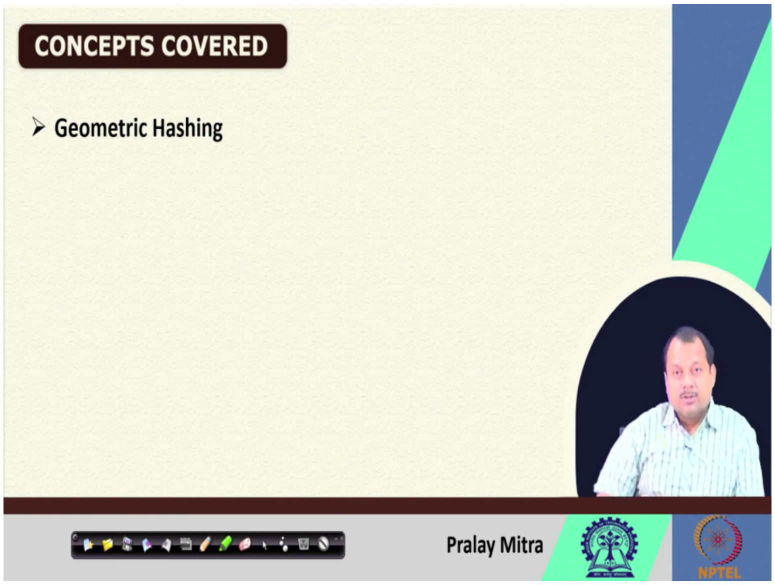
Algorithms for Protein Modeling and Engineering
Professor. Pralay Mitra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No. 11
Geometric Hashing

Welcome back. This week we are planning to discuss geometric hashing. And we will discuss the genetic algorithm in the context of surface matching. Similar to the last week, we wish to extend to protein complex modeling. That is why the module name is the same as the algorithmic technique for modeling. And in this lecture, we will start with the concept of geometric hashing.

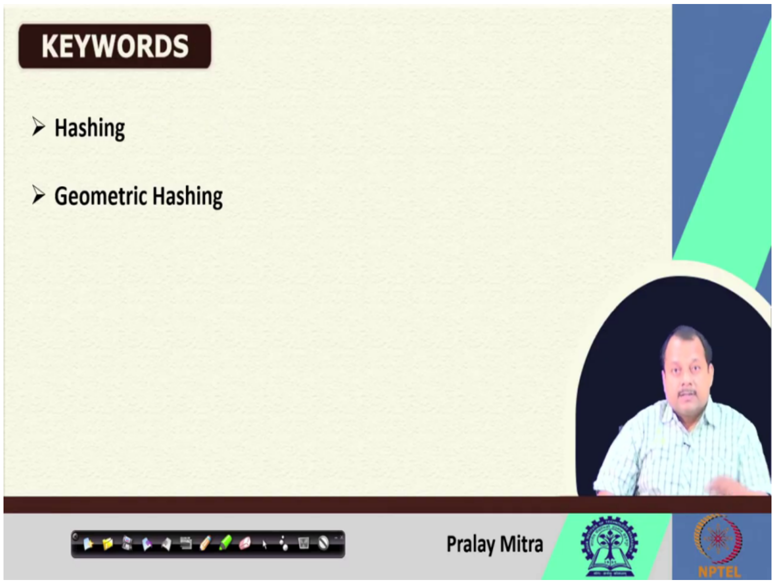
Regarding the hashing, in the last lecture, we give a brief introduction to the hashing. Specifically, the concept which will be relevant for our purpose, but hashing is very vast. We are not going into the details of that one. Only the portion which is required that we have discussed and we will be using that concept.

(Refer to Slide Time: 1:13)



CONCEPTS COVERED

- Geometric Hashing

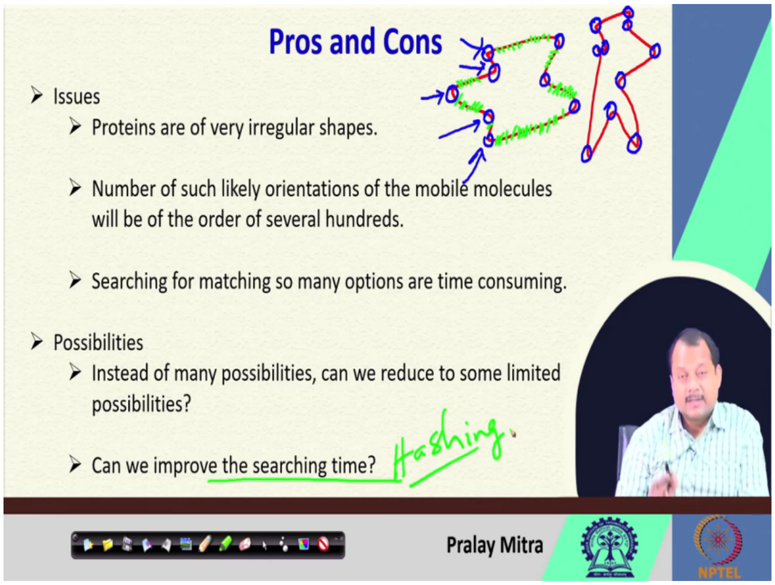


KEYWORDS

- Hashing
- Geometric Hashing

Let us start with the concept we are planning to cover is geometric hashing. We will cover in a couple of lectures, not in only one lecture. And as a keyword, I am using the hashing and the geometric hashing for the time being.

(Refer to Slide Time: 1:29)



Pros and Cons

- Issues
  - Proteins are of very irregular shapes.
  - Number of such likely orientations of the mobile molecules will be of the order of several hundreds.
  - Searching for matching so many options are time consuming.
- Possibilities
  - Instead of many possibilities, can we reduce to some limited possibilities?
  - Can we improve the searching time?

A few advantages and the disadvantages - one thing probably you noted as of now is protein is of a very irregular shape. If there is no regularity, then it is difficult also to model the surface. Hence, we mentioned in the last lecture that probably the simplest way to represent the surface will be to digitize the protein molecule and then identify the cell in the grid which is on the surface.

While doing that we also discussed that some sort of approximation will be considered and it is true that when you are digitizing, then definitely you are approximating. But, we have no other option, because protein does not have any regular shape.

On a secondary structure level (although we did not discuss it), there is some regularity/patterns in the form of hydrogen bonding. But if I look at the geometric level - it is very difficult. That is why some of the initial attempts to model the protein structure using some sort of shape descriptor or using some sort of topology analysis did not succeed much. Thus, it is one of the concerns.

Another concern is the number of likely orientations of the mobile molecule will be of the order of several hundred. That we also discussed in the last class. Corresponding to each protein molecule, there are 6 degrees of freedom 3 translations, and 3 rotation3. If you assume that the translation step is 0.5 angstrom and the angle of the rotation step is 1 degree, then taking together 3 different translations along the X-axis, the Y-axis, and the Z-axis and 3 different rotations about the X-axis, the Y-axis, and the Z-axis will be several 100,000.

Then the question is whether all of them are relevant or not? Percentages of true positive cases are the cases that will be biologically probable and will be very less in number. That is true. But if you do not generate the true positive case or biologically valid orientation, during your generation or stage. Where we are exploring all sorts of orientations, then it will not be possible for you to identify the correct one.

Nevertheless, when two protein molecules are given to you, whose orientation is not known to you, then you are clueless. So, you have to generate as many possibilities as possible. That is why it is good or advantageous from the fact that if I generate several hundred or thousands of orientations. At the moment, I go for that one. Then the computation time will also increase. So, what to do with that one?

Next is searching for matching. Many options are also time-consuming. So, if I give you one simple example say. I have generated say 340 thousand orientations or say if I assume that 9 million orientations I have generated. I am given a very large protein molecule and truly speaking I do not have any knowledge of the binding site or the probable orientation where they will go and bind if that is a situation and you think that 9 million orientations have been generated. Even if I go for the simplest way of calculating the surface matching that we have discussed in the last class, then also it will take about a second or so. If I assume without any loss of generality that it is not only the geometric surface matching. But some sort of physicochemical information is also attached where I am matching that one to check the geometric compatibility as well as the physicochemical compatibility of the interacting area or the interface which will take about a second. Then to test 9 million cases, the value will be 9 million seconds!!!

Even with a parallel programming language, computational time will be high. Also if the score value varies from 0 to some finite number, then for one score value multiple solutions will be there. If I rank them based upon the score then also the percentage of the true positive cases will be very less compared to the total cases.

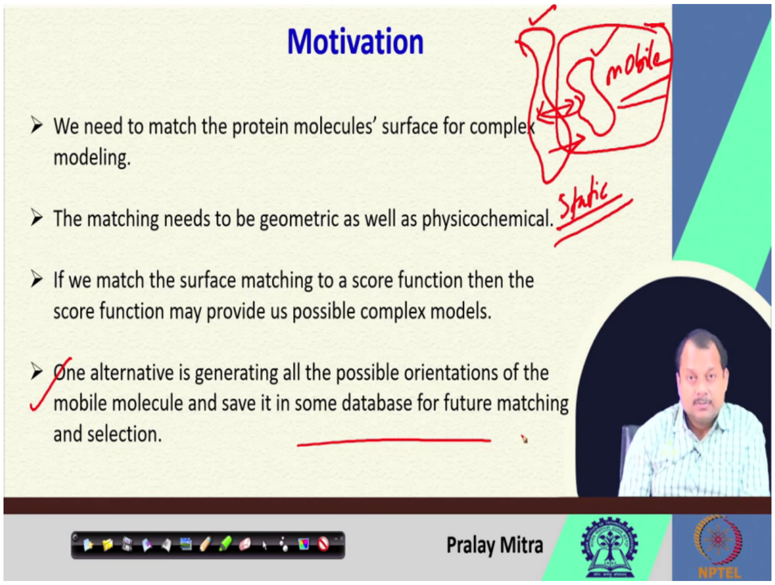
So, what is the trade-off? What is the balance? How should I proceed in such a way that during the generation phase we will not miss out on any true positive case, at the same time the number of generations of such orientation will also be limited. The good news is that instead of many possibilities, if I exploit some of the biological information and declare some of the points as the important points or the critical points which specifically describe the shape or structure of a molecule, then perhaps it will be an advantage. Let me give you one example, I am drawing one shape like this. This is a very regular shape another shape maybe like this. Now, instead of all the points inside this, I consider these are my important points. My logic is the point where the inclination changes from one direction to another direction. Then perhaps, I can declare some shapes.

Similarly, if I match then this will match, this will match, this will match, this will match, this will match. What is next? I can think of as if these points are not present. Now, you see that the number of residues or amino acids which are several hundred is now reduced to only a few points to be considered. Hence, the size of the problem reduces. But, I have to make sure that the selection of this point serves my purpose in capturing the geometric shape and the physicochemical information of the molecules so that during the generation phase I do not miss out on many true positive opportunities.

It is fine if I am missing out on a few true positive cases, but I should not miss out on a large number of true positive cases. Because while I am generating you should remember that hardly

1-2% or if you are fortunate enough, then at most 5% true positive cases will be there. Now, if I increase the percentage of the true positive cases then the chances will be high to get the hit. The last point is that can we improve the searching time? Is it possible to search quickly? Here we shall exploit the hashing technique.

(Refer to Slide Time: 11:20)



Motivation

- We need to match the protein molecules' surface for complex modeling.
- The matching needs to be geometric as well as physicochemical.
- If we match the surface matching to a score function then the score function may provide us possible complex models.
- One alternative is generating all the possible orientations of the mobile molecule and save it in some database for future matching and selection.

Pralay Mitra

The motivation is we need to match the protein molecule surface for complex modeling. The matching needs to be geometric as well as physicochemical. If we match the surface matching to a score function, then the score function may provide us, with possible complex models. One alternative is generating all the possible orientations of the mobile molecule and saving it in some database for future matching and selection which is a very good idea.

So, what it suggests is that if you are given one static and one mobile molecule you know how to translate, rotating both the molecules. I mean transforming both the molecules are same as keeping one as a static and another as a mobile or say this is my mobile which means this molecule will be translating and rotating and it will be fixed. Now, if it is the situation, then the last point suggests that for this mobile molecule, what if I generate all the possible translation and rotation and save those translation and rotation information in some database. How I am going to save that I shall discuss later, but for the time being, you can assume that there is a technique through which I can save that one in some database.

My next step will be to give the static molecule you look for all the orientations of the mobile molecule and perform one-to-one searching by matching their surface. Thus, you need a surface matching score function for which you can use the geometric score function that we discussed. Also, we use some electrostatic information or combine other physicochemical information with the score function. Here, I am sticking to that particular point that when I am matching the surface, then the geometry will come fast - no doubt. But some physicochemical information can also be integrated to get a combined score. Now, for each pair of orientations - one static and another transformed mobile molecule, I am getting one score value.

Similarly, you can see that, unlike the last time when we are generating and scoring the surface, here, we are generating and storing that model and then we are computing the score function or computing the surface matching score. This is kind of edge scoring or after the generation, I am scoring. Last time it was integrated scoring.

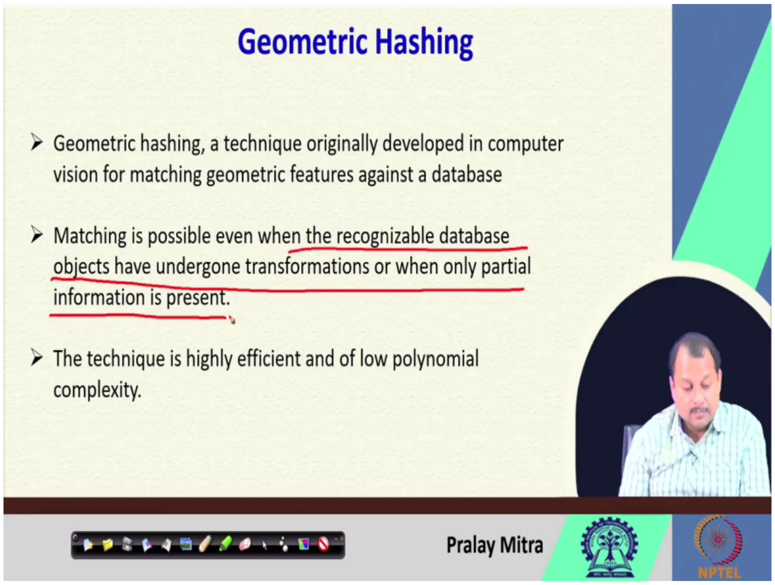
This suggests that if I can do that one, then perhaps that is a good idea. Well, no doubt that is a good idea. But this also incurs two more problems. Number 1: when I am storing and I admitted the fact that say several hundred or thousands if not millions of orientations will be generated. What is the efficient way of storage so the searching will also be easy? That again triggers the point that perhaps hashing will be a solution. Stick on that.

Next when I am generating several thousand or millions of the orientations and storing them in some database. How much storage space will be required? If I say instead of the complete orientation, you can store only the transformation matrix, then that is fine. Then you have to apply the transformation matrix and generate the orientation for scoring. This approach is fine as long as I am given a sufficient amount of storage space and I have developed some hashing technique that is fast enough for database searching.

Geometric hashing is a technique originally developed in computer vision for matching geometric features against a database for the same purpose. Matching is possible even when the recognizable database objects have undergone transformation or when only partial information is present. This point is very important for our purpose also. It says that if there is a transformation.

In the last class, I give you an example of one protein molecule. This is another protein molecule. I know that this is the orientation they will be in. Now if I take this molecule and give some orientation like this, then definitely this is not going to interact or this is not going to be the correct one. Now, if I take this one that is not correct. Now, if I rotate, if I give an inverse transformation of what I have done, then only I shall get the correct one.

(Refer Slide Time: 16:30)



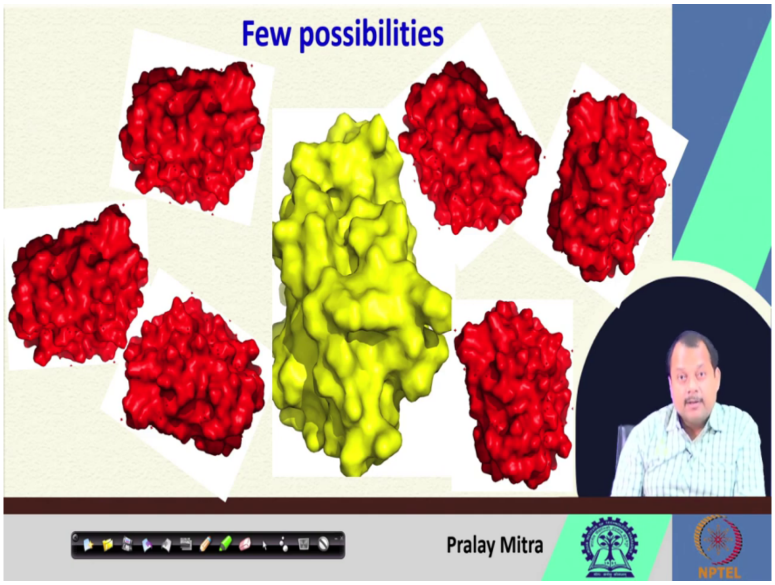
Geometric Hashing

- Geometric hashing, a technique originally developed in computer vision for matching geometric features against a database
- Matching is possible even when the recognizable database objects have undergone transformations or when only partial information is present.
- The technique is highly efficient and of low polynomial complexity.

Pralay Mitra

I should have one technique where the transformation that I have applied has no effect if it is biased by the transformation, then I will be in trouble. Now, the technique is highly efficient and of low polynomial complexity. Shortly we shall apply hashing technique for this purpose.

(Refer Slide Time: 18:09)



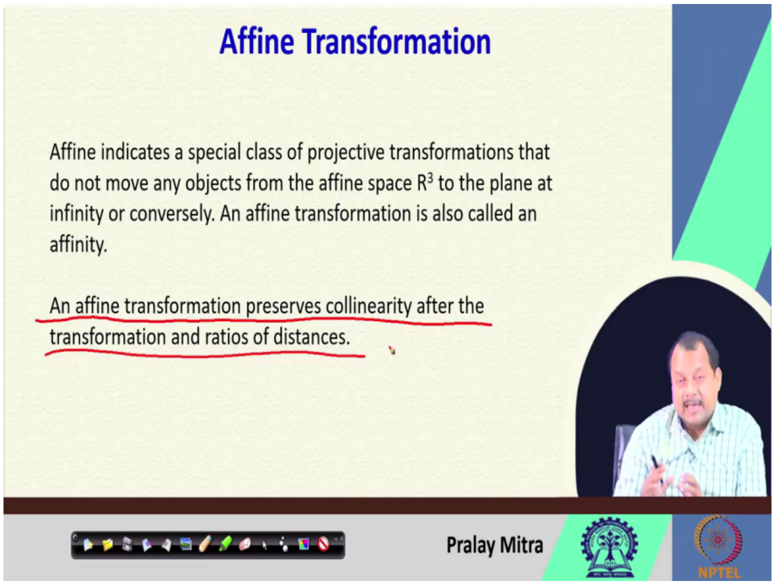
Few possibilities

Pralay Mitra

Let us assume this is my static molecule and the mobile molecule can have one orientation like this, can have another orientation like this, can have another orientation like this, can have another orientation like this, like this, like this.

You are familiar with this protein molecule from the beginning of the class. You can quickly identify that it is the same one with a little transformation and translation. It is fine, but when you are writing a program, which will take two protein molecules - yellow and red, and you will be asked which is the correct orientation or, which is the correct orientation for which the surface matching score is the largest? If that kind of question will be asked then, in the program, you have to implement that. That is why you have to come up with some idea that is kind of invariant to that transformation.

(Refer Slide Time: 19:35)



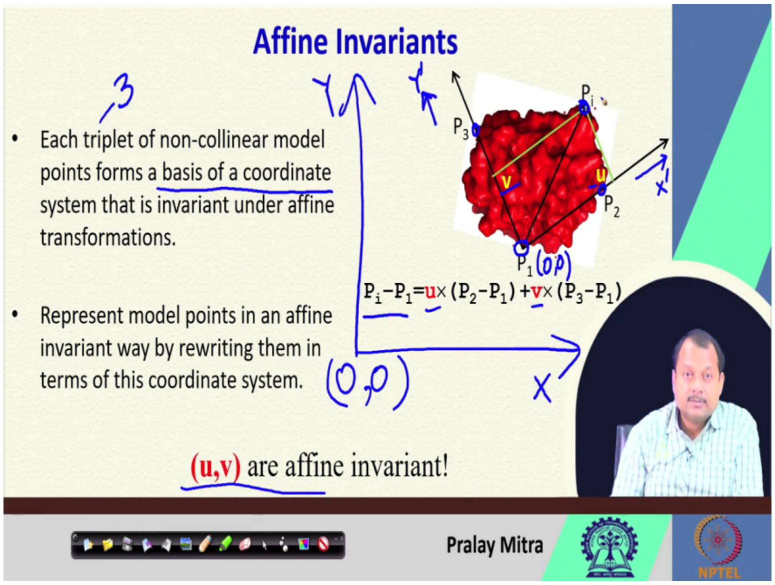
Affine Transformation

Affine indicates a special class of projective transformations that do not move any objects from the affine space R^3 to the plane at infinity or conversely. An affine transformation is also called an affinity.

An affine transformation preserves collinearity after the transformation and ratios of distances.

Regarding this invariance, we are defining the affine transformation. Some of you if you have done the course on computer vision or object recognition, then are familiar with this affine transformation. It says that affine indicates a special class of projective transformation that does not move any objects from the affine space R^3 to the plane at infinity or conversely. An affine transformation is also called an affinity. An affine transformation preserves co-linearity after the transformation and ratios of distances which is very important for our purpose. Because, for our purpose, we are not interested to scale the object, only translation and rotation are required.

(Refer Slide Time: 20:42)



Affine Invariants

- Each triplet of non-collinear model points forms a basis of a coordinate system that is invariant under affine transformations.
- Represent model points in an affine invariant way by rewriting them in terms of this coordinate system.

$P_i - P_1 = u \times (P_2 - P_1) + v \times (P_3 - P_1)$

(u,v) are affine invariant!

Pralay Mitra

Here is an example of affine invariants. I have taken one orientation of the mobile molecule and each triplet of non-collinear model points form a basis of a coordinate system. In this case, 3 triplets, triplet means 3. 3 non-collinear points viz., P1, P2, and P3. These three points are on the mobile molecule. Those three non-collinear triplets - you can see this P1, P2, and P3 form the basis of the coordinate system. For me one axis like this another like this. We are demonstrating in 2-D which I believe you can extend for the 3-D also. Represent model points in affine invariants way by rewriting them in terms of this coordinate system.

Now, you see this P1, P2, and P3 in some coordinate systems. There is coordinate information in the protein databank structure file with angstrom dimension and when those coordinates are presented in X, Y, and Z, then you understand the existence of one reference frame to what that molecule is given.

Without any loss of generality, I am assuming that this is my coordinate this is my assumption and this is X and this is Y. In this coordinate system P1, P2, P3, Pi, or all the atoms in this mobile molecule are given where P1 has one value for (x, y) P2(x, y), P3(x, y) it says that if I consider any triplet, which is not collinear then I can consider that as a basis for coordinate system and that coordinate system I am defining like this. This P1 is my origin. This is my origin now. This is X’ axis this is Y’. This X’ and Y’ is another coordinate system, where the origin is (0,0) which is my P1.

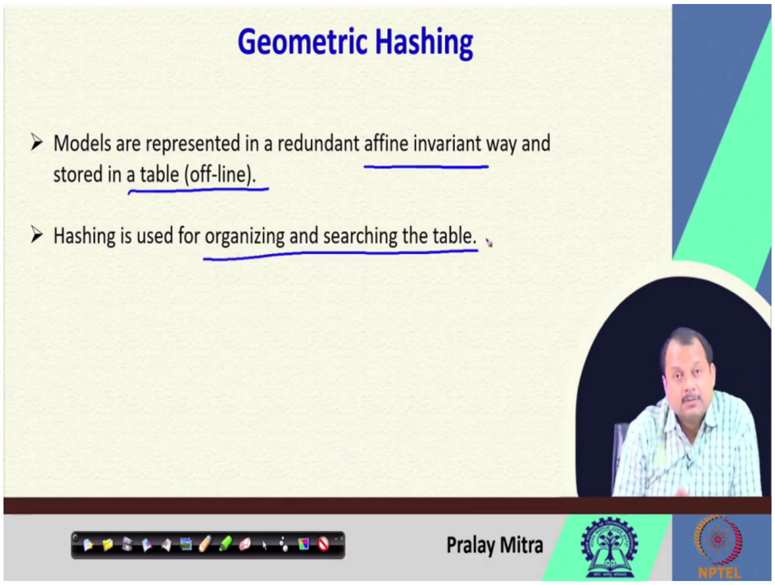
If I consider that one, then I can represent model points Pi that is here in affine invariant way. How?

Pi - P1 = u × (P2 - P1) + v × (P3 - P1)

From here I am taking the projection on the X-axis and the Y-axis and from that projection, I am getting u and v. Using u and v and by that projection, I am computing or representing the model point Pi in an invariant way. Now u, v is affine invariant.

I can represent Pi using this u, v, and these in this same like basis coordinate system I have selected - P1, P2, P3 triplet. All together P1, P2, P3, u, and v will give me the information regarding this Pi. Is this clear? Based on it we are going to develop our next step. Despite it being demonstrated in 2-D, I believe you can extend it for 3D by adding another dimension.

(Refer Slide Time: 25:12)

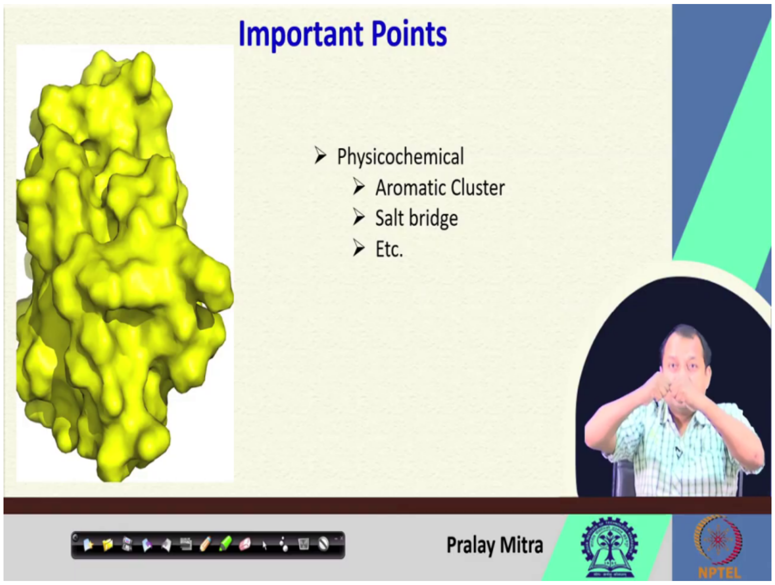


Geometric Hashing

- Models are represented in a redundant affine invariant way and stored in a table (off-line).
- Hashing is used for organizing and searching the table.

Now, we discuss geometric hashing. Models are represented in a redundant affine invariant way and stored in a table offline. Please note I mentioned that models are represented in a redundant affine invariant. What is affine invariant? You know by this time and stored in a table offline, this table we are going to tell is a hash table so that the searching will be fast. We noted that if have a very good hash function without many collisions, then searching can be done in constant time. Just calculate one mathematical expression. Hashing is used for organizing and searching the table. You remember that hash function.

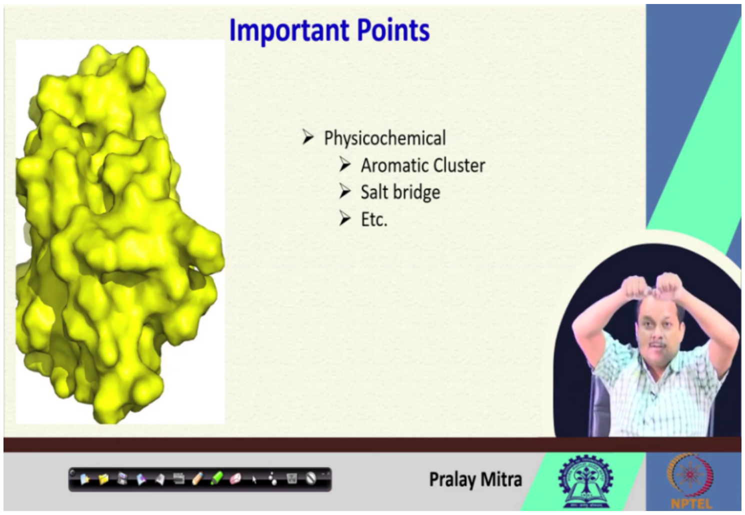
(Refer to Slide Time: 26:10)



Important Points

- Physicochemical
  - Aromatic Cluster
  - Salt bridge
  - Etc.

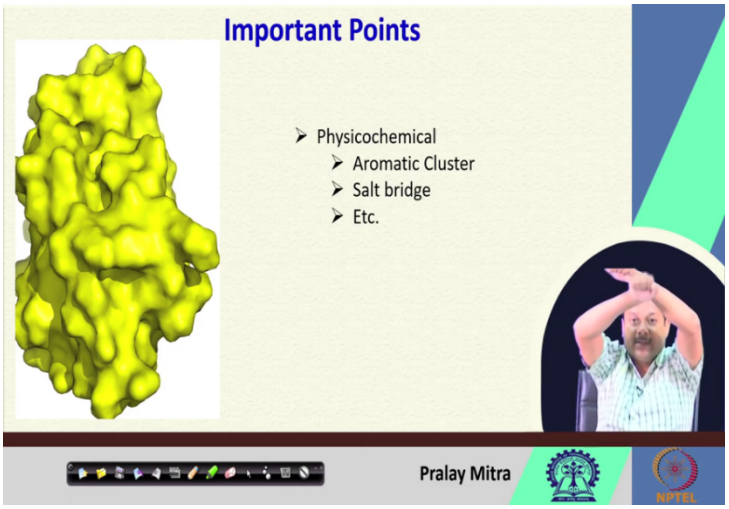
Pralay Mitra



Important Points

- Physicochemical
  - Aromatic Cluster
  - Salt bridge
  - Etc.

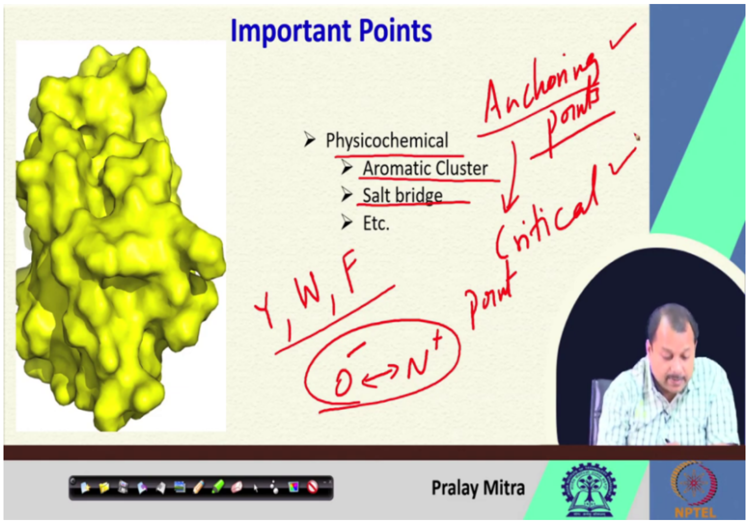
Pralay Mitra



Important Points

- Physicochemical
  - Aromatic Cluster
  - Salt bridge
  - Etc.

Pralay Mitra



Important Points

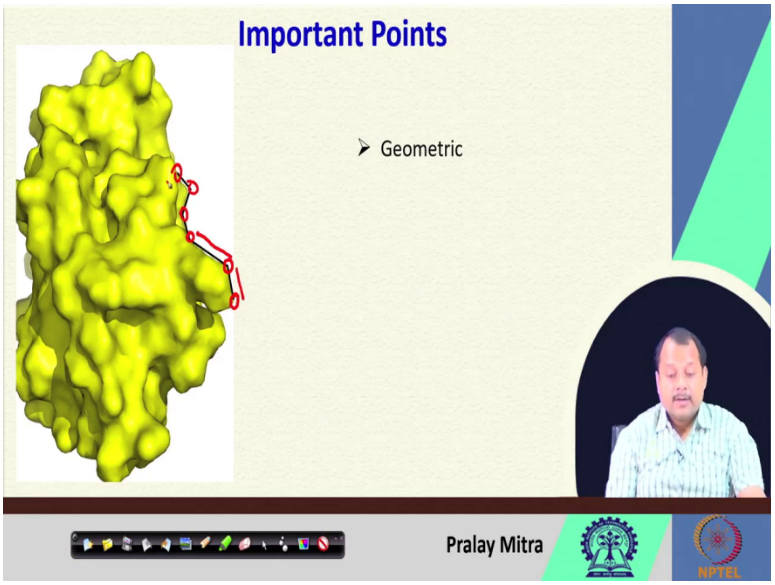
- Physicochemical
  - Aromatic Cluster
  - Salt bridge
  - Etc.

Now, the important points: let us consider one protein molecule. If this is one molecule, again you can see that it is a very irregular shape, there is no regularity. If no regularity is there, then it is very difficult to represent using some topology or using some shape descriptor. Nevertheless, the biologists suggest that when you are going to generate the orientation, then all possible orientations are not required to generate because they will not occur.

Theoretically, I said that if this is my protein molecule, this is my other protein molecule, then these are with different possible orientations. Theoretically, this is one orientation, this is another orientation, this is one orientation. But biologist says that there is some rule which is followed by the nature and using that may not be a correct biological orientation. Why? Because the amount of the contact is less. This amount of the contact I understand only when, I compute the surface matching score whereas, the, if I go by the fact that for this protein molecule, there are some anchoring points then, that anchoring point will tell me that they are the critical points. When you are generating the orientations, a few points will act as an anchor. An anchor means that at that point it will go and bind. If that is the situation then I can also call that a critical point. According to the biologists, these critical points are with some physicochemical properties like aromatic clusters, and salt bridges. Detail we shall introduce when we shall discuss. But what is this aromatic cluster? So, what are the aromatic residues? Those residues like tryptophan, tyrosine, and phenylalanine where there is an aromatic ring at the side chain are called aromatic residues. You mark all the aromatic residues that are on the surface. Then, there is a tendency that charged residue will interact means, acidic residue will interact with the basic residue. If that kind of electrostatic interaction happens, then you also mark all the charged residues that are on the surface.

First of all, from the given protein molecules, which are there, you are taking a subset of the amino acid molecules which are on the surface. From those surface molecules or precisely if, I focus only on the atoms, then for all the atoms which are on the surface, I am considering those atoms which are part of aromatic residue or charged residue then, I am considering only the side chain where there is oxygen or say nitrogen who can form the salt bridge. Following this, you will understand that the number of anchoring points or the critical points will be very less compared to the total number of atoms in the protein molecule. Now, this is about the physicochemical information. What is regarding geometry? Here you can see that some of the physicochemical points I have marked here this is some symmetric ones. So, do not take any correlation with the correct molecule. But something like this.

(Refer to Slide Time: 31:14)



Important Points

- Geometric

If I go by the geometry then I can look for the points where the inclination changes. For example, if I look carefully. I have drawn one line here. Now, from here I shall go this way next to another black line. So, here this one, this one that is why the point where the inclination changes at that particular point, I am selecting. This is one important point, this is another, this is another, and if I allow then this will go another. I hope you can able to follow, then this is another, then this is another, that way it is going this and this. These are that geometric features. Thank you.