

Modeling Visual Search Time for Soft Keyboards

Lecture #14

Topics to cover

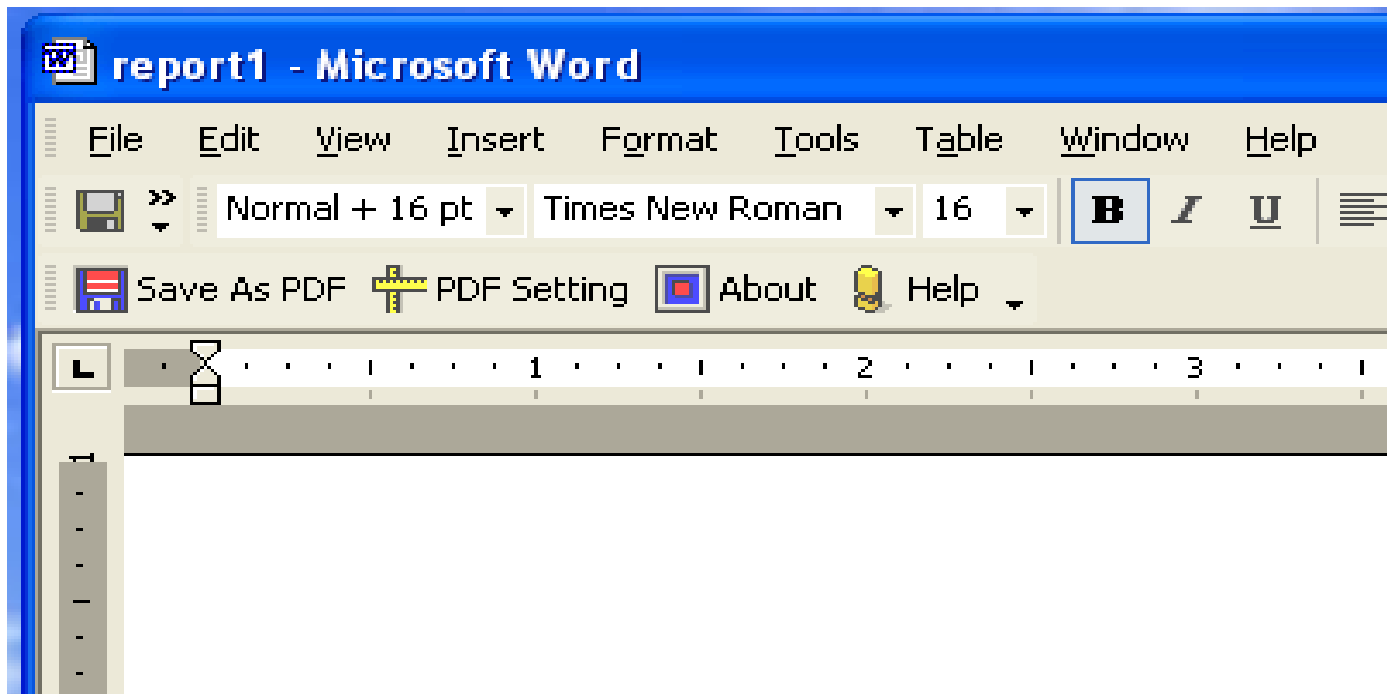
- Introduction
- Models of Visual Search
- Our Proposed Model
- Model Validation
- Conclusion

Introduction

- What is Visual Search?
- Types of Visual Search
- What is a Soft Keyboard?
- Why it is important?
- Issues in designing a Soft Keyboard

Introduction (cont..)

- **What is Visual Search?**
 - Visual search is our ability to detect a target among distractors
 - Example: Searching File menu in *MS-WORD*

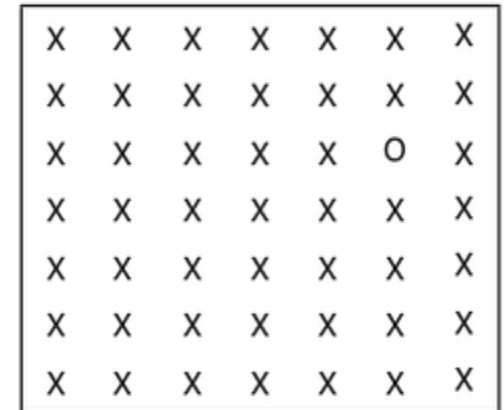


Introduction (cont..)

- **Types of Visual Search**

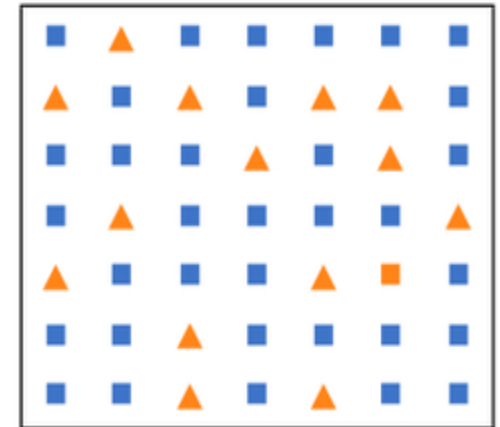
- *Feature Search*

Feature Search is the process of searching for targets defined by a single visual feature, such as color, size, orientation or shape, sometimes called *Parallel Search*



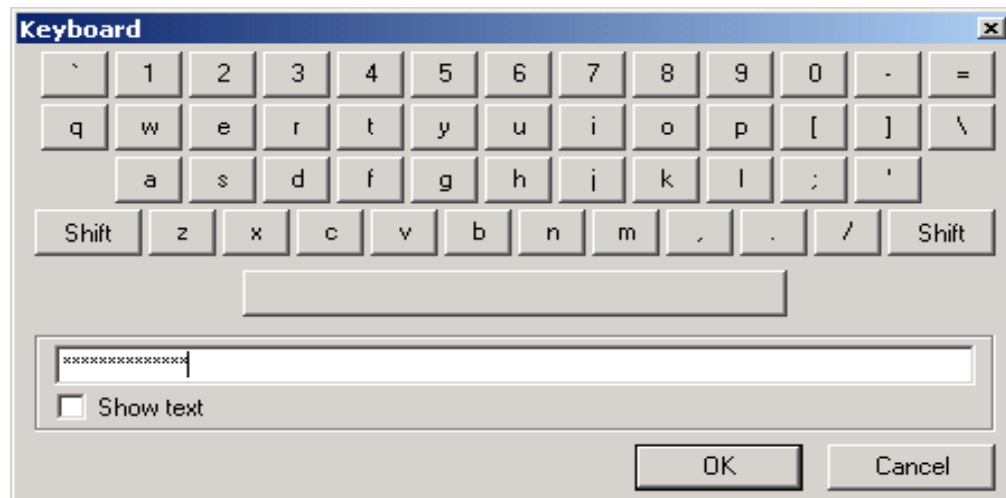
- *Conjunction Search*

It occurs when a target is defined not by any single visual feature, but by a combination of two or more features, sometimes called *Serial Search*



Introduction (cont..)

- **What is a Soft Keyboard?**
 - Soft Keyboards are on-screen representation of physical keyboards
 - The time to locate one key in soft keyboard interface is known as *Visual Search time in Soft Keyboard*



Introduction (cont..)

- Why a Soft Keyboard is important?
 - In Hand-held devices
 - For motor-impaired people to use computer



Introduction (cont..)

- Issues in designing Soft Keyboard
 - Problems in *user testing*
 - Fitts'-digraph model
 - Deficiencies in *Hick-Hyman Law*

Introduction (cont..)

- **Fitts' Digraph Model**
 - Used to model virtual keyboard performance
 - There are 3 components in this model
 - *The visual search time component (RT)*: The time to locate the target character key on the interface. The time is modeled using the *Hick-Hyman law*.
 - *The motor movement time component (MT)*: The time to move the finger or stylus (common input method used on soft keyboards) from the last selected key to the target key. The time is calculated using the *Fitts' law*.
 - *Digraph probability of character pairs (DP)*: The digraph probability is calculated from a corpus of text.

Fitt's-digraph Model

Fitts'-digraph model

- Movement time using Fitts law

$$MT_{ij} = A + B \times \log_2(D_{ij} / W_j + 1)$$

- Visual search time using Hick-Hyman law

$$RT = A' + B' \times \log_2 N$$

- Digraph probability (calculated from a corpus)

$$P_{ij} = f_{ij} / \sum_{i=1}^N \sum_{j=1}^N f_{ij}$$

Fitt's-digraph Model

- Average movement time

$$MT_{MEAN} = \sum_{i=1}^N \sum_{j=1}^N (MT_{ij} \times P_{ij})$$

- Performance (CPS)

$$Novice = 1 / (RT + MT_{MEAN})$$

$$Expert = 1 / MT_{MEAN}$$

- Performance (WPM)

$$WPM = CPS \times (60 / W_{AVG})$$

Introduction (cont..)

- **Hick-Hyman Law (Hick, Hyman)**

- States that the time required to make a choice among N objects given a target is a logarithmic function of N
 - N is the number of possible responses,
 - c and d are constants

- **Response time**

$$RT = c + d \log_2 N$$

- d was set to the reciprocal of the slowest speed of text entry (i.e. 1/5 or 0.20)

Fitt's-digraph Model

- **Drawbacks of Hick-Hyman Law**
 - No. of alternatives (N) should be based on the no. of keys in the keyboard rather than the number of letters.
 - Visual search time on a soft keyboard interface is affected by various other factors like familiarity of the organization of the letters, number of letters on each key, grouping of letters etc. in addition to number of elements on the interface

Models of Visual Search

1. Simple Search Model (Neisser)

This model predicts that the response time will increase linearly as a function of the total number of objects that must be examined

Mean Predicted Response Time

- $RT_a = \alpha_a + \beta (m_s)$ -----when target absent
- $RT_p = \alpha_p + \beta (m_s + 1)/2$ -----when target present

Where

- m_s = The total number of items in the display
- β = The amount of search time required to examine each item in a serial, non-overlapping search
- α = A constant amount of time required for other processes not related to the number of items in the display (which might change depending on whether the target is absent or present)

Models of Visual Search (cont...)

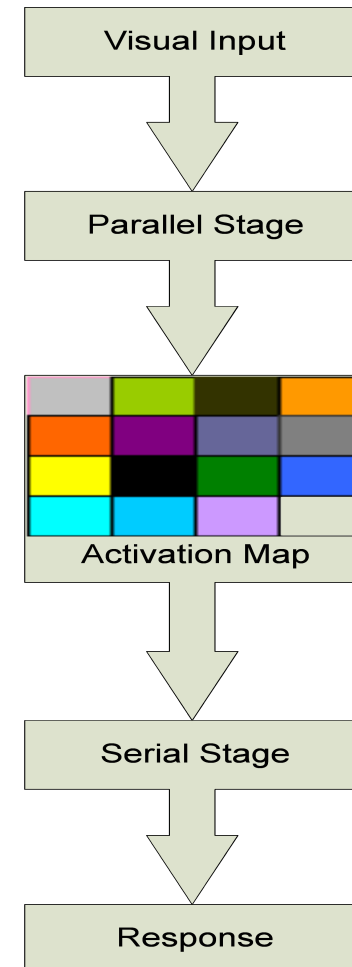
2. Feature Integration Model (Treisman)

- Early in the processing of visual information, a separate *feature map* is created for each primary feature (such as size, color, shape, and orientation)
- These feature maps are searched in parallel, pre-attentively, for the presence of a single object that possesses a simple feature corresponding to one of the maps
- If the target item does not possess a distinguishing primary feature, then the model must resort to a serial search process much like simple search model

Models of Visual Search (cont....)

3. Guided Search Model (Cave and Wolfe)

- The Guided Search model had a pre-attentive stage and an attentive stage. Information from the first stage could be used to guide deployments of selective attention in the second stage
- In the parallel stage, the element that resembles the target and differs most from other elements is identified



Models of Visual Search (cont...)

- Search time is linearly related to the number of steps for locating the target

- The time for Visual Search then calculated by the formula

$$RT = \text{No. of Steps} * \text{Cycle time} + \text{Overhead Time}$$

Where,

- RT= reaction time or response time, the time required to search an element in the interface
- Cycle time = the constant time required for processing of an individual element in the serial search. This value can be taken from the cognitive science
- Overhead time = time for key pressing during experiment

Models of Visual Search (cont....)

- The activation score for each and every element can be calculated by the following equation.

$$A_i = \exp \left(\frac{p}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^n \frac{|f_i - f_j|}{|d_i - d_j|} \right) - q |f_i - t|$$

Where

A_i = Net activation of element i for a visual feature

p = Bottom-up coefficient

q = Top- down coefficient

f_i = Activation for element i for that feature

f_j = Activation for rest of the other elements

$|d_i - d_j|$ = Absolute difference in the distance between the i_{th} element and j_{th} element

t = Activation for the target for that particular visual feature

n = No. of elements in the interface

Identified factors for Soft Keyboard

1. Grouping
2. Size of the keys
3. Number of groups
4. Distance between the groups
5. Size of the groups
6. Number of elements
7. Shape characteristics

Our Proposed Model

- Selecting Model Parameters and Factors
- Collecting Experimental Data from Search Trials
- Parameter Fitting
- Calculations of Activation Score and Visual Search Time

Model Parameters and Factors

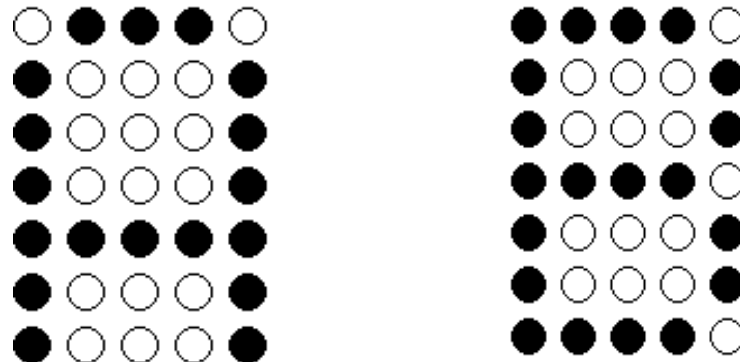
- **Factors**

- *Grouping Factor*

Whenever, there are large number of elements present in the interface, grouping and labeling of elements have decrease the visual search time

- *Shape Characteristics*

It describes how much amount one element differs from other elements by shape.



Model Parameters and Factors (cont..)

■ Parameters

- *Activation Scores* for each and every elements for each and every factor
- *Bottom-up coefficient* for the corresponding factor
- *Top-down coefficient* for the corresponding factor
- *Cycle Time*
- *Overhead Time*

Collecting Experimental Data

- **Step 1**

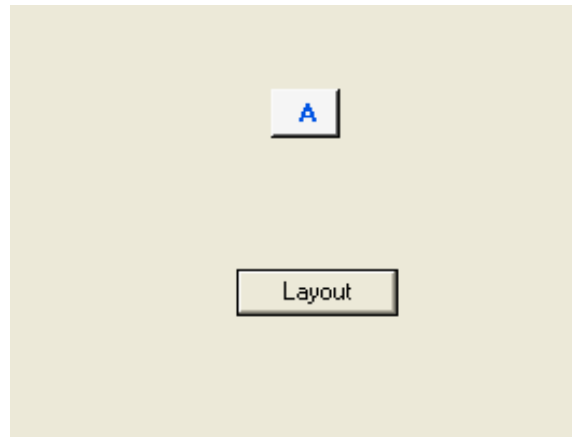
Keyboard hooking program was run by which all the key press events were stored in a file in a temporary folder along with the timestamp

- **Step 2**

After finishing the first step, participants were shown a screen which is shown below

- **Step 3**

As soon as the user was ready for search, the control was set on to the “layout” button. Then, the user hit the <ENTER> button



Experiment (cont..)

- **Step 4**

After hitting the <ENTER>button, a screen which contained the keyboard layout came. Then, the participant had to search for the target

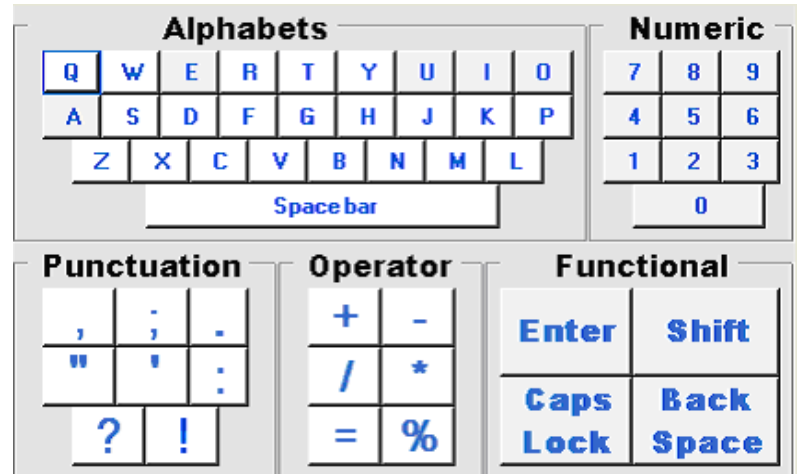
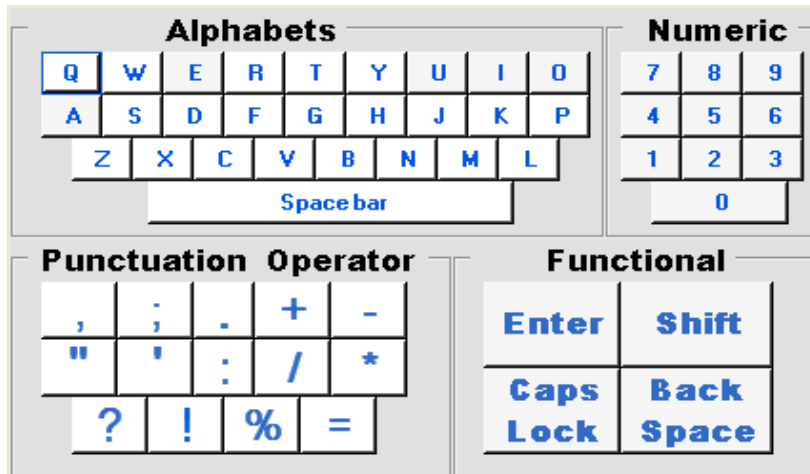
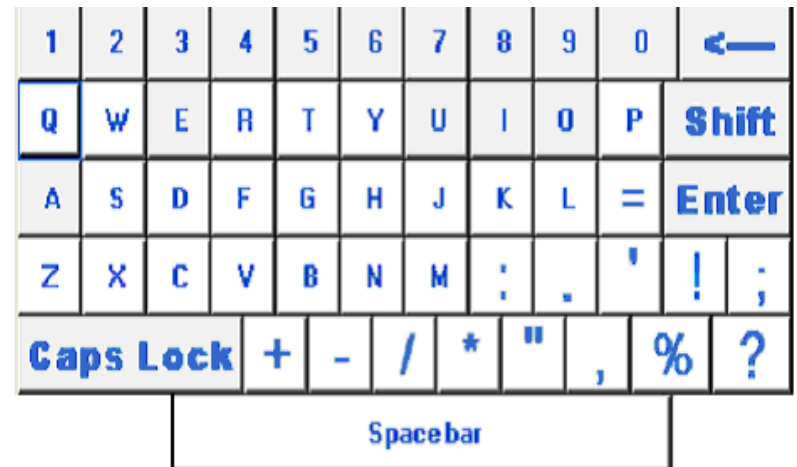
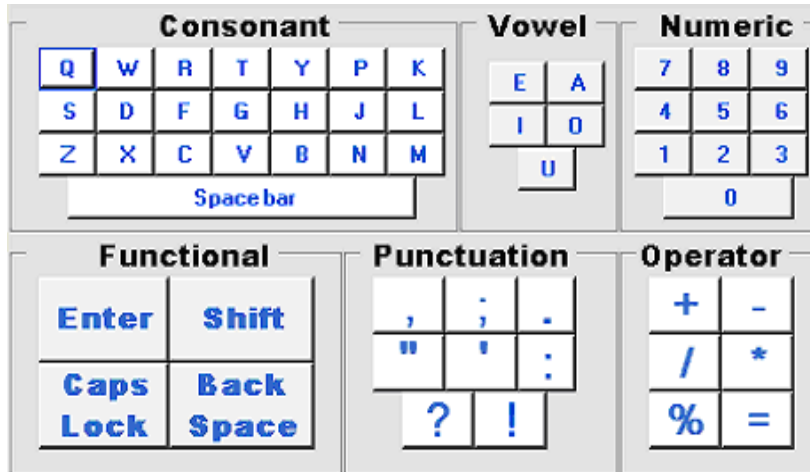
- **Step 5**

As soon as the participant found the target, he again pressed the <ENTER> button

- **Step 6**

Some verifications were also done to investigate whether the participant actually saw the target or not

Layouts for Experiment



Parameter Fitting

- The average search times across the 10 participants in all the search trials were used to fit the model predictions
- With a given set of model parameters, the model can compute a prediction of the visual search time for locating the target
- The predicted search times were then correlated with the experimentally measured search times, and the *squared correlation coefficient* was computed
- A single model parameter was then changed, and the predicted search times for all targets were computed again

Parameter Fitting (cont..)

- These predicted search times were used again to calculate the squared correlation coefficient with the experimental search times
- If the parameter change led to an increase in the squared correlation coefficient, the change was kept otherwise change was undone
- This process iterated until it seemed that no further changes in model parameters led to an increase in the squared correlation coefficient
- The model parameters were first set manually

Calculations of Activation Score and Visual Search Time

- After representing the elements in the *pixel matrix* form, initial activation for shape can be calculated as follows
 - Binary values can be calculated by taking the blank pixels as '0' and filled pixels as '1'
 - From the binary representation, decimal values can be calculated for each element
 - From the initial activations, we can calculate the final activations from the activation formula
- After the calculating activations for individual factors, total activations can be calculated by,
$$\text{Activation (Total)} = \text{Activation (group)} + \text{Activation (shape)}$$

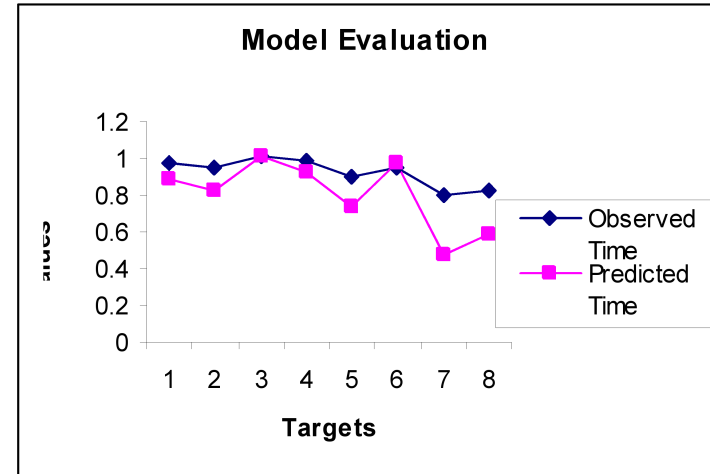
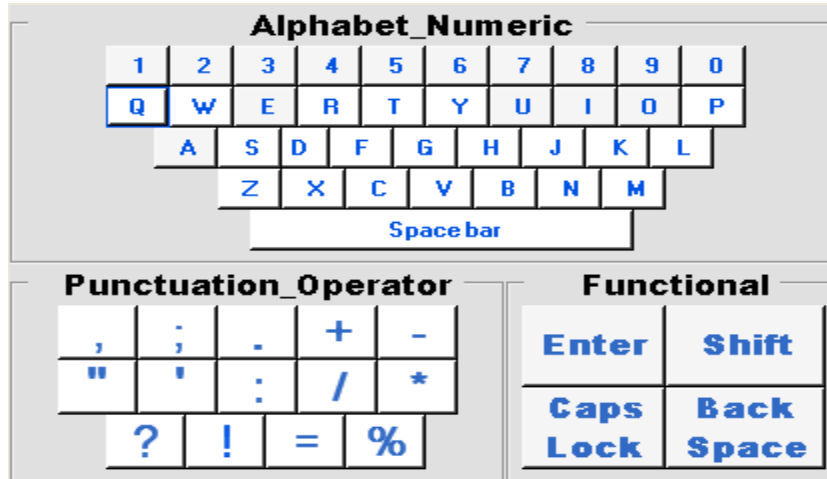
Calculations of Activation Score and Visual Search Time

- Distance between two elements has been calculated by the *Euclidian distance*
- Visual search time or Response time (RT) can be calculated as follows.

$$\text{RT} = \text{No. of Steps} * \text{Cycle time} + \text{Overhead time} + \text{System response time}$$

- Cycle time= 30 msec (Taken from *Cognitive Science*)
- Overhead time = 50 msec
- System response time = Time delay between successive <ENTER> key presses = 150 msec
(calculated)

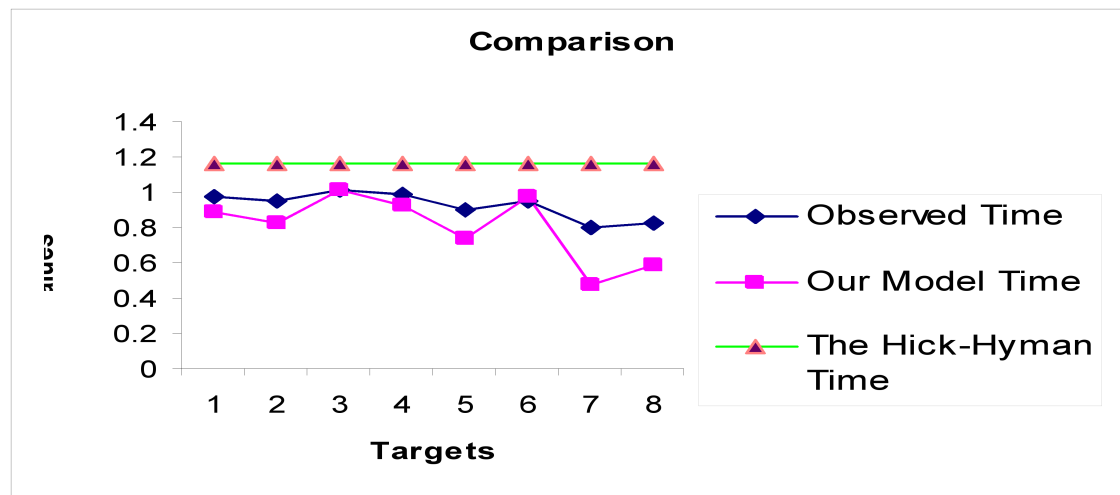
Model Validation



Targets	Observed Time	No. of Steps	Predicted Time
A	0.980	23	0.89
B	0.947	21	0.83
division	1.016	27	1.01
G	0.984	24	0.92
I	0.906	18	0.74
S	0.953	26	0.98
Z	0.805	9	0.47
Shift	0.828	13	0.59

Comparison Between Hick-Hyman Law & our Proposed Model

Targets	Observed time	Predicted time (The Hick-Hyman law)	% Error	Predicted time (Our model)	% Error
A	0.98	1.1563	17.9898	0.89	9.183673
B	0.947	1.1563	22.10137	0.83	12.3548
division	1.016	1.1563	13.80906	1.01	0.590551
G	0.984	1.1563	17.51016	0.92	6.504065
I	0.906	1.1563	27.62693	0.74	18.3223
S	0.953	1.1563	21.33263	0.98	2.833158
Z	0.805	1.1563	43.63975	0.47	41.61491
Shift	0.828	1.1563	39.64976	0.59	28.74396



Conclusion

- Comparison shows that our proposed model predicts visual search time more accurately than Hick-Hyman law.
- Some more factors to be considered like
 - Size of the groups
 - Size of the keys
 - Spacing between the groups
 - Number of groups