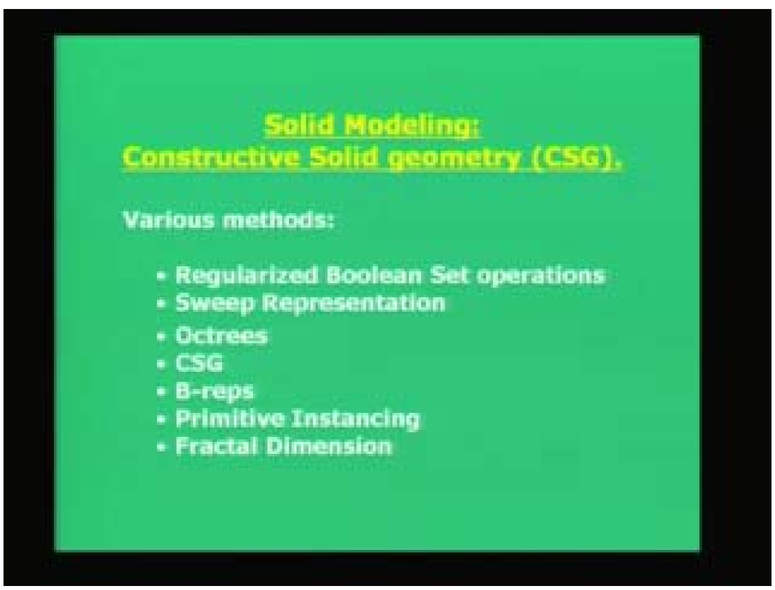
Computer Graphics
Prof. Sukhendu Das
Dept. of Computer Science and Engineering
Indian Institute of Technology, Madras
Lecture- 23
Solid Modelling

Hello and welcome everybody again to the lectures on Computer Graphics. Today we are going to discuss methods to represent solid objects and see how you represent solid object structures in this field of computer graphics. We have to know methods by which we represent them or model them and then position them in the first stage during pipeline before we apply normal transformations.

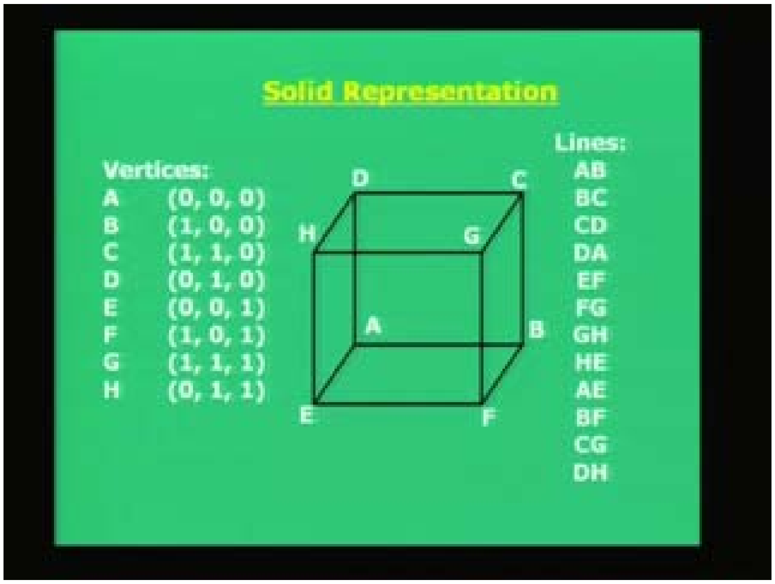
Objects which we see before us typically are both natural and artificial objects and there are various methods by which these are represented. Depending upon the application we sometimes need to represent solid objects, sometimes need to represent soft objects also. But we will restrict our discussion today based on methods to represent solid objects. So solid modeling is the title of discussion today and out of the various categories the most significant one will be the constructive solid geometry or CSG.

(Refer Slide Time: 02:28 min)



So if you look into the slide the various methods which are used to represent or model solid objects are; regularized Boolean set operations, sweep representations, octrees, CSG or constructive solid geometry which is also one of the methods of representing solids or modeling solids then B-reps or boundary representations which we will see. I think we will mainly discuss the first five methods depending upon the scope and time available to us. There are methods of primitive instancing and fractal dimension which I leave for users to read themselves, which are the fractal dimensions are method by which you represent the natural object structures not necessarily solids. So we first look into a method by which a solid is represented.

(Refer Slide Time: 03:22)



Let us take an example of a simple cube. Let us take the example of a simple cube we call structures and which has eight vertices at the endpoints of the cube, four on the bottom, four on the top or four on the left and four on the right. The corresponding vertices are A B C and D E F G and H. And if you visualize that the vertex A of this unit cube, unit cube means the length of the solid is unity, the length, height and breadth. And if the vertex A is at the origin of the coordinate system then the corresponding vertices, the coordinates of rest of the vertices are also given on the left hand side of the cube.

We assume that the line AB lies on the X axis whereas the line AD or EH lie on the Y axis and correspondingly the line AE or BF they lie parallel to the Z axis. So these are the three orthogonal axis on which the cube is lying and the vertex A is at the origin of the coordinate system. So based on this concept you can visualize the left hand side table which talks about the coordinates of the different vertices on the left hand side. So a set of eight vertices can be represented as solid, in this case a cube let us say. So you have to basically store the set of vertices and that is the sufficient information in this case to represent a cube.

But let us say, if we talk about arbitrary objects it is not always possible to represent the solid object structures with the help of vertices only. Of course you need to store vertices. But you also need to find out which of the vertices are connected by lines and which of the lines form a particular polygonal shape or a square shape in this particular case of a cube. Let us say the above is connected to the vertex B at 1 0 0 by a line. As you see there is no mention of a line joining vertices A and C because it does not exist in the picture. So you can see lines such as AC DE they do not exist on the right hand side column.

If you count the number of line how many lines do you think exist in a cube. You know, four on the bottom four on the top and the four along the side so there should be twelve lines or twelve edges which exist in a cube and you can count on the right hand side that there are actually there are twelve lines. So these twelve lines tell you which of these vertices are connected. That is another way by which the solid object also can be represented and that means you not only need to specify the vertices but you also need to specify the lamp lines which connect the corresponding vertices. So this is one way by which a solid can be represented with the help of vertices and lines.

Of course but this is not a complete representation. We actually need some more information in terms of trying to know which of these lines form an enclosed polygon. So solid we know is bound by surfaces so we also need to define the polygons of the vertices which form the solid. It must be also a valid representation. If you take any solid in front of you let us say you take a furniture of a classroom, you take the chairs and the tables and in some cases you have the board, you may have computer systems being used for practical laboratory classes, you may have apparatus, electrical and mechanical instruments being used.

If you look at those artificial solid objects typically any such solid object preferably it is easy for you to visualize the box type of a structure or a cube for that matter. In the simplest case it is always bound by surfaces. So actually on the top level of representation you need to define surfaces which encloses a particular solid. And when you talk of surfaces, in this case of course we are assuming very simplistic planar surfaces or planes which bind the particular solid, any solid for that matter. And when we talk of planar surfaces or planes which constitute surfaces to bound the solid then of course you need the representation to represent those surfaces or planar surfaces and planar surfaces can be represented using polygons we know that.

In the case of a cube of course the polygons are simply squares of unit length. But it could be any arbitrary solid which you can visualize which should be bound by the surfaces and so we need to hence define polygons with the help of vertices to which we should represent a surface. So let me again clarify that the solid is bound by surfaces.

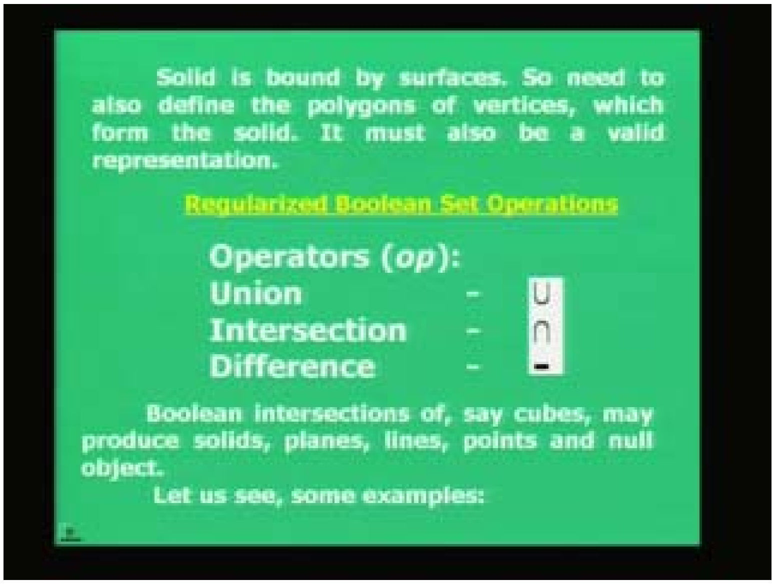
There are n number of surfaces which bind a particular solid. Each of the surfaces will be represented by polygons. So each polygon will have n different vertices or lines which define that polygon. And since as you have seen in the previous slide that once you define the vertices with the help of vertices you can define lines which join the corresponding vertices and then you can pick up a set of lines or edges which constitute a polygon. Those polygons actually define surfaces which bind the solid.

So I hope you can understand that at the top you are talking of surfaces then the edges or lines which define this polygonal surfaces and then of course vertices which define the corresponding line or the edges which join the pair of vertices to form a line. So at the very low level you have vertices then lines or edges and then polygonal surfaces to represent a solid structure. Or you can look from whether we are from top to bottom the solid structure is represented with the help of polygonal surfaces.

Polygons are defined with the help of lines or vertices and at the very low level we have the vertices which define lines as well. So if you look back, I repeat the sentences once again, a solid is bound by surfaces so we also need to define the polygons of the vertices which form the solid. I hope this statement is very clear now and of course we also need to check whether it is a valid representation of a solid because the solid is bound, there is no open space for a solid but typically when we see a solid structure it should be bound by surfaces on all sides such as left, right, top, bottom, front and back and so on and so forth. A cube can be the typical example.

We cannot visualize that a solid exist like a single plane. We cannot define a solid with a single plane because it is not binding any particular solid volume or a structure. So we will see methods by which we need to validate the representations of solids and we will see with one such method the help of the first representation which we look is regularized Boolean set operations in solid.

(Refer Slide Time: 12:12)



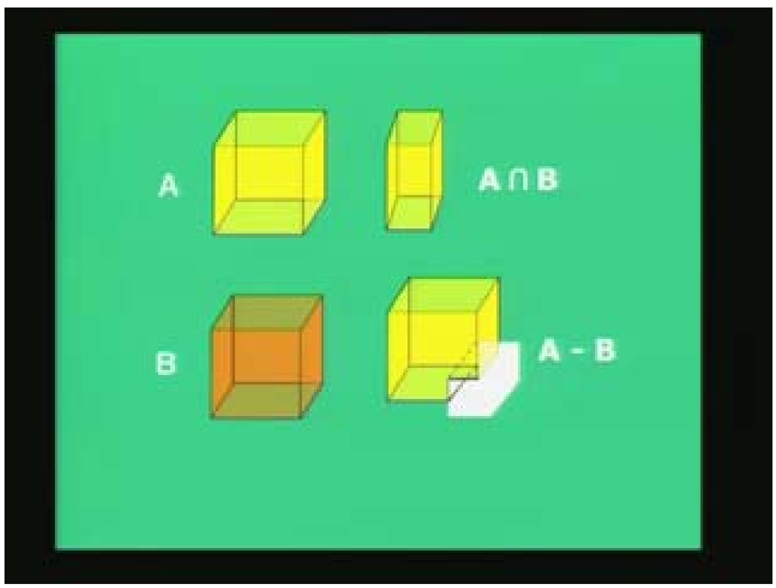
Before we move on to valid representations or methods to validate we will look into another representation of solids and methods by which you can create different type of solids from simple structures. Let us look at this simple example now. Regularized Boolean set operations are of three categories.

You basically define a set of operators which operates on solid structures and the three operators are simply used or called as the union intersection and the difference. The corresponding symbols are given on the right hand side, this is the symbol for union intersections. It is a simple set based operation on a set and the difference between the two. So Boolean intersections of cubes because we actually know right now how to represent a cube at least because we can define the cubes with help of about eight surfaces.

Each such surface will be defined with the help of four polygonal edges or squares. So the number of the surfaces which you have although are six. The number of edges lines which we have seen for the polygons are twelve and the numbers of vertices are also eight. So remember these figures, this will come back.

I repeat again, for a cube we have six surfaces, twelve edges or lines and eight vertices. So remember these figures. And once you define this cube we will say that with this operations of union intersections and difference mainly the intersections of cubes we may also produce other types of solids. We can also produce planes, lines, points and what we call as null objects and we will see with an example what do you mean by this. Let us see some examples of regularized Boolean set operations. But we just see the examples of Boolean set operations first and then we will see what you mean by the term regularization on these Boolean set operations. Let us take two cubes, the same cube which you have seen couple of slides back.

(Refer Slide Time: 19:09)



Now I have taken two such cubes. Each of this cube are of the same size and dimension. Each of them have eight vertices and six surfaces and twelve lines which constitute each of these cubes. I remember these values once again and let us say there are two such cubes represented by two different colors of identical size A and B. If that is so then we look at the intersection of A and B.

Now, depending upon the position of A with respect to B you will get different values of A intersection B. We have to visualize that this intersection can be obtained by placing this one cube within the other where there is a overlap of one portion of the solid portion or the volume covered by A and also with B. So the A intersection B which you are seeing is a common part of the volume occupied by both A and B when A and B are not only placed adjacent but one is little bit inside the other. So if both A and B occupy the same volume that means they occupy the same space then the intersection produces the same object.

Now if you start sliding one with respect to other assume that both A and B on the same place, remember we are working in a virtual environment. Computer graphics is supposed to produce a virtual reality or a virtual environment and although in the realistically true world which we live, not in the virtual world, we cannot put one solid object within another one unless you have a vacuum inside, if we have a solid structure say rock or a table we cannot put another object like a chair or a pen inside that where the surfaces intersect, it is not possible.

But remember, this is a virtual scenario, a virtual reality, we are living in a virtual world with computer graphics and here it is possible to have one solid penetrating inside the zone of the other one. That is virtually possible. Whether it is valid or not we will see later on but right now assume that it is possible. If it is possible let us say if A and B are two separate cubes as shown in the slide, a couple of minutes earlier in the last slide which we have seen if we place A and B in such a manner that both of them occupy identical or same volumes in the coordinate system. That means the we have one of the vertices at the origin of the coordinates system of both these cubes and the corresponding three axis which comes out of that particular vertex gets aligned with the X Y and Z axis. That is the case then both A and B occupy the same position and in that particular case the intersection of A and B will produce the same output.

However, if you take B and start sliding it with respect to B along any of one orthogonal axis X Y and Z then what will happen is if you don't take of course the B completely out of A there after sliding a little bit we will have some common volume which will be or some volume in 3d space which will be not a cube but a rectangular parallelepiped type of a structure which will be common to both these volumes A and B.

This is a structure which is shown in the slide here (Refer Slide time: 16:03 min). So what I have done is basically put B on A at the same position and started sliding little bit. The more you slide lesser will be width of the intersection and when you have just did a little bit and there is huge amount of volume which is common to A and B then the intersection volume also will be very very large. So you have to visualize this that the intersection A and B is a common volume between both A and B. Now, if you keep on sliding it quite a lot what will happen is at some point of time A and B will be adjacent to each other. Adjacent means there is just one common plane which is just too common to both volumes A and B and the common intersection actually could result in a plane. Depending upon the relative positions of A and B these two solids the intersection which is also a solid output in general will depend on the relative positions of A and B. The A and B the relative positions dictate the intersection volume.

I repeat, when A and B occupy the same place we have the same output of structure of both A and B starts sliding, you start of intersection volumes of rectangular parallel pipette coming out. The more amount of slide which you give lesser is the common volume which comes out and at one point of time when both A and B are just adjacent where there is just one plane between the two solids A and B you will have to visualize that you have just one common plane. It is possible that I can position A and B in such a manner that there could be just one particular line common to A and B or even one particular point common to A and B. Just one of the vertices could be shared common between two structures.

Visualize this, an object A or a solid, in this case a cube and I put another solid B of the same structure in such a manner that there is one vertex common to both these volumes. So depending upon the relative location of A and B you will generally have intersecting solids which is the common volume between these two solids A and B. You could have a resultant plane, you could have a common volume as a resultant line or a vertex point or the other option is when two solids A and B are separated wide apart where there is nothing common between the two you could result in a, what is the common space which you will be resultant? You will be having a null object.

You could have a null object as an output when A and B do not share anything in common in terms of volumes, surfaces, lines or even points. That was the point in the last slide in the statement which we made. That intersection produces volumes, it could produce planes, it could produce lines, it could produce a point or it could also produce a null object.

In this particular case the example is shown, I have just shown the example where you have a common intersecting volume like a rectangular parallelepiped but you can visualize that if I keep sliding B more with respect to A, the common volume will start shrinking in size and at some point I have plane, I can have a line or a particular point.

We look at an operation called the difference between A and B, A minus B is the difference operation. Here what I have done is I have taken the solid A and I have taken the solid B and positioned it in such a manner that the low right volume portion of A is the same common volume as the top left portion of B. And if you do that and subtract from A the common part or the basically the intersection between A and B, subtract that portion you result in a difference operation. As you can see that white volume portion is the part of B which is basically common or to A and you result in this structure A minus B as the difference operation produced by this. This is another operation, the difference operation. You can visualize the difference operation as you produce an intersection first between the two solids A and B and subtract that common part from the structure of A. This is how you can produce another type of a structure using the difference operation.
So I hope you will able to visualize again.

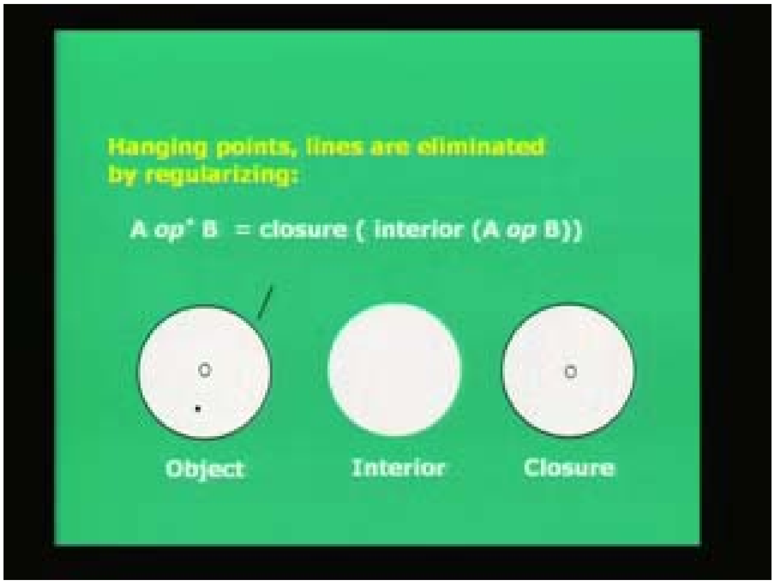
I repeat again, the relative positions of A and B will dictate the resultant volume which you will get in the difference operation A minus B as you would have got in the intersection volume of A and B. The relative positions of A and B as was dictating the intersection volume, resultant volume due to the intersection operation A and B, also the relative position will also dictate what you will get by the difference operations. These are the two operations. Of course you can visualize what is going to happen in the case of union, union will be the aggregate sum of the volumes.

So if you have A and B next to that or with a common part or just even an adjacent, the total volume in what you have is A and B put together. And of course again resultant volume which you will get, resultant solid structure or the volume which you will get as the result of union operation basically is dictated by the relative locations of both A and B. So you have seen three operations out of which I have illustrated only two. If you have understood these two the union operation is easy to visualize into regular solid structure. With the help of these cubes you can generate other solids and then you can actually keep generating more and more type of complex structures with the help of this.

As for this example, in this case you could use A intersection B along with the A minus B the difference between A and B to produce more complex structures by positioning them relative with respect to each other and then creating either union difference or intersection. These are the three Boolean set operations on solid structures. In this case we have started with cubes but as you can see the resulting structure which you have got is solids but they are not cubes any more. So you can create more and more complex structures with the help of this Boolean set operation and then we move into constructive solid geometry. We will see how with the help of Boolean set of operations we can create few more complex structures. Now what is the process of regularization and the question is why you need regularization in a solid?
As I said before, with the help of such operation or intersections you could have hanging points, lines or even surfaces of solids coming out due to Boolean set operations.

I was talking about this operation called intersection where you put A and B adjacent in such a manner that there is a common plane or you could have a common line or you can have a common point. If that is the case, that is not the solid because a line or a single surface or definitely a point cannot enclose a volume, it cannot bind the particular solid. It cannot create a solid with just with a help of one or even two. Two surfaces or lines or points cannot create a solid or you cannot define a solid with the help of just one or two points lines or surfaces. And in reality of course there is no question of a single surface or a single point or a line hanging in the free space and you need to represent such a typical because when we talking of solids these are structures or volumes which are bound by surfaces. You definitely need more than one or two surfaces which bind. So if there is a result of any Boolean set operation mostly the difference operation or the let us say the intersection operation may produce an isolated point or line or a surface. You do not need to keep that because either it exists along with a solid structure or without it, it cannot exist in practice.

(Refer Slide Time: 22:19)



So you need some regularization operations on Boolean set operations to remove these hanging points, lines and surfaces to create an ideal solid which is bound by the surfaces. You need these operations and that is what we are talking about, regularization here. And if you look at the structure here I have produced A operator star B where I will say this star indicates the regularization operation. Whereas in the right hand side if you see I have used an operation A op B where this op operator basically is a Boolean set operation without regularization.

What we have discussed earlier are operators intersection union difference which are Boolean set operators but without regularization because these produce hanging points and lines and surfaces. And we need to implement regularization on this which will help us to produce what we call as regularized Boolean set operations. So let us see with this example.

Let us say the left hand side is an object structure which is produced with the help of some operators which is non-regularized and we have a solid structure. But in addition we have a hanging line or it could be a hanging surface and also we have an isolated point here on this object. So we need to remove these two and get only this object structure. It may exist with a hole, there is absolutely no problem of a solid structure having a hole. But it should not have this hanging line or a hanging surface or hanging point.

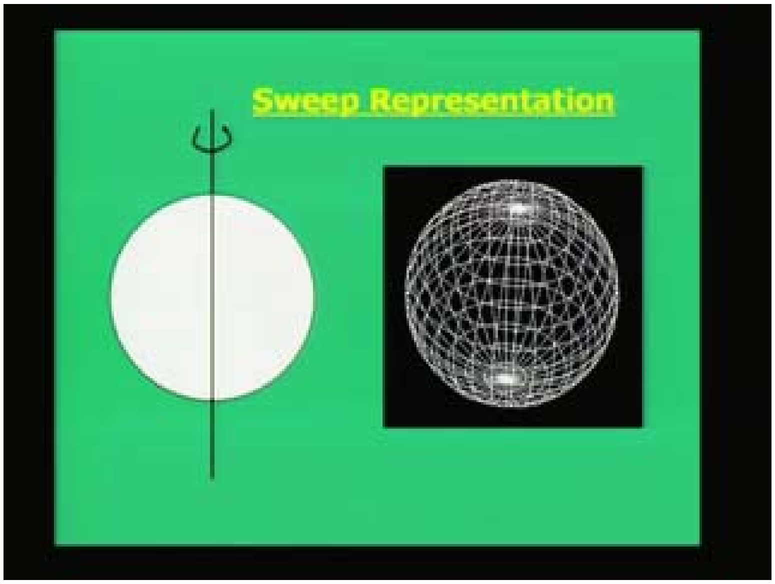
Of course in this case I am asking you to visualize in 2d but you can visualize this object also in 3d that as if this is a concentric cylinder where there is a hole in between. So, we define an operation called an interior on the resultant operation on this object produced by A operated on B and that produces this object on the left hand side and then we have this interior operation. This interior operation is the one which helps to remove what we call as boundaries of objects. Boundaries are certain surfaces which I will say is common to the set of points defining in the object and the complement of the object. Complement of this object is the space outside the object. So this is the surface which I will say as common to the object and it is complement to the object. That is what it is. If I use to define a boundary then an interior operation is an operation which will throw away these boundary points and that will help us as you can see to throw away all boundary points including the boundary of the object itself. And of course the hanging plane of the line goes and of course what will happen is these isolated points will go but it might create a small hole here because we are looking at interior operations which is this object minus the boundary. So any points or lines which are defined as boundaries on this object are removed to create this interior point so we may have two holes but this object is big as a boundary.

Now what we do is we implement a closure operation after the interior operation which will restore this boundary back. So the closure operation is the operation after the interior operation takes the solid structure and defines the boundary, includes the boundary or attaches the boundary on the solid structure. So if you have a small hole which are just boundary points around and if it does not exist in the structure those boundary points will be added and it will be created as an interior part. This closure operation will include boundary points. It will also close the small hole or gap which exists due to this isolated point here. So the regularized Boolean operation as you can see is a successive operation of an interior operation and a closure operation.

I would request you to go to books and see few more examples of regularized Boolean set operations. This is picked up from the examples from the book by Foley Van Dam and so I repeat again, regularized Boolean set operations followed by interior and closure operations.

I repeat again, Boolean set operations, unions intersections and difference are three examples we have seen followed by interior and closure operation implements what we call as the regularized Boolean set operations. So this is how we create solids which are of valid solid now because that solid will not have hanging points, will not have hanging lines in the object and the object will be close by the bounding planes. This is an example of how you implement regularized Boolean set operations. We will move to the next representation of a solid, which is the Sweep Representation.

(Refer Slide Time: 29:01 min)



Sweep Representation is a method by which you can generate solids with the help of 2D out layers or 2D structures. In this particular case I imagine you to visualize the left hand side structure which is a circle consisting of set of points on the circumference. So we define a set of points on the circumference of a circle which is not a solid, it is just a circle, is just a 2D arc circle. So you have just a circumference and you have a set of points.

For the time being you assume that these points are uniformly spaced and then what you have to visualize is, I take an axis a vertical axis passing through the center of the circle and I now rotate this 2D circular arc in 3D about the axis. I hope you remember 3D transformations of rotation. When we talk of rotation we defined an axis of rotation and then an amount of rotation about that axis.

Now what we do is we have taken a 2D circular arc which is just a set of points around the circumference that is what could represent a circle. You can visualize that these are the points generated by the Bresenham's midpoint circle algorithm. Remember midpoint circle algorithm, midpoint ellipse and midpoint line so out of those three think about midpoint circle algorithm. So, as if you generated instead of 2D points lying on the circumference of the circle. And now you visualize a vertical axis the Z axis which will rotate this particular structure and then what you do is you rotate it by a finite amount of steps.

Therefore, you take the circular arc and rotate it by a small amount delta theta and then get this new point in 3D. Just store the set of new points which these points on the arc of the circle will create. Rotate it by the same amount delta theta again and then store all these amounts. So, if you visualize what will happen is, these points as you keep rotating, of course you need to rotate just by about 5 radians or 180 degree in steps of course of the small amount. That small amount could be 5 degrees, 10 degree or even 30 degrees as largest as that. But typically to get better resolution it is better to have incrementally small amount of rotation by theta. It could be as those about 1 degree or even ½ degrees.

So what it means is now although we have taken points on the circumference of the circle which is in 2D you have to visualize that this circle lies in a plane in 3D which could be the ZX plane or ZY plane. Assume the ZY plane and the circle is lying in the ZY plane because in 2D and when you rotate the circle in 3D just by a small amount delta theta, a small rotation which you will give, what will happen to these points on the circle is it will create a new set of points in 3D which will be different from its original position. So store these points and these new points which will be generated are new vertices of which will be used to represent this figure and again rotate by another amount. Keep doing this till you have rotated in steps as you can see here by about 180 degrees not in steps of 180 degrees you need to increment by delta theta, 2 delta theta, 3 delta theta and so on in steps and you stop when the total amount of rotation which you have given to this 2D structure is 180 degrees what will result in.

At each point after rotation by delta theta you need to store this 3D coordinates of this point which you have created on the circumference of the circle. And when you create that and you store those points all those points after all those incremental rotations you store you have basically got the points on the surface of a sphere. We have got the points on the surface of this sphere. If we look back this will be the resultant sphere which will be generated by giving this 2D circle a sweep or a rotation about a three dimensional vertical axis. Of course you can rotate about any arbitrary axis. It could be a horizontal axis or it could be an inclined axis it does not matter, still you are creating a operation which I have just described called the sweep operation and with the help of the sweep operation what you have created is a set of 3D points.

A set of 3D points on the surface of the object and these set of 3D points are nothing but the vertices of those polygons which will enclose this particular solid. In this case you have generated a sphere from a circle. So you need to sweep any arbitrary in the general case. When you talk of the general case you can visualize any arbitrary two dimensional structure or a polygon for that matter and when you talk of a sweep you can rotate with respect to a particular three dimensional axis.

You can translate it, you can give it any sort of transformations which you have studied under three dimensional transforms category. You can give a combination of transformations typically a rotation and a translation, a rotation and a scale or you can actually sweep it along in the arbitrary line or a curve or an arc or just simply apply a rotation which we have just seen in this case which helps you to create a sphere from a circle.

A sphere from a circle is the best simple example for you to visualize. It is something like if you can create a circular ring and hang it with the help of a row and just simply give it a twist. We can almost visualize that when the ring spins it will create an impression of a volumetric structure of a sphere and the bounding surface, the spherical surface will be traversed by the arc of their particular ring which you will be spinning about Z axis.

I repeat again, imagine a ring created by a simple thin iron rod or even a stiff rope type of a structure if you can visualize, but the iron ring let us say take the large iron ring with the help of the iron wire create a ring suspend it from the string and then give it a rotation. If you give a very fast rotation not a spin it will create an impression of a spherical structure which you will get.

What it basically means is that the reality which will happen in computer graphics literature for sweep in this case is to create a surface from a circle what you are doing. You are taking points Bresenham's algorithm for midpoint circle algorithm, remember that, take points on that arc and at each step give it incremental rotations or at each point of rotation you just note down or store the 3D points on the arc of the circle in 3D because when you rotate each point in 3D you will get different set of points XY YI and ZI.

For each point XY YI ZI after rotation it will go to a new point XI prime YI prime ZI prime. Give it another rotation and that point goes to XI double prime YI double prime ZI double prime and so on. And this is true for all the points I on the circumference of a circle as you keep on rotating this is what we will create. And here I have shown the arc of rotation. Rotation could be in any direction, there could be any arbitrary axis.

I repeat; in this case I have taken a vertical axis but you can take any axis and this is the structure of the sphere which will be generated. If you look at the structure you would have seen this type of structure in any many places not only in computer graphics literature but you could have also seen that in the case of the map of a globe. When we talk of the latitudes and longitudes in a global map or in a globe type of a structure then what you will basically have is latitudes and longitudes.

Latitudes and longitudes are parallel lines parallel to the equator which never meets and then you have the vertical arcs which basically intersect at the North Pole and the south pole of the particular solid. This is what is created with the help of sweep representation for a circle in the case of generating a sphere. And the vertical lines, the circles intersecting at the north and South Pole in the case of a globe. And of course you have horizontal lines concentricus reducing circles which keep reducing radius as we move towards the north and the South Pole. Imagine the case of a globe, that is the very simple example to visualize this with the help of a globe structure and that is what is an ideal example of a sweep representation to generate a sphere out of a circle. Now, the question is what should be the radius of the circle which you choose to generate this sphere and how much is that the values of delta theta which you have to select to give it incremental rotations as you keep on sweeping. That depends upon two aspects.

What is the size of the structure which you are generating? That means the radius of this sphere, in this case you simply need to choose the radius of the circle that is very simple and straightforward. The other aspect is what this delta theta is. The lesser you choose the amount of delta theta value you will get more amount of set of 3D points which will enclose this sphere or more number of polygons which will enclose the spherical surface.

If you choose a larger value of delta theta you will have lesser amount of points and hence lines and hence polygonal surfaces which define the spherical surfaces.

The higher the degree of resolution you want the greater the degree of approximation you want because what is happening in this particular case is if you even take, let us look back into the slide. (Refer Slide Time: 39:35) If you look into a particular small area, a set of four vertices which will enclose, keep watching my mouse as it keeps pointing to four different vertices which will enclose and give you a particular polygonal shape those four vertices will actually create a polygonal shape. So it will create a plane and at that region that plane is going to approximate that part of the small spherical structure or part of that sphere or the surface of the sphere at that region will be approximated by that polygon. And polygon is always a plane and all parts of the sphere is a curved object it is never planar.

The sphere surface is always the case where it is never a planar surface, it is always curved. So, at any point of time you are always approximating a section out of small part of a curve structure, in this case of course the spherical surface by a planar shape or a plain or that polygon.

Now, how close you want to have your approximation depends on how much you can afford because you can take a very small amount of delta theta what will in fact happen is this polygonal approximation area will become a smaller and smaller and you are having a greater degree of approximation but at what cost. The cost is that you have to store larger and larger number of vertices and lines which are used to represent this solid in the case of a sphere.

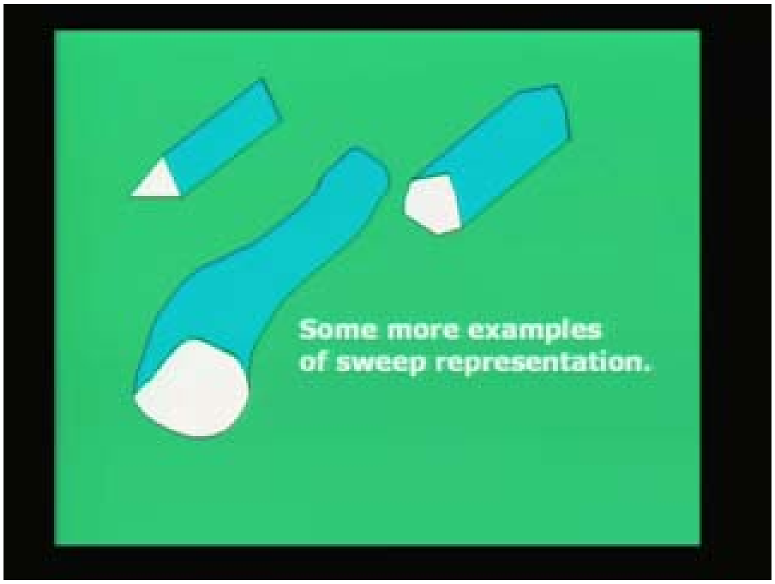
Thus, if you want a very good approximation that means the more closer to the curved object which you mean to represent, more smoother approximation which you want to do then you have to go for smaller amount of delta theta. Larger values smaller amount of delta theta for the sweep which means larger number of points and lines you need to store for the to represent that object structure. Whereas if you cannot afford a larger space for storage for your solid structure because you need to use this solid structure for various other operations like create various other operations complex structures with the help of Boolean set operations then you need to of course find out what is the shading so that you can shade the surface with certain color. We will see this when we talk of shading models or visible surface determination. Algorithms will talk of those and they could become very complex and you need lot of computation time in the number of polygonal surfaces increases quite a lot. So the storage is a requirement and that also enhances the computational time whether you can afford that is the another question so all this will dictate the amount of smoothness or coarseness.

When I talk of coarseness I am talking of this particular example, larger values of delta theta which will create lesser number of points and lines to represent and store for the solid object and in that case that will reduce the number of computations require for creating higher level complex structures or even shading this particular object. But the object structures will require very coarse. It will be a very crude approximation, low level approximation of this curved object, if you can afford it, if you want it, it is fine but if you want if you can afford greater amount of storage, if you have lot of computational storage available and if you can afford lot of computational time in your system then you will have to have a very good approximation done based on your application and then you have to have small values of delta theta. This is true for any sweep, we will see for other examples of sweep representation.

I am just asking you to concentrate on this particular case where I have generated a surface. It is a sphere that has been generated by sweeping two dimensional structures which is in this case a circle. A circle is sweep to generate a sphere in this particular case and if you can afford larger amount of complexity in terms of storage space as well as computational time as the requirement says then in your application you need to have a very good approximation of a curved object and that can be only done with smaller step sizes of your delta theta in this particular case. So it is a compromise between the approximation or the finest or the resolution you need to represent the object and the computational time requirement. So you better choose whether you go for a final resolution and then choose smaller values of delta theta. If you cannot afford it then you make your delta theta much larger and of course approximation.

Thus, we move forward to look into other applications of sweep representation and I have just few more examples for you, generate it to visualize the sweep representation. But I hope I have sufficiently described how you represent an object using sweep.

(Refer Slide Time: 44:39)



These are some more examples of sweep representation. There are three different structures been talked about in the first left example I have taken a simple triangle and the sweep is just a straight line an axis in 3D along which I translate this object. That generates a sweep. So you can visualize that I have these three structures, this structure 2D is a triangle with three vertices and as I keep sweeping along a particular direction or vector in 3D.I can do it in steps, fine steps more complex structure in terms of the number of space requirement.

I can have larger replacements where it read as in a core structure and lesser number of points to store. I can afford to do this in this particular wedge type of a structure generated by a triangle because I am approximating planar surfaces. When I am representing planar surfaces with the help of the sweep in fact I can go to a very coarse approximation where I can afford to take the first position of triangle and the last position of triangle and I do not take any positions in between. So I sweep from a starting position to the finishing position or the end position and I can still define this structure because in this case I am representing a solid which is bound only by planar surfaces or plains.

When I am representing curved surfaces, curved structures or surfaces bound by curved surfaces rather than planar ones I have to go in for a finer representation and then I have to sweep in steps. May be in finite steps if the curvature is more for this solid. If the curvature is slowly moving or it is a plane the step size could be larger. So come back again here as I was saying that in this particular case of the wedge I can have the starting point and finishing point that is just sufficient. That is not true for these structures. Of course in the structure on the right hand side which is till the sweep direction is a line I can have the starting point and finishing. But that is not the position and that is not the case in this arbitrary structure.

As you can see a bony type of a structure where along when I am sweeping I am not only sweeping along a curved line but I am also giving the structure a scale. So it is a simple example where you can visualize. I can create a cone also with the help of a circular structure when I sweep along an axis which passes through the centre of the square assuming an axis passing to the centre of the square I sweep along that I will get a cylinder but when I scale up or down I will get a conical structure. So in this case I can generate various types of solids with the help of structure.

The third case example, the largest one in this case, an arbitrary structure chosen and dragged or swept along an arbitrary curve and when I am sweeping along this curve line I am also giving it a scale. I am increasing it larger and larger giving it a zoom and I am expanding it out and that is what I need. These are certain examples again as I was taking. You can create a pyramidal structure with the help of this square structure.

(Refer Slide Time: 47:55)

When I am sweeping I give it a zoom I create a simple cone this structure is a very simple example like a conical ice cream where I can have this but this structure is a cone where I can take a circular structure, sweep it along an axis which passes to the centre of the circle and when I am sweeping I also give it a scale or a zoom. And I can also have an arbitrary path for the sweep that is also possible nobody says that you have to sweep only about an axis and then sweep to the circle to create a sphere, sweep it an axis along passing to the center to create a cylinder or a pyramid or a cone and all that.

To create a very regular structure I need to have two combinations. A combination of a path or a sweep which could be very arbitrary not only along a particular axis it could be an arbitrary helical structure which we have just seen and when I am sweeping along that particular structure I give the structure a scale. If required I can give that to produce more arbitrary structure. So a very general example in this case as you see in the slide is that the sweep could be an arbitrary path.

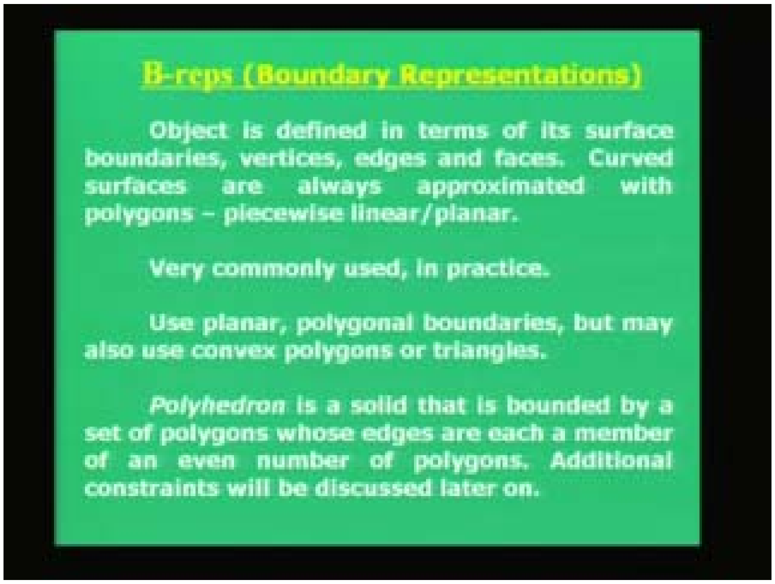
In this case a helical structure and that too not very smooth it could be very arbitrary and you can visualize what sort of the structure which you need. I ask you to imagine, I am requesting you to imagine a structure which will be generated when I am taking a circle in 2D and then sweep about another circular arc in 3D. Let us say the circle is in the XY plane. Assume an XY plane in front of your monitor, the XY plane and Z axis going towards an XY plane and assume a circle in this XY plane and I take a circular arc based on which I am sweeping and that is in the ZX plane. Z is towards you, X is on the right hand side, Y is the vertical axis let us say. So assume a circular sweeping arc, the path of the sweep to lie on the ZX plane and the circle in the XY plane. So if I provide a sweep along this you can visualize that I have a ring type of a structure but I get a structure which is called a toraidotorus.

That is the structure which I can generate various types of models using sweep both regular and in fact very highly regular structures. The only thing I should have is I should be able to represent an arbitrary solid with the help of a sweeping arc or the path of the sweep you have to define. And also you must define the structure which you need to sweep. In fact for the structure which you need to sweep then the path along which you need to sweep and then when you are sweeping the structure along the path whether you do give other transformations like scale the path could be a simple translational path or a curve linear path based on a rotation or a combination of rotation and translation. It could all happen and you can provide scale and shear as you move along the path. These are the combinations of this 3D transformation which we have seen earlier which is necessary and is all coming back now to provide this sweep representation. This is how you provide a sweep representation.

I have just given one example with the help of the circle and the sphere but I have given also other illustrations of pyramid, the cone, the wedge and of course I have visually asked you to imagine in the case of a torus. And of course other arbitrary structures and one or two examples I had given. So this is how you generate. And what did you store as an output of this sweep representation is the case when you stored the XYZ vertices generated by the sweep of the structure so that is how you generate solid with the help of sweep.

In the remaining time available to us in this lecture we continue solid modeling and constructive solid geometry discussions today. As well as in the next class we will go to the next representation which is probably the most popular methods of representing solids called boundary representations or B-reps. Boundary representation or B-reps we talk of an object which is defined in terms of surface boundaries, vertices, edges and faces. This is nothing of course new because when we talk of a solid the solid is also defined and it is bound by surfaces and the surfaces are again defined in terms of edges or faces and edges and faces are again defined with the help of vertices so you know that. But this representation is different than the case of sweep so curved surfaces are always approximated with the help of polygons.

(Refer Slide Time: 54:18min)



We talk of piecewise linear or planar approximations of curved surfaces with the help of boundaries and these are very commonly used in practice. This is nothing new we discussed about solid object in the case of a cube which is defined with the help of surfaces. There are surfaces which bind this particular solid on the top of course in the cube. We had two surfaces, one on the top and the bottom, two on the left and right, two are on the front there were about six surfaces and there were eight vertices if you remember and there were twelve lines which define. So try to imagine boundary representations closer to that particular case of a cube where the cube was bound by certain surfaces, polygons, lines, edges or vertices. We use the term faces also here to represent solids.

Unlike in the case of sweep representations where we used a sweep to represent curved objects, here in the case of boundary representation always go for a piecewise linear or planar approximation. Sweep also does that I must admit here. But always in this case we are talking about polygons which approximate curved surfaces as well. So we use planar polygonal boundaries but may also use convex polygons or triangles to represent in the case of B-reps and we define a new term based on polygons where we define a term called polyhedron.

A polyhedron is a solid that is bounded by a set of polygons whose edges are each a member of an even number of polygons. This is interesting, we will see and we will discuss of additional constraints later on in this class. But we will wind up a lecture here today with the discussion of what is called a polyhedron. That is the new term which we use to define a solid under the category of B-reps or boundary representation of solids where we say that a polyhedron is a solid of course it has been bounded by the certain surfaces. It is a solid that is bounded only by set of polygons and those polygons are again defined with the help of edges and those edges are each a member of an even number of polygons.

So we say that we can have edges which define the polygons. Polygons are the planes or faces which bind the solid. But in this case the first restriction which we put is that each edge of such polygon should be shared or it must be common to more than one not only more than one but we should have an even number of polygons. We will see how that constraint holds good with the help of example when we visit the next lecture on solid modeling after this. Here again we wind up the lecture today with the definition of polyhedron once again where we say that the polyhedron is a solid that is bounded by a set of polygons whose edges are each a member of an remember this even number of polygon edges are each a member of an even number of polygons and that is the first constraint we put on boundary representations.

We will see other constraints with respect to the edges that is the lines of the polygons and also the surfaces of the polygons. And the vertices we will put additional constraints to define what we called remember the beginning of lecture a valid solid. A valid solid should have certain constraints based on which we can say that is a valid solid object and not a open or a closed one and that constraints will define with the help of planes, lines, edges and vertices. The first constraint which we have seen today for the polygon is that an edge must have or must be shared by an even number of polygons. We will look into the additional constraints later on in the next class when we meet to discuss further on boundary representation for representing solids. Thank you very much.