves is getting ahead of the other because we're not tying them together in height or anything, we're only tying them together in how far are we from where we started. So that's a problem.

And then another problem is we're going to divide by that. Of course, if that's zero then we're out to lunch. And I make a note of that now because we'll need that in a minute. OK, constant size steps in the image. How about constant size steps in 3D?

So that means we want that to be 1 and so that means we need to divide by that quantity. And so for that, where's that come from? That's this thing here. So this gives us delta x, this gives us delta y, this gives us delta z. And if we want the sum of the squares of those to be one, then we divide by that.

And again, this has the same problem or spatial case that if Rp and Rq are zero, then that is zero and so on. So how about if we step in isophotes in contours in the image-- contours of brightness. So here we had contours on the surface. z was the constant along each of those curves.

But it might be interesting to step in the image from one brightness level to another. So well, I won't go too much into detail of that, but we basically then have to divide by that quantity. Remember the two gradients, the one in the image and the other one in the reflectance map?

Well, that's the dot product of those two. I don't know if that means anything, but it's just interesting to note. And we won't go into too much detail, but obviously that's another interesting speed control in that we're moving from contour to contour in the image and one advantage of some of these is that they tend to make it easier to tie together neighboring solutions. So in this case, these curves, these wavefronts, would be just isophotes in the image plane.

OK. And in terms of the numerical analysis aspect of it, again, we're not doing any fancy methods for solving ordinary differential equations. We're just saying if the slope is m and we take a step delta, then the change is m times delta which is the lowest order, crudest thing you can do. But as I mentioned, we don't really expect to get much better by using something very sophisticated.

You may have heard that there's results recently about three bodies. So you all know that if you have two bodies then they orbit in elliptical fashion and the ellipses are stable and all that good stuff from starting from Newton and Kepler and Copernicus and so on. If you have three bodies, chaos ensues. All sorts of things can happen, and mostly the orbits are not periodic.

And so even if you want to know-- suppose you live in a world with three suns and they're orbiting each other and suppose you want to know whether at some one of them will run into another and blow up your world, there's no periodicity so you can't use simple method. Anyway, there's a wonderful science fiction book called *The Three Body Problem* where people live in such a place.

And curiously, just recently, someone has actually put to good use one of these gigantic supercomputers. So you know nations compete with each other in various stupid ways and one of them is I can build a bigger computer than you. And so periodically the US has the biggest and then, I don't know, Japan, and then China. So I think at this point it might be China. Anyway, a lot of times then you ask, well, OK, they've got this fantastic computer.

What are they getting out of it? Well, you can do some things. You can do weather simulations better than anyone else. You can solve some quantum equations of more than one particle on it. Well, what he did, this person, was the three body problem. And so he was interested in finding periodic solutions.

And for a hundred years it's been known that there's some periodic solutions, but they were very special. Things go in particular figure eight patterns and asteroid orbits, but there's a very small number of these solutions known. I forget what. I don't know, six or something. Well, he used the supercomputer to find, I don't know, 68 and you know, it's amazing. It's fantastic. They're wonderful orbits.

And you might say, well, wait a minute, this is really something that should be done analytically because you can do the numerical simulation but how do you know that it's really, really periodic? Well, having the supercomputer, he is able to do things that mere mortals wouldn't normally do like approximate the Taylor series to a thousand terms, you know? You'd usually stop at two or three and maybe if you're on the computer you might do, I don't know, eight.

But he didn't stop at any particular point. He simply kept on adding until he didn't need to go any further. And the same with the calculations, the eight order Runge-Kutta, that's a very sophisticated method for solving ODs. Well, he used thousand order and in the process, he discovered these periodic orbits.

Anyway, where was I going with this? Well, the point is that there are very sophisticated methods for numerically solving equations and if you're trying to say, for example, figure out how long our solar system is stable, it's very difficult. You need to use much more sophisticated methods than we do.

And Gerald Sussman actually built a machine to do that, but because it's chaotic, you can't be sure. But he can say that nothing bad is going to happen in the next hundred million years, so you can rest assured that things will be fairly safe. Fortunately, we have a case where we don't need to have anything like that kind of numerical precision.

OK. We do need an initial curve though, so let's talk about-- which is a nuisance because the whole point is to explore the surface using optical machine vision methods, not go there with a measuring tape. And so having an initial curve is not desirable. I mean, it's better than actually having to measure the whole surface because you're only measuring one curve on it and then the rest is filled in using the image.

But actually here we have an even worse problem, which is we're carrying along not just x, y, and z, but we're carrying along orientation as well so we get these characteristic strips. So shouldn't it also be an initial strip? In other words, we're not just going to be forced to supply z, y, and z, but that makes it even worse. That means you have to measure that curve and also at every point on the curve, measure the orientation.

Well, fortunately, that's not necessary and there are two reasons for that. One is that on the initial curve, we have the image irradiance equation. So we have e of xy is r of pq. Or you're on the curve, you look in the image, what's the brightness there? Doesn't tell you the orientation, but it gives you one constraint on the orientation.

And if we look at our reflectance map, then it means we're on some curve. So we have one constraint. So it's not like it could be any p and q, it has to be one of those. But then the other thing is that we have this curve and we're told that that curve's actually in the surface.

So that means that if I differentiate dz, d eta should be d, dx, d eta plus q. Right? By the chain rule because p is dz dx and q is dz dy. And since someone magically gave me this initial curve, I can compute these derivatives, dx d eta, dy d eta, dz d eta and amazingly, this is a linear equation.

So what's my job? My job is to recover the unknowns p and q, and I have two equations, two unknowns, and so there we are. I can solve for p and q. Well, you might say this first equation is likely to be nonlinear-- you know, you look at Lambertian equation. But there's one linear equation. So then by Bézout's theorem, what matters is what order is this equation.

And if it's second order, that means you might have as many as two solutions, but two solutions is better than an infinite number of solutions. OK. So in practice, we don't really need an initial strip. We can get along with an initial curve because we can find the orientation using those two equations.

OK. But we'd really like to get rid of this initial curve business. It's really annoying. And so what do we do? Well, it'd be great if there were some special points on the object where we know the shape, orientation, something. So here is our prototypical object, the image of this prototypical object, and so question arises, are they some-- so in most places, we don't really know what the orientation is.

Like we go here and we measure brightness e, I don't know, 23, and we go to the reflectance map and we get a contour. There's a constraint, but we don't know what the orientation is. So are there any places here where you could tell me what the surface normal is? The edge. Right. So the thing I draw here, I guess the official word is occluding boundary. Sometimes the image version of it is called the silhouette.

Why? Well, because that's where the object curls around and the part over here is visible and then the part where it curled around is not visible and the terminator that separates them, I can draw a surface normal perpendicular to this curve and the surface normal at that point on the object will be parallel to that.

So what I'm saying is that if I go all along the occluding boundary I can construct a vector in the image plane and on the object, the corresponding surface normal will be parallel to that. So that's different from other places where I don't have local information on surface orientation. And of course, in perspective projection it's a little different, but we're talking about orthographic projection.

OK. So I could perhaps use those as starting conditions. I could start my solutions there. Well, the problem is that the slope is infinite there, right? If you think about approaching that edge, you fall off. dz dx and dz dy become infinite. The slope is infinite, so that's obviously going to be a problem if we try to somehow incorporate that in an equation. So what's interesting is that the ratio is known because the ratio just defines this direction.

And so that's a funny thing where p and q are infinite, whatever that means, but we know their ratio. But unfortunately, it turns out that we can't use that. We have these equations that tell us how p and q are changing as we take a step, but if the slope is infinite then that doesn't work. So the occluding boundary tells us something, but we can't start the solution there and we'll get back to using the recruiting boundary. So that's number one.

Now number two is if we look at-- imagine a beach ball painted white and the sun is behind you and you're looking at it. There'll be some spot on it that's brighter than any other spot and you can, from your knowledge of inversion surfaces, say right away what the surface orientation there is. Right? Right? Because its brightness is cosine of the incident angle.

The cosine doesn't get bigger than one and it does so for zero angle, so that's when the surface normal and the direction to the light source are the same. So that's kind of a special thing and so unique. I mean, it doesn't happen anywhere else. So let's see how to formalize this unique, global, isolated, extremum.

So if I go back to, for example, Lambertian surface, I have a reflectance map like this and here is my unique global isolated extremum. Right? So most brightness measurements don't tell me the orientation. If I measure this brightness, well, it could be any one of those. If I measure this brightness, it could be any one of those.

But if I measure that brightness, I have the surface orientation. So that's very special and these things are called stationary points. And why that? Well, because the places where the derivative is zero in the reflectance map. And we'll see there's another reason for calling them stationary points.

So maybe we can start the solution there and get rid of this problem about needing an initial curve. Well, if it's an extremum, that means that r sub p and r sub q are zero. Well, if it's smooth in the extremum.

And we could consider the case where we have some sort of nondifferentiable r of p and q, but let's not do that. Let's stay real. OK, fine. What's the problem with that? Well, the problem with that is that our five differential equations included these two. Right?

So at this particular point, suppose I put my solution, my solver down at that point in the image-- corresponding point in the image. It's not going to go anywhere because r sub p and r sub q are zero. And actually, also if we consider the image itself-- so that's the reflectance map and here's the image itself. Well, corresponding to that point in the reflectance map, suppose here's my beach ball. There's this point.

Well, that's an extremum in the image and so here, by the same argument, if e of x and y is smooth and that's supposed to be an extremum, then the derivatives there will be zero. And so what does that mean? Well, that means that dp d-- that these also don't change. Right? And since z is dependent on those, nothing changes. We're just stuck at that point. I mean, it would have been perhaps that, well, you had that point but p and q changed and after a while there will be a change in x and y.

But no, that doesn't work that way. Everything is zero there. So stationary points are very interesting because they give us local information about surface orientation, but they don't directly allow us to start the solution. We can go on with this. I said extremum rather than maximum because for Lambertian it's a maximum, but for the scanning electron microscope it's not. It's a minimum, right?

For the scanning electron microscope, we had reflectance map that looked like this and this was the magic point and there, the brightness is a minimum. Right? Remember the objects were the edges, the occluding boundaries were bright in the image, and the part facing you was dark? So in the case of the scanning electron microscope, we do also have stationary points but they correspond to minima rather than maxima. But anyway, OK.

So what to do? Well, if we can get away-- if we can get a little bit away from this point, then those conditions aren't going to be true anymore and those quantities may be small, but at least we can move. And of course we can control the speed. So suppose the quantities are small, big deal, we just multiply the step size. So as long as we can get away from that point, but how do we get away? So here's our stationary point. One thing we can think of doing is constructing an approximation of the surface. Let's suppose--

OK, so here's a story. We know the surface orientation there, of course, is one of those stationary points and now we want to get away from it a little bit so we can start the solution. So the idea is we want to start the solution from this curve. So we know the orientation so we can construct a small plane and I don't know, make a radius epsilon and then start the solutions there. Is that going to work?

Well, if it's a plane, then all parts of it have the same orientation, they all have the same brightness, and we're just exactly the same problem out here than we were there. So that idea doesn't quite work. So the answer is, well, let's have a curved surface. So we still have that special point, but now let's suppose that the surface is curved and we're going to construct a small curved shape around that point and we'll start the solution from there.

So this sounds kind of, I don't know, specialized, weird. Why these points and so on? But actually, these points are very important in human perception as well. So you can do experiments where you show someone a picture of a vase and they have a very good idea of what its shape is. I mean, not metrically accurate, but generally pretty good.

And then you Photoshop out the bright spot and they still have a shape in mind, but it's changed. And so actually, it turns out that we use these stationary points as well. Another example is where you have cut out-- so you have some blob in the world like this, but now you're showing someone a picture of only, I don't know, say that.

That doesn't include the bright point. It turns out that this is much more ambiguous that then if you included that bright point. So it's a real thing. It's not just something that affects our particular method of solution, it's important for there to be a unique solution or a small number of solutions as opposed to an infinite number of solutions.

OK. So this is going to be some sort of a curved surface and we need to find out what its curvature is in order to construct it. Right? And so it's like, Oh my God, now we not only need to guess at the surface, but we need to know the curvature of the surface. But actually, it turns out that's possible, and so let's see how that might work.

So the idea is that we are going to have a small patch, and I'm going to make this as simple as it can be. So we're going to assume, first of all, that we have an SEM type of reflectance map just to make it really simple, and then let's suppose we have a surface like this. And this will have a stationary point at the origin.

And let's see. So you got p is dz dx is 2x. q is dz dy is 4y. And then the reflectance map gives us p squared plus q squared is 4x squared plus 16y squared. And by the image irradiance equation, that's actually the image. OK? And I'm going to take the gradient of the image-- it'd be 8x-- and not surprisingly, the gradient is zero at the origin.

And that corresponds to it being an extremum, so that just confirms that we have, in fact, set it up so that we have an extremum at the origin. OK, now can I use the gradient to estimate the shape, local shape? Well, no, because the gradient is zero right at the origin. And so let's take the second derivative.

So the plan is the gradient will be zero, so that's useless. Brightness itself we've already used to determine that it's a stationary point, but if we take the second partial derivatives of brightness we get some information about the shape. Then we're going to try and recover the x squared plus 2y squared from this. So we might say, well, how can I measure the second derivative? Well, of course, just apply the first derivative twice. And we also already talked about convenient computational molecules for doing that, so there's one for exx.

So the plan will be we find the stationary point, we estimate the local shape by looking at the second-- not the gradient but the gradient of the gradient, so to speak-- and construct a small cap of that shape around the stationary point and then start the solutions from there. But I didn't quite get done with that, so we'll finish that next time. And then as I said, then we'll have a real big change of pace and we'll start talking about some industrial machine vision methods and the patents that describe them.