UNIT 6
Introduction To Animation And Gaming

1) Explain deletion of segment with suitable example

currently no segment is opened.

Syllabus Topic : Segment Deletion

10.3.2 Segment Deletion

UQ. 10.3.6 Explain with illustration how segments are deleted.

SPPU - May 10, Dec. 10, May 11, Dec.19

Like segment creation we can delete a segment also.

If a segment is no longer needed or there is no work left of that segment, we would prefer to delete that segment to save space. But when we are deleting a segment at that time other segments should not get changed. Refer Fig. 10.3.2 which shows display file before and after deletion of a particular segment.

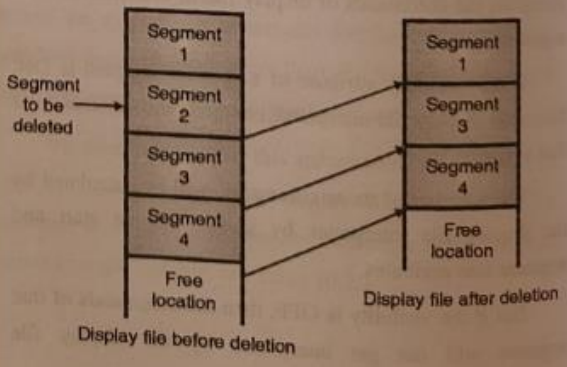


Fig. 10.3.2

We will stick to our original example of hills, bird, sun and tree which is shown in Fig. 10.1.1. We will store each sub picture in a separate segment. So there will be four segments.

Suppose the image of bird is stored in segment 2. But after some time we want to delete that segment 2 i.e. we don't want to display bird's image.

This we can achieve by just setting visibility attribute of bird's segment to OFF. But this will be temporary deletion, i.e. the commands to draw bird will be there in display file.

In later part, if we are not using those commands then it's better to delete those commands from display file itself. So that it will save the space on display file also. When we are deleting any segment, other segments should not get disturbed.

For segment deletion, we will first check segment name in segment table. If the segment name is valid and present in segment table, then we will check that segments size.

The size of segment to be deleted must be non-zero; then only we can delete the segment. But if the size of segment is zero it means there is no command in that segment. So there is no need as such to delete the segment.

From the segment table only we will come to know what will be starting point and size of the segment to be deleted. At the starting point of the segment to be deleted our next segment should start.

So we will shift next all segments up and overlap the segment to be deleted, by next segments commands till free location arrives. See Fig. 10.3.2 we have to start shifting these commands from segment three's starting point to segment two's starting point. It means we have to move all the commands up by size of segment two, i.e. segment to be deleted.

Table 10.3.1

Before deletion of segment 2

Segment Name	Segment Start	Segment Size	Visibility	.....
1	1	5	ON	
2	6	3	ON	
3	9	5	ON	
4	14	6	ON	

After deletion of segment 2

]	Segment Start	Segment Size	Visibility	.....
1	1	5	ON	
2	6	[0]	OFF/ON	
3	9 – 3 = 6	5	ON	
4	14 – (3) = 11	6	ON	

Size of segment to be deleted.

So by shifting segments up, we are filling the gap which will be created by deleting a segment. But this will be just for display file. Now we have to update segment table also which contains information of images. If we want to delete segment two, then the status of display file, before and after deletion of segment two is shown in Fig. 10.3.2. Similarly the pictorial view of segment table before and after deletion is shown in Table 10.3.1.

After deleting segment two, we have to modify the segment starting address of the segments which are after segment two, by the size of segment two. For segment two we have to write segment size as zero, to indicate that segment two does not contain anything. The visibility attribute of the segment which we want to delete may be kept as OFF or ON, anything as anyway the size of that segment is zero.

We can also delete all the segments at a time. For that we have to set size of all the segments to zero and initialize the free index of display file to the first location. We have to also set the starting position to one.

For deletion also we have to check a valid segment name, if name is correct then check whether it is open or not. Open means the segment is still in use, we can't delete a segment which is already opened.

2) What is Morphing and write the applications of Morphing

In case of ... frames are divided into individual components or objects called cels (celluloid transparencies). Then in-between frames of individual objects are generated using interpolation. Many times, we use complex transformations, in which shape of object may change. For example, in object like cloth, facial features the shape of the object may change. If such object are specified with polygon meshes, then the number of edges per polygon can change from one frame to the next. Thus, the total number of line segments may be different in different frames.

Syllabus Topic : Morphing

10.6.4.1 Morphing

UQ. 10.6.5 What is Morphing ?

SPPU - Dec. 12, May 17, May 19

UQ. 10.6.6 Explain morphing? What is simulating acceleration. SPPU - May 13

Morphing is a special effect in motion pictures and animations that changes one image into another through a seamless transition. It is transformation of object shapes from one form to another.

When object is described using polygon, the two keyframes for which in-between frames are to be generated compared. The keyframes are compared in terms of number of vertices, number of edges and number of line segments. If they are unequal they are added or deleted to match the count as preprocessing steps. This is illustrated in the Fig.10.6.1. The Fig. 10.6.1 shows two keyframes, K and K+1. The frame K has one line segment while frame K+1 has two line segments. Since keyframe K to balance the number of vertices in two frames as a preprocessing step. Then using linear interpolation we can translate the added vertex gives us the in-between frames as shown in the Fig. 10.6.1.

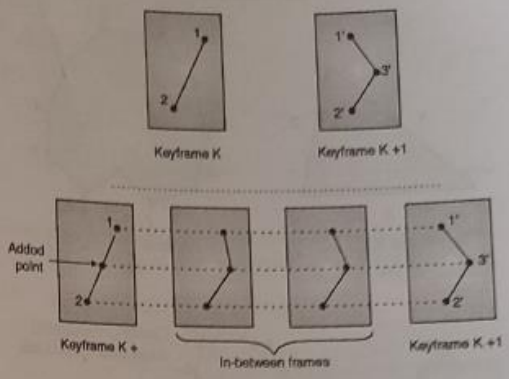


Fig. 10.6.1 : Generation of in-between frames using linear interpolation

The Fig. 10.6.2 shows transformation of triangle into quadrilateral using linear interpolation.

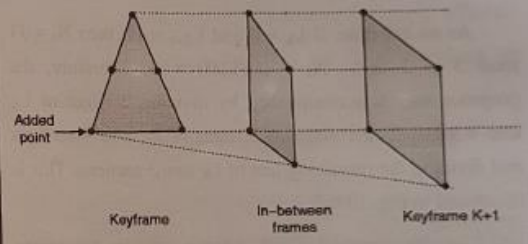


Fig. 10.6.2 : Linear interpolation for transforming a triangle into a quadrilateral

In general, to equalize two frames we can define

$$L_{max} = \max(L_K, L_{K+1}), L_{min} = \min(L_K, L_{K+1})$$

and

$$N_e = L_{max} \bmod L_{min}$$

$$N_s = \text{int}\,(L_{max}/L_{min})$$

Where L represents line segment.

With these definitions, the preprocessing is accomplished by

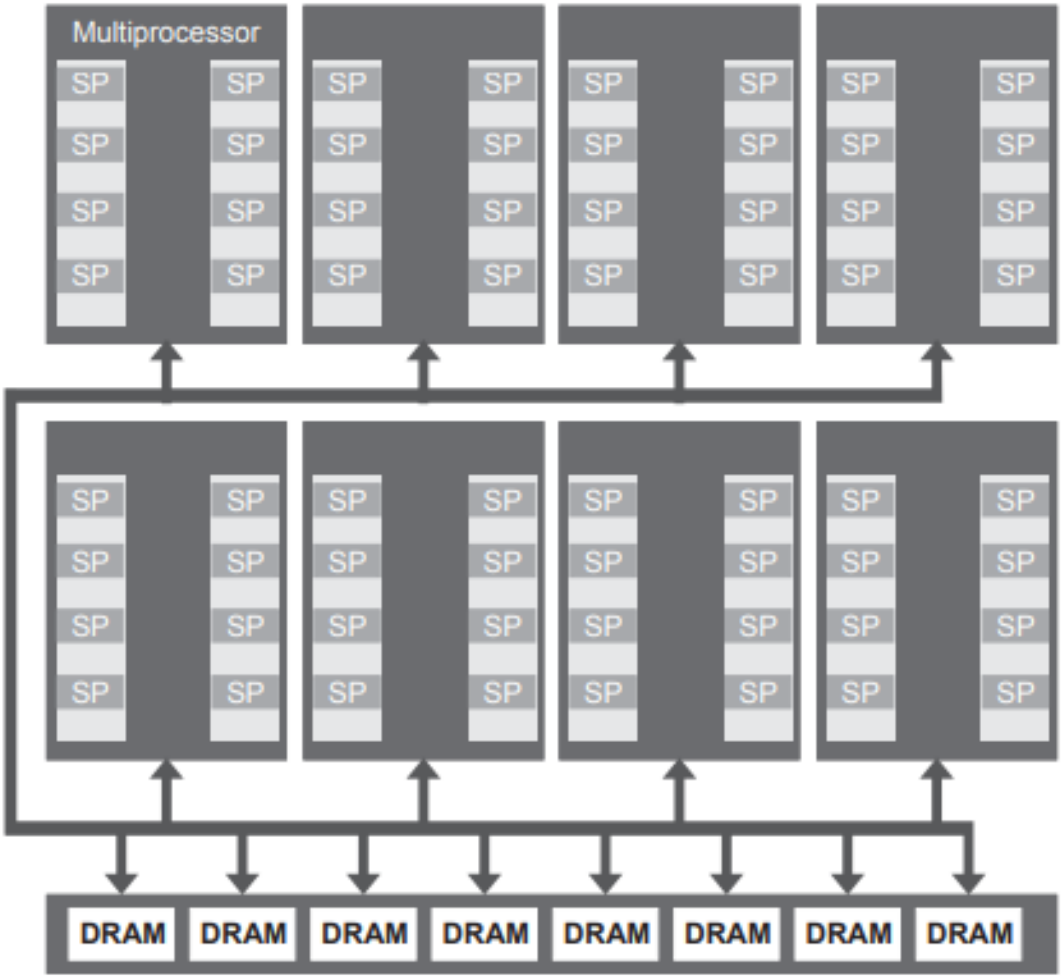
1. Dividing N_e edges of keyframe$_{min}$ into N_{s+1} sections.
2. Dividing the remaining lines of keyframe$_{min}$ into N_s sections.

☞ Applications of morphing

UQ. 10.6.7 What are the applications of morphing ? **SPPU - Dec. 12, May 15**

1. In medical applications digital image morphing is used to know how a particular organ will look like after a surgery.
2. It is heavily used in TV and movies for special effects.
3. It is commonly recognized for Face morphing in advertising world.

3) Draw block diagram of NVIDIA workstation and explain it in brief



NVIDIA workstations are based on GPU's with CUDA support. NVIDIA has mainly Tesla workstation, Quadro Workstation, Maximus workstations based on GPU and CUDA. (Source: NVIDIA)

10.12.6.1 Tesla Workstation

The Tesla workstation is a desktop computer that is backed by NVIDIA and built by Dell, Lenovo and other companies.

It is meant to be a demonstration of the capabilities of NVIDIA's Tesla GPGPU brand; it utilizes NVIDIA's CUDA parallel computing architecture and is powered by up to 2688 parallel processing cores per GPGPU,[1] which allow it to achieve speeds up to 250 times faster than standard PCs, according to NVIDIA.

NVIDIA Tesla GPU Accelerators turn standard PCs and workstations into personal supercomputers. Tesla workstation products are supported under

- Windows Vista, and Windows 7 - 32-bit (C2075 only) and 64-bit
- Linux 32-bit and 64-bit

4) Write a short note on motion specification method based on.

i) Geometric and kinematics information.

ii) Animation languages

10.11.1 Method Based on Geometric Information

This is the straight forward method for motion control. In this method there is direct specification at the motion parameters. In this we explicitly give the rotation angles and translation vectors. Then the geometric transform matrices are applied to transform coordinate positions. The alternative way is an approximating equation to specify certain kinds of motions. We can approximate the path of a bouncing ball with sine curve.

$$Y(X) = P\,|\sin(WX + \theta_0)|\, e^{-mx}$$

where P is the initial amplitude, W is the angular frequency, θ_0 is the phase angle, m is the damping constant.

10.11.2 Method based on Kinetic Information

The animation sequence can also be constructed using kinematic description. With this description, we have to specify the animation by giving motion parameters that is position, velocity, and acceleration without reference to the forces that cause the motion. The motion of rigid bodies in a scene try giving an initial position and velocity vector for each object will be designed. If the velocity is specified by any vector, that vector gives the direction for the motion path and the speed. If acceleration is also specified then we can generate speed ups, slow downs, and curved motion paths. Kinematic specification of a motion can also be gives by describing the motion path. This is done using spline curves.

The alternative to this is to use inverse kinematic. In this case the initial and final position of objects at specified times and the motion parameters are computed by system. For example, in zero acceleration, we can determine constant velocity that will accomplish the movement at an object from initial to final position.

Transition animations

- Use to show relationships between states. Animating state changes makes them easier to understand and appear smoother.
- Make sure transitions have natural mappings. For example, an opening window transition should be upward and expand; a closing window transition should be downward and contract.
- Must complete within a half second or less.

Feedback animations

- Must have clearly identifiable completion and failure states.
- Must stop showing progress when the underlying process isn't making progress.

10.9.2 Don'ts

- Don't use animations that can noticeably affect performance. Consider performance over slow network connections or when many objects are involved.
- Don't draw attention to things that aren't worthy of attention.

For critical animations there are many more guidelines. Many times we have to give priority to visualization than to strict rule. Use of color is often very important and must be chosen properly. The timing of animations is often driven by computing time instead of by final appearance. Allot as much time as possible to aesthetic considerations in the generation of animation. Overall feeling in animation is very important aspect.

Syllabus Topic : Animation Languages

10.10 ANIMATION LANGUAGES

UQ. 10.10.1 *Give different types of animation languages.* SPPU - Dec. 11

UQ. 10.10.2 *Write a short note on Animation Languages.* SPPU - Dec. 12

For describing animation there are many different languages. These languages are broadly categorized into three parts :

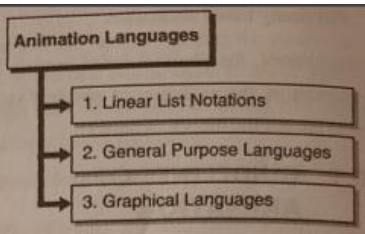


Fig. C10.4 : Animation Languages

10.10.1 Linear List Notations

For animation, each event in the animation is described by a starting and ending frame number and an action that is to take place. A typical statement will be

30, 50 A ROTATE "TREE", 2 by 45

Means "between frames 30 to 50, rotate the object called TREE about axis 1 by 45 degrees, determining the amount of rotation at each frame from table A". Since the statements describe individual actions and have frame values associated with them, for the most part their order is, irrelevant. However the order may matter if the two actions are applied to the same object at the same time.

10.10.2 General Purpose Languages

In general purpose programming languages we can embed animation features. The values of variables in the language can be used as parameters to whatever routines actually generate animations, so the high level languages can actually be used to generate simulations that then generate animations as a side effect. But the main issue here is for this programming expertise is needed. ASAS is an example of such a language. It is built on top of LISP. A point of view consists of a location and an orientation for an object.

ASAS also includes wide range of geometric transformations that operate on objects. They take an object as an argument and return a value that is transformed copy of the object. These transformations include up, down, left, right, zoom-in, zoom-out, forward and backward.

10.10.3 Graphical Languages

In case of textual languages used for animation, it is difficult for an animator to see what will take place in an animation just by looking at the script. Whereas graphical animation languages describe animation in a more visual way. These languages are used for expressing, editing, and comprehending the simultaneous changes taking place in an animation. Substitution of visual paradigm is the important point for such languages, rather than writing descriptions for actions.

5) Write any three important features of NVIDIA gaming platform

14.2.1.3 Features of NVIDIA Gaming Platform

1. Highest level performance and the smoothest experience possible from the moment you start playing.
2. Enables developers to add amazing graphics effects.
3. Dedicated ray tracing hardware enables fast real-time ray tracing with physical accurate shadows, reflection, refractions and global elimination.
4. Provides variable rate shading and faster frame rates.

6) Explain renaming of a segment with suitable example

10.3.3 Segment Renaming

UQ. 10.3.7 Explain with illustration how segments are renamed.

SPPU - May 10, May 11

UQ. 10.3.8 Write an algorithm to rename a segment. Draw a sample segment table.

SPPU - May 19

This operation is easy for implementation. Generally when we are using motion pictures or animated pictures, we are first displaying an object, then deleting that object, changing its position and again displaying the object. Consider again the same example of bird, sun, hill and tree which is shown in Fig. 10.1.1.

Suppose we want to move bird then we are creating a bird segment, displaying it, then deleting it, changing slightly its position and again creating a new segment for a bird and displaying it. The problem with this is during the time after first image is deleted and next image is complete, only partially completed image may get displayed. So to avoid this problem we should not delete a segment till the replacement for it is completed. This means that both the segments must be present in display file at the same time.

We will build a new invisible image under some temporary name. When it is completed we can delete the original image and make replacement image visible. The idea of maintaining two images, one to show and one to modify is called as *double buffering*.

Suppose we are maintaining four segments and if we want to rename the second segment by new segment then we are creating segment five, as a temporary segment with its visibility as OFF.

So that it will not get displayed. Once the segment five is completed then we are deleting segment two and copying all the attributes of the segment two to new segment five; as if at the same location of segment two we are writing commands of segment five.

Display file status

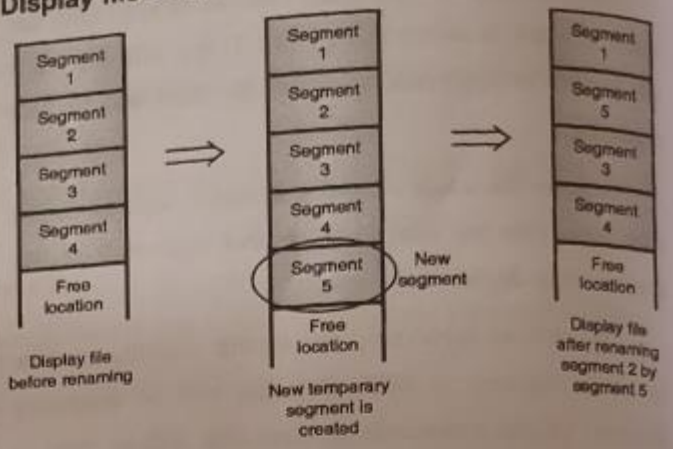


Fig. 10.3.3

After deleting segment two, we have to make the size of segment two as zero to indicate that it is deleted. In Fig. 10.3.3 both display file and segment table's status is shown.

Table 10.3.2

Segment table before renaming

Segment Name	Segment Start	Segment Size	Visibility	....
1	1	4	ON	
2	5	7	ON	
3	15	5	ON	
4	20	5	ON	

Segment table having both segments two and five

Segment Name	Segment Start	Segment Size	Visibility	....
1	1	4	ON	
2	5	7	ON	
3	15	5	ON	
4	20	5	ON	
5	–	–	OFF	

Segment table after renaming - Segment two as segment five

Segment Name	Segment Start	Segment Size	Visibility	....
1	1	4	ON	
2	5	[0]	ON/OFF	
3	15	5	ON	
4	20	5	ON	
5	[5]	[10]	[ON]	

Remember that renaming means giving

7) Explain architecture of i860

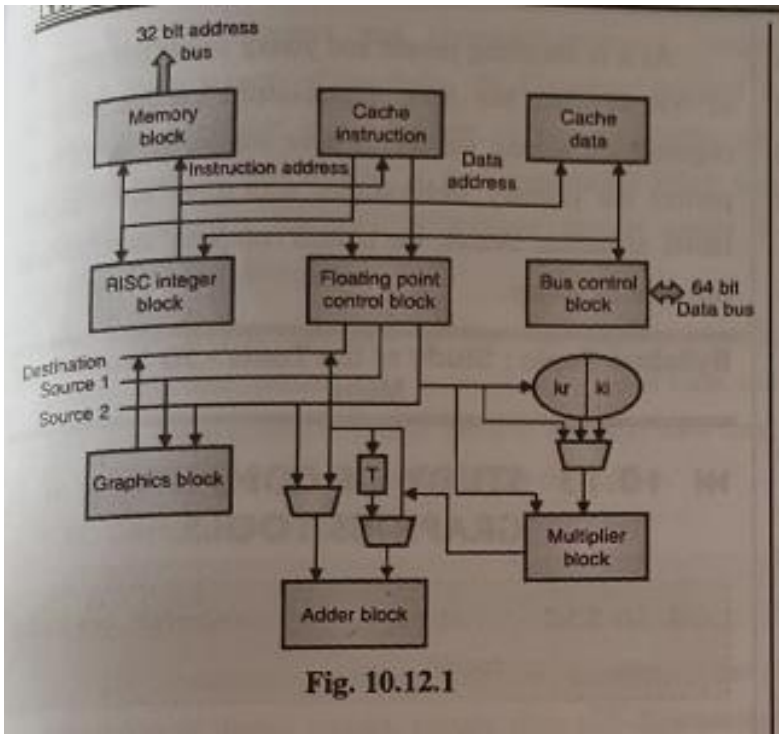


Fig. 10.12.1

The Intel i860 is also known as 80860. The i860 was a RISC microprocessor from Intel, first released in 1989. i860 is a 32-bit microprocessor with 64-bit external data bus. The i860 was one of Intel's first attempts at an entirely new, high-end instruction set. Diagram is copyrighted by Intel. Refer the following modified diagram. (Source: Intel).

Some of the technical features of i860 are as follows :

- The i860 combined a number of features that were unique at the time, most notably its very long instruction word (VLIW) architecture and powerful support for high-speed floating point operations.
- The design mounted a 32-bit ALU "Core" along with a 64-bit FPU that was itself built in three parts: an adder, a multiplier, and a graphics processor.
- The system had separate pipelines for the ALU, floating point adder and multiplier, and could hand off up to three operations per clock.
- The CPU could execute the majority of floating-point instructions either in pipelined mode or in scalar mode.
- In scalar mode the instructions executed one after another, which took from 3 to 4 clock cycles per instruction.
- In pipelined mode instruction execution was broken into three or four stages, and the CPU could execute different stages of different instructions at the same time. As a result one floating-point unit could provide new result every clock cycle.
- Two floating point execution units (FP adder and FP multiplier) could work in parallel, that is theoretically the CPU could provide two results every cycle.

- The processor could load and execute two instructions at the same time - one integer instruction and one floating point instruction. Also, certain floating-point instructions performed more than one operation, using both FP execution units simultaneously.
- All of the buses were at least 64 bits wide. The internal memory bus to the cache, for instance, was 128 bits wide. Both units had thirty-two 32-bit registers, but the FPU used its set as sixteen 64-bit registers. Instructions for the ALU were fetched two at a time to use the full external bus.
- Intel i860 instructions acted on data sizes from 8-bit through 128-bit.
- The graphics unit was unique for the era. It was essentially a 64-bit integer unit using the FPU registers as eight 128-bit registers.
- It supported a number of commands for SIMD-like instructions in addition to basic 64-bit integer math.
- Experience with the i860 influenced the MMX functionality later added to Intel's Pentium processors.
- One unusual feature of the i860 was that the pipelines into the functional units were program-accessible, requiring the compilers to order instructions carefully in the object code to keep the pipelines filled.
- In traditional architectures these duties were handled at runtime by a scheduler on the CPU itself, but the complexity of these systems limited their application in early RISC designs.
- The i860 was an attempt to avoid this entirely by moving this duty off-chip into the compiler. This allowed the i860 to devote more room to functional units, improving performance.
- As a result of its architecture, the i860 could run certain graphics and floating point algorithms with exceptionally high speed.

Overall processor performance was highly dependent on how well the code is optimized.

As a result the i860 was much more popular in areas where the code doesn't change often or doesn't change at all as an embedded or graphics processor, and had very limited use as a CPU. Normally it is used in parallel computer systems.

8) Write a short note on motion specification methods based on :

i) Geometric and kinematics information. ii) Specification methods based on physical information

10.11.3 Method based on Physical Information

The description of object behavior under the influence of forces is referred to as a physically based modeling. The dynamic are examples of physically based animations. Dynamic description require the specification of the forces that produce the velocities and acceleration object motions are obtained from the forces equations describing physical laws. The general form of Newton's second law for a particle of mass m is

$$V_f = \frac{d}{dt}(mv)$$

Where V_f is the force vector and V is the velocity vector, if mass is constant, then

$V_f = m A$ Where A is acceleration vector.

Application of physically based modeling are complex rigid body system and non rigid systems as cloth, plasticity.

8) Compare Conventional and Computer based Animation

Sr. No.	Conventional Animation	Computer based Animation
1	Conventional animation uses methods that don't involve any kind of digital tools.	Computer Based animation uses digital tools for animation.
2	Conventional animation uses physical materials and activities	It uses virtual materials in a digital space.
3	Conventional animation generally involves hand-drawing, hand-inking, and hand-painting each frame on physical paper.	Computer animation can be either 2D or 3D.
4	It is more time consuming and needs team of special artists.	Less time is required as compared to conventional animation.
5	Accuracy is compromised in conventional animation.	More accurate as compared to conventional animation.