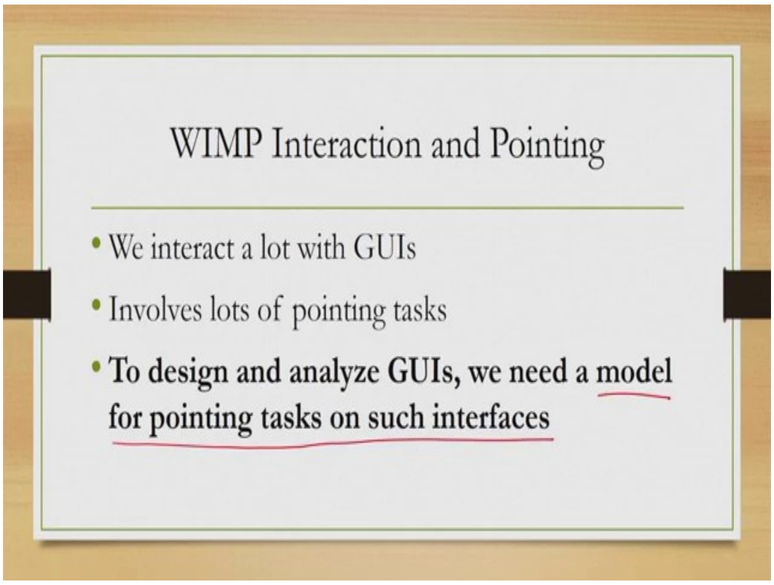
User-Centric Computing for Human-Computer Interaction
Prof. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture – 18
2D and 3D pointing models

Hello and welcome to lecture number 18 in the course User-Centric Computing for Human-Computer Interaction. So, in the previous week we have gone through the GOMS family of models, namely the KLM and the CMN-GOMS, the Fitts law and the hick Hyman law. If you may recollect at the beginning of the week we mentioned that these are classical models. These were developed long ago, some were not even developed for user centric systems as we have mentioned, but found wide usage in the design of user centric systems.

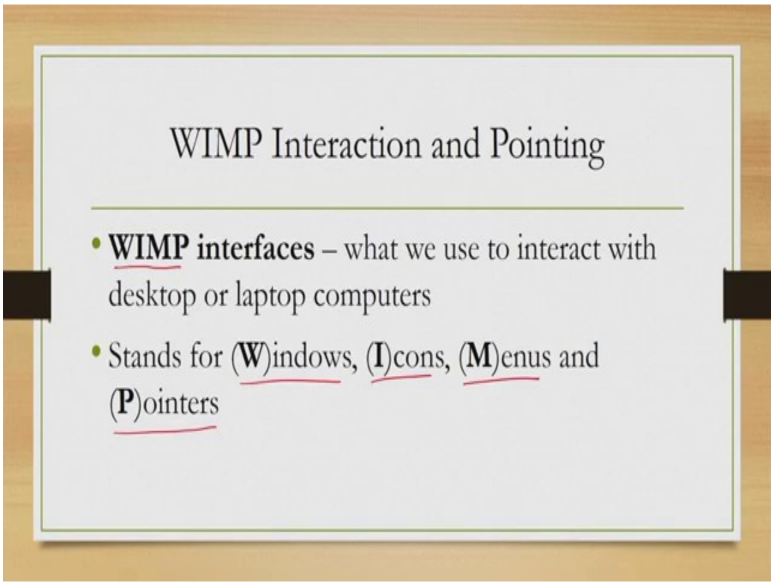
However, all of them were developed long ago, primarily to cater to the early generation of user centric systems and interfaces. In this week, we are going to discuss few models which are of much recent origin and are primarily applicable to contemporary systems and interfaces. We will start with a model which we can use for 2D pointing.

(Refer Slide Time: 01:48)



As you know we all use GUIs or Graphical User Interfaces heavily in our everyday interaction with computer. Now, this interaction with GUIs involve lots of pointing tasks. So, what we need is to design and analyze GUIs we need a model for pointing tasks on such interfaces. So, you may wonder whatever models we have discussed before whether they are applicable or not. In order to understand the question further, let us try to understand the nature of interaction with the GUI.

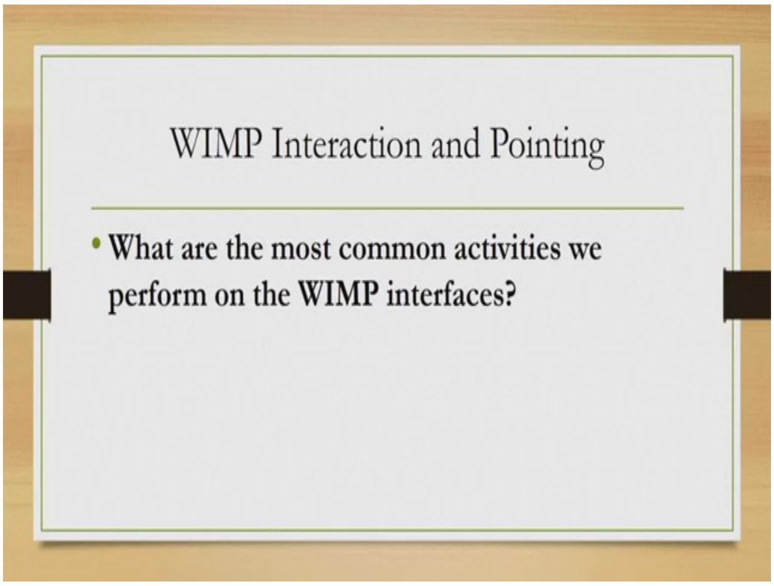
(Refer Slide Time: 02:18)



So, when we are talking of GUI what we are referring to primarily is called WIMP interfaces. This is the, this is a very popular term used to refer to say GUIs. It is called the WIMP interfaces where WIMP is an acronym which stands for windows, icons, menus and pointers. So, when we are talking of interacting with the GUI essentially what we are referring to is interacting with a WIMP interface.

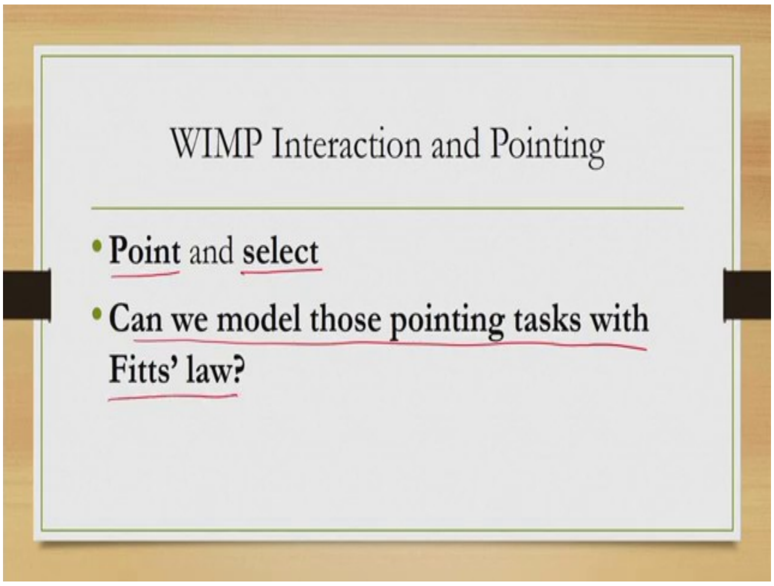
Now, the characteristics of that interaction is that you will be presented with windows, the windows will have icons which are essentially metaphors for certain elements. Then there will be menu based interaction, so there will be some menus through which we will interact and all these interactions will take place through the use of a pointing element or pointers. So, to interact we use pointers.

(Refer Slide Time: 03:11)



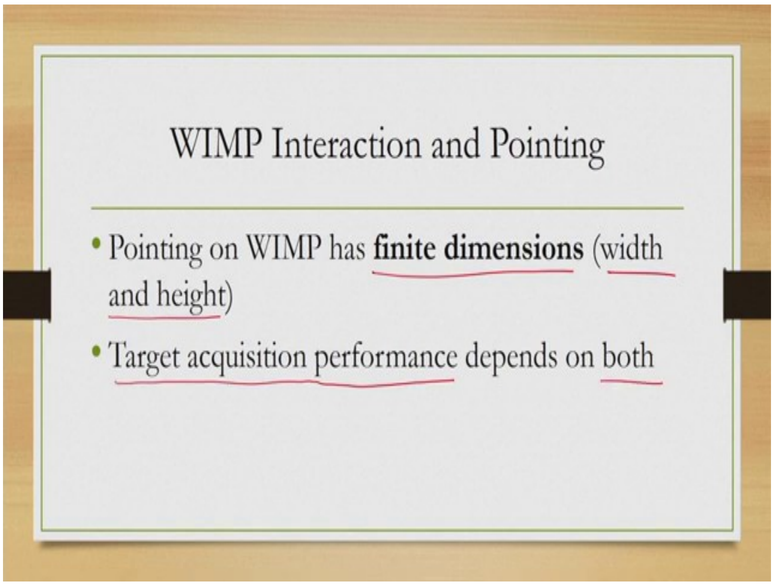
Now, with these type of interfaces what are the common activities we do? The answer is obvious we point and select.

(Refer Slide Time: 03:22)



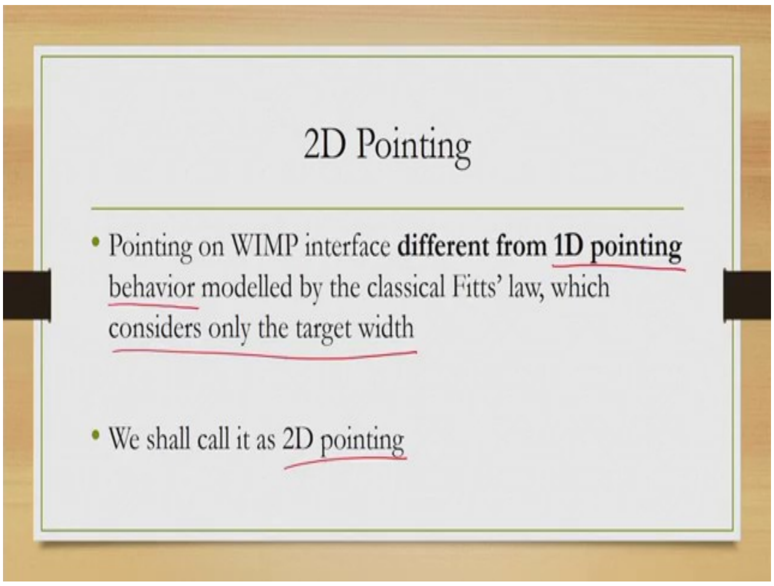
So, the most common activity that we can talk off in the context of a WIMP interface is pointing and selection. Now, the question is can we use the Fitts law to model those activities. So, can we model those pointing tasks with the Fitts law? Let us try to answer this question. There is one difference which we should keep in mind when we are discussing about pointing in WIMP interface and pointing in the context of the Fitts law that we have learned in the previous lecture.

(Refer Slide Time: 03:55)



So, when we are talking of pointing on WIMP interfaces what we are referring to is a pointing task that is done in a finite dimension that is the width and height of the entire reference frame is finite. And it is quite intuitive that when we are talking of target acquisition the performance of target acquisition depends both on the width and height of the dimension.

(Refer Slide Time: 04:28)



So, that is not necessarily what is considered in the context of Fitts law which can be more appropriately called 1D Fitts law or 1D pointing behavior that is modeled by the law which does not consider the finite dimension instead it considers only one element of the dimension that is the width. So, when we are talking of pointing in a finite area or pointing that involves finite dimension, so we have two elements to consider width and height.

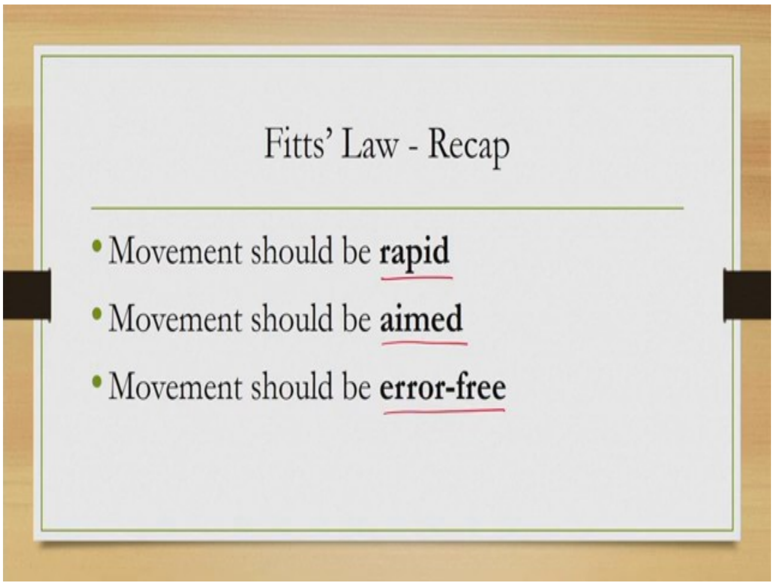
The classical Fitts law that we have learned in the previous week in one of the previous lectures, does not consider both instead it considers only one element that is the width. So, when we are talking of pointing having both elements we will call it 2D pointing to differentiate it with the notion of 1D pointing behavior that is modeled by the classical Fitts law.

Before we try to understand the model for 2D pointing let us have a quick recap on the Fitts law, what it assumes, what it does, then it will be easier to put the discussion into proper perspective, proper context.

(Refer Slide Time: 05:50)

So, if you may recollect in the Fitts law we made three assumptions. These three are very crucial assumptions.

(Refer Slide Time: 05:58)

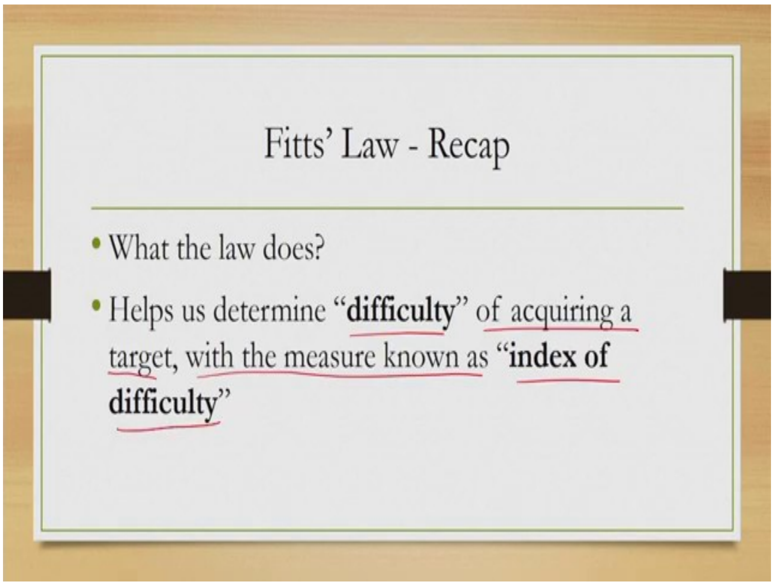


That is the movement should be rapid, the movements should be aimed, and the movements would be error free. So, these assumptions were made to avoid constellation of cognitive activities that may happen if we violate these assumptions.

(Refer Slide Time: 06:18)

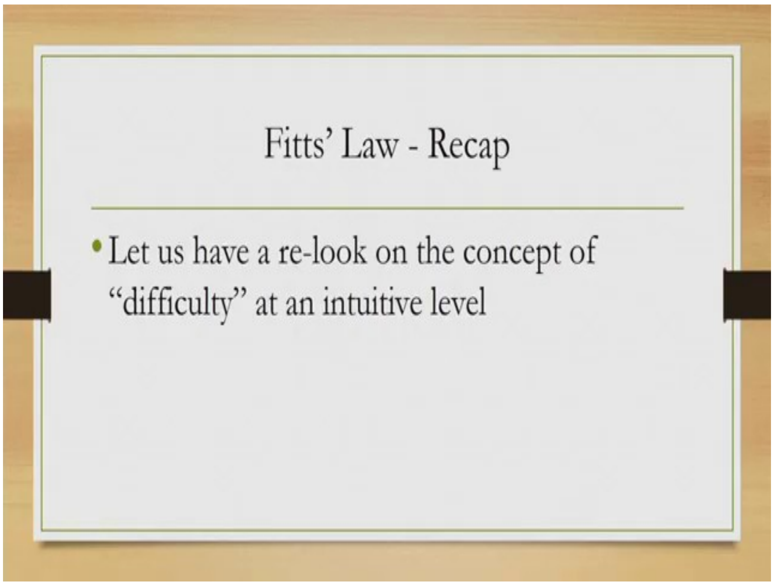
Now, the error free assumptions of course is something which we can deal with separately. As we have discussed in the speed accuracy trade off part of the lecture that if we are violating the assumption of error free condition then we can actually work width effective target with rather than actual target width and this is applicable when the miss rate is more than 4 percent. However, other two assumptions we must have to adhere to be able to apply Fitts law in modeling pointing behavior.

(Refer Slide Time: 06:56)



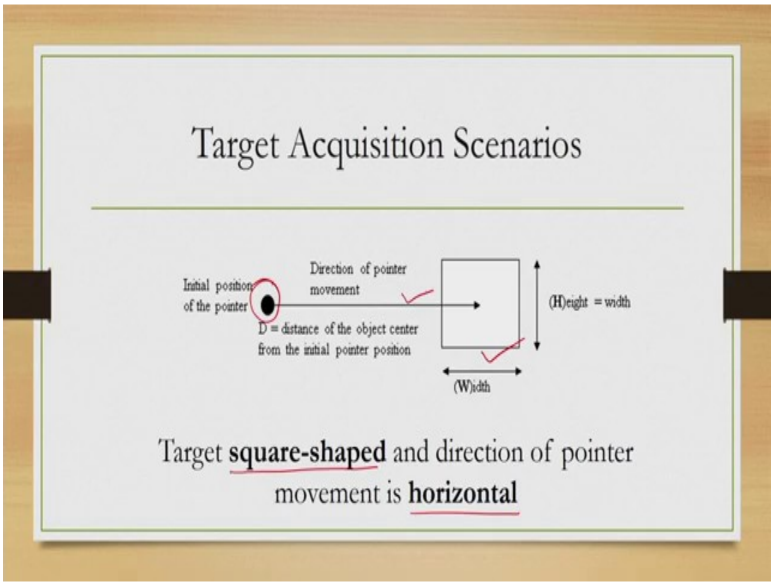
So, what the law does? It helps us determine difficulty of acquiring a target with the major known as the index of difficulty that is in summary what we have learned earlier that under those assumptions we have a major called index of difficulty which can be used to measure the level of difficulty in acquiring a target.

(Refer Slide Time: 07:24)



So, what is the problem with this particular approach to modeling pointing behavior? Now, let us have a re-look at the concept of difficulty at an intuitive level.

(Refer Slide Time: 07:35)



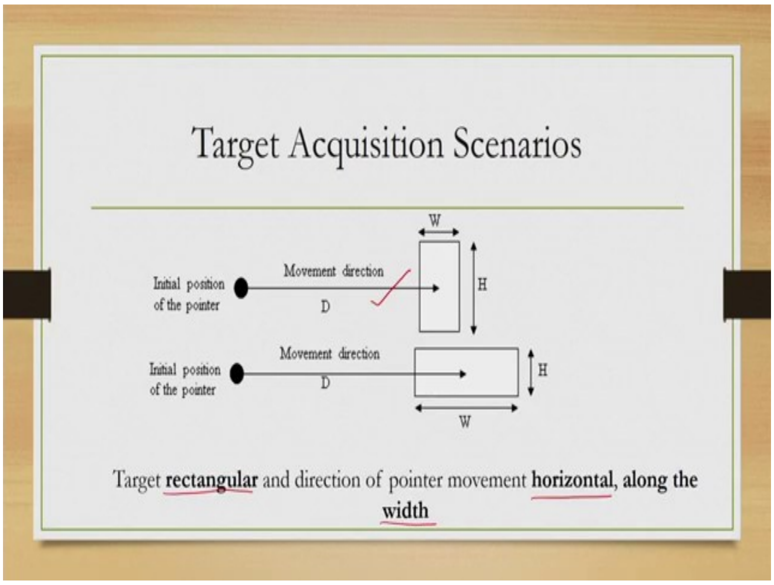
Consider this scenario, a target acquisition scenario. Here this is the initial position of the pointer which can be your finger, hand or most pointer and this square represents the target. This is a square region and this is the distance of the object center from the initial pointer position, the arrow indicates. This arrow indicates the direction of pointer movement.

So, here the target is square shaped and the direction of movement is horizontal. So, in this case we can apply Fitts law. This case actually matches very closely with the behavior that is assumed in applying Fitts law. So, if on a WIMP interface we are performing some activity, that matches with these type of situation where we are aiming to acquire a square shaped target by moving the cursor in a perfectly horizontal position or perfectly vertical position then we may apply the classical Fitts law. However, that is not necessarily the situation always.

(Refer Slide Time: 08:48)

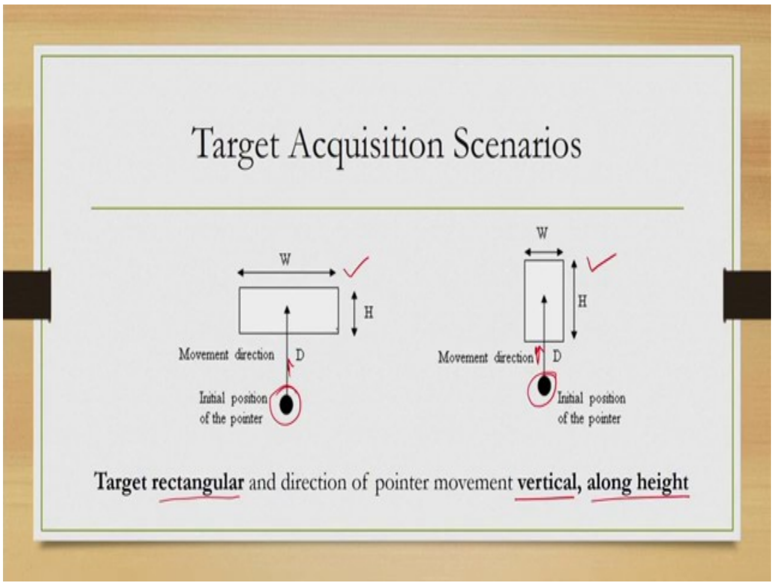
Let us see some more practical scenarios that we encounter while trying to select a target or trying to point to an object in a way which may not be similar to the one we have just seen.

(Refer Slide Time: 09:03)



Considered these two cases. Here we have two types of targets, two rectangular targets. So, both the targets are rectangular and in both the cases the movement direction is also horizontal, however the orientation of the targets are different. See in one case the width is less than height and the direction is along the width in other case the width is more than height and the direction is along the way again and horizontal. So, the direction of movement is horizontal along the width and the target is rectangular, but the orientation of the object is different in each case.

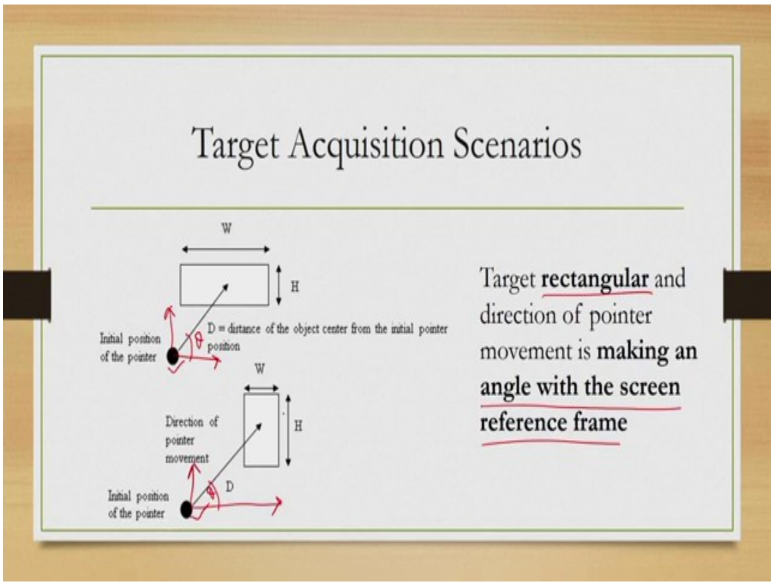
(Refer Slide Time: 09:42)



This is the another situation. Here the movement direction is vertical as you can see. So, this is the our initial position in both the cases. The direction of movement is vertical shown by this arrow and in this case also both the targets are rectangular; however, the orientation is different.

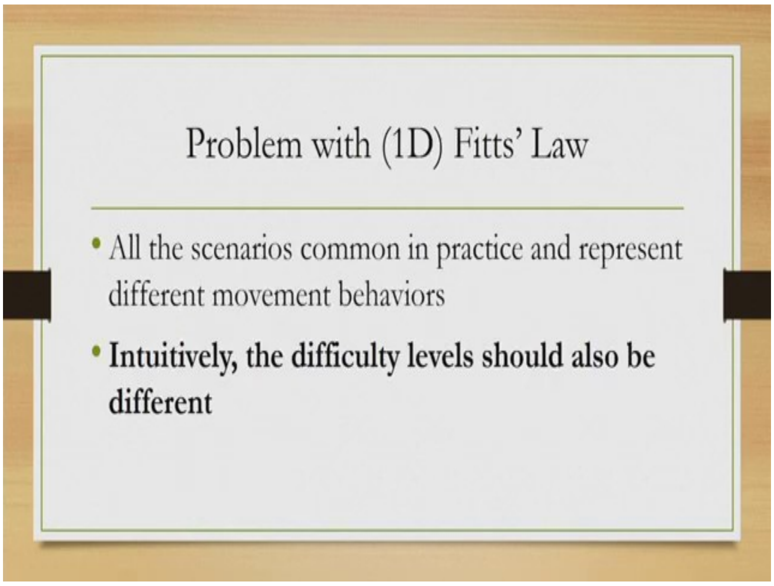
So, in one case the width is more than height, so height is less than width in the first case here and the direction of movement is along the height, in the second case here the height is more than width and the direction is along the height. So, the movement direction is vertical along height and the objects are rectangular, but the relationship between width and height is different in each case.

(Refer Slide Time: 10:26)



Let us see a third scenario. In this case as you can see this is the initial position, now the direction of movement is not vertical or horizontal instead it is making some angle with the screen reference frame. Both the targets are rectangular, but the direction is making an angle with the screen reference frame. So, if we assume if the reference frame is like this then this is there is an angle in both the cases. So, that is one difference. Other difference is the movement is not along any of the width or height unlike in the previous situations. All these three scenarios that we have just seen are fairly common in WIMP interactions.

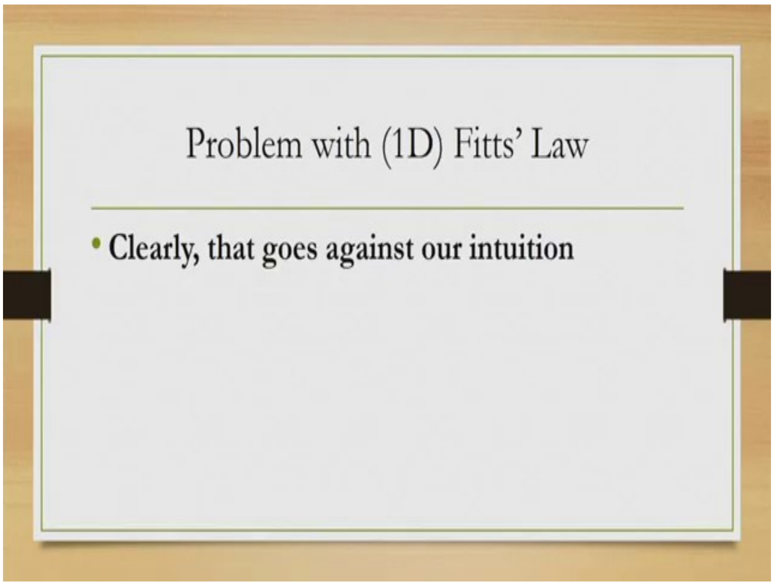
(Refer Slide Time: 11:12)



Now, if we apply Fitts law to model the difficulty level in these tasks that are shown in the scenarios, what do you expect? Ideally, with each setting we should expect different levels of difficulty different degree of difficulty different values for the major of difficulty.

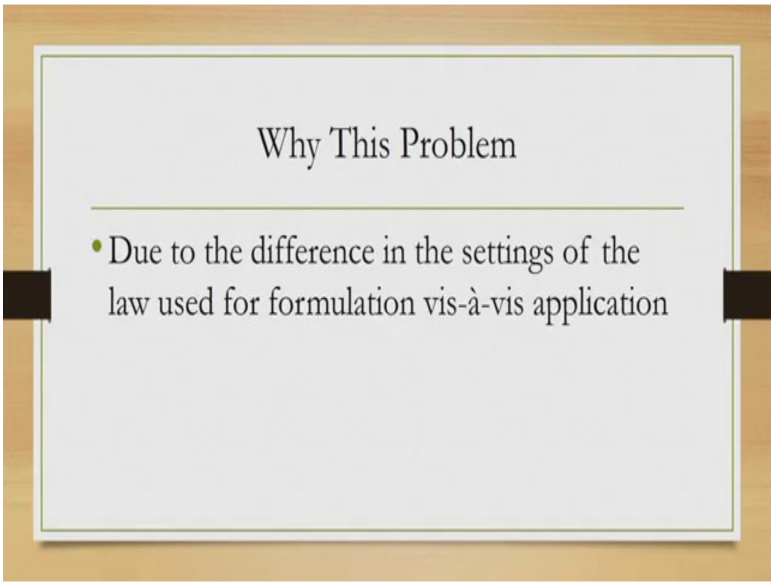
However, if the parameter remains the same, distance and width remains the same then the classical Fitts law will give you the same value, irrespective of the direction irrespective of the alignment of the movement direction with the width or height which obviously is counter intuitive. This is against our intuition.

(Refer Slide Time: 12:02)



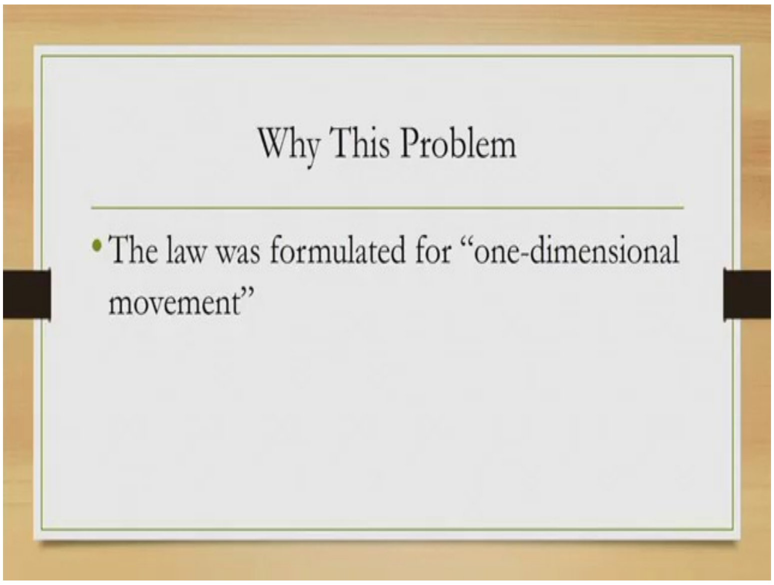
So, why this problem is happening? It is happening because we are not considering the setting that is involved in modeling the behavior.

(Refer Slide Time: 12:15)



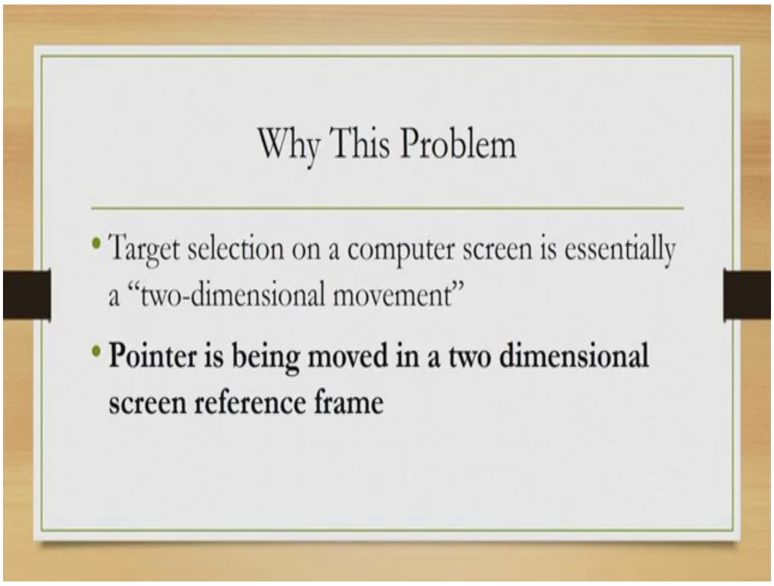
So, the setting that was used to model pointing behavior using Fitts law is different than the setting we have just seen in those situations where we want to apply the Fitts law in its classical form.

(Refer Slide Time: 12:31)



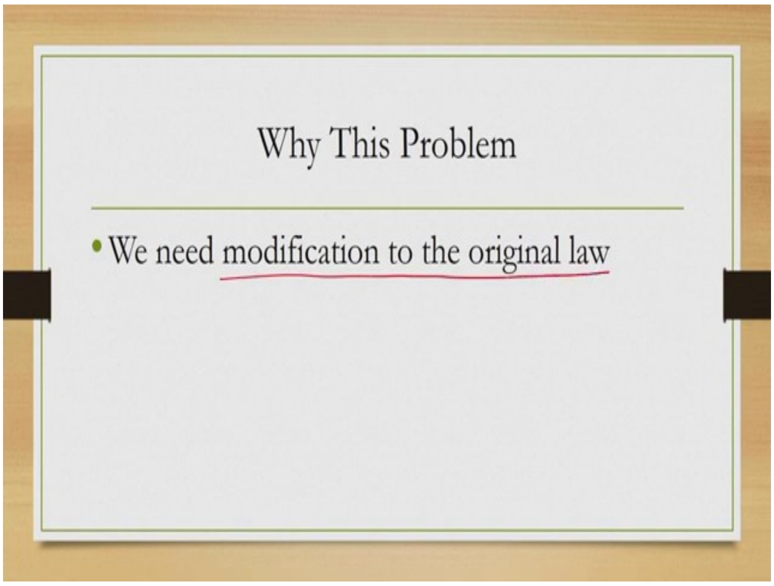
The law was formulated for one-dimensional movement, the first situation that we have seen, but we are trying to apply it for two-dimensional moments, in a two-dimensional screen reference frame which is not the right thing to do.

(Refer Slide Time: 12:37)



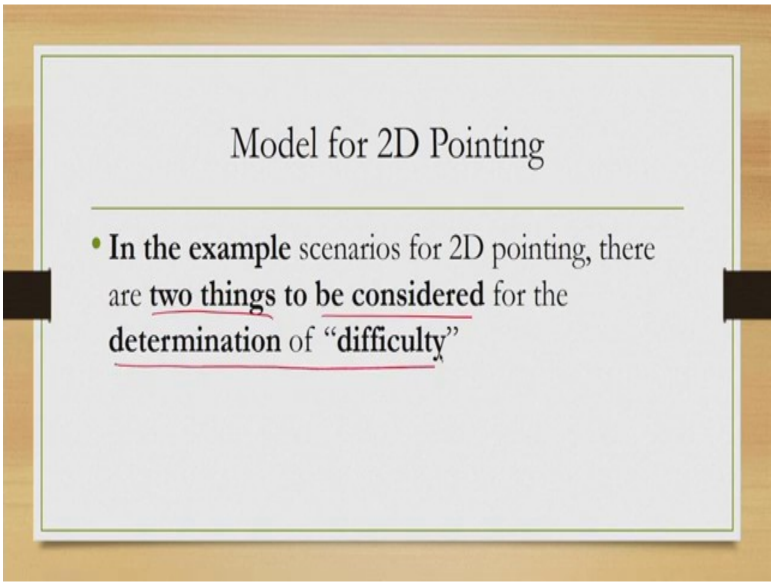
So, what we need to do is a modification in the original originals.

(Refer Slide Time: 12:49)



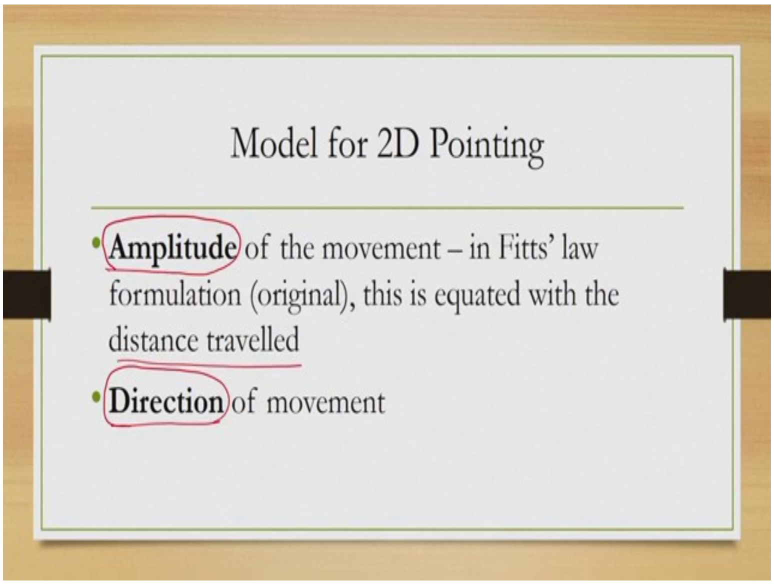
So, we want to have some modified law and that is the objective of this lecture to learn about a law that is applicable for 2D pointing tasks.

(Refer Slide Time: 13:05)



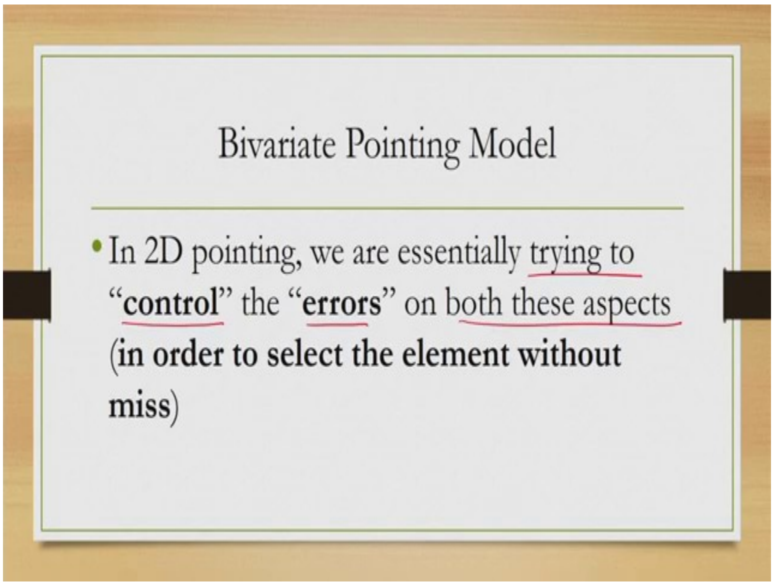
So, in the example if you may have noticed there are two things to be considered for determination of difficulty or the level of difficulty. So, we need to consider two things. What are those two things?

(Refer Slide Time: 13:19)



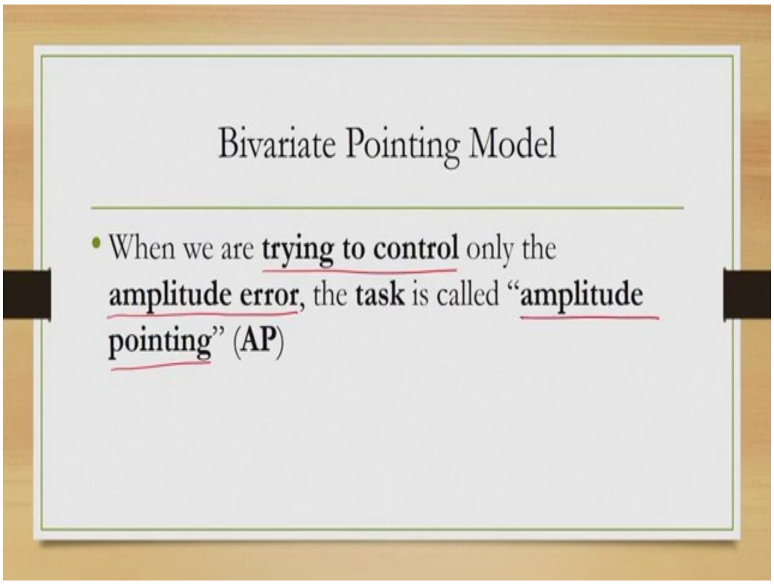
First one is amplitude that is the distance that needs to be traveled from the current position to the target position and the second thing is direction in which direction this distance is being traveled. So, both these things are important in the context of modeling 2D pointing behavior.

(Refer Slide Time: 13:43)



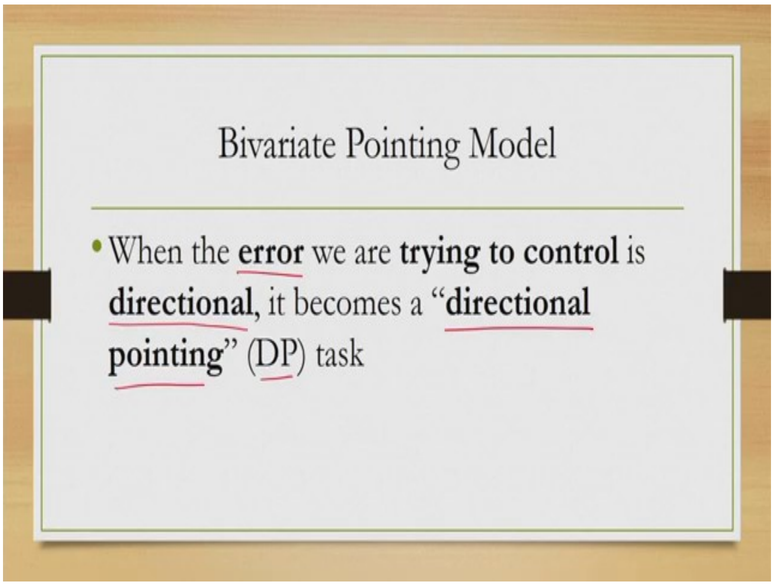
So, what we are trying to do in a 2D pointing task? What we are trying to do is trying to control the errors on both these aspects. So, we are trying to control errors while traveling a distance and we are trying to control errors while traveling to the distance along a particular trajectory in a specific direction. So, our objective is to basically control errors on both these aspects.

(Refer Slide Time: 14:17)



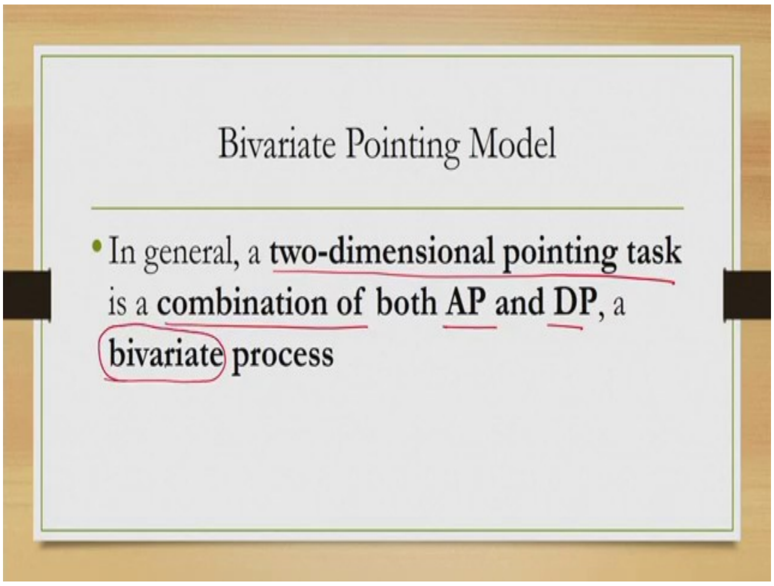
Now, there are terms attached to these concepts. When we are trying to control error for the amplitude or we are trying to control the amplitude error the task is called amplitude pointing. So, when your objective is to basically control error by considering only the distance irrespective of the direction then that is known as amplitude pointing or AP in its short form.

(Refer Slide Time: 14:43)



So, when we are trying to control the error that may come due to direction or the directional error, we call it directional pointing or DP. So, in a 2D pointing tasks then we are actually having two sub tasks amplitude pointing and directional pointing. And our objective is to perform these, both these sub tasks in order to perform the overall tasks without missing the target.

(Refer Slide Time: 15:13)

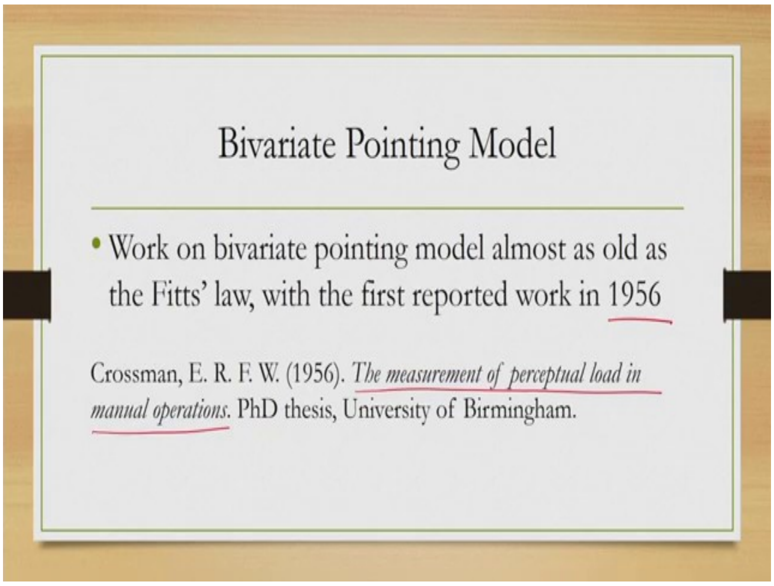


So, what we can say is that a pointing tasks is a combination of AP and DP. So, there are two variables, so it is a bivariate process.

(Refer Slide Time: 15:27)

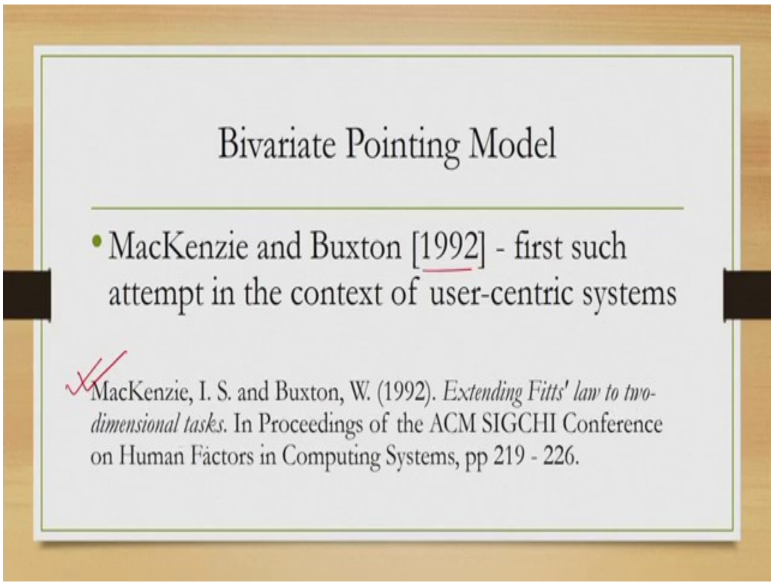
So, what do we need is a bivariate model of pointing rather than the univariate model that we have seen in the context of Fitts law in its classical form.

(Refer Slide Time: 15:37)



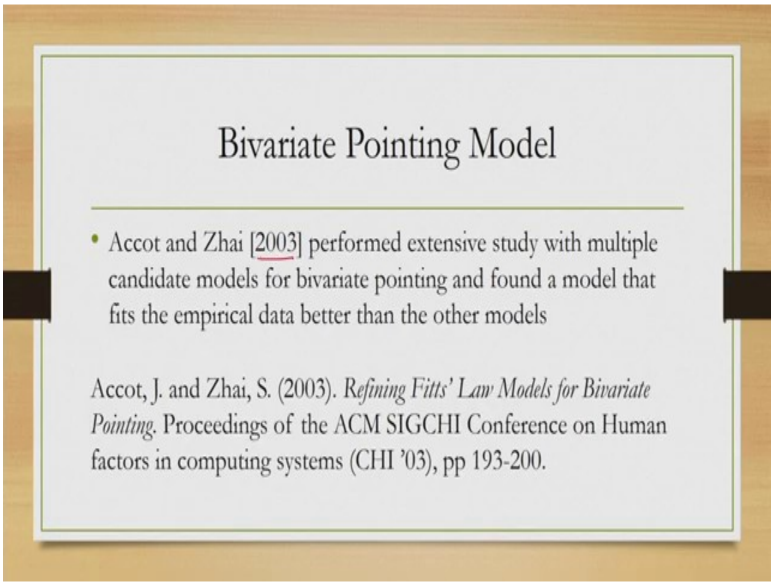
Now, this bivariate consideration to model pointing tasks is not new. Actually the idea came long ago. In fact, as soon as the classical model was proposed almost at around the same time these bivariate models was also reported in 1956 in this PhD thesis, it can be found the measurement of perceptual load in manual operations. But note that the objective was not to come up with a model for WIMP interactions, instead it was taken up as a general modeling activity. In the context of user centric system design bivariate model for 2D pointing has found wide interest and the work started long ago.

(Refer Slide Time: 16:27)



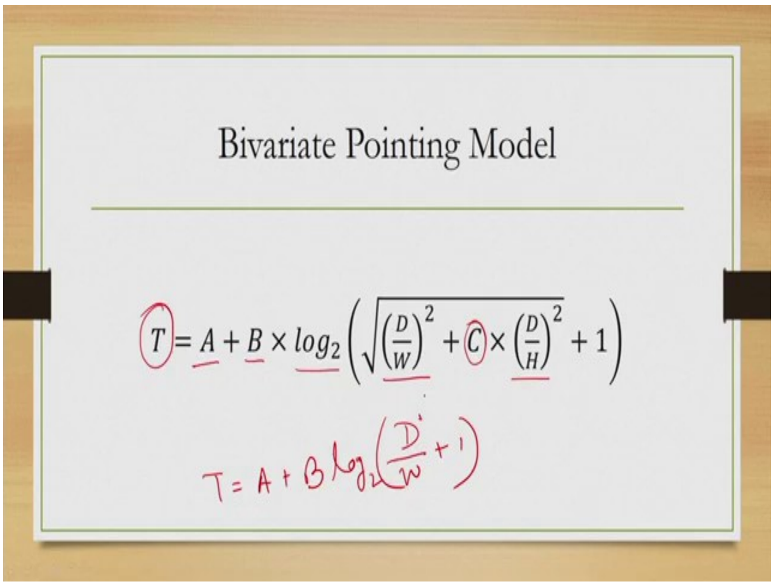
In 1992, MacKenzie and Buxton first reported such work in this article. If you are interested you can go through these article extending Fitts law to two-dimensional tasks which appeared in the proceedings of SIGCHI.

(Refer Slide Time: 16:49)



In 2003, Accot and Zhai in their paper refining Fitts law model for bivariate pointing reported a model after performing extensive study with multiple candidate models. So, they have come up with one model which fits the empirical data much better than other models as per their claim and we are going to have a look at this model.

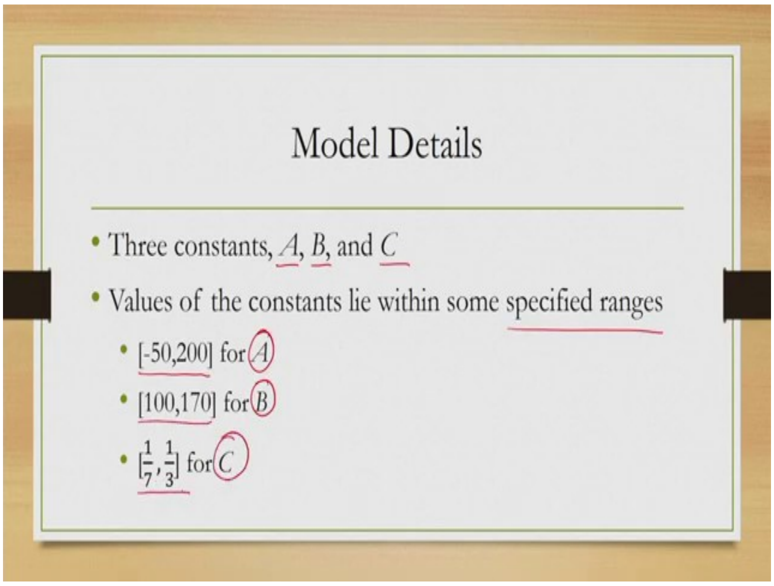
(Refer Slide Time: 17:14)



So, the model looks something like this. Here T is the movement time as usual, A and B are constants and there is this logarithmic time as in the classical Fitts law. However, unlike in the classical Fitts law where we have only an expression like (D/W+1) that is the classical Fitts law: $T = A + B.\log_2(D/W+1)$.

Now, you compare this with the bivariate pointing model here the logarithmic term is much more complicated it contains two quantities D/W and D/H, another constant C and these quantities these ratios are squared and overall square root is taken. So, it is a much more complicated a much more detailed model than the classical Fitts law which is much simpler.

(Refer Slide Time: 18:17)

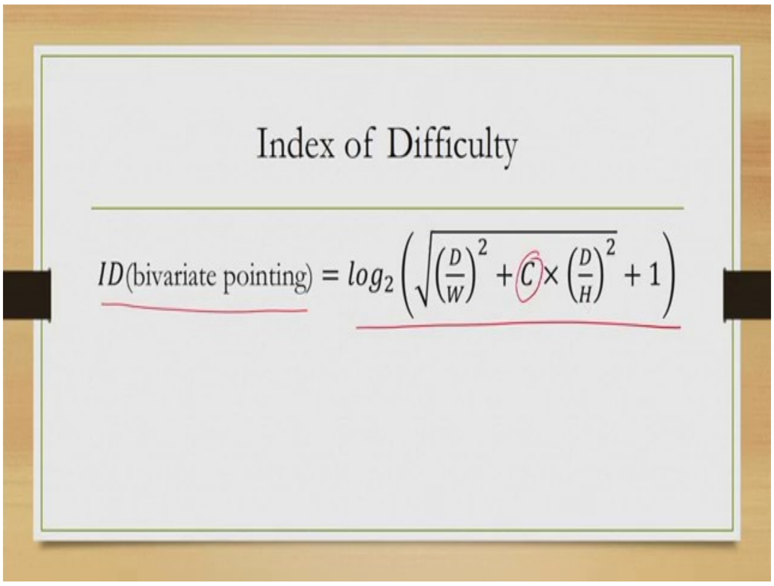


Now, these three constants A, B and C have some specified ranges as found out by the authors of the paper that is Accot and Zhai. So, what they reported is that the constant values should belong to some specific ranges for A the ranges between -50 to 200, for B it is between 100 to 170 and for C should be 1/7 and 1/3. So, with this we can actually capture the behavior the motor behavior in 2D pointing. Unlike, in the case of Fitts law which helps us to capture motor behavior in 1D pointing.

Now, here let me point out that in the discussion on Fitts law we discussed one example in the context of WIMP interaction that is selection of a file menu item. There, I mentioned that although we are discussing it in the context of classical Fitts law it is actually a 2D pointing task and this task would be modeled with a 2D pointing model.

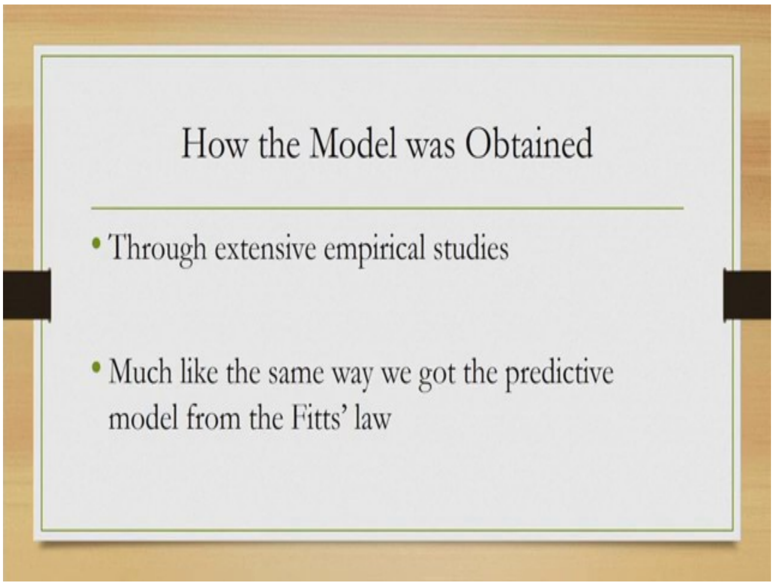
So, we can use the model that we have just now learned, the 2D pointing model or the bivariate pointing model, to more accurately model the example task that we have introduced in the discussion on the classical Fitts law in the last week.

(Refer Slide Time: 19:44)



So, if we compare the Fitts law and the bivariate pointing model, we can see that the index of difficulty for bivariate pointing can be represented with this part with this logarithmic term and as I have already explained D is the distance, W and H are width and height of the target object, C is the constant which has a specified range.

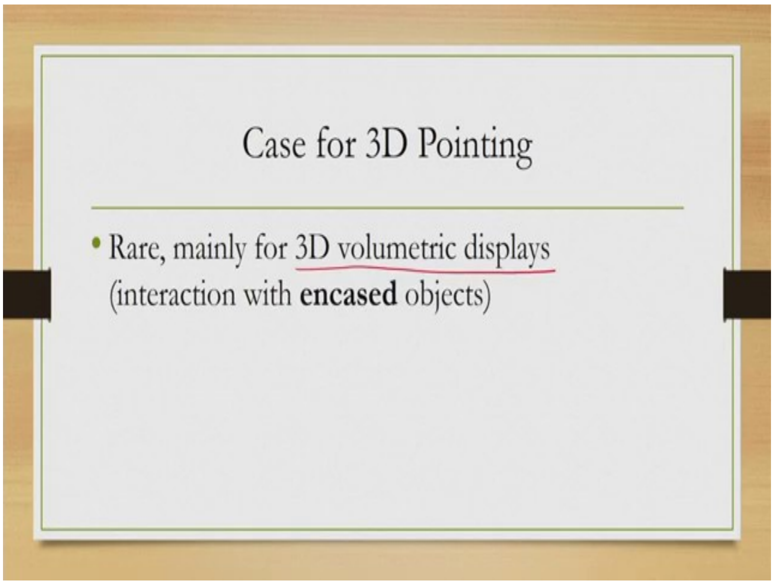
(Refer Slide Time: 20:09)



And how we can obtain these values constant values? So, we mentioned some specified range, but actual value how to obtain essentially through empirical studies as we have discussed in earlier lectures on Fitts law as well as hick Hymans law. Now, that was all about 2D pointing.

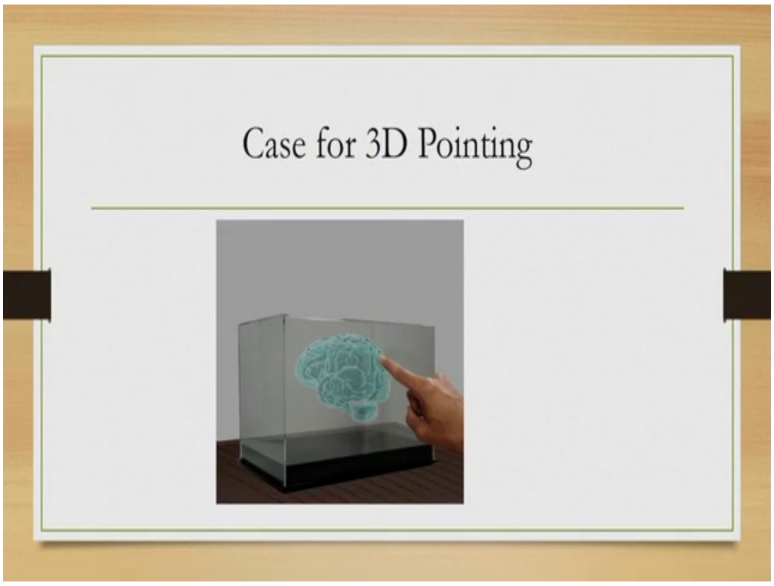
So, to recollect what we have learned is that 2D pointing task which is frequently performed on most or all WIMP interfaces are different than the pointing behavior that is modeled with the classical Fitts law. So, there are differences accordingly. We required a new model a modified model which is a bivariate pointing model which we have introduced in this lecture. Now, that was about pointing in 2D. Along with that there are models to represent our pointing behavior in three-dimensional space. So, we will briefly have a look at one such model.

(Refer Slide Time: 21:16)



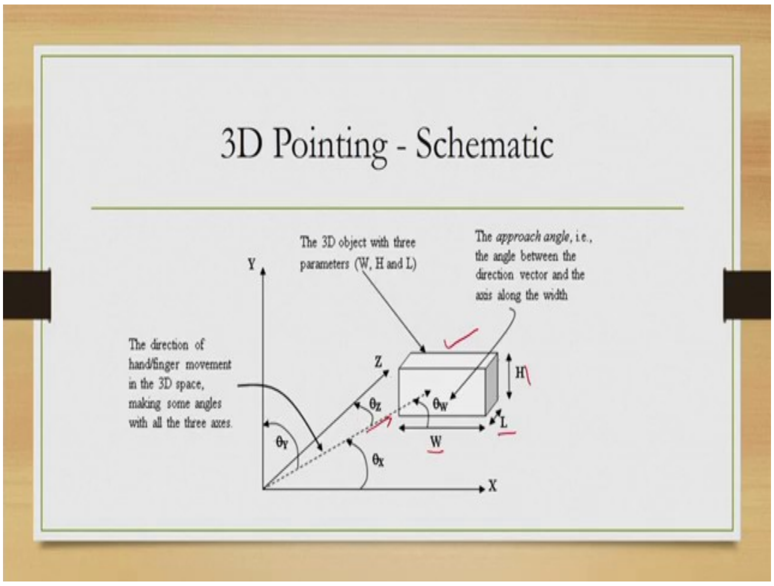
However, it may be noted that these 3D pointing activities still in the context of user centric systems is very rare. And there are certain interfaces which are occasionally called 3D volumetric displays where people tried to make those displays interactive through pointing and other gestures and the 3D pointing model refers to that type of interaction.

(Refer Slide Time: 21:44)



One example of this 3D volumetric display is shown here. So, there is a casing glass casing, within this there is a volumetric display of human brain and you can actually perform gestures to rotate the object to bring some other side to your front as well as you can point to specific locations on the object. So, here in order to point and select you need to make 3D pointing task and we will discuss one model that represents this type of tasks this type of motor behavior to carry out the tasks.

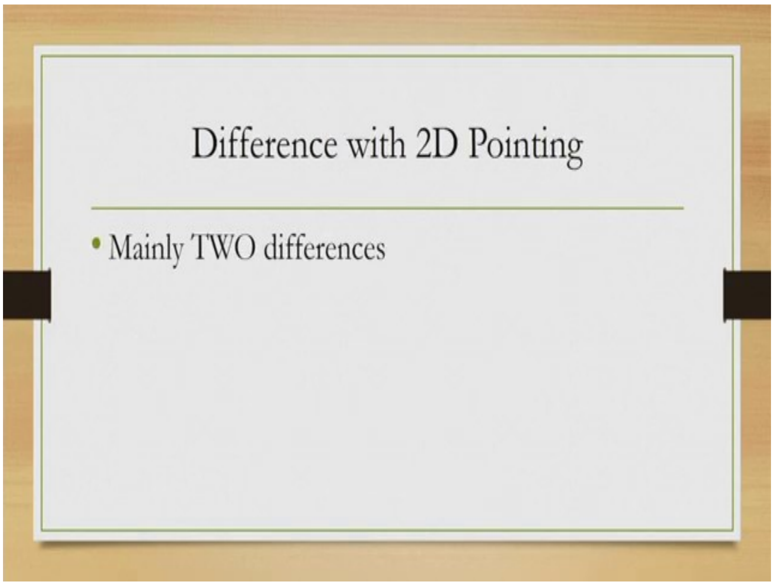
(Refer Slide Time: 22:29)



So, the figures source an illustration of the entire task setting. Here as you can see there is a 3D reference frame represented by the X Y Z axis. This is the 3D object within this frame, now there are several quantities involved. So, the dotted arrow is the direction of pointer movement or direction of finger movement.

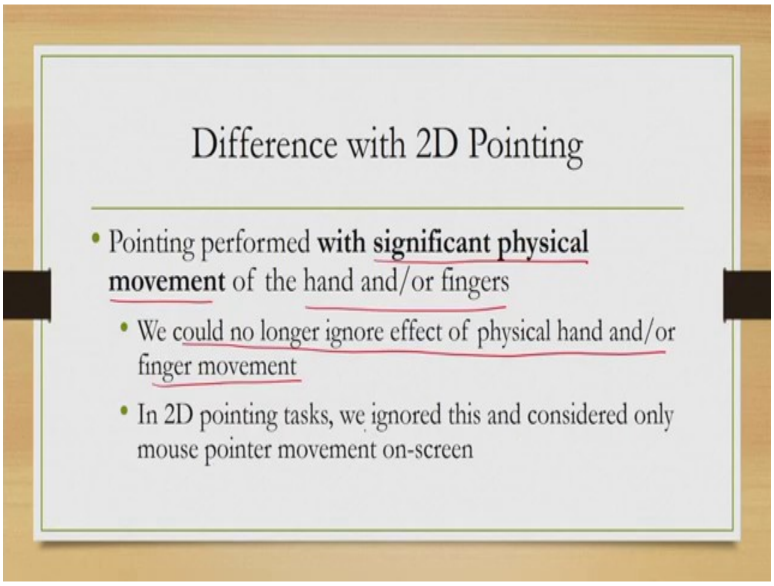
Now, the 3D object has three quantities as is obvious for any 3D object the length, breadth or width and height. Now, there are several angles that this direction of movement makes with different axes. So, one angle is called the approach angle that is the angle it makes with the axes representing the width and there are other angles it makes with the overall or the world reference frame of X Y Z, having the axis X Y Z.

(Refer Slide Time: 23:29)



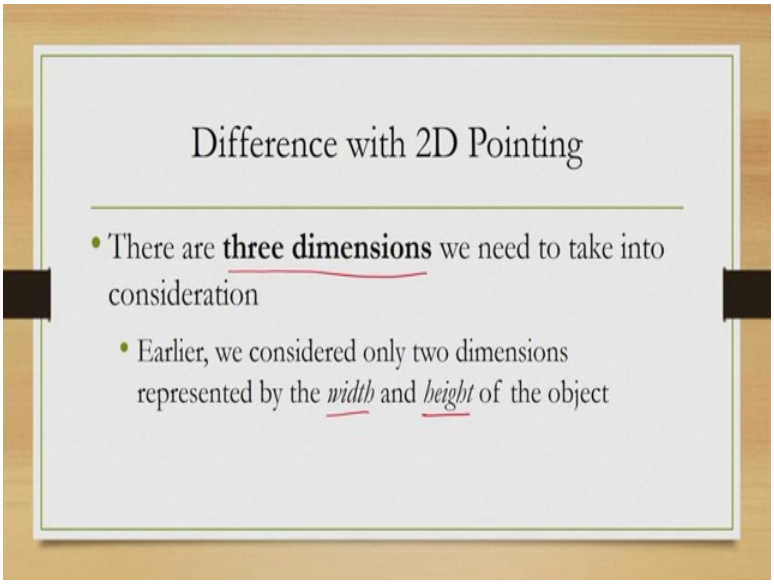
So, there are mainly two differences when we consider 3D pointing with respect to a 2D pointing task. Now, remember earlier when we are discussing 2D pointing task we discuss the differences of 2D pointing width 1D pointing. Now, we are going to discuss the difference of 3D pointing width 2D pointing.

(Refer Slide Time: 23:49)



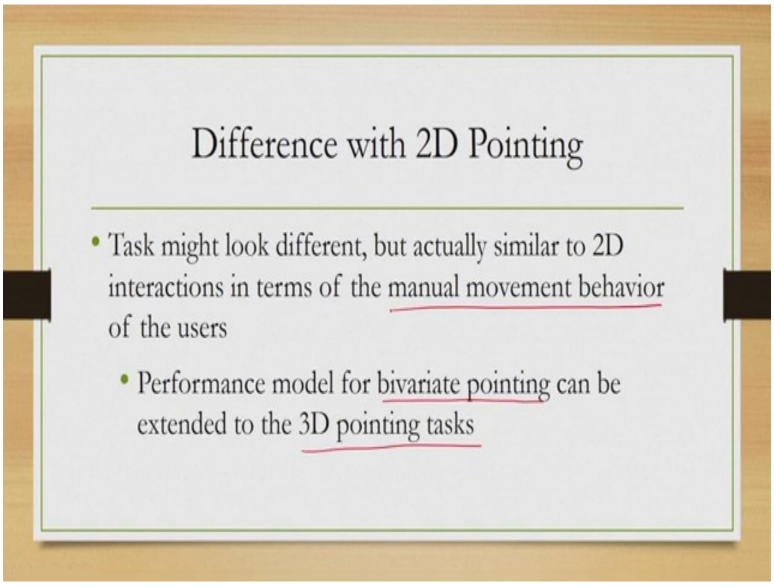
So, there are two differences mainly. First one is now in order to perform the pointing significant physical movement of the hand or finger is involved and we could no longer ignore the effect of the physical hand or finger movement. In case of 2D pointing, we ignored this additional thing, this additional hand or physical movement, we assume that to be negligible.

(Refer Slide Time: 24:17)



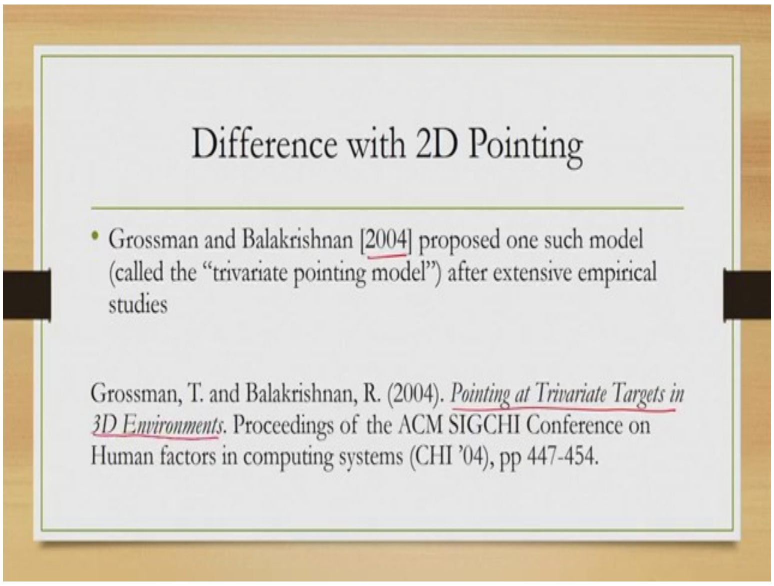
Secondly, there are three-dimensions. In contrast earlier we had two-dimensions width and height and the model was framed accordingly, now we have to deal with three-dimensions as we are dealing with 3D objects. So, these two differences make the 2D pointing and 3D pointing different things.

(Refer Slide Time: 24:36)



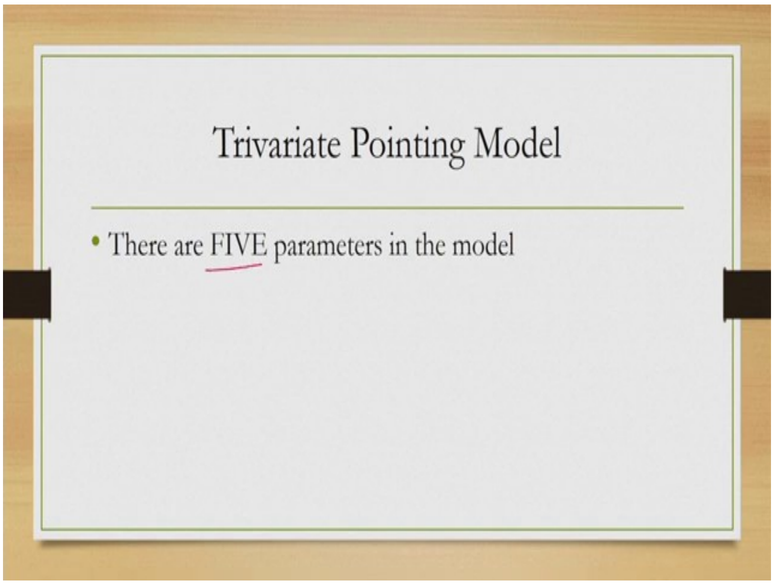
Although, in terms of these two differences the 3D pointing may look totally different from 2D pointing, but that is actually not show. In fact, in terms of the manual movement behavior the 3D pointing is similar to 2D pointing. So, we can actually extend the bivariate pointing model to model the 3D pointing tasks. We will not go into the details of how they are similar, but it is sufficient to know that we can extend the 2D or pointing model or the bivariate pointing model to model 3D pointing tasks based on certain similarities between the two tasks.

(Refer Slide Time: 25:25)



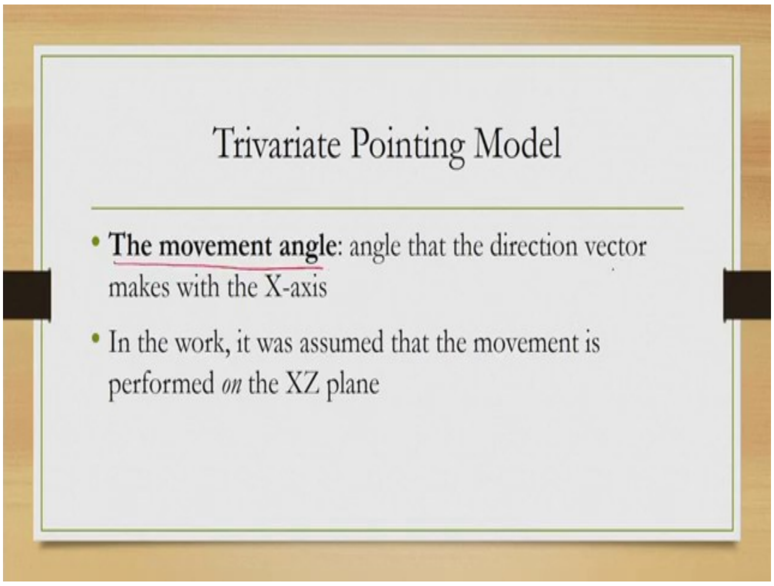
Now, one such model was proposed by Grossman and Balakrishnan in 2004, and their model was reported in this paper pointing at trivariate targets in 3D environments published in the proceedings of CHI 2004.

(Refer Slide Time: 25:46)



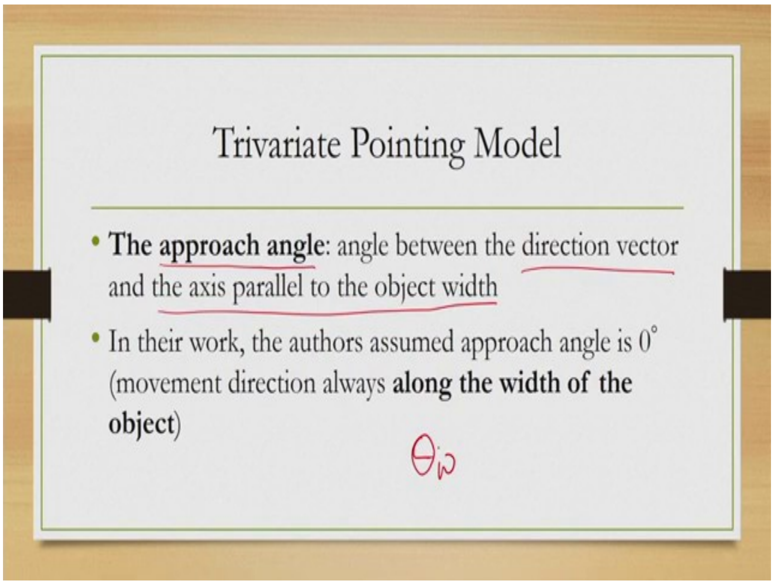
So, in their model there are five parameters. What are these parameters?

(Refer Slide Time: 25:57)



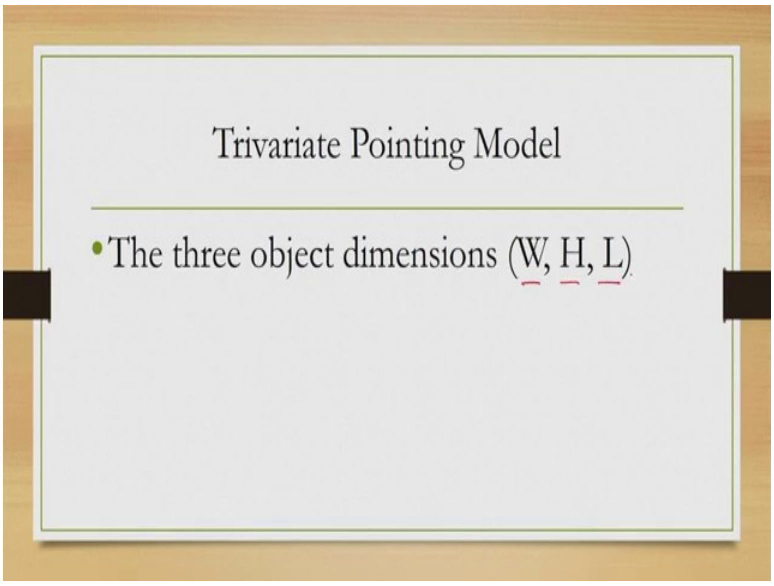
First one is the movement angle this is the angle that the direction vector makes with the X-axis. So, if we go back to the illustration then we will get to see that the movement angle is denoted by theta X. So, the direction that the angle that the direction vector makes with the X axis of the world reference frame is the movement angle. So, that is one of the parameters in the model proposed by Grossman and Balakrishnan.

(Refer Slide Time: 26:34)



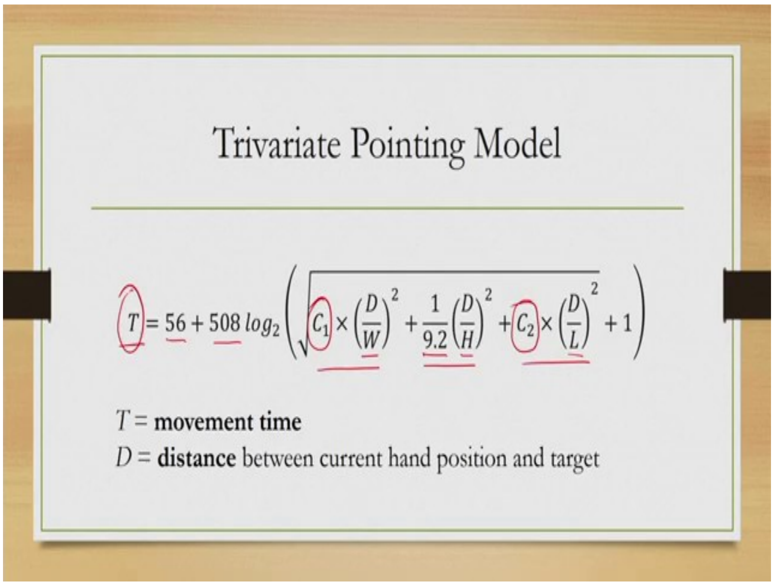
The other parameter is the approach angle. Now, this is the angle between the direction vector and the axis parallel to the object, width. And this we have already pointed out earlier in the schematic which we denoted by theta W. So, in their work the authors have made few assumptions. First of all they assumed that the movement is always performed on the XZ plane and the approach angle is always 0, that means, that movement direction is always along the width of the object. Based on that they proposed certain things that we will discuss.

(Refer Slide Time: 27:26)



So, there are five parameters in the model, one parameter we discussed as the movement angle, one parameter is the approach angle, the other three parameters are the three-dimensions of the object denoted by width, height and length; width or breadth, height and length.

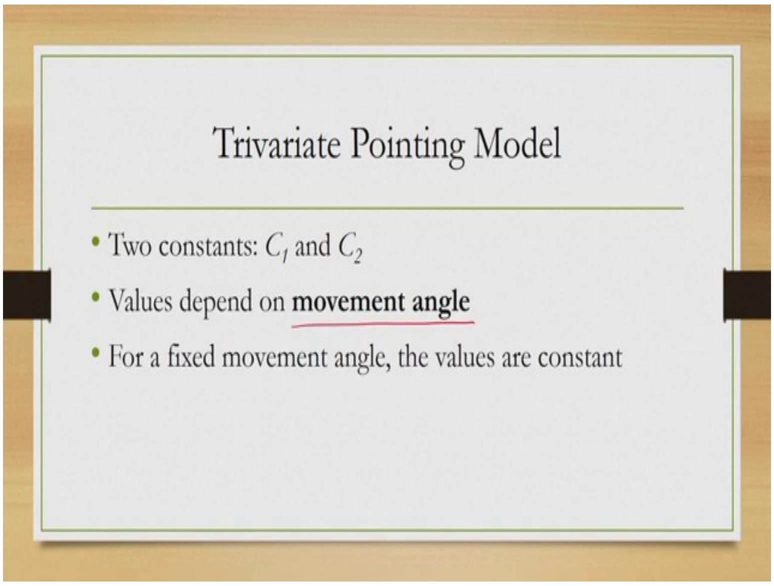
(Refer Slide Time: 27:37)



So, with these five parameters, Grossman and Balakrishnan reported this model with the two assumption that movement is always along the direction of the width that is the approach angle is 0 and it is always in the XZ plane. So, based on these assumptions they proposed a model which is something like this. T is represented, T is expressed as a mathematical equation of this form, where T represents the movement time and D represents the distance between the current hand position and the target.

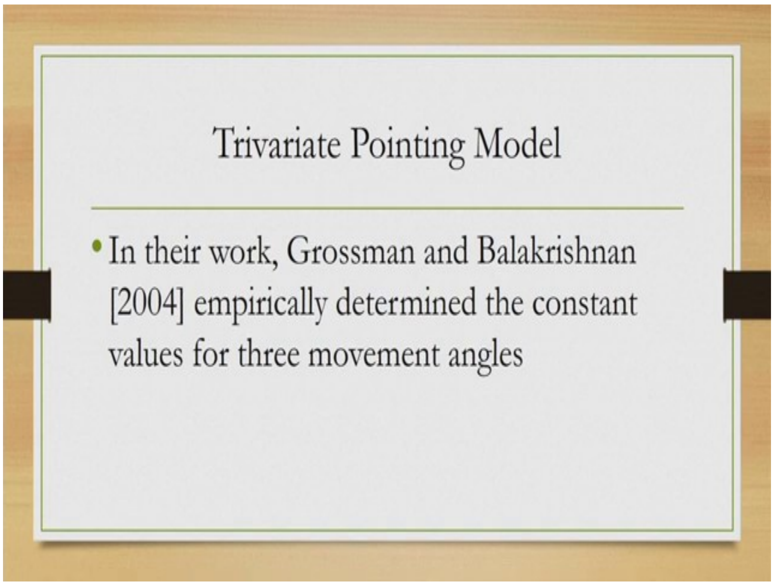
And there are three components now, W is the width, H is the height, L is the length and there are two constants now C1 and C2. So, if we compare it with the 2D pointing model then we can see that one component has been added with associated constant value, otherwise its overall appearance is similar.

(Refer Slide Time: 28:46)



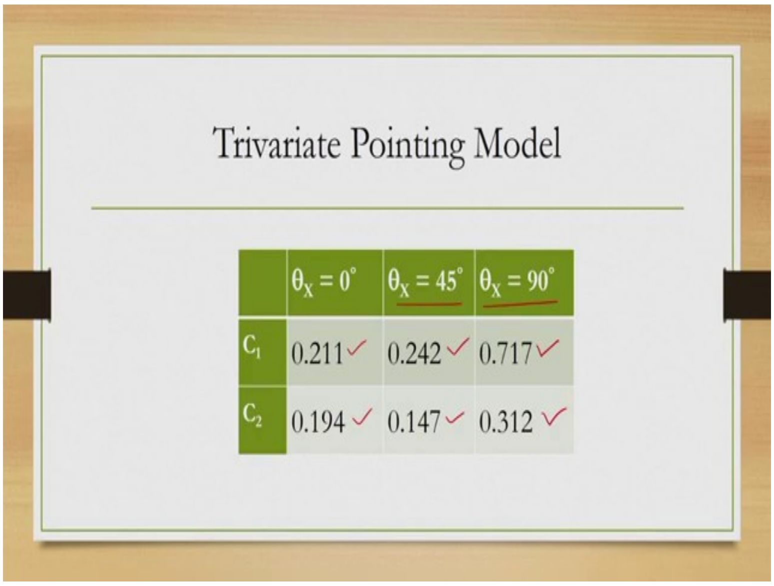
Now, the constants C1 and C2 their values depend on the movement angle or theta X that we have seen earlier. So, for a fixed movement angle the values are constant. So, if we fixed the movement angle then for any tasks that have the same movement angle will have constant values.

(Refer Slide Time: 29:08)



And the authors reported the constant values for various movement angles after extensive empirical studies.

(Refer Slide Time: 29:17)

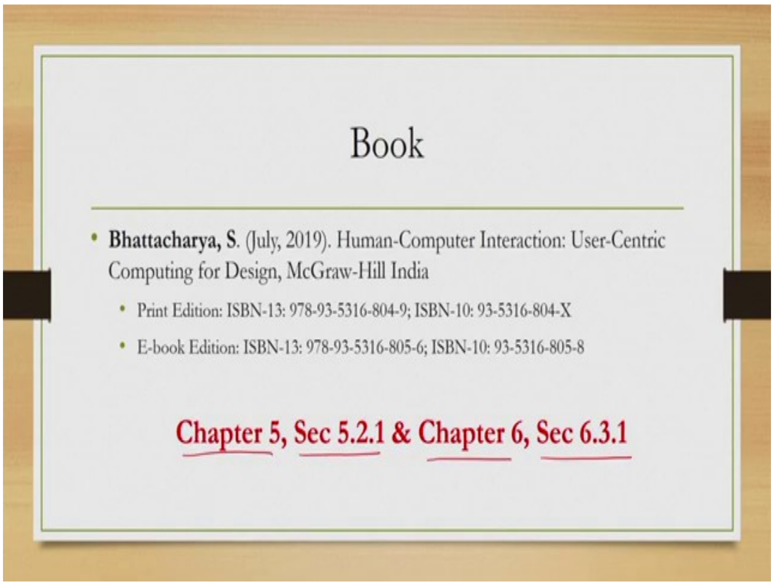


Their results are reported in a tabular form. As you can see when the movement angle is 0 degree the values of C1 and C2 are 0.211 and 0.194 respectively. When the movement angle is 45 degree then the C1 and C2 values are 0.242 and 0.147 respectively. When the movement angle is 90 degree then the constant values are 0.717 and 0.312, respectively

So, if we know the movement angle then we can use these values in the trivariate pointing model is shown here to get the movement time because all other values are already empirically determined. So, this is another predictive model for 3D pointing behavior.

So, we have covered then 1D pointing which is modeled with the classical Fitts law, 2D pointing which is modeled with the bivariate pointing model and 3D pointing modeled with the trivariate pointing model. So, all the materials that I have covered in this as well as the previous lectures that dealt with one day and 2D pointing can be found in this book.

(Refer Slide Time: 30:38)



So, interested readers are advice to refer to chapter 5, section 5.2.1 for 2D pointing; chapter 6, section 6.3.1 for 3D pointing. And the relevant material for 1D pointing that is the classical Fitts law can be found in chapter 4 as I have already mentioned in my previous lecture.

Thank you and goodbye.