Computational Geometry

Prof. Pankaj Aggarwal
Department of Computer Science & Engineering
Indian Institute of Technology, Delhi

Module No. # 14
Shape Analysis and Shape comparison
Lecture No. # 01
Shape Representation

So, I am giving the last two lectures. And, I will try to stay away from being too technical in these last two lectures. We have covered many different concepts. And, the goal is to bring some of them together and also to give you some motivation, what you have many listed that, what you have learnt, what it is good for. I have chosen one underlying theme to illustrate some of the techniques. And also, you will learn some of the new algorithms in the next two lectures. But, I think, this is something that is becoming important, increasing the importance. And, there is some and in many different communities, research communities, you will see research papers and various conferences on these topics.

So, one way of to motivate it, is that, suppose you want to create a virtual world or virtual physical world, whether, it is on the internet although, many other application that armies are very interested in doing it for training exercise, also Government and commercial entities. So, if you want to build a virtual, physical world, then, which consists of a, has very much lot of different geometric objects and then, of course there is much more to it that just pure geometry, but since, we are focusing on Geometry, I want to talk about the geometric aspects of virtual world.

So, first question is, how do we put the things together? And, where they are done? Is basically, what we have in the physical world? You have various sensors, you have various machines, devices or mapping devices, we use them and try to sample. Yeah. So, examples, sort of vary, from let say, three dimensional model generators and for example, there is, I do not know if you have heard of it, I mentioned in this class, Also there is something called Radar technology.

So, what you can do is, in the aeroplane or in some vehicle or ground vehicle, you can put a, what is called a laser finder. What it does is, it is sort of sensor laser scan and it gets the laser back and by the reflection, the distance can figure out how far it is. Object is from where? It is like radar, but it is laser frequency, laser stuff.

And, in different forms, this is been used a lot to generate three dimensional models. So, for example, all the... So, for example, I just give you an idea, the latest version of F-16, which is one of the US fighter planes is going to have eight video cameras, then four radars to collect so much data. And, but they want to... You collect all the data. And once, you done few times and you collected very high resolution three dimensional model. And, for them it is very important to have the three dimensional model, then you can plan and figure out how to do better feel management. Now, this is military application, but you can think also very commercial applications or civic applications. For example, more and more, what is happening when you are in disaster zone… Robots which have laser finders, they go and explore the three dimensional world. They have, which has damaged area, and then you figure out what is going on there. And, you can do the transcrimination much more effectively than sending human beings and they cannot reach all those places. You can think of very F minus applications in the commercial world.

For example, all the new airports that have been built, first you build a very detailed, high resolution three dimensional model, that is true, also for planes. For example, Boeing 777, which was designed to a single wide level, the distance model was built before anyone started thinking of manufacturing it. So, all these sort of basically, what is sort of is happening, building these huge three dimensional geometric models and then, geometric virtual worlds, such you are developing. So, what are sort of question comes out, how do you assemble such a world? Imagine the act of composing any music. Am I right. First, you have notes from or text you have letters, words and etcetera. You have hierarchy of representation, where higher level of abstractions and the same thing, happens here.

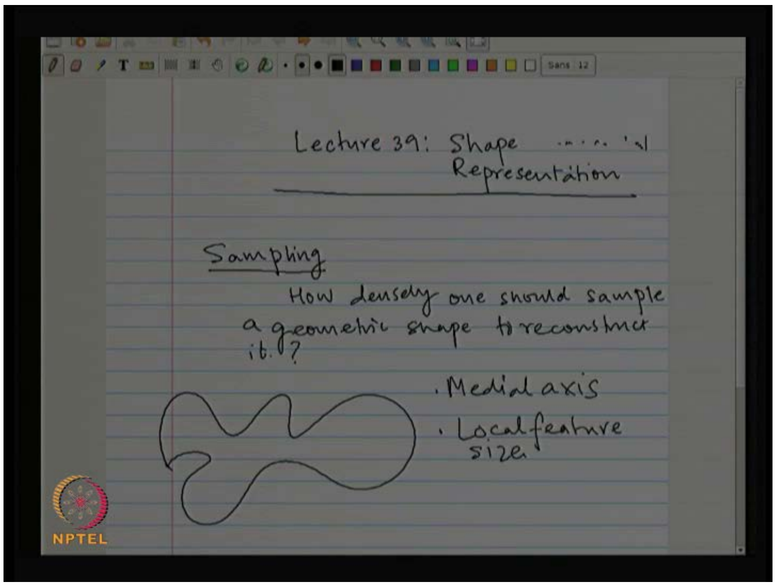
Then, the question comes is, what is hierarchy there, in this geometric representation? So, I will not address all those questions, but I want to touch upon some of these issues that for has to deal with.

So, first thing comes is that how do we start with very basic objects. And then, you build a database of objects. And, once your database objects built, you can club them together and then start building more complex objects.

So, there are various things that you need, tools that you need. One is how you represent a three dimensional object… Represent a shape and then we build is and then we need a hierarchical representation of the shapes. The second thing you need is, how we compare two shapes. Am I right. Because you want to talk about like, I need this shape, I need that shape, you need some mechanism, so you can compare two shapes.

Like, for complete dictionary, when you type in Google, you type some word and then you get something like, compare the two strings. Right. Same thing, you need to do in some model, for some way of comparing shapes. So, once you have compared shapes, the next stage comes is, you want to classify shapes; that all these objects are similar like, clustering. Right. So, you want to do clustered three dimensional objects. Then, what comes is that is basically, is a searching. What is called indexing databases? That is, you may have some database not trying to build a world, I need some object like this, so, but you do not want to give the whole representation of the shape, you want to give some keywords. Am I right. And, then you want the database to search and give you the objects, which represent to some world. Now, existing data, most databases, you type a word and then it gives you the shape of, which the word. The ability to search in the three dimensional world by representing, by catching a three dimensional object is returning the objects that are similar to this. That... Stage, none of the commercial products give you, does not do a very good job of doing it. So this is also called, what is called the Virtual or Visual information or Geometric information. How do you build this geometric world?

(Refer Slide Time: 08:47)



Lecture 39: Shape Representation

Sampling

How densely one should sample a geometric shape to reconstruct it?

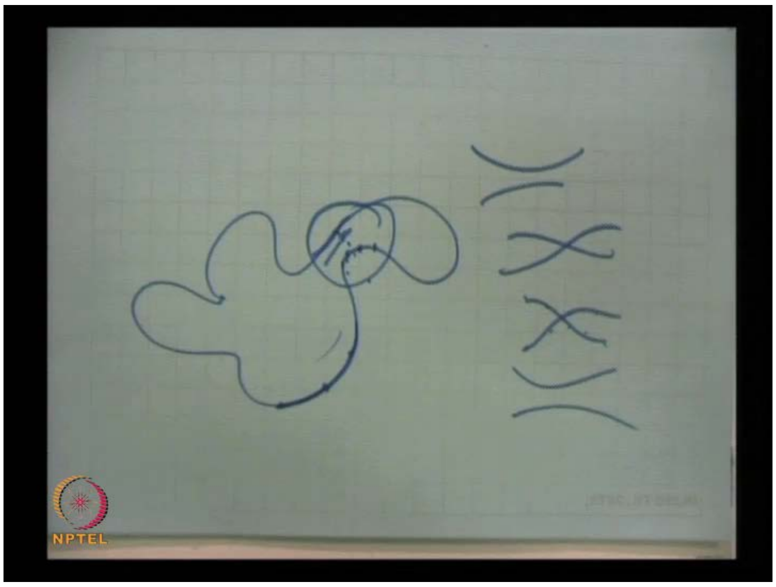
· Medial axis

· Local feature size

So, in these two lectures, I want to give you some very basic sample of very few of this questions and give you some idea of, what is being done there. So, let us talk about the very first thing is Shape representation. So, the very basic... Is, suppose you have some device, you have laser finder or some other way of measuring or sensing device, then the question is you want to Sample. What you do is… Let see in this room, you put a laser finder here, you will scan all around and you will sample parts. The first question comes is, how does we should sample? Am I right. That is the first thing; sampling question comes. For example, all you know, if you take any, ... Signal processing is called that if you want to reconstruct the signal, then it has to be twice of frequency, highest frequency, to reconstruct our signal back. So, what is the signal notion in this case, geometric nation or geometric world? That, how does we should sample. Ok.

And, that basically leads to, so first let us talk about sampling. So, how densely one should sample the geometric shape to reconstruct it? Even, this very basic question is not fully understood. And, I will not give you all the technical and what is the best result, I just want to give you an idea.

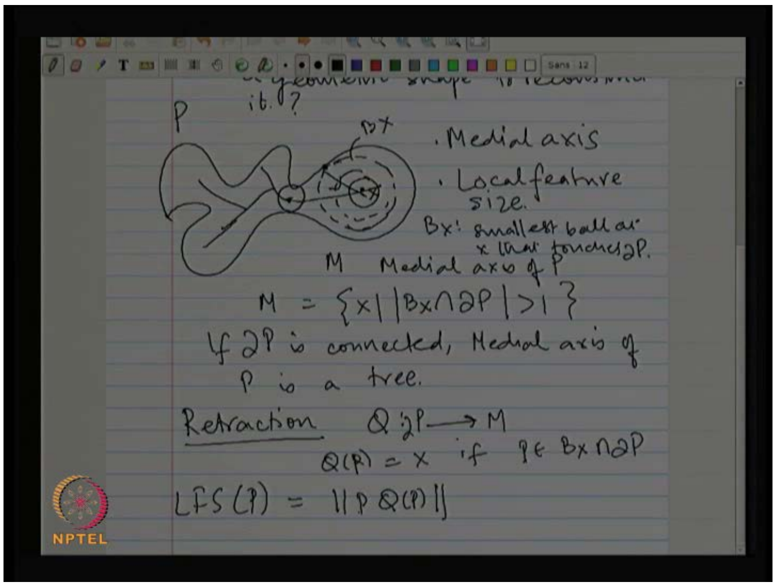
(Refer Slide Time: 09:20)



And, this is closely related to the notion of one more diagram that we have seen. So, there is something called Medial axis. So, let me just give you some idea. For example, suppose we have an object intuitively, without sort of being mathematically precise, you do not have to sample. I should do it here. So, intuitively, what is…? In this region, you do not have to get the sample too close. Am I right. You can get away with this parts sampling. But, when you come to this region, here you have to dense sample very densely because if you do not sample densely, you will not be able to construct it, where the object looks something like this or you have something like this shape. Am I right.

If I just give you some points like this, then you do not know whether the curve is going something like this, or it is going something like this. So, you should be able to distinguish these two. So, question is, how do you distinguish these two cases. For that, I need then two notions, what is called the medial axis, which is generalization of Warnier/Orr diagram and based on that, this notion of what is called Local feature size. So, Medial axis... So, we, Warnier/Orr diagram there will be at least, I define it in the class; I defined it for finite point sets. Then, you can think about Medial axis. Medial axis is a generalization for an infinite point set of curve, for other shapes and surfaces.

(Refer Slide Time: 11:07)


it. ?

- Medial axis
- Local feature size.

Bx: smallest ball at x that touches ∂P.

M Medial axis of P

$M = \{x \mid |B_x \cap \partial P| > 1\}$

If ∂P is connected, Medial axis of P is a tree.

Retraction $Q: \partial P \longrightarrow M$

$Q(p) = x$ if $p \in B_x \cap \partial P$

$LFS(p) = \| p\, Q(p) \|$

So, let us look at, suppose you have this, let us call object. Let say this shape, I have P. If I take a point inside, to the general point inside this x and I start drawing a disc centered at this point x. Am I right. In the beginning, it will be empty; it lies completely inside the object P.

As, we increase the radius continuously; it will start expanding. And, at some point, it will touch a point, boundary of the P. Am I right. Now, if I do it from a general point, what will happen? It will just go and touch at one point. But, there will be some points inside that. If I do this, it will touch two points. Right.

For example, so, if I take, let say I draw it here, let say if I start increasing the disc, it may touch this. So, the points where, I will not define again, when can I make this notion more mathematically precise; so, we can talk about, what is a limiting ball? What is a smallest empty ball? Which is a smallest ball at which it touches the boundary? It touches boundary. If it touches the boundary at two different points that point is said to allow in the Medial axis.

So, Medial axis of P, so let me define this ball as a B x, this notation. So, the smallest ball centered at x that touches the boundary. So, B x; that is a boundary of P. Medial axis of ball is of P is set of point x such that, if I take the B x and I take the intersection with the boundary, it is greater than 1. Right. So, I will not prove it. If you draw the medial axis, it would be a kind of one dimensional network.

So, it will be basically, what will happen? It will look like, something like this. So, this is, if you think about Warnier/Orr diagram, if you look at the edges and the vertices and look at… that set of the… leads to the medial axis of a finite set of points. And, here, that is what you will get. Some... If you have simple polygon, which is the case, all like, boundary is simply connected, then what you can argue? Then, that the medial axis will be a tree.

So, if P is a simple polygon, if P is or if delta P is connected, then medial axis is a tree. Is this making sense? Or, any questions about the definition of medial axis? Now, it is not easy to compute medial axis. Now, in two dimensional, it is no too hard. So, if it is a polygon, you can compute it; not it is like algebraic ... It becomes, starts becoming more complex. But, this notion that I have defined of medial axis is not only in two dimensional; if you have a surface, some shape three dimensional objects, you can define the medial axis. And, there the problem is still open, how fast you can compute the medial axis. And, there is no robust algorithm; so that, both theoretical problem that is still open, how fast you can compute the medial axis, what is a bastion algorithm. Fine. I do not care. If you are a practitioner, you do not care about the worst case studying time, but you need robust software to compute medial axis. And, that is still not, does not exist.

All the works, basically what they do is, do some kind of a patch work at some level. They compute the… patch it and do not compute the exact medial axis. And, the reason being that, even if you have a polygons, polytopes that, that is it; some piece-wise linear from the triangles, the medial axis may consider by cubic patches.

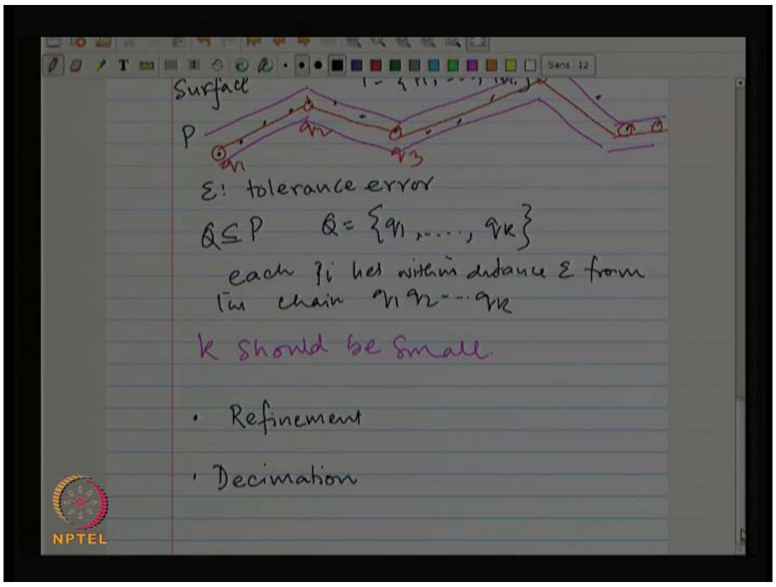
As a result, algebraic complexity, the bit manipulation, you have to deal with that becomes quite a nightmare. And, that is a challenge. So, that was a slight detour. Now, if I have the medial axis, then what you can do is, there is something called Retraction. In Mathematics, if there are no sharp corners, if this shape is smooth; mathematically smooth means that, you think as a curve, all the derivatives adjust. Then, what will happen is that, those for every point on the bound of the polygon, there is a unique point on the boundary axis; so that, B x will touch that point. Right.

So, what you can do is, you can do One-to-One mapping. From every point of the boundary, you can map it to the medial axis. That is, from that point, if you touch the boundary; rather, ball B x will touch at that point.

So, the retraction that is basically, if you want to define that phi is from P to the, boundary P to the… I define, did not use the notation of medial axis. Let me call it as M, then phi of P is x; if p belongs to B x, so that, it is a mapping from the boundary…. And, the Local feature size which is L F S of P, is equal to the distance between the x and P and phi P.

So, that is the distance between the medial axes to the point where it will track…. And, now if I go back to here, if I look at, what you will notice is that the Local feature size here, will be pretty small because medial axis will pass through here and the place is here, and the local feature size will be quite large. And, that is the reason that I have to do the sampling denser here and I have to do sampling.

(Refer Slide Time: 19:18)



ε: tolerance error

$Q \subseteq P$ $\quad Q = \{q_1, \ldots, q_k\}$

each q_i lies within distance ε from the chain $q_1 q_2 \cdots q_k$

k should be small.

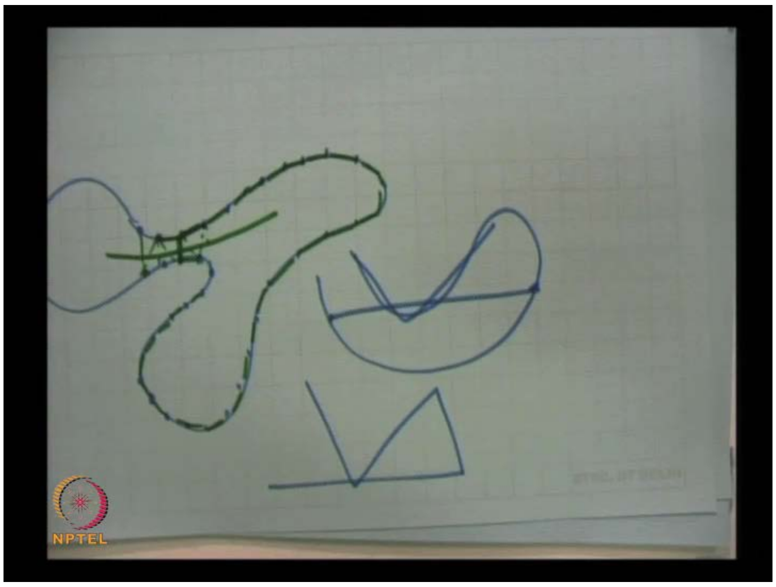
- Refinement
- Decimation

So, what is known is that, now, so sampling density as you can assume, is inversely proportional to the Local feature size P. So, where you have the local feature size is small, you have to do dense sampling. When, away from the Local feature size is large, you can get away parts sampling. And, this is basically a question. You have signal high frequency components; you have to do dense sampling. When, you have local small frequencies that you can get away by doing various sparse sampling. So, that is about sampling.

So, that is out of… Really very good question…. So, that was, I was going to come to next is. so, if you are very conservative, you compute the minimum value of this. Am I right. That is the maximum density you need to do. So, if a signal, I ask a question, when your signal is coming, at what frequency I should sample? If I know what is the maximum frequency, if you say that conservative, for example, at that frequency, so, what sort of…?

So, before I answer your question, let me put a question on the table. Let me, just sort of make another statement and then I will answer. I will elaborate on your question. Now, why… Again, I will not give you the proof, which is much very technical. But, let me give you an intuition, why that is a case, why this makes sense, why does this statement make sense. Here is an intuition.

(Refer Slide Time: 21:27)



That… suppose if this is your… and if you do the sampling, and suppose you compute the Delaunay triangulation of these points x, so let say what I did was to sort of address his question and what I will do is, suppose I need a polygon, of course …gone through the polygon that is why I am trying to do the sampling reconstruction. But, suppose I have the polygon, I did the some more, some oracle to meet what you have to sample at least, at this density.

So, suppose I know what is the smallest Local feature size of this polygon and with that keeping in mind, I did the sampling with that, let say enough density such that, Local feature size condition is met. So, I have done, that is enough.

Now, I compute the Delaunay triangulation. What I am saying is not quite true, but that sort of, gives an intuition because one has to be more careful than what I am saying. But, suppose I compute the Delaunay triangulation of these dense point sets, then what will happen is, you will have edges; you will have all these edges because if you can convince yourself with the density is small enough, this edges will, there will be empty balls passing through these two points.

Basically, but, then there will also be some cross edges, something like this. But, the way I have chosen, the way I have defined the Local feature size because what you know is that there is an underlying medial axis passing through this. And, the way the Local feature size is defined, the length of this edge will be at least twice a Local feature size. And, the length of these edges will be less than the size of a Local feature size.

So, what you can do is, you can throw away all the edges which are bigger than the Local feature size. And, keep the one through size length is smaller than the Local feature size. Then you can, basically what will happen is, if you do that you will reconstruct the shape back.

Now, I should again calculate same, what I said is not quite true; because one has to be more careful because if I do what I said, I will get some spurious edges. But, that sort of… That is too technical and that is idea. And, that is how you will reconstruct the shape.

So, what you do is, you do the sampling, dense enough sampling. You get an estimate what the minimum Local feature size will be. Based on that, you decide the sampling rate. And once, your sample have got dense enough samples, then you compute the Delaunay triangulation and keep right set of edges, small edges and throw away the long edges and then you can get the shape back. Ok.

Yeah…medial axis… Medial axis is a connected network. No, it is not obvious. One has to sort of argue. Now, in order to see that, I have to sort of to make some assumptions because the way of thinking about is, if you think about this point set, disc of as infinite dense of point sets, finite point sets and the Warnier/Orr diagram; when you look at the Warnier/Orr diagram, then if you look at the edges of Warnier/Orr diagram they found to be connected. Right. And, … The way you argue is that you sampled points and then you take the limit, then because what you know is that if this is finite point sets, then the medial axis is connected and then you just keep on taking the limit you reach and it remains connected. But, you are right, it is not, it requires a proof. It is not obvious. Ok

So, that is the one does it. Now, this is a small polygon. For example, there are, so there will be the number of projects. Those projects are at Stanford. What they did was, they went to Italy and they scanned through laser scanner; that is, scan all the old Mythological sculptures. And, each of the billions of polygons, points because they have to do; because if you have seen those sculptures, they are quite detailed. Am I right. They have hair, nails and toes and all the details. In order to capture all those details, they have to sample very densely. And now, if you have billions of points… compute to do it. So, outputs has its, has huge size.

And, if you want to do any processing on them after that, if you want to do interaction and walk through in such a museum, which has all these virtual museum, it is going to be very slow because wandering among these billions of polygons for each of them. So, then what the question you would like to ask is, ok, I did this.

Now, you reconstruct the shape and the precise statements that we have made, if you know, some of you know, I will write an apology. Then, what you can sort of, the statement one can make is the algorithm I described with some… the shape that you get is a homeo topic to the original shape.

It means if the structure will look like the same. And, you can also measure that is, if you, how much sampling dense you did, you can also say that how much geometric error you will have; because if you done the dense you will not get the actual curve back. But, you will get a piece-wise approximation of that curve of the shape. The error will be at most epsilon. And, you can define what the value of epsilon...

Now, it has too many points. So, what is the next topic and that is again the topic is whole research. And, active research is, can I think away from the points because this goes that to the question he asked, is that Local feature size is different for different places. So, I do not need to keep all the points because I did the density sampling with respect to the smallest Local feature size.

So, the places where the local feature size is large, I should be able to throw away some of the points. So, that leads to the whole area what is called curve simplification or surface simplification. So, for example, this is very different application; where it is being used a lot, is the following. Now, every car has GPS; like, cell phones have GPS. And, GPS is easy to track people. Right. So, you want to… and the GPS you want to again want to sample very densely; will GPS be a position? Now, if I connect them, I get a trajectory. Now, unless the trajectory is very changing a lot, if the trajectory is more or less the same, there is no reason, I need to keep all the points that I have sampled. I should be able to throw away most of the points.

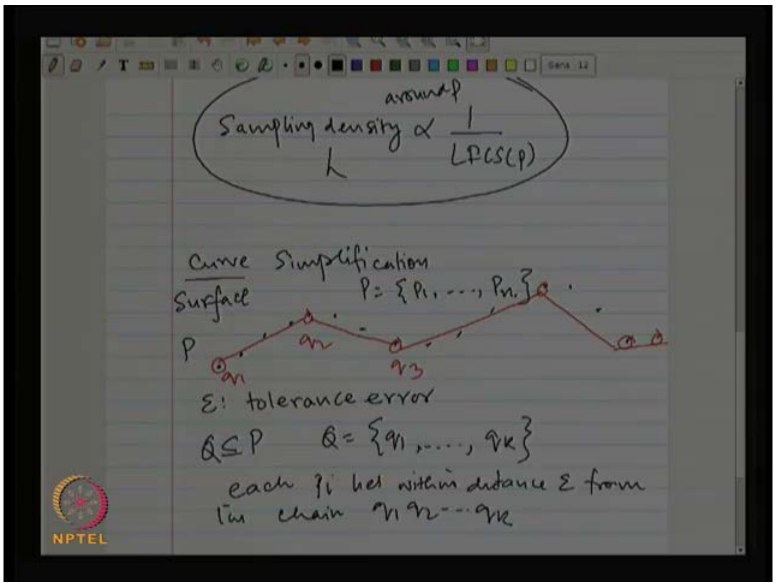
So, I do not know, whether it is happening in India. But, I know that happens in many European countries and US. What they do is that the lot of people who commute a lot in the cars, every sort of day, they commute or they go in a same pattern. Now, the question is that you want to do a road network. Generate a road network and you not only just want to get the shapes, but also want to know what is congestion on the street, what is the speed limit on the average during different type of data. We want to get your knowledge.

So, what they done is that, basically I will pay you some money, so, since you are going, let me collect your data from GPS, let me use your GPS and collect the data. So, I have the people who are going through various commuters and they collect those data and from that, they are building the road network. Right.

Now, do not want to show the road network as those point sets of the… because you want to generate the roads. Now, I have connected those points and I got the trajectory, how do I build the road. And, you do not want to throw away the most points, except the few points that you want to keep. So, that also comes to the question of curve simplification. Other… What I am saying is this is a form of data compression. There is an inherent conflict between accuracy and efficiency, the more information, more data you have; you have more accurate representation. But, it takes more time to process it.

So, how much loss I am willing to give to speed up my algorithm? So, the whole different ways of doing it, the second is because GPS… again let me go back to GPS, for example. When you are going on a road with GPS, GPS has some of statistical error, some measurement error, so, even if I you are going on a line, the points that you will get will not be a point on a line. It has some error. And, so ... What you like to do is you like to clean it up and give you a straight line. So, again that is, basically I want to simplify the curve.

(Refer Slide time: 33:14)



Sampling density $\underset{h}{}$ $\propto$ $\overset{\text{around } p}{\frac{1}{LFCS(p)}}$

Curve Simplification

Surface

$P = \{p_1, \dots, p_n\}$

ε: tolerance error

$Q \subseteq P \quad Q = \{q_1, \dots, q_k\}$

each q_i lies within distance ε from the chain $q_1 q_2 \cdots q_k$

So, next in topic in line is curve or surface. Let me talk about the curve first and then I will talk about the curve simplification. So, problem is the following. Will be so. Either you have a, you can think about that I have sequence of points. So, and what I would like to say is, I will give you tolerance error, epsilon error.

What you want to do is that, so this is at P, choose a subset Q of P, and so, let us say P is P 1 P n. Q is let us say q1 q k; so that, if you connect, let say, so this is q 1, q 2, and so on. And, you connect it, let say, then, so when each P i lies within the distance epsilon from the chain q 1, q 2… q k. Right.

So, basically what it says is that, every, all the points lie very near the polygonal chain. So, suppose if you have draw this, I am, see, if you draw this buffer zone of a error epsilon around this chain, all this points should lie inside this, in this buffer zone. Am I right. And, k should be small, so because this is a… You can, of course talk about we determine, I give you set P i, give you epsilon, return me the smallest size of q, so that all the points lies within the distance epsilon from this polygonal chain. So, this is the goal.

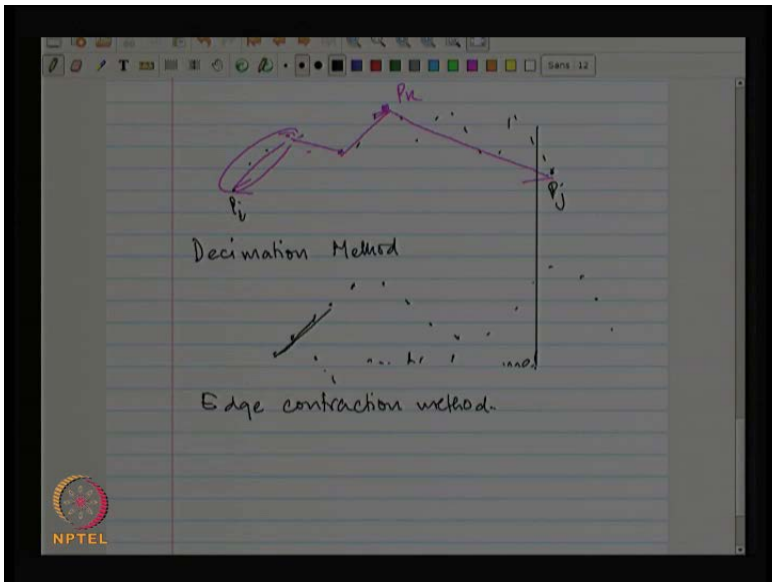
Now, there is dual problem also. That, you may have fixed budget. Am I right. Because let say that I can give you only three points, you can store only three points and then find the three points, so that errors are as small as possible. So, this is called minimum k problem that I give you an epsilon, find the smallest k. dual problem will be, I will give the value of k and find the, choose the k points, subsets of points, input points; so that, if you connect them, then the errors is as small as possible. Ok.

So, let me give you first two popular techniques that I used. And then, I will talk about what is theoretically known. So, there are two general techniques that I have used both for curve simplification. And, those methods also extend to surface simplification, when we talk about three dimensional.

So, the two methods or called approach is refinement and the second one is called decimation. It occurs to be that, also say that these problems also arrives in machine learning because what machine learning people also want to do? One of the problems they have is, you have lot of examples and from these examples, you want to generate a very simple concept that capture all the examples, the classification problem… Lot of them do this. And, I will not have time to talk about relationship, but a lot of this also… so, this simplification problem.

Ok. So, as the name suggests refinement is that, what you do is you start with very coarse representation and you refine it until the errors becomes at most epsilon. And, the decimation you start with very dense representation and slowly, gradually keep on coarsing it, until you cannot do anymore because errors are more than epsilon.

(Refer Slide Time: 36:55)



So, let me give you the refinement method. And, this is a popular heuristic method called… There are many of them, but I am going to just give you only one of them. That is called Douglas-Peucker algorithm. Douglas-Peucker is pretty simple. So, here is a set of points, example, you have what you do is a recursive algorithm, which may do the following.

Take the first point, let say P 1. Let say P i because I am going to define recursively. So, you have subsequence of points P i to P j; Connect this, then try to connect them as a single segment. And now, check if all the points between P i and P j, they are within distance epsilon from this segment. Ok.

If that is the case, if all of them are within the distance epsilon, we have nothing to do. You can return that segment. Now, in this case, they are not. Then, what we do is and then you need to break this segment. One Segment is not enough. Then, you need to sort of, need more than one.

So, you need to split this point set x at some point and what do you think is a natural choice? What do you… median? Yeah so intuitively… you can come out of different method, but is a, well, since you are looking at geometric error, so take the point where the geometric error is the most. So, take away the point, which is farther from this segment.

Let us call this point. Let this be a point and do P k. And now, you recursively solve this problem at this P. So, you do not add this segment. You now, you try. Let, I now continue doing it. So, suppose this is also not good enough, then this is the point or may be here, you have the left side; you had done because this is within distance of epsilon. So, you stop here. But, let see here, you will take this point let say and these two segments are enough. And, recursively you will do it here. Am I right.

So, that is the algorithm. It is pretty simple algorithm and it is used quite a bit in practice. It is one of the most popularly used Math for curve simplification, refinement method.

What is the running time? So, k well, I do not know the value of k. Am I right. In the worst case, k can be as large as… If you are really unlucky. So, you are right. So, in the worst, if I do ... Implementation it will be because at each step, what I need to do is indeed to steer out, look at all the points within the distance, see whether it is in the distance epsilon. So, you spend a linear time.

Now, if you are unlucky and recursive call is very unbalanced, it is like quick sort. Then, the worst case running time can be quadratic. If file output size is k, it could be k times n. Right. Now, by being more careful and using some data structures, which I have not talked about, you can make sure that the running time is always n log n. But, it is not known what is the lower bound? So, it is not known, whether the lower bound is n log n or whether one can do linear. There was a paper, which claims n log star n algorithm for this problem. But, I believe it has, it never appeared in the journal. It was in a conference, I believe it has some technical issues.

So, it is not known, whether you can implement this algorithm, better than n log n times. n log n is not hard. It requires some, let me put this way, you can do it. If I give a final problem, you should be able to do it; exam problem because it does not need more than what you learnt in this class. So, this is an example of refinement method. Any questions?...

No. no. yeah. Other issue is that, which are going to say, come later is the algorithm does not necessarily give you optimal. What the minimum value of k because you can ask an optimal question that for a given values of epsilon, what is the smallest value of k.

And, you can construct an example, where it will do really badly that there is a constant size output, but that gives you the almost linear. So, it is not only optimal, but it is not found near optimal, in the worst case. But, in practice, it does very well. And, that is why people use; because it is simple to implement it. You can also implement it in a more sort of, it does not require any space much. You can do it in online manner. You can do it in distributive manner. There is lot of nice thing one can do that. No. First I will finish them. I will give you an algorithm, which is recursive optimal.

They are some versions of that are not known to P; which, if I have time, I will talk about it. So, let me talk about decimation method. So, decimation method is, you want to start with all the points and slowly throw away one point after another. So, these are all the points. Am I right. If I want to throw points one after another, what do you think we should do? So, the error is at most epsilon. Then, you want, you can throw it away. Now, since if you want to do incremental error… In what order, should I throw away points? There may be many points, they may satisfy this. That is one way of doing it. We do some, start from the scanning left to right. Example, there is one way of doing is, you start from left to right and let I start from here.

And, the question is, whether I need the second point. So, as you said, you connect to the third point and you see the error is here. I can sort of, throw it away. Now, if I can throw it that is fine. I restrict to this. Now, ask the question, I want to do this. So, I can do this, you can…So, that is one way of doing left to right. Another way of doing it is, some sense, think about that reverse of this origin Douglas-Peucker… What you do is, what do you think about is, here what I did was I have the segment, I split the segment into two segments. That is what I did. Am I right.

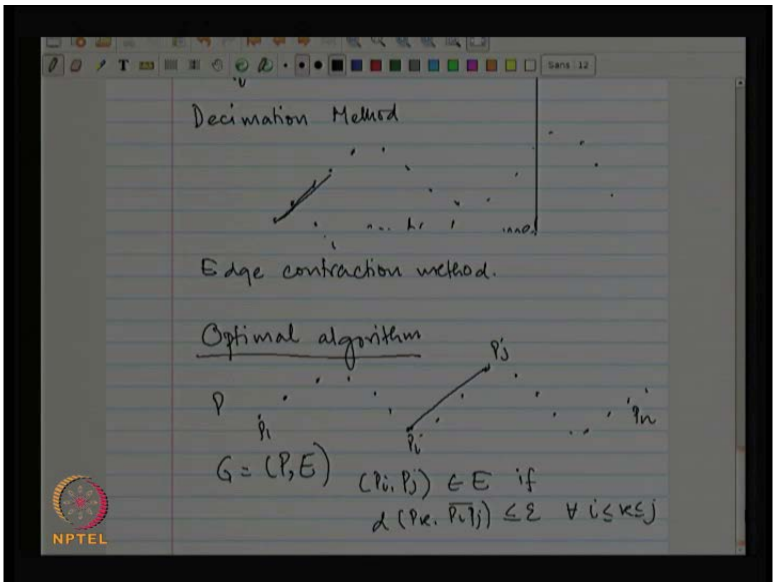
What you can do is, you can take two consecutive segments and if can I fuse them into a single segment. And, you choose them if the error introduced is not more than epsilon, it is so, that is what you can do. So, that is called Edge contraction because you are taking two edges, contract them to a single edge.

Now, again in the worst case, this may take n square time. And, but again you can by being a little more carefully, you can implement a n log n times. Again, this has a same problem as a Douglas-Peucker… In the sense, that is does not guarantee that it returns anything near optimal. It may be returned again pretty bad output, in the worst case. So, but both of these heuristics are used quite a log factors.

So, let me sort of say, let me call it Optimal algorithm. Now only, I should say I have been doing some cheating here. If you look at mine, if you look at the pictures are drawn here, all these. For example, if you look at the disc picture that I drew, then if you look at this picture and look at the third picture, these are all x monotone curves. But, the picture that I have drawn here is highly non monotone. And, no one is asking the question because and it is deliberate. It is not accident because when things are not the monotone, you run into some problems that I will address later.

So, for now, I will continue to assume that the points are as x monotone. So, basically if you increase the index, x coordinates are monotonically increased. So, what I am doing to do is the following.

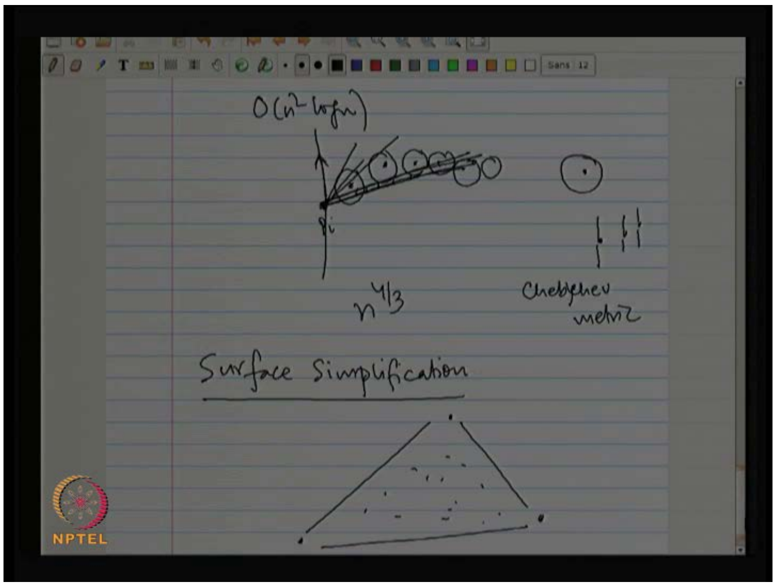
(Refer Slide Time: 47:11)



So, again this is at P, it starts with P 1, goes all the P n. So, what I am going to do is, I am going to discuss at a graph, whose vertices are points themselves and your set of edges. So, pair P i P j belongs to E, if distance from P k to the segment P i P j is lesser than epsilon for all. It me, what I am saying is, basically if this is P i, let say this is P j, you have an edge in the graph, if all the points from P i to P j, they are within distance of epsilon from the segment. right.

So, we construct the graph. Good question. So, what I mean by Optimal algorithm is, you tell me the, given an epsilon, return the polygonal chain with the fewest number of edges; so, optimal in terms of k. Yes. Now, if I construct this graph, what should you do to reconstruct the curve with the…? How do you solve the problem? Pardon. Say it. I cannot hear you. So, maybe you can say, somewhat like, yeah. You cannot do a programming, but you can give me one line. You can reduce to see one. Yes. You want to find the shortest path, when even number of edges from P 1 to P n. yeah. That has minimum number of lines.

(Refer Slide Time: 49:52)



So, find the shortest path from P 1 to P n in G. So, how much time will this take? What is the running time of this algorithm? Yes. That is right. So, the worst case running time, if I do naively, it is n cubed, as you said because the n square P s. I need to check for each of the...

Now, one way I, so n cube is pretty straight forward. Am I right. Is it clear to everyone or I should explain why n cube. So, n cube is straight forward. Now, can you, did you see to n square log n or n square of n actually, you can did, you see to n square log n, I will do not care about the log factor.

I just want to hear some idea that you can bring it down to n square. So, somehow you want to construct, do not want to construct; so, think about the following. Right. Now, what I am saying is, the way you get n cube that you have constructed the n square edges. All right. And, you are saying right now the linear time of each edge, but I really respond to linear time construct to each edge.

You think about all the edges, let say, which are going from P i to the higher side; P i to P i plus 1, P i to P i plus 2,… I am repeating lot of work, if I am going to spend on linear time. I should be able to reuse that work I have done. So, how would you do that. So, think about the following. Let me give you a hint. So, here we starting with P i and here these points, what is it mean that, all these points lie within distances of epsilon; so, from the segment, which I try to draw.

As a following question, what is the set of segment or set of range? So, I am thinking about things, which are going to the right. Am I right. What is set of range, for which the distance from all these points at most epsilon. Am I right. Because that is what you want to ask the question; because the condition is that you want to connect them by segment. If and only if, all the points they lie within the distance of epsilon.

So, what you do think about this? Draw the disc, it is epsilon around each point and if the line that starts from here, think about a line, so that all these points within distance of epsilon from that line. What is that mean? That, entire disc should intersect the line. Now, if I ask the following question, look at the line, what is the set of lines that intersect all these discs?

So, for example, think about the following this way. They may not be such a line, it is an empty set, but, suppose of such a set lastly. In this case, there would not be a set, but let us, let me make it little simpler. So, this line, if you take the vertical line, then that is not intersecting.

Now, I start rotating around this point. Am I right. At some points it will become a tangent, but still it does not tangent, it starts intersecting this one. And, this one intersects this one. Now, then I will continue doing it. So, it will continue intersecting it, it will intersect something else. So, this happens at this point. right.

And, look at the first line when we continue rotation. The first line rate that intersects all the discs; so, that will be something like this. Am I right. So, this is the line that intersects all the discs. And, there is a first such line and what will be the last such line that will happen? No. That is right. So, what is, what yes, you are… that is true and not only that is true for disc, this is true for such circle. Am I right. So, what you want to do is, we have a cone. Am I right. So, the next will be something this, that will be last line.

So, what you do is, you have these cones. As a sort of sets for each circle, you draw the two tangents from the point P i. And, these are the cones, which have the same apex. And, if you look at the intersection of these cones, this itself will be an error cone. So, this is, this set of lines.

Now, when you add another one, you get another cone. And, it is the intersection of these two cones. So, if that cone remains non empty, it means you can add this edge. And, at some point the cone will become empty, then there is no such line and you do not want to put the segment. So, that is all you have to do. When you maintain this cone as a starting point, whenever you add the cone and start marching the points from P i to the right onwards and you maintain as soon as the cone remains empty. Non-empty, you add that segment and when it stops, then you basically stop idea. So, you get these errors.

So, this why, you can spend only a constant time at each point. … a cone, so it takes n square time total. So, you can construct the graph. Now, question is, can you do better than n square? Now, this graph is not arbitrary because you have edges going from P i to P i plus 1, P i plus 2. There is a sub sequence graph. So, this graph has lot of a structure. So, if you go from P i, if I will start from P i, P i plus 1, P i plus 2, every time you add a disc, you are spending constant time to add a disc. Am I right. So, yes, so, in the worst case, I will spend linear time. So, I in the linear time, I can add all the edges that, whose left end point is P i and then I repeat it for every point.

So, the total time will be n square. Now, but this graph has a lot of a structure, it is not arbitrary graph. So, it is no reason to say that you have to represent; you have to spend n square time to find a shortest path in this graph.

And, it is not known, whether you can do better than n square. That is the known, nice theoretical open problem. Now, if I change the rule of the game slightly, differently, which is used a lot that, suppose these are the function values that you sampled and if it is a x monotone cone, it is reasonably stuff, you assume a sample function. So, what is called as Chebyshev matrix? It is called… Matrix. The Chebyshev matrix is a place instead of thinking it in the disc; you think about it, I just care about the function value at that point. That is it.

So, I replace this disc by vertical segments. And then, again the condition becomes; this line should intersect all these vertical segment. That is called Chebyshev matrix. So, you are looking at the arrow under Chebyshev matrix. Do it and then it turn, you can do better. Then, you can find the shortest path in time n to the power four-third. So, this is sub-quadratic algorithm. But, when we look at the… matrix, it is these are no sub quadratic algorithm. And, that is, it will be nice to understand of the real right boundaries.

There been a number of irony papers on this problem. So, this one is the sort of, peoples sort of, came up with condition and it turns out to be false. So, in terms of time optimal t, we do not know what we can do. Any questions.

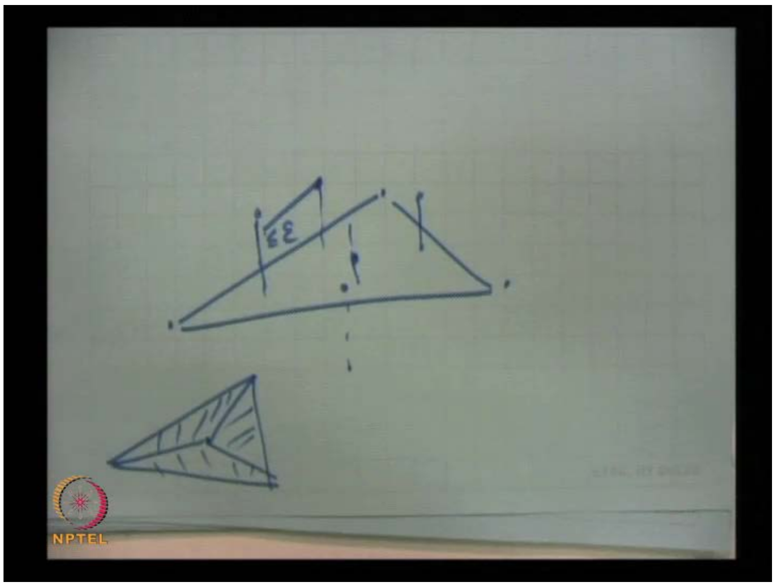
So, now let me sort of, I am really going out of time soon. So, let me talk about surface simplification. Yes. Unweighted graph. That is right. So, will be minimum number of edges. Now, before I talk about surface simplification, I should say the following. I should say the remark that I made about monotone verses non-monotone. Now, what happens is that, we start with the simple polygon. Am I right. But, what may happen is that, if you had something like this and if you replace this by simplify; suppose, arrow mark is large, simplify. Then what will happen is, you may get something like self-intersecting curve.

So, is no longer intersecting. You start about simple curve, simple polygon. But, once you simplified, you got self- intersecting curve and finding an optimal algorithm. So, what is not known is, whether you can compute a non-intersecting curve, which is a smallest size for given epsilon, whether it is a, this column is in P that is not known, as well as I did not know, it is not going to be in P.

Now, if I talk about surface simplification, so that both of this methods refinement and decimation methods, they can be extended to surface simplification. And, here is what you do is, refinement method is, let say that all the points, lines, had a triangle. So, you have, it is not a, if that is not the case of many of the circumstances of this problem; so, what I mean by is that, I take the points, so, now, if you have surface that points are sitting in three dimensional, you project them on the plane, let say all the projection lines had a triangle. Ok.

Now, in the case of Douglas-Peucker, what we are doing, the question is looking the first and last point. Now, there is no sequence. Am I right. But, there is sort of, you will be a kind of ordering and you ask the question, whether that the segment approximated all the points. Now, you will ask the following question. Take these three points in the triangle, whether this triangle approximates, all the points to all the points inside this triangle, whether the distance epsilon from the triangle, let say vertical height.

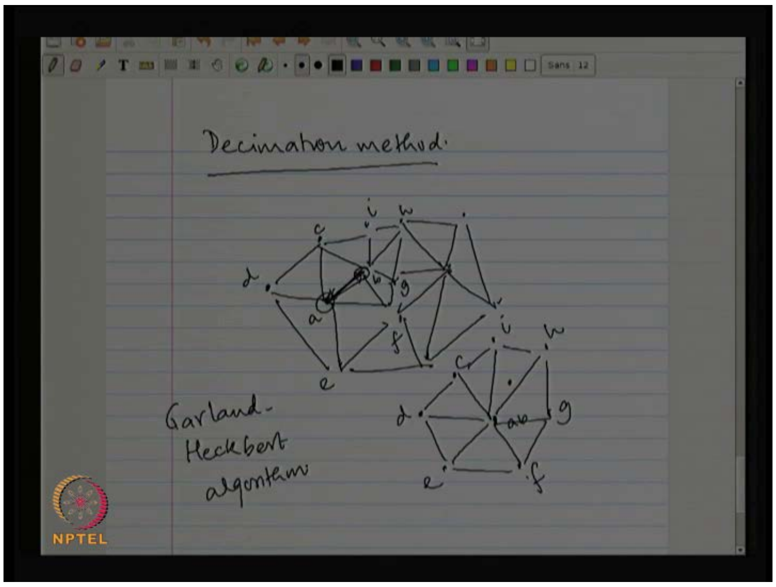
(Refer Slide Time: 1:02:00)



So you took the three points, you have the triangle. So, this triangle suspended in three space and you have all these points, whose projection lies inside this triangle. And, you ask, whether this distance is at most epsilon. If it is this case, then this triangle approximates all the points. You are fine. Otherwise, you do the same as Douglas-Peucker. Take the point that is farther from this triangle. And, let say and what you going to do is you going to replace that. This is a point; you are going to replace this triangle by three triangles. So, it is easier to think in terms of a projection that you have projection here and solve the problem recursively for each of these three triangles.

And then, you can take this point and map it back in the three.. And then, you start with one triangle replaced by three triangles and you ask the same question. And, you can keep on doing it, until you find your edge at triangle. So that, all the points have distance epsilon. So, that is the straight forward extension of Douglas-Peucker programming algorithm in three dimensional. I will not go into detail. This is not used a lot in practice in three dimensional. What is used in three dimensional is a, lot is a Decimation method.

(Refer Slide Time: 1:03:30)



So, what now happens is the following. And, you think about with the projection because it is easier to think about the… So, if you take this, what now, what you do is, you have a set of points, so, here is an algorithm and you start the… We have set of points in three dimensional. But, they are following a surface x monotone and y monotone surface, project them on the plane, compute a Delaunay triangulation and lift it back in three dimensional, you get a surface. And, instead of thinking in three dimensional, think in, just in Delaunay triangulation itself.

So, you have a set of points, which is the projection of the points from three dimensional and you construct the Delaunay triangulation. Let say, this is Delaunay triangulation. Now, it is not sort of, polygon change that, you just take two consecutive edges and do it. Now, but there is a notion of edge contraction here also; so, think about let us, I want to contract this edge. So, it has two end points, vertices. You fuse them into single end points. So, let us, it is a two, it is, think about a, b, c, d, e, f, g. So, these are the neighbors of a and b. once you feels it, in this sort of, becomes the, contracts to a single point. Then, what happens is that these two triangles, which were adjacent to the edge a, b, they get fused and then they become like two edges.

So, let say this is a fuse vertex a, b. Then, you get c, get g, then you have this d, e, f. Sorry, not g, f and the rest of the graph…luminously. Am I right. I hope this is right. I did not make a mistake. Yeah. Which seems right? Now, the many questions here, where should this fuse vertex set? And, the simplest thing is you make it as a set on a or you make it set on b. But, you should bring here…, bringing this vertex here or you think about bringing this vertex to here.

And now, what you have to say is, basically you have last 1 vertex. Fuse the two vertices become one vertex. And, you check that. So, this vertex which let say, you bring that point a, then the vertex b lies somewhere. And, you check what are that. Now, distance between b and the triangle if the error is at most of epsilon; you do this contraction. If it is more than an epsilon, then you do not do this contraction. That is the basic rule.

And, this is been used enormously. This is simple accurate in practice. And again, since the question is, which one you should do next; you basically maintain priority queue, what is error. And, pick up the vertex, which introduce the minimum error and contract that edge. And then, you keep on doing it. There is a very, there is a publicly available software for this problem and this was called… The history has been defined earlier. But, this is known as Garland Heckbert algorithm.

Those are the one, who did very nice…They are the one, who gave up the lot of nice DTS algorithm and details. And also, the implementation and the, you can get the software from their web pages. There are many different extensions of this algorithm. Now, let me go back to the question of the optimal algorithm, which I talked about two dimensional. That if I have given error epsilon and what is the minimum number of triangles? Now, the question you ask is, you can get that presentation.

And, that surprising the problem is not known. People suspect it is NP-hard, but the proof seems to be very harder to prove that it is NP-hard. And, there are lots of and its people have worked a lot. On to prove that, NP-hard and not only that. There is no approximation algorithm known, also.

So, suppose I tell you that if I give you error epsilon, there is a surface of with the k triangles that has error at most epsilon; Can you generate a surface with k log k triangles or even k square triangles, even a k cubed triangles, nothing is known. What is known is the following. If you do not insist that the vertices of the triangle should be the input points, then there are approximation algorithms.

But, somehow if you insist that the vertices of the triangle should be the subset of the input points, then nothing is known. No approximation algorithm is known. It is not known, whether the problem is NP-hard. And, that doing measurements and now you want to have a sparse representation to, basically of this surface measurement of Sciences. And then, the question… And also, it is not a centralized and you do not want to send all the data from sensors to the base station. You want to do local computation. So, the question is, is there distributed algorithm to do such surface of approximation. And, that is also an open problem. It is not known. So, let me stop here.