

Computer Vision and Image Processing -Fundamentals and Application
Professor Doctor M.K. Bhuyan
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati
Lecture 38
Gesture Recognition

Welcome to NPTEL MOOCs course in Computer Vision and Image Processing Fundamentals and Applications. In this class I will discuss one important application of computer vision, that is gesture recognition. Gesture recognition has many important applications. One important application is human computer interactions. So, I will discuss all these applications and also what are the major challenges of gesture recognition, that I am going to explain.

And mainly I will discuss the concept of hand gesture recognition. And in this discussion, I will discuss two important algorithms. So, briefly I will highlight the concept of DTW, that is Dynamic Time Warping. And also, the concept of the hidden Markov model. So, briefly these two concepts I will be explaining, related to gesture recognition. So, let us see what is gesture recognition in my next slide.

(Refer Slide Time: 01:32)

What is Gesture?

A movement of a limb or the body
as an expression of thought or
feeling.

--Oxford Concise Dictionary 1995

So, this is the definition of gesture, a movement of a limb or the body as an expression of thought or feeling. So, this is the definition from the Oxford concise dictionary, that is the definition of gesture.

(Refer Slide Time: 01:47)

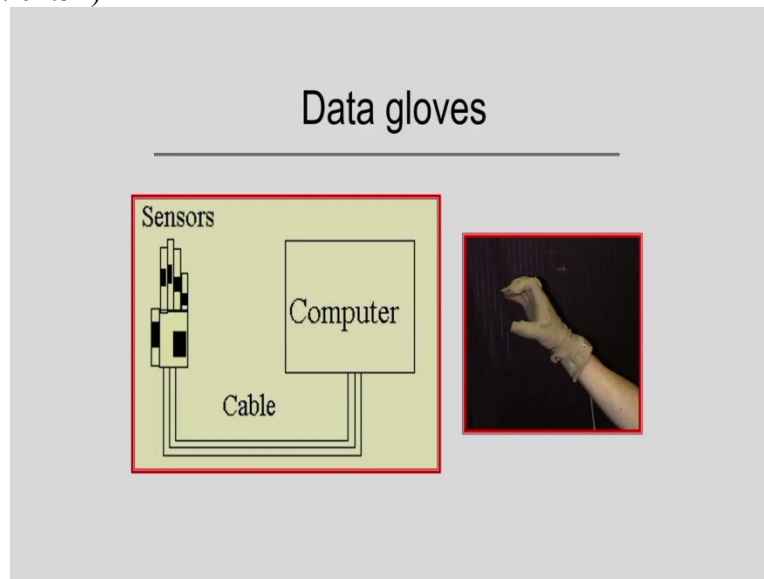
Human Computer Interface using Gesture

- Replace mouse and keyboard
- Pointing gestures
- Navigate in a virtual environment
- No physical contact with computer
- Communicate at a distance

So, there are many applications of gestures. So, if I consider only hand gestures, so hand gestures also have many applications, one is the replace mouse and the keyboard. That means, without using the mouse and a keyboard, I can interact with the computer, with the help of hand gestures. Also, I may consider pointing gestures, maybe for robotic interactions, the human robot interactions or maybe for window menu activations.

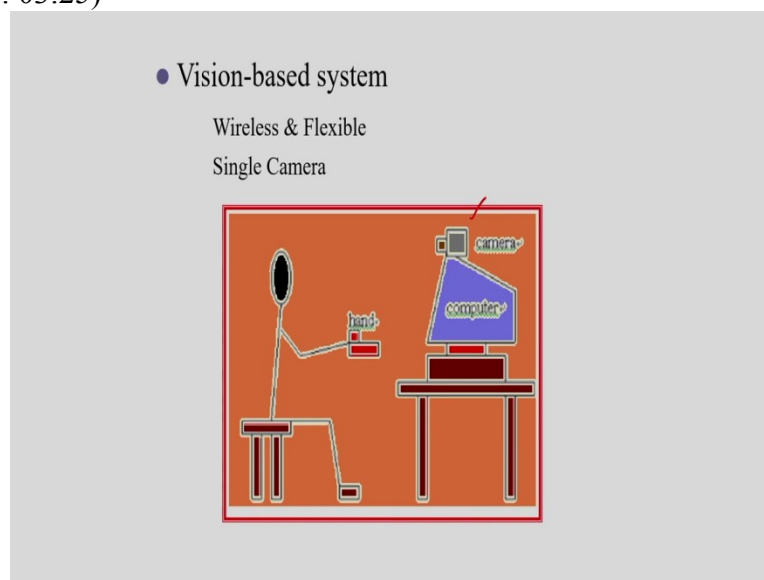
So, I can select the window menus based on the pointing gestures. And like navigate in a virtual environment that is the VR application, virtual reality application. And no physical contact with computer, because I can interact with the computer by using hand gestures. And communicate at a distance. So, these are the, some human computer interface using gestures, by hand gestures.

(Refer Slide Time: 02:51)



So, in this case, I am showing one interaction system that is by considering the data gloves. So, here you can see, I have the data gloves, in the data gloves I have optical or the magnetic sensors. So, these sensors detect the movement of the fingers, the movement of the hand and the glove is connected to the computer. So, computer will get the signals corresponding to the movement of the fingers, the movement of the hand and based on this I can recognize different types of gestures, this is the glove-based system.

(Refer Slide Time: 03:25)

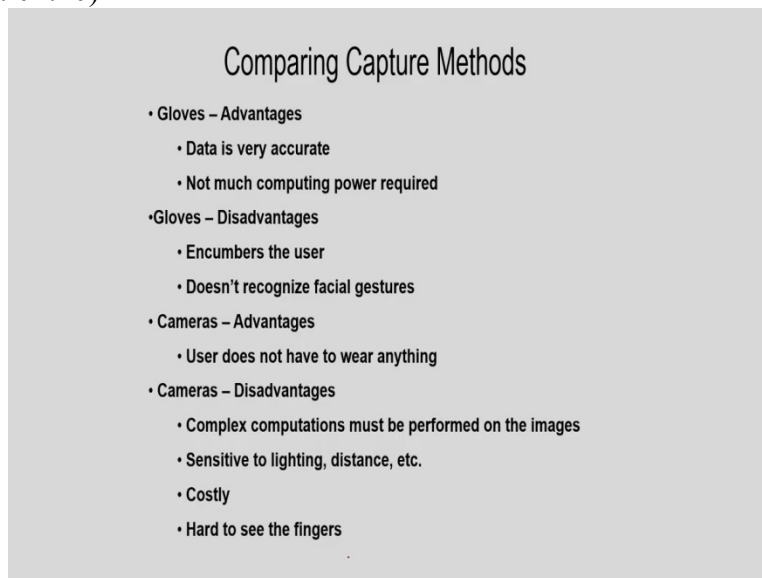


The next system is the vision based system. So, here you can see, I have a camera. So, here in the figure you can see I have a camera. So, the camera detects the movement of the hand, the

movements of the fingers and after this, the computer can recognize the movement of the hands or the gestures or it can recognize the finger's feeling.

So, this is a pattern classification problem, because I can recognize different gestures, different finger's feeling, and in this case, I may have one camera or maybe two camera or multiple cameras. So, in this case, glove is not important. So, the camera will detect the movement of the hand and also it can detect the movement of the fingers.

(Refer Slide Time: 04:10)



Comparing Capture Methods

- **Gloves – Advantages**
 - Data is very accurate
 - Not much computing power required
- **Gloves – Disadvantages**
 - Encumbers the user
 - Doesn't recognize facial gestures
- **Cameras – Advantages**
 - User does not have to wear anything
- **Cameras – Disadvantages**
 - Complex computations must be performed on the images
 - Sensitive to lighting, distance, etc.
 - Costly
 - Hard to see the fingers

So, if I compare these two systems, so one is the glove-based system, another one is the camera-based system, that is the vision-based system and the glove-based system. You can see advantages and disadvantages of, if I want to compare these two systems, one is the glove-based system another one is the vision-based system. They both have advantages and disadvantages. So, if you see the glove-based system, the advantages are, data is very accurate.

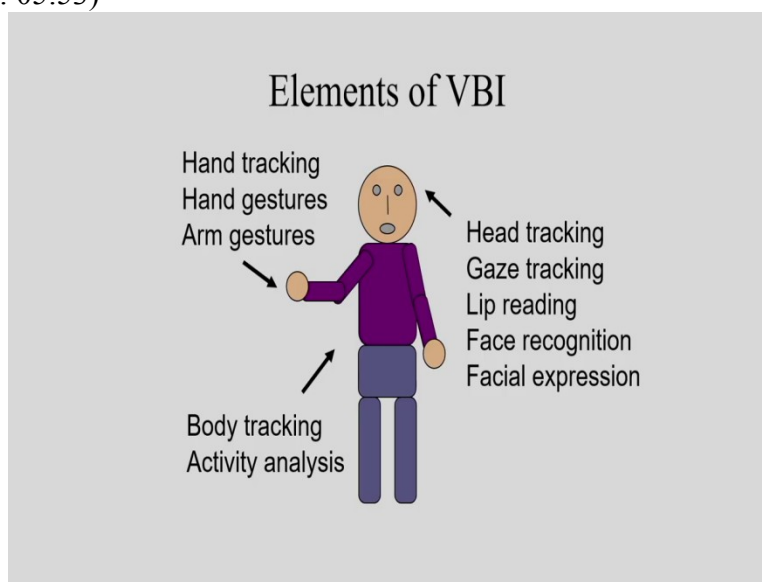
Because I am getting the signal from the glove. And mainly we have the optical sensors and the magnetic sensors. So, that means I am getting a very accurate signal from the data gloves and not much computing power required, in case of the glove-based system. But the disadvantages are, because user has to wear gloves. That is the disadvantage, and this glove-based system is not available for recognizing facial gestures.

So, these are the advantages and the disadvantages of the glove-based system. And if I consider the vision-based system, one main advantage is that user does not have to wear anything, that is

one important advantage and disadvantages like we have to do complex computations, because we have to do the image processing. And since, we have to consider different lighting conditions, the cluttered background.

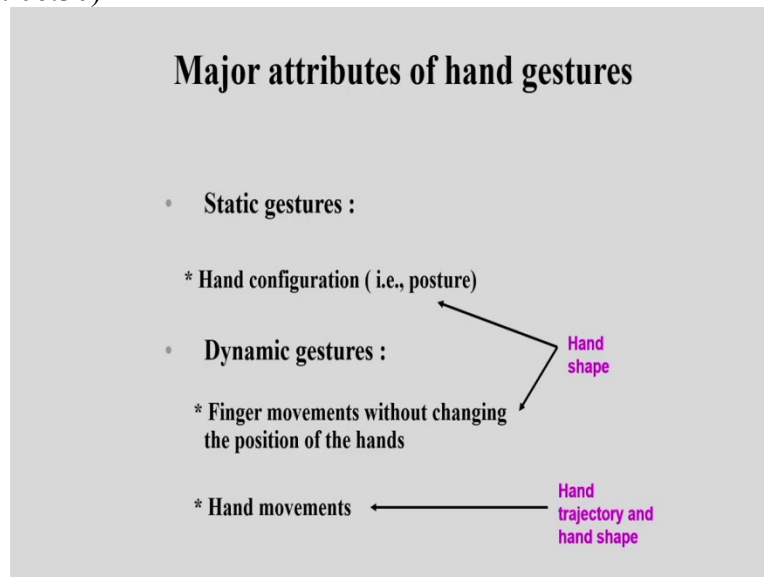
So, that means, we have to do lots of image processing and that is very difficult and sometimes there may be a problem of occlusions. So, hard to see the fingers by the camera, and this is nothing but the occlusion. So, these are the comparisons between the glove-based system and the camera-based system.

(Refer Slide Time: 05:53)



So, I want to show the elements of the vision-based interface. VBI means the vision-based interface. So, these are the research area in which, you can do research in computer vision. So, like hand tracking, hand gestures, arm gestures, head tracking, gaze tracking, lip reading, face recognition, facial expression recognition, body tracking, activity analysis. So, these are the elements of vision-based interface. So, any one of these topic you can do research in computer vision or maybe in machine learning.

(Refer Slide Time: 06:30)



Now, I will consider the major attributes of hand gestures. So, I may consider static gestures or dynamic gestures. For static gestures, hand configuration is important that is the posture and for this I need the information of hand shape. In case of dynamic gestures, finger movements, without changing the position of the hands. Then in this case I can consider the hand shape information, but if I consider hand movements in space, then I have to consider the hand trajectory and the hand shape.

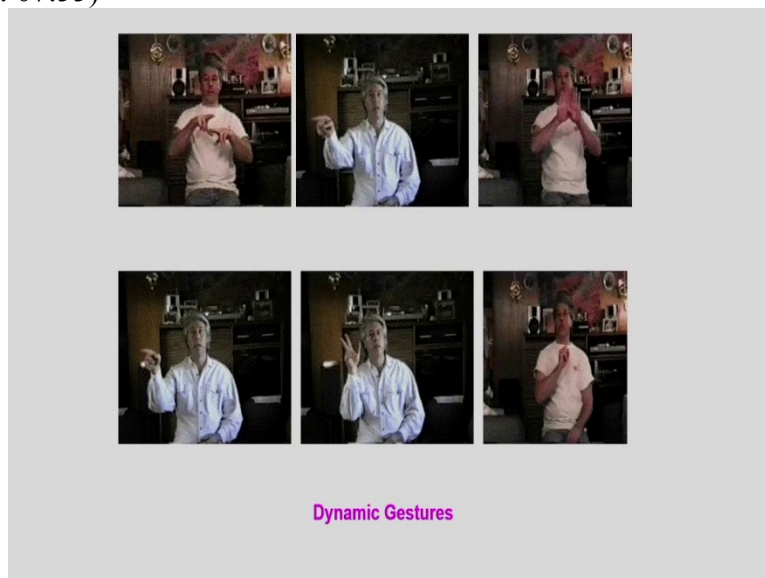
That means, in case of the static gesture, only we have to consider the hand shape information, but in case of a dynamic gestures, it may have local motions or the global motions or maybe the combination of the local motion and the global motion. In case of local motion only I have the motion of the fingers, in case of the global motion I have the movement of the hand and generally if I consider the sign language suppose, then in this case it is a combination of both local motion and the global motions. So, these are the major attributes of hand gestures.

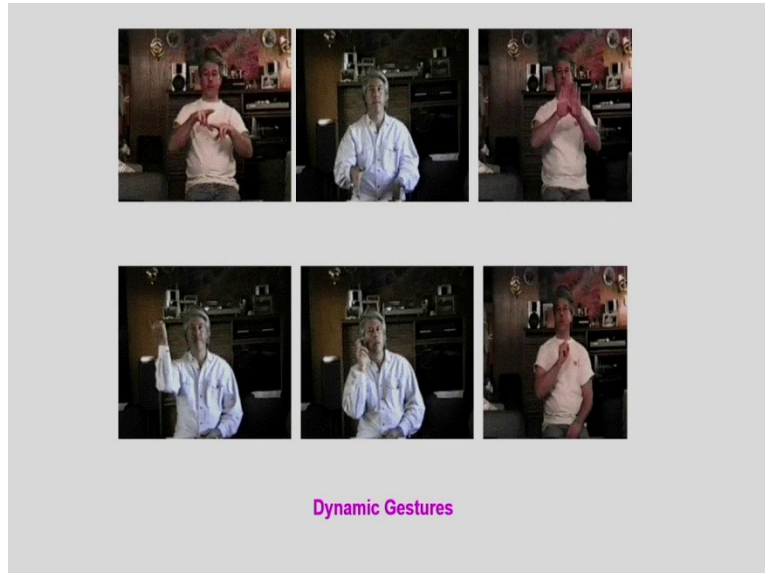
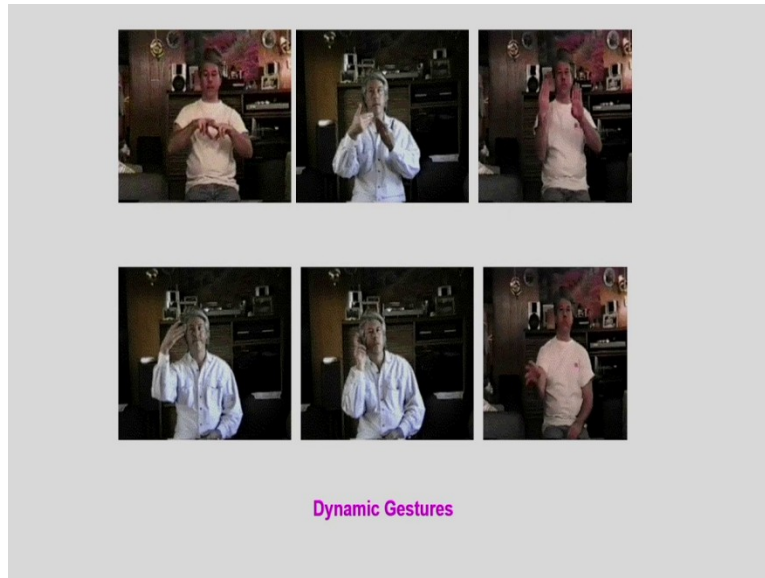
(Refer Slide Time: 07:40)



So, here you can see I am showing some static gestures of American Sign Language. So, in this case, I need the information of the hand shape. So, with this information, that means the shape information, I can recognize different static gestures.

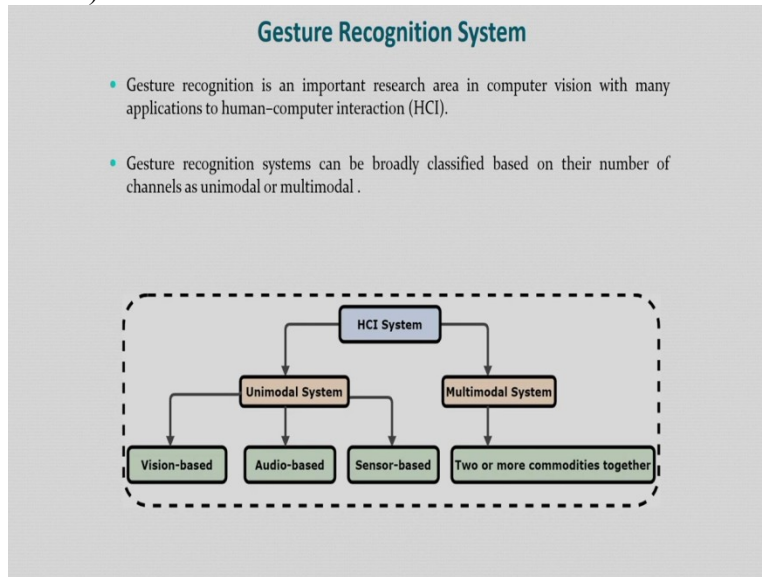
(Refer Slide Time: 07:55)





And here you can see I am showing some of the dynamic gestures. So, here you can see the moment of the hand and also the finger movements. So, that means, in this case I have both, local motion and the global motion. So, this is one example of dynamic gestures. So, in this video I have shown that sign language, you can see some American Sign Language and in this case the computer has to understand these signs of the American Sign Language.

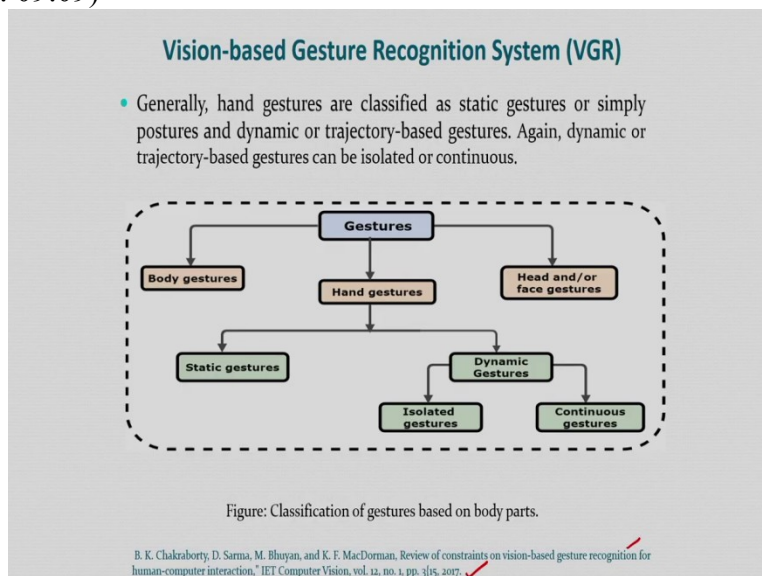
(Refer Slide Time: 08:28)



For this gesture recognition system, that is the human computer interface system, it may be a unimodal system or a multimodal system. So, for this we can consider the vision-based system, that means the camera or maybe the audio-based system or maybe the sensor based system like the data gloves.

In case of the multimodal system, we can consider one or two modalities. That means, I can combine the vision based and the audio based or maybe the audio based plus sensor based I can combine. So, in case of the human computer interface system, I may have unimodal system or the multimodal system.

(Refer Slide Time: 09:09)



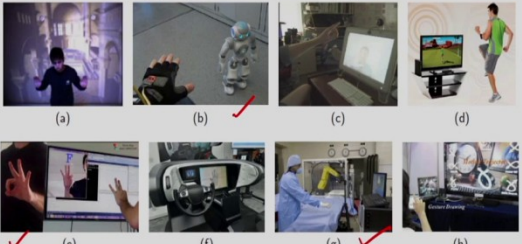
And if you consider the gestures, so, there are many types of gestures. One is the body gestures, one is the hands gestures, one is the head and or the face gestures. So, in my discussion I am mainly considering the hand gestures. As explained earlier I may have static gestures or the dynamic gestures and I can consider the isolated gestures or the continuous gestures. So, in case of the continuous gesture I can perform all the research continuously, that is like the fluent finger spinning I can do, or the, the continuous gesture I can perform.

But in case of the isolated gestures, I can perform a particular gesture at a particular time. So, to understand these concepts, mainly the constraint on vision-based gesture recognition for human computer interaction, you may see this research paper, this is our research paper. So, I think you can see this research paper, you can understand the concept of the vision-based gesture recognition for human computer interaction.

(Refer Slide Time: 10:16)

Application of VGR System

- In recent years, VGR has become a key research topic in HCI with its diverse application in different areas.
- Sign language recognition ✓
- Robotic control and healthcare ✓
- Human-Computer Interaction (HCI) and
- Augmented Reality (AR) and Virtual Reality (VR) etc.



Applications of gesture recognition system.

So, there are many applications of the vision-based gesture recognition system, VGR means the vision-based gesture recognition system. So, one important application is sign language recognition. Another one is robotic interactions. And also, for healthcare, there are many applications like laparoscopic surgery, by using gestures, or maybe the window menu activation by considering the hand gestures.

One is a human computer interactions, the human computer intelligent interactions and some applications in augmented reality and virtual reality. So, here you can see the sign language recognition, here this is the sign language recognition and this application is in virtual reality and

this is in the healthcare applications. This is the robotic interactions. So, there are many applications of VGR, that is the vision-based gesture recognition.

(Refer Slide Time: 11:20)

Major challenges of gesture recognition

- Segmentation ✓
- Occlusion. ✓
- Depth information. ✓
- Illumination change
- 3D translation and rotational variance.
- Temporal variation.
- 3D shape variation.
- No explicit indication of starting and ending points.
- Co-articulation.

Major challenges of gesture recognition

- Segmentation ✓
- Occlusion. ✓
- Depth information. ✓
- Illumination change
- 3D translation and rotational variance.
- Temporal variation. |
- 3D shape variation. |
- No explicit indication of starting and ending points.
- Co-articulation. ✓

But the major challenges of gesture recognition, I am going to explain now, the in case of the vision-based system, we will consider only the camera. Glove based system has the advantages, but the vision-based system has the flexibility because users need not wear the data gloves. So, the camera will detects the movement of the hand or the movement of the fingers. But there are many challenges of the vision-based gesture recognition.

So, one is the segmentation of the hand from the background. So, in this case, we have to consider different illumination conditions, random illumination conditions and also, we have to consider the cluttered background. So, segmentation of the hand from the background, the cluttered background and under different illumination conditions is one research problem. And also, the second problem is the occlusion.

So, there may be self occlusion between the fingers, because the camera cannot see all the points of the hand or all the position of the fingers. So, we have to consider the occlusion, the self occlusion is one important aspect. So, that we have to consider. And sometimes the depth information is also very important. So, if I consider only one camera, it is not possible to determine the depth information.

But if I consider maybe the stereo vision system, then in this case we have to estimate the depth, with the depth information that recognition performance increases. So, how to determine that depth information that is also one important salience of gesture recognition. And already I have mentioned about the illumination changes, because for different illumination, I have to do the segmentation. The segmentation of the hand and also the tracking of the hand and the background maybe cluttered background.

And also we have to consider the 3D translation and the rotational variance, that means we have to consider the translation variance and the rotational variance. So, we have to extract the features, which should be invariant to translation and the rotation. That means we have to extract the hand features, which are invariant to a (T, R, S) . That is the translation rotation in the scaling.

So, another problem is the spatial temporal variation. That means if a, if a particular gesture is performed by different persons or the different users, there will be spatial temporal variations. Even the same gesture is performed by the same user or the same person, there will be spatial temporal variation. So, this spatial temporal variation we have to consider. And also, if I considered a continuous gesture, then in this case, we have to find the starting point and the ending point of the gesture.

So, how to do the segmentation, if I do the gestures continuously, that is the continuous gesture. So, how to do the segmentation? That is one important point, that is the segmentation of the

continuous gesture. And one important point is the co articulation, that means the current gesture is affected by the preceding and the following gesture. So, one gesture maybe the part of another gesture. So, this problem also we have to consider, that is the co articulation problem. So, these are the major challenges of gesture recognition.

(Refer Slide Time: 14:41)

Major challenges of DG recognition

- **Segmentation problem:** Accurate segmentation of hand or the gesturing body parts from the captured videos or images remains a challenge in computer vision.
 - illumination variation ✓
 - background complexity ✓
 - occlusion
- **Gesture spotting:** Gesture spotting is used to locate the starting point and the endpoint of a gesture.
 - segmentation ambiguity
 - spatio-temporal variability ✓
- **Difficulties associated with image processing techniques:**
 - extracted features should be rotation-scaling-translation (RST) invariant.
 - processing of a large amount of image data is time consuming and so real-time recognition may be difficult.

Now, again, I am explaining the major challenges of the dynamic gesture recognition. So, first problem already I have explained, that is the segmentation problem. So, for this, we have to consider the illumination variation and the complex background also we have to consider and also the problem of the occlusion. And in this case the self occlusion is very important, that means occlusion between the fingers, occlusion between the two hands suppose if I consider two hands, that occlusion also we have to consider.

Another one is the gesture spotting, that means we have to consider spatial temporal variations, if I consider a continuous gesture, we have to find a starting point and the ending point of this particular gesture, that is called the gesture spotting. And also, the difficulties associated with image processing techniques. So, the extracted feature should be rotation scaling and the translation invariant.

That means, the hand features should be invariant to rotations, translation and the scaling. And a processing of a large amount of image data is time consuming. And so, real time recognition is really very difficult. So, these are difficulties associated with image processing techniques.

(Refer Slide Time: 16:00)

Major challenges of DG recognition

- **Problems with continuous stream of gestures:** There are some non-gestural movements that generally occurs in a continuous stream of gestures.
 - “gesture co-articulation” that occurs between two gestures and the current gesture is affected by the preceding or the following gestures.
 - “movement epenthesis” is the unwanted movement that occurs while performing a gesture.
 - “sub-gesture problem” when a gesture is similar to a subpart of a longer gesture.
- **Problems related to two-handed gesture recognition:** two-handed gesture recognition faces some specific difficulties:
 - computational complexity ✓
 - occlusion due to inter-hand overlapping ✓
 - simultaneous tracking of both hands ✓

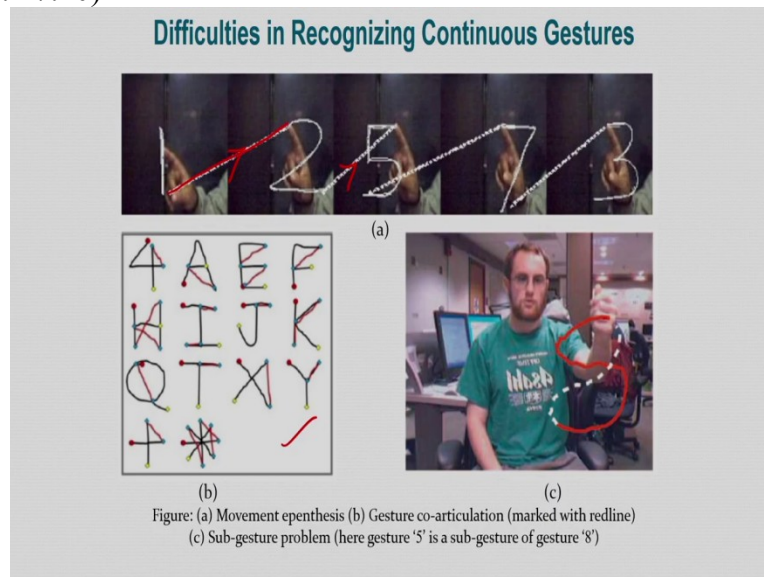
And if I consider the major challenges of the dynamic gesture recognition, if I consider the problem with continuous gestures. So, one problem is the co articulation problem that means, the current gesture is affected by the preceding or the following gestures. And another important problem is the movement epenthesis, that is the unwanted movement that occurs while performing a gesture.

And another problem is the sub gesture problem, when a gesture is similar to a sub part of a longer gesture. So, this problem I am explaining in my next slide, the one problem is the co articulation problem, one is the movement epenthesis problem and another one is the sub gesture problem. And one important problem is, that is the problems related to two handed gesture recognition.

It is very difficult, in case of a two-handed gesture recognition, one is the computational complexity and one is the occlusion due to inter hand overlapping. That already I have explained, that is if I considered the overlapping of two hands or maybe the overlapping of the fingers then, that is nothing but the occlusion, the self occlusion. And also, simultaneously we have to track both the hands.

So, we have to do the tracking and we have to develop the tracking algorithm, so, that it can track both hands. So, these are the major challenges of dynamic gesture recognition. Now, I will show in my next slide, what is the co articulation, what is movement epenthesis and what is sub gesture problem.

(Refer Slide Time: 17:40)

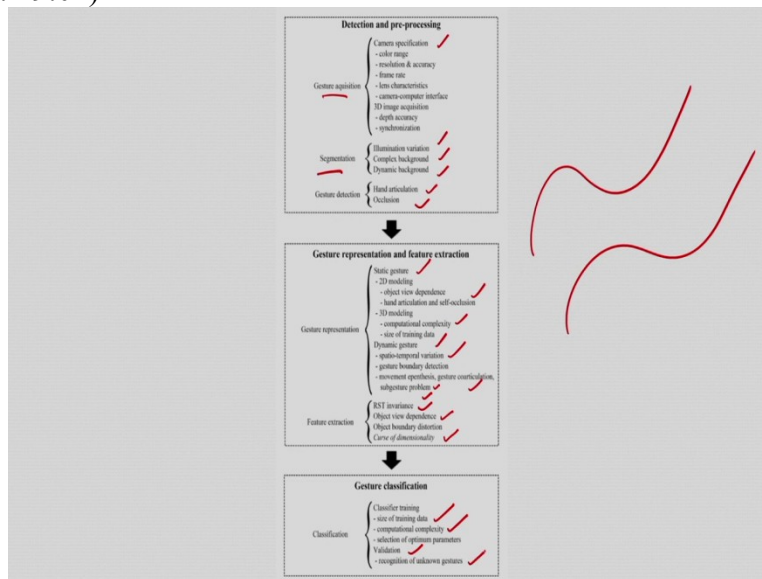


So, here you can see, I am showing the gestures one, two, three like this, 1, 2, 5, seven three like this. So, if I perform one and after this, if I perform two like this, so one is this extra movement between these two gestures. So, this is the extra movement between these two gestures. And similarly, from 2 to 5 this is the extra movements between these 2 and 5. So, this is called movement epenthesis.

In figure b I have shown the problem of the co articulation that means, the current gesture is affected by the preceding or the following gestures. So, here I have shown the example of the coarticulation and finally, I want to show the sub gesture problem in c, in figure c. So, my gesture is eight, but here you can see 5 is a sub gesture of gesture eight. So, if I want to draw eight, so, you can see the five is a part of 8.

So, that is called the sub gesture, but my gesture is the eight, but by mistake the recognizer can recognize that gesture as 5. So, this is called the sub gesture problem. So, these are the major problems in case of the continuous gesture recognition. One is the co articulation. One is the movement epenthesis and one is the sub gesture.

(Refer Slide Time: 19:01)



So, in summary, I am showing in this figure the, the problems of the gesture recognition. So, the major challenges of the gesture recognition. So, first one is gesture acquisition regarding the camera specification, color range, resolution, frame rate, lens characteristics and regarding the 3D image acquisition, depth accuracy and also the problem of the segmentation, like illumination variation, complex background, dynamic background.

And also, for the gesture detection, we can consider hand articulation and a, and an occlusion that is nothing but the self occlusion. After this gesture representation and the feature extraction. So, we may consider static gesture, dynamic gestures and we can consider the 2D modeling of the gestures or maybe 3D modeling. But in case of the 3D modeling, it is computationally complex.

And in case of the 2D modeling, it is object view dependence. And in this case, the problem of the self occlusion and the hand articulation. So, 3D modeling is more accurate, but it is computationally complex. And for dynamic gesture we have to consider, the spatial temporal variations. So, what is the spatial temporal variation? Suppose, if I want to perform the gesture like this, this is a gesture.

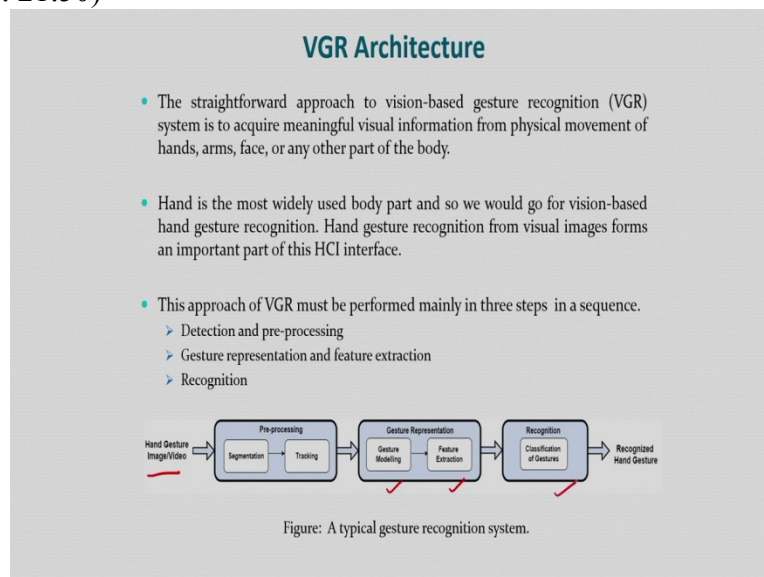
So, if I repeat this gesture, if I repeat this gesture suppose like this, there will be a spatial temporal variation. That means, variation in the space and variation in the time. The same gesture is repeated by different users, there will be spatial temporal variations. And even the

same gesture is performed by the same user, or the same person still there will be a spatial temporal variation.

And for the continuous gesture, we have to consider these cases, one is the movement epenthesis, co articulation and the sub gesture problem. And in case of the feature extraction, we have to consider the feature should be invariant to (())(21:09) that is RST invariance. And also, the object view dependence and also that we have to reduce the dimensionality of the feature vector, that is nothing but the curse of dimensionality.

After this we have to do the classification. So, who is classified as good? That we have to also consider. So, maybe the size of the training data, that is important. The computational complexity is very important and selection of optimum parameters for validation that is also very important. And the recognition of unknown gestures. So, these are the main challenges of gesture recognition.

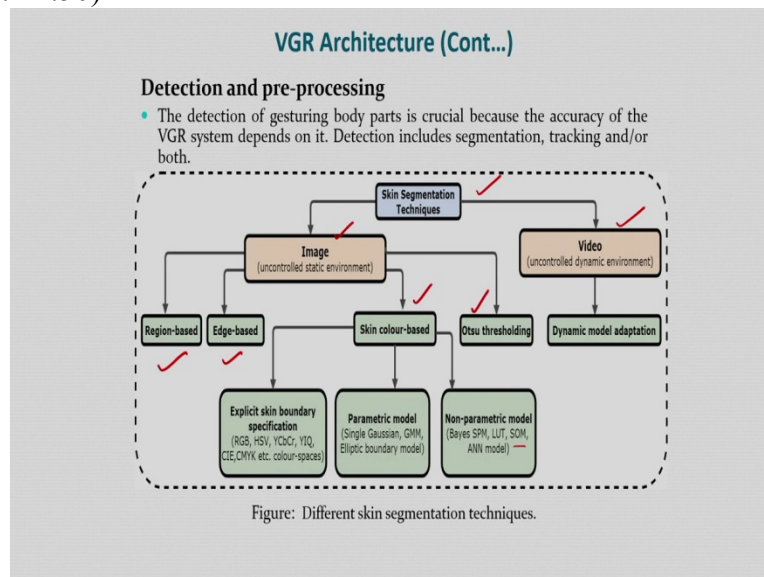
(Refer Slide Time: 21:50)



So, if you see the vision-based gesture recognition architecture, the first the hand and the image video that can be captured by the camera, after this we have to do the segmentation, the segmentation of the hand and after this we have to do the tracking. So, already I have explained this problem is quite difficult because we have to consider dynamic, background maybe the cluttered background and also illumination variations I have to consider.

And if I consider two handed gestures, I have to track both the hands. So, tracking is also a problem and segmentation is also a very difficult problem, because I have to consider elimination variation and the dynamic background and the cluttered background. After this I have to consider gesture representation. So, for this I have to do the modeling, gesture modeling, I have to extract features, the feature extraction. And finally, I have to select the classifier for recognition. So, this is a typical gesture recognition system.

(Refer Slide Time: 22:50)



After this we can consider the detection and the pre processing. Detection means the detection of the hand, that is the segmentation of the hand and also, I have to do the tracking. So, we may consider these cases. So, skin segmentation techniques we can apply for the image and the videos. And in case of the image, an uncontrolled static environment, because the background may be cluttered and there may be illumination radiations.

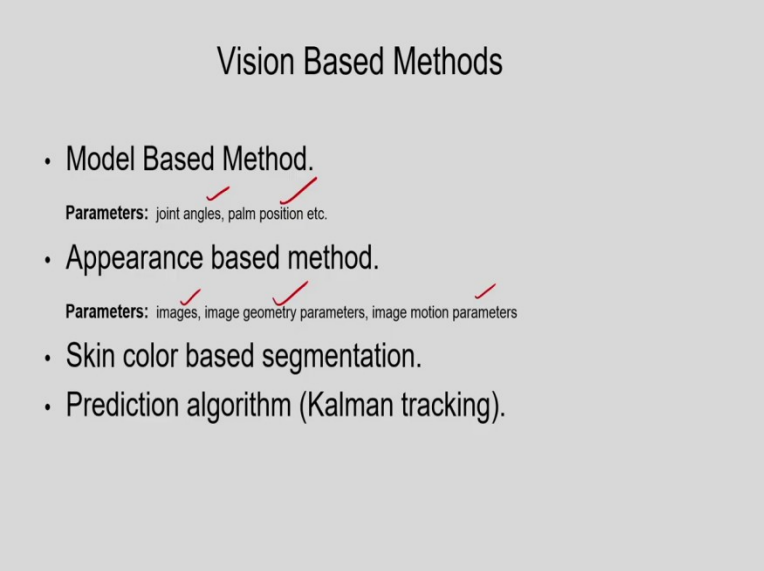
And in case of the video, we have to consider uncontrolled dynamic environment, the dynamic background may be there, the random illumination variation will be there, the cluttered background may be there. So, all these cases we have to consider and for image we may consider region-based technique, the edge-based technique or maybe the skin color based technique we can consider, for segmentation or maybe the otsu thresholding technique also we can use.

And in case of the video, we can consider dynamic model adaptation technique. So, for skin color-based techniques, we can consider different color models, maybe the RGB color model,

HSV color model, YCbCr color model, YIQ model like this all these models you can consider, to determine the skin colors and based on the skin color, we can segment out the hand from the background.

And for a parametric model we can consider a single Gaussian model, the Gaussian mixture model or maybe the elliptical boundary models we can consider. And that is corresponding to the skin color based segmentation or maybe the nonparametric models we can use like, the base skin probability map, SPM means the skin probability map we can use. The lookup table we can also use, the self organizing map SOM also we can use, or maybe the ANN model we can use for skin color segmentation. So, based on the skin color, we can segment out the hand from the background for different conditions.

(Refer Slide Time: 24:53)



Vision Based Methods

- Model Based Method.
Parameters: joint angles, palm position etc.
- Appearance based method.
Parameters: images, image geometry parameters, image motion parameters
- Skin color based segmentation.
- Prediction algorithm (Kalman tracking).

So, for vision-based methods, so we can consider the model-based methods. So, for this we can consider a parameter like, the parameters of the hand, like the joint angles, the palm positions like this we can consider or maybe we can consider the appearance-based model. So, for this we can consider the parameters like images, the image geometry parameters, images motion parameters, we can consider these parameters. And already I have explained about the skin color-based segmentation.

So, we can apply this technique. So, with the help of the skin color, we can do the segmentation of the hand from the background or maybe for a tracking, we can use the Kalman filter or maybe some filters like particle filters, we can use for the tracking. So, there are many tracking

algorithms, maybe the mean shift algorithm also we can used. But, one tracking algorithm is the Kalman tracking or maybe we can use the particle filter tracking, these techniques we can use for hand tracking.

(Refer Slide Time: 25:51)

VGR Architecture (Cont...)

Tracking:

- Tracking can also be considered as a part of pre-processing in the hand detection process as both tracking and segmentation together help to extract the hand from the background.

Various methods are:

- Using pixel-level change: ✓
 - background subtraction, ✓
 - inter-frame difference,
 - three-frame difference.
- Model-based: ✓
 - mean-shift or continuous adaptive mean-shift (CAMShift),
 - Kalman filter,
 - particle filter.

Some methods combine 2/3 techniques for tracking.

And for tracking already I have explained. So, the methods maybe like this. So, using the pixel level change, we can consider the background subtraction method, we can consider inter frame difference, the three-frame difference. So, these classical techniques also we can used. And maybe like the mean shift algorithm or the CAM shift algorithm, the Kalman filter, the particle filter. So, these techniques can be used for hand tracking.

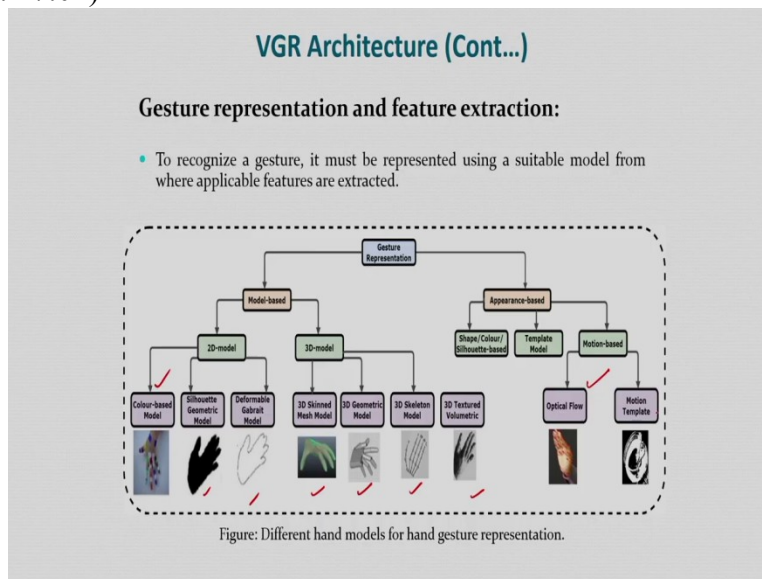
(Refer Slide Time: 26:21)

Hand Gesture Modeling

Representing the same hand posture by different hand models. (a) 3-D textured volumetric model; (b) 3-D wireframe volumetric model; (c) 3-D skeletal model; (d) Binary silhouette; (e) Contour model.

And for hand gesture modeling already I have explained. So, we can consider different types of models, like in the figure a, I have considered the 3D textured volumetric model. So, that is the first model, the 3D textured volumetric model we can consider. The second b, the 3D wireframe volumetric model, I can consider. In figure three, 3D skeletal model also we can consider. In figure d, this is the binary silhouette, we can consider. And maybe we can consider a control model as shown in figure e. So, these models we can use for hand gesture modeling.

(Refer Slide Time: 27:01)



So, here again I am showing these models. So, that is the gesture representation feature extraction. So, you can see the color models, the first I am showing that is the 2D model. The silhouette geometric model, that is the second one. That is a 2D model, deformable model also we can consider. And similarly, we can consider 3D models like 3D skin mesh model, 3D geometric model, 3D skeleton model, 3D textured volumetric model.

So, these are the model-based techniques and if I consider an appearance based model, that means, in this case we can consider shape, color silhouette based or maybe we can consider template models or maybe we can consider motion based models. So, for this we can consider optical flow, for motion representation or maybe we can consider motion templates, like motion history image, motion energy image. So, by using this motion templates, the motion history image and the motion energy image we can model a particular gesture or we can represent a particular gesture.

(Refer Slide Time: 28:14)

Depth-based Methods on RGB-D Data

- Depth information is largely invariant to illumination variation and skin colors and offers a quite clear segmentation from the background.
- So, the major problems in segmentation like illumination variation and occlusion can be handled nicely with the help of depth data to a great extent.
- **Kinect-based methods:** Kinect R obtains RGB-D data
- **Other depth sensor-based methods:** Leap motion controller (LMC) by Leap Motion, Senz3D by Creative and DVS128 by iniLabs and Intel RealSense are the most used RGB-D sensor for HCI applications apart from Microsoft Kinect.

And also, we should consider the depth base method on RGB depth data, because if I considered a depth information, the accuracy of recognition will increase. And it will also solve the problem of self occlusion or it may solve, or it may partially solve the problem of the self occlusion. So, the major problems in segmentation like illumination variation occlusion can be handled nicely with the help of the depth data.


And for this we can consider the Kinect based methods. So, we can consider the Kinect sensors to get the depth data or maybe other depth sensors, maybe the leap motion controller we can consider, senz3D 3D by creative depth, that also we can consider. So, different types of depth sensors are available. So, from this we can get the RGB depth data, the RGB D data. So, with the depth information, some of the problems can be partially or nicely eliminated like the problem of the self occlusion, the problem of the elimination variation. So, these problems can be ended, with the depth information.

(Refer Slide Time: 29:26)

VGR Architecture (Cont...)

Gesture representation and feature extraction:

- Feature extraction involves selection of proper features and their extraction for classification. The selection of features is purely application specific.
- Many authors have identified different features for representing particular kinds of gestures. Most of the features can be broadly classified as follows:
 - a) shape, (e.g., geometric features or non-geometric features) ✓
 - b) texture or pixel value, ✓
 - c) 2D/3D model-based features ✓
 - d) spatial features (e.g., position and motion velocity).
 - e) spatio-temporal interest points (STIP).



And after this for gesture representation and the feature extraction. So, we have to extract the features and the features should be invariant to (())(29:34). So, maybe the features may be something like this, the same information we can consider. So, geometric features or the non geometric features we can consider. Texture or the pixel value we can consider, 2D or 3D model based features we can consider, special features like position and the motion velocity we can consider.

So, special features maybe, suppose if I consider the hand is moving, so that means we can determine the motion trajectory. And from the motion trajectory, we can determine the dynamic features like velocity acceleration. So, all these features we can determine and also the position of the hand we can also determine. So, this is the motion trajectory and also we can consider spatial temporal interest points, that also we can determine and maybe we can determine the shape feature also, that is also one important feature that can be applied for hand gesture recognition.

(Refer Slide Time: 30:31)

VGR Architecture (Cont...)

Gesture representation and feature extraction:

- Feature extraction involves selection of proper features and their extraction for classification. The selection of features is purely application specific.

Feature type	Examples	Static	Dynamic	Advantages	Limitations
Spatial domain (2D)	Finger tips location, finger direction, and silhouette Motion chain code (MCC)	✓	✓	<ul style="list-style-type: none"> • Easy to extract • Rotation invariant. 	<ul style="list-style-type: none"> • Unreliable under occlusion or varying illumination • Object view-dependent • Distorted hand trajectory distorts MCC also.
Spatial domain (3D)	Joint angles, hand location, surface texture and surface illumination	✓	✓	<ul style="list-style-type: none"> • 3D modelling can most accurately represent the state of a hand, and thus can give higher recognition accuracy 	<ul style="list-style-type: none"> • Difficult to accurately estimate 3D shape information of a hand.
Transform domain	Fourier descriptor DCT descriptor Wavelet descriptor	✓	✓	<ul style="list-style-type: none"> • RST invariant 	<ul style="list-style-type: none"> • Not able to perfectly distinguish different gestures.
Moments	Geometric moments Orthogonal moments	✓	✓	<ul style="list-style-type: none"> • Moments can be used to derive RST invariant global features. 	<ul style="list-style-type: none"> • Moments are in general global features. So, moments cannot effectively represent an occluded hand.
Curve fitting-based	Curvature scale space	✓	✓	<ul style="list-style-type: none"> • RST invariant. • Resistant to noise. 	<ul style="list-style-type: none"> • Sensitive to distortion in the boundary.
Histogram-based	Histogram of gradient (HoG) features	✓	✓	<ul style="list-style-type: none"> • Invariant to geometry and illumination changes. 	<ul style="list-style-type: none"> • Performance is not so satisfactory for images with a complex background and noise.
Interest point-based	Scale-invariant feature transform (SIFT) Speeded up robust features (SURF)	✓	✓	<ul style="list-style-type: none"> • RST and illumination invariant 	<ul style="list-style-type: none"> • They are not the best choice for real-time applications because they are computationally expensive.
Mixture of features	Combined features	✓	✓	<ul style="list-style-type: none"> • Incorporates the advantages of different types of features. 	<ul style="list-style-type: none"> • Classification performance may degrade due to curse of dimensionality.

So, in this table I have shown different types of features, they are rarely used in the research papers. So, special domain 2D features, maybe the finger tips, location, finger detection, silhouette and maybe the motion chain code can be used like this. So, we have already discussed about the chain code. And for spatial domain 3D features, we can consider joint angles, hand location, surface textures, surface illumination, like this we can consider.

And also, we can consider features in the transform domain, like the Fourier descriptors, DCT descriptors, wavelet descriptors. And also we can consider the moments, the geometric moments, the seven moment invariants we can consider. And like the histogram-based HoG features, the SIFT features, the SURF features or maybe the combined features we can used. And all these features have advantages and disadvantages. So, you can see the advantages and disadvantages of these features from the slide or maybe from my research paper you can see. So, what are the advantages of these features and the limitation of these features?

(Refer Slide Time: 31:42)

VGR Architecture: Recognition

- The final stage of a VGR system is the recognition stage where a suitable classifier recognizes the incoming gesture parameters or features and groups them into predefined classes (supervised) or by their similarity (unsupervised).
- There are various classifiers used for both static and dynamic gesture recognition, each with its own limitations.
- And based on the type of input data and the method, hand gesture recognition process can be broadly categorized into three sections:
 - Conventional methods on RGB data ✓
 - Depth-based methods on RGB-D data ✓
 - Deep networks - a new era in computer vision

And finally, we can consider the classification model that is the recognition model. So, after extracting the features, we have to recognize the gestures. So, we can apply the supervised techniques or maybe the unsupervised techniques. And maybe we can consider like the conventional methods on RGB data, we can consider or maybe the depth-based methods on RGB depth data or maybe the deep networks that we can use for recognition.

So, the classical machine learning techniques maybe the support vector machine, the K nearest neighbor. So, all these techniques can be used for gesture recognition or maybe the deep networks like the CNN, the convolution neural networks. So, this can be used or this can be employed for gesture recognition.

(Refer Slide Time: 32:40)

Conventional Methods on RGB Data

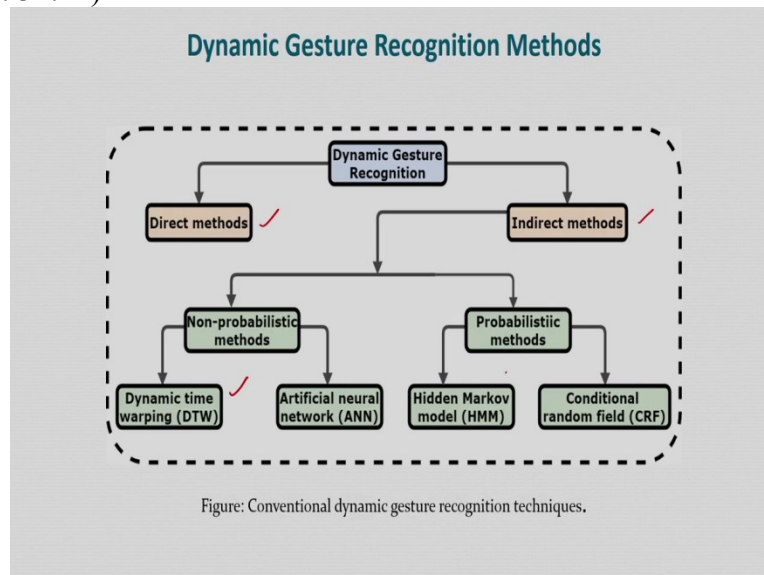
- **Static gesture recognition:** Static gestures are basically finger-spelled signs in still images without any time frame. Used classifiers are-
 - Unsupervised k-means.
 - Supervised k-NN, SVM, ANN.
- **Dynamic gesture recognition:** Dynamic gestures or trajectory-based gestures are basically gestures with trajectory having temporal information in terms of video-frames. Used methods are divided into direct and indirect one.
 - **Direct approaches** first detect the time boundaries of the performed gestures and then apply standard isolated gesture recognition. Typically, motion cues (e.g., velocity, acceleration, and trajectory curvature) or specific start and end marks, an open/closed palm can be employed for boundary detection.
 - **Model-based Indirect methods** can be of two types: non-probabilistic i.e. a) Dynamic programming/ Dynamic time warping, b) ANN; and probabilistic i.e. c) HMM and other statistical methods, d) CRF and its variants.

So, for static gesture, so for static gesture recognition we can apply maybe the supervised classification techniques like k nearest neighbor, support vector machine, artificial neural networks. For the unsupervised maybe we can consider the k means algorithms. And for dynamic gesture recognition because for dynamic gesture recognition, we have to consider the motion parameters. That means, we can consider the motion trajectory and from the motion trajectory, we can determine the dynamic features like the velocity, we can determine the acceleration we can determine, trajectory curvature we can determine from the motion trajectory.

And based on these features, we can do the classification. So, maybe we can consider some techniques like the dynamic programming, or maybe the dynamic time warping, this is a very popular algorithm for gesture recognition, maybe we can consider the artificial neural networks and the probabilistic framework, maybe we can consider the Bayes classifier also or maybe the hidden Markov model or other statistical methods we can consider.

And maybe we can consider CRF that is the Conditional Random Field, that is also very important, you consider research paper on conditional random field. So, by using the CRF also we can recognize gestures. So, in this class only, I will explain briefly the concept of the dynamic time warping and the concept of the hidden Markov model.

(Refer Slide Time: 34:11)



Now, I will discuss the conventional dynamic gesture recognition techniques. So, here in the figure you can see, I have shown the dynamic gesture recognition techniques, we may have direct methods or maybe the indirect methods. In case of the direct methods, we have to extract gesture features. And based on these features, we can do the classification, we can do the recognition.

In case of the indirect methods, we may consider non probabilistic methods or the probabilistic methods. For example, in case of the non probabilistic methods, we can use dynamic time warping, DTW and also maybe the artificial neural networks. In case of the probabilistic methods, popular methods are hidden Markov models, that is very important. And another one is the conditional random field.

So, these are some examples of conventional dynamic gesture recognition techniques. So, in this class mainly I will discussed these two algorithms, one is the DTW, Dynamic Time Warping, and another one is the brief introduction of hidden Markov model. So, these two algorithms briefly I will explain.

(Refer Slide Time: 35:32)

Dynamic Time Warping (DTW)

- Dynamic time warping (DTW), a template matching application of dynamic programming, has been widely used in isolated gesture recognition.
- DTW algorithm calculates the distance between each possible pair of points in two time series in terms of their feature vectors.

* Two time series P and Q:

$$P = p_1, p_2, \dots, p_M$$
$$Q = q_1, q_2, \dots, q_N$$

where q_i, p_i are feature vectors for the i^{th} element of the corresponding time sequences.

- * Construct $N \times M$ matrix D with distances $D_{ij} = d(p_i, q_j)$.
- * Warping path W is a contiguous set of matrix elements $w_k = (i, j)_k$.
- * Define warping between P and Q

$$W = w_1, w_2, \dots, w_K$$

where $\max(M, N) \leq K \leq M + N - 1$

- * Find:

$$DTW(P, Q) = \min \sqrt{\sum w_k}$$

So, what is DTW? You can see in my next slide. So, this is a Dynamic Time Warping algorithm. So, this algorithm is mainly the template matching. And it is used in isolated gesture recognition. Suppose, in case of the gesture recognition, I have one trajectory that is the gesture trajectory. Because the hand is moving. So, from the video I can determine the gesture trajectory. So, we have the trajectories of all the gestures. These are called the template gestures or the template trajectories.

So, when the new trajectory is coming, suppose a new trajectory is this. So, I have to do the matching. So, the matching between the template trajectory and the input trajectory, that is the test trajectory. And based on this matching, I can determine or I can recognize a particular gesture. So, I am repeating this, that means corresponding to a particular gesture sequence, gesture video, I can extract the gesture trajectory.

And I have the number of trajectories corresponding to different gestures. So, suppose you pick one, consider suppose, if I write A like this or maybe the B like this, so for all these we have the gestures trajectories. And for recognition, for classification we have to match the input trajectory, that is the test trajectory with the template trajectory. So, one algorithm is very popular that is the DTW, the Dynamic Time Warping algorithm.

So, it is a template matching algorithm. So, mainly the concept is, suppose I have a two-time series, one is suppose P and another one is Q. So, I have two-time series P and Q and I can find the similarity between these two time series. So, for this I can compute the distance between P

and Q you can see, I am finding the distance between P and Q. So, I can consider Euclidean distance or maybe the Manhattan distance.

So, any distance I can consider and based on this distance, I can find a similarity between these two time series, one is P and another one is Q. So, for this I am considering the warping part W and corresponding to this I have the W matrix suppose, and here you can see the matrix element is w_k . So, define warping between P and Q like this. So, W is nothing but w_1, w_2, w_k like this. So, for this what I have to consider?

I have to find the DTW between P and Q in this expression you can see I am finding the DTW between P and Q, that is nothing but the minimum distance between the time series P and Q. So, based on this, I can find a similarity between the time series P and Q. And based on this I can recognize. So, the actual concept I am going to explain in my next slide.

(Refer Slide Time: 38:40)

Metric Distances

- What properties should a similarity distance have?
- $D(A,B) = D(B,A)$ Symmetry
- $D(A,A) = 0$ Constancy of Self-Similarity
- $D(A,B) \geq 0$ Positivity ✓
- $D(A,B) \leq D(A,C) + D(B,C)$ Triangular Inequality

So, for matching, what we can consider? We can consider a distance, maybe we can consider Euclidean distance or any other distance, city block distance also we can consider. But what properties should a similarity distance have? So, we can consider a metric distance, the property of a distance measure should be like this. The distance between two points, the point A and B, $D(A,B)$ should be equal to $D(B,A)$, that is the symmetry property.

And the distance between the same point $D(A,A)$ that should be equal to 0, that is the self similarity. And also, the distance between the point A and B should be greater than equal to 0

that is the positivity. And this is also you know that is the triangular inequality. So, distance between these two points $D(A, B)$ should be less than equal to the distance between the point A and C plus the distance between the point B and C. So, these are the properties of the metric distances. So, based on this, we can consider the Euclidean distance.

(Refer Slide Time: 39:56)

Euclidean Similarity Measure

- View each sequence as a point in n-dimensional Euclidean space (n = length of each sequence)
- Define (dis-)similarity between sequences X and Y as

$$L_p = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

p=1 Manhattan distance ✓

p=2 Euclidean distance

So, in the next slide you can see I am considering the distance between suppose two points or maybe the two sequences, I am considering X and Y sequences, and I am finding the distance between these two. So, L_p is the distance and suppose the p is equal to 1, then it corresponds to Manhattan distance. If p is equal to 2, that corresponds to Euclidean distance. So, this distance measure I can consider to find a similarity between the time series P and the time series Q.

(Refer Slide Time: 40:28)

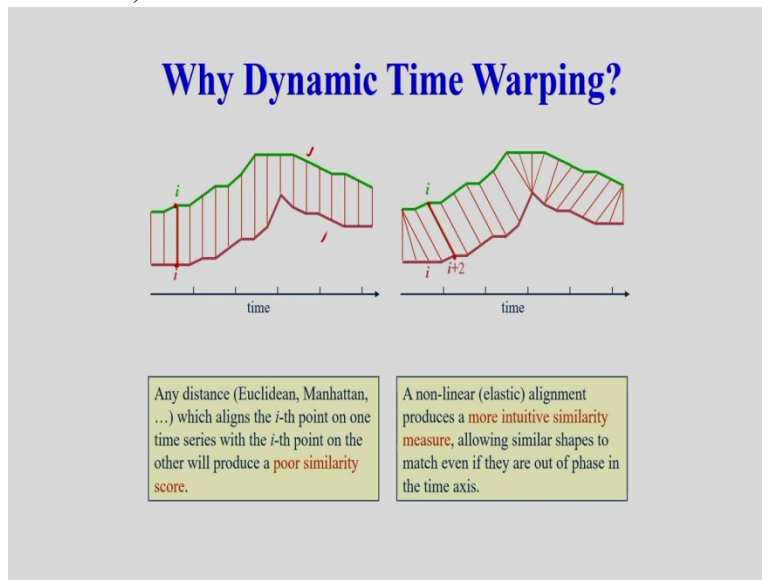
Dynamic Time Warping

- Allows acceleration-deceleration of signals along the time dimension
- Basic idea
 - Consider $A = a_1, a_2, \dots, a_n$, and $B = b_1, b_2, \dots, b_n$
 - We are allowed to extend each sequence by repeating elements
 - Euclidean distance now calculated between the extended sequences X' and Y'
 - Matrix M , where $m_{ij} = d(x_i, y_j)$

So, for this the basic idea is let us consider the series, the time series A that is a_1 up to a_n and another series I am considering B that is b_1 up to b_n . And we are allowed to extend each sequence by repeating elements. So, that means, we can extend the sequence. After this we can calculate the Euclidean distance between these two extended sequences, one is X' another one is Y' .

So, the X' means the extended sequence and Y' is the extended sequence that is the y sequence. So, we can find the Euclidean distance between these two sequences. And in this case, we are getting the matrix, the matrix is m that is nothing but the elements are like this m_{ij} is the element of the matrix M. What is the element m_{ij} ? That is nothing but the distance between X_i and Y_j .

□□(Refer Slide Time: 41:32)

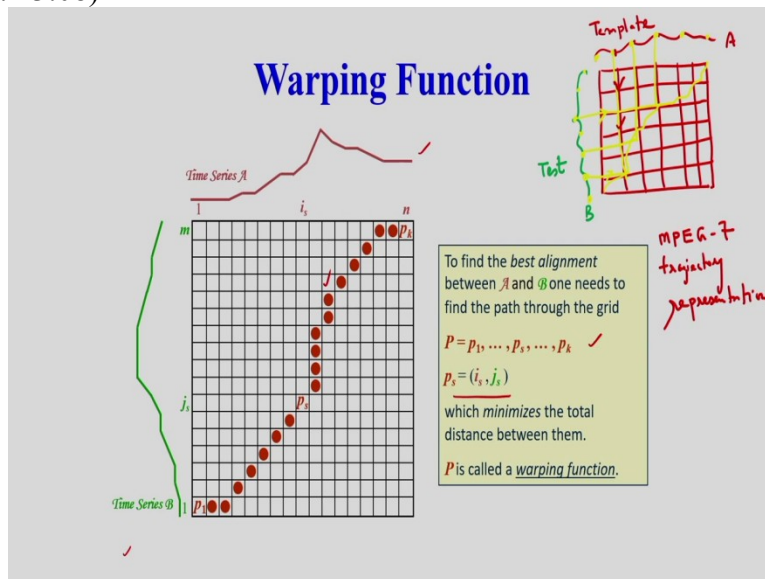


Suppose, if I consider these two-time series I am considering, one is the green another one is the red. So, what is the importance of dynamic time warping, that I want to show here. And suppose, if I want to match these two sequences, point to points. Suppose, the point i is match with the point i in the second sequence. Then in this case it gives poor similarity score. The point I of the first sequence is match with the point i of the second sequence.

So, it gives the poor similarity score, but actually what is happening here, the point i of the first sequence corresponds to the point i plus two in the second sequence. So, we have to find a matching between i and the i plus two. So, that is why the nonlinear elastic alignment is important. In the previous case in the first case, I am just considering the point i of the first sequence and the point i of the second sequence. So, it gives poor similarity score.

But in the second case what I am considering? I am considering a nonlinear elastic alignment, then, then in this case I am getting the good measure, the similarity score, because the point i of the first sequence actually corresponds to the point i plus two in the second sequence. So, that is why the importance of dynamic time warping, that means we need a nonlinear elastic alignment.

(Refer Slide Time: 43:08)



So, in this figure you can see I am considering the warping function, I am considering the time series A and the time series B, you can see here the time series A and the time series B I am considering. Now, I want to find the best alignment between A and B. And for this I need to find a path through the grid. So, that means I have to find a path, the path is represented by P is equal to p_1, p_2, p_3 like this. So, P_s is nothing but is comma j_s .

So, what is the best path? That means the best path minimize the total distance between them. So, what is the best path, the base path minimizes the total distance between the two sequences. One is the time series A another one is the time series B. And the P is called the warping function. So, in this figure you can see I have shown the, the warping function p_1, p_2, p_3 like this.

So, this is the warping function. So, in case of the gesture recognition already I have mentioned, so I have to compare the test trajectory and the template trajectory. So, suppose I am considering the DTW algorithm. Suppose, I am considering one trajectory something like this, this is the gesture trajectory A suppose. And that is nothing but the suppose template. So, I am considering this is a template.

Template gesture trajectory, and after this I am considering the another trajectory that is the test trajectory something like this. So, trajectory is B suppose this is test trajectory. And by this DTW algorithm, I can find the alignment between A and B. So, I can find a correspondence between

the template A and the template B, that is the template and the test trajectory. So, suppose this is the alignment between A and B that we can find, that is the warping function.

So, the best alignment we can find based on the distance between the time series A and the time series B. So, based on this we can recognize a particular gesture. So, we have the template gestures, we have the templates and whenever the new gesture is coming, we have to match the test gesture with the template gestures. But it is time consuming because for one gesture I have to compare each and every templates.

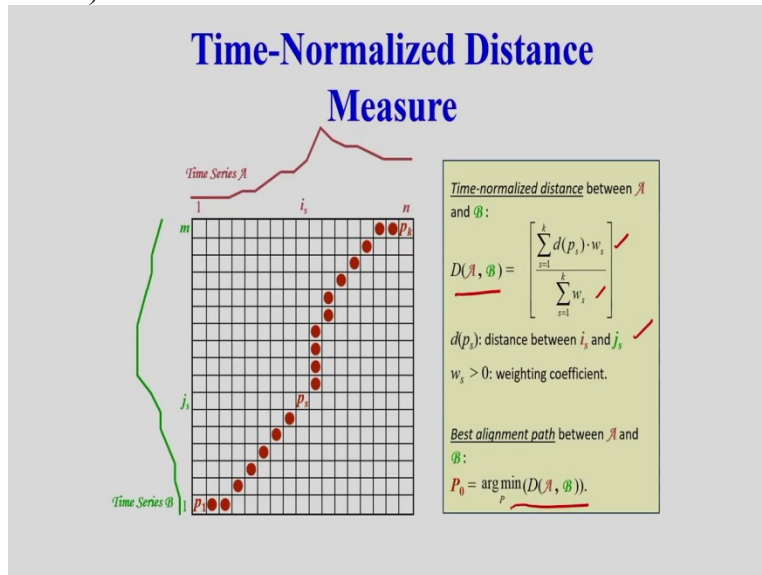
So, that is why it is computationally complex. And suppose in the template, if I can identify some of the key points, suppose these are the key points in the template and from this I can determine gesture features. Suppose I can extract some features like orientation feature, the length feature or maybe the dynamic feature like velocity acceleration I can determine. And similarly, for the test trajectory also I can determine the, the key points, the key points I can determine and these key points I can match, these key points I can match based on the warping function.

And by this process, I can determine the feature corresponding to the test gesture trajectory. So, you can see I am just doing the matching based on the key points. One is the template trajectory, another one is just the test trajectory. In the template trajectory I can find the key points. So, maybe something like the MPEG 7 the trajectory descriptors, trajectory representation, MPEG 7 trajectory representation we can consider.

So, we can determine the some of the key points like this. So, I have shown the key points and I can find the correspondence between the key points and after determining the key points in the test trajectory, I can determine the gesture features. The static feature and the dynamic feature, I can extract from the key points because from these key points of the template, I can find the corresponding key points of the test trajectory.

And after this we can determine the, the feature the static feature like linked between the key points or maybe the orientations or maybe the dynamic feature like velocity acceleration I can determine and based on this I can recognize a particular gesture. So, this is one example how to recognize a particular gesture. So, we have been discussing about the warping function. So, in the next slide also I will explain the concept of the warping function.

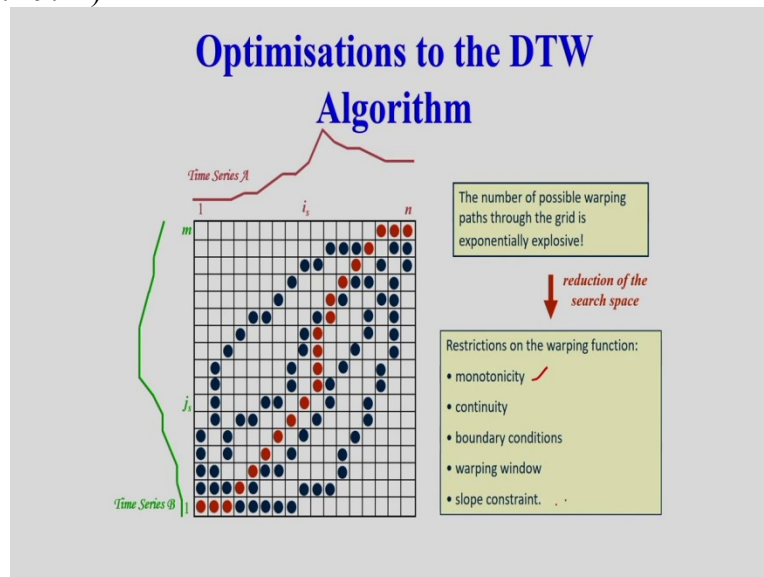
(Refer Slide Time: 48:13)



So, here you can see I am finding the time normalized distance between A and B, the time series A and time series B. So, depth distance I am determining. So, $D(A, B)$ is the time normalized distance and here you can see the distance is divided by the sum of the coefficients. That is the weighting coefficients. So, what is $d(p, s)$? $d(p, s)$ is the distance between i_s and the j_s , that is the time series A and the time series B.

So, for this I am finding the distance between i_s and the j_s . And I am considering the weighting functions or weighting I am considering the weighting coefficients W_s . So, that is the time normalized distance between A and B because the distance is divided by the sum of the coefficients. So, what is the best alignment path between A and B, the best alignment path is nothing but the minimum distance between the time series A and the time series B. So, based on this I can find the best alignment path between A and B.

(Refer Slide Time: 49:21)



So, in this case, you can see the optimization to the DTW algorithm, because number of possible warping path through the grid is exponentially explosive. So, maybe I can consider this warping path or maybe I can consider this warping path like this I can consider many, many warping path. But which one is the best, I have to determine. So, that is why I have to do the optimization, optimization to the DTW algorithm.

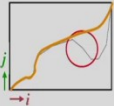
So, for this I have to consider some restrictions on the warping function. So, this restriction we have to consider, one is the monotonic condition, continuity condition, boundary conditions, warping window size consideration and the slope constraint. So, I am going to explain one by one. So, based on this I can find the best the warping path.

(Refer Slide Time: 50:15)

Restrictions on the Warping Function

Monotonicity: $i_{s-1} \leq i_s$ and $j_{s-1} \leq j_s$

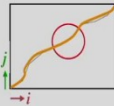
The alignment path does not go back in "time" index.



Guarantees that features are not repeated in the alignment.

Continuity: $i_s - i_{s-1} \leq 1$ and $j_s - j_{s-1} \leq 1$.

The alignment path does not jump in "time" index.



Guarantees that the alignment does not omit important features.

So, first one is the monotonic condition. So, i_{s-1} should be less than equal to i_s . And similarly, j_{s-1} should be less than equal to j_s . The meaning is the alignment path does not go back in time index. So, that means, it should be monotonically increasing, the warping path, the warping function should be monotonically increasing. That means, it guarantees that features are not repeated in the alignment. So, the feature should not be repeated during the alignment.

So, that is why we are considering the monotonic condition. Next, I am considering the continuity condition. So, here you can see I am considering a discontinuous alignment path, the alignment path does not jump in time index. So, that is one important point, that the alignment path does not jump in the time index. So, there should be continuity. So, in the figure I am showing the discontinuity.

So, what is the importance of the continuity? So, guarantees that the alignment does not omit important feature. So, if I consider the discontinuous alignment path then that means, I may miss some important feature. So, that is why the continuity condition is important. So, these two conditions one is the monotonic condition, another one is the continuity condition these are very important. So, next slide you can see I am showing the, the monotonic condition here you can see. So, it is monotonically increasing and also, I am showing that continuity path that is I am showing the continuity that this alignment path does not jump in time index, it should be continuous.

(Refer Slide Time: 52:03)

Restrictions on the Warping Function

Boundary Conditions: $i_1 = 1, i_k = n$ and $j_1 = 1, j_k = m$.

The alignment path starts at the bottom left and ends at the top right.

Guarantees that the alignment does not consider partially one of the sequences.

Warping Window: $|i_i - j_i| \leq r$, where $r > 0$ is the window length.

A good alignment path is unlikely to wander too far from the diagonal.

Guarantees that the alignment does not try to skip different features and gets stuck at similar features.

Next one is I am considering the boundary conditions. So, that means, in this figure you can see this alignment path, it is starting from this point and ending at this point, but it should not be like this. So, the condition is the alignment path start at the bottom left and the ends at the top left. So, that should be the condition, but in the figure, I am showing there, here you can see the alignment path is starting from this point and it is ending at this point.

But actually, it should start at the bottom left and ends at the top left. So, what is the importance of this. So, it guarantees that the alignment does not consider partially one of the sequences. So, that is why we have to consider the starting point like this i_1 should be equal to 1 and i_k should be equal to n and similarly, j_1 should be equal to 1 and the ending point should be j_k is equal to m. So, here you can see I am showing the actual alignment path based on the boundary condition.

So, the second one is the, the alignment path considering the, the boundary conditions. Next point is warping window. So, that means in this case I am defining the window size. So, which one is the best window I am considering. So, is minus js should be less than equal to r. And r should be greater than 0. So, that means I am considering the window length. So, a good alignment path is unlikely to wander too far from the diagonal.

That means the best alignment path will be very close to the diagonal. So, alignment paths should be close to this diagonal. So, that means, what is the importance of this? It guarantees that the alignment does not try to skip the parallel feature and get stuck at similar feature. So, that is

why the window size is important. So, here in the figure you can see I am showing the, the window, the length of the window I am considering r is the length of the window. And based on this window, I can see the alignment path like this, it is close to the diagonal. So, it should not wander too far from the diagonal. So, that is the concept of the warping window. So, this condition also we need to consider.

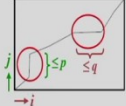
(Refer Slide Time: 54:34)

Restrictions on the Warping Function

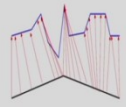
Slope Constraint: $(j_p - j_{p_0}) / (i_p - i_{p_0}) \leq p$ and $(i_q - i_{q_0}) / (j_q - j_{q_0}) \leq q$, where $q \geq 0$ is the number of steps in the x -direction and $p \geq 0$ is the number of steps in the y -direction.

After q steps in x one must step in y and vice versa: $S = p / q \in [0, \infty]$.

The alignment path should not be too steep or too shallow.



Prevents that very short parts of the sequences are matched to very long ones.



And finally, the last a constraint that is a slope constraint I am considering. So, this equation is nothing but you can see, I am considering the slope and also, I am continuing considering the slope. So, what is p ? p means, the number of steps in y direction and what is q ? The q is the number of steps in x direction, I am considering both the directions, one is the x direction and another one is the y direction.

So, in this case the q is the number of steps in the x direction and the p is the number of steps in y direction. So, that means in this case the condition is the alignment path should not be too steep or too shallow. So, what is the importance of this constraint? That means, it prevents that very short parts of the sequences are match to very long ones. So, that is the slope constraint. So, we have to consider all these constraints, the monotonic constraint, the continuity constraints, the slope constraint, window size all this we have to consider, corresponding to the warping function.

(Refer Slide Time: 55:45)

The Choice of the Weighting Coefficient

Time-normalized distance between \mathcal{A} and \mathcal{B} :

$$D(\mathcal{A}, \mathcal{B}) = \min_p \left[\frac{\sum_{s=1}^k d(p_s) \cdot w_s}{\sum_{s=1}^k w_s} \right]$$

← complicates optimisation

Seeking a weighting coefficient function which guarantees that:

$$C = \sum_{s=1}^k w_s$$

is independent of the warping function. Thus

$$D(\mathcal{A}, \mathcal{B}) = \frac{1}{C} \min_p \left[\sum_{s=1}^k d(p_s) \cdot w_s \right]$$

can be solved by use of dynamic programming.

Weighting Coefficient Definitions

- Symmetric form
 $w_s = (i_s - i_{s-1}) + (j_s - j_{s-1})$, ✓
then $C = n + m$. ✓
- Asymmetric form
 $w_s = (i_s - i_{s-1})$, ✓
then $C = n$. ✓

Or equivalently,

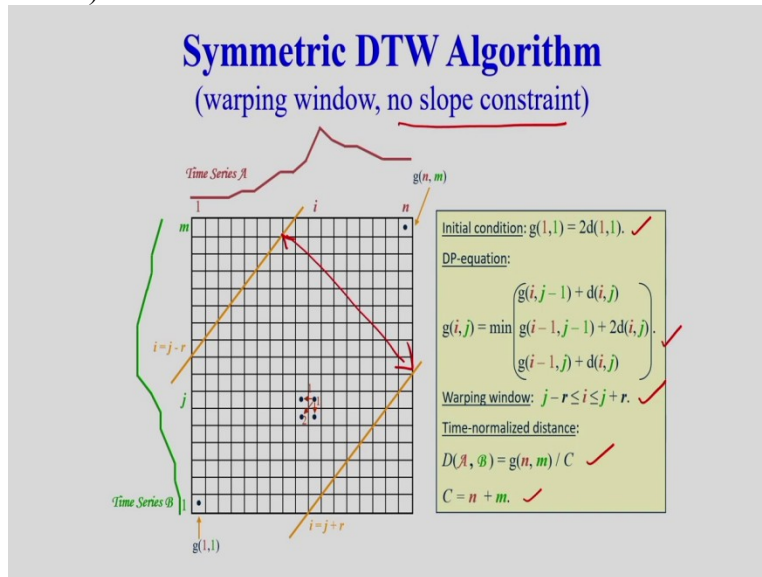
- $w_s = (j_s - j_{s-1})$, ✓
then $C = m$. ✓

So, here I am considering the time normalized distance between A and B. And here you can see the distance between A and B I am considering and this distance is normalized by the summation of the warping coefficients. So, now, I am considering C is equal to the summation of w_s , it is starting from s is equal to 1 to k. So, w_s is nothing but the weighting coefficients. And this weighting coefficient function, it should be independent.

So, I am getting C, C is nothing but the summation of w_s , s is equal to 1 to k. And it should be independent of the warping function. So, that is why since it is independent of the warping function, so I am taking it out. So, it is 1 by C minimum and this, I am the distance, this I am considering. So, that means, I am considering the distance between A and B. And that is nothing but a time normalized distance.

And this, the distance between D AB, I can determine and it can be solved by considering the dynamic programming. So, there are two forms, one is the symmetry form another one is the asymmetric form. In the symmetric form the weighting coefficients, we can determine like this, $i_s - i_{s-1} + j_s - j_{s-1}$. And C is defined like this, C is equal to $n+m$. And similarly, if I consider the asymmetric form corresponding to the asymmetric form w_s is this. And C is equal to n or maybe C is equal to m. So, I may consider the symmetric form or the asymmetric form for solving the dynamic programming problem.

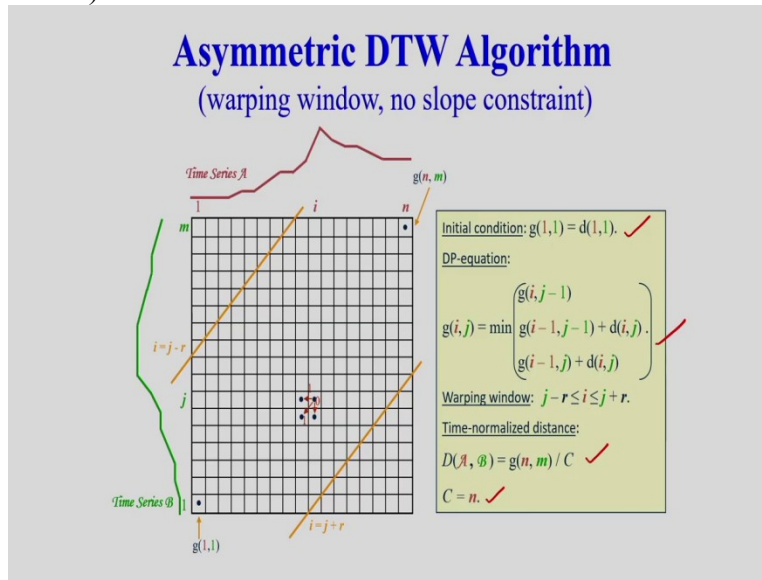
(Refer Slide Time: 57:39)



So, here you can see I am considering the symmetric DTW algorithm. And I am not considering the slope constraint, I am not considering. And you can see the warping window that, that is defined between these two yellow lines. So, this is the warping window I am considering between these two yellow lines. And initial condition also we have to consider, that is $g(1, 1)$ is equal to, to $d(1, 1)$. That is the initial condition for the dynamic programming. So, this DP equation, the dynamic programming equation we can employ.

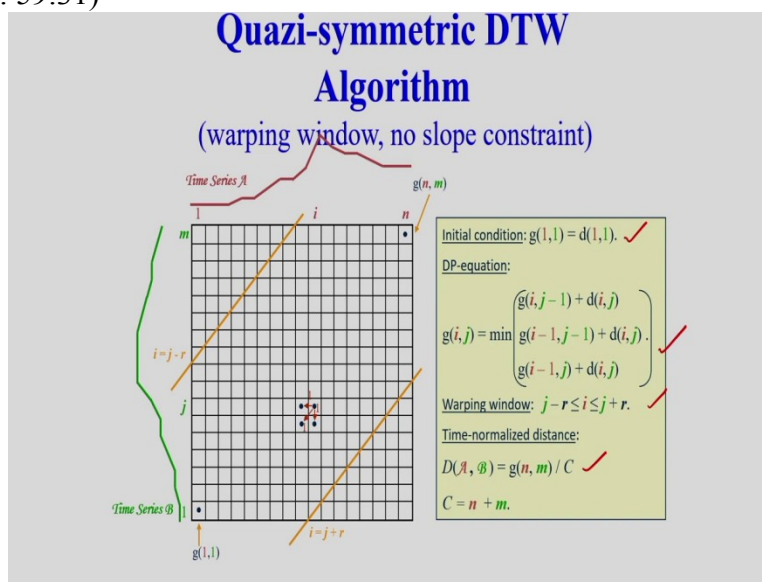
And this warping window we have, we have defined. And we can determine the time normalized distance. And C is nothing but $n + m$. So, this symmetry DTW algorithm we can employ. And that means, we are considering the distance between the time series A and time series B. And we have shown the, window, the warping window we have shown.

(Refer Slide Time: 58:40)



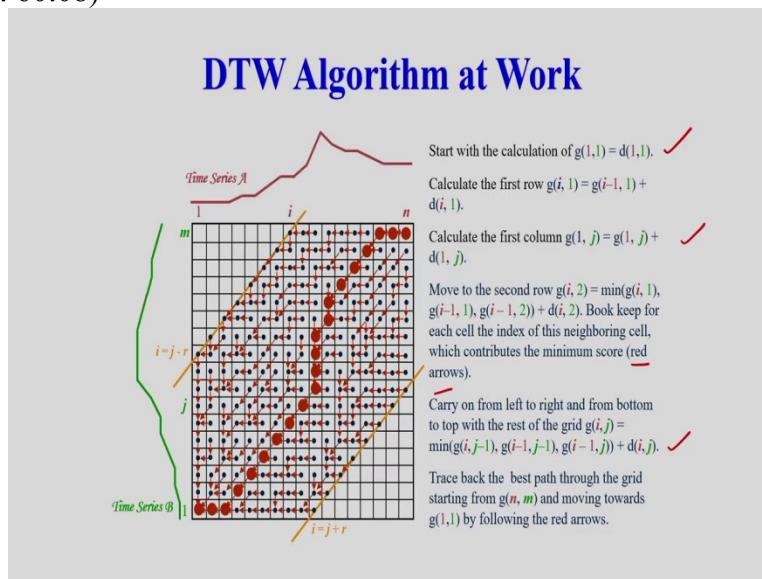
And similarly, we can consider the asymmetric DTW algorithm. Again, in this case I am defining the window, that is the warping window I am considering. And slope constraint I am not considering. And this is the initial condition for the asymmetric DTW algorithm and we can consider the DP algorithm, the Dynamic Programming equations corresponding to asymmetric DTW algorithm. And from this we can determine the time normalized distance $D(A, B)$, we can determine. And in this case, we can consider C is equal to n . So, this symmetric DTW algorithm or the asymmetric DTW algorithm we can employ, to find the time normalized distance. That means, I want to find the alignment between time series A and the time series B.

(Refer Slide Time: 59:31)



And also, we can consider the Quazi symmetric DTW algorithm, the concept is very similar, but the initial condition is this $g(1, 1)$ is equal to $d(1, 1)$ that is the condition. And we can consider the dynamic programming equations like this, $g(i, j)$ we can consider. So, this is from the dynamic programming equations. And we can consider the warping window and we can determine the time normalized distance, we can determine. So, this DTW one is the symmetric, one is the asymmetric and one is the Quazi, Quazi symmetric DTW we can consider.

(Refer Slide Time: 60:08)



So, now, let us see one example, how we can find a best alignment between time series A and the time series B. So, here I am showing the time series A and the time series B. And I am also showing that window, that is between two yellow lines. So, first let us start with the calculation. So, first we are considering $g(1, 1)$ is equal to $d(1, 1)$, that I am considering. After this the calculate the first row, you can see in the figure here I am calculating the first row by this DP equation, $g(i, 1)$ I have calculated.

The first row I have calculated, after this calculate the first column by using this DP algorithm. The DP equations, move to the second row that means, I am considering the second row if you see that figure, move to the second row that is $g(i, 1)$. We can determine $g(i, 2)$, that is the minimum of $g(i, 1), g(i-1, 1), g(i-1, 2) + d(i, 2)$. And after this keep, book keep for each cell, the index of this neighboring cell, which contributes the minimum score.

So, that means the minimum score, the distance we have to consider and that is shown by the red arrows. So, in the figure you can see, I am showing the red arrows. So, book keep for each cell, the index of the neighboring cell which contributes the minimum score, that we are considering. And carry forward, carry on from left to right and from the bottom to top with the rest of the grid, we have to consider. And based on this $g(i, j)$ we can determine by this DP equations.

So, pictorially again I am showing here. So, first we have calculate, the first row and after this, we considered the first column after this, we are considering the second row, after this we have to consider $g(i, j)$ like this we are, we are computing by using the DP equations. After this, trace the best path through the grid starting from g_{nm} and moving towards $g(1, 1)$, following the red arrows.

So, that means, I am finding the best path I am determining. So, how to find the best path? You can see it again. So, trace back the best path through the grid starting from g_{nm} and moving towards $g(1, 1)$ by following the red arrows. So, that means, I am getting the best alignment path between the time series A and the time series B. So, this is the brief concept of the DTW algorithm. So, if you want to see the details of this DTW algorithm, you can see the book, the even in the speech recognition book you will get this algorithm, that DTW algorithm.

The book by a (63:10) or maybe other research papers you can see for this algorithm. So, DTW algorithm. So, briefly I have explained the concept of the DTW algorithm and how it can be used for gesture recognition. But the main problem already I have explained, the main problem is the computational complexity. Because I have to compare the test template with all the template trajectories. So, it is computationally expensive. So, this is about the DTW algorithm.

(Refer Slide Time: 63:41)

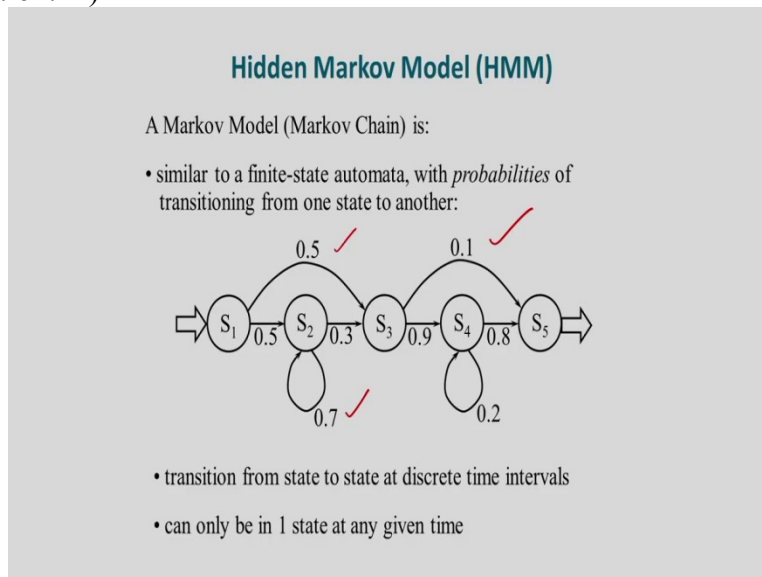
Hidden Markov Model (HMM)

- A particular gesture trajectory is represented by a set of feature vectors, where each feature vector describes the dynamics of a hand corresponding to a particular state of a gesture.
- The number of such states depends on the nature and complexity of a gesture.
- The global HMM structure is formed by connecting in parallel the trained HMMs $(\lambda_1, \lambda_2, \dots, \lambda_G)$ where G is number of gestures to be recognized.

Now, let us consider the concept of the hidden Markov model. So, briefly I will explain the this concept, the hidden Markov model. So, a particular gesture trajectory is represented by a set of feature vectors. Where each feature vector describes the dynamics of the hand corresponding to a particular state of the gesture. So, that means we are considering the feature vector describing the dynamics of the hand corresponding to a particular state of a gesture. That we are considering.

The number of such states depend on the nature and the complexity of the gestures. So, number of states of the hidden Markov model depends on the nature and the complexity of a particular gesture. That global HMM structure is formed by connecting in parallel the trained HMM, that is a train HMM sorry, λ_1, λ_2 like this, where g is the number of gestures to be recognized. So, for each and every gesture I have the train HMM.

(Refer Slide Time: 64:44)



So, now, I am showing the hidden Markov model. So, Markov chain I have shown. So, it has number of states S_1, S_2, S_3, S_4, S_5 like this. And I have shown the transition from one state to another state, the transition from S_1 to S_3 , the transition from S_1 to S_2 , the transition from S_3 to S_4 like this you can see. And with the, the probability suppose the transition from S_1 to S_3 with a probability the 0.5.

And also I have shown the self transition you can see in case of the state S_2 the self transition is taking place with a probability, the probability is 0.7. So, transition from one state to another state I have shown here, that means a hidden Markov model has finite setup states and each of which is associated with a multi dimensional probability distribution. And also you can see already I have defined that is the transition among the states are governed by a set of probabilities called transition probabilities.

So, in the figure I have shown the transition probabilities like 0.5, 0.1, I have shown the transition probabilities. In a particular state an outcome or observation can be generated. So, corresponding to particular state, I can see the outcome or the observation can be generated, according to the associated probability distribution. And why it is called a hidden? I will explain. So, it is only the outcome, not the state visible to an external observer and therefore, states are hidden to the outside.

Hence, the name is the hidden Markov model, because I am considering the Markov chains and why I am considering the term hidden? Because the states are not visible to an external observer. So, what is visible? Only the outcome is visible. So, that is, it is only the outcome not the state visible to an external observer. And therefore, the states are hidden to the outside and that is the name is the hidden Markov model. So, that is the concept of the hidden Markov model, the brief concept.

(Refer Slide Time: 67:09)

Hidden Markov Model

Elements of a Hidden Markov Model:

- clock $t = \{1, 2, 3, \dots, T\}$ ✓
- N states $q = \{1, 2, 3, \dots, N\}$ ✓
- M no of observationsymbol $O = \{v_1, v_2, v_3, \dots, v_M\}$ ✓
- initial probabilities $\pi_j = P[q_1 = j]$ ✓ $1 \leq j \leq N$
- transition probabilities(A) $a_{ij} = P[q_t = j | q_{t-1} = i]$ $1 \leq i, j \leq N$
- Observation probabilities(B) $b_j(k) = P[o_t = v_k | q_t = j]$ $1 \leq k \leq M$
 $b_j(o_t) = P[o_t = v_k | q_t = j]$ $1 \leq k \leq M$
- $A =$ matrix of a_{ij} values, $B =$ set of observation probabilities, $\lambda_1 \rightarrow a_{11}$
 $\pi =$ vector of π_j values. $\lambda_2 \rightarrow a_{12}$
- Entire Model: $\lambda = (A, B, \pi)$ ✓

And what are the elements of the hidden Markov models? So, first the clock. So, the clock is defined like this, the t is equal to 1, 2, 3 this is, this is the clock. And corresponding to this I have the n number of states, q ? Q is the states 1, 2, 3. So, n number of states are available, m number of observation symbols. So, observation symbols I am considering O , v_1, v_2, v_3 like this. So, number of observation symbols I am considering.

And the initial probabilities of the states is defined by π_j . So, this is the initial probabilities of the states, that is also defined that is the π_j . And the transition probability, that is a_{ij} . So, transition from one state to another state that is also defined, the transition probabilities. And the observation probabilities also it is defined, that is the B . b_j it is defined the observation probabilities corresponding to a particular state.

So, in this case, so what are the main elements now? What are the main elements of the hidden Markov model? One is the matrix A . So, in the matrix A , the elements are a_{ij} . So, what is a_{ij} , that

is the transition probability. What is the B? The set of observation probabilities, and what is π ? The π is nothing but vector of π_j values, that is the initial probabilities. So, that means, the hidden Markov model is defined by this λ .

So, lambda is A comma B comma pi. In case of gesture recognition for each and every gesture, I have one hidden Markov model. So, suppose the λ_1 corresponds to the gesture one, λ_2 corresponds to gesture two. So, like this I have number of hidden Markov models, corresponding to each and every gesture. So, hidden Markov model is represented by these parameters, so one is A, another one is B, another one is pi. So, one is the matrix of a_{ij} value. So, what is this? This is nothing but the transition probabilities, the B is nothing but the observation probabilities, the π is nothing but the initial probabilities. So, the model is represented by A B pi.

(Refer Slide Time: 69:35)

Hidden Markov Model (HMM)

- i. The state transition probability distribution A, which gives the probability of transition from the current state to the next possible state. ✓
- ii. The observation symbol probability distribution B, which gives the probability of observation for the present state of the model.
- iii. The initial state distribution Π , which gives the probability of a state being an initial state.

(A, B, Π)

So, the hidden Markov model you can see the state transition distribution A, which gives the probability of transition from the current state to the next possible state. So, in case of a hidden Markov model, you can see here the state transition probability distribution A, which gives the probability of transition from the current state to the next possible state. So, this is the transition probability distribution A.

What about B? The observation symbol probability distribution B which gives the probability of observation for the present state of the model, that is nothing but the observation symbol probability distribution, the initial state distribution that is pi which gives the probability of a

state being an initial state. So, that is the definition of A, B, π . So, in case of the hidden Markov model, so these parameters are important one is A , another one is B , another one is π . So, this is the hidden Markov model.

(Refer Slide Time: 70:37)

- * **Evaluation:** Given the model $\lambda = (A, B, \Pi)$, what is the probability of occurrence of a particular observation sequence (gesture sequence) $O = \{o_1, \dots, o_T\} = P(O|\lambda)$? This is the classification/recognition problem. This is actually the determination of the probability that a particular model will generate the observed gesture sequence when there is a trained model for each of the gesture classes (forward-backward algorithm).
- * **Decoding:** Determination of optimal state sequence that produces an observation sequence $O = \{o_1, \dots, o_T\}$ (Viterbi algorithm).
- * **Learning:** Determination of the model λ , given a training set of observations, *i.e.*, find λ , such that $P(O|\lambda)$ is maximal. Train and adjust the model to maximize the observation sequence probability such that HMM should identify a similar observation sequence in future (Baum-Welch algorithm).

In case of the hidden Markov model, we have three problems, the basic problems. The first problem is the evaluation. So, given the model, the model is represented by (A, B, π) . So, what we have to determine, what is the probability of occurrence of a particular observation sequence, that is the gesture sequence. Gesture sequence is represented by O_1, O_2, O_3 like this. So, this is a gesture sequence. And we have to determine the probability of O given λ , λ is nothing but the model.

And this is a classification or the recognition problem. This is actually the determination of the probability that a particular model will generate the observed gesture sequence when there is a trained model for each of the distinct classes. So, this can be obtained by the algorithm, the algorithm is the forward, backward algorithm. So, I am not explaining these algorithms. So, you can see the research papers on this.

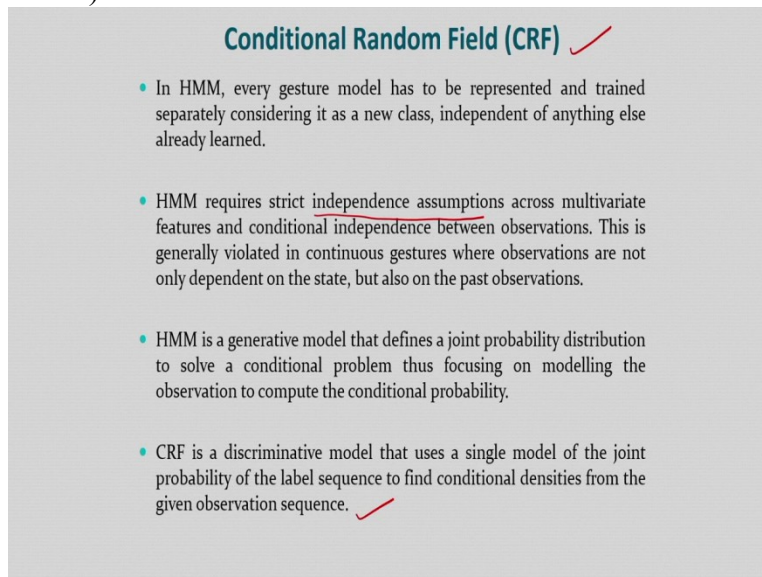
So, what is the forward backward algorithm? The problem is the probability of O given λ , that we have to determine. So, that is the recognition problem, what is the decoding problem? The decoding problem is determination of optimal state sequence, that produces an observation

sequence. So, this algorithm is called the Viterbi algorithm also you can see this algorithm, what is the algorithm. But this problem is called a decoding problem.

But one important problem is the learning problem, that is the training of the hidden Markov model. Determination of the model λ , given a training set of observation. That means, we have to find lambda such that the probability of O given lambda is maxima, maximal this is nothing but the training of the hidden Markov model. Train and adjust the model to maximize the observation sequence probability such that HMM should identify a similar observation sequence in the future.

So, this Baum Welch algorithm also you can see. So, these algorithms are very important, one is the forward backward algorithm, for this problem, the problem is we have to determine the probability of O given lambda. And the decoding problem that is nothing but the Viterbi algorithm and the learning problem that is the training of the hidden Markov model. So, we have to maximize the probability of O given lambda. So, for this we can consider the Baum Welch algorithm.

(Refer Slide Time: 73:09)



Conditional Random Field (CRF) ✓

- In HMM, every gesture model has to be represented and trained separately considering it as a new class, independent of anything else already learned.
- HMM requires strict independence assumptions across multivariate features and conditional independence between observations. This is generally violated in continuous gestures where observations are not only dependent on the state, but also on the past observations.
- HMM is a generative model that defines a joint probability distribution to solve a conditional problem thus focusing on modelling the observation to compute the conditional probability.
- CRF is a discriminative model that uses a single model of the joint probability of the label sequence to find conditional densities from the given observation sequence. ✓

So, in case of the hidden Markov model, what are the main problems? So, we must know all the possible states in advance. That means, we must know possible state connection in advance. So, for all the gestures, we have to consider this. That means, we should know all the possible states in advance. Also, we should know the possible state connection in advance. And this HMM cannot recognize the gestures or the things outside the model, because already I have explained.

So, first we have to form the model, we have to train the model by using the Baum Welch still algorithm and after this we can recognize. But it cannot recognize things or the gestures outside the model and it must have some estimate of state emission probabilities and state transition probabilities. That means, it must have some estimate of state emission probabilities and state transition probabilities. And the hidden Markov model make several assumptions.

So, these are the problems with the hidden Markov model. So, in case of a hidden Markov model every gesture model has to be represented and trained separately considering it as a new class, independent of anything else already learned. And in case of the hidden Markov model requires strict independent assumptions across multivariate features. And the conditional independence between the observations.

So, this is one important requirement. This is generally violated in continuous gesture recognition. So, for continuous gesture recognition, we have to modify the hidden Markov model. So, in the research paper you can see the, in case of the continuous gesture also we can apply the hidden Markov model. The hidden Markov model is generative model that defines a joint probability distribution to solve a conditional problem. So, that is why, we can consider another model that is the conditional random field.

This is very popular in case of gesture recognition. CRF is a discriminative model that uses a single model of the joint probability of the label sequence to find conditional densities from the given observation sequence. So, the main concept of the CRF, it is the discriminative model and it uses a single model of the joint probability of the label Sequence, to find conditional density from the given observation sequence. So, this concept you can also see from the research papers. Just I am mentioning the concept of the hidden Markov model and the conditional random field.

(Refer Slide Time: 76:04)

Deep networks - a new era in computer vision

- Recently, deep learning has irrupted in action and gesture recognition fields achieving outstanding results and outperforming "non-deep" state-of-the-art methods.
- Deep learning (DL) provides a plausible way of automatically learning multiple level features, by using multiple processing layers to learn image representations with multiple levels of feature abstraction.
- Need of DL techniques: huge data requirement with good processing unit.
- Convolutional neural networks (CNN) for images, ✓
- 3D-CNN (C3D) model for videos,
- Long-term video prediction-RNN/LSTM/GRU:
 - Recurrent neural network (RNN) ✓
 - Long short-term memory (LSTM) ✓
 - Gated recurrent units (GRU)

And finally, nowadays, our most recently, the deep networks are used in case of gesture recognition. So, maybe we can use something like a convolution neural networks or maybe the recurrent neural networks, the long- and short-term memories, the concept in the recurrent neural network. So, these types of networks are recently used for gesture recognition, you can see the recently the deep learning has erupted in action.

And gesture recognition fields, (())(76:12) being outstanding results and outperforming non deep state of the art methods, like the state of the art method is the hidden Markov model DTW like this. So, now, it is or recently the deep learning techniques are used. And the popular networks are the CNN or maybe the recurrent neural networks. So, these types of networks are used for gesture recognition.

So, in this class, I discussed the basic concept of the gesture recognition and I have highlighted some of the applications of gesture recognitions. I discussed the concept of aesthetic gesture recognition and the dynamic gesture recognition. For dynamic gesture recognition, we can consider the hidden Markov model or maybe the DTW. So, briefly I have explained the concept of the DTW algorithm and also the hidden Markov model.

So, for more detail, you can see the related research paper on gesture recognition. So, there are many new techniques available for gesture recognition. So, you can see all these concepts in the research papers. So, let me stop here today. Thank you.