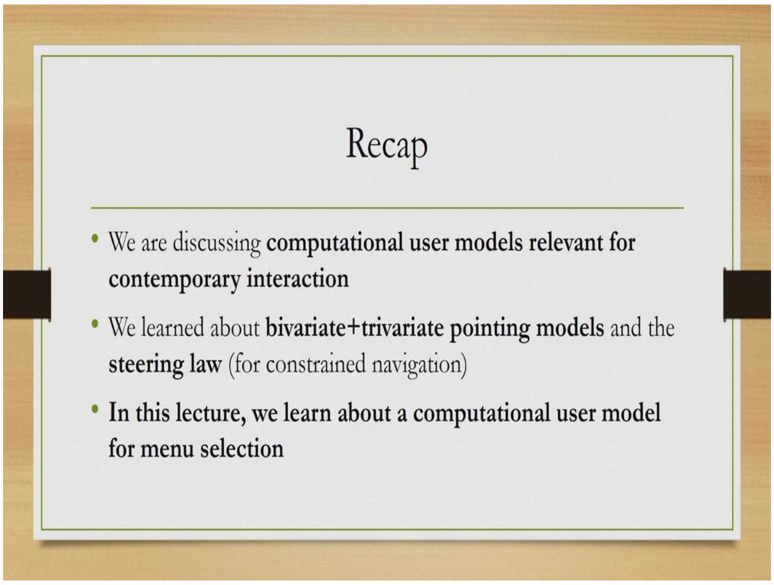
User-Centric Computing for Human-Computer Interaction

Prof. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture – 20
Model for hierarchical menu selection

Hello and welcome to lecture number 20 in the course User Centric Computing for Human Computer Interaction. So, we are discussing about various computational user models that are of recent origin. In the previous week, we have discussed about some classical models Fitts's law, the GOMS family of models, the Hick Hyman law. This week, we are discussing about some models of recent origin which are equally applicable in contemporary interfaces and interactions.

(Refer Slide Time 01:16)



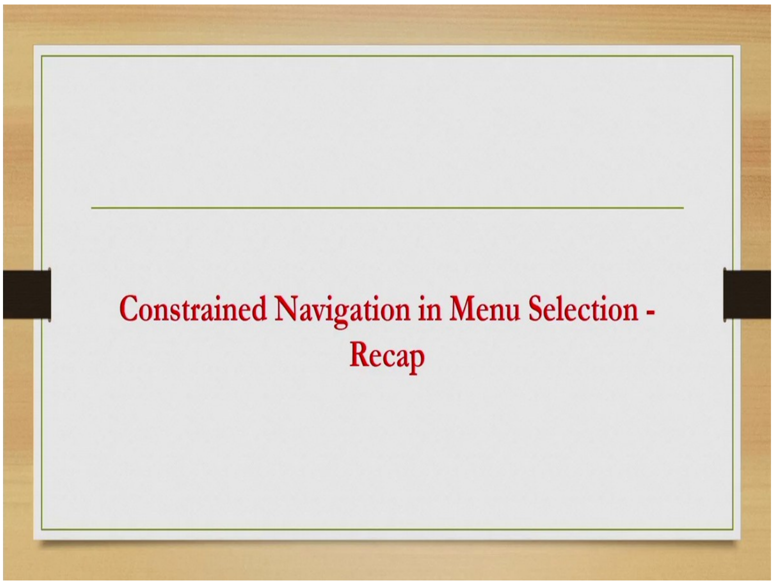
Recap

- We are discussing **computational user models relevant for contemporary interaction**
- We learned about **bivariate+trivariate pointing models** and the **steering law** (for constrained navigation)
- **In this lecture, we learn about a computational user model for menu selection**

In the previous few lectures in this week, we have learned about the bivariate pointing model, the trivariate pointing model and the Steering law; these are related to contemporary interactions. The Steering law that we have learned in the previous lecture refers to constrained navigation. Now, in constrained navigation if you may recollect from the previous lecture we discussed about two examples; one is related to putting a boundary line around a group of objects that is one type of constrained navigational tasks done on interfaces and the other is menu selection.

So, in the previous lecture we did not talk about modeling menu selection behavior; instead what we discussed was the performance modeling of constrained navigation when we are concerned about putting a boundary line around group of objects. Today, we are going to learn about a computational model for menu selection performance.

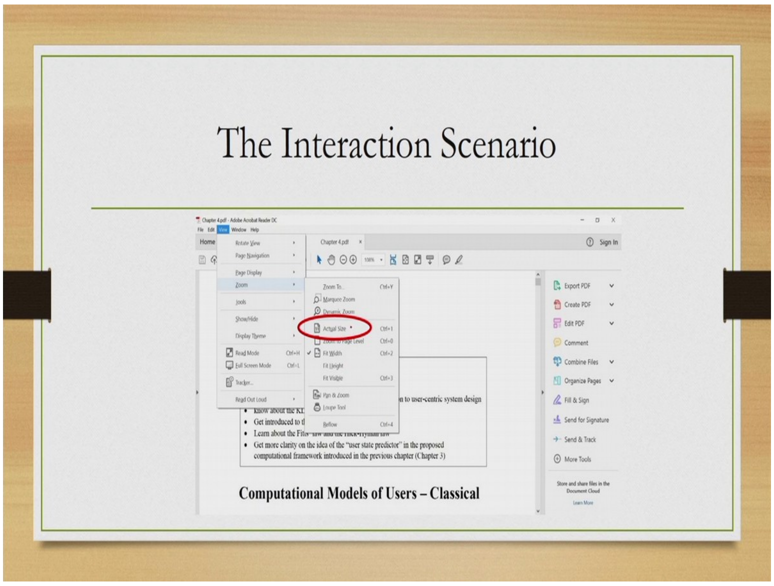
(Refer Slide Time 02:21)



Constrained Navigation in Menu Selection - Recap

Now, before we go into the discussion of the model; let us quickly recap what are the things that we have discussed under menu selection.

(Refer Slide Time 02:29)



The Interaction Scenario

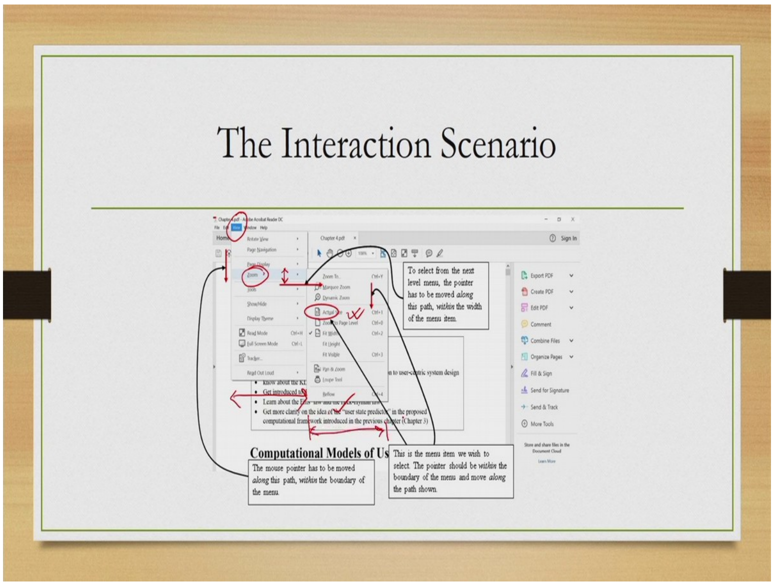
n to user-centric system design

- know about the KL
- Get introduced to t
- Learn about the Fitts' law and the Hick-Hyman law
- Get more clarity on the idea of the "user state predictor" in the proposed computational framework introduced in the previous chapter (Chapter 3)

Computational Models of Users – Classical

So, suppose in this example we are asked to select the actual size sub menu option. So, for that first we have to select one top level menu option view; under this we have to select the zoom option, then this sub level menu appears under the sub level menu we have to select the actual size option. So, the tasks involve multiple tasks; what are the sub tasks?

(Refer Slide Time 03:03)



The Interaction Scenario

To select from the next level menu, the pointer has to be moved *along* this path, *within* the width of the menu item.

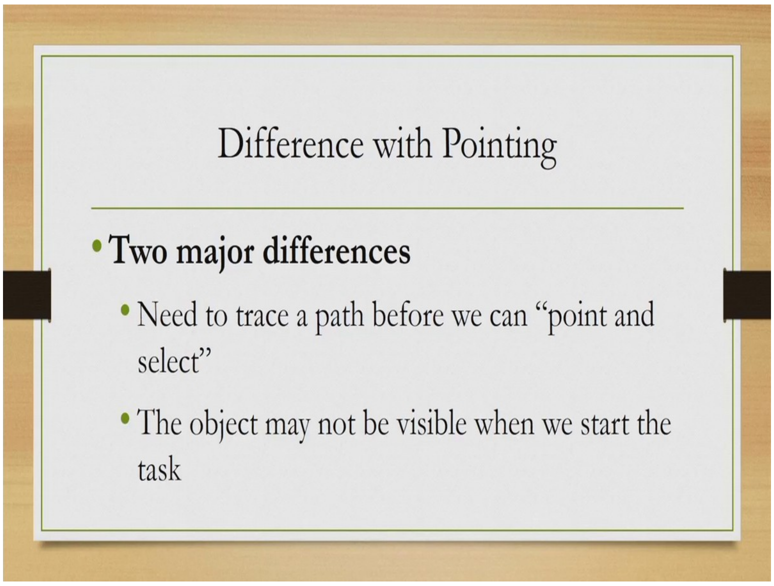
The mouse pointer has to be moved *along* this path, *within* the boundary of the menu.

This is the menu item we wish to select. The pointer should be *within* the boundary of the menu and move *along* the path shown.

First, we perform a vertical downwards movement; here we means we have to take our mouse pointer vertically downwards; after clicking on the blue top level menu option we take it vertically downwards on the drop down menu that appears; till we reach the zoom option. Now, one this option the second level menu appears; this one and here we perform a horizontal movement towards the second level menu. Once we reach the second level menu from top we start performing; again another vertical downward movement to reach to the particular menu item that we used to select in this case actual size.

So, then what it tells? These series or activities tell us that selection of the actual size menu option is not a direct point and select activity. So, we need to use different models to compute the performance of such selection task. So, with the classical Fitts’s law model we can model only behavior when we can directly point and select. But in the case of menu we cannot directly point to the item instead we have to follow certain steps.

(Refer Slide Time 04:31)



Difference with Pointing

- Two major differences
  - Need to trace a path before we can “point and select”
  - The object may not be visible when we start the task

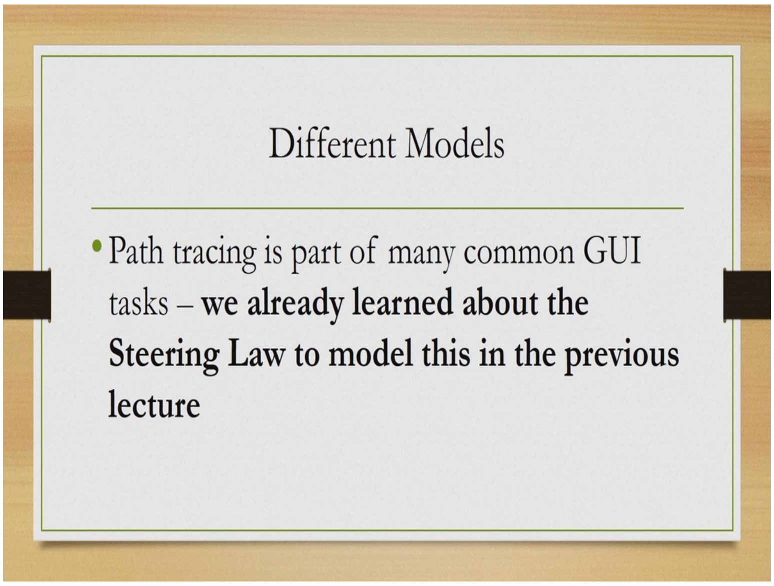
So, in that case the application of the classical Fitts’s law may not be appropriate. So, there are two major differences one is we are following a constraint path. So, in this case if you can recollect; so when we are saying that we perform a particular downward movement of the pointer or horizontal movement of the pointer, we have to ensure that during these movements the point remains within specified boundary.

So, in case of vertically downward movements on first level menu; so the boundary is defined by the menu boundary and the pointer should remain within this boundary. In case of horizontal movement, it should remain within the boundary defined by this zoom menu option. In case of second level vertical movement, the pointer that should remain within the boundary defined by the second level menu.

So, that is what we need to ensure while moving the mouse pointer; so that is the first difference we direct by select task. The second difference is unlike in point and select task in case of menu selection; we may not be able to see the target immediately. So, it is not the case that all the items are visible at once as we keep on performing the sub tasks the items becomes visible; so, everything is not visible at the beginning and we cannot directly point and select the actual size menu option.

First, we have to click on view to get the first level menu option; then we have to select zoom to get the second level menu option and then only will be able to see the actual size menu option and select it. So, these two differences make it distinct from the behavior that is modeled by the classical Fitts's law and that necessitates the development of a separate model.

(Refer Slide Time 06:26)

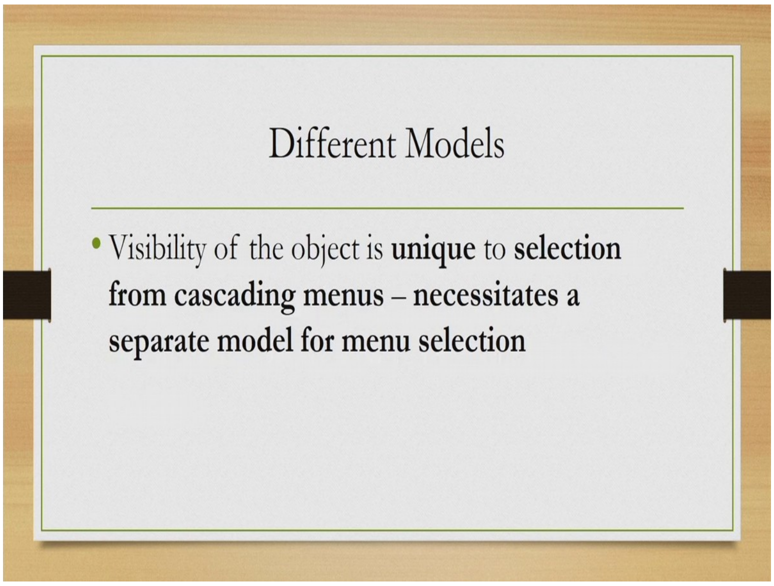


Different Models

- Path tracing is part of many common GUI tasks – **we already learned about the Steering Law to model this in the previous lecture**

Now, in the earlier lecture one part of this difference has been addressed with the development of the Steering law. So, when we are moving the pointer in a constrained path that can be modeled with the Steering law; however, the second problem that is not all the items are visible at once and we need to select as soon as it becomes visible that behavior cannot be modeled with only the Steering law and we need a separate model. So, for my new selection; we need a separate model then only the application of Steering law.

(Refer Slide Time 07:00)

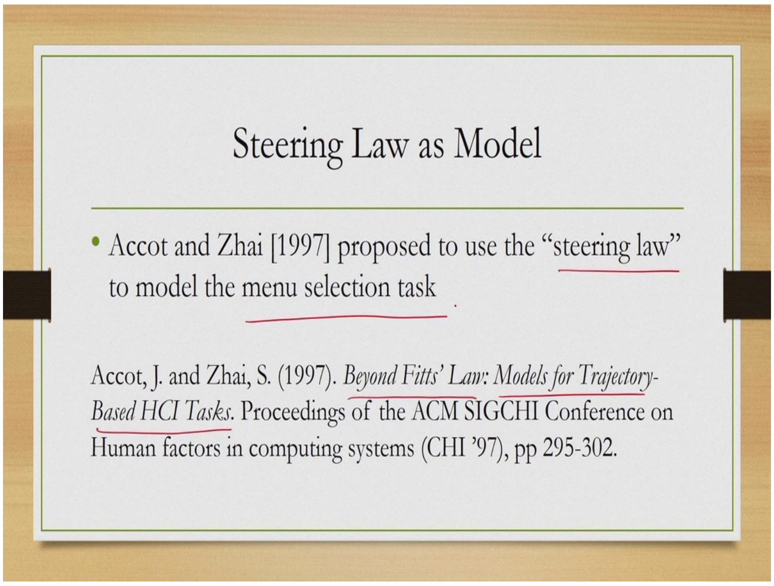


Different Models

- Visibility of the object is **unique** to **selection from cascading menus – necessitates a separate model for menu selection**

And in this lecture we are going to learn such a model.

(Refer Slide Time 07:03)



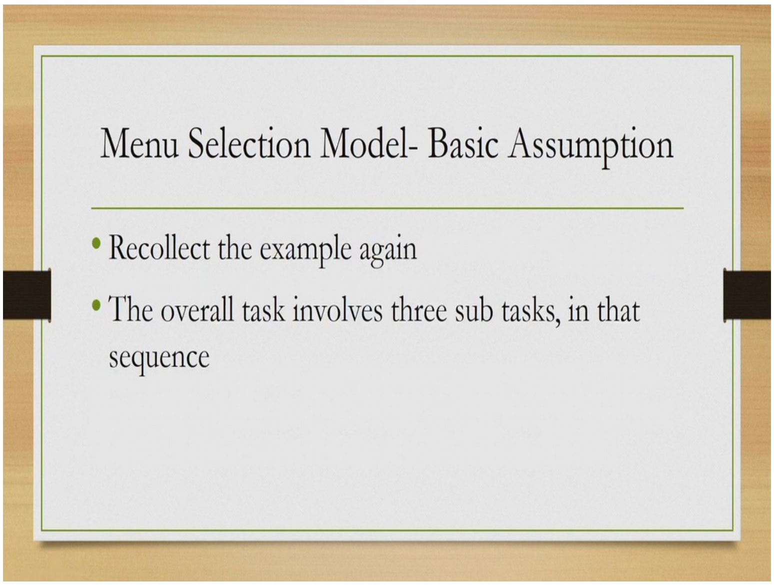
Steering Law as Model

- Accot and Zhai [1997] proposed to use the "steering law" to model the menu selection task

Accot, J. and Zhai, S. (1997). *Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks*. Proceedings of the ACM SIGCHI Conference on Human factors in computing systems (CHI '97), pp 295-302.

Interestingly, the authors of the paper beyond Fitts's law models for trajectory based HCI tasks only Accot and Zhai who proposed the Steering law for use to model performance of constraint navigational tasks, have also proposed to use the law; the Steering law for modeling menu selection task as well.

(Refer Slide Time 07:32)

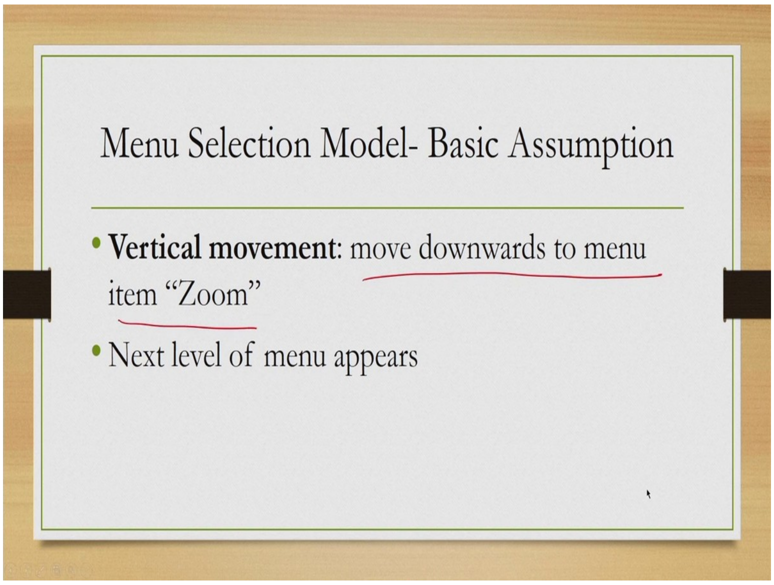


Menu Selection Model- Basic Assumption

- Recollect the example again
- The overall task involves three sub tasks, in that sequence

What they proposed is that each movement can be modeled as a Steering law. So, in the example; if you may recollect we talked off three movement's two vertical movements and one horizontal movement. Each of these movements can be modeled as a Steering law as per the proposal of Accot and Zhai. Now to aid your understanding, let us quickly recap these moments.

(Refer Slide Time 07:59)

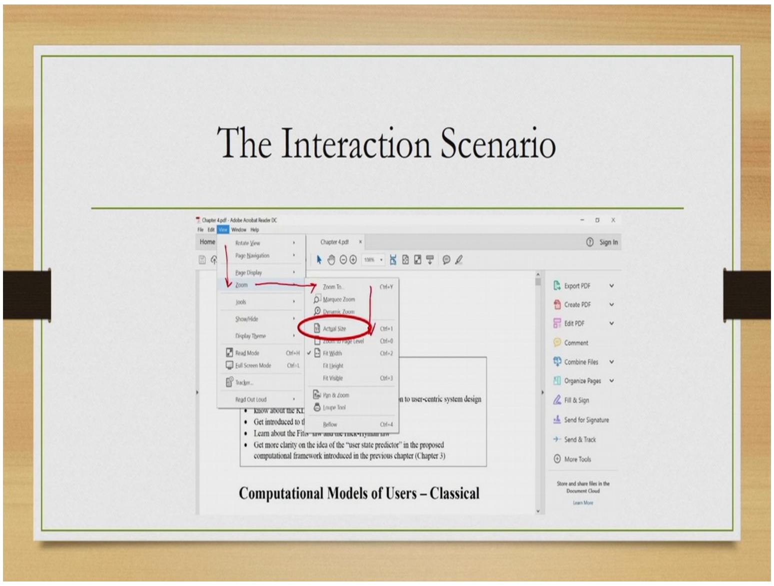


Menu Selection Model- Basic Assumption

- **Vertical movement:** move downwards to menu item "Zoom"
- Next level of menu appears

So, there is one vertical movement; first vertical moment where we move downwards to menu item zoom which brings the second level menu item.

(Refer Slide Time 08:09)



The Interaction Scenario

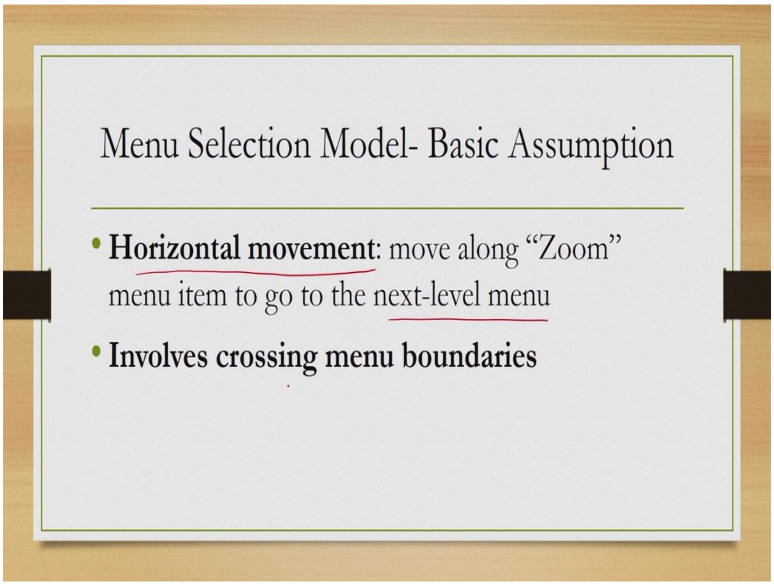
n to user-centric system design

- know about the KL
- Get introduced to t
- Learn about the Fitts' law and the Hick-Hyman law
- Get more clarity on the idea of the "user state predictor" in the proposed computational framework introduced in the previous chapter (Chapter 3)

Computational Models of Users – Classical

So, here we move downwards up to zoom; then the second level item appears.

(Refer Slide Time 08:16)

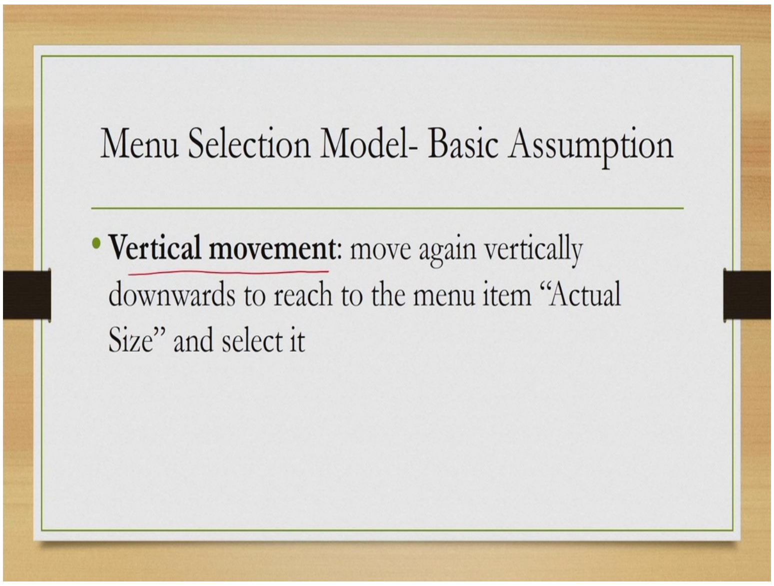


Menu Selection Model- Basic Assumption

- **Horizontal movement**: move along "Zoom" menu item to go to the next-level menu
- **Involves crossing menu boundaries**

Then, we make a horizontal movement to go to the next level menu; it involves crossing menu boundaries. So, here after reaching to zoom, after vertical movement make a horizontal movement which involves crossing the boundary of these two menu items or these two sub menus.

(Refer Slide Time 08:39)

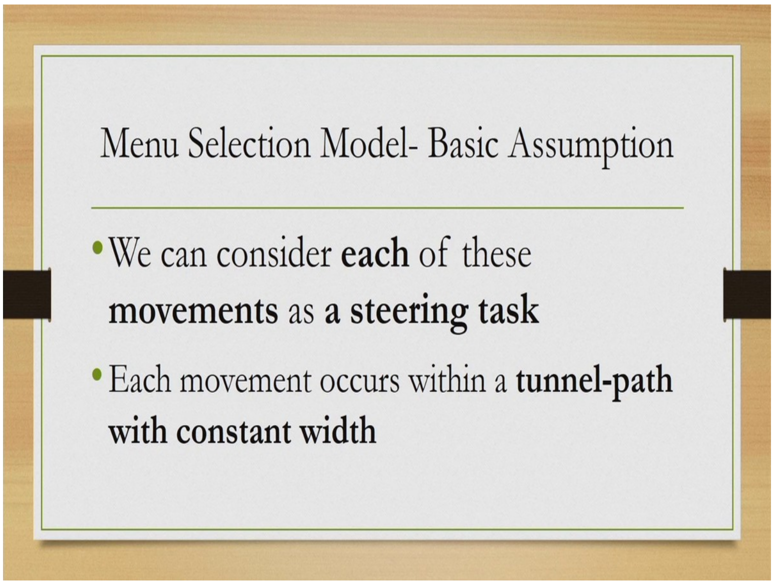


Menu Selection Model- Basic Assumption

- **Vertical movement**: move again vertically downwards to reach to the menu item "Actual Size" and select it

And finally, there is one vertical movement along the sub menu to select the item of interest that is actual size. So, here we perform this downward, then horizontal and then another down or vertical movement to select; to select actual size.

(Refer Slide Time 09:02)

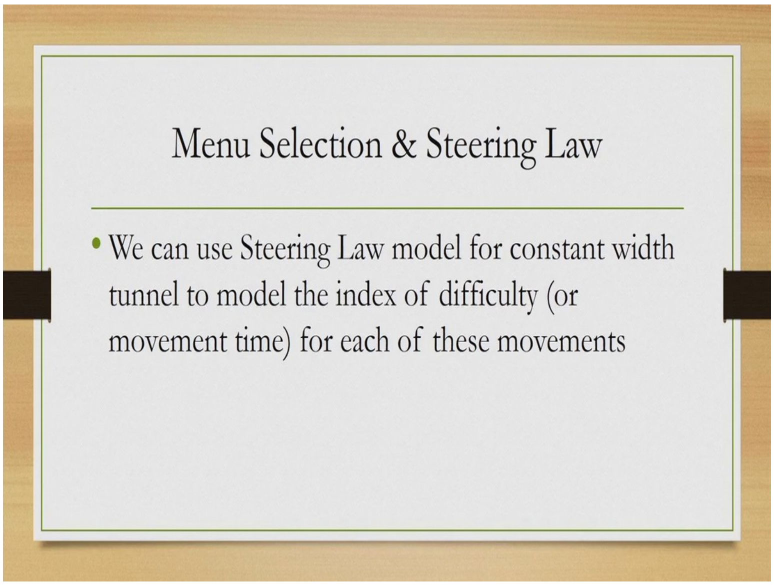


Menu Selection Model- Basic Assumption

- We can consider **each** of these **movements** as **a steering task**
- Each movement occurs within a **tunnel-path with constant width**

Now, each of these movements as I have explained before involves constraint navigation. So, we have to ensure that we are always within some specified boundary. So, intuitively it appears that each of these movements can be modeled with the Steering law and the overall performance can be modeled as a combination of the individual movement models. And in this case, if you may notice that the movements are taking place within tunnels of constant width. So, all the tunnels through which the movement takes place are having constant widths.

(Refer Slide Time 09:48)



Menu Selection & Steering Law

- We can use Steering Law model for constant width tunnel to model the index of difficulty (or movement time) for each of these movements

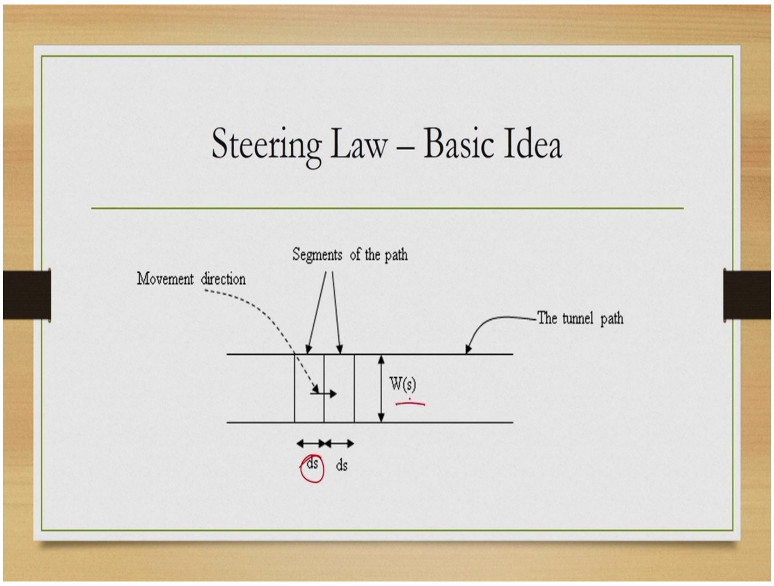
So, in that case we may use the Steering law for tunnel like movements; for movements through tunnel paths having constraint width.

(Refer Slide Time 09:53)

Steering Law - Recap

So, what is that law; let us quickly recap the law before we try to understand its implications.

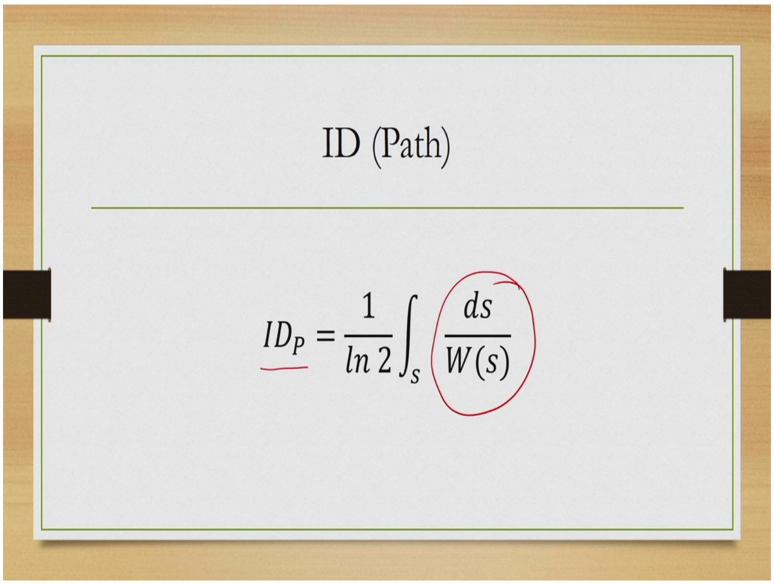
(Refer Slide Time 10:00)



Steering Law – Basic Idea

So, we assume that when we are moving along it constrained path which can be represented as a tunnel; we may consider it to consist of infinitesimally small segments and the segments have very small length, but they may have large width.

(Refer Slide Time 10:25)

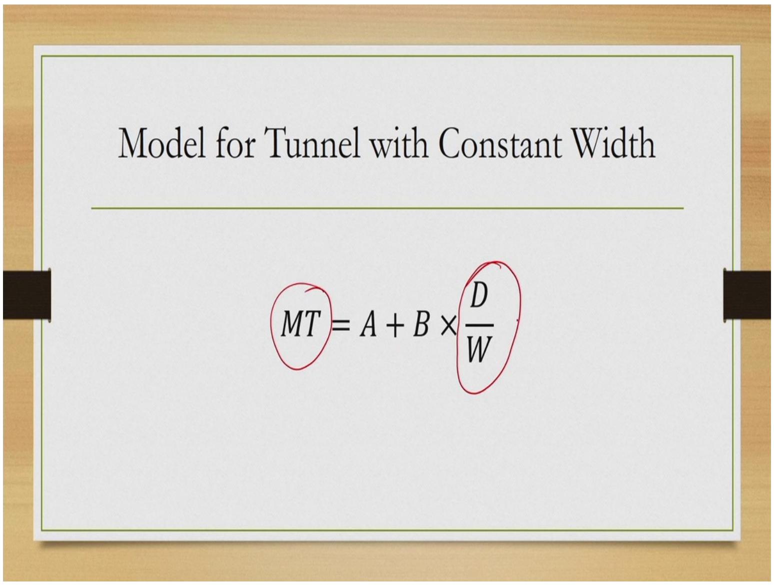


ID (Path)

$$ID_P = \frac{1}{ln\,2}\int_S \frac{ds}{W(s)}$$

The widths maybe different in the most general case and in that case; we can compute the index of difficulty for moving along the tunnel as an integral over this ratio ds by Ws; that is the most general formulation.

(Refer Slide Time 10:45)

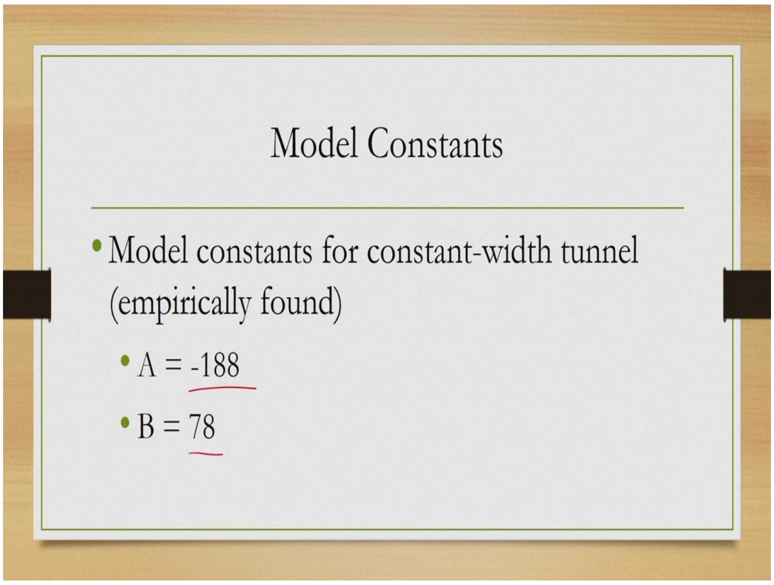


Model for Tunnel with Constant Width

$$MT = A + B \times \frac{D}{W}$$

However, when the tunnel width remains the same throughout the movement that means tunnel width constant width; we can simplify the index of difficulty in this form D/W and it was in this form, we can compute the movement time for moving along the entire tunnel with this expression A+B*(D/W) where the constant values A and B are empirically determined and they have these values; A equal to -188 and B equal to 78.

(Refer Slide Time 11:04)

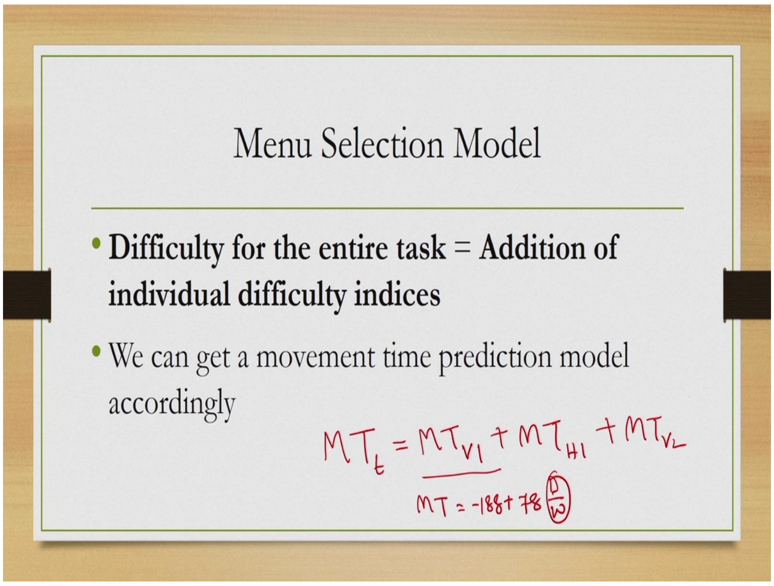


Model Constants

- Model constants for constant-width tunnel (empirically found)
  - A = -188
  - B = 78

So, this is the predictive formulation of movement performance for constraint navigation along a tunnel path having constrained width.

(Refer Slide Time 11:21)



Menu Selection Model

- **Difficulty for the entire task = Addition of individual difficulty indices**
- We can get a movement time prediction model accordingly

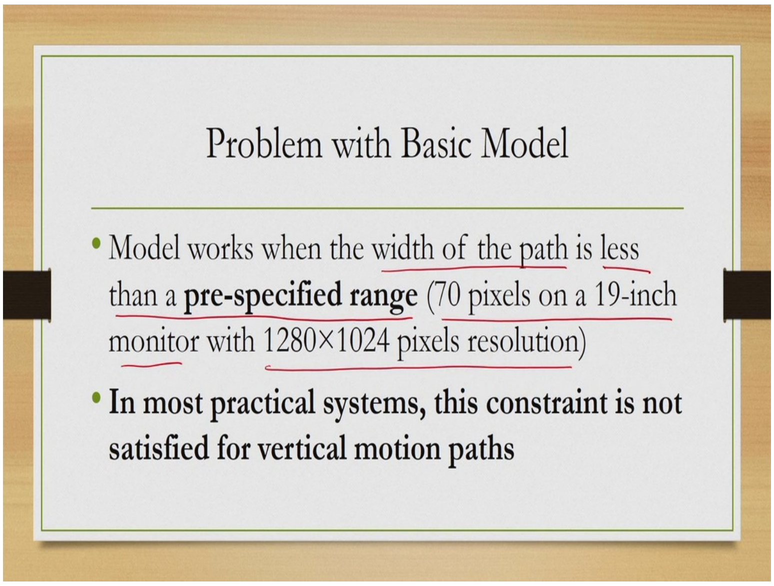
$$MT_t = \underline{MT_{V1}} + MT_{H1} + MT_{V2}$$

$$MT = -188 + 78 \boxed{\frac{D}{W}}$$

Now, in case of menu selection what happens? So, if we assume that individual movements can be modeled with Steering law; then the entire movement starting from the top level menu to the final selection can be modeled as a combination of these individuals Steering law models. And then we can simply add individual movement times modeled by the Steering laws to get the overall movement time for the entire selection task.

So, in other words we can get the total movement time as the movement time for first vertical task plus movement time for first horizontal task plus movement time for the second vertical task. Now, each of these movement times can be modeled as a Steering law task which is MT = -188+78*(D/W) and D and W are dependent on that tunnel. So, this is the most intuitive application of Steering law for menu selection.

(Refer Slide Time 12:29)



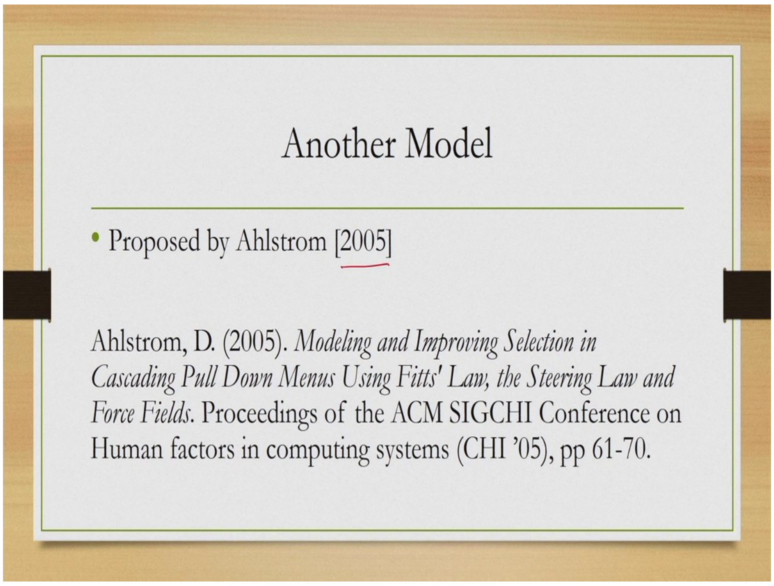
Problem with Basic Model

- Model works when the width of the path is less than a **pre-specified range** (70 pixels on a 19-inch monitor with 1280×1024 pixels resolution)
- **In most practical systems, this constraint is not satisfied for vertical motion paths**

However, there is a problem; in the same paper they have reported it has been found empirically that the approach of using Steering law to model the individual movements and then summing up to get the overall movement time is not applicable in general, it works only under certain constraints. What is that constraint?

It works only when we have the path length less than a pre specified range that is defined as 70 pixels on a 19 inch monitor or with this specific resolution; 1280 by 1024. In other words, if the menu width do not satisfy these constraints; then we will not be able to apply the Steering glove to model these menu selects and task. Clearly, in most practical system it may not be possible to adhere to this constraint and we require some other model.

(Refer Slide Time 13:25)



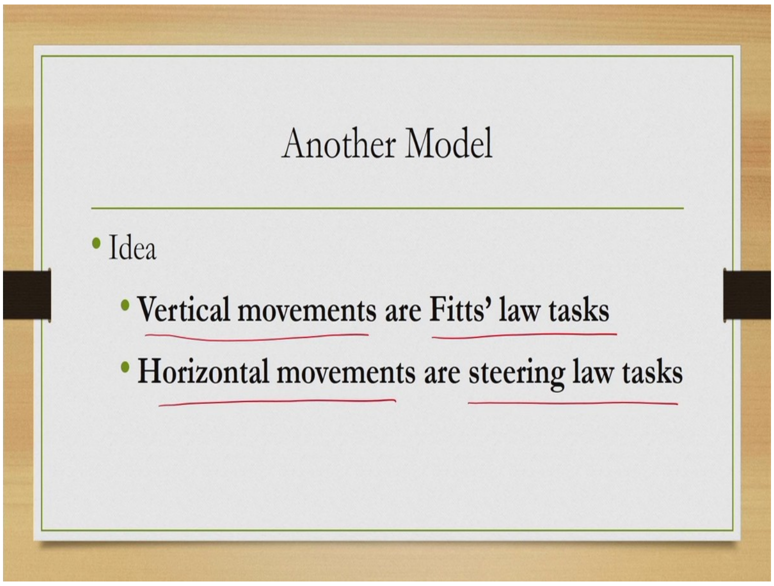
Another Model

- Proposed by Ahlstrom [2005]

Ahlstrom, D. (2005). *Modeling and Improving Selection in Cascading Pull Down Menus Using Fitts' Law, the Steering Law and Force Fields*. Proceedings of the ACM SIGCHI Conference on Human factors in computing systems (CHI '05), pp 61-70.

So, one such model was proposed by Ahlstorm in 2005; if you are interested you can refer to this original paper that is Modeling and Improving Selection in Cascading Pull Down Menus Using Fitts's law; the Steering Law and Force Fields; it was published in the proceedings of CHI 2005.

(Refer Slide Time 13:42)



Another Model

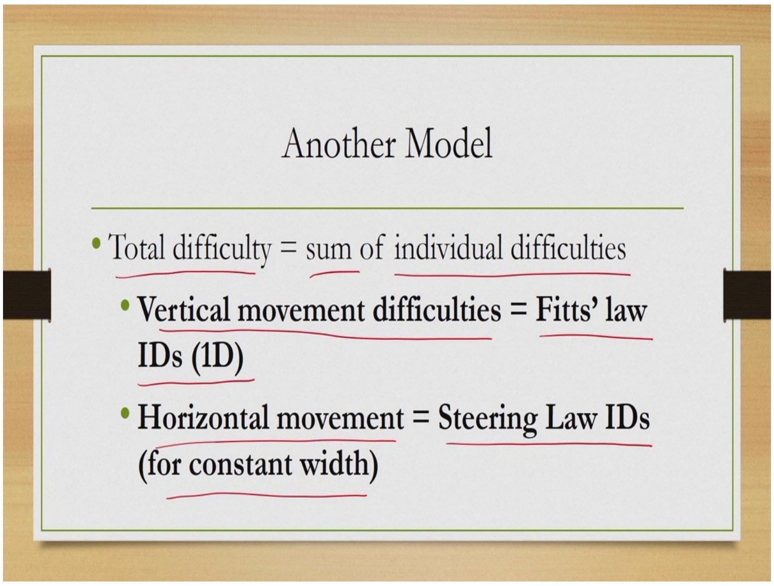
- Idea
  - **Vertical movements are Fitts' law tasks**
  - **Horizontal movements are steering law tasks**

So, what is the basic idea? The idea is that the menu selection task as we have seen in the earlier example comprises of two types of movements; vertical movements and horizontal movements. Now, according to this proposal the vertical movements are

Fitts’s law tasks and can be modeled with the Fitts’s law; horizontal movements are Steering law tasks and can be modeled with the Steering law; so that is the difference.

Earlier, it was assumed that all the movements can be modeled with the Steering law; in this new proposal it is assumed that only the horizontal movements can be modeled with the Steering law and vertical movements can be modeled with the Fitts’s law.

(Refer Slide Time 14:20)



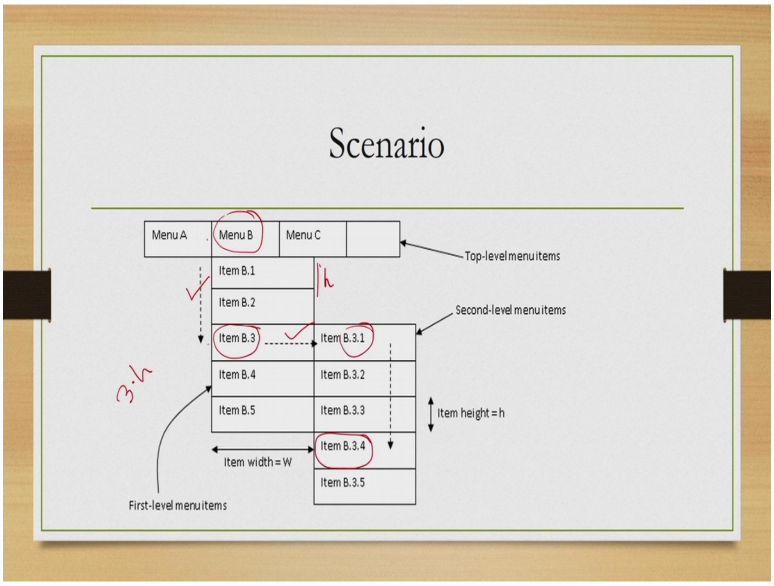
Another Model

- Total difficulty = sum of individual difficulties
  - Vertical movement difficulties = Fitts’ law IDs (1D)
  - Horizontal movement = Steering Law IDs (for constant width)

So, the overall model will be a combination of individual models and the total difficulty of the task can be found out by summing up the individual task difficulties, where we can find the vertical movement has difficulties by applying the Fitts’s law task difficulty measure that is index of difficulty measure and horizontal movement task difficulties by applying the Steering law index of difficulty measure for constant width.

So, by applying these index of difficulty measures we can find out that develop difficulty for individual movements and then we can sum up all these difficulties to get the overall difficulty; using that overall difficulty we can come up with a model to predict the overall movement time.

(Refer Slide Time 15:05)



Scenario

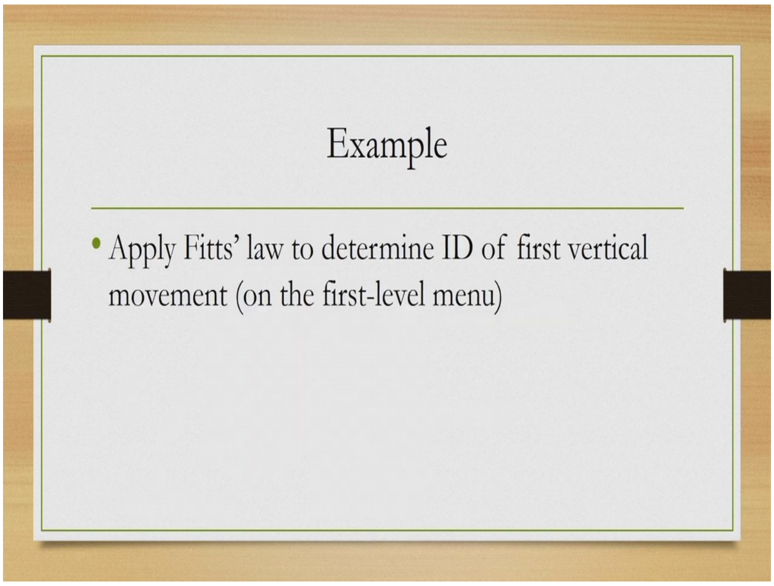
Menu A
Menu B
Menu C
Top-level menu items
Item B.1
Item B.2
Item B.3
Item B.4
Item B.5
Second-level menu items
Item B.3.1
Item B.3.2
Item B.3.3
Item B.3.4
Item B.3.5
Item height = h
Item width = W
First-level menu items

Now, let us try to understand the derivation of this model in terms of an example. Consider this scenario here we are trying to select a menu item; it is a drop down menu once we click on this top level menu item, then the first level drop down list appears. There we want to select this menu item, then the next level dropdown list appears we perform the horizontal movement to go to this second level menu item and then perform another vertical movement to finally, select this item.

So, here like in the previous example we have one vertical; vertical movement that is vertically downward movement of the pointer, then one of our agent tell movement from item B.3 to B.3.1 and then one more vertical downward movement of the pointer.

Now, as the model assumes the vertical movements can be modeled with the Fitts’s law and the horizontal movement can be modeled with the Steering law. Now, let us try to first model the vertical movements. So, in the first case if we may notice; we are trying to move to this point from this point. So, this is essentially a movement of see if height is h; then this is essentially a movement of 3 h. So, we are trying to cover a distance of 3 h and the width is width of the object is W.

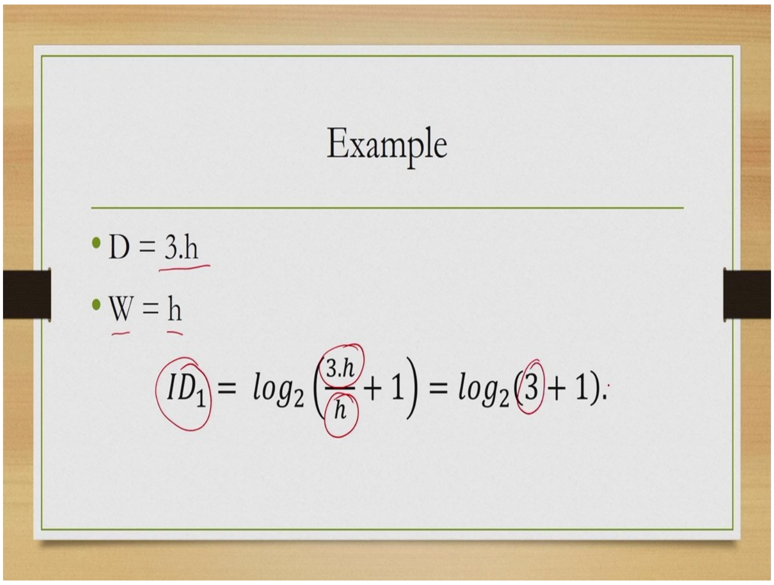
(Refer Slide Time 16:48)



Example

- Apply Fitts' law to determine ID of first vertical movement (on the first-level menu)

(Refer Slide Time 16:53)



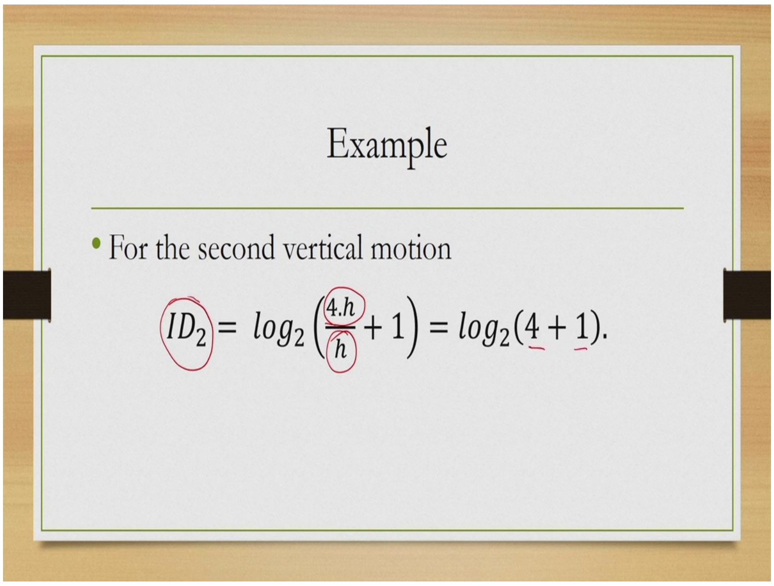
Example

- D = 3.h
- W = h

$$ID_1 = log_2\left(\frac{3.h}{h}+1\right) = log_2(3+1).$$

So, if we try to apply Fitts's law on the first vertical movement then what we will get is D is 3h and the width is h. So, the ID1 as per Fitts's law is $\log_2(D/W+1)$ which can be simplified as $\log_2(3 + 1)$.

(Refer Slide Time 17:15)



Example

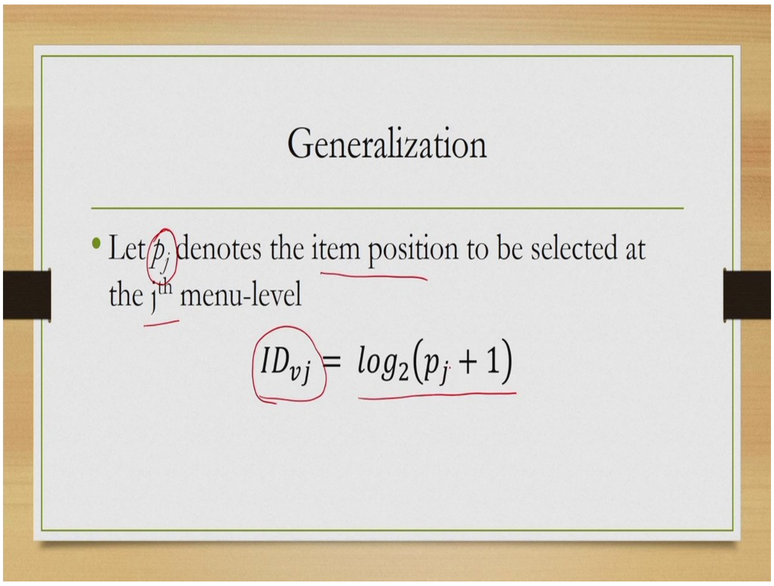
- For the second vertical motion

$$ID_2 = log_2\left(\frac{4.h}{h}+1\right) = log_2(4+1).$$

What happens in the case of second vertical movement? In a similar way, you can actually find out that we can compute the index of difficulty for the second vertical movement as $\log_2(4h/h+1)$ or $\log_2(4+1)$.

So, then from these two examples we can try to come up with a generalized expression for computing the index of difficulty for any vertical movement during a menu selection task.

(Refer Slide Time 17:47)



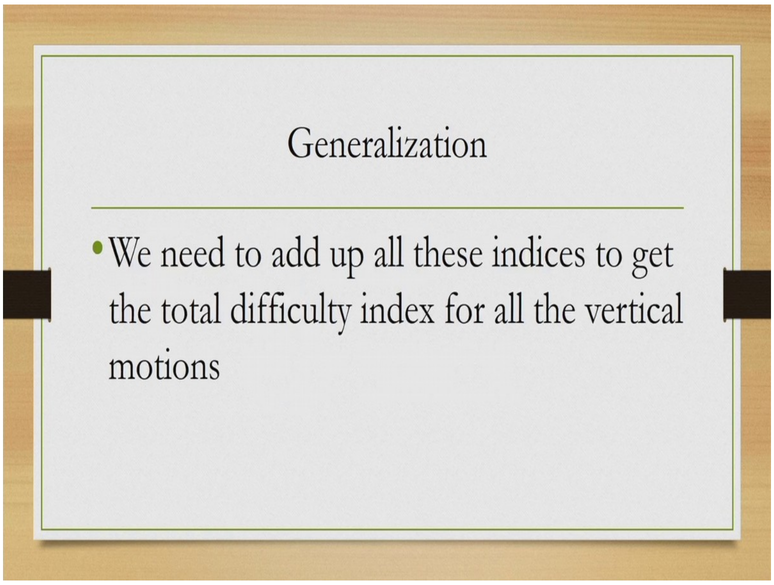
Generalization

- Let p_j denotes the item position to be selected at the j^{th} menu-level

$$ID_{vj} = log_2(p_j+1)$$

What that will look like? Let us assume that pj denotes the item position in the particular menu which is at the jth level. So, first level, second level, third level jth level and pj it denotes the item position that is first, second, third, fourth something. Then the index of difficulty for that vertical movement to select that particular menu item can be represented by simply by this expression $\log_2(pj+1)$, where pj is the position in that particular menu.

(Refer Slide Time 18:21)

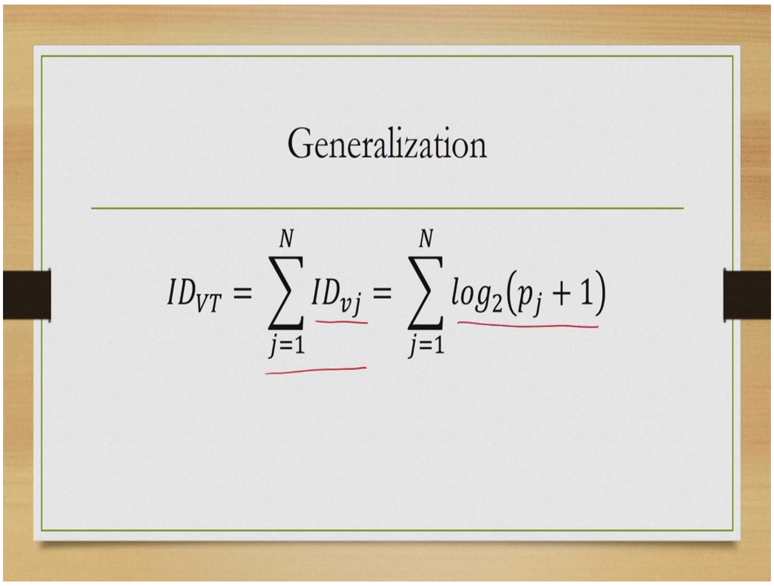


Generalization

- We need to add up all these indices to get the total difficulty index for all the vertical motions

And we need to add up all this index of difficulties to get the overall index of difficulties for all the vertical movements which is a simple addition that can be represented in this way.

(Refer Slide Time 18:32)

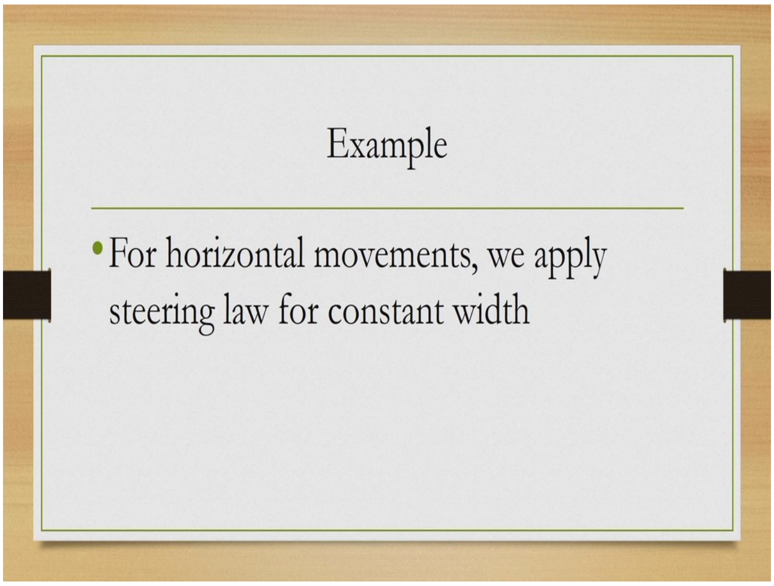


Generalization

$$ID_{VT} = \underline{\sum_{j=1}^{N}} \underline{ID_{vj}} = \sum_{j=1}^{N} \underline{log_2(p_j + 1)}$$

For all j, we can compute the ID and then we add up where each ID is represented by $\log_2(pj+1)$; so this is for vertical movements. Then let us try to compute the index of difficulty for horizontal movement; for horizontal movement we need to apply the Steering law particularly the law for movement along a tunnel with constant width.

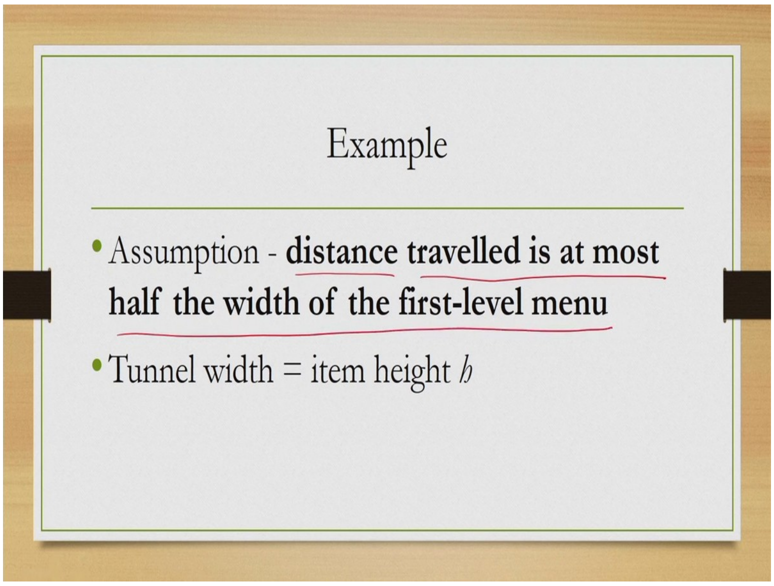
(Refer Slide Time 18:48)



Example

- For horizontal movements, we apply steering law for constant width

(Refer Slide Time 19:04)

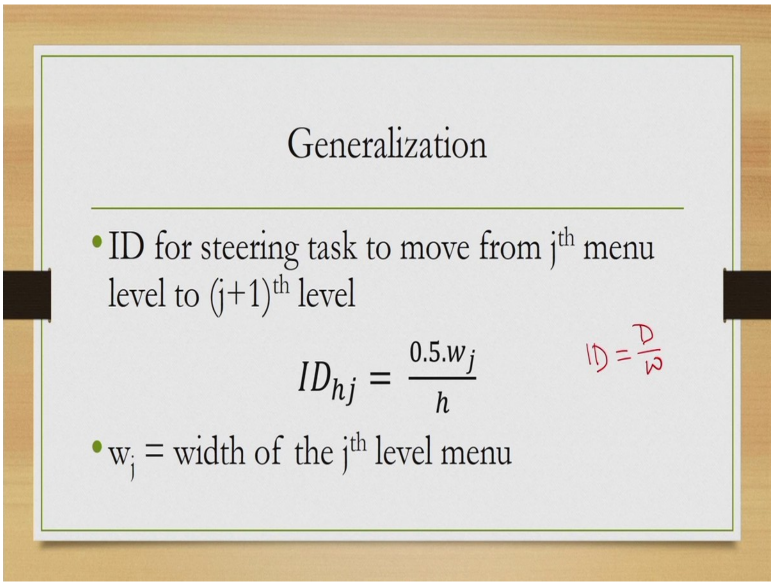


Example

- Assumption - **distance travelled is at most half the width of the first-level menu**
- Tunnel width = item height *h*

Now, in this case we need to move from one menu to another menu by crossing the boundary. Now, here we can make one simplifying assumption that is while making this horizontal movement; the total distance travelled can be assumed to be at most half the width of the first level menu and the tunnel width is the item height h.

(Refer Slide Time 19:26)



Generalization

- ID for steering task to move from jth menu level to (j+1)th level

$$ID_{hj} = \frac{0.5.w_j}{h}$$

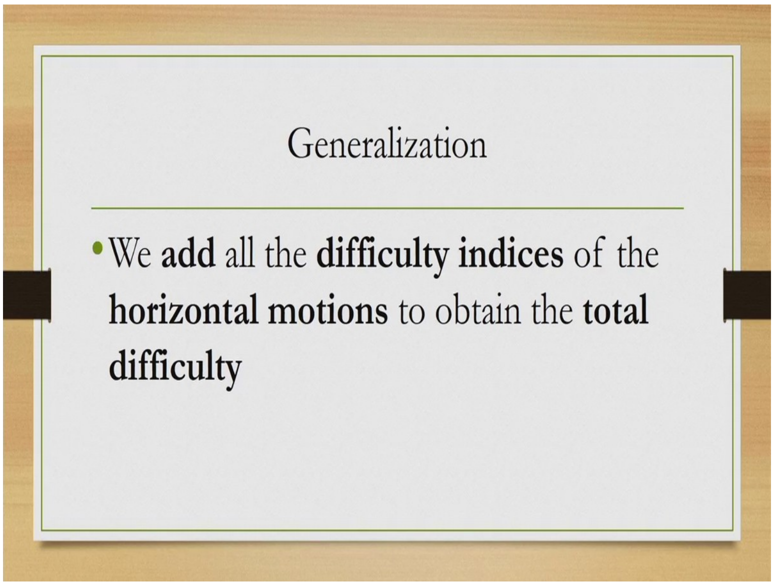
$ID = \frac{D}{W}$

- w$_j$ = width of the jth level menu

In that case, then if you may recollect that the ID for Steering law tasks having tunnels with constant width is D/W. Now, in our case based on the assumption; we can see this is 0.5*wj, where wj is the width of the jth level menu and h is the height.

Like in the case of vertical movements we add all the indices of difficulties for all horizontal movements to get the viral index of difficulty for all the horizontal movements.

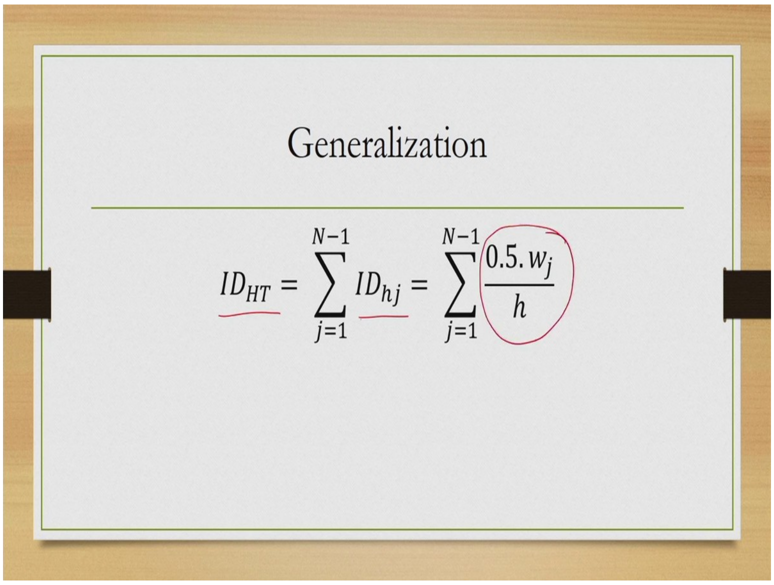
(Refer Slide Time 20:06)



Generalization

- We **add** all the **difficulty indices** of the **horizontal motions** to obtain the **total difficulty**

(Refer Slide Time 20:08)

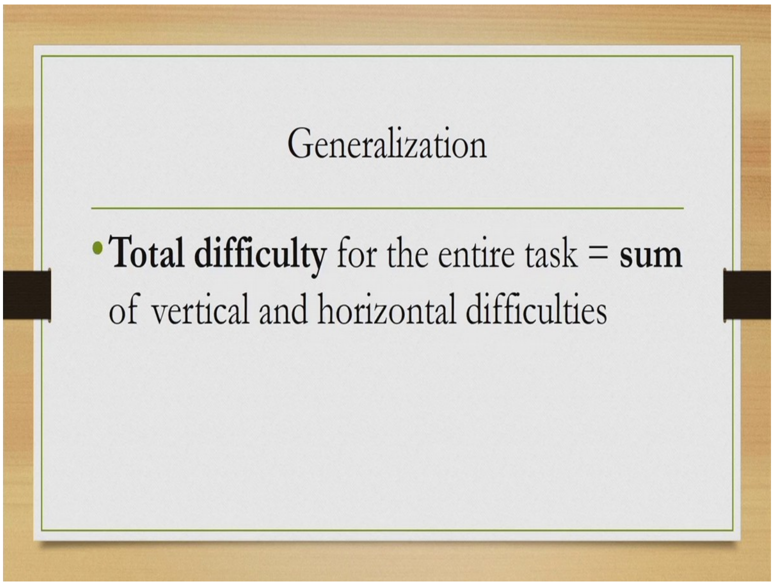


Generalization

$$\underline{ID_{HT}} = \sum_{j=1}^{N-1} \underline{ID_{hj}} = \sum_{j=1}^{N-1} \boxed{\frac{0.5.w_j}{h}}$$

That is again a simple summation of the form shown in this expression where ID indicates the overall index of difficulty for all horizontal movements taken together which is summation of all individual horizontal movement IDs and for each ID; we can compute it with this formula.

(Refer Slide Time 20:28)

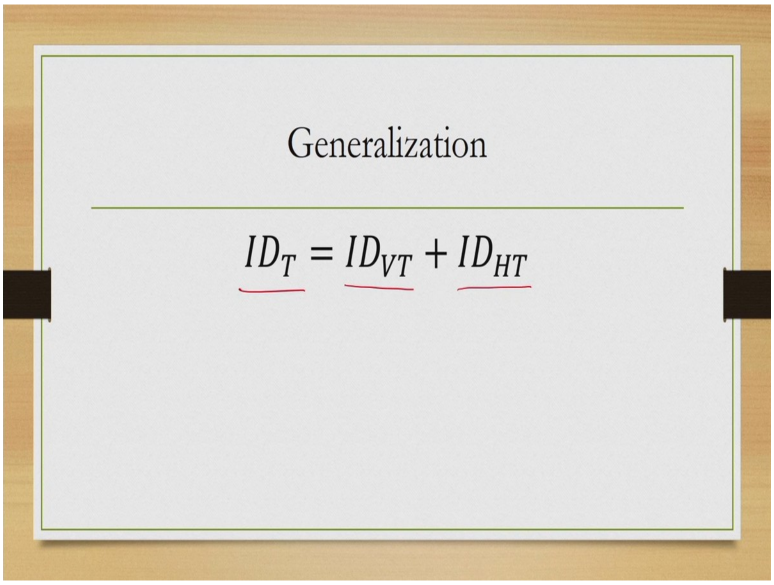


Generalization

- **Total difficulty** for the entire task = **sum** of vertical and horizontal difficulties

So, once we get this overall difficulty level for all horizontal movements and overall difficulty level for all vertical movement; what is left is just to add these two to get the overall difficulty for all movements. So, the total difficulty of the entire task is summation of the total difficulty for all vertical and the total difficulty for all horizontal movements.

(Refer Slide Time 20:51)

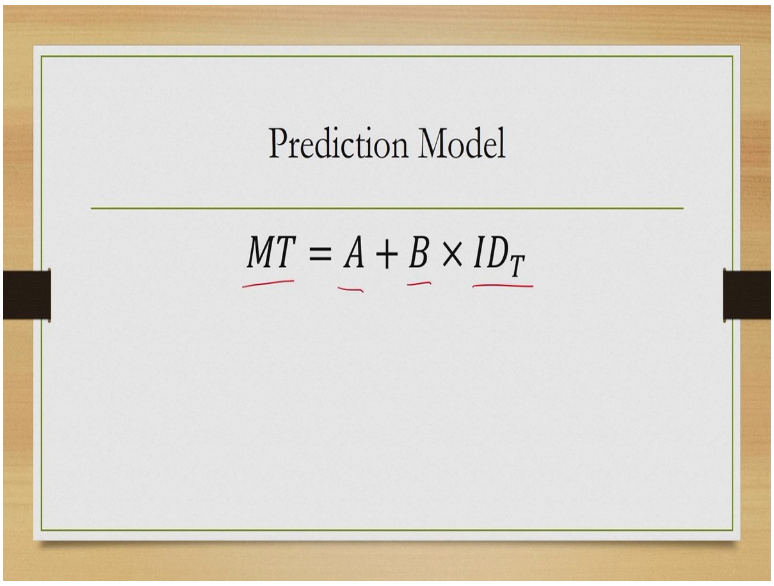


Generalization

$$\underline{ID_T} = \underline{ID_{VT}} + \underline{ID_{HT}}$$

In other words, index of difficulty for the total task is index of difficulty for all vertical tasks and index of difficulty of all horizontal tasks summed up together.

(Refer Slide Time 21:09)



Prediction Model

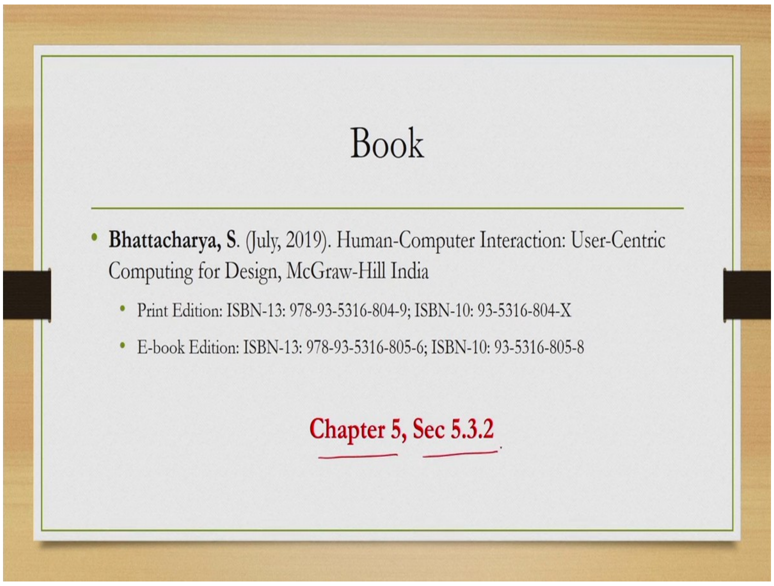
$$\underline{MT} = \underline{A} + \underline{B} \times \underline{ID_T}$$

Now, with this calculation of index of difficulty for the entire task; what we can propose is the predictive model to compute performance in terms of movement time. So, we can come up with a model of the form A+B*ID_T; where ID_T we have shown how to calculate and A and B are constants which maybe derived empirically.

So, this is the model to predict the movement time during selection of a menu item and this model is a combination of classical Fitts’s law and the Steering law. To recollect the model assumes that there will be series of vertical and horizontal movements which is actually the case; the vertical movements can be modeled with the Fitts’s law the horizontal movements can be modeled with the Steering law.

So, in order to get the overall movement time we need to identify or compute the individual movement times that we can do by applying the Fitts’s law for individual vertical movements and the Steering law for individual horizontal movements and then we can add them together to get the overall movement time.

(Refer Slide Time 22:19)



Book

- **Bhattacharya, S**. (July, 2019). Human-Computer Interaction: User-Centric Computing for Design, McGraw-Hill India
  - Print Edition: ISBN-13: 978-93-5316-804-9; ISBN-10: 93-5316-804-X
  - E-book Edition: ISBN-13: 978-93-5316-805-6; ISBN-10: 93-5316-805-8

Chapter 5, Sec 5.3.2

So, that is in a nutshell what the model tells us about selection from a menu item. Whatever we have discussed so far can be found in this book and interested reader is advised to go through chapter 5 section 5.3.2 to get all the materials that we have covered today. So, the book contains both the models and you can learn more about those models including other references and details of the references from the sections and chapters mentioned.

Thank you and goodbye.