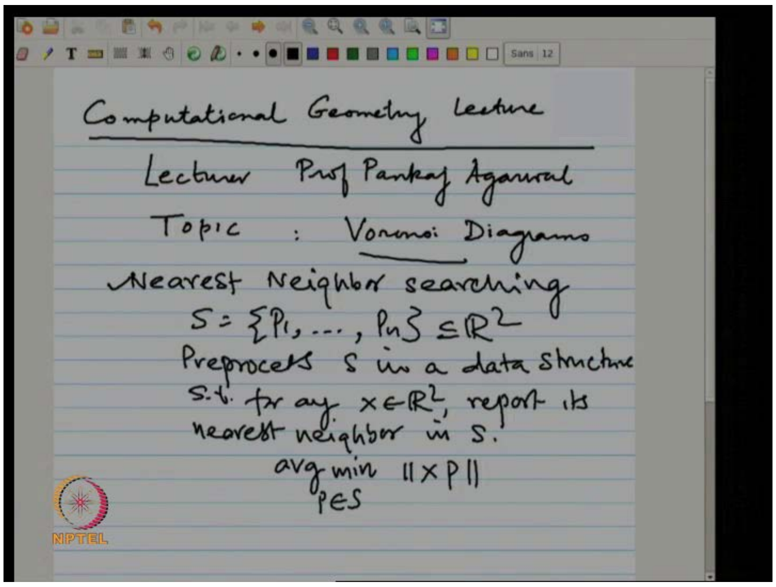
Computational Geometry
Prof. Pankaj Aggarwal
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

Module No. # 08
Voronoi Diagram and Delaunay Triangulation
Lecture No. # 01
Voronoi Diagrams Nearest Neighbour Searching

Today, we will start talking about Voronoi diagrams and Delaunay triangulation.

(Refer Slide Time: 00:33)



Computational Geometry Lecture

Lecturer Prof Pankaj Agarwal

Topic : Voronoi Diagrams

Nearest Neighbor searching

$S = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^2$

Preprocess S in a data structure s.t. for any $x \in \mathbb{R}^2$, report its nearest neighbor in S.

$\arg\min_{p \in S} \|xp\|$

And, the way I planned to do is in today’s lecture, I will tell you what Voronoi diagrams are; tell you some basic properties of Voronoi diagrams. In the next lecture, I plan to give you an algorithm; how you compute Voronoi diagram and Delaunay triangulation. And, on Friday, I plan to talk about applications and give you some sort of flavor why, what they are good for.

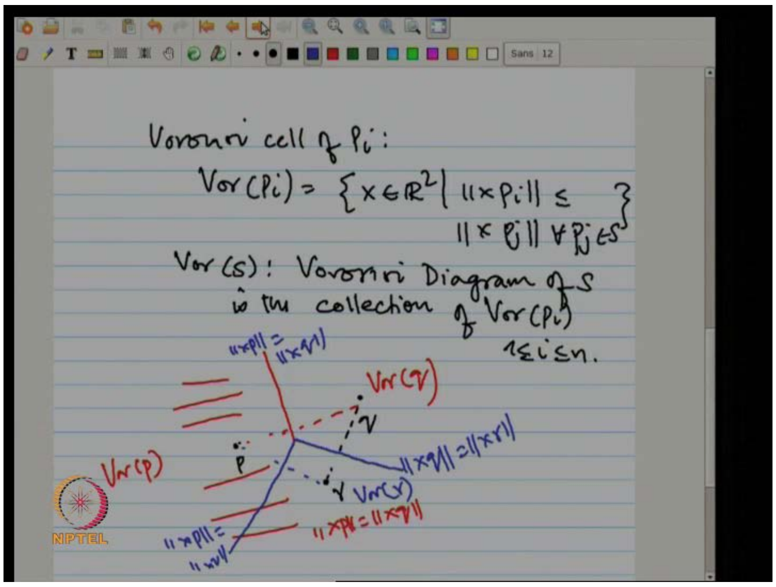
Let us begin. Also, let me start. If you think about what is called nearest neighbor searching… By the way, there are no stupid questions and feel free to ask me questions. And, if you do not ask, I will ask you. Nearest neighbor searching is defined as follows;

that you are given let us say set of points in two dimensions; but, they can be in any dimensions. And, actually in many applications, the points are in very high dimensions. But, let us stick to 2D for now. You have these points; you store them in a data structure. They are given to you once in advance. I will not talk about insertion, deletion for now. But, then you want to ask many queries. And, the query is that for any X belongs to R square, report its nearest neighbor in S. So, what you want to do is, if I have a set of objects, then I give you a query object; tell me which is nearest.

Now, nearest is a distance and one can… Here we are talking about Euclidean distance. So, what it… By mean is report the point that minimizes the distance. In general, it need not be Euclidean distance; you can define what are the distance functions you like; and, you do not want to do that. Now, when you do Google, when you do type a keyword, when it gives you answer, it is nothing but nearest neighbor searching, because you are trying to find some tax or something that is closest to what you typed, the keywords you typed. In databases that is what happens all the time. You want to look for something similar.

In molecular biology, you have some motif and you want to find the protein that has this motif; and, that is the closest way. So, it is all around. So, it is not surprising that the nearest neighbor searching has been studied in many different fields and for a long time. So, what is a relationship between Voronoi diagram and nearest neighbors (Refer Slide Time: 03:40).We started with the Voronoi diagrams and we are talking about nearest neighbor. Now, one way of answering the nearest neighbor is you do some pre-cooking. For each point in the input, each P i you store, what are all the points that are nearest to that point? And, you store that information for each P i. And then, when I get a query point, you figure out in which region it lies and then you answer it. So, I am going ahead, but, that is an idea.

(Refer Slide Time: 04:20)



Voronoi cell of P_i:

$Vor(P_i) = \{x \in \mathbb{R}^2 \mid \|x P_i\| \leq \|x P_j\| \; \forall P_j \in S\}$

$Vor(S)$: Voronoi Diagram of S is the collection of $Vor(P_i)$ $1 \leq i \leq n$.

With that in mind, let me define Voronoi cell of P i. We define as Voronoi P i is a set of all points such that x P i is less than or equal to x P j for all P j belonging to S; that means, that is a Voronoi cell of P i, is a set of all points in the plane for which P i is the nearest neighbor. Then, we can define Voronoi diagram of S is the collection of Voronoi P i. So, it is a… You take all the Voronoi cells; that together form the Voronoi diagram.

Let us start with very simple. Suppose you just have two points. Let us call them as p and q (Refer Slide Time: 05:43). Then, what (( )) due to the Voronoi diagram look like? Where you have bisectors? It should connect and draw this by perpendicular bisector. So, this is an equation. This line basically has a property that is equidistance from all the points on this line, equidistance from point So, this side of this line – this is a Voronoi cell of p; and, this side is a Voronoi cell of q. Now, let us add one more point. Let us add a new point r. Then, what happens?

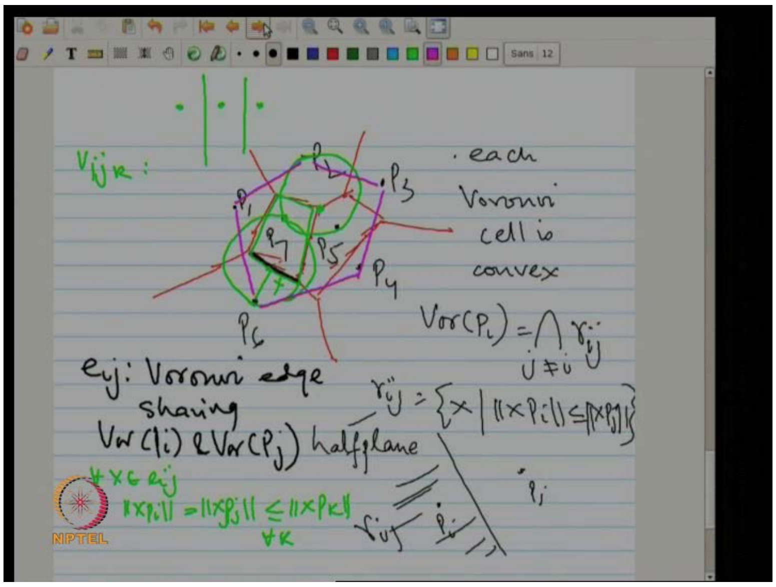
[Not audible] (Refer Slide Time: 06:37)

Yes. What will happen is that, as you said that, you take… It will change here. So, this becomes now the Voronoi cell of r. So, this is the equation x p is equal to x q; this is x q is equal to x r; and, this is x p is equal to x r. So far, so good. Any questions? It is not the centroid.

[Not audible]

Circumcenter, yes. I will come to that you are a step ahead of me.

(Refer Slide Time: 08:15)



V_{ijk}:
p_1
p_2
p_3
p_4
p_5
p_6
p_7
· each Voronoi cell is convex
$Vor(P_i) = \bigcap_{j \neq i} \gamma_{ij}$
e_{ij}: Voronoi edge sharing $Vor(P_i)$ & $Vor(P_j)$
$\gamma_{ij} = \{x \mid \|xP_i\| \leq \|xP_j\|\}$
half plane
$\forall x \in e_{ij}$
$\|xP_i\| = \|xP_j\| \leq \|xP_k\|$ $\forall k$
γ_{ij}
p_i
p_j

Let me try somewhat bigger example and hopefully I will complete it. Let us take… Can you see? I think this is a good approximation. And, let us label them these points: P 1, P 2, P 3, P 4, P 5, P 6, P 7. So, a few things to notice that each cell is convex first of all. Does someone know why each cell is a convex? Polygon?

[Not audible]

That is right. If you look at the intersection, if think about the Voronoi polygon or let us say Voronoi P i… If I go back to the previous (Refer Slide Time: 10:59) page and if you look at the equation; if I look at this equation, then it says that x P i should be utmost x P j for all j. So, what that means is that, this is intersection of half space. So, if I look at the… Let me… So, gamma i j is set x at x P i is less than or equal to x P j. And, this is a half plane as you said. So, if you have a P i, P j, it should be a perpendicular bisector. But, this is gamma j, because this is set of points; that dash region is set of points, where the point P i is closer than P j. So, this half plane is a convex object and the intersection of convex object is convex. So, it is a convex polygon.

Now, let us look at those edges. Let us look at an edge. By construction, if you know that, these edges are portions of perpendicular bisectors… So, let us write (Refer Slide Time: 12:45) e i j is a Voronoi edge sharing P i P j. So, let us say e i j is an edge, which shares the boundary between P i and P j – Voronoi cells of P i and Voronoi cells of P j. Now, we know that this edge is a portion of the perpendicular bisector of P i and P j. So, if you take any point, what we know is that this point is equidistant from P i and P j. In this case, P 6 and P 7. So, if you take a point x on this edge and draw a circle with the distance x P 6, it will also pass through P 7 as well, because both are equidistance. But, we also know that since it is in the Voronoi cell of these two, it will not contain any other point. So, I will not write this on page. But, basically, if you take a point on e i j, draw a disc of radius x P i or x P j, which are same, then… So, for all x on e i j, x P i is equal to x P j, which is less than x P j for all k. That is (( ))

Now, let us look at a vertex. This is equidistant from three points. So, let us denote it, call it V i j k (Refer Slide Time: 14:54) is the Voronoi vertex at the intersection of three Voronoi cells. As someone had said that, this is the circumcenter of these three points, because all these three points: P i, P j, P k are at the same distance. So, if I draw… pass through three points. Now, you notice that, in this Voronoi diagram that I drew, all vertices have degree 3. Why is that? Why they are at the vertices of higher degree; or, let me ask a question – when will you have a vertex in this diagram, whose degree is more than 3.

[Not audible] (Refer Slide Time: 15:48)

(( )) a point is described by three lines.

Two lines.

Two lines actually.

Suppose you have a square and (( )) Then, if you consider these edges as these points (( )) did you have…

Four touches. Then, what does it mean by that four? The four points are cocircular. So, what he was saying was something closer to what he wanted to say. Do you want to revise the statement that you made? You are on the right direction.

Actually, we cannot have more than three lines (( )) Simplex on a plane can be described by three lines.

But, it is not simplex; it is a circle, because for every three points, there is a circumcircle. But, in general, if you take four points, they are not cocircular; they have to be in very special position. Randomly, if I draw three points, then you can pass a circle through it; if you randomly draw four points in a plane, then you will not a circle pass through it. Circle is defined by three parameters. Three points define a unique circle.

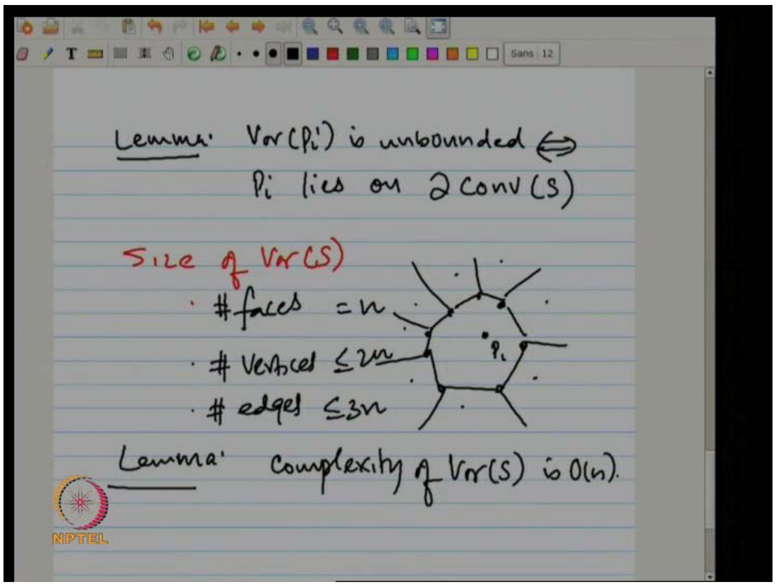
Now, there can be degenerate configurations as he mentioned like, if you take the vertices of a square, then it can be sort of from the center, all four vertices circular. So, what you can say about the Voronoi diagram, if no four points are cocircular, then degree of every vertex will be 3. Now, if you have three collinear points; if I have three points that are collinear; then, if you draw the bisector of these three points, they will parallelize this; they will not meet. But, if there are no three points cocircular, then all the bisectors will meet each other. So, in practice to simplify, to understand for simplicity, we sometimes assume there no three points are collinear, no four points are cocircular. But, you have to keep this thing in mind. But, sometimes this may not be the case; and, you may have large (( )) vertex.

Another property of Voronoi diagram that you notice is that some cells… That is a problem that you cannot see when I point something. So, if I look at this (Refer Slide Time: 18:34) cell, this is bounded polygon. But, if you look at the Voronoi cell of P 6, that is unbounded. So, some cells are bounded, some cells are unbounded. And, here is… It is not a coincidence. If I take the unbounded cells, I connect them. What do you get? That is something that you seen in this class. Take the point of all the…

[Not audible]

Yes, you get the convex cell (Refer Slide Time: 19:11).

(Refer Slide Time: 19:23)



Lemma: $Vor(P_i)$ is unbounded $\Leftrightarrow$ P_i lies on ∂ conv(S)

Size of Vor(S)

- #faces = n
- #vertices $\leq 2n$
- #edges $\leq 3n$

Lemma: Complexity of Vor(S) is O(n).

Conv I write used to convex cell and this is a boundary sign – partial sign. That is a boundary on the convex cell. So, the point as on the boundary on the convex cell, the Voronoi cell be unbounded; otherwise, it will be bounded.

Now, the…

[Not audible]

Yes, this requires the proof, which I am not going to give. It is not very hard, but it is not trivial. It is not obvious. I might give this as a homework problem since you asked this question.

Now, the million dollar question is, what is the size of the Voronoi diagram? Now, this each cell (Refer Slide Time: 20:54) is a convex polygon and they are n cells (Refer Slide Time: 21:04). Number of faces is precisely n. But, how many vertices and edges you will have? And, the issue is, because you know that each edge is a portion of a bisector; but, there are n square bisectors. So, that is… Also, you can have a face, which is pretty large, because if you take a point P i and put rest of the points surrounded let us say; then, if you look at the Voronoi cell of this point. Size of a face can be as large as n. So, that shows that… So, the number of vertices on this Voronoi face is n or n minus 1 if you want to be picky. So, you know there are n cells, which of them can have utmost n minus

1 vertices. So, total number of vertices is utmost n square – order of n square. But, that is a pretty loose bound.

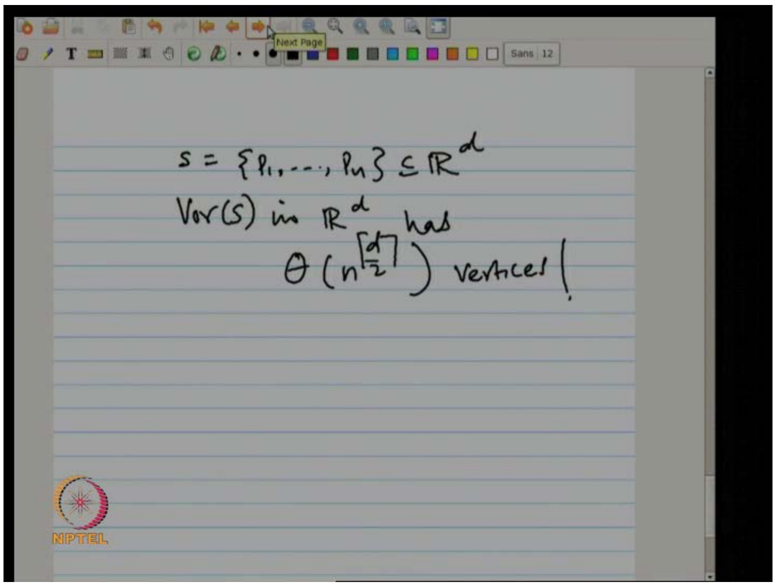
The question is, will it be n? Can it be n square? It is a planar graph. So, planar graph and the degree of every vertex is at least 3, because if you allow degree 2 vertices, there is a planer graph with n faces can have n square vertices, because you can add as many vertices you want. But, since the degree of each vertex is, that is, 3; and, it is a planar graph; then, number of vertices… So, everyone knows Euler's formula and planar graph; or, there is someone… or I should go through it? So, number of vertices will be utmost 2n and number of edges will be utmost 3n. So, main…

Now, if I go back to the nearest neighbor searching question, we are almost there in 2D (Refer Slide Time: 23:55). Remember, the question was that when I give you a query point, you want to know what its nearest neighbor is. Now, suppose you have Voronoi diagram (Refer Slide Time: 24:07) you computed, which I did not talk about how you are going to compute the Voronoi diagram. But, suppose you could compute the Voronoi diagram. And then, what you need to do is when I give you a query point, you want to know in which few faces it lies. And, if I remember, you have already seen point location; how to answer point location queries. So, if I can compute the Voronoi diagram, we know, the Voronoi diagram – it has a linear size it has only n vertices, n edges, out of n faces. You can preprocess this Voronoi diagram into a point location data structure and you answer the point location queries – tell me which faces the point contains. And, you know you can do that in log n time. So, that is basically in 2D. Using Voronoi diagram, you can answer nearest neighbor queries fast.

Now, if I go back to the (Refer Slide Time: 25:01) definition of Voronoi diagram, there is nothing magic about 2, because I could say this in high dimensions as well, because definition extends let us say in three dimensions; you do not have points in two dimensions, but you have points in three dimensions. And then, you can define similarly Voronoi cell. And, the whole thing sort of goes through; basically, define a Voronoi cell in this manner; and, you can talk about Voronoi diagram.

Now, almost if I have time, I will talk a little bit about Voronoi diagram in high dimensions. But, in 3D, this is not so a case. Number of cells is still n, because you will have now instead of a convex polygon, you will have convex polytopes. So, there are n convex polytopes. And, you can still argue that each convex polytope will have utmost order of n vertices. But, now, what can happen is, you no longer have a planar graph; you have some subdivision in 3D. And, as a result… And, there is a lower bound construction that almost each polytope may have linear number vertices. So, a Voronoi diagram of n points in 3D can have quadratic complexity; that means a sort of just mention it for now. And then, if I have time, I will talk little more about it.

(Refer Slide Time: 26:46)



$S = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d$

$Vor(S)$ in $\mathbb{R}^d$ has

$\Theta(n^{\lceil \frac{d}{2} \rceil})$ vertices!

Now, how many of you know anything about convex polytopes? Does anyone know about convex polytopes in high dimension? OK you and? OK. So, what? In 2D, when you have a polygon, what do you have? If you have a polygon, you have a polygon; then, you have edges and then vertices. Now, when you are in 3D, if you have convex (( )) think about this room as a convex polytope; you have all these looks and corners, which make you nonconvex. But, if you had a cube, then what happens is, you have a three-dimensional cube, then you have two-dimensional faces, edges – one-dimensional edges and then vertices. Now, when we go to higher dimensions indeed and four dimensions, what happens is that you have a four-dimensional convex polytope, three-dimensional faces, two-dimensional faces, one dimension and zero dimension. So, in d dimensions, you have faces from vertices of zero dimensions, edge is of one dimension, face is two faces. And, they go have high dimensional faces (( ))

Now, what is known is that the Voronoi diagram (Refer Slide Time: 28:11) has theta of n to the power d over 2 vertices. So, the size of Voronoi diagram is exponential in d. And, this bound as you saw is tight; it is always utmost n to power n d over 2; and, the points that for which that will be the case. So, that is a reason that Voronoi diagrams are not used for answering for nearest neighbor queries in high dimensions, for example, when you deal with the information retrieval. How many of you know how Google works? How does Google answer queries search? That you should know; come on; that is much more important than the geometry. If you do not know geometry, your life will not change, but Google you use every day. So, you should read the Google page rank algorithm if you do not know. It is how the Google works, how does search works. But, anyway I do not want… I think the whole thing is not relevant. But, what happens is that you have document let us say; you have keywords and you define that is what is called a feature vector in the very high dimension space.

[Not audible] (Refer Slide Time: 29:41)

That is right, yes. So, it becomes an index in… So, a document is mapped to a point in very dimensions in let us say in 1000 of dimensions – 1000. So, you have let us say 10,000 such documents or million documents. If you are going to compute the Voronoi diagram, it is just not feasible, because you have 1000 over 2 like million to the power 500. That is not going to be feasible. So, that is why Voronoi diagrams are very useful in low dimensions in two or three dimensions. But, when you go to high dimensions, when you want to answer nearest neighbor searching, this is not a viable approach; and, other methods are used.

Let me now go back to 2D (Refer Slide Time: 30:32). Let me erase some of the stuff from here.

[Not audible]

Go ahead.

What time the (( ))

That is a very good question and there are still papers are being published every year. And, I do not think I will talk about in this course. We are not going to talk about k-d trees or we are going to talk; we will talk (( )). So, there are some data of k-d trees and… But, certainly, when I am going to teach this course on geometry, then I will talk about this – some higher dimension. So, did they have some… So, that is up to 50 dimensions. If you want to answer nearest neighbor searching up to – let us say up to 3 or 4, you can still use Voronoi diagrams up to 4, because the thing is, the bound that I have mentioned n to power d over 2 – that is a worst case bound.

If you take the Voronoi diagram let us say, choose n points at random uniformly distributed in d dimensional space, then the complexity is not n 2 power d over 2; it is some quite small. So, even at a worst case, it can be quite large. But, if you have random points, the Voronoi diagram will… complexity will be small. So, in practice, up to 4 to 6 dimensions you can still go to Voronoi diagram. But, when you want to go to 8 to 10 dimensions, then Voronoi diagram is not feasible; then, you switch to k-d tree. And, that works until 50-100 dimensions. Then, you go beyond 100 dimensions, even k-d tree is not viable; you have to use other approaches. And, what happens is, in most of those applications, where you want to search nearest neighbor queries in 1000 dimension, you do not care about the exact nearest neighbor; you are quite happy with some approximate nearest neighbor. And then, you work with approximate nearest neighbor data structures; you do dimensional reduction techniques for example; that is used quite a bit. You do some kind of clustering and you to bring the dimension down and use some kind of clustering and partitioning approaches.

And, there is a plethora of papers. If you are interested, I can give you some survey papers. There is a survey paper by Piotr Indyk. There is also a survey paper by Ken Clarkson – a nearest neighbor in metric space, because in many applications, you have not even worked with Euclidean metrics; you have some in the arbitrary metric space; and then, there are the techniques that you use. So, that is a very nice survey by Ken Clarkson called nearest neighbor in metric space searching in metric space. This you can download from his web page and you can read many techniques there.

[Not audible] (Refer Slide Time: 33:11)

No, because you see I said you can find a nearest neighbor in order.

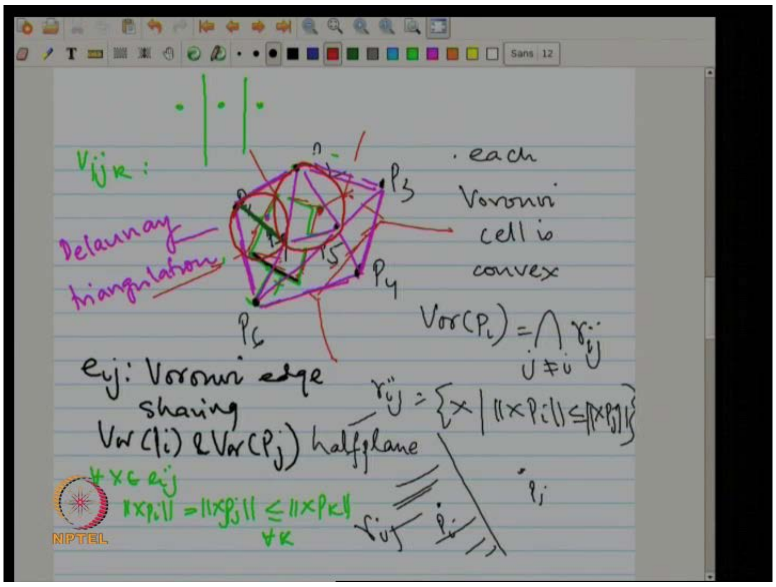
You can make (( ))

No, I did not say that.

The size is (( ))

It says size is only n, because no you can do it in n log n. And, you are a step ahead of me. When I talk about in next class, talk about Voronoi diagram, then I will show you n log n algorithm and I will say that n log n is a lower bound; and, I will say because it is as hard as conflicting convex cell; and, that is, I will say it is a lower bound, is n log n and 2d.

(Refer Slide Time: 34:01)



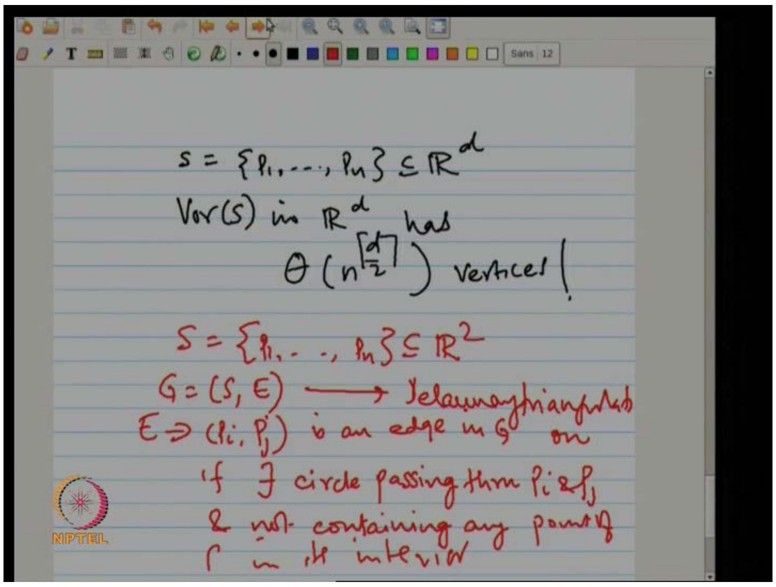
V_{ijk} :
Delaunay triangulation
p_3
p_4
p_5
p_6
· each Voronoi cell is convex
$Vor(P_i) = \bigcap_{j \neq i} r_{ij}$
e_{ij}: Voronoi edge sharing $Vor(P_i)$ & $Vor(P_j)$
$r_{ij} = \{x \mid \|xP_i\| \leq \|xP_j\|\}$
halfplane
$\forall x \in e_{ij}$
$\|xP_i\| = \|xP_j\| \leq \|xP_k\|$ $\forall k$
r_{ij}
p_i
p_j

As many of you observed that, Voronoi diagram is a planar graph. So, for every planar graph, there is also a dual graph; that is, you connect two points by an edge if they share a Voronoi edge. I already did for unbounded cells, because that is why you got the convex cell. But, that is completed. So, these two points P 1 and P 7 share an edge; I connect them by… Similarly, Voronoi cells of P 6 and P 7 share an edge. So, I connect them by an edge. So, I do this. What did I get? I got a triangulation of the convex cell. And, it is not a surprise that it is a triangulation, because the easiest of thinking about is you say because the degree of every vertex is 3. And, you know that if in the planar graph, in the dual graph, a vertex corresponds to a face. And, the number of edges on the face in a dual graph is as same as the degree of a vertex. So, here the degree of every vertex is 3. So, in the dual graph, each face is a triangle. And, this triangulation you got is called Delaunay triangulation.

Now, let me sort of… It is a beautiful triangulation. And, with many nice properties, it is somehow it is kind of magic. Let me start with by mentioning a few properties of Delaunay triangulation. What you notice, if you recall, if what I wrote is Voronoi edge property that I have written (Refer Slide Time: 36:42) that if you take any (( )) if you recall that if I take any point on a Voronoi edge and I draw its circle to its site, the circle passes those two points and does not contain any other points in its interior. And, the Delaunay edge is a dual of a Voronoi edge; that for every Voronoi edge, there is a Delaunay edge. For example, if you look at the Delaunay edge: P 1, P 7, what you know is there is a disc; there is a circle that passes through P 1 and P 7; and, that does not contain any other point.

Similarly, if you look at the triangle – if you look at the circumcircle of this triangle, this does not contain any other point in its interior, because this was center versus Voronoi vertex of this circumcircle (Refer Slide Time: 37:55). So, what it sort of says is that, when you look at the Delaunay triangulation, for every edge, there is a disc that passes through its end points and it does not contain any points in the interior. And, for every triangle in the Delaunay triangulation, if you look at the circumcircle, that does not contain any point in its interior. So, that is a property. But, the nice thing is which is not at all obvious is, its converse is also true; it is if and only if condition. By that what I mean is the following.

(Refer Slide Time: 38:36)



$S = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^d$

$Vor(S)$ in $\mathbb{R}^d$ has

$\Theta(n^{\lceil \frac{d}{2} \rceil})$ vertices!

$S = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^2$

$G = (S, E) \longrightarrow$ Delaunay triangulation on

$E \ni (p_i, p_j)$ is an edge in G

if $\exists$ circle passing thru p_i & p_j

& not containing any point of P in its interior

Let us go back that, you have a set of points in two dimensions. What I will do is, you construct a graph on the point set. So, P i, P j is an edge in G. If there exists a circle… So, you have certain points. For every pair of points, if you can draw a circle that passes through these two points, it means its center should lie on the perpendicular bisector of these two points and that does not contain any point in its interior. If you can do that, draw an edge. And, what you get is a Delaunay triangulation. So, this is an independent definition of Delaunay triangulation; I did not introduce Voronoi diagram anywhere. I just gave you a rule that which edges you should put; that is, we compute… So, this is a characterization of Delaunay triangulation.

Now, what you can do is, it is up to you; if I give you set of points, you compute its Voronoi diagram; you compute its dual graph; you get Delaunay triangulation; or, you can compute Delaunay triangulation first using this definition; and then, you compute its dual graph; you get the Voronoi diagram. And, it turns out that in practice – I will talk about in the next lecture – it is easy to compute a Delaunay triangulation first and then you do… And, Delaunay triangulation – they are used in their own right in many applications. So, for example, here is one application that (( )) few of you are seen. How many of you have seen a 3D printer or how many of you know a 3D printer?

[Not audible]

Only on you tube. So, there are sort of technology exist that I can take a scan of you, but through a 3D range finder. It will put sample points. It will compute a triangulation of that point set. So, that will be a model of your body. Send it to a 3D printer; you will get your sculpture. It is not a photograph; you will get a 3D sculpture of yours. And, it is pretty cheap nowadays if you can…

[Not audible] (Refer Slide Time: 42:05)

No, (( )) It is a… on a 3D model you get. It is a sculpture; you get a sculpture. It is doing it and it is a… So, those printers now do not cost very much; they cost about 10,000 dollars – those printers. And, the material has also become cheap. And so, it is used a lot in CAD for prototyping when doing it. For example, triangulation that they use, a surface model they use – they are variant of Delaunay triangulation actually a lot. So…

Sir, you say you have this Voronoi diagram and then you do the triangulation. You get a particular Voronoi diagram (( )) in your point edges; in vertices of your diagrams are centre of some circles. But, when you do the triangulation and then you convert (( )) you get a different kind of diagram; do not you?

No.

So, there is something I stepped under the rug and you caught me there.

[Not audible]

No. What you do is the following. You are right. There are some details that I did not say and I am glad that you caught me on that. So, what I do is the following. Suppose I have a triangulation; what I know, the Voronoi vertices are the circumcenters of this triangle. So, you take all these triangles; you connect the circumcenters. And, once you have circumcenters, then for each edge, you can add those two circumcenters by an edge and then you get the Voronoi diagram. Since I have about 5 minutes pointer, let me say a little more why Delaunay triangulation? It is better to compute Delaunay triangulation than compute the Voronoi diagram.

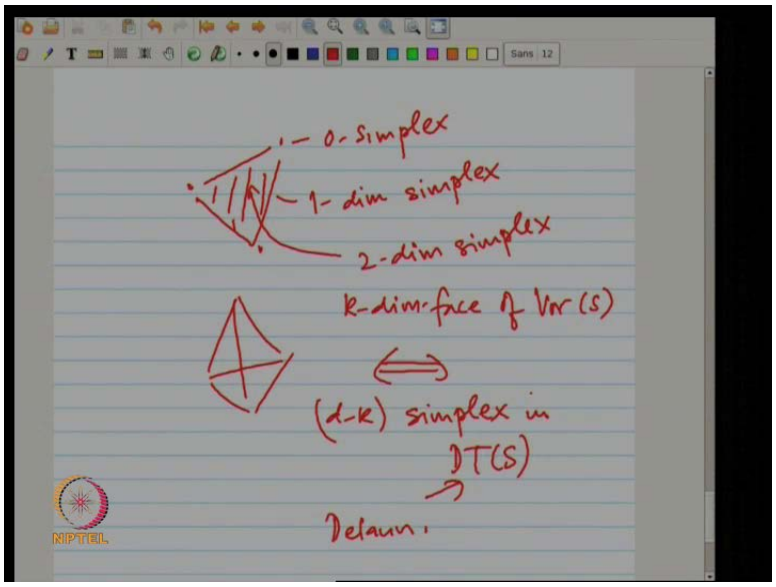
Now, unlike other algorithms that you do, (( )) algorithms that you have seen, in geometry, when you are working with you have points and points are coordinates; that means they are real numbers. So, when you do mathematics, it is easy to reason in terms of real numbers, real arithmetic. But, real arithmetic has infinite precisions; and, computers do not have infinite precisions, because we have only certain number of bits you do. As a result, if you just do floating point arithmetic, you compute the structures; or, computation you are doing, you are doing imprecisely, because there are some rounding of errors. And, this rounding of errors – when you are doing geometry, they are not just introducing some errors. Number of times what you do is, you do the following. For example, if I do nearest neighbor query, when I am trying to do is, I want to ask a question – in which face the point lies? And, for that, you will ask is, which side of the perpendicular bisector I lie?

Now, if the point is far away from the bisector, you do not need many bits to do this comparison. But, if the point lies very near the bisector, you need very high precision. And, if you did not have enough precision, you may get a wrong answer. And, number of times what happens, you can easily construct an example; it is not that only that your algorithm will give a wrong answer, but your algorithm will crash. And, most of the CAD softwares that are available, they suffer from this problem. You can easily crash them, because of the precision problem. So, what you want to do is, you want to do computation with as little precision as possible.

Now, when you do Voronoi diagram, you are working with the circumcenters. And, intuitively, what you think about it is that, when you have two points – let us say each point I give you b bits to present each point; you need about 2 b bits to represent the bisector; and, you need about 3 b bits to compute that to represent the circumcenter. Actually, it is more, but let me be sort of conservative. So, you need 3 b bits. Now, on the other hand, when you are doing Delaunay triangulation, you need fewer bits. So, the software is more robust, because you are working with basically the question. And, I will show you the algorithm how you compute Delaunay triangulation; I will give you a simple algorithm. Doing it, it requires fewer bits. And, that is why it is better to do Delaunay triangulation than compute the Voronoi diagram.

Let me just conclude by making a few more remarks. The notion of Delaunay triangulation extends (Refer Slide Time: 46:47) to high dimensions as well.

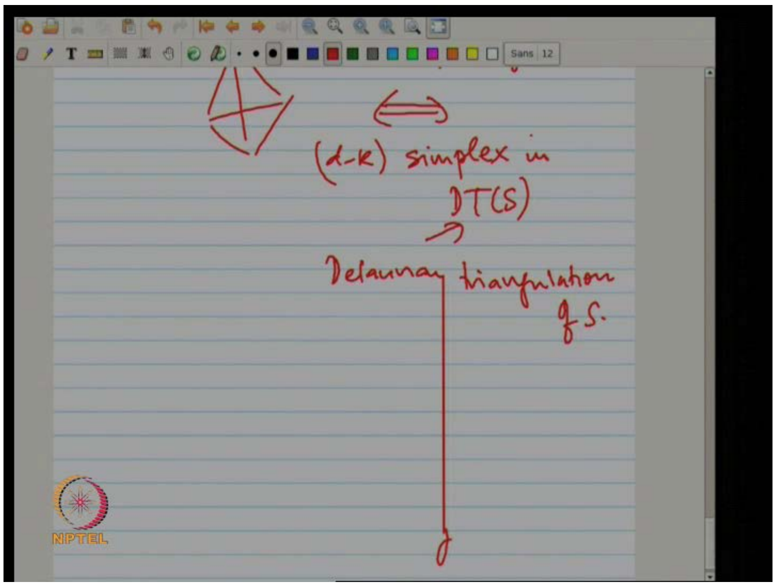
(Refer Slide Time: 47:02)



In 2D, when you do the Delaunay triangulation, it is the points, edges and triangles. So, points are… These are 0 simplex – 0-dimensional simplex; this is a 1-dimensional simplex; and, this is a 2-dimensional simplex. Our formal definition of a simplex is, it is a convex hull of d plus 1 – a finely independent points. So, it is a… In 0–dimensional, you have only one point and the convex cell have one point, is a set point in… If you take two points, take the convex cell; you get this line segment. And, if take the 3 points, convex cell is this triangle.

Now, what happens in 3D? If you take 4 points in 3D and if you take the convex cell, what do you get? Tetrahedra. So, you have a tetrahedra (Refer Slide Time: 48:01). Now, you can do in d dimensions also; you get higher dimension simplexes. So, in two dimensions, you had the triangulation. You take the convex cell; it decomposes into triangles. In d dimensions, you will have a d dimension convex cell and it will be decomposed in simplexes. And then, again it is a dual of Voronoi diagram.

(Refer Slide Time: 48:50)

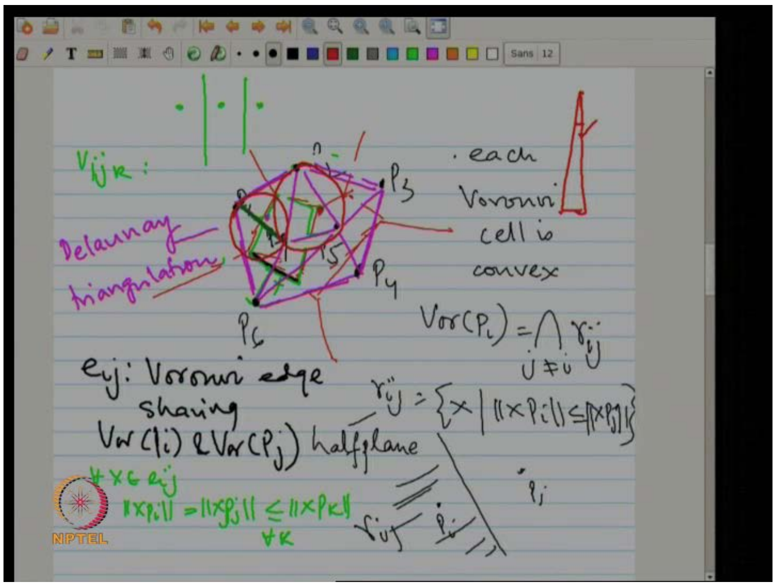


k-dimensional face maps to a d minus 1 simplex in DTS. So, DTS is a Delaunay triangulation; I will use the notation. So, there is a one-to-one mapping there also in Delaunay triangulation.

Now, one of the most widely used application of Delaunay triangulation is the following. If you have done any course on numerical analysis, when you want to do any simulation, what you normally do is you have some domain; you sample some points. And now, you want to interpolate in the rest of the places. So, you want to have some kind of interpolation simplex. And, Delaunay triangulation is used a lot. So, for example, the example that I will tell you, that is, scanner will scan your body; and then, you want to have 2-dimensional representation of a body, you do some triangulation. So, for example, lot of modeling people do, they use Delaunay triangulation.

For example, in people in biology – they study for example, heart; they want to understand (( )) cardiology, the model – the model heart; and, they also – how the heart pumps; and, they want to understand heart attacks. So, what they do is, they build sample points on heart and then they will create a 2-dimensional surface, which is (( )) more or sometimes you mean the 3D surface, because they want not only the surface, but they also want to measure interior. And, they use Delaunay triangulation. And then, also (( )) in this case, because when the heart pumps, it is deforming; the points sort of move. And, what you want to do is, you have the Delaunay triangulation of the points; moving points you want to maintain. So, these are called meshes. So, lot of work in mesh generation is used (( )) Delaunay triangulation is used.

(Refer Slide Time: 50:53)



V_{ijk} :
Delaunay triangulation
p_3
p_4
p_5
p_6
. each Voronoi cell is convex
$Vor(p_i) = \bigcap_{j \neq i} r_{ij}$
e_{ij}: Voronoi edge sharing $Vor(p_i)$ & $Vor(p_j)$
$r_{ij} = \{x \mid \|xp_i\| \leq \|xp_j\|\}$
halfplane
$\forall x \in e_{ij}$
$\|xp_i\| = \|xp_j\| \leq \|xp_k\|$
$\forall k$
p_j
p_i
r_{ij}
NPTEL

One nice thing about Delaunay triangulation is, if you look at the triangle… If you are doing interpolation, for example; you think about it; you like the triangles to be nice and nice shift. That if you look at these triangles (( )) you do not have… You do not like long and skinny triangles when you are doing interpolation, because these are not compact; they are not nice triangles. And, I will not go in much more technicality here, because you want to avoid these skinny triangles when you want to do interpolation on surface generation.

Nice thing about Delaunay triangulation is now the nice… The bad portion of this triangulation is that they have some very tiny angles. And, here (Refer Slide Time: 51:20) if you look at the… all the angles are large. So, Delaunay triangulation maximizes the minimum angle. The question I ask is, you have a set of points; you look consider all possible triangulations on this point set and determine one that has, in which the minimum angle of a triangle is as large as possible. And, that is Delaunay triangulation. Again, it is a property that is not obvious at all. But, I will not prove it. So, that is the reason that is quite useful in interpolation and finite element methods.

Let me stop here. And, in the next class, I will talk about how you compute Delaunay triangulation in Voronoi diagrams.