PROFESSOR: All right. Let's get started. So we are continuing the theme of unfolding polyhedra, and the general picture we are thinking about in terms of edge unfolding versus general unfolding which are these two pictures. Top is an edge unfolding of the cube. Bottom is a general unfolding of the cube.

If you have a complex polyhedron, we found general unfoldings. There are like four of them. They work. We proved one of them.

If you want an edge unfolding, that's the centuries-old open problem. For non-convex polyhedra, we know this is too much to hope for. Even if it's topologically convex, there's not always an edge unfolding. That was the tetrahedral Witch's Hat. But for general unfolding, we don't know.

So today's lecture is actually mostly about these two open problems and different variations of it that we know how to solve, special cases, and changes in the model. At the end, there'll also be some stuff about the reverse direction folding, but mostly, it will be about unfolding. So there's a third kind of unfolding which we call vertex unfolding, and it's kind of like a hinged dissection. So instead of normally in unfolding, something like a cross, it's a nice, connected polygon. The faces are joined along edges.

But what if I allowed disconnecting along edges, but I just wanted things to stay connected along vertices, like at a hinge. So it still should be one piece in the sense that this is still connected, and in this case, I still want an edge unfolding. You're only allowed to cut on edges. And in fact, we will cut on all the edges because that'll be the most flexible. Every edge gets cut, but you're going to leave intact certain vertices to make one, quote, "piece" at vertices. So we still want one piece.

So this is vertex unfolding, and it's always possible. We've even implemented this algorithm here. A bunch of random points on a sphere. Take the convex hull, and then, take a vertex unfolding. So you get this nice chain of triangles, don't intersect each other, and this will fold up into that 3D polyhedron on the left. Here's some bigger examples, hundreds of vertices. Amazing.

All right. So how do we do this? All right. Let me state a theorem. Every connected triangulated manifold-- this is a very general result. It actually holds in any dimension. We're going to think about two dimensional sources in 3D, but it could be D dimensional surfaces in D plus 1 dimensions. --has a vertex unfolding.

So this works both for convex and for non-convex. The only catch is that every face has to be a triangle. It's an open problem for something like a cube where you actually have quadrilateral faces.

So the way we prove this is to construct what we call a facet-path. This is a path that alternates between visiting faces, triangles, and vertices. So the idea is you have some-- say we're doing a tetrahedron or something.

You start at a facet. Then, you go to one of its vertices. Then, you go to one of the facets that shares that vertex. Then, you go to one of the vertices.

Then, you go to a facet. Then, you go to a vertex. Then, you go to a facet. This is a facet-path. It should visit every facet, every triangle, exactly once.

Vertices you can visit multiple times, although I think it's a bad idea to visit the same vertex twice in a row. Other than that, you can visit a vertex more than once. Maybe I would visit it again coming up here. If there was another triangle like this, I could go up to this vertex again and over to the triangle.

So the claim is facet-paths always exist for any triangulated, connected surface. Once you have this facet-path, you're basically golden because you can lay it out without overlap. And that's the next picture.

So if you have a triangle and you have some corners of it that are hinged to adjacent triangles, you can rotate that triangle 'til it fits in a vertical slab. And the hinges are on the ends of the slab. And so each triangle lives in its own slab. Slabs don't intersect. No intersection.

The hard part is really getting this path. Once you have that, you can just lay it out.

It's a little bit nontrivial with obtuse triangles. They don't necessarily just lie along the horizontal line. You have to set things up so that those guys are on the boundary, but you can always rotate it so that that's possible.

So the real question is, how do we construct a facet-path? All right. So I have a bunch of triangles. In general, they make some surface. And maybe I should draw a running example over here.

Let's think about a triangulated cube, something simple. Imagine there's more triangles in the back. It's a little hard to think about the 3D picture. I would really like to two dimensionalify it, and because this theorem works, not only for a polyhedron-- it works for any manifold, any sort of locally, two dimensional thing-- it would be nice if I could just sort of cut this open and think about a disc instead of a sphere, topologically.

And I can. I mean this theorem's supposed to work for discs just as well. It should work for anything that's connected. So ideally, I cut it all apart into lots of little triangles, but it has to stay connected.

So I'll unfold it. Why not? So I can add some cuts until I get down to a spanning tree of the faces, a spanning tree of the dual graph. OK. For whatever reason, this is the triangulation I chose, and this is the unfolding I chose. But just keep cutting edges until cutting an edge would cause it to disconnect.

So the maximal set of cuts, there are many ways to do it. When I unfold, it might overlap because we don't know whether edge unfoldings exist, but that's fine. You can think of it as a two dimensional picture, but it might need a few layers. We're not going to actually fold this thing, but we're going to cut this up into a facet-path.

This could actually be our input. Instead of being given a polyhedron, we're given a disc like this, a triangulated disc, and we have to deal with it. Somehow we've got to construct a facet-path here, visit every triangle exactly once, and passing through vertices.

AUDIENCE: So if you have overlap, it doesn't really matter.

PROFESSOR: Yeah, overlap doesn't really matter. We're going to cut it up, and then, we're going to splay out the triangles.

AUDIENCE: Distances don't matter. Right?

PROFESSOR: No, uh.

AUDIENCE: If you're trying to just find a path--

PROFESSOR: Right. From the perspective of facet-path, distances don't matter. It's just topology. Yeah. You could think of it as a circle with some decomposition into triangles, if you like, but that's maybe harder to think about. I got to think about it this way. It's useful because then, we'll get to Mickey Mouses, as we'll see.

So cut edges until you can't anymore, so until cutting would disconnect. So this means what you're left with will be sort of a tree of faces. There'll be no cycles because if there was a cycle, you could cut one of the edges, and it wouldn't fall apart. So obviously, there's a tree here.

Now, trees have leaves. That's our favorite lemma lately. They have at least two leaves.

In the case of a triangulated polygon, we usually call them ears. So here's a fun term. So let's say color the ears. These are leaves and in that tree which call the dual tree.

So in reality, we're thinking about a dual graph which looks something like this, where there's a vertex for every triangle and edges for every edge connecting triangles, and there's leaves of that tree. But in the original thing, we think of this triangle as being an ear, and this triangle is an ear. And this triangle is an ear, and this triangle is an ear. I like ears because they're kind of on the boundary, on the surface, so they're just triangles that are adjacent to only one other triangle.

Now, the next step is to color what are called the second-level ears. I'll call them the remaining ears, if you remove those ears, what would, then, become an ear. All right. Now, I wish I had another color. Yeah, that's kind of yellowish.

So this would become an ear. This would become an ear. This would become an ear, and this would become an ear. And I'm going to stop there, just two levels.

What could I get in this process? I mean what it looks like I'm getting is I get an ear and then, a second-level ear. It could be a little more general than that.

Maybe, for example, if I had a triangle like this, both of these would be first-level ears, and then, this would become a second-level ear. Turns out that's all that can happen. Because these are triangles, second-level ear, at this point, it's only adjacent to one other thing.

So what was it adjacent to before? Well, maybe, one ear, certainly not zero because it wasn't a first-level ear. So at least one ear, maybe two ears. That's it.

AUDIENCE: Unless there's only four pieces left. Right?

PROFESSOR: Sorry, what do you mean?

AUDIENCE: You could have three ears.

PROFESSOR: You could have three years. Yeah, so here I have two. I mean the first-level ear and a second-level, or it could be a first-level-- Oh, you're saying at the end. Right. It could be like this. That would only happen at the very end. But yeah, this could be first-level ears, and this is the second-level ear. Good point.

Most of the time, we will either get something like this, the rest of the polygons over here-- so this is a first-level ear. This is a second-level ear. --or I get-- this is the Mickey Mouse picture. Probably not allowed to say Mickey Mouse. It's under copyright, but there you go.

So two first-level ears, second-level ear. Boom. This is, most of the time, what you'll get in the base case of this induction. I'm going to pluck off these ears and keep making the thing smaller. At the end, there are a few cases to think about. I have them drawn here somewhere, base cases. Nothing. Yeah, well.

This won't work out if you have one triangle, if you have zero triangles, which is this empty picture, or if you just have two maybe. Well, I guess-- Yeah, because then, these are both first-level ears. It doesn't look quite the same. Or maybe just a Mickey Mouse because those are all-- Well, it probably works. Anyway, these are the sort of cases you worry about.

But for these cases, I just need to check that I can find a facet-path. So for example, this one, I just visit the triangle. This one-- I don't know-- I do it like that. In fact, I can make it a cycle if I want to go crazy. This one-- I don't know-- something like this. That'd be one way to do it.

(SINGING) Doo, doo, doo. Hm. Got to be a little careful. [INAUDIBLE]. Something like that.

All right. But really, I care about these two cases. So what I'm going to do for each of those-- I guess this is still step four-- is I actually want to make cycles for these guys. I care about that. Am I doing something wrong? Oh, no. Over there. Good.

So in both of these cases, I can make cycles. In this one, in the base cases, not always. Like this guy, hard to make a cycle. But for the two general cases, if I find two ears or three ears, I can just draw that in. So I'll do that for this example, though, by now, it's gotten a little messy, so let me redraw it.

So we have-- All right. So I'm going to connect these guys and connect these guys. This case, I only get the two-ear case, but I think we'll get another case shortly. OK? So obviously, I haven't finished it, and these are disconnected. There's lots of things left to do.

But then, the idea is repeat. So imagine those guys as being done. I'm left with these four triangles, actually, a little boring because I don't get the Mickey Mouse case. But then, this will be an ear. This will be a second-level ear. And so I'll end up doing this. And this will be an ear, and this will be an ear, second-level here. I mean this is actually the base case, if you will.

So in general, I just pluck off two or three triangles, repeat until I get one of the base cases. Now, what I have is not connected, but it's a bunch of cycles. But there's one cycle here, one cycle there, one cycle there.

You could actually connect some of them together, like these two. You could go around like this and then, go like that, so that could be a bigger cycle. But these cycles are not even attached to these cycles, so it's kind of a problem.

That's step five is we're going to fix all the problems. That'd be a great title. Connect cycles together. We're going to do that by local switches, so here's the general picture.

Suppose you have two adjacent triangles and two completely separate paths. So it's a cycle, so some cycle over here. And if this guy shared a vertex, then I could actually connect them together. So suppose it doesn't share a vertex. That means it has to use these two, and then, it-- something like that.

What I want to do is remove this edge and remove this edge and, instead, add this edge and that edge. So that's a local change, and now, it will be one big cycle. We've probably seen this trick once or twice before, I think in the Mountain Valley assignment stuff.

So here I have, for example, these two triangles, which are adjacent, but the paths don't meet. So I'm just going to erase this edge, put in that one, erase this edge, put in that one. Lo and behold, I have a bigger cycle now, like that.

Still not everything, so like these two triangles don't touch. So erase that edge. Erase that edge. Put in that one. Put in that one.

I'm preserving, at all times, that I'm a facet path, and I'm merging components. So by the end, I'll have one big component. Now, I haven't necessarily described this as a big cycle, but as I've kind of been hinting, what I do now is take an Euler tour around this thing. We've seen Euler tours.

All the vertices in this graph will have even degree. And so I can take an Euler tour, and really, I should take what's called a non-crossing Euler tour. Point is you've got a bunch of cycles that touch.

You just walk around the outside. That will visit all the cycles. It'll visit all these edges exactly once, and that will actually be a facet-path. And we're done.

Now, I have a facet-path for a triangulated cube. So I can splay out those triangles, get a non-overlapping chain like this. Actually, it's probably the top one, something like that, might be. I may not have matched exactly what's in the textbook. Cool.

One slight detail is I was sort of waving my hands, assuming there was actually a cycle here. I really only need a path. In fact, there won't always be a cycle because at the very end, you're not able to construct a cycle.

So that will actually cause you to make-- if I do something like this, so at least it can connect to other things. That will cause you make two vertices of odd degree. But that's OK because there's still an Euler path that starts at one of the vertices of odd degree, visits all the other edges, and then, comes to the other vertex of odd degree. And I just need a path.

You can actually characterize when you get a cycle. It's when the original thing is not too colorable, I think. Anyway, that's vertex unfolding. It's kind of easy.

I think we solved most of it in like an afternoon. We had this idea. It was cool, and then, we solved it kind of quickly. That's how it often goes. Once you have a cool problem, it falls quickly.

There's lots of interesting open problems remaining about vertex unfolding, and they seem a lot harder. So for example, what if I have non-triangulated polyhedra? And natural version is to think about what we were originally trying to attack, convex polyhedra. This turned out to not require convexity.

But what about convex polyhedra, not triangulated? Is there always a vertex unfolding? We don't know about edge unfoldings. And the answer is no. Well, no that's not right. Sorry. The answer is we don't know.

What's annoying about this example is that there's no facet-path. So there are two things that could go wrong. One is that the facet-path doesn't exist, and that can happen in this more general scenario. And the other is that when you lay it out, it overlaps. Both of these things could go wrong in the general convex case.

So in this example, this is a truncated cube. There's eight triangles and only six octagons. And if you look, once you're at a triangle, you have to go to an octagon because there's no adjacent triangles.

So at best, you could alternate triangle-octagon-triangle-octagon if you want to pack all those triangles in, but you run out of octagons to get there. So there's no facet-path, but maybe, you don't need facet-path. You could make a tree, just a little trickier.

Obviously, if you triangulate that surface, you're done. So the analog of general unfolding with vertex unfolding is trivial. You just triangulate, and then, you use the edge case. But if you really only want to cut along edges, this example can be done. I mean it has an edge unfolding, so if it has an edge unfolding, it definitely has a vertex unfolding, but we don't know how to prove it.

The other thing that could go wrong is once you have octagons, even if you could find a way to lay them out-- that's not a very good octagon-- you can't-- let's say if your two hinges were here and here, if that's where you attach two adjacent pieces, you can't fit that in a vertical strip. There's no way to turn it so it fits in a vertical strip. So also this layout problem doesn't work if you have something more than triangles. So some pretty fascinating questions.

It's even open for non-convex. If I take non-convex polyhedra, and I want a vertex unfolding-- let's say all the faces are convex. At the very least, you need to forbid faces having holes. If you remember this example, the box on a box, this also doesn't have a vertex unfolding because still, this guy has to fit in that little square hole.

But as long as the faces don't have holes-- let's say the faces are convex-- would be a nice version, like the Witch's Hat, the spiked tetrahedron. That probably has a vertex unfolding.

AUDIENCE: If you triangulate that, [INAUDIBLE].

PROFESSOR: If you triangulate this, it'll definitely have a-- if you triangulate anything, it'll have a vertex unfolding. Yeah. All right. That's all I want to say about vertex unfolding, and that's sort of addressing edge unfolding of convex polyhedra, or actually, both of these.

Now, I'd like to go to sort of the real problem. This is one of my favorite problems, I think. A bunch of us posed it in '98, which was right when I was-- or '99, right when I was starting out.

See, it's tantalizing, in some ways more natural, because it's nice to allow cuts anywhere. And it could potentially work for everything. I conjecture every non-convex polyhedron without boundary can be generally unfolded, and there's some really good evidence for this as of a couple years ago.

That's what I want to talk about, which is orthogonal polyhedra. This is one of my favorite unfolding results. It's by Mirela Damian, Robin Flatland, and Joe O'Rourke. They've done a lot of work in this unfolding area.

An orthogonal polyhedron is one where all the faces are perpendicular to one of the three coordinate axes. So for example, here is a orthogonal polyhedron I drew this morning. If you want to draw orthogonal polyhedra, Google SketchUp makes it really easy, and you can add shadows and texture.

So there are three kinds of faces. There's the ones perpendicular to x, like these guys. There's the ones perpendicular to y. That's all the yellow faces. And there's the ones perpendicular to z. That's these top guys. So we call them x-faces, y-faces, z-faces.

What we're going to do-- so the theorem is these have general unfoldings. That's pretty awesome because every polyhedron is approximately orthogonal if you can voxelize it. So this is really a lot of stuff. I would love to generalize this approach arbitrary polyhedra, but that's the big open question.

So what do we do? Well, we're going to single out, from this color coding, the y-faces. Just color them yellow. Then, there's all the other faces. Well, they form bands. They're cycles. They go around in a loop. A lot of them here, I've just drawn as rectangular loops, but in general, all those wooden faces, the x and z-faces, will form a bunch of loops. And then, there's the y-faces which we're going to have to deal with, but sort of ignore them for a while. I should have drawn it with them erased. It would look cool.

But if you think about just those bands, this is sort of how they're connected together, this tree. There's a big one out here, and then, it has two children. There's this one and this one. And then, this child has one child hanging off of it, and this one has two children hanging off of it. So that's these two guys.

In general, this guy might have some children hanging off the back side as well. I'm not going to try to represent that in this picture. There's just a bunch of children. There will be some front children and some back children. You pick some root arbitrarily, and then, you have children going off of there.

Now, if you're orthogonal polyhedron has genus 0-- it's topologically a sphere-- this will be a tree. If it's like a doughnut, it will have a cycle. So this theorem only applies for our genius zero. So we're going to exploit that that dual drawing, how the bands are connected together, is like a tree. So why don't I write down, a band is a cycle of x and z-faces, and they are connected together in a tree.

Now, the rough idea is we're going to take a depth-first traversal of this tree. We're going to start at the root, work our way down and come back, work our way down, and something like that. It's not going to be quite so simple.

The challenge, I guess you could say, is avoiding overlap. OK? If you wanted to unfold a band, obviously, a band can just unfold straight. It's like a nice, long strip.

So each band, individually, is fine. It's how do you piece those bands together and then, have room for the yellow faces to attach on the sides, no overlap?

But it can be done with the awesome, crazy idea that we'll get to shortly. It's going to start out kind of innocent, but the general approach is always proceed rightward in the unfolding. So the unfolding will look something like this. OK, whatever. Always going to the right. We start here, and we might go up and down, but we never go left.

And then, that's going to be all the band faces. All the band stuff will be connected like that, and then, there's going to be yellow faces that can just hang off the sides. So these are the y-faces. As long as I get the band to do this, y-faces can hang up and down. It's not going to intersect anybody. Pretty clear? So it's clear if we could do this, we can get non-selfintersection.

The amazing thing is that this is possible. What this is essentially a limitation on is how far you could turn your bearing. So you start going right. You can turn right, but I could not turn right again. I have to turn left next. I can actually do two left turns in a row, as long as I was initially going down.

I can turn left twice, then, I could alternate left-right. That's always good. Or I could turn right twice, but only if I'm initially going up. That's the rules. As long as I adhere to those rules, I'm fine.

Now, we're going to heavily exploit that we can do general unfolding, that we can subdivide those strips into lots of little pieces. We're going to subdivide into a lot of little pieces, an exponential number of pieces, so this is kind of hard core. So here is one example.

This is a leaf. So trees have leaves, and at the end, we're going to have to visit a leaf. So this is one box. There's this funny view, so you can see like a mirror on the bottom and a mirror on the right and on the side. So this is if you had one box, here's what you would do.

And we're actually assuming-- notice this side does not get covered. The idea is that site doesn't exist. That's attached to the rest of the polyhedron, so that's where our parent lives. And our parent tells us you have to start at s, and it says you better finish at t. And I want the property that if initially, I think, initially, you're going right. No, it looks like initially, you're going up. It matters.

You can do stuff, and then, at the end, you should still be facing up. So you have to visit all these faces, but not turn in total. And normally, that would be hard to do. If you just tried to visit one face at a time, you can't do that, but if you visit faces multiple times and kind of weave around in a clever way, you can do it.

So maybe I'll point with this thing. Yeah. So I start at s. I go up. I turn right. Now, I better turn left. I go down over here, up there. I turn left. Then, I turn right.

So if you follow along here, I just turn right here. So now, I go down here. And that is a left turn because it's on the bottom, a little hard to think about. So I turn left here, and then, I go, turn right. And then, I go down, turn left and then, right. This is confusing that I'm upside down. And I come to t, and lo and behold, I'm facing up again. In fact, I basically just zigzagged.

This would also work if it was rotated 90 degrees. I'd initially be going right and, then, down and right and down and right and down. So this can kind of go in a couple of different orientations. It's really powerful.

It also works if t is on the other side of s. You could do sort of the mirror image traversal. Now, obviously, I didn't cover the entire surface. I'm leaving room for later, but if this was all I was going to do, I would actually sort of fill out all those strips, just kind of extend them. It just makes this kind of fatter. So this got a little bigger. I've got the first half and then, the second half. This is really glued up there.

But you can imagine once you have these sort of paths that visit everything, you just fatten out, and it's no big deal. And also, in this case here, we're imagining-- oh, this is actually two of them. Fun. Two of these strips joined together.

And so there's a few side faces. They just attach up and down. They're not going to intersect anything because this is not actually below this. This is way over here. So that's the leaves, and I still haven't gotten to the exciting part.

So imagine you have a band. Just going to represent that by this big rectangle, and it has a bunch of children. Remember, it can have front children. This is down in the y-coordinate. And it could have back children up in the y-coordinate.

And suppose I'm actually attached to some parent, let's say, as a down neighbor, and I start at some s. Let's say I'm told you have to start here. You have to finish here, and you should preserve orientation because I don't want to have to think about whether it turned, don't want to have to depend on that.

So you can do it. Initially, you must be facing up because immediately I'm going to make two right turns. And I could handle two right turns, as long as the next thing I did was a left.

So I come into this thing saying, look, you're facing-- this is facing up. Now, you're facing down. You better turn left next, and by the end, I still want to be facing down. OK.

Now, I make two left turns. Now, I'm facing up again. I tell this guy, you better turn right next, and preserve. At the end, you should be facing up again. Now, I make two right turns. Now, I'm facing down, and so on and so on. And here, I'm wrapping around to here because this is actually a band that cycles around.

Then, I go in here. It's the same thing. It's just a little hard to see because I'm drawing it on a flat surface. But if it was on a ring, it would be much clearer. Just going left and right and left and right, alternating direction, so that I preserve my orientation.

At some point, I get to here. I loop around. I make a little wiggle at some point, and then, I can visit all the top neighbors. You just have to slightly switch your orientation, but again, preserve that you're doing left-left-right-right, left-left-right-right, left-left-right-right. Then, you will preserve your orientation. You tell each of the children which way you're initially going, and they can deal with it. It's basically telling you whether s and t is like this or the other way around.

OK? So now, we end up way up there where the lavender edge is at t10. Now what? We want to come back here, and I'm not allowed to sort of intersect myself. That would be the paper going into two parts of this unfolding, so that's not good. But I have all this space, so natural thing is to just wander from there back down to here, using up the space.

So it's going to look like this. Everything that you did, you just undo. Now, where this is painful is not only do I have to undo it in this diagram, but I have to recursively undo everything I did in here, everything I didn't in here, and everything I did in here.

So this recursive thing from this structure ends up getting doubled. At the parent structure, it will also get doubled. At every level of the tree, you're going to double what was below you, so that's why you get exponential, in general.

If your tree is ugly like something like this, you'll start with something nice and small down here, maybe constant number of terms. Then, you'll double. Then, you'll double again. Then, you'll double again. Then, you'll double again, and it will be exponential. So if this is n, the number of things you're doing here is 2 theta n.

On the other hand, if your tree happens to be nice and balanced, doubling is not so bad because here you'll have constant. This will double everything below. This a double everything below, but there's only log n levels. So is that linear? It should be about linear. It's certainly 2 to the theta log n, and it matters what this constant is. I think it's n or maybe n squared, but not too bad.

So if you're lucky and the just the structure of your bands is balanced, it's good. In general, though, it's going to be exponential. Open problem. Can you deal with these situations and sort of balance them and make them only make a polynomial number of cuts. Certainly be nice. Exponential number cuts is a lot, but it works.

You can unfold every orthogonal polyhedron this way. I would love to see an implementation of this algorithm. You could only do it in a computer because you'd be splicing into all these little things, and it would fall apart. Jason?

AUDIENCE: You've been making these, I guess, gadgets [INAUDIBLE] voxel would attach just by a side. You could imagine it attaching at a corner attaching to multiple sides.

PROFESSOR: So you're worried about-- not quite sure. What we think about is one band, which looks something like this, attaching to another band. That will always happen on a face. I'm not quite sure what you're imagining. Maybe something like that where they share a partial face here?

AUDIENCE: Yeah, but it could also be inset into the [INAUDIBLE].

PROFESSOR: If it's inset, I'm cutting with every-- I maybe didn't mention that-- through every vertex, I'm going to slice with a y-plane. So that will cut into lots of little strips, and then, there's no sort of overlap with the strips. I'm subdividing into little substrips. So that sort of deals with that issue if this was moved that way. Then, there will be three strips, one over here, one where they're overlapping, and one on the right. Yeah. Good question. I forgot to mention the subdivision at the beginning.

Speaking of subdivision at the beginning, this leads to another notion which we call grid unfolding. So the grid of an orthogonal polyhedron is what you get when you subdivide by extending every face into a plane and cutting with that plane. So that's sort of what I was doing here when they're overlapping. Even in this picture, there's like a face here, this vertical face, and so I'd end up slicing through this thing with a band in that direction.

And I don't know, this vertex you'll end up slicing through this guy. So hopefully, you can imagine that. With every vertex, slice x, y, and z. That's another way to do it. And that subdivides in sort of a nice grid where every face will now be a rectangle, and rectangles always meet whole edge to whole edge. So it's a nice simplification.

What I'm proposing is add those edges to your polyhedron. It's kind of like assuming that you started with unit cubes and build something but kept all the edges of all the cubes. I want an edge unfolding of that. Do those exist? These are what we call grid unfoldings.

So grid unfolding is an edge unfolding of the grid. This only makes sense for orthogonal polyhedra. Open question. Do grid unfoldings always exist? It's essentially the orthogonal analog of this question, edge unfolding a convex polyhedra. If you want to go to orthogonal, which is like in between convex and general non-convex, maybe you could hope for grid unfoldings.

Edge unfoldings obviously don't work. We had lots of examples where those fail, like the cube with little bites taken out of the edges. But grid unfolding, you get lots of subdivision. It might be easy. Well, it's not easy.

I would guess, actually, it's not possible. The next best thing you could hope for is to refine. So you take each of the grid rectangles and divide it into k by k, so subgrid. So ideally, k is one, and you're not subdividing at all. But maybe, you take every rectangle, divide it in half. Maybe that's enough to then be edge unfoldable. That would be sort of a refined level two grid-like unfolding.

There are a ton of results about this. They're all partial. Obviously, in the general situation-- I mean this algorithm we just covered-- you can achieve a refinement of only 2 to the theta n exponential.

When can you do better? Ideally, you get 1 by 1, but maybe, you can do something. What? OK. Interesting. One thing you could do, with merely 5 by 4 refinement, is something called Manhattan Towers.

Let me show you a picture of Manhattan Tower. No that's not a picture of Manhattan Tower. This is more crazy examples of what it's like to visit. This is not-- this probably is complete, but here, there isn't too much doubling because there's only a single child, more or less, everywhere. But the unfolding looks something like that.

Here is Manhattan Tower. So it has a connected base on the x-y plane. And I want to consider z-slices as I go up in z-coordinate, and I want those z-slices to get smaller and smaller, always contained in what I had before. So I never have overhang. That's a Manhattan Tower.

And in that case, 5 by 4 refinement is enough to unfold these things. So that's pretty good, still not perfect, but pretty good. And this is by the same authors, like a year before the general result.

Let's see. I think maybe I have a movie. Yeah. So this algorithm has been implemented, at least in some simple examples. And it kind of nicely unrolls. You can see a 5 by 4 refinement in that little staircase. It's, again, to make everything keep going to the right, but here, they find a clever way to visit all the faces without having to revisit, basically, at all, just visiting each face a constant number of times.

And then, we can zoom out, and you get the unfolding. So it looks very similar in spirit. Of course, the details is how do you do all that visiting. I'm not going to cover that here, but you get substantially less refinement for that special case.

Yeah. Another case looks like this. Boom!

AUDIENCE: Woah.

PROFESSOR: Isn't that cool? I'll play it again. This is just slightly more special. So again-- I have three of them. They're so much fun. It's like exploding a city. Boom!

So here, the floor is a rectangle. That's the only additional requirement, and again, as you slice upwards, things only get smaller. Here's a bigger one. Boom! So exciting. I could do this all day.

So I'm not going to describe how this works, but you could almost reconstruct it from these diagrams. This is what we call an orthogonal terrain. This result by Joe O'Rourke. Here, you don't need any refinement, grid unfolding, one by one. That's pretty sweet.

All right. Next one is what we call orthostacks. These are like a stack of a bunch of orthogonal polygons, a bunch of bands, basically. This is where the idea of bands came from. This is from an old paper in 1998, from the beginning.

So it's just I have a band, and then, I stack another band on top. So each z cross-section is connected. So that's a little different. With towers, I could have multiple towers here. I really only want one tower built slab by slab.

These things we don't know how to grid unfold. That's an open problem, but if you refine just in z by a factor of 2, that's enough to unfold. So 1 by 2 refinement is enough for orthostacks.

Now, you could go a little crazier and allow vertex unfolding of orthostacks with some grid refinement, and in that case, you don't need any refinement. So grid vertex unfolding orthostacks was the title of paper. These guys, John Iacono and Stefan Langerman.

And it looks something like this. He uses vertex unfolding to fix the direction. Here you were going up, but you really wanted to go right. So it makes things easier.

And in fact, the other guys, Damian, Flatland, and O'Rourke-- It's got to be awesome doing geometry and your last name is Flatland, unless she does a lot of three dimensional stuff. Irony.

You can do vertex grid unfolding of any orthogonal polyhedron is what I have written here. I haven't actually read that. I should read that paper. What else do I have?

Orthotubes. Orthotubes, this is again in the old paper. Orthotube is just sort of thickness one orthogonal tube. It could even be closed, I think, in a loop, but here I've shown it open.

And here, grid unfolding is enough. You just do all the grid refinement. You could even just do it locally. Technically, there's a slice here that might slice over here, but you don't have to worry about that. You just subdivide into a bunch of boxes, and this can be unfolded in a sort of zigzag fashion. So 1 by 1.

You can generalize this to trees also, though, it's not totally known. As long as you have a tree of cubes with fairly long connectors in between the branch points-- those are called well-separated orthotrees-- that works grid unfolding. If you don't have that condition, so just have cubes connect in some treelike fashion, it's open whether you can do a grid unfolding. We've worked on that in the past.

My conjecture's, in general, you need to omega n refinement. My belief is that this kind of exponential blow up is necessary, but that you can do it only a balance tree. So this would give like linear refinement. I think that's necessary, but we can't even prove that you need 2 buy 1 refinement in any example. We don't have anything where grid unfolding is definitely impossible. It's quite sad state, I guess.

It's very hard to prove that there aren't unfoldings, accept by exhaustive enumeration, and that's hard to do because it's slow. We've come up with lots of candidate examples, but eventually, we unfold them all. I have a bunch of other open problems.

This was genus 0. Interesting question is can you do genus higher than 0? Orthogonal polyhedra. I would guess so, but I'm not sure. I think the biggest question is, can you make this non-orthogonal? But then, the bands get messy. Haven't been able to do that.

All right. I think those were the main problems. Any questions? I'm going to take a break from unfolding now and switch the other direction of folding.

So with folding, we're imagining we're given some polygon, and we'd like to make a polyhedron out of it. It's exactly the reverse of what we've been thinking about. When is this possible?

Now, the rules of the game here are different from origami. With origami, we showed from any shape, you can make anything if you scale it down. But here, I really want exact coverage. Every point here, or let's say every little patch here, should be covered by exactly one layer over here, not allowing multiple layers. So this means not everything is possible. I also care about scale factor, but--

This one, of course, you can crease like this, and more importantly, you glue these edges together. You glue these edges together. The opposite of cutting is gluing. We'll be more formal about defining gluing, I think, next lecture. This is just a sort of prelude.

But you end up gluing-- I want to make something, let's say-- in fact, we're always going to talk about folding convex polyhedra. There's very little work on the non-convex case, though, there was actually a recent result. I'll mention that next lecture.

If you want to make something convex, and therefore, sphere-like, you have to get rid of all the boundary, so you've got a glue every edge. You don't have to glue whole edges to whole edges. Maybe you just glue part of an edge to another part of an edge, but somehow, boundary has to get glued up. And if you want something sphere-like, in fact, those gluings have to be non-crossing. I have to be able to draw a picture like this.

Question is when do these gluings make a polyhedron? That is the question we will be answering next class. But first question is suppose I gave you one of these pictures. I give you a polygon, and I give you a gluing. What this tells you is sort of how to locally walk around.

So if I'm here, I could walk over here. I could walk over here, teleport over there, walk over here, teleport over here. Whatever. OK? The gluing tells you locally what the surface looks like, even though you don't know what it looks like yet in 3D.

In particular, you can compute shortest paths here. I could compute the shortest path from this point to this point you might think is a straight line. But no, it's not. I don't think. Or maybe it is. Let's see. A little tricky.

AUDIENCE: Should be diagonal with the bottom square.

PROFESSOR: Diagonal with the bottom square because these guys are both in the bottom. Very good. So this point is the same as this point, or no, this point. Because these get zipped together. This point is the same as this point, so in fact, that diagonal is the shortest path between those two points. So you have to think about it for a while, but it turns out, in polynomial time, you can do that. That's cool.

What I want to show now is that suppose you could make a convex polyhedron in this way. I claim you can only make one, never more than one from the same gluing. So I've defined locally what this thing is. It's like a piece of paper. I can mangle it around. If I want to make something convex, there's only one thing it could possibly make.

Finding out what that one thing is quite a challenge, but at least, we can prove that it's unique. Sorry about the screeching. This is Cauchy's rigidity theorem. I think we mentioned it in a previous lecture when we were talking about rigidity and three dimensions and like why domes stand up. But now, we're going to use it. We're actually going to prove this theorem, and we're going to use it to study these kinds of gluings and say there's, at most, one way to do this.

There's a lot of ways to state this theorem, but one way is to say, suppose you have two convex polyhedra, and suppose they came from the same sort of intrinsic geometry. So there's the geometry of the faces, and there's how they're connected together. So here, initially, I drew a tree of how the faces were connected together, and then, I drew some other connections like this. So the dual here is, of course, an octahedron.

But if you have two convex polyhedra and they are combinatorally equivalent, they have the same way that things are connected together, the same graph, and they have congruent faces, so the geometries match also, then, they are actually the same thing, the same polyhedron.

AUDIENCE: Is that something that--

PROFESSOR: Up to rotation and translation.

AUDIENCE: Is it the same set of congruent faces?

PROFESSOR: Oh, yeah. So when I say congruent faces, I mean according to this equivalence. If you take a face on one that has a corresponding face on the other, those two faces should be congruent, not just any pair, and they're not all the same. Different faces can be different, but they're identical in pairs.

So I'm just saying basically, if you have this picture, there's only one realization. So this picture defines what are the geometry of the faces, how are they connected together. And so if you had two convex polyhedra with that same underlying diagram, they have to be the same polyhedron. That's what we're claiming. That's what we're going to prove.

Is this one any better? Yeah, it's better. This is an old theorem. You may have heard of Cauchy, famous French mathematician. Cauchy-Schwarz inequality, all those good things. You don't need to know those. He proved a lot of things.

This theorem he didn't actually prove. He wrote a paper about it or, I think, partly a letter, partly a paper in 1813. Proof was wrong, and it was fixed in 1934, over 100 years later, by Steinitz. So sometimes it's called Cauchy-Steinitz rigidity theorem, although, usually, Cauchy. There's a lemma in here that's often attributed to both of them.

So it's sort of a proof by contradiction. We want to prove uniqueness. So we're supposing, well, maybe, there's two polyhedra, p and p prime, and they're combinatory equivalent and have matching faces congruent. What I want to do is look at corresponding vertices, let's say a vertex v and p and a vertex v prime and p prime. So there corresponding in the combinatorial equivalence.

And then, I want to slice the polyhedra, p and p prime, with an epsilon sphere, epsilon radius sphere centered at v and v prime. Quick mention, this is not true if you allow non-convex realizations. You may have seen that example before. These have exactly the same combinatorial structure, same geometry on each face, but one's non-convex, and they're different. But as long as they're convex, they're going to be the same.

In a convex situation, here's what the slice looks like. So here's little vertex, degree like 5, and I slice the polyhedron with this tiny sphere centered at v, not the interior, just the boundary of the sphere. What I get are these arcs. They'll be great circular arcs.

Here we can see all five of them. They're great circular arcs. There on the sphere. I get a polygon. I get a convex polygon on the sphere.

Convex spherical polygons, convex because the polyhedra are convex. I actually get two of them. Right? One for p, one for p prime. Polygons. What do I know about those polygons? I know they're edge lengths because the length of an edge here is equal to-- if this is a unit sphere, if I rescale the epsilon to be 1, that edge length is actually this angle. That angle is an angle of the face at the vertex.

So I know all the angles. I know all the geometries of the faces. I know they're congruent. So I know what these edge lengths are in the sphere.

What I don't know are these angles of the polygon. I know the lengths of the polygon, but I don't know the angles of the polygon because that's essentially the dihedral angle of that edge. And the worry is, well, maybe all the edge lengths match. We could do this at every vertex, but maybe the dihedral angles are different. Maybe the convex polyhedron could flex. That's why this is about rigidity.

Maybe it's flexible, and then, maybe there are two different states. All the faces are the same, and therefore, all those edge lengths are the same, but the angles might differ. If p and p prime are supposed to be different, then there must be two angles that differ. So I want to look at those angles, and I want to label vertex of a spherical polygon.

So I have n of these spherical polygons for p n of them and p prime. Let's look at them in p. I'm going to label it plus if the angle there in p is bigger than the angle in p prime. Remember I have a correspondence between everything in p and everything and p prime. Minus, if the angle is less, and 0, if the angles are equal.

So what I want to show is actually all the angles are equal. Therefore, the polyhedra will be identical. If all the faces are the same and all the angles at which you join them are the same and they're connected in the same way, they are the same polyhedron. There's no flexibility there, but it could be there are pluses and minuses.

But if there's going to be a problem with this theorem, there have to be pluses and minuses. That's the proof by contradiction. So let's look at one of them. Let's look at a vertex that has some pluses or minuses.

So we have-- it's a spherical polygon. I'm going to draw it more like a polygon, maybe some pluses, some zeroes, some minuses, whatever. First question is could it be all pluses and zeroes? Is that possible? No. It doesn't look good.

What does that mean? It would mean this is like a linkage. We're used to linkages. It just happens to be on a sphere. Forget this on a sphere. Think of it is as almost flat.

What that would mean is there's some other way to draw this thing. Basically, there's a way to flex this linkage so that all of these angles increase and this one stays the same. How could I get a polygon where all the angles increase and still be convex? Ain't possible.

Why is it not possible? I think we've used this fact a couple lectures ago. It's not possible by something called the Cauchy Arm Lemma. This is the part that Cauchy got wrong, so it's also called the Cauchy-Steinitz Arm Lemma.

And here's the thing. If you have a convex chain but open chain here. There's a missing bar. So suppose that even when you add the bar, it's a convex polygon. But then, I take a convex polygon, remove an edge. This is what we call convex chain.

And if you open all the angles, increasing all the angles, in a convex chain, then this distance increases. It's pretty intuitive. I open all these angles. So I put plus.

Some of them could stay the same, but then, this distance will increase, as long as you stay convex. I should mention that. But in this situation, we know that both the initial position in p and its target position in p prime are both convex.

OK. Let's just take that lemma as given. This is the one I don't like. Then, I know, in particular, I can't have all pluses because then, if I just pretend one of these edges wasn't here, I know that that distance must increase. But it can't increase. It's supposed to stay the same. The edge lengths are fixed.

So if they're all pluses and zeroes, or all minuses and zeroes, the same is true just viewing p prime as p and p as p prime. They can't be all pluses. They can't be all minuses. So if there's anything in there other than zeroes, there has to be at least one plus, at least one minus.

In particular, there have to be at least two alternations. Alternation is either going from plus to minus or from minus to plus. So it could be something like plus, plus, plus, minus, minus, minus, minus, plus, plus. Whatever. OK? Maybe that's your polygon, and that's your labeling.

Is that possible? So once you have at least one plus and one minus, you have to have these two switches at least. Could this happen?

AUDIENCE: No.

PROFESSOR: No. Right. Because you pick some chord. What do I do here? Just subdivide. All right. Whatever. Pick some chord like this one. I'm not sure it matters. Maybe here? Whatever.

These angles down here are decreasing. Therefore, this distance decreases. The angles up here are all increasing. Therefore, this distance increases. Can't have both. So this is also not possible.

So in fact, you have to have at least four alternations. It's always even. And so it has to be at least a bunch of pluses, then a bunch of minuses, then a bunch of pluses, then a bunch of minuses. And so we're counting these transitions from plus to minus.

This is a lemma that we used when we were locking trees, I think. We had a convex polygon. I said at least two of the angles have to decrease, or at least two of them have to increase, one way or the other. That's because there have to be at least two groups of pluses, not only at least two pluses and at least two minuses.

You might think, well, what happens if there's only three vertices. Then, you can't have these four alternations. Well, yeah, you can't have those four alternation because if you have a triangle, even on the sphere, triangles are rigid. So you would know if I had a degree 3 vertex, locally that thing is rigid. It can't flex at all. We're only interested in cases where it might flex locally at a vertex like the pentagon, like a quadrilateral.

All right. So what? This was true at every vertex that was not entirely zero. So if we look at-- let me write this down. I really just care about the non-zero edges, the ones that are plus or minus. So I'll call the subgraph of plus and minus edges. The number of alternations is at least 4 times the number of vertices. I'm going to denote the number of vertices by a capital V. OK? That's page one.

We're going to use a trick. It has many names, I suppose. I usually call it double counting in combinatorics, where you have one quantity, namely, the number of alterations. We're going to count it in two different ways. We'll get two different answers, but we know they must end up being the same. And then, we'll get a contradiction.

So what's the second way of counting. Well, we counted local to a vertex. The other natural way to count angles is by looking at the faces. There are also faces. It's sort of the dual perspective. Every phase has a bunch of angles that have some degree or whatever.

They're really kind of the same thing. Oh, here was Cauchy's Arm Lemma. Beautiful. If you look at the alternations as you walk around a vertex versus as you walk around a face, you'll end up counting them the same number. Right? Here was an alternation from plus to minus.

What's interesting here-- before we were thinking of labeling the vertices of the spherical polygon, but in fact, whatever this edge does, it does the same thing local to that vertex as the thing local to that vertex. So really the labels are on the edges of the graph. They could be zero, plus, or minus.

And if I have an alternation from plus to minus, view from the vertex, it's also an alternation as I walk around the face. So instead of counting by walking around all the vertices-- which are just did, and I got at least four at every vertex-- let's do it from the perspective of the faces. And we're in this weird subgraph of plus and minus edges, so assume there are no zeroes.

All right. If I have a face of 2 k or 2 k plus 1 edges, then it will have, at most, 2 k alternations. So I already have a lower bound number of alternations. I'm going to try and prove an upper bound, sandwich it between, and show that, actually, the upper bound is smaller than the lower bound, and that's a contradiction.

So this is kind of obvious. Right? If you have 2 k vertices, no more than 2 k alternations, slight, the place where we're making a little improvement is for the odd case. We know because a number of alterations has to be even you can't get up to 2 k plus 1 alternations. Has to be even because for every plus to minus, there has to be a matching minus to plus because it's cyclic. So even here, you can only get 2 k alternations.

That helps us a little bit because now, we can talk about number of alternation is at most two times the number of triangles, f sub 3 is going to be the number of faces of degree 3, plus 4 times the number of quadrilaterals plus 4 times the number of pentagons plus 6 times the number of hexagons plus-- why'd I write seven? I wrote seven. I'm being sloppy.

At that point, I don't care. 7 f 7, 8 f 8, 9 f 9-- I'm not going to try to be clever from 6 on, but I'm going to be clever at 5 and 3. Why 5 and 3?

As you may remember from way back when, you have a planar graph-- because polyhedra are planar graphs. They're convex. The average degree is 5? Slightly under 6, 4, 3, 2, 1? One of those numbers. Let's see. Should be like 3 n minus 6 edges, so that should be 3. Yeah, three. So most of the faces are going to have low degree. That's the points. So 3 and 5 really matter, but out here, it doesn't matter so much.

This is kind of a magical proof. It shouldn't be intuitive where it came from, but it's really beautiful. You'll see as it all comes together. Fun. What do I do next? Right.

I want to relate-- I have a vertex count here. I have a face count here, and I know Euler's formula. Hopefully, we know it. It's a cool formula. V minus E plus F is 2. This is the number of vertices, number of edges, number faces is two. [? For ?] connected, planar graphs.

There are other versions when you have multiple components or when you have tori, genus, whatever, but for convex polyhedra, this is true. So this conveniently relates vertices to faces, but it involves edges. So somehow, I have to bring edges into the mix. All right. Well, edges.

I want to count the number of edges in terms of the number of faces. I could do it in terms of vertices or faces. The number of edges is half the sum of the degrees of the vertices. Remember, that's handshaking lemma from way back when.

It's also half the sum of the degrees of the faces. If I look at every face and I count the number of edges, I will end up counting every edge twice, once from each side, so this is half the number of faces. Well, number faces I want to write in this form. So this is half 2-- No, sorry.

Not the number faces. What am I doing? Half the sum of the degrees of the faces. So this is half-- what is the degree of degree 3 faces? 3. What is the degree of degree 4 faces? 4. And so on. Exactly.

So now, things are starting to look similar, and I want to get some cancellation going on. Use my cheat sheet here. I'm going to rewrite this formula as V equals 2 plus E minus F. Hopefully, that's the same. I put E over here. I put F over there. We'll get this.

OK. So now, I have E minus F. E is this. F-- I have F. Am I missing something here? I'll write the next one, and I'll figure out where it came from.

Oh, duh, I just skipped a step. F is the sum of f3 plus f4 plus f5 plus f6. OK? All I did here was decrease by-- well, because there's a half out here, I decrease each coefficient by 2, nothing surprising. Whew! So I took E, I subtracted F, just took away one of each. Now, I have this formula. That's V.

Now, I also know that 4V is at most, the number of alternations. Hm. So I could get a formula for 4V here. Right? 4V is going to be 8 plus this is going to be like 2. So I get 2 f3 plus-- oh boy, 1 times-- 4 times f4 plus-- just double these numbers-- 6 times f5 plus-- what's the next one? 4-- 8 times f6, and so on. Yes. OK.

Now, these guys look very similar to these guys. Now, here it gets bigger. 8. It's going to go 10, instead of 6 and 7. We also have a plus 8. We don't know whether there any faces of degree 6 or more, so we can't rely on that, but we have plus 8.

So somehow, 4V, which is, at most, the number of alternations-- that's the reverse of what we said before-- 4V is, at most, this number, and yet, it's also equal to this number. It can't be both. This number's at least 8 larger than this number. It could be even more larger, but at least that contradiction done.

This works as long as there's at least one face, meaning there's at least 1 plus or minus because we're only looking at the subgraph plus and minus edges. And that is Cauchy's rigidity theorem.

Let me quickly tell you in our situation, here, we don't actually necessarily know where the creases are. We just know how things are glued together. Even in that situation, you could sort of figure out where the creases must be because as I said, you can compute shortest paths once you have the gluing. So you compute the shortest paths between all pairs of vertices, something like this picture, except you don't know what it looks like in 3D.

You can still compute the shortest paths. You know every edge must be a shortest path. So the edges are some subset of these guys. And so you've got lots of little convex polygons here. We know it must make a convex polyhedron. If it made two, Cauchy's rigidity theorem would tell you that they're the same.

So even once you fix the gluing, you know that there's a unique convex realization, and there will be a unique set of edges from the shortest paths that actually realize it. The next class will be all about how to actually find those gluings and know that they actually will fold into some convex shape and how to find that convex shape, but that's it for today.