Introduction to Computer Graphics
Dr. Prem Kalra
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
Lecture - 35
Animation (Contd...)

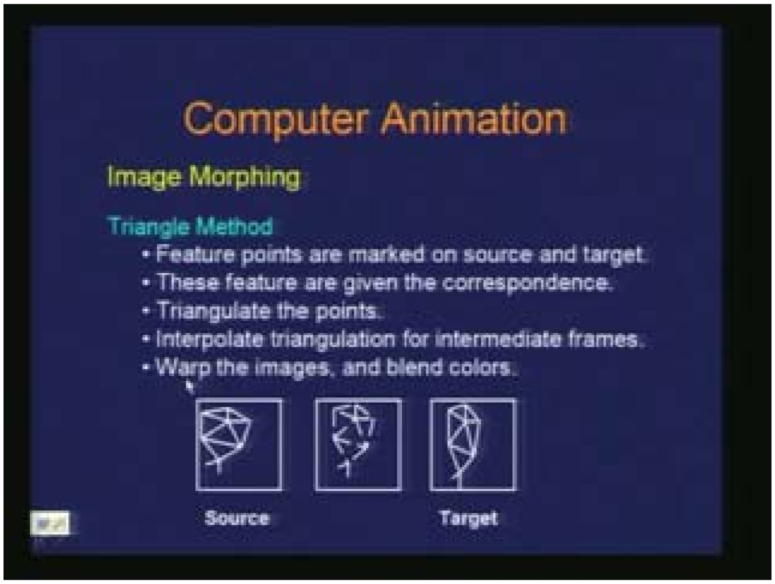
We have been talking about animation. If you recall we basically have looked at various animation techniques.

(Refer Slide Time: 01:05)

The major techniques could be classified into techniques of rotoscopy which is extremely data intensive. Then we have techniques involving key framing which is very popular. Here the idea is that you try to generate certain key frames and then using some sort of interpolation you create animation through those key frames. Then we also looked at parametric animations where the key framing could be formed as parameters which drive animation rather than the complete shape of the key frame. And there are also methods which are algorithmic which more or less attempt to simulate the process being governed by a physical law or certain heuristics you have in mind.

In fact one can also look at these techniques in a different form. For instance, rotoscopy in modern day animation can be looked at as a technique of motion capture where you are trying to capture the motion from a real performer and then try to use that information for generating animation. You can also have scripted animation where you write a script consisting of various commands involving key frame animation or parametric animation. And you can also have procedural animation where you set the procedure of animation may be based on law of Physics or law of motion or it could be a procedure which is designed by you so it is a procedural animation. Therefore, these are similar ideas as these but with a different nomenclature. We have looked at an example of key framing animation in the context of image morphing.

(Refer Slide Time: 03:50)

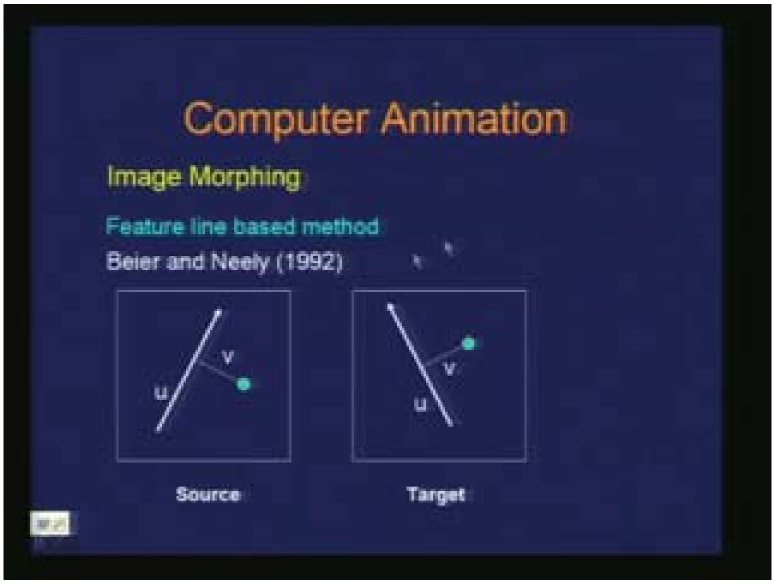


Just to recapitulate what we looked at last time, the problem was stated in the form that you have a source image and you have a target image and what you want to generate is a metamorphism between the source and the target. That means you start from the source and you reach the target preserving what the target is and in between you have some sort of a blend between source and target. And in order to have the realistic transition from source to target it is necessary that there is some sort of a feature identification which you or the animator provide as the corresponding feature between the source and the target so that you preserve those features during transition.

Once these feature points are identified then in the triangular method you triangulate these points which are feature points. That means you basically establish a coordinate system for the rest of the points in terms of the feature points you have given as corresponding points. So, triangular method is one of the ways of doing it. And once you have done that then you perform an interpolation of the triangulation for getting the intermediate frames.

Perform an interpolation both in terms of the spatial information which gives you some sort of a warping function from one frame to another frame and also for the color or the intensity information to find the intermediate frame. There is another way to do the similar process.

(Refer Slide Time: 06:03)



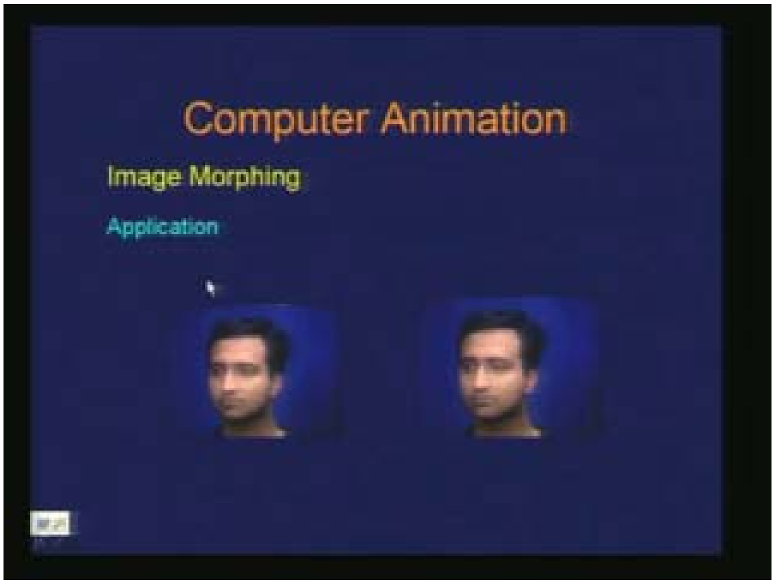
This is similar to what triangular method is except that you are establishing the features as feature lines in the image instead of specifying points and thereby using triangles as corresponding primitives between the images you are using these feature lines. So you are saying that this line matches to this line and this is the corresponding line in the target. This is due to Beier and Neely in 1992 and in fact that is where they had created the famous animation or the morphing clips of Michael Jackson. This appeared in [SIGGRAPH of 7:10] in 1992. Here again the problem is that you need to find out given this pixel in the source how do you transform this pixel into the target image knowing that this line corresponds to this line in the target image.

Again you need to have some sort of a coordinate system which would locate this point in the relation to the feature line you have specified so that you can do the proper interpolation for the points. Again this coordinate system is established through the parameters u which is along the line and then v which is the orthonormal distance from the line. So by this way you locate this point. Similarly the same u and v would locate this point. So these are equivalent to some sort of Barycentric coordinates which we have seen in the triangle method.

This is with respect to single line but there will be situations where you would actually specify multiple lines of feature correspondence between the source and the target. So there could be a situation when there is one line here and there is another line here and so on. So a point here would actually get affected by more than one line with this system of establishing coordinates. So you need to have some sort of a weighting or weight factors associated to these coordinates. So one of the ways by which you can do is that you can define a decay function. Closer to the line you get influenced by this line and as you go farther from this line the influence is reduced. So basically this mid point may get influenced by some sort of an attenuation of distance from the various lines.

This is another way of doing the same thing. Instead of using a triangle method you use feature line. In fact it is thought that sometimes it is easier to identify primitives like line which would be considered as corresponding features rather than points. So that way these methods have certain advantage to point wise method. The application of these methods was also dealt with where you saw the morphing.

(Refer Slide Time: 10:49)



Here in this video you see that there is an utterance of a sentence which is given as an input as a text which was processed to figure out the atomic units of speech which are the phonemes and the visual counterparts of those phonemes which we call as visiems. Then what you have is a snapshot of certain characteristic visiems such as ha, oh, p and so on. There is a chosen set of these visiems which are just snapshots or stills. And then what you are trying to do is some sort of a morphing between those visiems when you run through a sentence.

For instance, if you are saying “to” then it is a combination of t and o then you mix these visiems during your animation. And the correspondence since we are taking the entire image is actually hard to do either using triangular method or using feature line based method. In fact there is a method known as optical flow which helps determining the correspondence for the dense pixels. It actually gives you correspondence for each pixel in the source and the target so using that you establish the correspondence and then do the morphing.

Obviously when you look at the eye motion it was not done using that because when these utterances where taken for vision both the times the eyes were open. So this was like an additional feature overlaid on this.

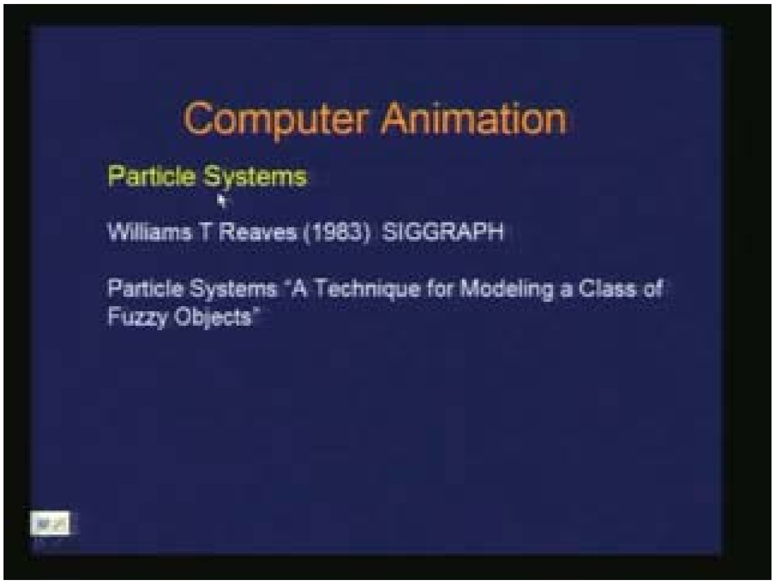
For instance, one can do masking of eyes so you do not display the eyes or you can have one key frame as a closed eye and then one key frame as an open eye and then you again run through morphing. So this is an overlay on the speech. Here is another clip. The head motion you observe is actually an artifact and not an intention. The subject is, not supposed to move the head because otherwise you lose the information of correspondence. When you are doing the correspondence you want the snapshots to be in the same neutral head position. But the feature here in fact which was intended was to demonstrate the articulation of speech.

What we mean by articulation is, basically when you utter a word depending on the preceding and succeeding phonemes the current phoneme shape is formed. It is not an independent entity after all, it very much depends on the preceding and succeeding phonemes. That means one has to take some sort of a window in which you can perform the blending of these phonemes. It is not just a pair-wise blending but you may have to take a window of phonemes where you need to perform the blending. So, one needs to perform some sort of a ranking of phonemes. There are phonemes which are dominating, there are phonemes which are less dominating so those problems are addressed for co-articulation.

In fact another thing which you observe is, this is completely image based so everything is done on an image. But one can do a similar process using a model. Basically you construct a three dimensional model of a face and in order to add a realistic display of that you may use texture mapping. That is to map the information of such a realistic face on to the model which can be achieved through texture mapping. Then it becomes a matter of moving those points on the mesh which is giving you the model of the head which could be done parametrically or which could be done by other methods.

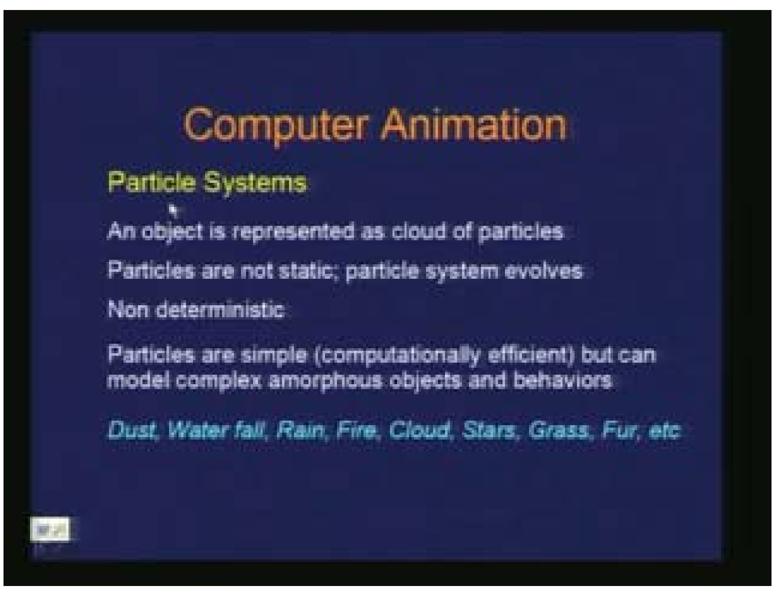
In fact you can go to the extent of defining muscular structures within those models and then pull these muscles giving forces to those muscles to see the deformation of the mesh. This can become as complicated as you want it to be. So those are model based techniques. So the advantage there is that you have much more flexibility because you have a model. Here you are fixed with a particular subject and the database of that subject. So there is a plethora of research activity around this. This has been on for about twenty or thirty years. Here is another way of doing animation. In fact we are going to talk about particle systems which can be used for animation and also be used for some other purposes, modeling in general.

(Refer Slide Time: 18:10)



So the initial attempts with respect to particle systems started in 1983 from a SIGGRAPGH paper by William T Reaves where he gave particle system as a technique for modeling a class of fuzzy objects. Fuzzy means something similar to random fractals. So the nature is stochastic. What happens in a particle system is that, first of all the representation of an object is in terms of cloud of particles.

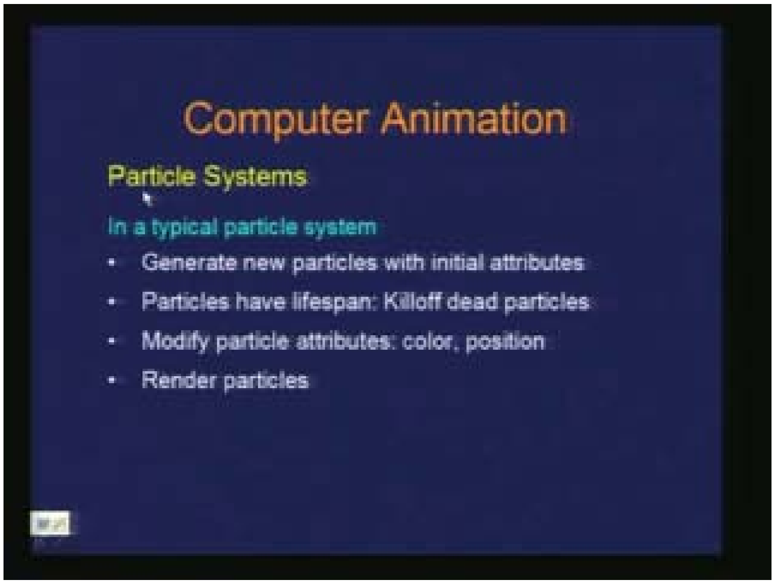
(Refer Slide Time: 19:11)



So you are thinking of an object to be represented as a bunch of particles. And these particles are not static in general particularly when you are performing animation. That means this particle system which is a collection of these, hundreds and thousands of particles evolves over a period of time and it is nondeterministic, it is stochastic so there is randomness in the process. The idea behind is that you are basically trying to do a representation in the form of particles which are simple in shape, simple in performing computation. So, to gain some computational efficiency you choose this representation of just collection of points. But they can actually model very complex amorphous phenomena, objects and behaviors.

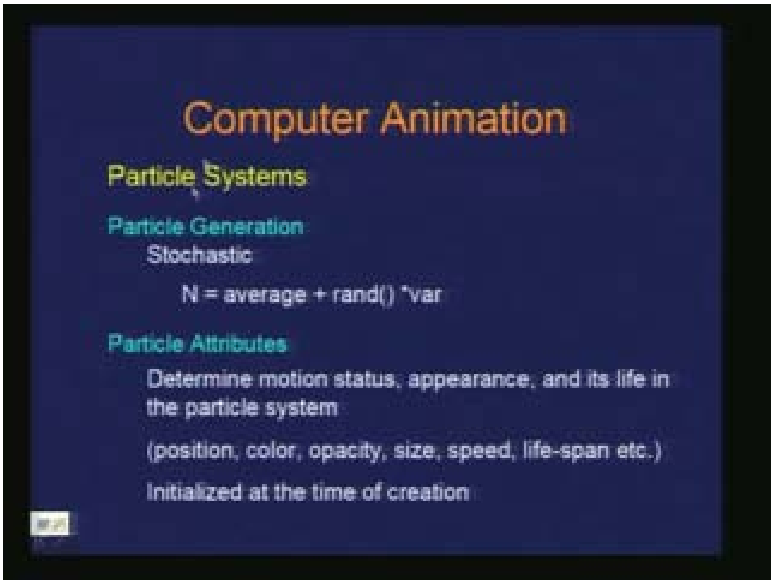
Some of the examples are dust, waterfall, rain, fire, cloud, stars, and grass and so on. Otherwise if you choose some geometric primitive to do the proper modeling of these phenomena it is going to be extremely computationally intensive. These particles systems allow you to capture the characteristic behavior of such phenomena.

(Refer Slide Time: 21:33)



Therefore when we are dealing with these particle systems, in a typical particle system you have a generation of particles, so you generate new particles with initial attributes and these particles have a lifespan. That means they die after a certain time or they disappear. In terms of computation you may still have the allocation for that particle or in the memory but you do not display it. You kill off these dead particles, you do not show them or remove them. And then the evolution requires you to have a modification of the attributes of the particle in terms of position and in terms of color and other attributes and then there is a process of rendering these particles which stay in the system.

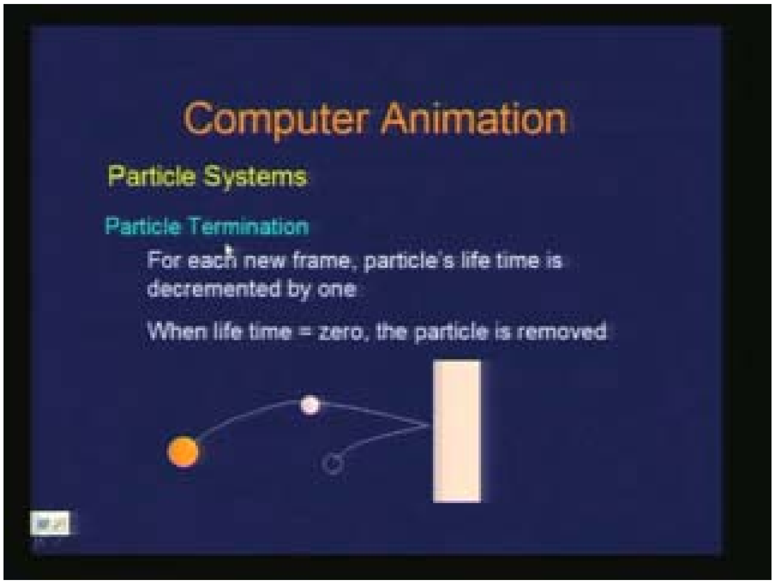
(Refer Slide Time: 22:52)



We are looking at a process of particle generation which is stochastic, stochastic in the sense that at certain time you generate some number of particles in a particle system which could be determined by the average number of particles at that instance. There is some randomness and then there is this variance which will keep changing depending on at what instance of time you are generating this particles. That makes the process of getting these particles as stochastic.

Now the attributes basically determine the motion status of the particles namely the position, the velocity, the appearance of these particles, the size, shape, color, opacity, speed and the lifespan. These attributes need to be initialized at the time of creation. You have certain values to be given at the time of creation of a particle system. We said that the particles have a lifespan so they would terminate at some instance of time. For each new frame basically you have a subtraction of the time which has been elapsed from the life of the particle.

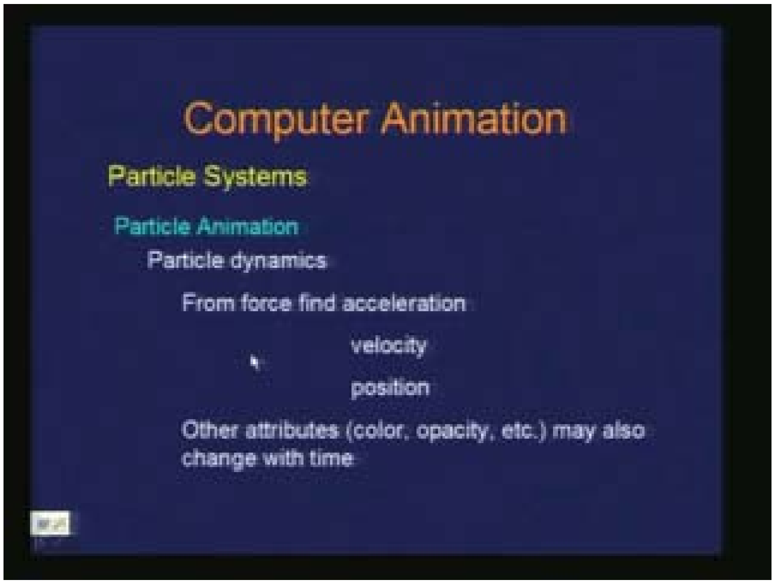
(Refer Slide Time: 24:53)



So particle’s lifetime is decremented by 1 or by some factor or whichever procedure you have determined. And when the lifetime may equals to 0 that means the particle has to be removed, it is dead. For instance, at the time of the creation a particle you may have, with these attributes the size and the color and the position then you have a law of motion or the procedure in which this particle has to move over a period of time. So it goes from here, at some point of time this is the state of the particle where the attributes have changed according to the procedure or the law you had given so the size has reduced, the color has changed and so on.

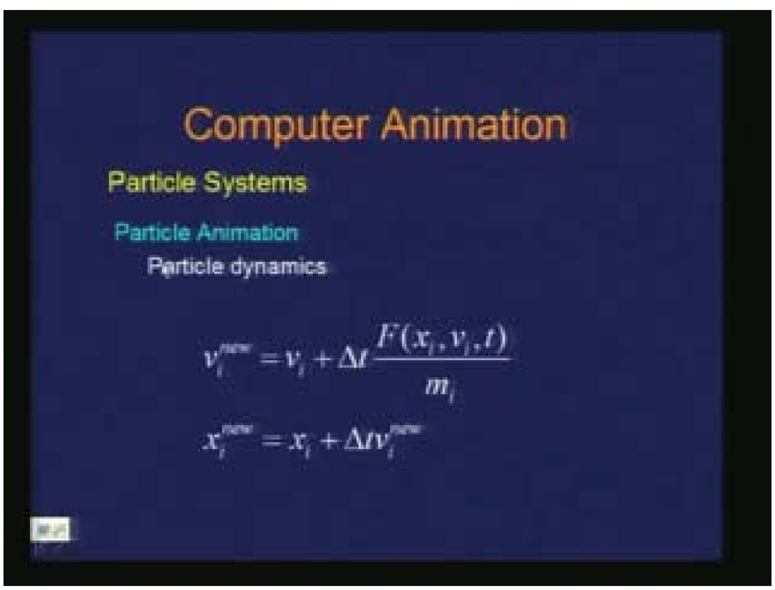
It hits this and comes here and at this instance the particle is dead. This sort of shows the evolution of the particle. Therefore, to be able to do animation of particles either you may have a procedural animation where you have described a procedure for the animation of particle or you may actually use some dynamics or law of motion which performs the animation of particle.

(Refer Slide Time: 26:43)



For instance, if you want to use dynamics then we are looking at finding out the terms like acceleration, velocity and position given certain forces on the particle. And simultaneously you may change attributes of particles. For instance, you may change the color of the particle, the opacity of the particle with time. This is what performs in the sense of evolution of particles system.

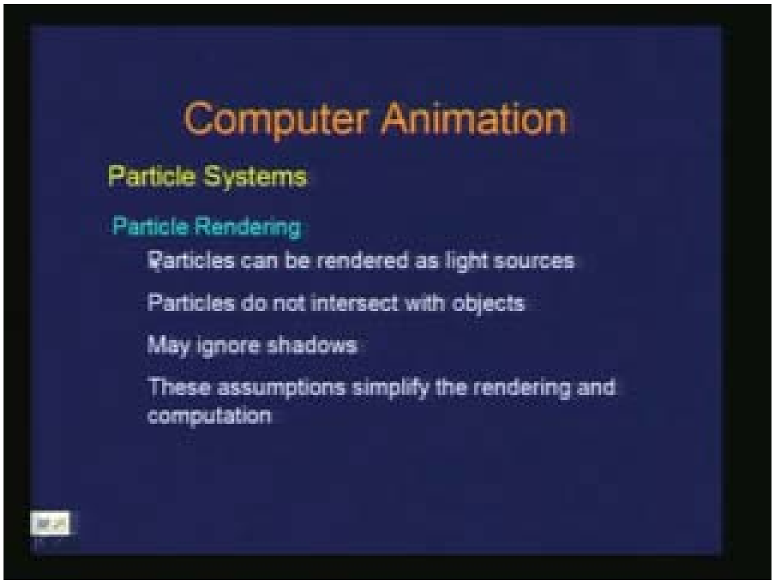
(Refer Slide Time: 27:48)



Basically you can determine terms like acceleration, velocity and position of a particle. So, if we are looking at the basic law of motion of Newton f is equal to ma so we are looking at this f which is the force term determined through the position at instance i the velocity at i and time t and this term gives me the acceleration f/mi where m is the mass of the particle. This is sort of the Euler approximation when you do the numerical solution of the differential equation because f is equal to ma is nothing but a differential equation where a is your dx square by dt square.

So when you solve this differential equation you have this approximation where you compute the new velocity from the old velocity at the time interval delta t. So, if you have delta t as the time interval this is your new velocity and from this new velocity you can compute the new position of the particle. It is a simple law of dynamics you can use for finding out the new position and the new velocity of the particle. Now the question of rendering of these particles is concerned so we simplify lots of things to be able to render a huge number of particles.

(Refer Slide Time: 29:49)

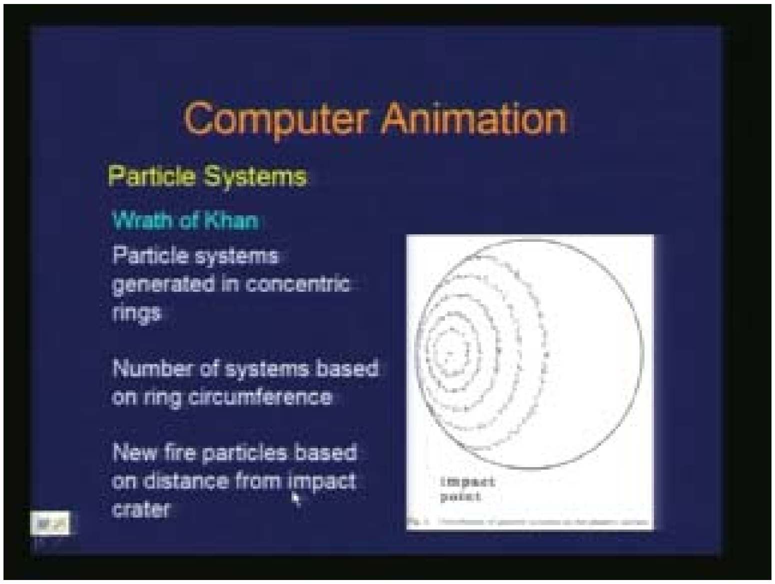


For instance, particles are considered as light sources. That means they will be emitting lights and nothing else will be happening so they are just light sources. They would not intersect with objects, particles and particles would not collide otherwise finding out collision between the particles is computationally intensive. You may even ignore shadows of particles on particles, you may have shadows of the entire particle system on a ground plane or something which may be easier thing to compute but you may not have particle to particle shadows which mean self shadows. These are assumptions just to simplify the process of rendering of these particles just because the number which we use for particle systems is huge, thousands and sometimes millions of particles are used. These are some examples. The fire crackers and other fireworks can be actually modeled using particle systems.

(Refer Slide Time: 31:09)

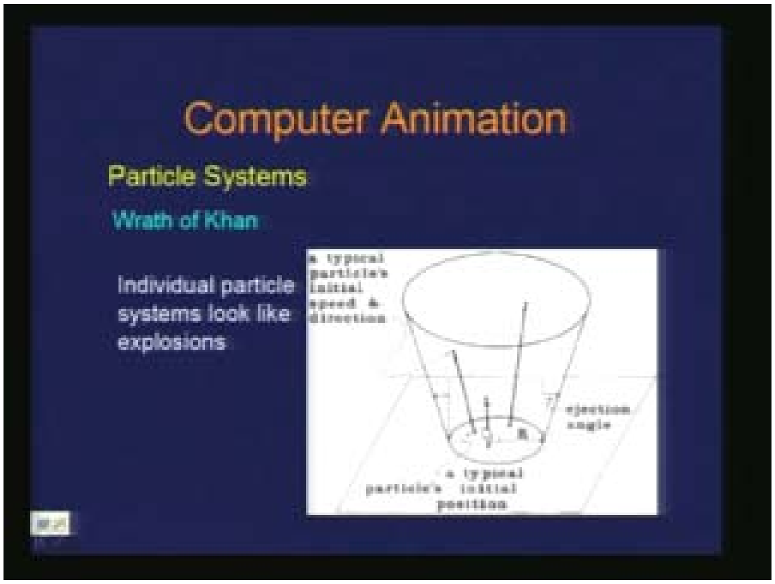
Here, in fact you have the origin of the particle system where these particles are getting generated and during the evolution the color attributes keep changing with the position and velocity and so on. Then at certain point of time they just disappear and that is why you get the impression of some particles being alive and some particles not alive. So this decay of particles is very important to the process of evolution of particle system. So this is also some sort of a comet, this is a waterfall so this is all synthetic by the way there is no real picture here. All the particles are being sort of generated at some source, they follow a certain motion and along with the attributes keep changing over a period of time. So you can get a fairly nice simulation of these phenomena. This is the Wrath of Khan which is another example.

(Refer Slide Time: 33:04)



In the early eighties in star trek II there was this showing the process of genesis and there this was some sort of a model of the planet and the idea was that they were showing how this genesis takes place. So lots of particles were generated and emanated from the surface of this planet giving you the impression of an explosion and the fire from this. So the idea was that they got generated in concentric rings from the point of impact. So number of systems of particles were actually based on circumference of the rings and the new fire particles based on the distance from the impact creator would generate it. Then you also had some sort of a process of explosion. So you have some initial position and direction in which the particle would start.

(Refer Slide Time: 34:45)



And in a typical zone what you will have is some attribute of ejection angle so it is a sort of a frustum so you have particles which are straight, vertical and there are particles which are at an angle. So this ejection angle was also an attribute to those particles. There is also an initial speed assigned to these particles. This is the kind of impression created.

(Refer Slide Time: 35:38)

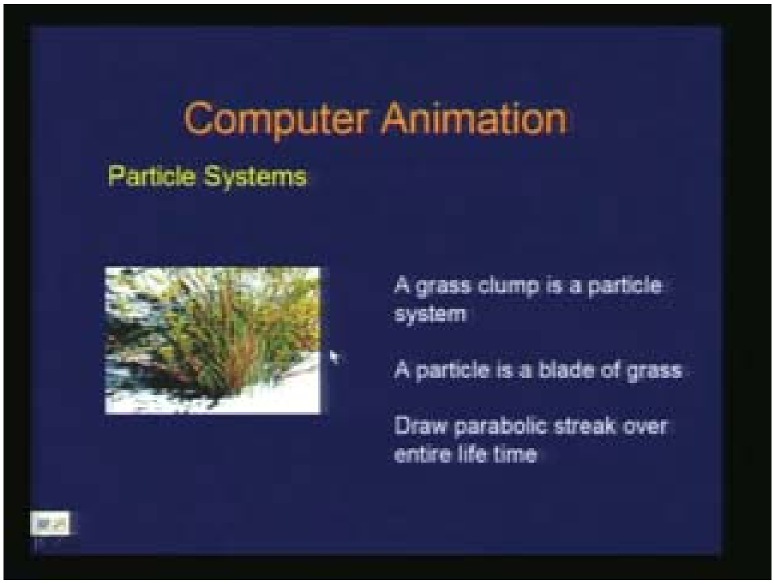
(Refer Slide Time: 36:17)

All these were done using particle system and this was done by Lucas film now-a-days called as ILN Industrial Light and Magic so it is actually a company which does lots of special effects. This is a simulation of a smoke coming from a cigarette. This is also done using a particle system. The attributes could be color, opacity and transparency, so that is an attribute you can use for simulating smoke and fire.

(Refer Slide Time: 36:48)

There is another example which is actually doing the modeling of this grass clump.

(Refer Slide Time: 37:20)



In fact a particle is considered as a blade of grass and you are drawing the parabolic streak over the entire lifetime of the particle. So it just keeps drawing these parabolic streaks which are the evolution procedure you have given to the particle which gives you the impression of this grass clump. Of course you can keep changing the color attributes.
So some of them may appear more yellow, some of them are green and so on. So here you notice that there is a shadow. There is a shadow actually on the ground plane of this grass clump. There are no shadows from grass to grass but there is shadow on the ground plane. So this could be done using some sort of a projection of these shaped primitives on to a plane. This is relatively easier than computing the self shadows of these particles. You can even do the modeling of natural phenomena not just the process of animation.

(Refer Slide Time: 39:07)

You can also further extend the idea of particle system for instance doing behavioral animation. For instance, you want to simulate the process of flocking of birds, you notice that when the birds fly and at some point they come together some sort of a banking of these birds and then they split and so on. They have a very known behavior when they do this flocking. That can be modeled using particle systems. Think of those particles now to be like birds and give them behavior through the procedure of particle animation.

Here of course the issues would come in terms of collision avoidance. There is a tendency, when you notice flocking there is an implicit leader to the flock, all birds tend to go towards that leader and the leader guides the whole navigation. However, there is no collision obviously of those birds. So when you are doing this process of simulating the flocking you will have to account for collision avoidance which is computationally quite intensive. Those birds are like particles. These particles are not just points in all situations.

In other examples where we observe these particles were more or less like point entities, clouds of points. Now you can actually use not just the point but any other primitive or an object to be called as a part of them.

All you are saying is that there is a cloud of primitives and then there is a behavior of these primitives as a system and as an individual primitive. You can also do modeling of things like deformable objects. For instance, here what we are basically trying to achieve is, if you are doing a modeling of cloth, so cloth is a deformable object it is an elastic kind of an object and it can deform. So here we are taking the particles in the form of node points in the mesh connected through springs. In other words, for instance, these are nothing but some sort of particles and these are connected to neighboring particles through springs.

(Refer Slide Time: 45:03)

So what will happen is that this point will deform as a function of the spring here in these four links and then each of these particles or node point will actually deform as a consequence of a spring action which is the connectivity of the two particles. Then you would observe that the sum of the total behavior would be a deformable mesh which can look like this which is a deformable cloth. One can do simulation of cloth or garments using particle systems. Similarly, you can also have things like hair or strands or fur. So there you are basically modeling a strand which is nothing but line connections with springs of these particles or the node points.

For instance, we are talking of these things. So here this is connected to this and this is connected to this and each one of them is something like a spring so they will contract and expand and thereby cause a deformation or bend. They may not always be deformable along the axis but they may actually bend.

(Refer Slide Time: 45:31)

For instance, that is a typical thing which will happen to hair. These are some examples where you can use particle systems.