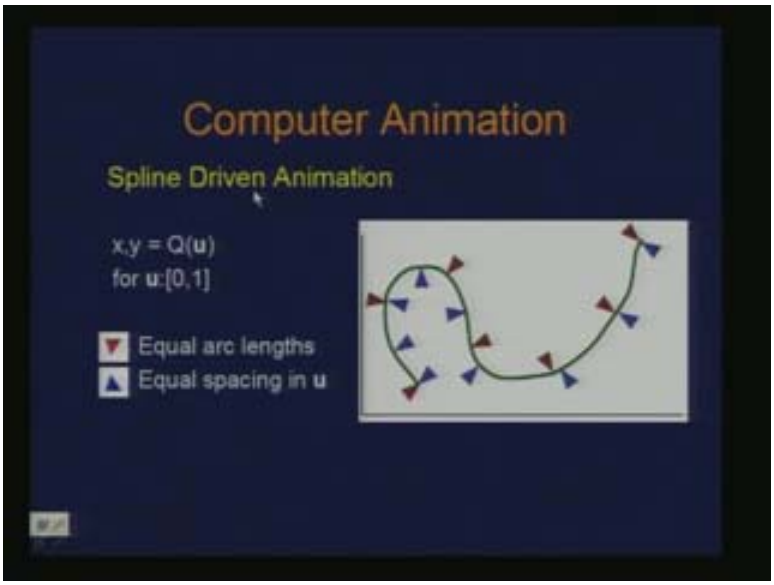
Introduction to Computer Graphics
Dr. Prem Kalra
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
Lecture - 34
Animation (Contd.)

We started on animation and we will continue talking about animation. What we basically started looking at was the usage of splines in the context of animation. So we are going to look at one of the applications where splines can be used for the purpose of animation.

(Refer Slide Time: 01:25)



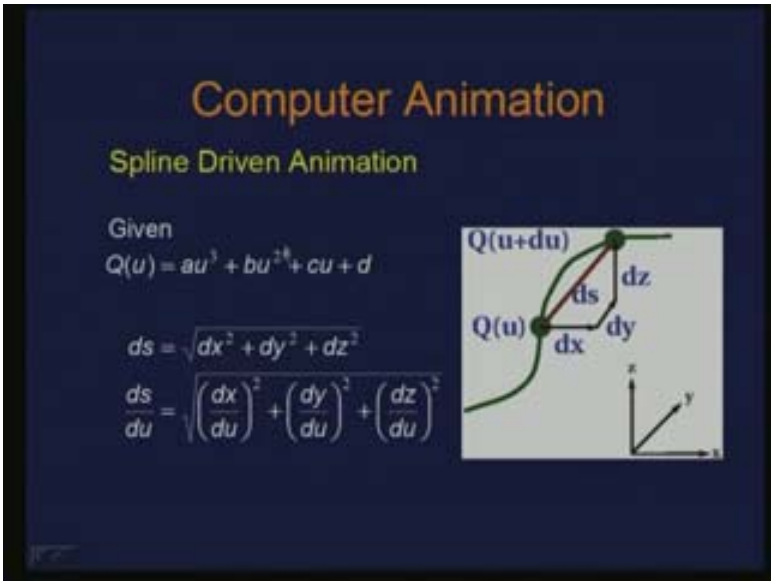
I call that as spline driven animation. For example, if you have a trajectory specified in space where the movement of the camera or the object is taking place and it is desired that the motion along the path generated by this spline has got uniform speed, and that is what we want to obtain. Now the issue here is that if I have a two dimensional case where x and y are represented through this Q(u) so it is parametric curve in u where u is between 0 and 1. Now if we want to generate the movement along this curve with the uniform speed what we really want is that, for a given time interval the distance which is traversed along this curve is the same.

If I fixed interval t is equal to 1 or 2 or whatever time then the same amount of distance i will be traversing along the curve. So basically it is an equidistance traversal along the curve. Now one may start wondering, the fact is that it is a parametric curve in (u) then will the uniform or equidistant interval in (u) would provide me an equidistance traversal along the path. The answer is no because it is actually a polynomial in (u).

If we are talking about a cubic case then it is something like a(u) cube plus b(u) square plus c(u) plus d. So, clearly that is not the case. It is not a linear segment where we are trying to traverse but it is actually a curvilinear path. So one thing which is desired from here is that I want to have equal arc lengths for some spacing in (u) and this equal spacing in u does not guarantee equal arc lengths that we have established. So in some sense I am trying to re parameterize this curve in such a way that, that parameter gives me the point or the location on this path at an equal interval if the parameter is within equal interval. That is a desirable situation.

Now the question is can we do this? We can. How can we do it? it is a very interrelated problem. We want to traverse equal distance here which is the arc length and that is what we want to use as a parameterization to give us the curve. We are sort of looping around. It is not analytical so I cannot solve it in an analytical way. Let us try to see how we compute the measure which we are using from a given curve which is the arc length.

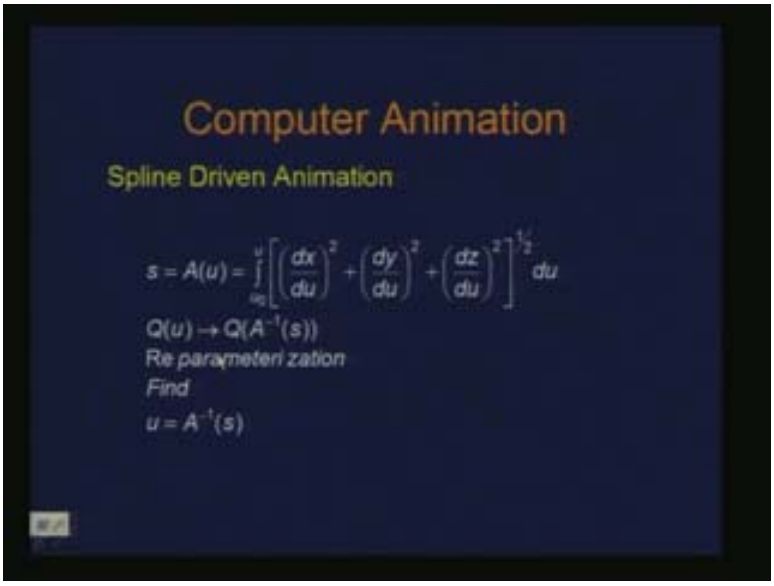
(Refer Slide Time: 06:02)



So, if I assume that the curve is a cubic curve represented through this a b c and d here would turn out to be the vector terms because I can get the x coordinate also using this, y coordinates and z coordinate. Given that I have a cubic spline curve represented in this fashion which is shown here and at a distance or an interval of du on this curve I have point a Qu and i have a corresponding point Q(u plus du) after the interval du.

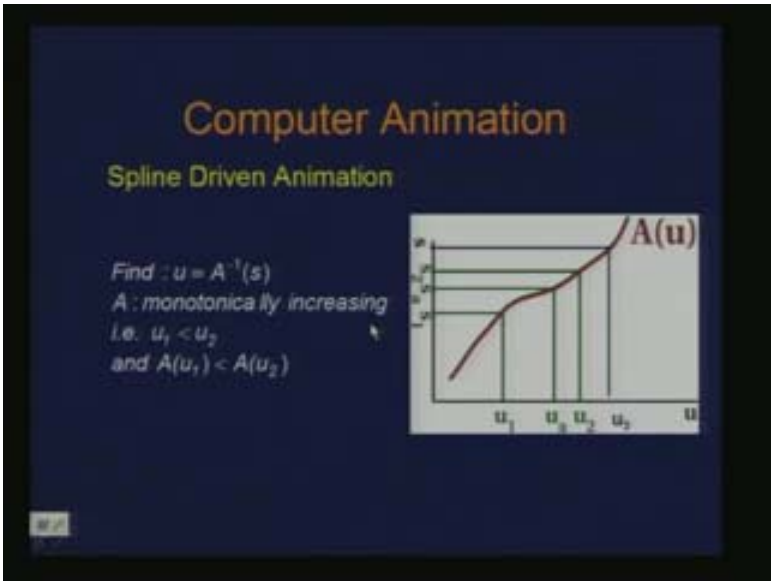
Now if this interval is small enough I compute ds which is the arc length so I am assuming this is equal to this for a small enough interval then this is nothing but the Euclidian distance which would be computed between this point and this point which is nothing but square root of dx square plus dy square plus dz square. Now the fact that I am using a parameterization in (u) I can always represent this is in this form it does not really change anything so I have ds that is the arc length for a point Q(u) and Q(u plus du) given by this.

(Refer Slide Time: 08:21)



Now I take an integral integration of that which I call as A(u) from some starting point for the parameter u_0 to (u) and I integrate this and this in fact gives me (s). I am basically establishing a relationship between (s) which is the arc length and the parameter (u) from where I compute the curve. In other words, Q(u) which is the parametric curve I have becomes this because I can get (u) as taking A inverse of (s). Now the fact that A is not an analytical function so A inverse (s) cannot be computed analytically so we are trying to find out this quantity which is u is equal to A inverse (s) in some fashion knowing that this is not analytical. In other words, we now re-parameterize in (s) with this relationship. Either you say (s) is equal to A(u) or you say u is equal to A inverse (s). So, as far as the formulation is concerned we are able to do a re-parameterization in (s) of Qu. Now we have to figure out how we do that computation.

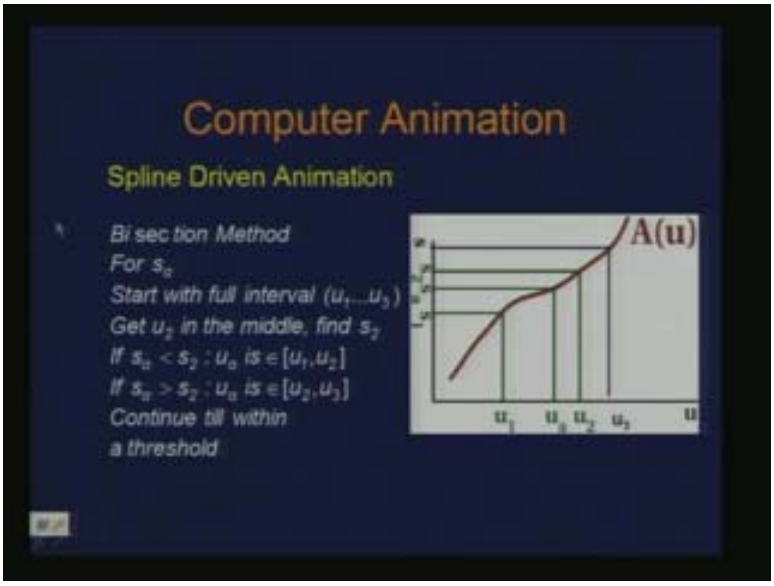
(Refer Slide Time: 10:23)



So what is the problem we are stating? We are stating the problem as find u is equal to A inverse (s) given that we can actually compute (s) in some fashion. There is a property which can be of help which is that this A which is the relationship function between (u) and (s) gives us a monotonically increasing value and that is the fundamental property we need to have for the parameterization in the sense that if I have u_1 less than U_2 then $A(u_1)$ is going to be lesser than $A(u_2)$. So this monotonicity has to exist. If I have A(u) given like this then it has to monotonically increase from u1 to some value U_3.

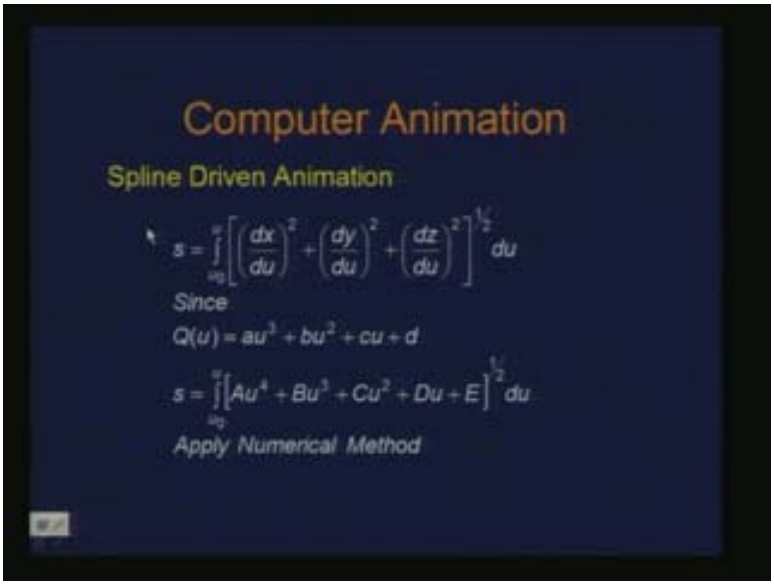
Now the statement of the problems become that given the total interval of u given this relationship as A(u) how do you find out (u) for some value of S_{alpha}. We can use this property. We can actually do this using called as bisection method. So it is a bisection method which exploits this monotonicity because for some S_{alpha} one can start with the entire interval you have of (u) in this case U_1 to U_3, you obtain U_2 which is in the middle find S_2 corresponding to this U_2.

(Refer Slide Time: 12:27)



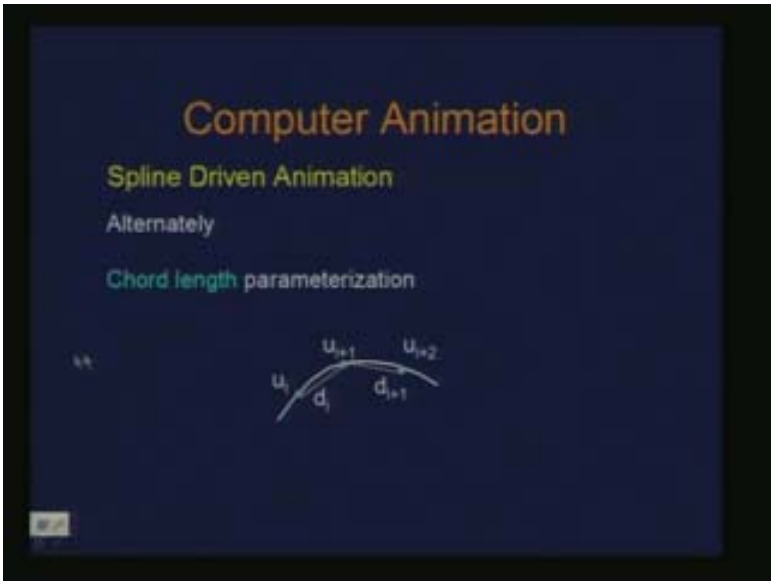
Now, for the given s alpha which is known to you, you look at how this s is related to S_2 and use the monotonic property stating that if S_{alpha} is less than S2 would basically imply that U_{alpha} is in the interval U_1 U_2, it is on one section of the interval and which is U_1 U_2. If S_{alpha} is greater than S_2 U_{alpha} is in the interval U_2 to U_3. So you keep doing this bisection depending on how you are located in each of these intervals. So once you have figured out that S_{alpha} is less than S_2 and this happens to be the interval and you are still far from S_{alpha} having computed S_2 so you compare S_2 to S_{alpha}. If the difference between them is large you again do the bisectioning and keep doing that until you satisfy some threshold. This is using bisection. Now we have basically computed U_{alpha} provided that we could get S_{alpha}. That is an underlying assumption. Now how can we get S_{alpha}?

(Refer Slide Time: 15:12)



So it is basically looking at how do we get s? So this is the formula for s which is an integration of this. Now Q(u) is known to you in parameter u and let us say we assume this to be a cubic case then I can actually do a substitution of the individual terms for x and y and z from Q(u) which will be in terms of u, take the derivative here with respect to u, substitute here, do the squaring, take the square root and get something like this. So basically this expression can be written in terms of u. Now what we need to do is get this s. Now this is sort of hard. Analytically you cannot find out therefore you have to apply some numerical method. One can do it but there are still problems. Even doing this is not so easy so alternatively what you can do is you can use chord length parameterization which is a straight forward method.

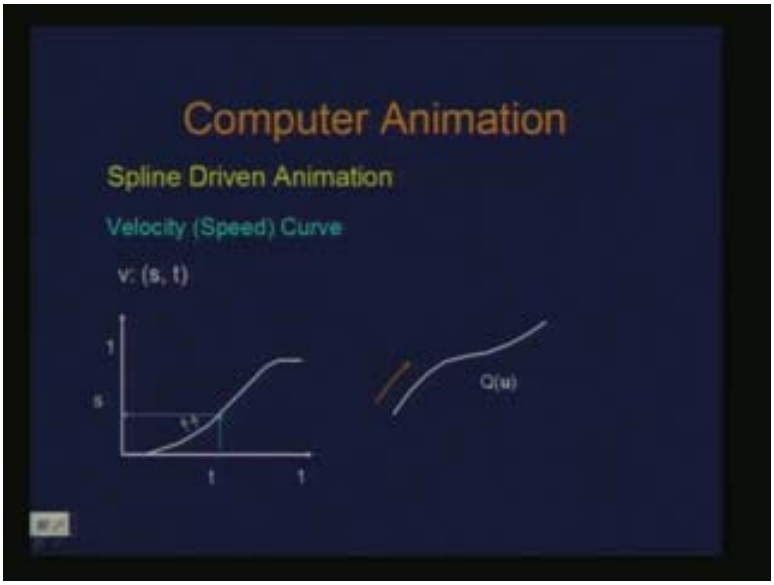
(Refer Slide Time: 17:16)



There we are saying that this parameterization (u) is actually determined by the chord length between the data. So if I have that data at different points all we are doing is computing this chord length which is the distance between the two consecutive points and use that as parameterization. Chord length parameterization in some sense approximates arc length parameterization if you think that these chords are small enough and would capture the distance along the curve. So computationally it becomes handy and you need not do any integration.

Now we basically achieved that there is a way by which you can find out the position of equidistant points on the curve using re parameterization. That gives us a possibility of obtaining uniform speed. But what if you want non uniformity? There is some sort of a notion of velocity or speed curve which actually relates (s, t) t is some parameter which is your time line and s is your positional parameter.

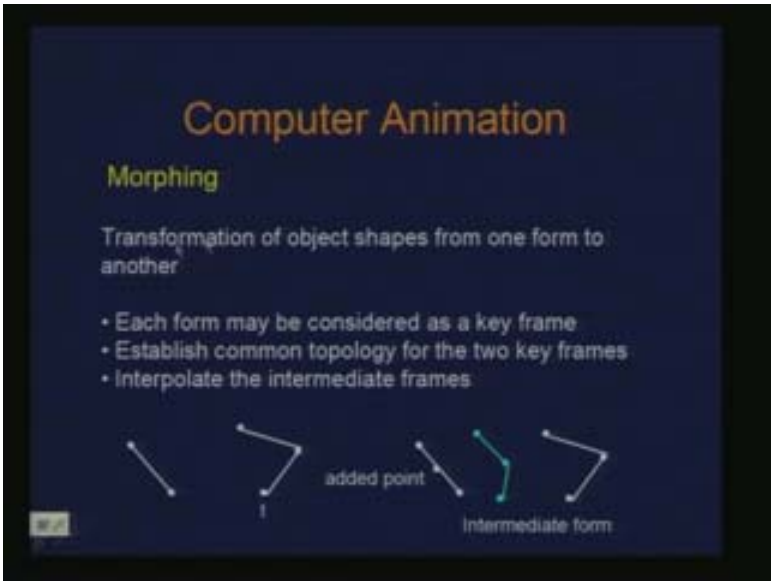
(Refer Slide Time: 19:40)



Therefore, the combination of this is basically determining the speed. So this curve for instance is an indicator to a slow start and a slow end. Now incorporating this is very easy because the first step is going to be given this t parameter which is the indication of your frame number or the time where you want to compute the position along the curve. You get the parameter s through this function which is known to you, use this s for getting this so one two and three that is the third step and that would look at your point here. So we can incorporate this velocity curve to locate points along this curve. And this itself could be a spline curve.

Splines can be used not only to give you spatial path but can also give you this speed path or speed curve. These are some of the applications we can think of using splines other than just doing an interpolation of the key frames like earlier we said that key frame in-betweening can be done by interpolation and that interpolation can be linear interpolation or it can also be using splines. But this is in a slightly different context because here we are specifying the trajectory or a path and we want to move along that. This is useful if you want to design a walk through of a scene. Here is an example of generating animation.

(Refer Slide Time: 23:25)



This is actually an example where we illustrate the concept of key framing and how do we then perform the in-betweening of those key frames. So here what we are saying is that morphing basically means some sort of a transformation of object shapes from one form to another. There are lots of commercials now-a-days in television where one form changes to another form or even the special effects in many of the films. Each form from where we want to change or which is a particular shape of the object is considered as a key frame.

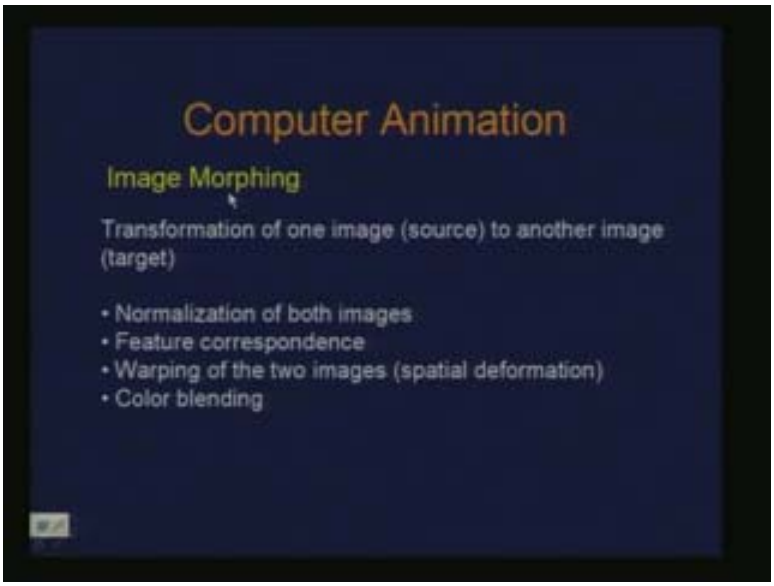
Now there is an issue that if I represent one object shape using a key frame and then I need to perform an in-betweening between the key frames I must have a common topology before I can apply a linear interpolation or an interpolation for that matter. What do I mean by common topology? Here is an example. You have a shape like this, you have a shape like this so this is one key frame and this is another key frame. What you are trying to achieve is a transformation from here to here.

Now the fact that this segment has got only two points and there are two segments here and three points I cannot apply just any interpolation unless I have same number of points and same number of segments. So one has to establish some common topology between them before you can perform an interpolation. So, one possibility could be that you add a point here. If you add a point here then you are saying that there are actually two segments three points here also two segments three points and then you can perform interpolation to get an intermediate stage. This is a very contrived simple example. Real life situations are definitely complex and hard.

Imagine there is a horse and there is a cat and they have different number of polygons different number of points and you have to morph horse to cat, it would be too hard but it is possible. There is another example of morphing. Morphing is a generic term which is a metamorphism or transformation of one shape to another shape which could be in 2-D or

in 3-D. Let us look into it through image morphing so we are restricting it to a two dimensional case. This illustrates to you the concept of key framing because there is a notion of assigning these key frames and then performing an in-betweening.

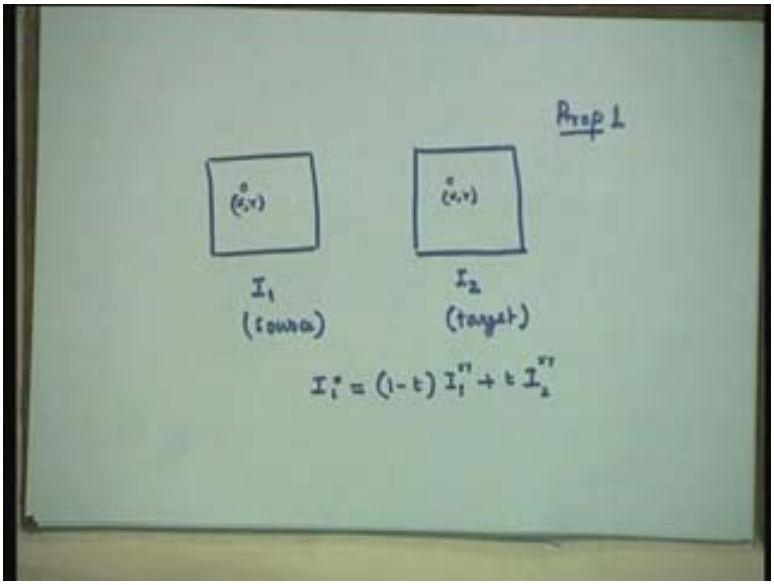
(Refer Slide Time: 28:07)



Here we are talking about image morphing. Here the problem statement is that you want to do a transformation of one image the source image to another image called the target image or the destination image. Again there are certain requirements you need. When I am talking about a transformation which is happening between one image and another image just like we were discussing about topology here one should have the sizes of the images to be the same. So I need a sort of a normalization which will make these two images of the same size.

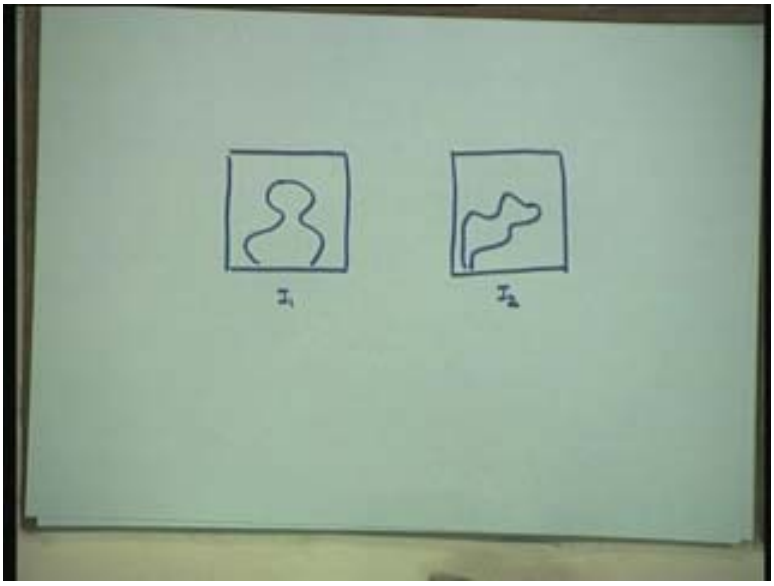
Here is a question, if you were asked to do image morphing without knowing anything about it, how would you do it? Have you understood the problem? There is a source image and there is a target image and all you are trying to do is obtain the in betweens of the source and target.

(Refer Slide Time: 32:15)



Here is a proposition, proposition number one. I have an image I call that I_1 and we have performed normalization. That means I have another image I_2 which is the target image. So this is the source and this is the target or the destination and they are of the same size, they are normalized. So proposition one says, do an interpolation of pixel colors. There is a pixel here and there is a pixel here. So you have (x, y) here and you have (x, y) here in the same location. So, at any intermediate stage call it Ii it is some sort of an interpolation, let us do linear interpolation, I call this as I1xy plus t I2xy. Let us take a concrete example now. I just put two images like a box, there is one creature here and another creature here. Basically I want to do some sort of a transition from here to here which we are calling as morphing.

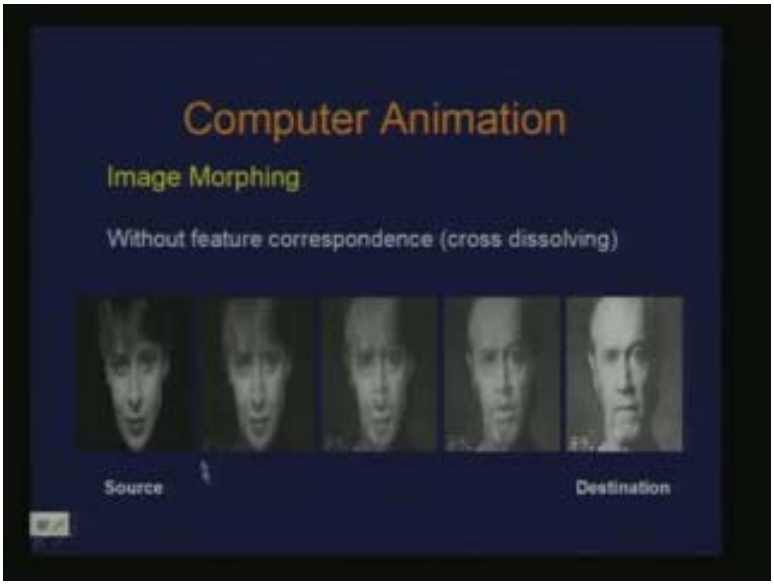
(Refer Slide Time: 34:16)



Now the problem is that, using this method you are not bothered about the spatial features. All you are doing is intensity interpolation of the pixel. So you are basically doing a cross dissolving of the two images overlaying one image on the top. So, if there are two human beings not necessarily fully aligned in the image the cross dissolving is going to have at some intermediate stage four eyes, two noses etc.

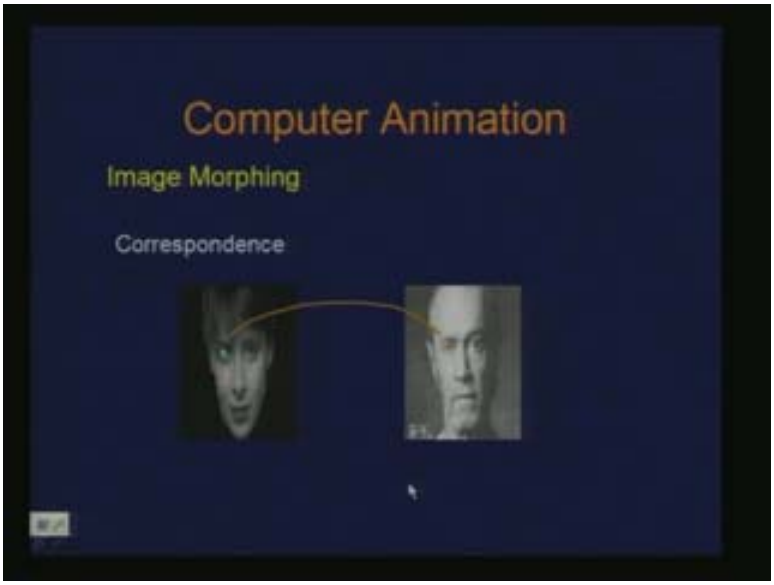
There has to be some feature correspondence and without that this will not work. And once you establish these features between the source and the destination then some spatial transformation or deformation is happening to carry the feature point from one key frame or from source to the destination because this feature may be located at different points. So there is a spatial transformation or deformation which is happening for the feature. Then you basically do this deformation for the intermediate frames as well through interpolation for instance and then do a color blending.

(Refer Slide Time: 36:48)



Color blending is what you proposed as to have the blending of the two images. So, this is the outline of the method. So let me illustrate it further. This was source image and this was the target image and if you see here there are 1 2 3 4 four eyes, two noses etc. So this is when no feature correspondence was done. It is just a cross dissolve. All was ascertained is that the size of this image and the size of this image are same. And if you look at grossly they are in some sense aligned. It is not ninety degree rotated or anything. Even then you have problems. This is actually called as ghost effect, two eyes turning to four eyes like a ghost.

(Refer Slide Time: 37:50)



Now, to fix this we need a correspondence. That means this eye needs to go to this eye and so on, nose has to go here. So I need to specify the correspondences between the source and the target. This can be done manually; this can also be done somewhat automatically. So if you have done image processing may be you can do some automation otherwise you have to specify that this point goes there. So what we have done is we have chosen a certain number of features which we want that they should correspond between the source and the destination. There are several methods and one of the method is using triangles for performing this which would enable you to specify or obtain the spatial deformation which I was referring to.

(Refer Slide Time: 39:02)

So what we have done is we have basically marked the feature points on to the source and the target image. Just marking these features points does not really solve the problem but we need to figure out what is the transformation which has happened. As long as these features are concerned you can establish the spatial transformation because you know that this point has gone there so you know the offset in (x, y) for a particular feature point but what happens to the rest of the pixels in the image. In some sense you need to obtain a deformation of the entire image through these feature points. So for feature points you are saying yes that this point goes there but for the rest of points you would use that information for the feature points.

Basically we obtain the warping function or the warping transformation for the image. So what you have done is you have given these points as the feature points which would correspond between the source and the target. So this point goes there, this point goes there and so on. Now what you have done is you have applied a triangulation of these points in order to cover other points in the image. In a way what we are saying is that this triangle corresponds to this triangle. So, the transformation is from this triangle to this triangle. Therefore for each triangle I know the transformation through the feature points.

Then it is a question of finding out for any pixel inside this triangle what the transformation is?

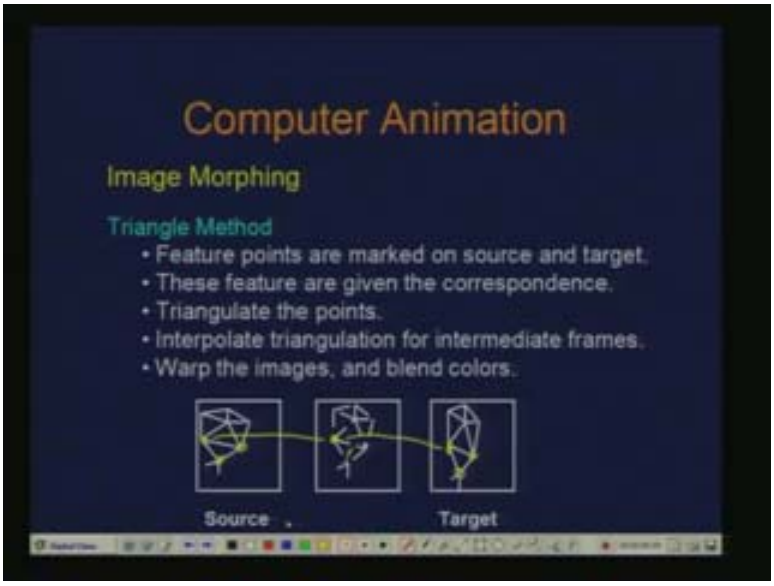
So, one thing is that this triangulation which we are applying here has got to be the same here topologically. if I join this, this, this point to make the triangle I should have a triangle joining these three points here that is what well establish the common topology of the triangulation.

If I obtain some other triangle joining this, this, this then I am in trouble. So I need to have the common topology between this and this. It is the same triangulation with respect to the vertices. It is just the location of those vertices which is changing but the connectivity of the vertices is the same. If it connects to vertex 1 2 3 here it also connects to vertex 1 2 3 here. This is what is happening to the source and this is what is happening to the target.

Now we are interested in getting some intermediate form. The intermediate form is basically given to you for a particular time interval. If I say that this is at t is equal to 0 this is at t is equal to 1 so I am looking at for some t which is 0.5 times so what happens here? All I need to do is use this time as the interpolation parameter and perform an interpolation between this and this which are the triangulations and get this. Now I am basically able to perform the shape transformation in time between source and destination.

For any frame between source and destination I can obtain spatial transformation which is nothing but an interpolation from source to destination. The next question comes, how do you render this? You obtain these triangles but how do you render this because ultimately what you need is to render the image here. As far as this point is concerned which happens to be the feature point marked, this point corresponds to this point and this point corresponds to this point and this point corresponds to this point and we obtain the intermediate frame like this.

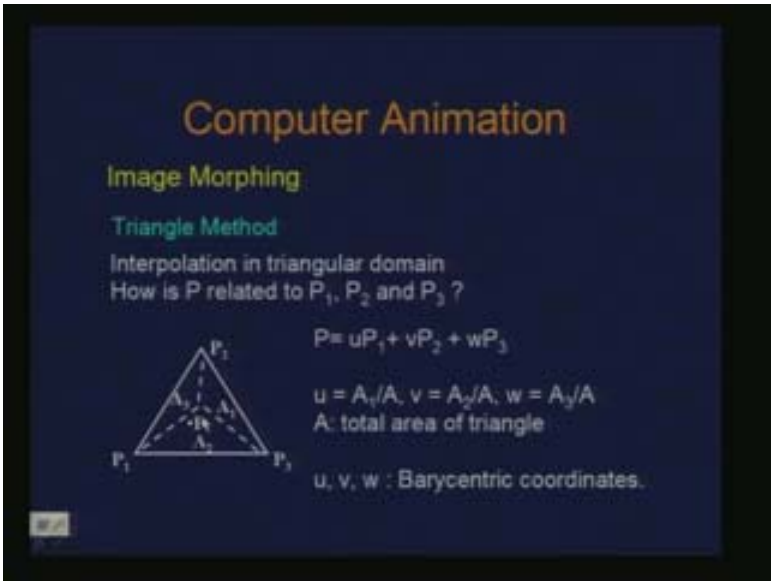
(Refer Slide Time: 47:58)



So this point is basically an interpolation from this point and this point. Similarly this point is an interpolation from this point and this point. So, if there was only a question of rendering this point then all I would have applied the same interpolation to the intensity. I know this point, where it is located; I know the corresponding colors in the image or the pixel values in the image, here also I know the point where it is located and therefore I know the color values of this and the pixel values of this. So all I need to do is perform a linear interpolation between the colors here and here to get the color for this. So it is basically interpolation not only at the position level but also at the color level for this point. When I say for this point therefore you can do for all the feature points the same thing, just an interpolation of the color intensities.

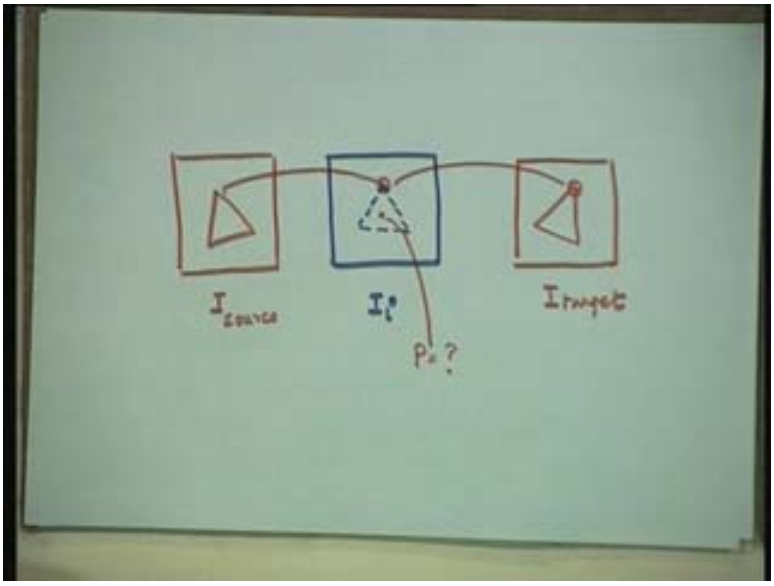
Now what would you do for this point?
Within a triangle if you want to perform such an interpolation what do you use? You use barycentric coordinates and that is what we are going to do here as well.

(Refer Slide Time: 48:35)



So basically we are looking at an interpolation in triangular domain. So given a triangle through P_1, P_2 and P_3 you are representing a point P inside the triangle using P_1, P_2 P_3 and that is through the barycentric coordinates u, v and w and this is how we compute the barycentric coordinates using the area. So how these barycentric coordinates help? We can locate a point inside triangle but how is it helping? The question is, there is an image which is the intermediate image.

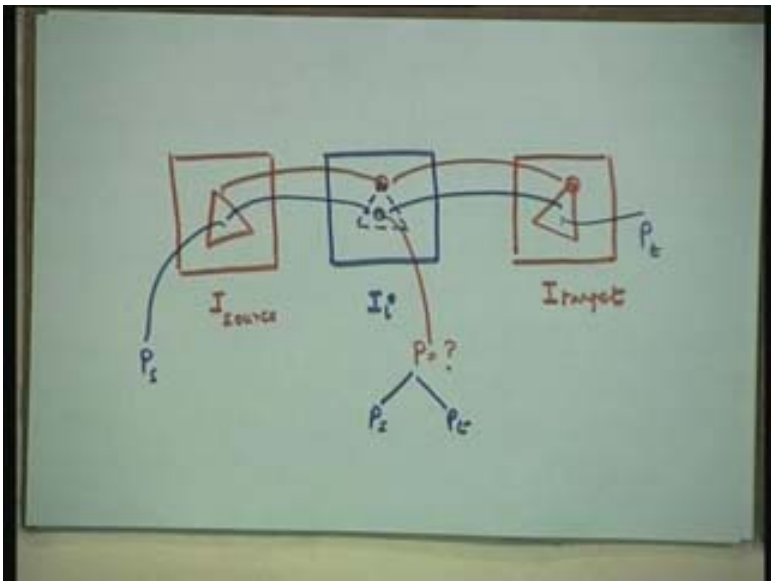
(Refer Slide Time: 51:43)



Now this triangle is obtained from these two triangles. So this is my source and this is my target. All we are saying is that this particular point corresponds to this, this particular

point corresponds to this and so on. So these points have been located now spatially where the (x, y) of this particular point is, we want to render this. And you already know the mechanism of rendering this point which is to take the intensity interpolation at these points. Now the question which is being asked is what happens to this point? Perhaps we can use barycentric coordinates, so how can we use barycentric coordinates? This point P corresponds to some point here and some point there, so let this be P target and this is P source.

(Refer Slide Time: 54:54)



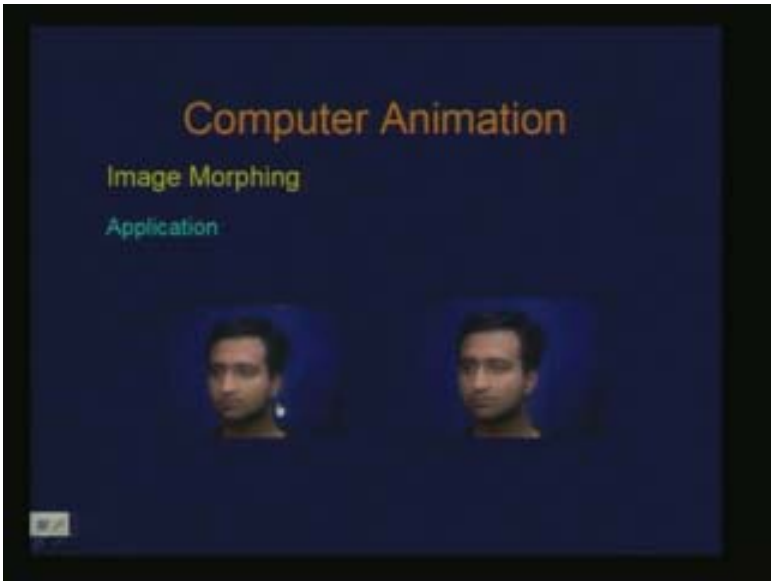
Now the question is, how this P is related to P_s and P_t? It is exactly the same as the vertex point. This P is inside this triangle. Given the point P here can I locate P_t and P_s using barycentric coordinates, so that is where we use the barycentric coordinates. Now I would render this triangle and that is the question we are trying to solve.

In other words, I am going to find out the pixel values of all the pixels within this triangle. So given this point or given this pixel inside the triangle I can obtain the corresponding points for the target and the source which are P_s and P_t and now I can perform linear interpolation or whatever interpolation I am using for the intensity colors for the point P_s and P_t. So using this I can render the triangle and if I can render one triangle I can render all the triangles and get the intermediate dimension. This is again the source, this is the destination and you see that the eyes and nose sort of correspond. Here is an example, this is a tower in London and this is your Eiffel tower so you want to make a morphing between these two towers.

(Refer Slide Time: 55:20)

Now there is another application you can use through image morphing. So here in fact an interface is built where an audio visual output is produced given the text of the speech.

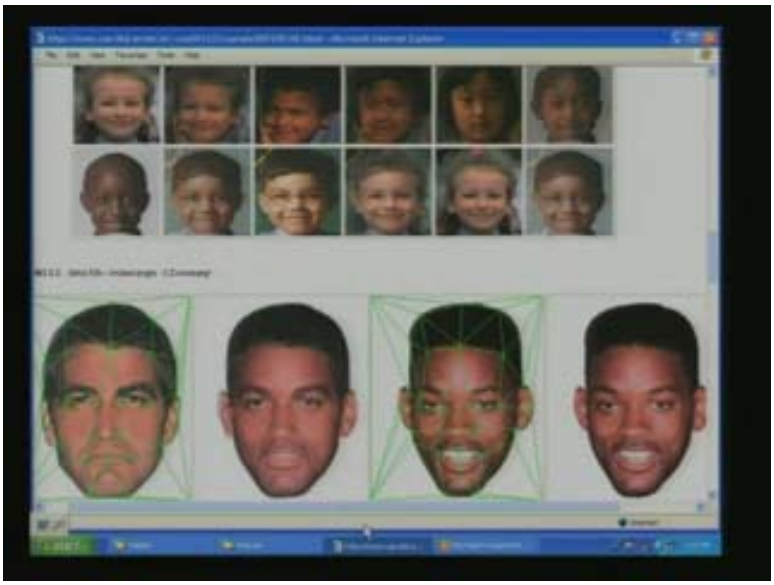
(Refer Slide Time: 56:01)



So there is a database made for different phonemes, phonemes are the unit of or the lowest component of your speech. So there is a corresponding visual corpus or counter part of the phoneme so when you say P then the lips are closed, when you say R then the mouth is open so these form the visual counter parts of phonemes so we call them visiems.

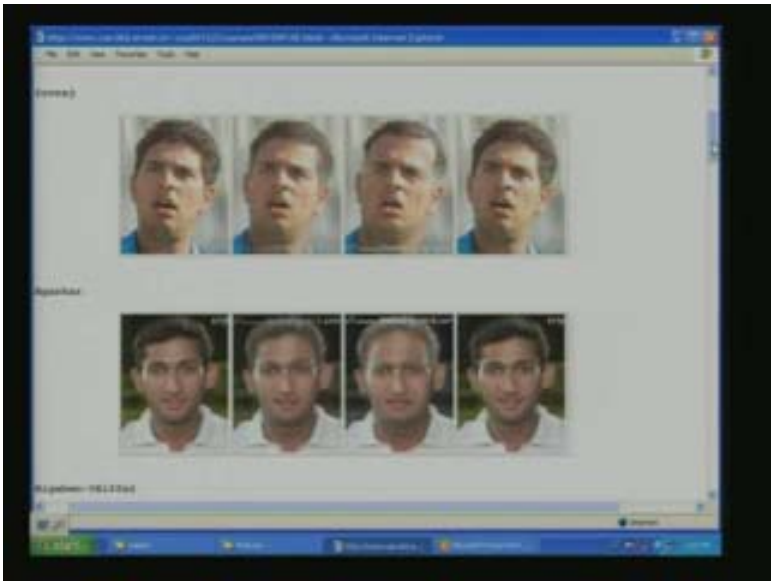
In fact you can do or perform a morphing when you know the sequence of phonemes required to alter a word or a sentence. You can process the speech getting these components in terms of phonemes therefore the visiems and then perform morphing of them. Here in the slide you could make out that he is saying something like how are you and similarly here so basically you get impression of visual speech. Here is another example of this morphing. Here this is one source and this is some sort of a target, this is an intermediate formed and this is the transitions you observe.

(Refer Slide Time: 57:45)



And here in fact number of key frames are used so it goes from 1 to 2 to 3 to 4 and so on. So this is through morphing. Actually here this is some sort of an aging process.

(Refer Slide Time: 58:48)



This is an image of Yuvaraj and this is an old Yuvaraj which somebody made, so here you see how aging is taking place as a process known as morphing.