Designing an Interactive System for the Disabled Users

Samit Bhattacharya

Research Scholar

Dept. of Computer Science & Engineering

IIT Kharagpur

Assistive Technology (AT)

- Technology to assist individuals with disabilities to carry out various activities
- Who needs such technology
  - Visually impaired
  - Hearing impaired
  - Speech and motor impaired
  - Mentally retarded

How AT can help?

- Education
- Interpersonal communication
- Daily activities
- Entertainment
- Creativity
- Knowledge aquisition

HCI challenge

- Traditional I/O techniques may not be suitable
  - Sensory/motor requirements may not be present in the disabled user
- New interaction methods and techniques are required

Examples

- Text to speech synthesis-screen reader
- Speech recognition
- Braille printer
- Haptic and Tactile devices for input/output
- Voice output communication aids

AAC: Augmentative and Alternative Communication

- Communication aids for the speech and motor impaired
  - Cerebral Palsy
  - Muscular Dystrophy
  - Friedrich's Ataxia
  - Quadriplegia
- Alternate input methods
  - Alternate method of direct input (eye tracker, head tracker, head pointing)
  - Scanning or sequential input

AAC systems

- Icon based
- Text based
  - Character level text composition
    - 0.5-5 wpm
  - Word level
    - Compansion (10-15 wpm)
  - Storage and recall of pharses, sentences, paragraphs)
  - Conversational modeling (storage of scripts, schemata, frames)
    - >60 wpm

Character level systems

- Characterized by
  - Slow entry rate
  - Tedious
- But required for natural communication
  - Creation of novel and spontaneous statements during conversation
  - Off-line writing tasks, i.e. essays, stories, letters and messages

Soft or on-screen keyboards

- Keys are arranged in rows and columns
- Operated by
  - eye tracking
  - scanning input methods
- Text entry rate
  - 0.5-5 wpm
  - 6-8 wpm with rate enhancement techniques (prediction, ambiguity, abbreviation expansion)

Gesture driven systems

- Continuous gesture used for text composition
  - Trackball EdgeWrite: 6-8 wpm
  - Dasher: 25 wpm (with eye tracker)

Our Focus-
Scanning Keyboards

- Soft keyboards operated with scanning input methods

Our Focus-
Scanning Keyboards

- Soft keyboards operated with scanning input methods

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	**Enter**
Z	X	C	V	B	N	M	__	<--	**Shift**

Our Focus-
Scanning Keyboards

- Soft keyboards operated with scanning input methods

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	**Enter**
Z	X	C	V	B	N	M	__	<--	**Shift**

Our Focus-
Scanning Keyboards

- Soft keyboards operated with scanning input methods

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	**Enter**
Z	X	C	V	B	N	M	___	<--	**Shift**

Our Focus-
Scanning Keyboards

- Soft keyboards operated with scanning input methods

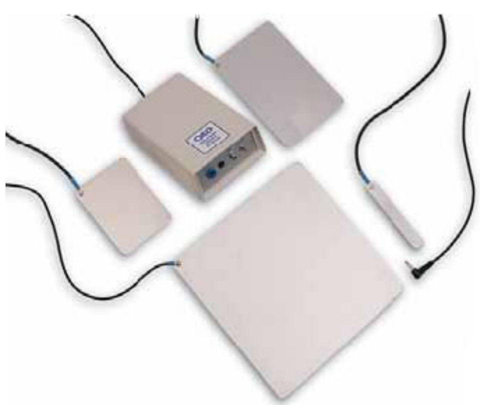
Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	**Enter**
Z	X	C	V	B	N	M	___	<--	**Shift**

Our Focus-
Scanning Keyboards

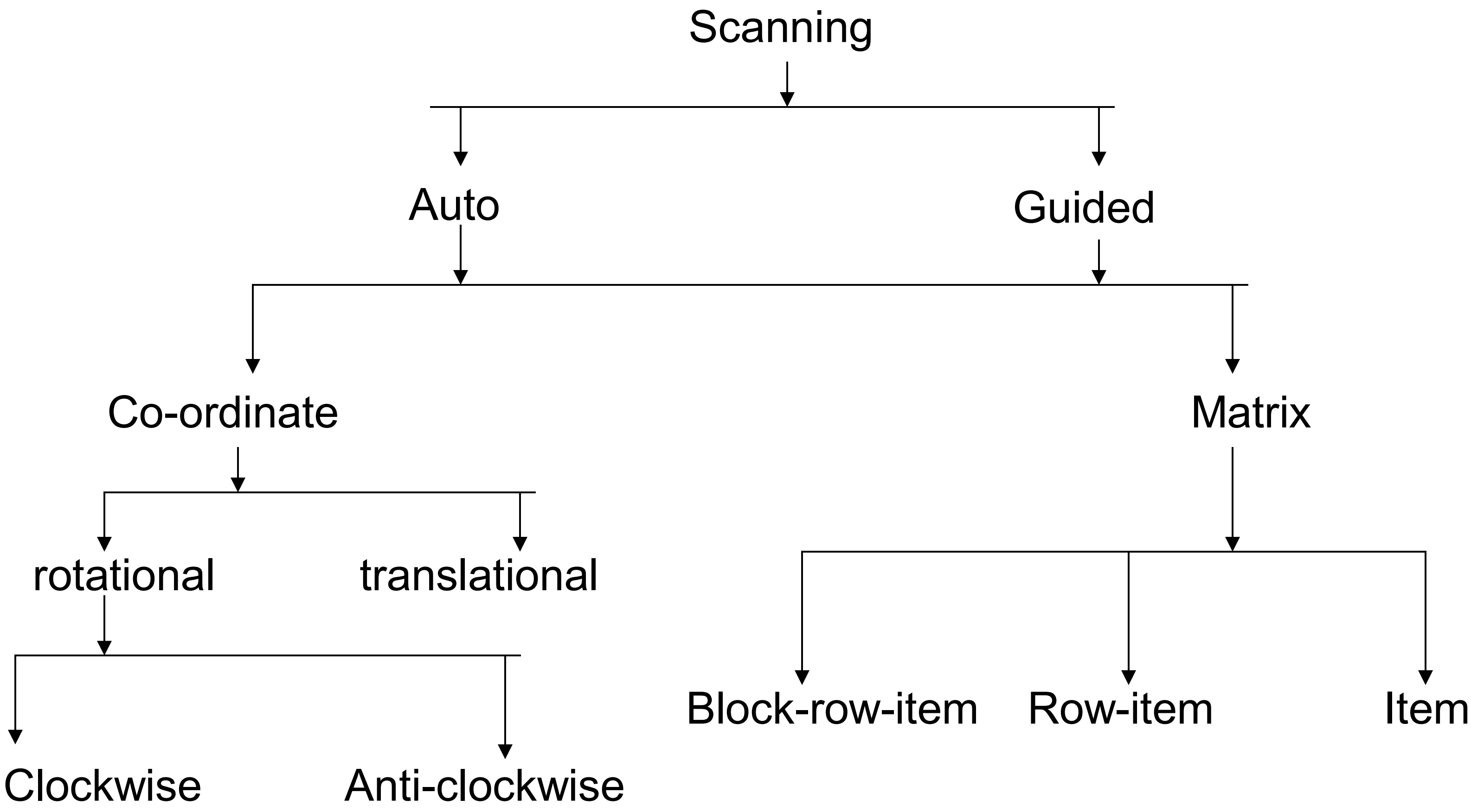
- Soft keyboards operated with scanning input methods

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	**Enter**
Z	X	C	V	B	N	M	___	<--	**Shift**

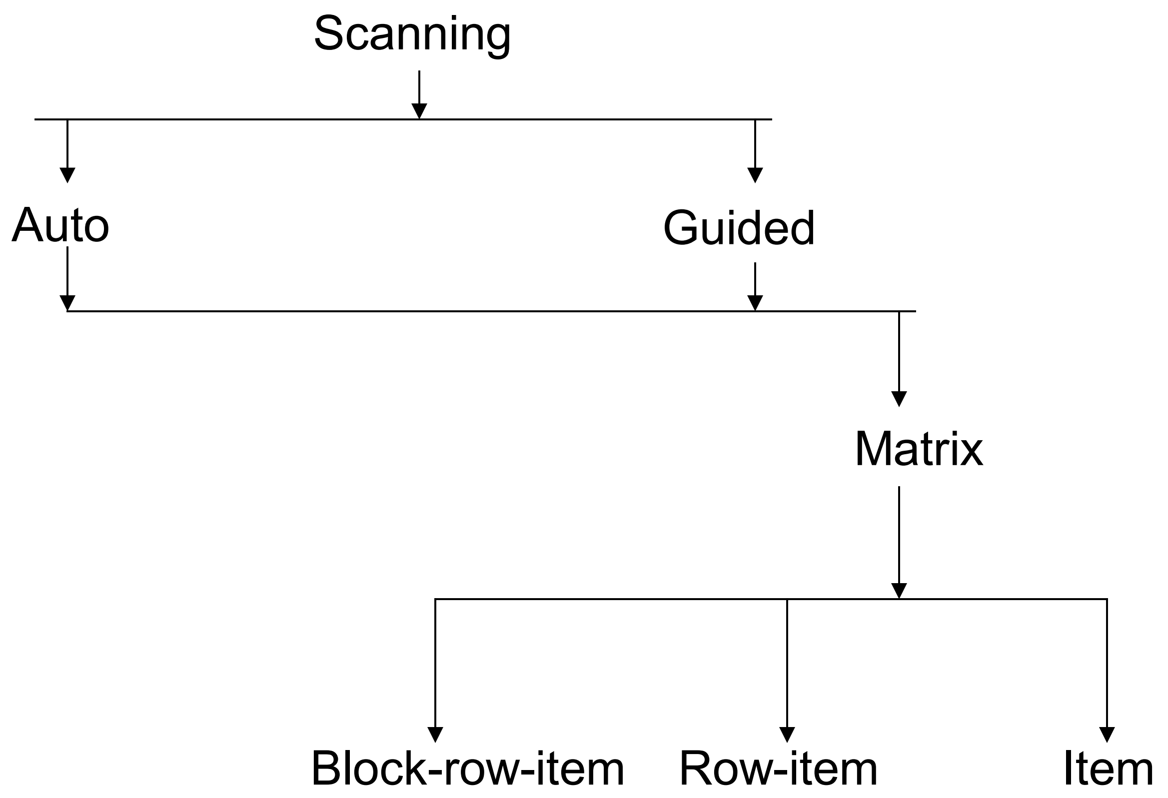
Input Devices



Scanning types



Scanning types



Design Challenge

- Large design space
  - With 27 keys, 27! Possible layouts
  - Each can be operated with either of the scanning methods
- How to choose a design that optimizes user performance?

Standard Design Method

- Reduce size of the design space based on experience and intuition
- Implement prototypes of the remaining designs
- Test prototypes with disabled users to determine the best

Problems

- Difficult to get disabled users for testing prototypes
  - Social pressure
  - Lack of exposure to computers
- Difficult to collect data large enough for analysis
  - Testing is physically demanding
  - Disabled users can not continue for long at a stretch
  - Data collection is slow

Model Based Design

- Evaluation of designs with user/performance models
  - Fast
  - Can be automated
  - Does not require user testing
  - Design space can be searched using the models
    - Removes dependency on designer's expertise to reduce design space

Performance Models

- The RG model by Rosen and Goodenough-Trepagnier (1981)
- Based on three components
  - L -- average no of language units per word
  - A -- average no of motor acts required to input each language unit
  - T – average time required to carry out each motor act
- T_w = average time to compose an word = L*A*T

Performance Models contd…

- Levine and Goodenough-Trepagnier (1990)
- Three performance models based on the RG model
  - Unambiguous keyboards
  - Soft keyboard with character encoding
  - Ambiguous keyboards

Performance Models contd…

- RG model considered only direct input, not scanning
- Damper (1984) extended the RG model for scanning keyboards
- According to Damper

$$T_W = \frac{L}{R}\sum_{i=1}^{K} p_i(s_i + 1)$$

L = as before

R = scan rate

Pi = unigram char probability

Si = no of scan steps from a home position

Performance Models contd…

- GOMS model by Horstmann and Levine (1990)
- KLM model by Koester and Levine (1994, 1997, 1998)
- Only interactions with direct input methods were modeled

Performance Models contd…

- The FD Model – Model for able-bodied users of soft keyboards (MacKenzie & Soukoreff, 1995; Soukoreff & MacKenzie, 2002; Zhai et al. 2002)
- Three components
  - Visual search time -- Hick-Hyman law
  - Movement time -- Fitts' law
  - Digraph probability -- from corpus

Performance Models contd…

- Movement time

$$MT_{ij} = a + b\log_2(d_{ij}/w_j + 1)$$

- Visual search time

$$RT = a' + b'\log_2 N$$

- Digraph probability

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

Performance Models contd…

- 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)

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

Comparison

- KLM/GOMS models require task description—inputting a string of characters
  - Tedious
  - Desirable that the models do not take task description as input--RG and FD model are more suitable
- Damper’s extended RG model
  - Considers only a particular scanning type
- FD model, appropriately modified, could alleviate these problems

Automatic Design Space Search

- Getschow et al. (1986) – greedy algorithm
  - Not very efficient
- Adaptive evolutionary search – Levine & Goodenough-Trepagnier (1990)
  - RG model for selection from a generation
- Dynamic Simulation, Metropolis algorithm (Zhai et al. 2002), Genetic algorithm (Raynal & Vigouroux, 2005)
  - FD model for selection

FD Model-Limitations

- Highlighter movement time instead of manual movement
- Switch input
- User errors

- We have addressed these issues in our work

Modeling Scanning Interaction

- Replace Fitts’ law with focus movement and selection time (FT)
- Assumptions
  - Each key holds single character
  - No prediction
  - Focus returns to the current block/row/item after each selection
  - No errors
- Let there are two keys: K=<b,r,c>, k’=<b’,r’,c’>
- Events between selection of k’ after k

Auto Scanning Events

Event	Notation	Time
b is highlighted again	FOC()	System dependent
Focus moves from *b* to *b’*	MOV(b,b’)	[B+(b’-b)]T: b’<b (b’-b)T: b’≤b
User activates switch to select *b’*	SEL(b’)	T/2
Row level scanning in *b’* starts	FOC()	System dependent
Focus moves from the first row to *r’*	MOV(r1,r’)	(r’-1)T
User activates switch to select *r’*	SEL(r’)	T/2
Item level scanning in *r’* starts	FOC()	System dependent
Focus moves from the first item to *c’*	MOV(c1,c’)	(c’-1)T
User selects *c’* once *c’* is focused	SEL(c’)	T/2

FT for Auto Scanning

- Sum of the individual event times

$$FT(k,k') = T_{so} + (X + C) \times T$$

T_{so}	Total time for three FOC() events
X	(b'+r'+c')-b
C	-0.5 b'≥b; (B-0.5) b'<b;

Guided Scanning Events

Event	Notation	Time
b is highlighted again	FOC()	System dependent
User shifts focus from *b* to *b'*	SFT(b,b')	$[B+(b'-b)][T_{GS}+FOC()]$: $b'<b$ $(b'-b)[T_{GS}+FOC()]$: $b'\leq b$
User activates switch to select *b'*	SEL(b')	T_{GS}
Row level scanning in *b'* starts	FOC()	System dependent
User shifts focus from the first row to *r'*	SFT(r1,r')	$(r'-1)[T_{GS}+FOC()]$
User activates switch to select *r'*	SEL(r')	T_{GS}
Item level scanning in *r'* starts	FOC()	System dependent
User shifts focus from the first item to *c'*	SFT(c1,c')	$(c'-1)[T_{GS}+FOC()]$
User selects *c'* once *c'* is focused	SEL(c')	T_{GS}

FT for Guided Scanning

- Sum of the individual event times

$$FT(k,k') = (X + C) \times (T_{GS} + FOC())$$

X	(b’+r’+c’)-b
C	1 b’≥b; (B+1) b’<b;

Calculation of T_{GS}

- Keates et al. (2000) proposed five steps for switch activation
  - Perceive focusing (perception) (100 ms, Card et al., 1983)
  - Decide to activate switch (cognition) (84 ms, Keates et al., 2000)
  - Activate switch (motor act) (105 ms, Keates et al., 2000)
  - Decide to deactivate switch (cognition) (84 ms, Keates et al., 2000)
  - Deactivate switch (motor act) (105 ms, Keates et al., 2000)
- $T_{GS} = 100+2(105+84) = 478$ ms

User Study

- Eight interfaces
- Two layouts
- Four types of scanning on each layout
  - 3-level auto and guided scanning
  - 2-level auto and guided scanning
- Eight subjects
  - Six with disabilities
  - Two without disabilities

Interfaces

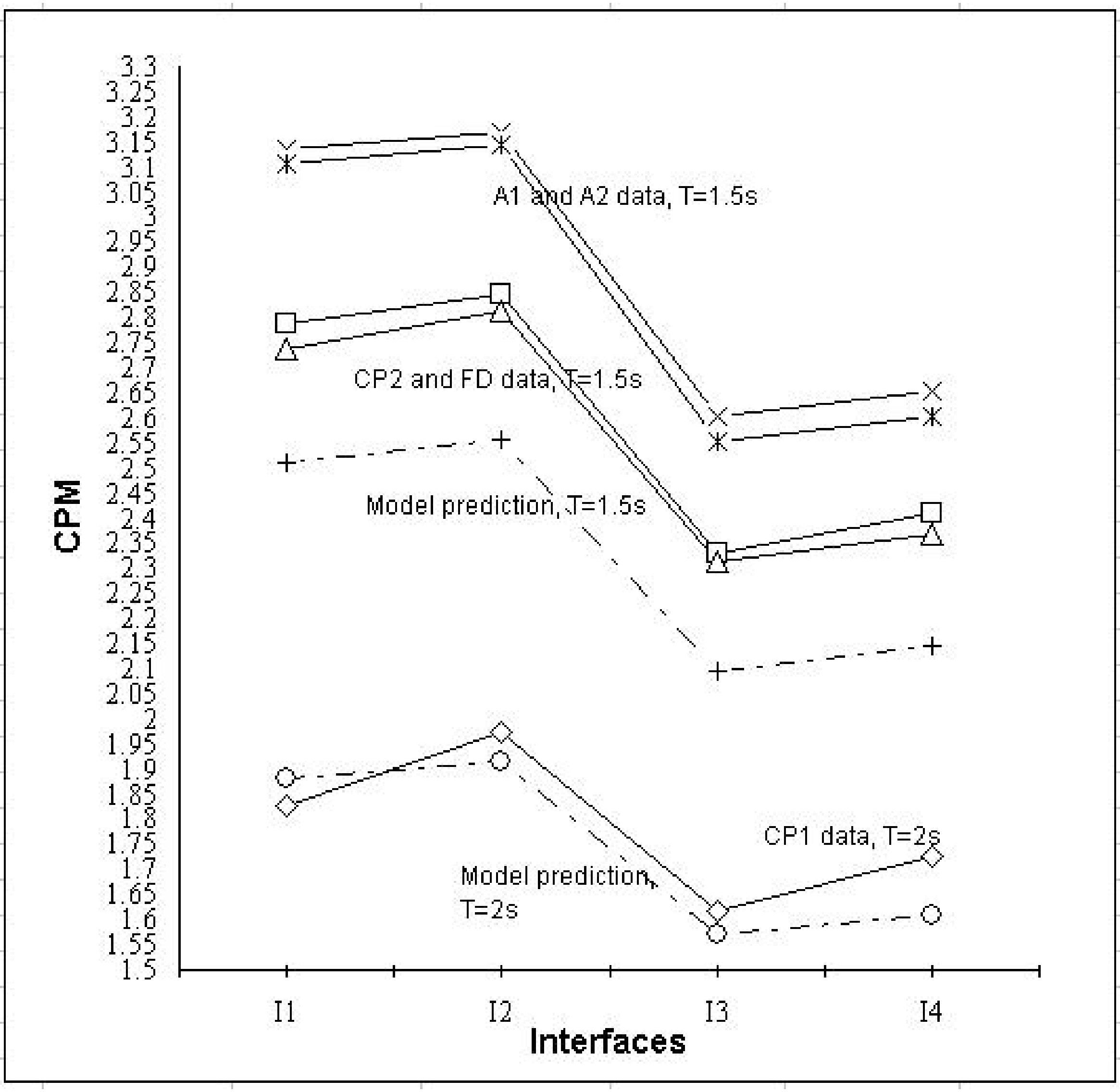
I1,I3,I5,I7

I2,I4,I6,I8

Resources

- Digraph prob. table for Bengali (size = 104*104 including non-alphabetic pairs like "Enter-Space")
- Average word length in Bengali (6 chars including space)
- Text *chunk* for data collection (630 chars)
- All the above from "Anandabazar" corpus
  - 96,012,779 characters

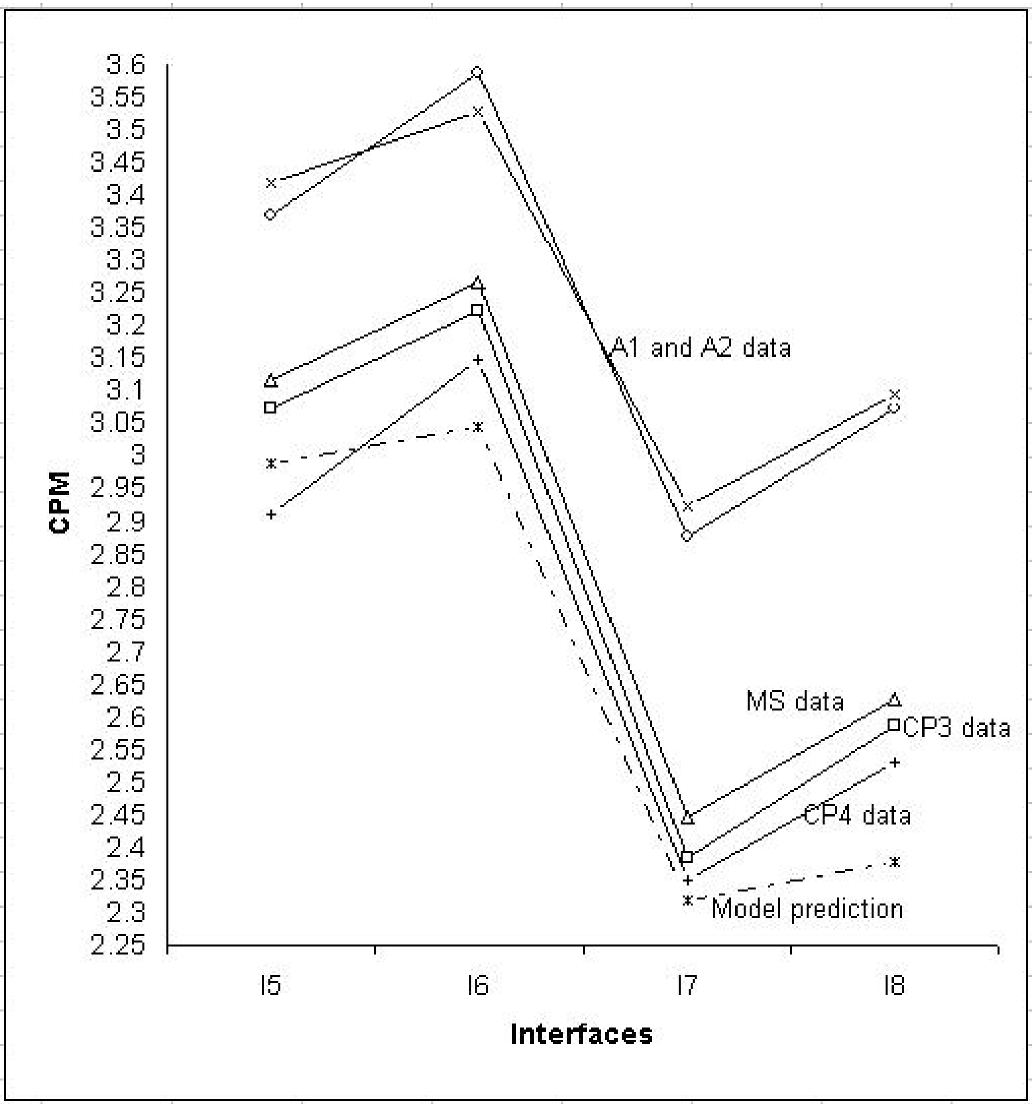
Results (Auto scanning)



Tso = 3*T for I1,I3

= 2*T for I2,I4

Results (Guided scanning)



$T_{GS} = 478$ ms

Discussion

- Difference between model and observations
  - Auto scanning - 5-10%
  - Guided scanning – 2-8%
- Reason??
  - SEL() = T/2
  - T_{GS} = 478 ms
  - Five step switch activation model
  - Visual search

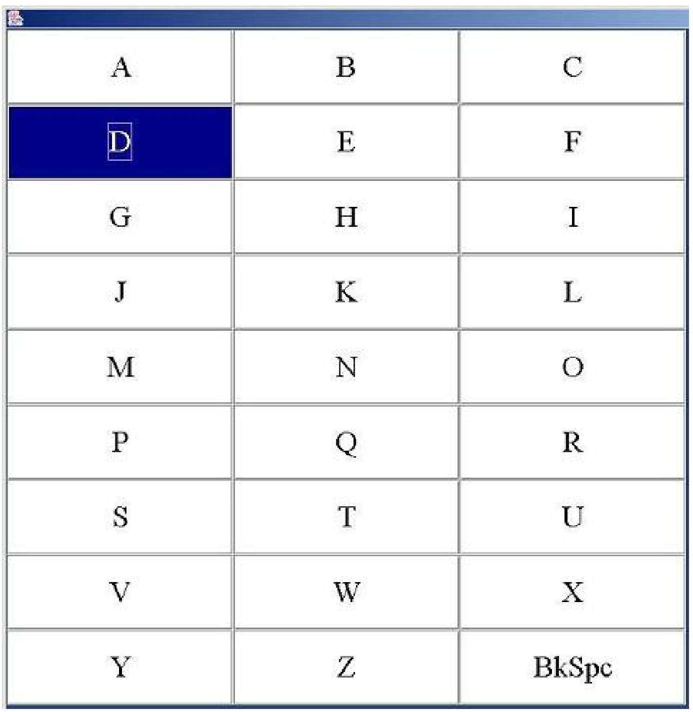
Error Study and Modeling: Background

- Trewin and Pine (1998)
  - Direct input methods
- Performance models do not take into account the effect of errors
- Reason: lack of data
- Result: limited practical usefulness of resulting designs

User Study

- Two layouts
  - Alphabetic organization
  - Single character each keys
  - No prediction
- 3-level, 2-level and 1-level auto scanning on each
- Six subjects with disabilities
- Printed texts of about 1000 characters for entry

Layouts


A B C
D E F
G H I
J K L
M N O
P Q R
S T U
V W X
Y Z BkSpc


অ আ ই ঈ উ ঊ
ঋ এ ঐ ও ঔ
ক খ গ ঘ ঙ
চ ছ জ ঝ ঞ
ট ঠ ড ঢ ণ
ত থ দ ধ ন
প ফ ব ভ ম
শ ষ স হ
য র ল ঢ় ড়
য় ৎ ং ঃ
১ ২ ৩ ৪ ৫
৬ ৭ ৮ ৯ ০
, । ? !
space bksp jkAsr ent

Experimental Method

- Two groups of experiments
- First group (English layout with three scanning types)
  - for data collection and model development
- Second group (Bengali layout with three scanning types)
  - validating results of first group

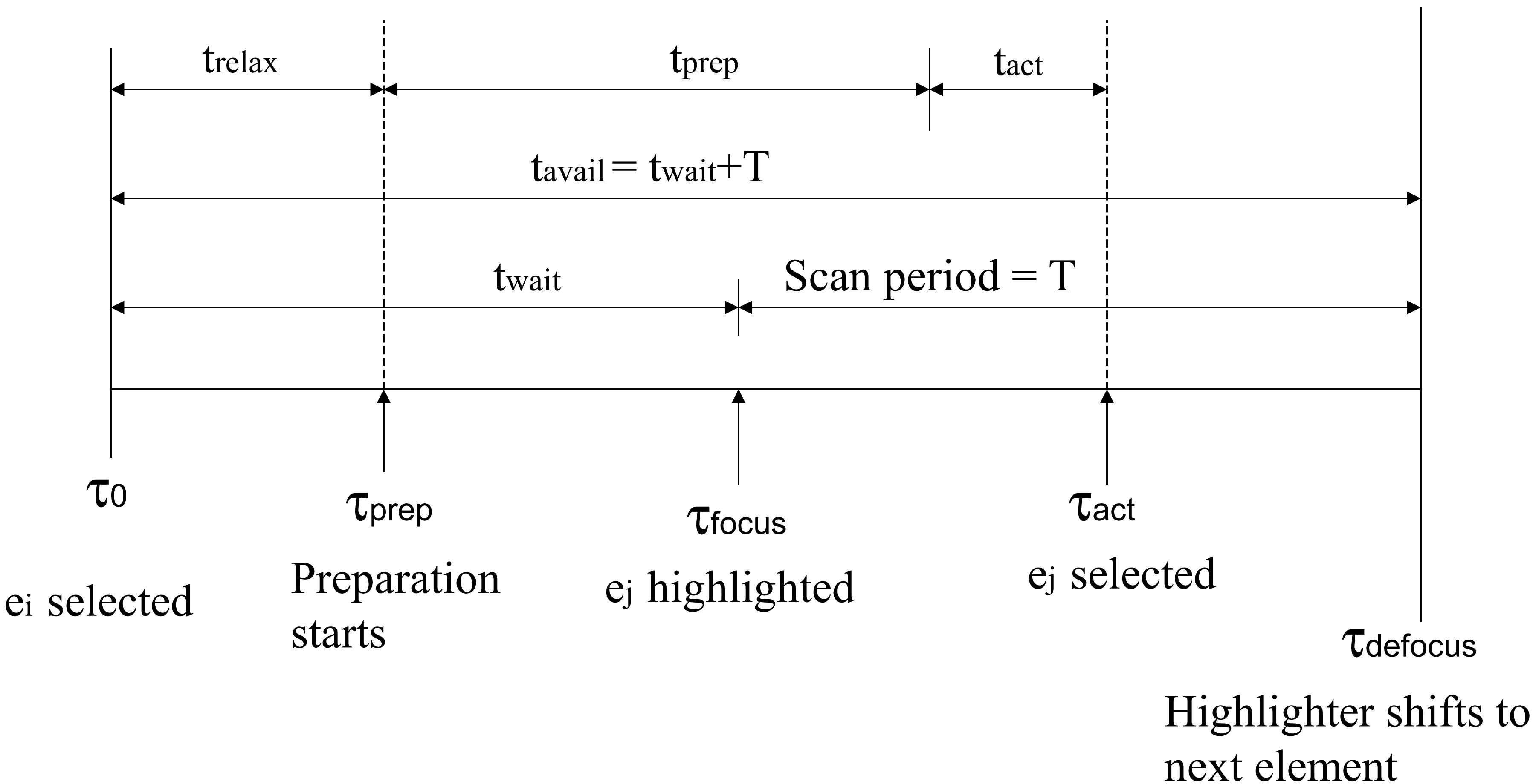
Observation

- Three types of errors
  - Timing errors (TE)
  - Selection errors (SE)
  - Transcription errors
- Transcription errors very rare and its effect can be ignored

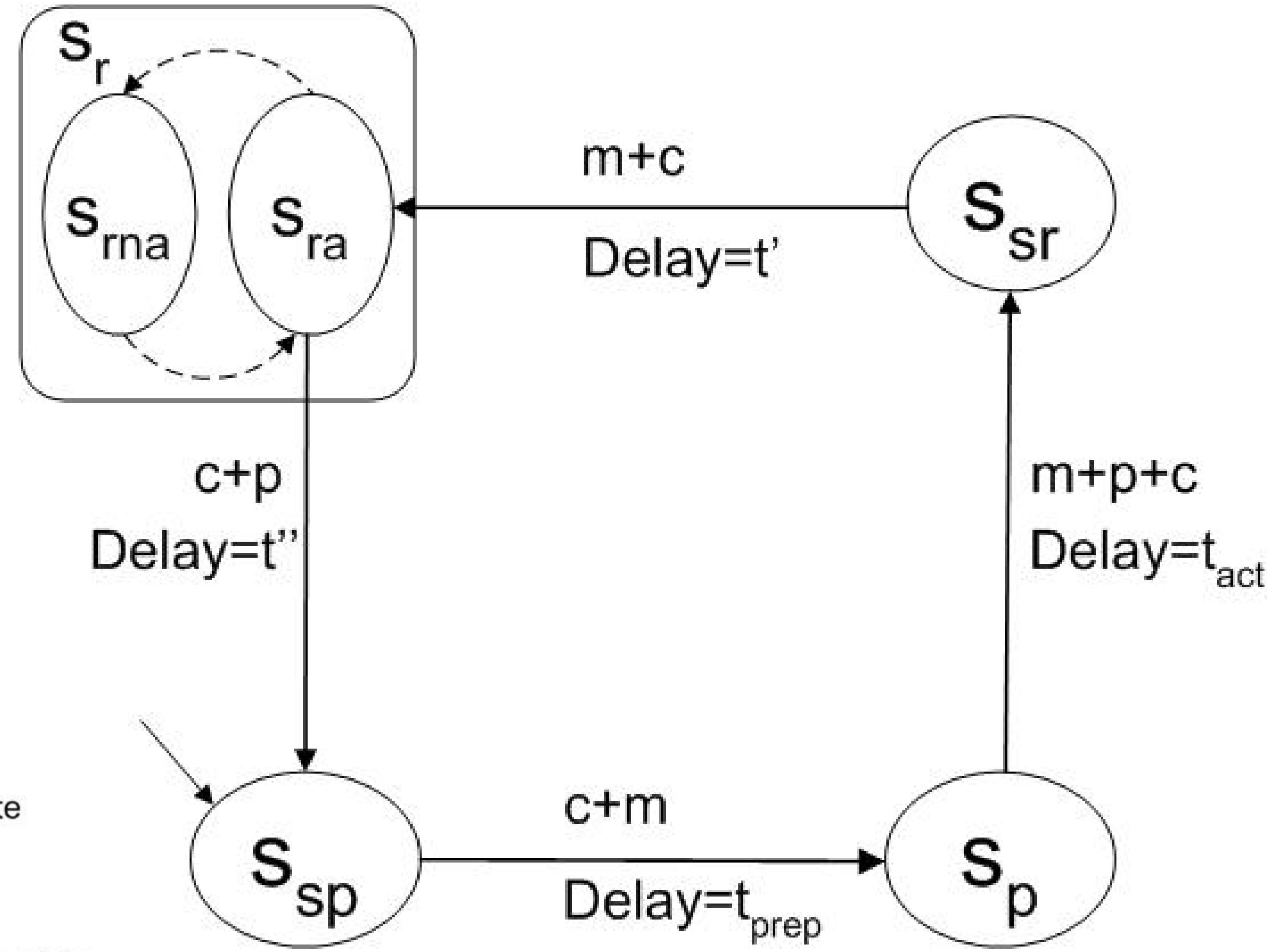
Effect of TE and SE

- Analyzed usage logs of the six subjects
- Increase in text entry time due to TE
  - 65% (approx) for 3-level
  - 45% (approx) for 2-level
  - 35% (approx) for 1-level
- Increase in text entry time due to SE
  - 35% (approx) for 3-level
  - 25% (approx) for 2-level
  - 15% (approx) for 1-level

Temporal Model of User Behavior



Finite State Model

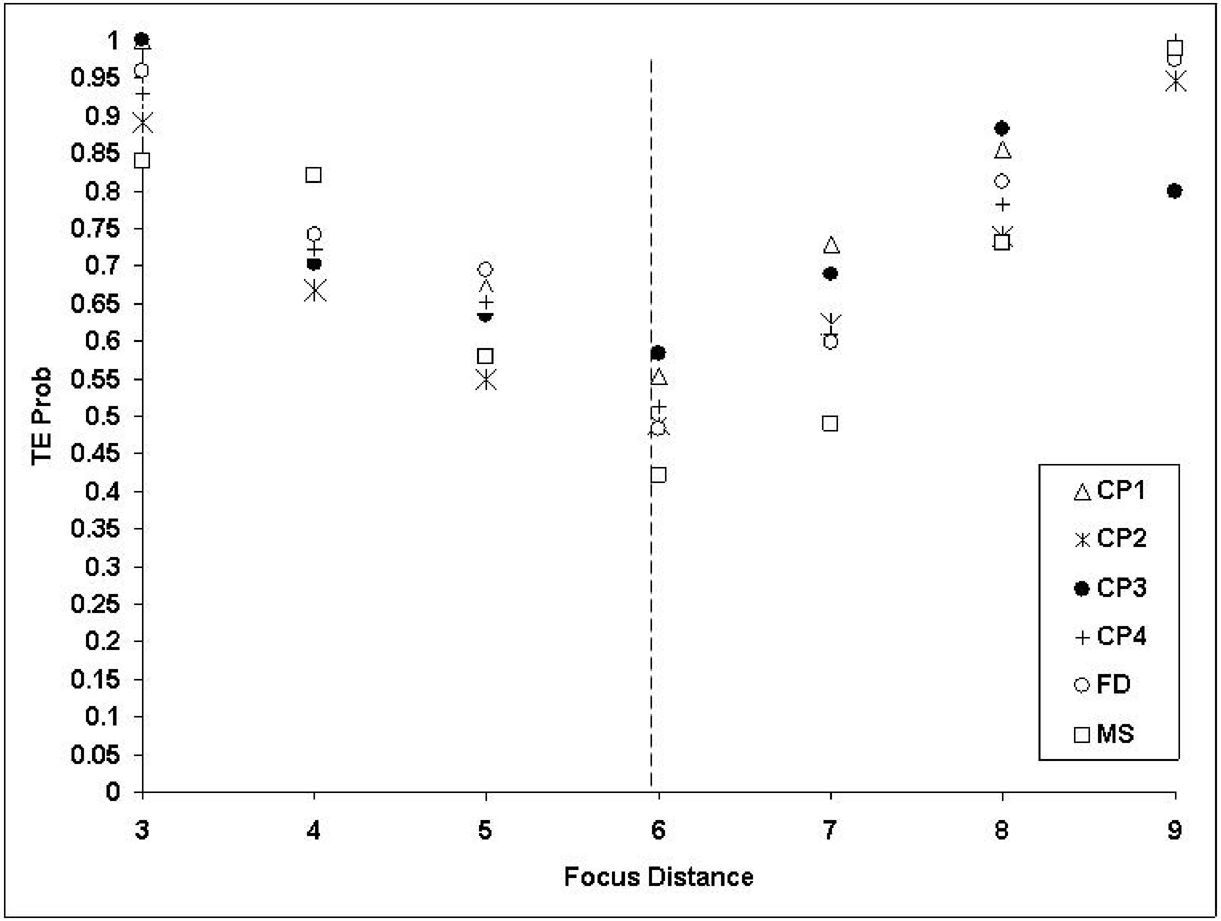


S_{sp}=*Start preparation* state
S_p=*Prepared* state
S_{sr}=*Start relaxing* state
S_r=*Relaxed* state
S_{ra}=*Relaxed and attentive* state
S_{rna}=*Relaxed but not attentive* state

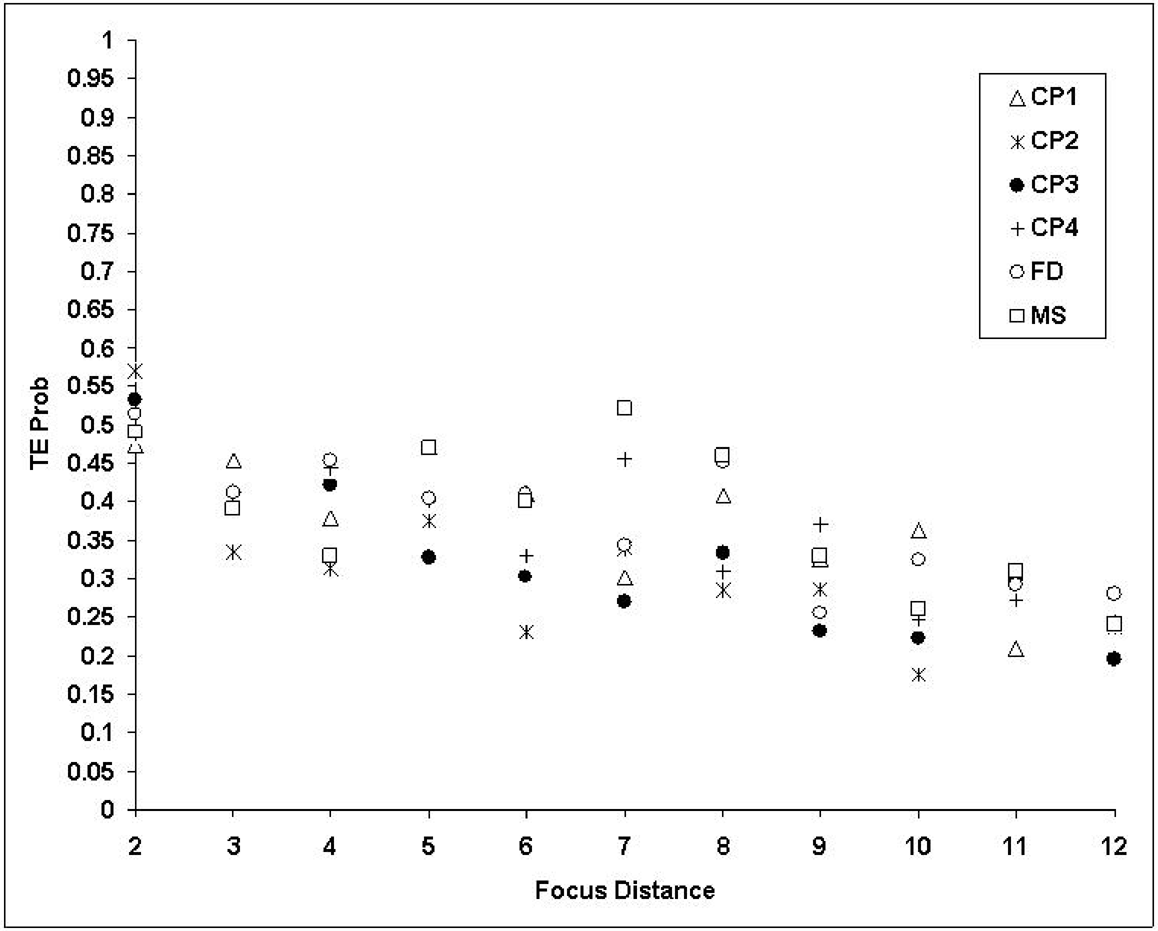
Model Prediction

- Focus distance between two elements
  - Number of highlighter shifts + switch activations
- Each scanning keyboard has
  - A minimum focus distance, f_{min}
  - A maximum focus distance, f_{max}
- User model predicts that
  - At f_{min}, high TE probability
  - TE prob. decreases till a critical focus distance, f_c
  - Then, it increases again till f_{max}
- No such pattern for SE, random in nature

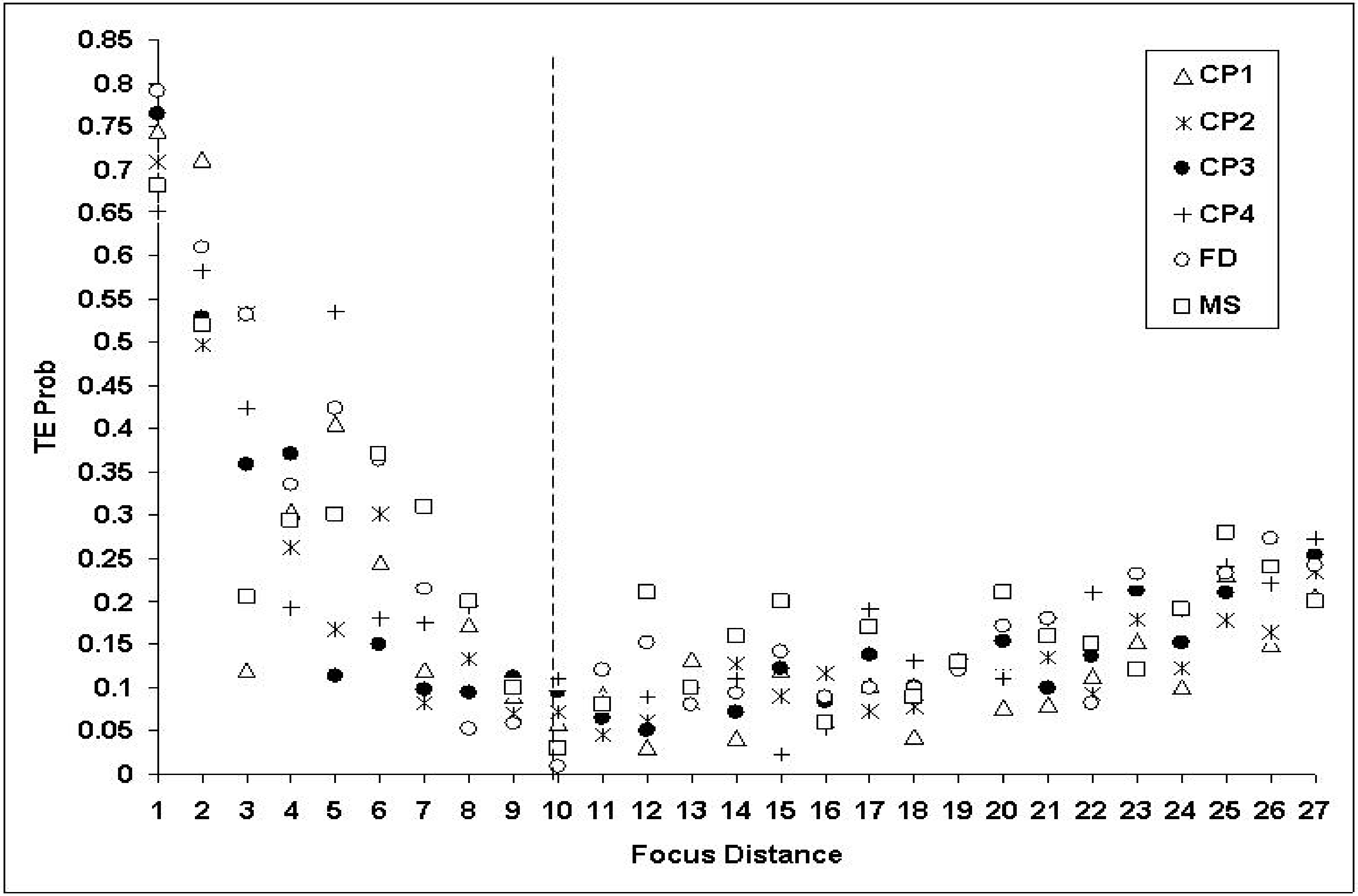
Observation: 3-level TE Dist.



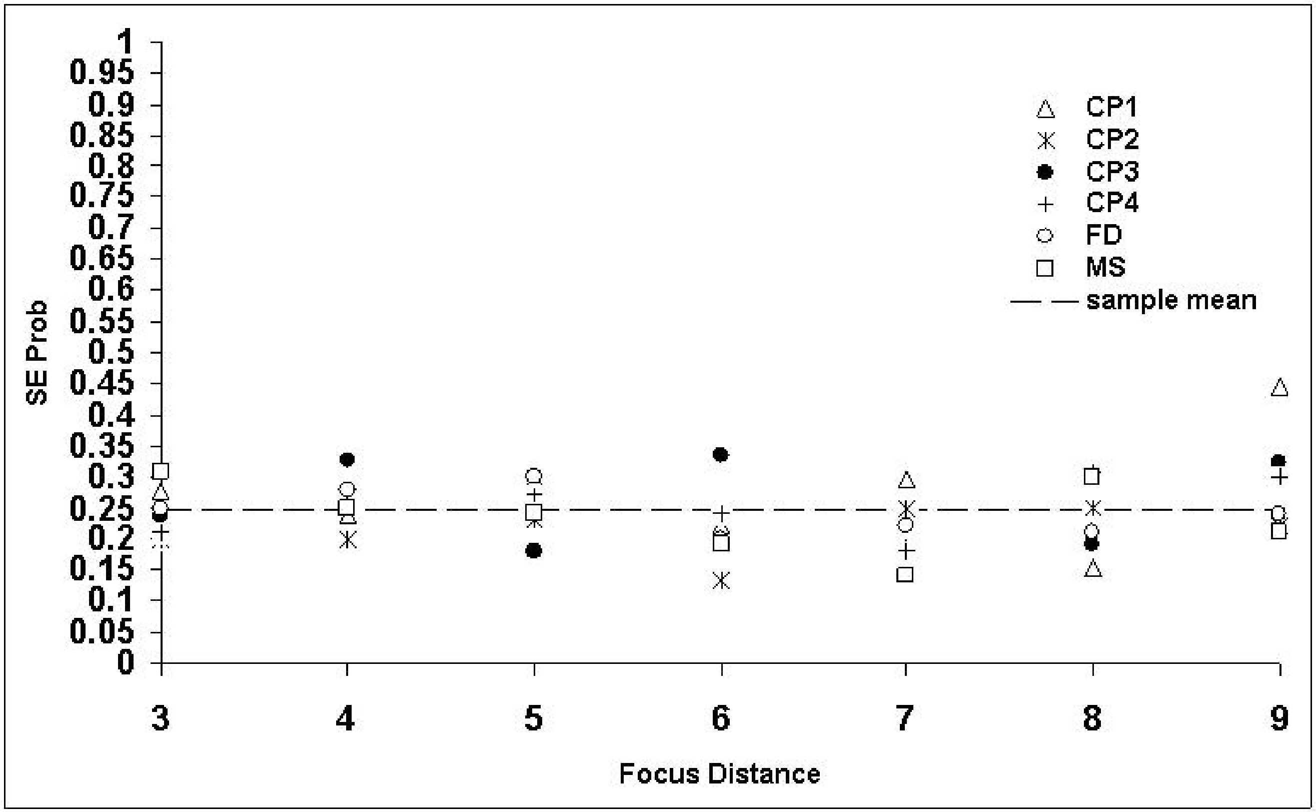
Observation: 2-level TE Dist.



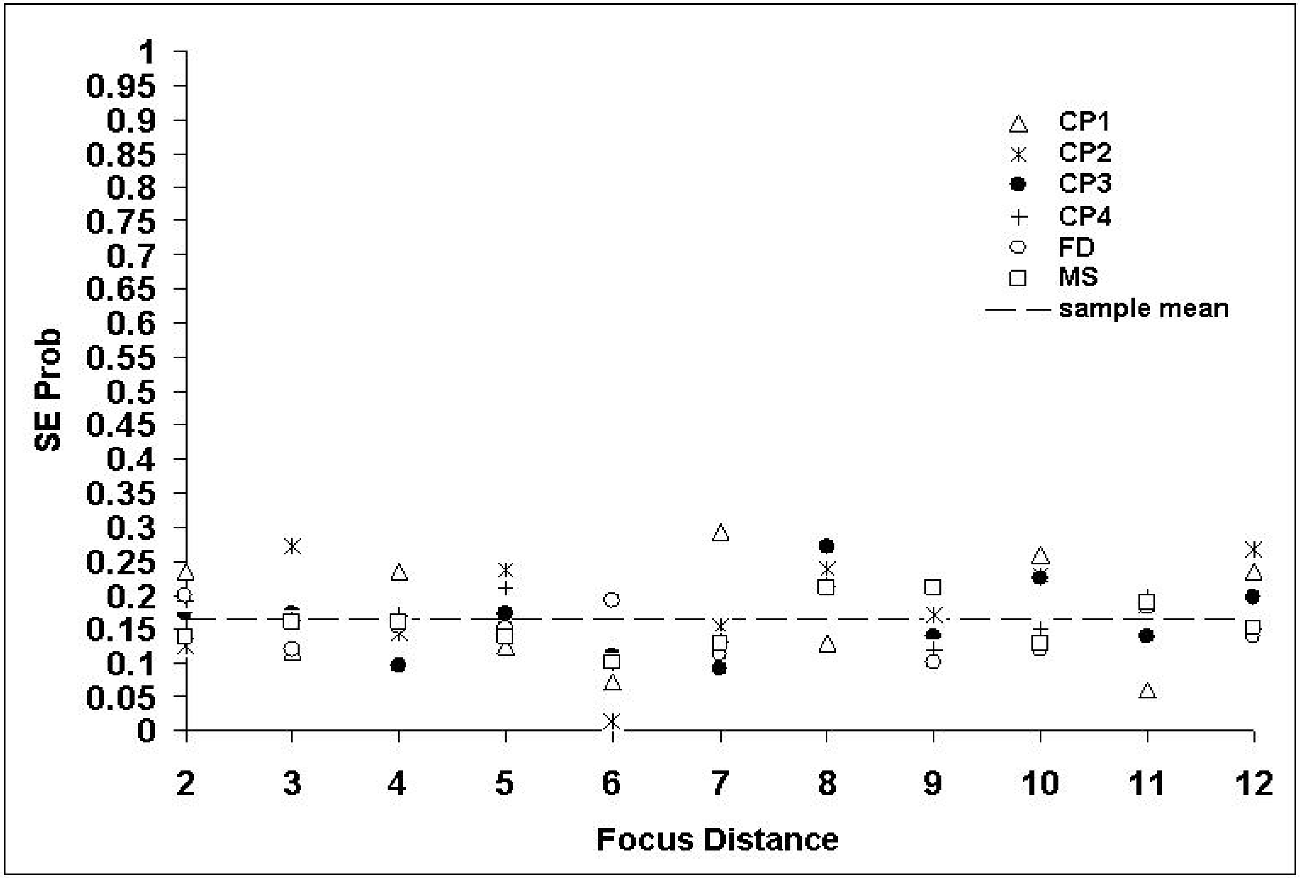
Observation: 1-level TE Dist.



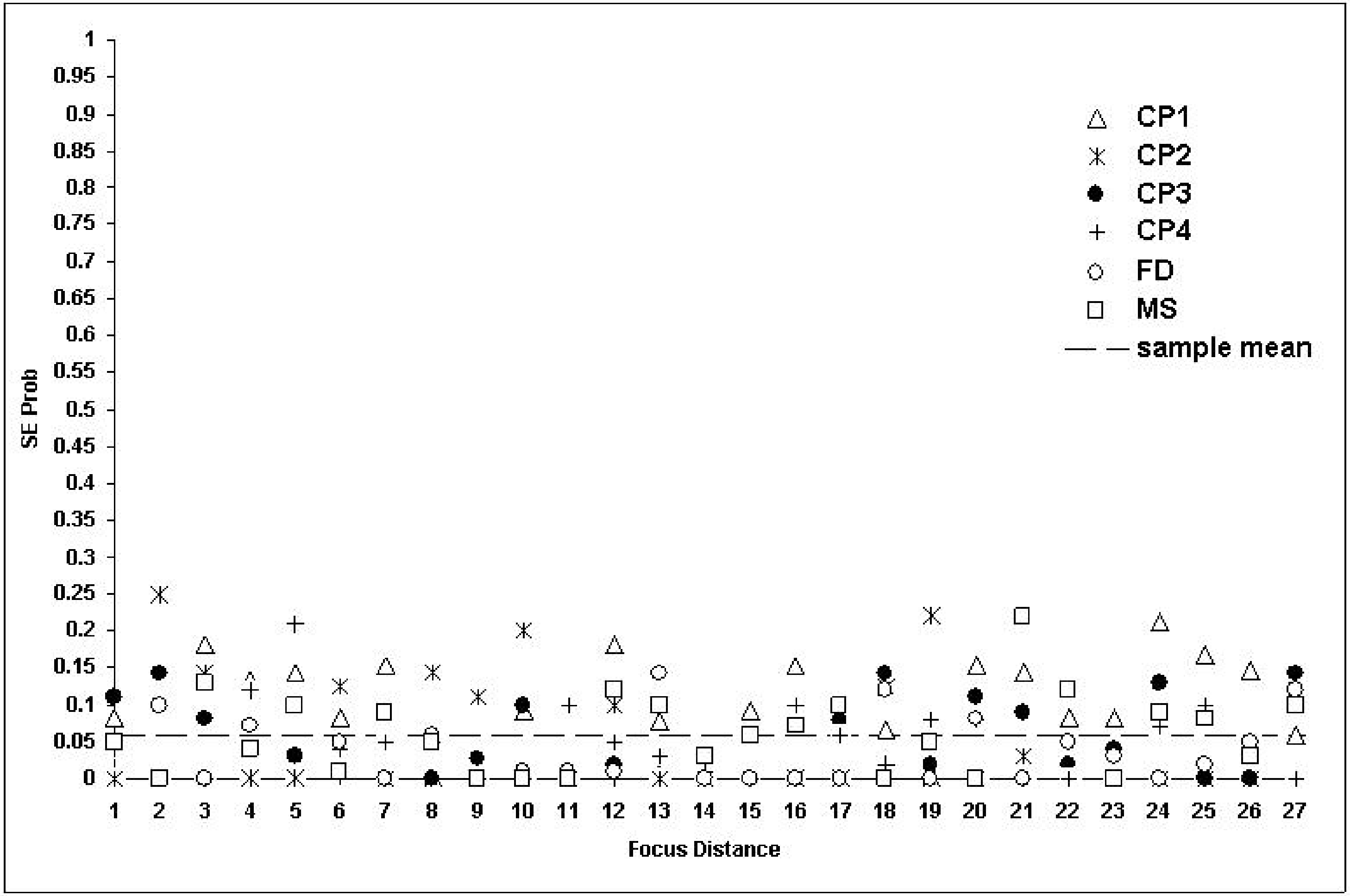
Observation: 3-level SE Dist.



Observation: 2-level SE Dist.



Observation: 1-level SE Dist.



Model Implication: Design Principles

- To reduce error, frequently selected char pairs should be placed apart by
  - $f_{min}+R_f/2$ for 3-level
  - $f_{min}+R_f/3$ for 1-level
  - f_{max} for 2-level
- Effect on text entry rate?
  - Interviewed subjects
  - They preferred high text entry rate if error prob is low (1 and 2-level), reduced error if error prob is high (3-level)
- Principle important for 3-level, apply with care for 1 and 2-levels

Model Implication: Computational Model

- Distribution function for TE

$$
\begin{aligned}
& P_{TE}(f) \\
& = P_0.e^{-\lambda_0(f - f_{\min})} \quad f_{\min} \le f \le f_c \\
& = P_1.e^{\lambda_1(f - f_c)} \quad f_c < f \le f_{\max}
\end{aligned}
$$

- SE modeled with sample mean since no pattern

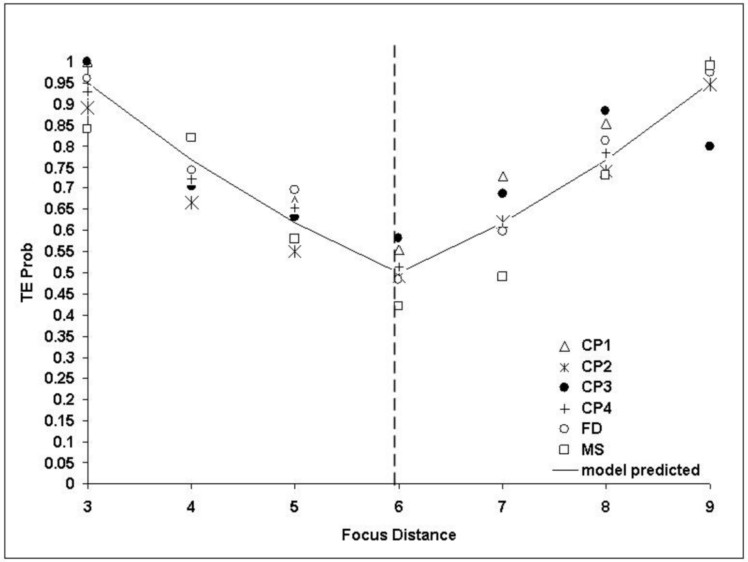
Computational Model

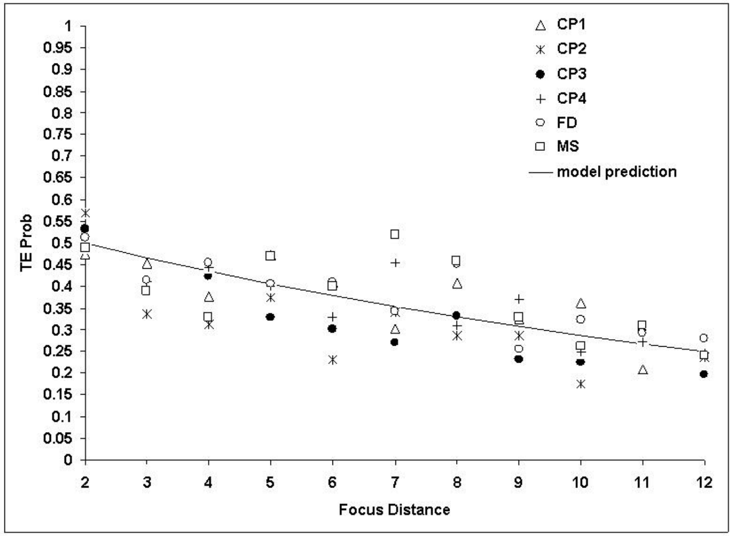
- Four parameters for TE
  - P_0, TE prob at f_{min}
  - f_c, the critical focus distance
  - P_1, TE prob at f_c
  - P_2, TE prob at f_{max}
- One parameter for SE, sample mean or P_m
- We have estimated their values from empirical data

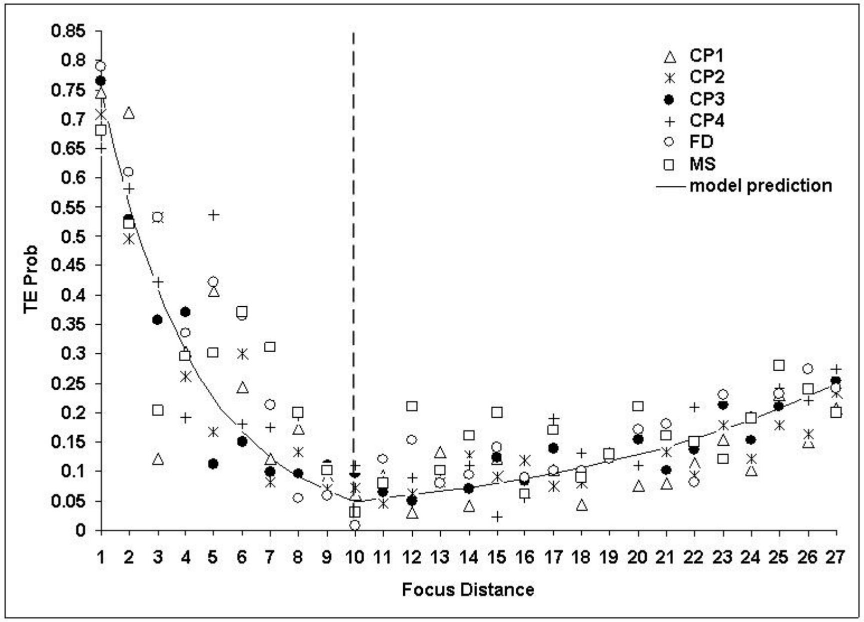
Parameter Values

Scanning type	Parameter values
3-level	$P_0 \approx 0.95$, $P_1 \approx 0.5$, $P_2 \approx 0.95$ $f_c \approx f_{min} + R_f/2$ $\lambda_0 = (1/(R_f/2))\ln 2$, $\lambda_1 = (1/(R_f-1)-R_f/2))\ln 2$ $P_m \approx 0.25$
2-level	$P0 \approx 0.5$, $P1 = 0$, $P2 \approx 0.5$ $f_c \approx f_{max}$ $\lambda_0 = (1/(R_f-1))\ln 2$ $P_m \approx 0.15$
1-level	$P_0 \approx 0.75$, $P_1 \approx 0.05$, $P_2 \approx 0.25$ $f_c \approx f_{min} + R_f/3$ $\lambda_0 = (1/(R_f/3))\ln 15$, $\lambda_1=(1/((R_f-1)-R_f/3))\ln 5$ $P_m \approx 0.05$

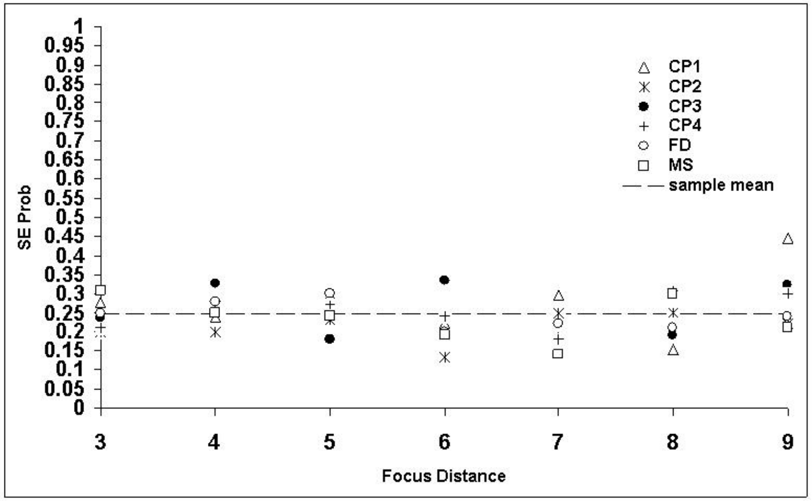
Observation: TE Dist.

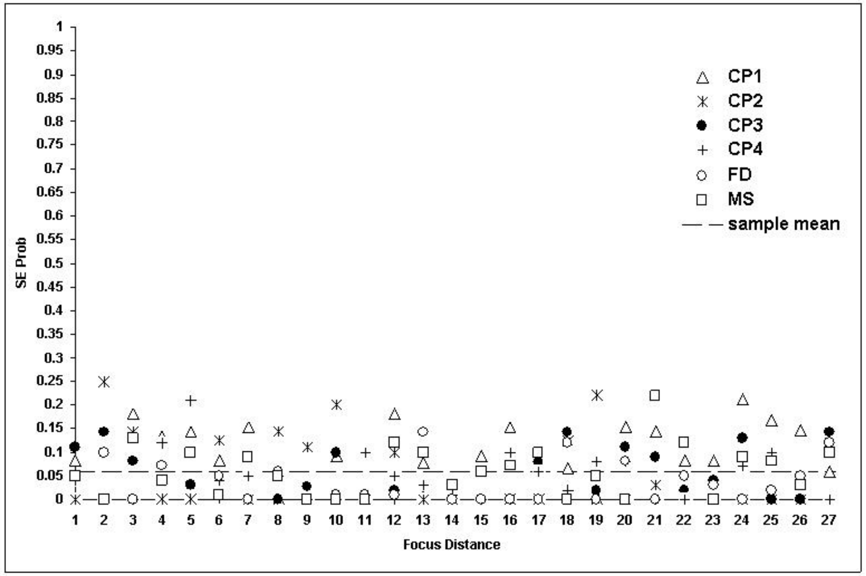


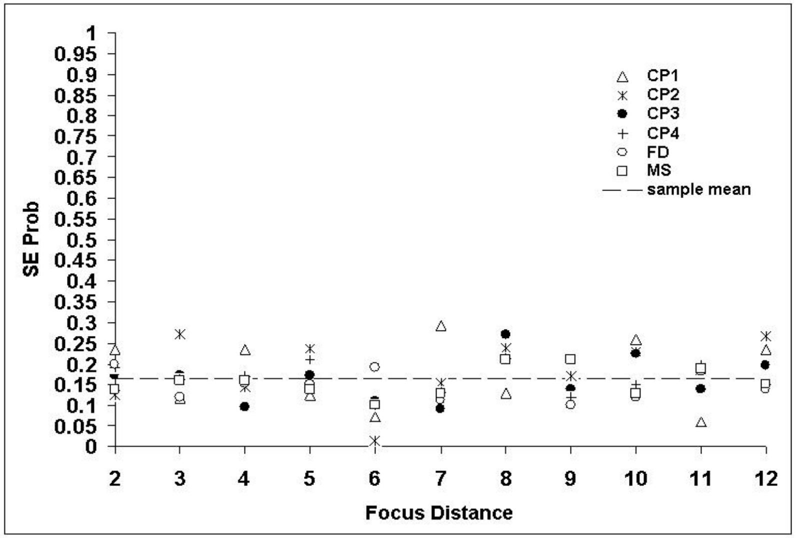




Observation: SE Dist.







The ErrorProneness (EP) Measure

- A numerical measure of the effect of errors for scanning keyboards
- Developed from the distribution functions
- EP of a scanning keyboards
  - Joint prob. of TE and SE
  - The higher the EP, the less the keyboards ability to prevent errors

EP Calculation

- Calculate average focus distance, f_{mean}

$$f_{mean} = \sum_{i=1}^{K} \sum_{j=1}^{K} DP_{ij} \times f_{ij}$$

- Calculate joint error prob for f_{mean}, assuming mutual independence

$$EP = P_m \times P_{TE}(f_{mean})$$

Comparing Interfaces

- Let we have a set of interfaces
- Compute the following for each interface
  - Error free text entry rate (t)
  - EP (e)
- Compare the interfaces based on these two measures

Relationship between two interfaces (s1, s2)

Notation	Relation
r1	t1<t2, e1<e2
r2	t1<t2, e1=e2
r3	t1<t2, e1>e2
r4	t1=t2, e1<e2
r5	t1=t2, e1=e2
r6	t1=t2, e1>e2
r7	t1>t2, e1<e2
r8	t1>t2, e1=e2
r9	t1>t2, e1>e2

Choosing the Better

- s1 better than s2 for r4, r7, r8
- s2 better than s1 for r2, r3, r6
- They are equal for r5
- For 3-level scanning, s1 better than s2 for r1 and vice-versa for r9
- For 1-level and 2-level, s1 better than s2 for r9 and vice-versa for r1

Design Space Search

- Previous method requires designer's expertise
- Solution: search design space with algorithm
- We want an algorithm that
  - Maximizes error free text entry rate
  - Minimizes error probability
- Associated problem – optimal grouping of keys for multi-level scanning

Optimal Key Grouping

- Extended the work of Foulds et al. (1987)
- Uses a modified definition of focus distance as
  - Total shifts and switch activations starting from first block (3-level) or first row (2-level)
- Focus distance of a key
  - 3-level, k (b,r,c) = b+r+c
  - 2-level, k (r,c) = r+c
- Minimizes total focus distance of a layout

Algorithm

Set $B_{MIN} = R_{MIN} = C_{MIN} = 2$, $B_{MAX} = R_{MAX} = C_{MAX} = \lceil K/4 \rceil$, $i = 1$

for $b = B_{MIN}$ to B_{MAX} **do**

  for $r = R_{MIN}$ to R_{MAX} **do**

    for $c = C_{MIN}$ to C_{MAX} **do**

      if $b \times r \times c$ K **then**

        generate layout L[i] with b, r and c numbers of blocks, rows and items respectively

      end if

      $i = i + 1$

    end for

  end for

end for

$i = 1$

while $i < \text{length}(L)$ **do**

  Calculate focus distance for each key in L[i]

  Sort keys in non-decreasing order of their focus distance

  Choose first K keys from this sorted list as the position of the characters in L[i]

  Calculate cost of the layout L[i]

  $i = i + 1$

end while

Select the layout with minimum cost from L

The Search Algorithm

- For simplicity, (1/text entry rate) was taken as one of the optimization criteria
  - Transformed to minimization problem
- Starts with a random layout; grouping algorithm used for multi-level scanning
- Initial temperature, $T_0 = -\Delta E / \ln P_0$
- T is decreased by a factor α (the cooling rate) till some minimum value

Acceptance Probabilities

- Both the measures are worse

$$P_1 = 1/(1+e^{((\Delta_1/(1/R')+\Delta_2/E')/T_i)})$$

- Text entry rate better

$$P_2 = 1/(1+e^{(\Delta_1/(1/R'))/T_i})$$

- EP is better

$$P_3 = 1/(1+e^{(\Delta_2/E')/T_i})$$

Δ_1=1/R'-1/R

Δ_2=E'-E

Algorithm

Generate a layout at random. Calculate text entry rate and EP of the layout.
Initialize temperature. Set minimum temperature, maximum iterations and α
Initialize iteration count.

repeat

repeat

Choose two keys randomly from current layout l and swap characters to generate l'

if $\Delta_1, \Delta_2 \leq 0$ then

set l = l'

else if $\Delta_1, \Delta_2 > 0$ then

calculate P_1, if $P_1 \geq$ rand(), set l = l'

else if $\Delta_1 > 0$ and $\Delta_2 \leq 0$ then

calculate P_2, if $P_2 \geq$ rand(), set l = l'

else

calculate P_3, if $P_3 \geq$ rand(), set l = l'

end if

update iteration count by one

until maximum number of iterations

update T; $T = T \times \alpha$

until T > TMIN

Select the current layout

User Study

- 27 keys keyboards (26 letters + space)
- 3-level and 2-level scanning
- Developed optimized layouts
- Compared with alphabetic and randomly perturbed layouts
- Eight subjects (six with disabilities, two without)

Optimum Grouping (3-level)

Block 1	K1	K2	K5	K11
	K3	K6	K12	K19
	K7	K13	K20	K27
Block 2	K4	K8	K14	K21
	K9	K15	K22	
	K16	K23		
Block 3	K10	K17	K24	
	K18	K25		
	K26			

Optimum Grouping (2-level)

K1	K2	K4	K7	K11	K16	K22
K3	K5	K8	K12	K17	K23	
K6	K9	K13	K18	K24		
K10	K14	K19	K25			
K15	K20	K26				
K21	K27					

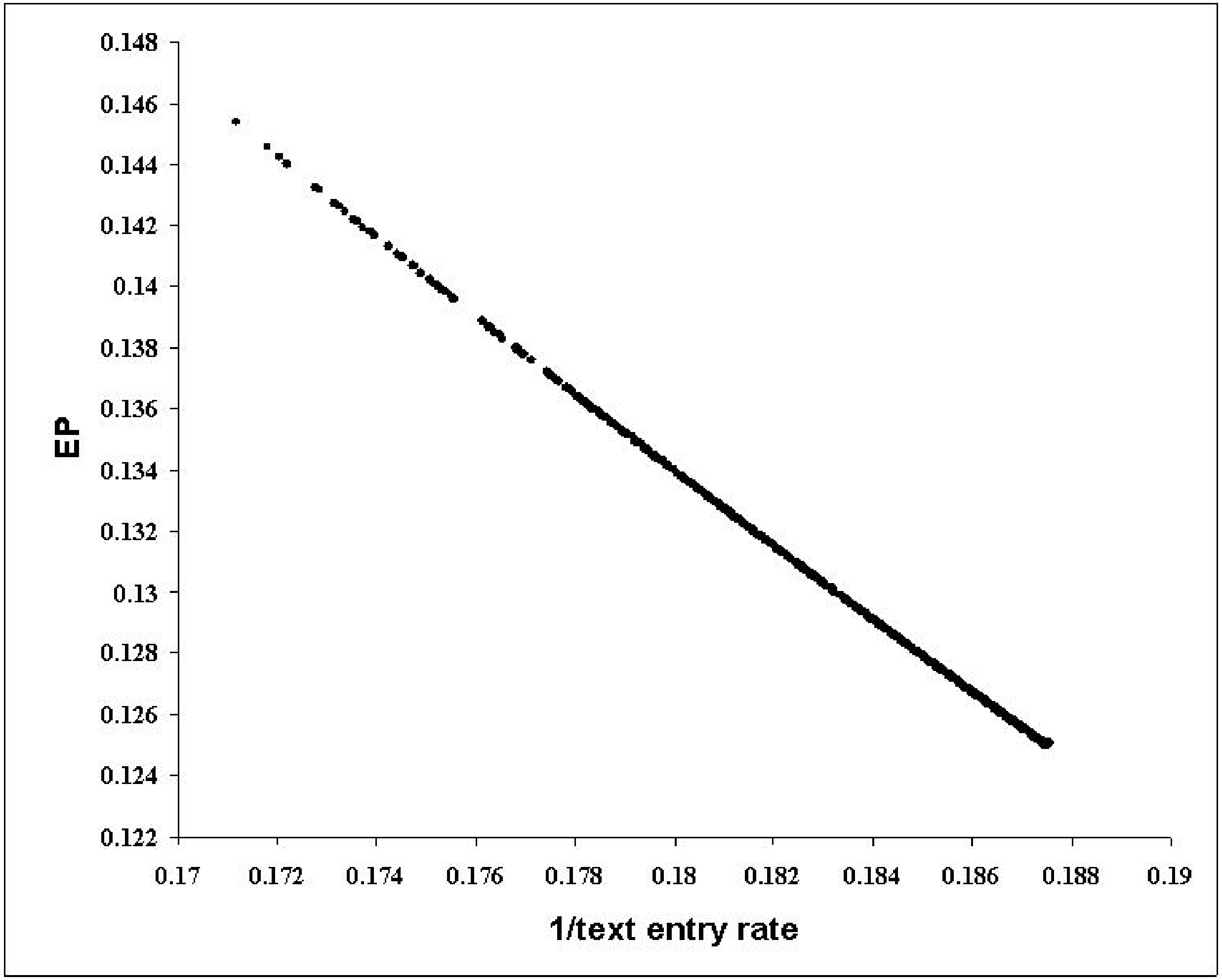
Optimum layouts

- $T = 1$ sec, $T_{MIN} = 0.01$
- $P_0 = 0.8$
- $\alpha = 0.99$
- Number of iteration for each temp = 1000
- ΔE = average text entry rate diff for twenty random layouts

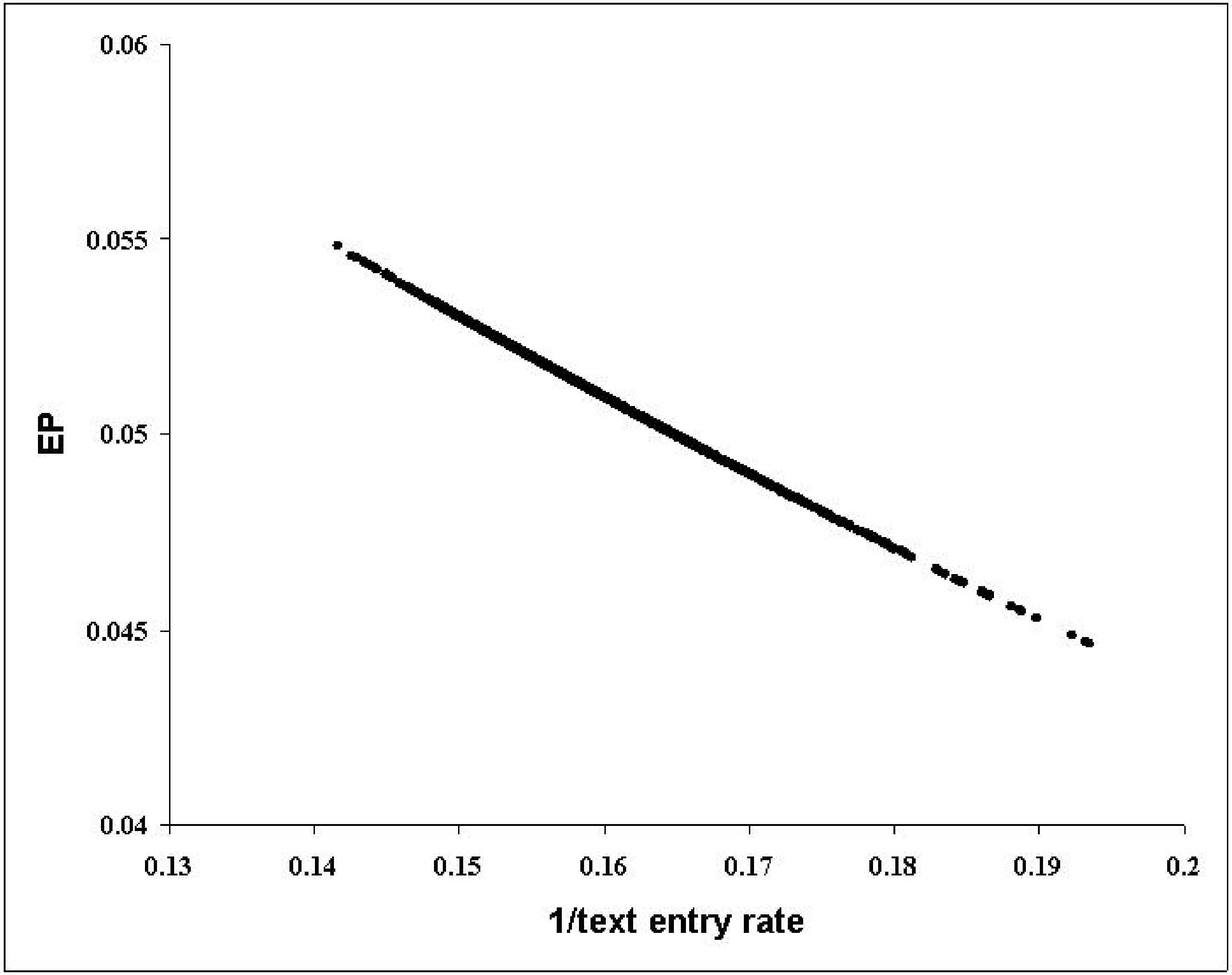
Some Statistics

- Each run considered 90,90,000 layouts
- Output of each run—a near optimum solution
- 1000 solution points generated for each
- These formed the Pareto fronts

Pareto Front for 3-level

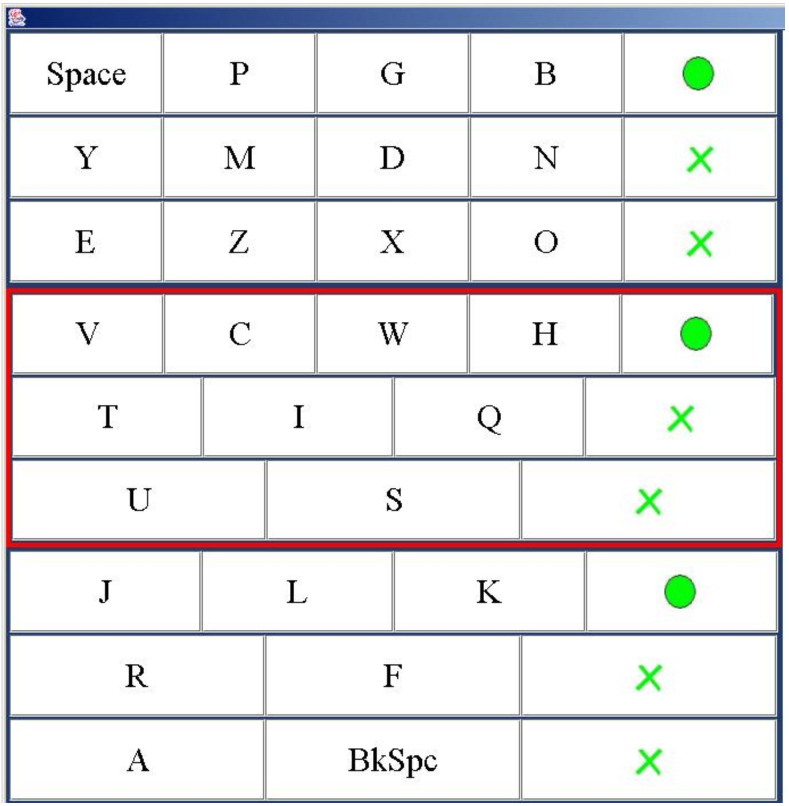


Pareto Front for 2-level



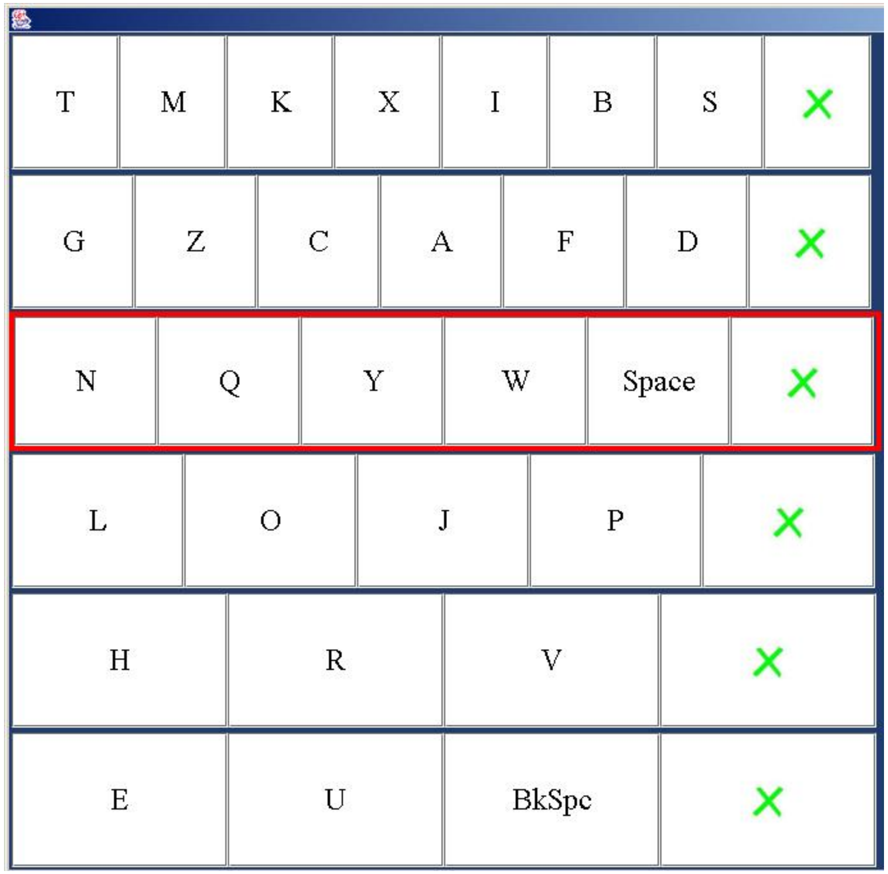
Final Designs

- Design with least EP for 3-level

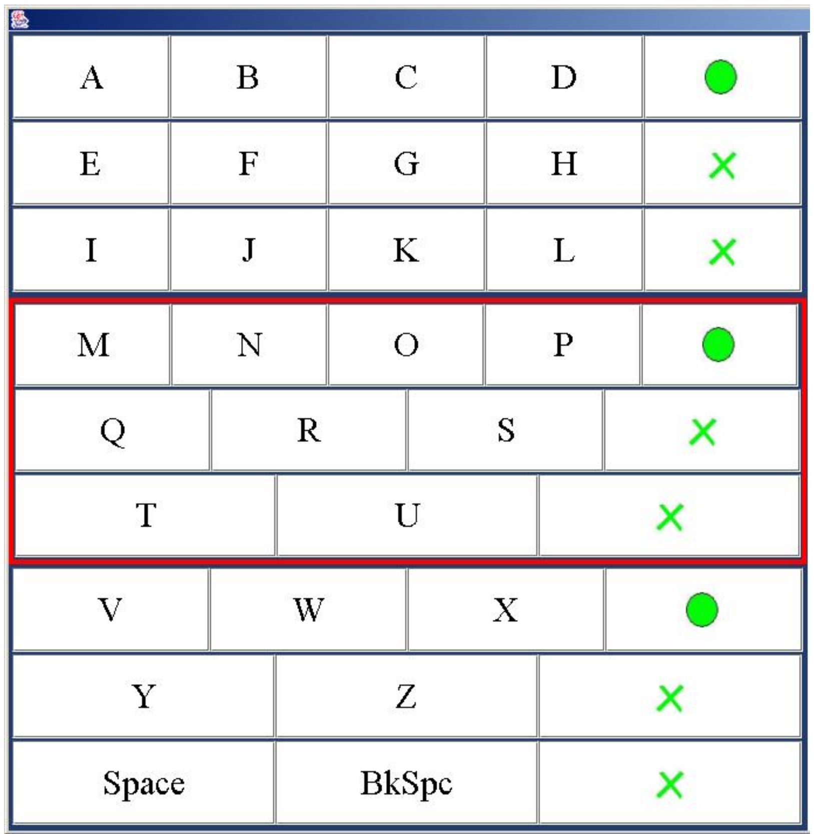


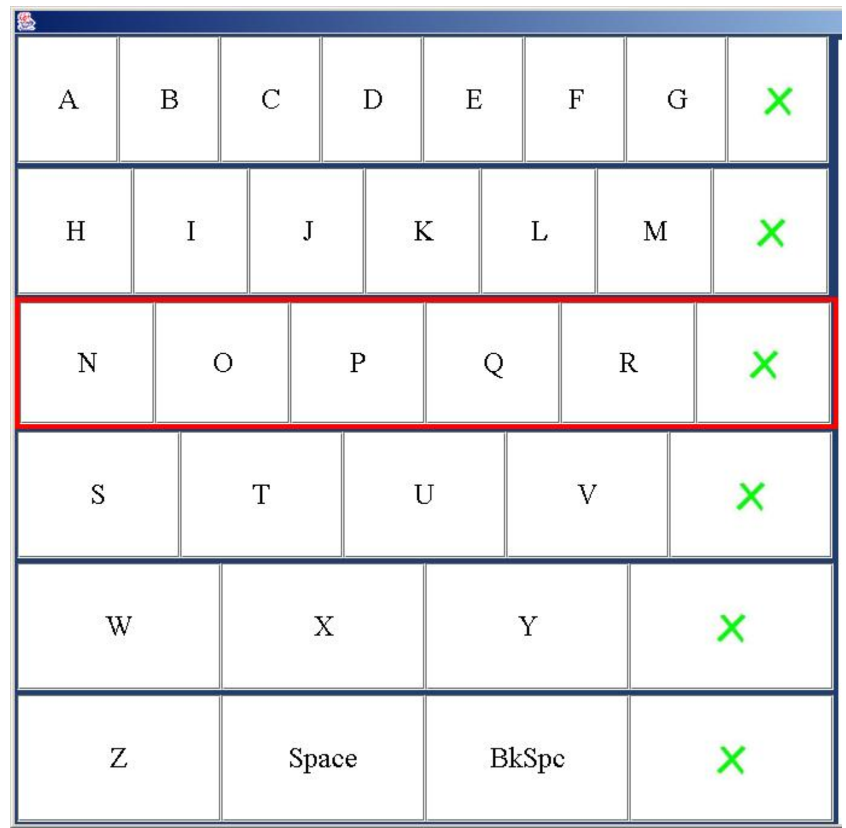
Final Designs

- Design with highest text entry rate for 2-level

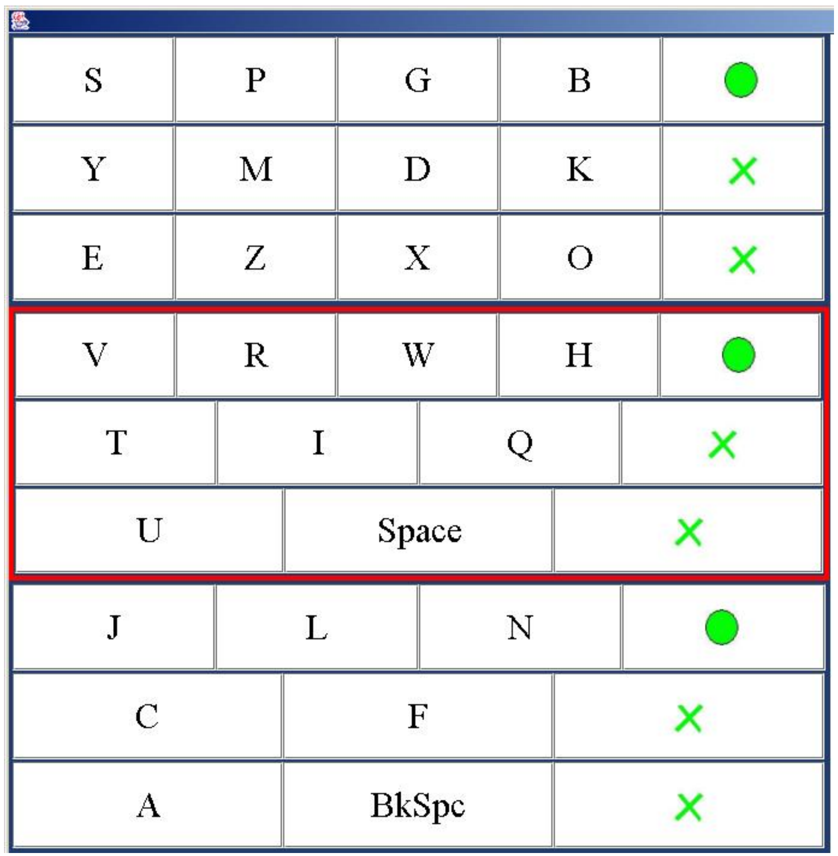


Other Layouts: Alphabetic





Other Layouts: Perturbed



Predicted Performance

Interface	Text entry rate	EP
OPT_3SK	**5.33**	**0.125**
ALPH_3SK	**5.11**	**0.135**
PERT_3SK	**5.27**	**0.127**
OPT_2SK	**7.06**	**0.054**
ALPH_2SK	**6.56**	**0.052**
PERT_2SK	**5.81**	**0.049**

Expected

- OPT_3SK should have less probability of error (i.e. lower EP) than the other two
- OPT_2SK should have higher text entry rates than the other two

Observations

- For 3-level, all the subjects had higher text entry rate and lower EP with OPT_3SK than the other two
- For 2-level, all the subjects had higher text entry rate with OPT_2SK. However, in a few cases, subjects had more errors with the optimum design

Further Work

- More data for refinement and further validation
- Visual search incorporation
- Other multi-objective algorithms for design space search
- Predictive and ambiguous keyboards
- Extension to other scanning aids
- Scan step determination
- Automatic usability evaluation framework

Thank You