Information theory as a unified tool for understanding and designing human-computer interaction

Wanyu Liu

To cite this version:

Wanyu Liu. Information theory as a unified tool for understanding and designing human-computer interaction. Human-Computer Interaction [cs.HC]. Université Paris Saclay (COmUE), 2018. English. NNT : 2018SACLT012 . tel-02004468

HAL Id: tel-02004468
https://pastel.hal.science/tel-02004468

Submitted on 1 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université
PARIS-SACLAY

NNT : 2018SACLT012

Thèse de doctorat

Information Theory as a Unified Tool for Understanding and Designing Human-Computer Interaction

PhD thesis of l'Université Paris-Saclay
Prepared at Télécom ParisTech

Doctoral school n°580 Sciences et technologies de l'information et de la communication (STIC)
Speciality : Données, Connaissances, Apprentissage et Interaction

Thesis defense in Paris, on 22 November 2018, by

Wanyu Liu

Committee members :

Joseph Jean Boutros	Professor, Texas A&M University	President
Roderick Murray-Smith	Professor, University of Glasgow	Reviewer
Géry Casiez	Professor, Université de Lille	Reviewer
Otmar Hilliges	Associate Professor, ETH Zürich	Examiner
Olivier Rioul	Professor, Télécom ParisTech	Thesis advisor
Michel Beaudouin-Lafon	Professor, Université Paris-Sud	Thesis co-advisor
Yves Guiard	DR Emeritus, CNRS Professor Emeritus, Télécom ParisTech	Thesis co-advisor

Wanyu Liu: *Information Theory as a Unified Tool*
for Understanding and Designing
Human Computer Interaction, © December 2018

DEDICATION

This thesis is dedicated to my family, mentors and friends across three continents and multiple countries, without whom, I would have not been able to arrive at where I stand today. I would like to acknowledge my appreciation to these amazing people.

I am deeply indebted to Dr. Bernd Ploderer who introduced me to Human-Computer Interaction (HCI). It was a bug from the course application platform, which resulted in me taking a not-supposed-to selective course – Usability Design and Interaction in the first semester of my master study, and met Bernd who has been a tremendous mentor for me. I still remember all these afternoons that we spent in his office in Doug McDonell building where Bernd walked me through on how to conduct research and write papers step by step. I appreciate all his time and patience during my master study and his strong encouragement for pursuing a Ph.D.

I am extremely grateful for my Ph.D. advisors: Professor Olivier Rioul, Professor Michel Beaudouin-Lafon, Dr. Yves Guiard, who have taught me, both consciously and unconsciously, how good research is done. Working with three advisors from three distinctive domains (Information theory, HCI, experimental psychology) has not been easy, but I have learned a great deal from these three excellent examples. All of them have been immensely helpful and motivational and made my Ph.D. experience stimulating and productive. I would like to thank them wholeheartedly, not only for their continuous support, but also for giving me so many wonderful opportunities to go to conferences and meet people. A special thanks goes to Dr. Dr. Wendy Mackay, whose enthusiasm for science has greatly influenced me for the past three years and will stay for life time.

My sincere gratitude also goes to all of those with whom I have had the pleasure to work during my Ph.D. Dr. Rafael Gregorio Lucas D'Oliveira, with whom I came up with the "BIG" idea, has been an amazing collaborator and a true friend. It took us lots of fights and beers to speak the same research language, but the outcome has become a cornerstone of my thesis. My co-authors Dr. Gilles Bailly, Professor Andrew Howes, and Professor Joanna McGrenere have provided me extensive personal and professional guidance throughout the thesis. A special mention goes to Professor Antti Oulasvirta for the valuable experience of a 2-week visit at Aalto University where we have had numerous inspirational discussions, as well as for his priceless and constant encouragement for better self development.

I would have never thought of teaching a course at the summer school without him. My Ph.D. was also enriched by the extraordinary interaction that I have had with the computational interaction community: attending Dagstuhl seminar, visiting Cambridge with Professor Per Ola Kristensson and ETH Zurich with Professor Otmar Hilliges.

For this dissertation, I would like to thank my thesis reviewers: Professor Roderick Murray-Smith and Professor Géry Casiez for their time, interest and helpful comments. I would also like to thank the two thesis examiners: Professor Joseph Jean Boutros, and Professor Otmar Hilliges for their time and insightful questions. Most of all, they have made this defense an enjoyable moment.

My time in Paris was made delightful in large part due to the many friends and groups that became a part of my life. Manos and Bruno have been like family to me. We went together to the summer school in Helsinki in 2016, CHI 2017 in Denver, after which, with Julien, we have had a lovely one-week road trip from central America to the west coast. Their presence reassures me that whatever happens, I am not alone. The three teams – Comelec, VIA, ExSitu that I was affiliated with have also been a source of friendships as well as good advice. I appreciate all the lunch discussions, sometimes intense debates, and coffee breaks with Julien, Alaa, Artur, Vincent and many other Comelec fellows; the time we spent at butte aux cailles every Thursday evening sharing visions about research and life with Marc, Michael, Arnaud, James, Sally, and sometimes Stacy, Jiali, Germán, Carla, Philip, Nacho, Chengcheng, Jean-Philippe, Janin, Mathieu and many others if I forgot to mention their names.

I would also like to thank all that I have met at the summer school and conferences: Abdo, Beren, Artem, Jan, Damien, Kashyap, and my friends outside of research: Laura, Xi and Miao, with whom I have shared moments of anxiety (rarely) but also of big excitement. The dancing community has also given me constant joy: Fabian, Nadya, Thomas, Anthi, Lucia, Alexis. Thank you guys very much, especially Fabian, who has taught me everything about ballroom dancing. A very special word of thanks goes for Alex, my dancing partner and diving buddy, with whom I spin and twirl, and discover the extraordinary underwater world.

This journey would not have been possible without the support of my family, who gives me unconditional love and encouragement. To my parents, thank you for raising me with a love of science, for supporting me in all my pursuits and for teaching me to always learn, laugh and love.

ÉCOLE DOCTORALE
Sciences et technologies
de l'information
et de la communication (STIC)

Titre : La théorie de l'information comme un outil unifié pour comprendre et concevoir l'interaction homme-machine

Résumé : La théorie de l'information a influencé un grand nombre de domaines scientifiques depuis son introduction par Claude Shannon en 1948. A part la loi de Fitts et la loi de Hick, qui sont apparues lorsque les psychologues expérimentaux étaient encore enthousiastes à l'idée d'appliquer la théorie de l'information aux différents domaines de la psychologie, les liens entre la théorie de l'information et l'interaction homme-machine (IHM) ont été peu explorés.

L'objectif de cette thèse est de combler le fossé entre la théorie de l'information et l'IHM en considérant que l'interaction entre les humains et les machines peut être considérée comme un processus de communication et peut donc être caractérisée à l'aide des concepts de la théorie de l'information. Les trois principales contributions de cette thèse sont : (1) une perspective historique détaillée sur la manière dont la théorie de l'information a influencé la psychologie et l'IHM, avec en particulier une discussion approfondie et une analyse de la pertinence de la loi de Hick pour l'IHM, (2) le cadre formel Gain d'Information Bayésienne (BIG pour Bayesian Information Gain) qui quantifie et exploite les informations envoyées par l'utilisateur à l'ordinateur pour exprimer son intention et (3) une illustration des avantages de l'utilisation des mesures de la théorie de l'information pour évaluer la performance des entrées et pour caractériser une tâche d'interaction. Cette thèse démontre ainsi que la théorie de l'information peut être utilisée comme un outil unifié pour comprendre et concevoir la communication et l'interaction homme-machine.

Mots clés : L'interaction homme-machine, la théorie de l'information, la psychologie expérimentale

Title : Information theory as a unified tool for understanding and designing human-computer interaction

Abstract : Information theory has influenced a large number of scientific fields since its first introduction in 1948. Apart from Fitts' law and Hick's law, which came out when experimental psychologists were still enthusiastic about applying information theory to various areas of psychology, the relation between information theory and human-computer interaction (HCI) has rarely been explored.

This thesis strives to bridge the gap between information theory and HCI by taking the stance that human-computer interaction can be considered as a communication process and therefore can be characterized using information-theoretic concepts. The three main contributions are: (1) a detailed historical perspective on how information theory influenced psychology and HCI, particularly an in-depth discussion and analysis of how relevant Hick's law is for HCI; (2) a Bayesian Information Gain (BIG) framework that quantifies the information sent by the user to the computer to express her intention; and (3) a further illustration of the advantages of using information-theoretic measures to evaluate input performance and to characterize the rich aspects of an interaction task. This thesis demonstrates that information theory can be used as a unified tool to understand and design human-computer communication & interaction.

Keywords : Human-computer interaction, information theory, experimental psychology

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

1. Wanyu Liu, Rafael Lucas D'Oliveira, Michel Beaudouin-Lafon, and Olivier Rioul. 2017. BIGnav: Bayesian Information Gain for Guiding Multiscale Navigation. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). ACM, New York, NY, USA, 5869-5880.
2. Wanyu Liu, Olivier Rioul, Joanna McGrenere, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2018. BIGFile: Bayesian Information Gain for Fast File Retrieval. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 385, 13 pages.
3. Wanyu Liu, Olivier Rioul, Michel Beaudouin-Lafon, and Yves Guiard. 2017. Information-Theoretic Analysis of Human Performance for Command Selection. In IFIP Conference on Human-Computer Interaction, pp. 515-524. Springer, Cham, 2017.

CONTENTS

OVERVIEW 1

Research Context 1

Outlining Contributions 2

I INFORMATION THEORY IN PSYCHOLOGY AND IN HCI 5

1 INFORMATION THEORY CONCEPTS 7

1.1 Information as Entropy 7

1.2 Mutual Information and Equivocation 8

1.3 Channel Capacity and Throughput 9

2 INFORMATION THEORY IN PSYCHOLOGY 13

2.1 Enthusiasm at the Early Stage 13

2.1.1 Measuring Information 13

2.1.2 Investigating Information Capacity 17

2.2 From Criticism to Abandonment 21

3 PSYCHOLOGY AND HCI – FITTS' LAW 25

3.1 Which Minimum Amount of Information? 26

3.2 The Average of Minimum or the Average of Average? 28

3.3 Does Throughput Equal Channel Capacity? 29

4 PSYCHOLOGY AND HCI – HICK'S LAW 33

4.1 HCI Applications of Hick's Law 34

4.1.1 Modeling Menu Performance 34

4.1.2 Modeling Text Entry 36

4.1.3 Interface Design Guideline 37

4.1.4 Summary: Inconsistencies 37

4.2 Choice-Reaction Time in Psychology 38

4.2.1 Before Information Theory 38

4.2.2 The Information Analogy: Hick & Hyman 39

4.2.3 Choice Reaction Time: Results 40

4.2.4 Clarifying Hick's law 43

4.3 The Choice-Reaction Paradigm and HCI 44

4.3.1 Decomposition of Time Measures 44

4.3.2 The Differences between Novices and Experts 45

4.3.3 Effect Size of Hick's Law 47

4.4 Implications For HCI 49

4.4.1 The Concavity of the Logarithm Contradicts the Hick-based Design Principle 49

4.4.2 xT Design Principle: A Matter of Convexity 51

4.5 Conclusion 52

5 OTHER INFORMATION THEORY ATTEMPTS IN HCI 55

5.1 Statistical Language Processing for Text Entry 55

5.2 Human Information Capacity 58

5.3 Pointing without a Pointer 60

6 SUMMARY 63

II A BAYESIAN INFORMATION GAIN (BIG) FRAMEWORK FOR QUANTIFYING INFORMATION 65

7 BAYESIAN INFORMATION GAIN FRAMEWORK 67
- 7.1 Motivation . . . 67
- 7.2 BIG Framework . . . 68

8 MULTISCALE NAVIGATION 73
- 8.1 Better Understanding Multiscale Navigation . . . 74
- 8.2 Exploiting Information Space Features . . . 75
- 8.3 Interpreting User Intention . . . 76

9 MULTISCALE NAVIGATION: BIGNAV 79
- 9.1 BIGnav . . . 79
- 9.2 Implementation: BIGnav in 1D . . . 81
- 9.3 Implementation: BIGnav in 2D . . . 84
- 9.4 Controlled Experiment . . . 86
  - 9.4.1 Participants . . . 86
  - 9.4.2 Apparatus . . . 87
  - 9.4.3 Procedure . . . 87
  - 9.4.4 Task . . . 88
  - 9.4.5 Data Collection . . . 89
- 9.5 Results . . . 89
  - 9.5.1 Task Completion Time . . . 89
  - 9.5.2 Number of Commands . . . 91
  - 9.5.3 Uncertainty and Information Gain . . . 92
  - 9.5.4 Trajectory in Multiscale Worlds . . . 93
  - 9.5.5 Qualitative Results . . . 94
- 9.6 Application: BIGMap . . . 95
- 9.7 Discussion . . . 96

10 HIERARCHICAL FILE RETRIEVAL 99
- 10.1 Personal File Systems . . . 100
  - 10.1.1 File Management . . . 100
  - 10.1.2 File Retrieval . . . 101
- 10.2 Enhanced File Retrieval Techniques . . . 104
  - 10.2.1 Visualization and Search . . . 104
  - 10.2.2 Prediction . . . 105
- 10.3 Adaptive User Interfaces . . . 108

11 HIERARCHICAL FILE RETRIEVAL: BIGFILE 113
- 11.1 BIGFile Interface . . . 113
- 11.2 BIGFile Algorithm . . . 115
  - 11.2.1 BIGFileOpt: Optimal Algorithm . . . 115
  - 11.2.2 BIGFileFast: Efficient Algorithm . . . 117
- 11.3 Pilot Study . . . 118
- 11.4 Study 1: Simulations . . . 120
  - 11.4.1 Simulation Settings . . . 121
  - 11.4.2 Simulation Results . . . 121

11.5 Study 2: Experiment 123
11.5.1 Method 124
11.5.2 Participants 124
11.5.3 Apparatus 124
11.5.4 Procedure 125
11.5.5 Data Collection 126
11.6 Results 127
11.6.1 Task Completion Time and Step Time 127
11.6.2 Number of Steps and Information Gain 128
11.6.3 Subjective Feedback 129
11.7 Discussion 130
11.7.1 Split File Interface 130
11.7.2 Comparisons with AccessRank 131
11.7.3 Limitations 131
12 DICUSSION AND FUTURE PERSPECTIVES 133
12.1 BIGnav vs. BIGFile 133
12.2 Collaboration and Adaptation 134
12.3 Future Perspectives 136

III INFORMATION-THEORETIC MEASURES FOR CHARACTERIZING INTERACTION 139
13 MOTIVATION 141
14 INFORMATION-THEORETIC MEASURES 145
14.1 Input Entropy $H(X)$ 146
14.2 Transmitted Information $I(X;Y)$ 146
14.3 Transmitted Information Rate TP 147
14.4 Equivocation $H(X|Y)$ 147
14.5 Advantages of the Information-theoretic Measures . . 148
15 COMMAND SELECTION 151
15.1 Simulations 151
15.1.1 Simulation Settings 152
15.1.2 Simulation Results 153
15.1.3 Summary 156
15.2 Data Reanalysis 157
15.2.1 Changing the Number of Items and Menu Organization 157
15.2.2 Changing Distribution 159
15.2.3 Summary 160
15.3 Controlled Study 161
15.3.1 Data Collection 161
15.3.2 Results 162
15.3.3 Information-Theoretic Analysis 164
15.3.4 Summary 166
16 TEXT ENTRY 167
16.1 Conditional Input Entropy 167
16.2 Intelligent Statistical Decoder 170

16.3 Keyboard Simulations 172
16.3.1 No Errors 172
16.3.2 Uncorrected Errors 174
16.3.3 Auto-correction 175
16.3.4 Summary 177
16.4 Pilot Study 177
16.4.1 Results 178
16.5 Summary 179
17 DICUSSION AND FUTURE PERSPECTIVES 181
17.1 Command Selection & Text Entry 181
17.2 Future Work 182
CONCLUSION 185
Looking Forward 189

IV APPENDIX 191
A APPENDIX 193
A.1 BIGnav Experiment Consent 193
A.2 BIGnav Post-Experiment Questionnaire 196
A.3 BIGFile Experiment Consent 198
A.4 BIGFile Post-Experiment Questionnaire 200

BIBLIOGRAPHY 209

LIST OF FIGURES

Figure 1 An overview of this thesis. 4
Figure 2 Shannon's communication scheme. 7
Figure 3 Noiseless binary and noisy channel. 10
Figure 4 Schematic diagram of the choice-reaction time as a model of communication. 15
Figure 5 Hick's and Hyman's data. 16
Figure 6 Channel capacity in absolute judgement tasks. 18
Figure 7 Schematic diagram of human motor system as a model of communication. 19
Figure 8 Results of Fitts' 4 experiments. 20
Figure 9 Card, Moran and Newell's description of Fitts' law. 25
Figure 10 Comparison between $\mathrm{ID}(\varepsilon)$ and ID_e. 27
Figure 11 Movement data as a function of ID in a wild study. 29
Figure 12 Movement time, accuracy rate and throughput accross 3 conditions. 30
Figure 13 Guiard and Rioul's WHo model describing speed-accuracy tradeoff of aimed movement. 31
Figure 14 Card, Moran and Newell's description of Uncertainty Principle. 33
Figure 15 Reaction time as a function of stimulus "information" in different tasks. 42
Figure 16 Time measures in controlled experiments. . . . 45
Figure 17 Reanalysis of data from Roy et al. [181] and Liu et al. [134]: reaction time as a function of stimulus uncertainty. 48
Figure 18 xT design guideline based on data categorization and xT function. 52
Figure 19 Writing in Dasher. 56
Figure 20 Token passing strategy. 57
Figure 21 Information capacity of repeated movements. . 59
Figure 22 Model of a user controlling sound using continuous analog sensors. 60
Figure 23 Pointing without a pointer. 61
Figure 24 The Bayesian Information Gain (BIG) framework. 68
Figure 25 Multiscale representations and interactions. . . 73
Figure 26 Space-scale diagrams. 74
Figure 27 Fitts' law in multiscale navigation. 74
Figure 28 GravNav and Content-aware scrolling. 75
Figure 29 "The River Analogy" to assist 3D navigation. . 76

Figure 30 SDAZ and Orthozoom. 77
Figure 31 Controlled experiment comparing Orthozoom with SDAZ. 77
Figure 32 Illustration of BIGnav. 79
Figure 33 Navigation with BIGnav in 4 steps. 81
Figure 34 Discretization of user input in BIGnav. 84
Figure 35 BIGnav experimental condition. 86
Figure 36 Task completion time by ID in BIGnav experiment. 90
Figure 37 Task completion time by distribution in BIGnav experiment. 91
Figure 38 Number of commands by technique in BIGnav experiment. 92
Figure 39 Uncertainty decrease and information gain in BIGnav experiment. 93
Figure 40 Time plot of the decrease in ID in BIGnav experiment. 94
Figure 41 BIGMap. 95
Figure 42 Three common views to represent files and structures. 99
Figure 43 File retrieval practices. 102
Figure 44 Search-based systems MyLifeBits and Stuff I've Seen. 104
Figure 45 Prediction list stability v.s. Average Rank and Percentage Revisitations predicted over three datasets. 107
Figure 46 Three augmented file interfaces: Icon Highlights, Hover Menus and Search Directed Navigation. 107
Figure 47 Experiment results of the three augmented file interfaces. 108
Figure 48 Ephemeral adaptation applied to menus. . . . 109
Figure 49 *SUPPLE* generates personalized interfaces for users with motor impairments. 109
Figure 50 Adapted interfaces for users with different ethnic backgrounds. 110
Figure 51 Static split menu and Adaptable split menu. . 110
Figure 52 Split, moving and visual popout interface. . . . 111
Figure 53 The BIGFile interface. 113
Figure 54 Algorithm BIGFileFast with a binary tree. . . . 117
Figure 55 Visualization of P12's Documents folder, a 5-level tree. 119
Figure 56 Simulations of BIGFileFast v.s. AccessRank. . . 122
Figure 57 BIGFile experimental condition. 125
Figure 58 Task completion time and number of steps in BIGFile experiment. 128

Figure 59 Uncertainty and information gain in BIGFile experiment. 129
Figure 60 NASA TLX scores and overall preference in BIGFile experiment. 130
Figure 61 Mixed-initiative system Lookout. 135
Figure 62 Technique *Glass* and *Glass+Skin* with finger identification. 142
Figure 63 Transmitted information and throughput in the experiment of *Glass* and *Glass+Skin*. 142
Figure 64 The decomposition of Shannon's entropy. . . . 143
Figure 65 Information-theoretic measures for examining human-computer interaction as information transmission from the user to the computer through the channel interface. 145
Figure 66 Simulation settings with three variables. 152
Figure 67 Simulation results when error rate increases. . 153
Figure 68 Simulation results when item height and starting distance are changed. 154
Figure 69 Simulation results when the number of items increases. 155
Figure 70 Simulation results when distribution is changed. 156
Figure 71 Information-theoretic notion of throughput and selection time for 3 organizations and menu lengths. 157
Figure 72 Three definitions of throughput in alphabetic, semantic and unordered menus. 158
Figure 73 3 distributions with names, multiset of item frequencies, the r-squared and pairwise distance. 159
Figure 74 Information-theoretic notion of throughput and selection time for 3 distributions. 159
Figure 75 Three definitions of throughput in 3 distributions. 160
Figure 76 Experimental condition for the command selection task. 162
Figure 77 Experimental results in the command selection task. 163
Figure 78 A toy keyboard with 7 keys. 168
Figure 79 Probability of individual character and Conditional probability of pairs of characters in language model L. 169
Figure 80 A substitution-only decoder. 171
Figure 81 A decoder for 'a' and 'b'. 171
Figure 82 Three keyboards with keys in different sizes: small, regular and large. 173
Figure 83 Comparison among three measures of throughput when key size and error type changes. . . 174

Figure 84 Auto-correction in text entry. 176
Figure 85 Comparison among three measures of throughput and KSCP with or without auto-correction. 176
Figure 86 Comparison among three measures of throughput and KSCP in the pilot study. 178

LIST OF TABLES

Table 1 A comparison of Hick 1952, Hyman 1953 and HCI studies that used Hick's law. 46
Table 2 Full-factorial ANOVA for 3 conditions on their respective reaction times. 48
Table 3 Notations in Bayesian Experimental Design (BED) and in Bayesian Information Gain (BIG). 70
Table 4 Calibration results used as prior knowledge about user behavior. 85
Table 5 Full-factorial ANOVA on TCT. 90
Table 6 Full-factorial ANOVA on the number of commands. 91
Table 7 Significant effects in the full-factorial ANOVA on TCT. 127
Table 8 Full-factorial ANOVA on the number of steps required to locate the target. 129
Table 9 A summary of how the three definitions of throughput change when different conditions change. 156
Table 10 A summary of different measures on the simulations of 3 keyboards. 173
Table 11 A summary of different measures on the simulations of 2 error types. 174
Table 12 A summary of different measures on the simulations with or without auto-correction. . . . 175
Table 13 A summary of different measures of the pilot study. 178

ACRONYMS

HCI	Human Computer Interaction
ID	Index of Difficulty
KSPC	Keystroke Per Character
WMP	Words Per Minute
BIG	Bayesian Information Gain
BED	Bayesian Experimental Design
MI	Mutual Information
RT	Reaction Time
MT	Movement Time
TP	Throughput
ERP	Event Related Potential
EEG	Electroencephalography

OVERVIEW

RESEARCH CONTEXT

Information theory, since it was first introduced by Claude Shannon back in 1948, has received much attention and successful applications in a number of domains, notably in electrical engineering. This mathematically described communication scheme outlines the information transmission process from a sender to a receiver over a noisy channel. Apart from the two remaining principles: Fitts' law and Hick's law (or the Hick-Hyman law), which came out when experimental psychologists were still enthusiastic about applying information theory to various areas of psychology, the relation of information theory to human-computer interaction (HCI) has not been clear. Even the two above-mentioned "laws" remain controversial in both psychology and HCI.

As users, we implicitly and explicitly send information to the computer to accomplish tasks and to express our intentions. Interestingly, this communication standpoint is supported by the ACM SIGCHI Curriculum for human-computer interaction [1], which points out that "Because human-computer interaction studies a human and a machine in communication, it draws from supporting knowledge on both the machine and the human side". In recent years, we have also started seeing information theory inspire or contribute to HCI research.

This thesis strives to bridge the gap between information theory and human-computer interaction. I argue that information theory can be used as a unified tool for understanding the human-computer communication process as well as for designing interactions with more efficient communication rates. Towards this goal, I propose a Bayesian Information Gain (BIG) framework to quantify the information sent by the user to the computer and I present two interaction techniques that use BIG to improve communication efficiency. I then illustrate the advantages of using information-theoretic measures to evaluate input performance and to characterize the rich aspects of an interaction task. These two contributions are not possible without a historical walkthrough of how information theory influenced psychology and HCI. I conclude with a plea for using information theory as a unified tool to understand and design human-computer communication & interaction.

OUTLINING CONTRIBUTIONS

This thesis is organized in 3 parts:

Part i provides a detailed historical perspective on how information theory influenced psychology and HCI. It starts with basic concepts of information theory that are used throughout the thesis and highlight how they are different from our ordinary understanding, particularly of the notion of "information". Then it goes through the history of how experimental psychologists were first excited by the ability to quantitatively measure "information", and then abandoned information theory completely under the criticism of the information theory community. Today, two principles born during this 1950's period are still used in HCI: Fitts' law and Hick's law. While Fitts' law has welcome a large number of applications, Hick's law remains rather controversial. This part continues with an in-depth discussion and analysis of how relevant Hick's law is for HCI. I argue that only by understanding the essence of information-theoretic concepts and by examining the ups and downs from a historical perspective, we can grasp the theory, clarify the misunderstandings and take advantage of it in the domain of HCI.

Part ii presents the Bayesian Information Gain (BIG) framework that is built on the scheme of human-computer communication: users send information to computers to express their intentions and interests. BIG is based on Bayesian Experimental Design using the criterion of mutual information from information theory and quantifies the information in the user input to reduce the computer's uncertainty in bits. By actively probing users for information at each interaction step, the computer can play a more active role and improve the interaction & communication efficiency.

The part first introduces the framework and demonstrates it with a 1D scenario where the computer tries to gain maximum information from the user. Then it goes in depth with two use cases in multiscale navigation and in hierarchical file retrieval respectively. We report two controlled experiments: a controlled experiment with 16 participants in multicale navigation comparing the BIG technique BIGnav with conventional pan and zoom; and a controlled experiment with 18 participants in hierarchical file retrieval comparing the BIG interface BIGFile with two other interfaces. Both experiments favor the BIG-inspired interaction technique and interface. Lastly we outline the possibilities for future work.

Part iii builds on and extends the concept that users send information to the computer through the input device or the interface, which constitute the communication channel. The information-theoretic measures quantify how much information can be transmitted (entropy), how much information is successfully transmitted (mutual information) and what is the information transmission rate (throughput). Compared to the conventional objective assessment of input techniques and interfaces, they offer a richer and more coherent description of an interaction task.

The part starts by going through some similar ideas in the HCI literature and introduces the information-theoretic measures. It then demonstrates how to use these measures in the context of command selection and text entry, comparing the information-theoretic notion of throughput with two existing definitions of throughput and outlines the coherence as well as consistency of the information-theoretic measures. Finally, I emphasize the benefits of using this general framework and discuss its potential use in other contexts as well as its limitations.

In summary, the main contributions of this thesis are:

- A historical walkthrough of information theory applications in psychology and in HCI with an extensive discussion of how relevant Hick's law is for HCI (Part i);
- A Bayesian Information Gain (BIG) framework to quantify information sent by the user to the computer (Part ii);
- An information-theoretic notion of throughput for characterizing information transmission efficiency (Part iii).

Overall, the aim of this thesis is to formally examine the human-computer communication process using the tools of information theory. The notations X and Y used in the chapters do not always correspond to the same meaning (Fig. 1), but in both cases, I only consider information transmitted from the user to the computer. I do not consider information transmitted from the computer to the user. I illustrate two use cases in Part ii and two use cases in Part iii and discuss future perspectives in the conclusion.

The content of this thesis is based on work already published or under review, and more specific details will be given at the beginning of each chapter. I do not provide a separate chapter for the related work, but prefer to refer to it as research context that motivates the work at the beginning of each treated subject.

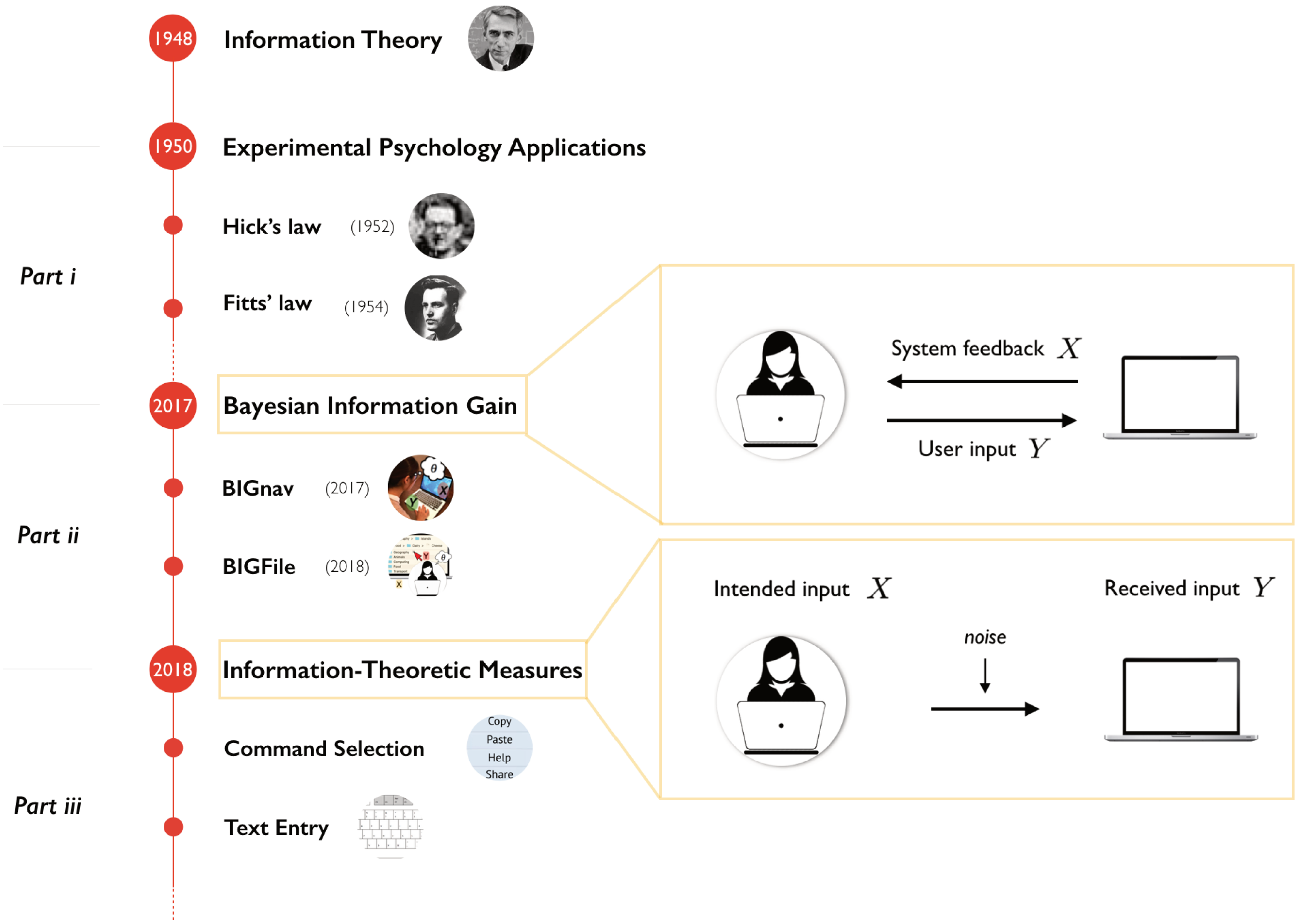

1948
Information Theory
1950
Experimental Psychology Applications
Hick's law (1952)
Fitts' law (1954)
Part i
2017
Bayesian Information Gain
BIGnav (2017)
BIGFile (2018)
Part ii
2018
Information-Theoretic Measures
Command Selection
Copy
Paste
Help
Share
Text Entry
Part iii
System feedback X
User input Y
Intended input X
Received input Y
noise

Figure 1: An overview of this thesis.

Part I

INFORMATION THEORY IN PSYCHOLOGY AND IN HCI

The goal of this part is to provide a historical perspective on how information theory influenced psychology and HCI. It starts with basic concepts of information theory that are used throughout this thesis. Then it outlines a number of information theory applications in psychology and in HCI and provides an in-depth discussion and analysis of how relevant Hick's law is for HCI.

I argue that by understanding the basic concepts of information theory and walking through history, we as a community can better understand why information theory has not been successfully applied in experimental psychology, clarify the misunderstanding that we hold so far and further take advantage of the theory in the domain of HCI.

1

INFORMATION THEORY CONCEPTS

This chapter provides a few key concepts of information theory that are used throughout the thesis. Readers familiar with these concepts can move onto the next chapter directly. We recommend *Elements of information theory* [34] for those who want to know more about information theory.

The communication scheme proposed by Shannon (Fig. 2) states that a *source* produces messages, which are adapted by a *encoder* and then are sent over a *channel* and are decoded by a *decoder* to the final *destination*. The pair of source and encoder constructs the *emitter* and the pair of decoder and destination constructs the *receiver*. The input of the channel by the *emitter* is X and the output of the channel to the *receiver* is Y. Since there might be *noise* in the channel, output Y does not always equal input X. The engineering process to transmit a source message X to the other side of the channel where the message Y is received, does not concern the semantic aspect of communication, but is only related to the probability of each possible outcome [188]. Therefore, the channel is completely described by the probability of Y conditional on X: $p(Y|X)$.

Information theory covers many aspects of the communication process including efficient encoding and decoding schemes that match the channel to ensure reliable transmission. The following introduces a few key concepts that are of importance to this thesis.

1.1 INFORMATION AS ENTROPY

"Shannon's theory does not deal with 'information'as that word is generally understood. Instead, it deals with data – the raw material out of which information is obtained." [41]

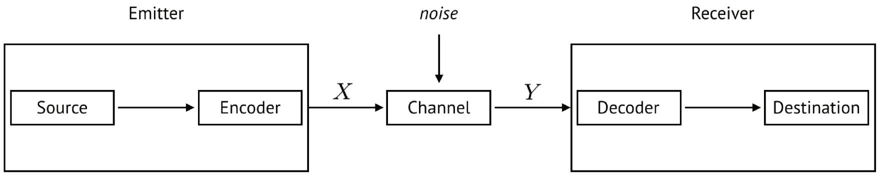


Figure 2: Shannon's communication scheme.

Information in the sense of information theory is defined against our common sense. We ordinarily think of information as a collection of facts, a file of meaningful data. The key to Shannon's theory is precisely that he deliberately avoided the question of meaning. Here information measures *randomness* or *uncertainty* of the outcome of a random variable and is captured by an entropy function, defined as follows (entropy of a discrete random variable X):

$$H(X) = -\sum_{x} p(x) \log_2 p(x). \tag{1}$$

where X is drawn according to the probability distribution $p(x) = P(X = x)$ and entropy $H(X)$ is measured in bit. The higher the entropy, the more uncertain the outcome, the harder the prediction. Entropy measures "information" in the sense that the outcome of a random variable will increase the receiver's knowledge (or decreases the receiver's uncertainty).

A simple example is the weather on the next day. If the chance of rain is 0% and the chance of sun is 100%, the entropy is 0 as it is a sure event. However, if the chance of rain and chance of sun are 50% each, the entropy reaches its maximum, 1 bit, as the uncertainty about the weather is maximal. In other words, a message brings maximum "information" to those who receive it. Equiprobable messages generate maximum entropy.

Entropy is bounded by sure event and maximum random event $0 \leqslant H(X) \leqslant \log_2 N$:

- Entropy is zero if the event is sure or it is impossible:
$$H(X) = 0 \qquad \text{if} \qquad p(x) = 0 \text{ or } 1.$$
- Entropy of a set of N equiprobable messages:
$$H(X) = \log_2 N \qquad \text{if} \qquad p(x) = \tfrac{1}{N}.$$

1.2 MUTUAL INFORMATION AND EQUIVOCATION

Since information is transmitted over a noisy channel, some information might get lost. The actually transmitted information, which is captured by mutual information, characterizes the amount of information that is *effectively* transmitted through the channel. Mutual information of two discrete random variables X and Y is defined as follows:

$$I(X;Y) = \sum_{y} \sum_{x} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = H(X) - H(X|Y). \tag{2}$$

where $p(x,y)$ is the joint probability function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively.

Mutual information is also bounded by two quantities $0 \leqslant I(X;Y) \leqslant H(X)$:

- If no messages get transmitted from the source to the receiver, mutual information is 0;
- If all messages get transmitted from the source to the receiver, mutual information is entropy $H(X)$.

Continuing with the weather example: if a person needs to tell a friend about the weather she experienced last week, she says"rainy on Monday, sunny on Tuesday, rainy on Wednesday, sunny on Thursday, rainy on Friday and sunny on Saturday". The information she is transmitting is $H(X) = 1$ bit, since $P(\text{rain}) = 50\%$ and $P(\text{sun}) = 50\%$. If her friend perfectly receives all the information, the mutual information is $I(X;Y) = H(X) = 1$ bit. But if her friend does not receive anything, the mutual information is 0. If her friend is distracted and hears "rainy on Monday, sunny on Tuesday, rainy on Wednesday, sunny on Thursday, rainy on Friday and *rainy* on Saturday", most of the information is transmitted but one day's weather condition is lost. Here the mutual information is between 0 and 1 bit.

The information lost in transmission is captured by equivocation $H(X|Y)$ (Equation 2). It describes the receiver's uncertainty about the source after the transmission given the channel output Y. In an ideal channel without noise, equivocation $H(X|Y)$ would be zero and mutual information $I(X;Y) = H(X)$: information is perfectly transmitted from the source to the destination.

Equivocation is related to how errors are made. Particularly, Fano's inequality [34, Theorem 2.4.1] relates the average information lost in a noisy channel to the probability of the categorization error:

$$H(X|Y) \leqslant H(E) + P_e \times H(Z|E = 1). \tag{3}$$

where random variable E represents errors, P_e represents error rate and random variable Z represents the noise in the channel that perturbs the effective transmission due to errors. We will provide a more detailed discussion in Chapter 15 when this notion is needed.

1.3 CHANNEL CAPACITY AND THROUGHPUT

The "information" channel has a certain capacity, computed as the maximum amount of mutual information $I(X;Y)$ conveyed by the channel. It is defined as (a discrete memoryless channel):

$$C = \max_{p(x)} I(X;Y). \tag{4}$$

where the maximum is taken over all possible input distributions $p(x)$.

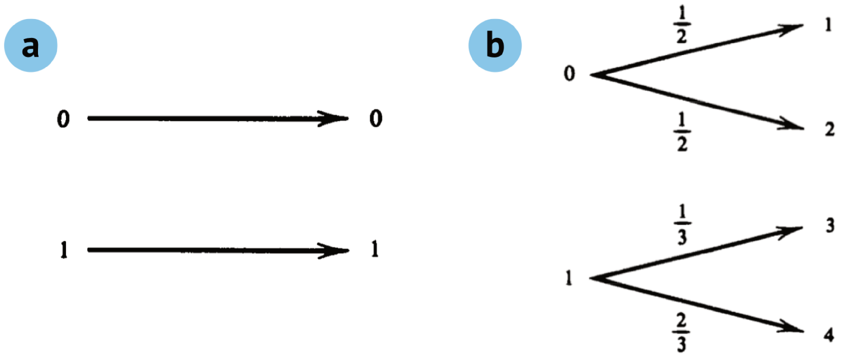


Figure 3: (a) Noiseless binary channel and (b) Noisy channel with nonoverlapping outputs (adapted from [34]).

Suppose we have a channel whose binary input is reproduced exactly at the output (Fig. 3 (a)). In this case, any transmitted bit is received without error. Therefore, 1 error-free bit can be transmitted per use of the channel, and the capacity is 1 bit. We can also calculate the information channel capacity $C = \max I(X;Y) = 1$ bit, which is achieved by using $p(x) = (\frac{1}{2}, \frac{1}{2})$.

Fig. 3 (b) shows two possible outputs corresponding to each of the two inputs. The channel appears to be noisy, but in fact is not. Even though the output of the channel is a random consequence of the input, the input can be determined from the output, and hence every transmitted bit can be recovered without error. The capacity of this channel is also 1 bit per transmission. We can also calculate the information channel capacity $C = \max I(X;Y) = 1$ bit, which is achieved by using $p(x) = (\frac{1}{2}, \frac{1}{2})$.

These are just two examples from Cover and Thomas' book [34]. Depending on the channel usage and operations, the computation of channel capacity differs. It defines, however, the tight upper bound on the rate at which information can be reliably transmitted over a communication channel.

Note that Shannon's theorem states that the natural metric of a discrete-time channel capacity is bits per channel use, such as in the above-mentioned examples. However, if we are told how far apart in time the discrete time instants are, e. g. one channel use per microsecond, then a capacity C bits per channel use can be stated as bits per second. If the inputs and outputs are continuous-time signals that occupy bandwidth W, channel capacity is naturally measured in bits per second per Hertz, or simply bits.

The well-known Shannon's theorem [188, Theorem 17], which inspired Fitts' law [63], applied the channel capacity concept to an additive white Gaussian noise (AWGN) channel with B Hz bandwidth and signal-to-noise ratio S/N, measured in bits per second:

$$C = B \log_2 \left(1 + \frac{S}{N}\right) \tag{5}$$

Furthermore, the theorem states that given a noisy channel with channel capacity C and information transmitted at a rate R, then if $R < C$, there exists a code that allows the probability of error at the receiver to be made arbitrarily small [34, Theorem 8.7.1]. This transmission rate R is widely used in wireless network communication, packet-based schemes, etc. to measure an effective speed of data transmission, which is also known as throughput (TP). One common computation of throughput is dividing successfully transmitted information (mutual information) by the time it takes to transmit such information. For instance, if a friend is telling another friend perfectly about the weather condition in 10 seconds (T), then the throughput in this case is:

$$TP = \frac{I(X;Y)}{T} = 1/10 = 0.1 \quad \text{bits/s.} \tag{6}$$

The notions of entropy and mutual information are used throughout the thesis. Equivocation and throughput will be discussed and compared in Part iii.

2

INFORMATION THEORY IN PSYCHOLOGY

Although information theory is still alive and well in a number of fields, it went through a rather interesting development in psychology: experimental psychologists were first swept by a wave of excitement for information theory during the 1950s and 1960s, then experienced a period of critical analysis and finally decided on the incompatibility between information theory and psychology. In the article *Whatever Happened to Information Theory in Psychology?*, Luce [136] explains that "... after an initial fad in psychology during the 1950s and 1960s it [information theory] no longer is much of a factor, beyond the word *bit*, in psychological theory." While it is still well applied in biology, engineering, computer science, physics, and statistics, it is true that psychologists today are no longer supporters of information theory.

In this chapter, I examine the dramatic changes of information theory in psychology.

2.1 ENTHUSIASM AT THE EARLY STAGE

Even though Shannon himself strongly preferred the term *communication theory* to *information theory*, psychologists in the 1950s and 1960s seemed to be thrilled by the ability to quantitatively measure information and to investigate human information capacity in various psychological contexts. Note that the applications of information theory during this period *exclusively* explored these two concepts but left most of the engineering parts (e. g. channel coding) of the transmission process aside.

2.1.1 *Measuring Information*

The Entropy of Language An important example of an information source is English text [188, 189]. If we assume that the alphabet of English consists of 26 letters and the space symbol, and ignore punctuation and the difference between upper and lower case letters, we can construct models of English using empirical distributions collected from samples of text [1]. Using such a method, Shannon [188] estimated that the entropy of English is 4.14 bits per letter.

---

[1] The frequency of letters in English is far from uniform. The most common letter E has a frequency of about 13% while the least common letters, Q and Z, occur with a frequency of about 0.1%.

We can also build more complex models by incorporating conditional probability as we know that the frequency of pairs of letters is also far from uniform. For example, the letter Q is always followed by a U. The most frequent pair is TH, which occurs with a frequency of about 3.7%. We can use the frequency of the pairs to estimate the probability that a letter follows any other letter. For example, to build a fourth order Markov approximation, we must estimate the values of $p(x_i|x_{i-1}, x_{i-2}, x_{i-3}, x_{i-4})$. Such a model gives an estimation of 2.8 bits per letter.

Similarly, Shannon estimated the word-entropy of printed English as 11.82 bits per word. Later on, Grignetti [85] estimated the word entropy in printed English as 9.83 bits using a different word sample. Miller et al. [154, 155] also studied the word context, particularly the extent to which the prior occurrence of certain verbal elements (word choice) influences the talker's present choice. For instance, if the talker has said "children like to," his choice for the next word in this pattern is considerably limited – elephant, punished, loud, Bill, and many other words are highly unlikely continuations.

These statistics of English are useful in decoding encrypted English text and in word prediction. A commonly used model is the trigram (second-order Markov) word model, which estimates the probability of the next word given the previous two words, as seen in intelligent text input and speech recognition systems these days. We can also apply the techniques above to estimate the entropy rate of other information sources such as images and other multimedia content.

The Information in Stimuli The relationship between the number of alternate stimuli and choice-reaction times was first reported by Helmholtz [95] in 1850, Donders in 1868 [42] and then by Merkel in 1883 [151]. Using 1 to 10 alternatives, Merkel discovered that it takes longer to respond to a stimulus when it belongs to a large set as opposed to a smaller set of stimuli. This was later on taken by psychologists, notably Hick [98] and Hyman [104], as an analogy to information theory: the display is the transmitter of information; each alternate stimulus the message; the sensory-perceptual system the channel; the participant the receiver, and the appropriate action the destination [119] (Fig. 4).

Hick was clearly motivated by finding a formula to capture the "reaction-time era" as other psychologists discussed the increase in reaction time with the number of alternatives and attributed it to such causes as the division of attention or a reduction in the effective intensity of the stimulus, but not with quantitative theory. The only reference to a mathematical relation between reaction time and number of alternatives was by Blank [20], where a logarithmic relation was mentioned without further explanation.

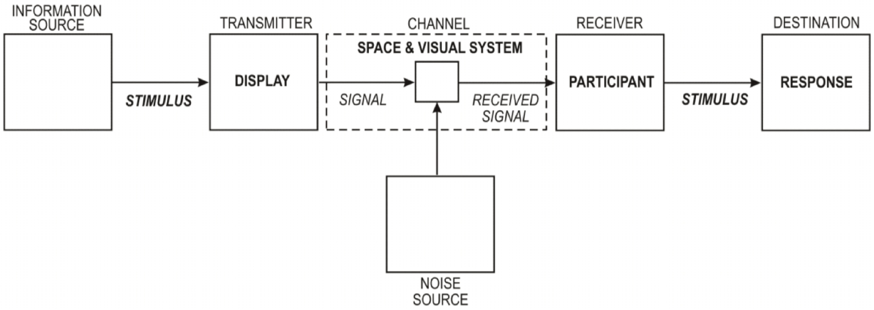


Figure 4: Schematic diagram of the choice-reaction time experiment as a model of a communication system. Taken from [119].

To find out this quantitative formula, Hick [98] conducted 3 experiments, replicating Merkel's experiment [151]. The appartus was the same for all 3 experiments: ten lamps were arranged in an irregular circle formation and connected to a device that was coded to light up one random lamp every 5 seconds [97]. Each of the participants' fingers was connected to a Morse key [2] corresponding to the lamps. The participants' task was to press the correct key for a lighting of a particular lamp. Both stimulus presentation and response were recorded in binary code by moving paper. A uniform distribution of stimuli was used.

The first experiment was carried out to confirm the fitness of $\log(n+1)$ to reaction time RT, rather than $\log n$, which is the entropy formula accounting for equally probable choices in information theory. Hick himself served as the only participant in the first experiment, varying the number of stimuli from 2 to 10. He trained himself over 8,000 trials before the experiment and removed incorrect reactions, so that the transmitted information equals the entropy of stimuli. He showed that $\log(n+1)$ indeed gave a better fit than $A + B \times \log n$.

The second experiment included Hick himself and another participant with the aim of testing the relationship between reaction time and partially extracted information. Both participants were well trained before the experiment and were instructed to make errors, which were included in the analysis. Hick estimated the joint probability distribution between the stimuli and the actual reactions and computed the transmitted information ($I(X;Y)$ in Equation 2). Then he introduced the notion of *degree of choice* n_e where $I(X;Y) = \log n_e$, and showed that in the case of partially extracted information, $\log(n_e + 1)$ also provided a better fit than $A + B \times \log n_e$. The third experiment confirmed that learning was not an issue. He therefore concluded that "the amount of information extracted R is proportional to the time taken to extract it, on average" where R is defined as:

$$R = \log_2(n+1) \quad \text{or} \quad R = \log_2(n_e+1). \tag{7}$$

---

[2] Also known as telegraph key: a switching device used primarily to send Morse code.

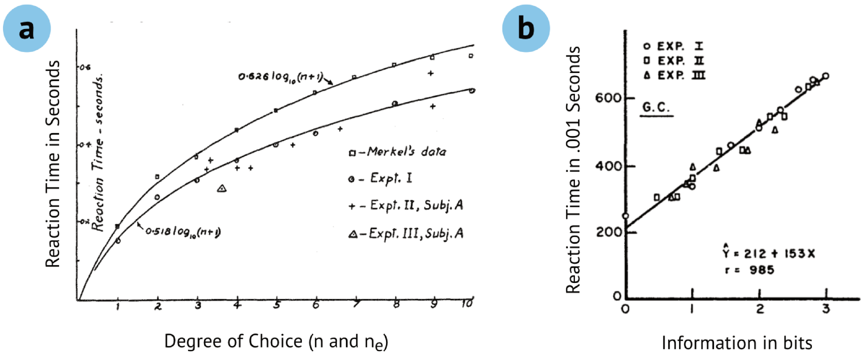


Figure 5: (a) Hick's data as a function of Degree of Choice and (b) Hyman's data (Subject 1) as a function of information in bits. Taken from [98] and [104] respectively.

where $n_e = n$ if no errors are made and all information gets transmitted. The "+1" accounts for the participants' uncertainty about a "no stimulus" condition. Reaction time RT $\propto$ R, therefore the constant rate of gain of information.

Interestingly, Hick noticed the potential discrepancy between the subjective probability and objective frequency. He mentioned that "Although the differences with respect to individual stimuli and responses suggest that the subjective – or perhaps one should say the psychologically effective – probabilities do not exactly correspond to the objective frequencies, it will be an enormous practical advantage if, for the purpose of estimating average effects, it can be assumed that they do. To discover the limits within which that assumption is justifiable will require a great deal of experimentation". Later research in psychology and cognitive science indeed pointed out that subjects react to task-relevant stimuli with subjective probability [46]. Unlike what Hick proposed "It may be found both practicable and valid, in some cases, to estimate subjective probabilities by some form of *guessing* technique", Duncan and Donchin [46] used the P300 wave [3] to determine the participants' reaction.

Hick plotted data as a function of the number of alternatives (n or n_e) and did not explicitly postulate a linear relationship between reaction time RT and the transmitted information R (Fig. 5 (a)). Crossman [35] in 1953, using a card-sorting task, presented data that were plotted as a function of R. In the same year, Hyman [104] also plotted data as a function of R and suggested that a linear function of stimulus information within the range of 0 to 3 bits could be considered (Fig. 5 (b)).

---

[3] The P300 (P3) wave is an event related potential (ERP) component elicited in the process of decision making. It can be recorded by e. g. electroencephalography (EEG).

Hyman used 8 lights in a matrix of 36 lights (6 rows by 6 columns) display and used names – Bun, Boo, Bee, Bore, By, Bix, Bev, and Bate – to designate them. At the beginning of each trial, the experimenter gave a warning signal and 2 seconds later turned on one of the 8 lights and started a timer. Participants responded by calling out the designated name of the light. A throat microphone attached to the participant activated an electronic voice key to stop the timer. Four subjects participated in the experiments and they all attended more than 40 experimental sessions over a 3-month period with approximately 15,000 reaction times recorded for each subject. All errorless performance.

The first experiment replicated Merkel's and Hick's experiment using 8 conditions with different numbers of equally probable alternatives. The second experiment had 8 conditions which involved different numbers of alternatives and different probabilities of the occurrence of these alternatives, therefore varying the average information content by altering the probability of occurrence of each choice. The last experiment also had 8 conditions and introduced sequential dependencies between successive choices of alternatives. In each condition, each of the alternatives had equal likelihood of occurring but its probability was conditional. For example, in condition 1, where two alternatives were used, the conditional probability of b given that a has occurred was $p(b|a) = 0.8$. These conditions yielded entropies ranging from 0.72 to 2.81 bits.

With the three experiments, Hyman [104, p.196] concluded:

> *"The reaction time to the amount of information in the (visual) stimulus produced a linear regression for each of the three ways in which information was varied."*

His formula is written as:

$$RT = a + b \times H_T. \tag{8}$$

where RT is reaction time, a and b are empirically determined constants, and H_T is the transmitted information captured by $\log_2 n$ for equiprobable stimuli or $-\sum_{i=1}^{n} p_i \log_2 p_i$ for non-uniformly distributed stimuli with probability p_i.

2.1.2 *Investigating Information Capacity*

Information Capacity of Working Memory *The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information* by Miller [153] is one of the most highly cited papers in psychology. Miller demonstrated that the number of objects an average human can hold in working memory is 7 ± 2.

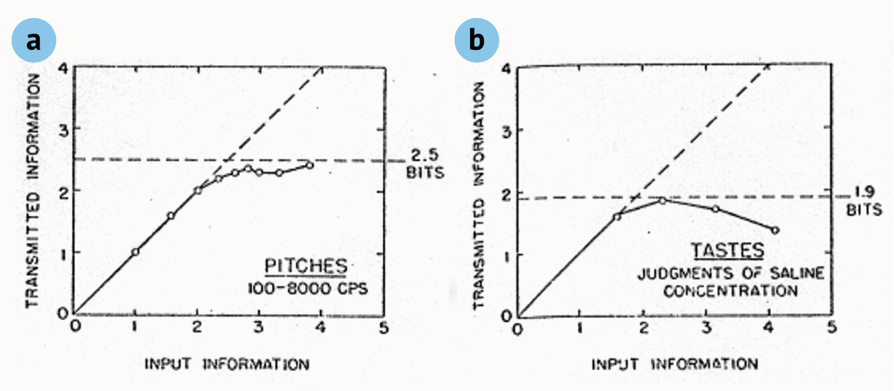


Figure 6: Channel capacity for subjects who make absolute judgments of (a) auditory pitch (Pollack [172, 173]) and of (b) saltiness (Beebe-Center, Rogers, and O'Connell [13]). Taken from [153].

Similar to the stimulus-response paradigm (Fig. 4), Miller [153] assumed that in experiments on absolute judgment, the observer is considered to be a communication channel. The amount of information in the stimuli is transmitted as input and the amount of information in the subjects' responses is the output. The experimental problem is to increase the amount of input information and to measure the amount of transmitted information. If the observer's absolute judgments are quite accurate, then nearly all of the input information will be transmitted and will be recoverable from his responses. If he makes errors, then the transmitted information may be considerably less than the input. Miller was interested in investigating the notion of channel capacity: it represents the greatest amount of information for which an observer can match his responses to the given stimuli on the basis of an absolute judgment.

Miller demonstrated the validity of this concept of capacity with various examples from previous studies using unidimensional stimuli: a subject is presented with a number of stimuli that vary on one dimension and responds to each stimulus with a corresponding response. Unlike the Hick-Hyman paradigm, however, the participants in these studies took as much time as needed to identify the stimulus.

These one-dimension stimuli include tones [172, 173], loudness [76], taste intensities [13], visual position [91], and the sizes of squares [50]. All of them illustrated the leveling off effect of transmitted information as a function of input information, some sooner than others. For instance, most of us can identify about 1 out of 6 pitches (2.5 bits [172, 173]) and 1 out of 4 saline concentration levels (1.9 bits [13]) before we begin to get confused (Fig. 6). Therefore, people's maximum performance on unidimensional absolute judgment can be characterized as an information channel capacity with approximately 2 to 3 bits of information, which corresponds to the ability to distinguish among 4 to 8 alternatives.

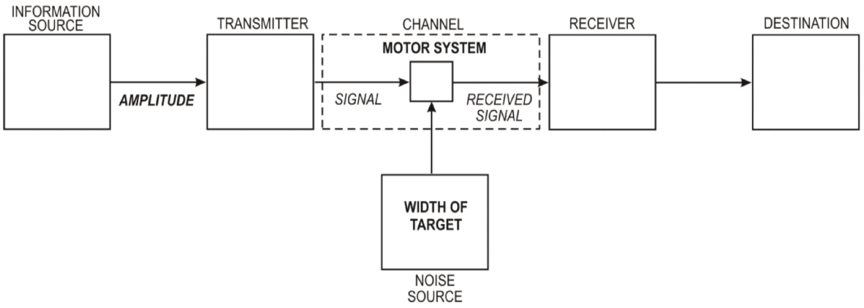


Figure 7: Schematic diagram of human motor system experiments as a model of communication system. Taken from [187].

Miller also examined absolute judgments of multidimensional stimuli and memory span where the magic number 7 does not apply. Memory span refers to the longest list of items (e. g. digits, letters, words) that a person can repeat back in correct order on 50% of trials immediately after presentation. Miller found that memory span is not limited in terms of bits but rather in terms of chunks. A chunk is the largest meaningful unit in the presented material that the person recognizes – therefore, what counts as a chunk depends on the knowledge of the person being tested. Although he did observe that memory span of young adults is approximately 7 items, Miller only used information-theoretic terms for interpreting the unidimensional absolute judgment tasks and mentioned that the correspondence between the limits of one-dimensional absolute judgment and of short-term memory span was only a coincidence [153].

Information Capacity of Motor Movement In 1954, Fitts published the paper *The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement*, which according to himself, was in line with Miller's reasoning [153] and was an analogy to Shannon's Theorem 17 [4] [188]. Fitts was obviously motivated by applying information theory as many other psychologists at the time, as he started the paper with:

> *"Information theory has recently been employed to specify more precisely than has hitherto been possible man's capacity in certain sensory, perceptual, and perceptual-motor functions. The experiments reported in the present paper extend the theory to the human motor system."*

Adopting Shannon's Theorem 17, Fitts conceptualized the human motor system as a communication channel, movement amplitude A as a signal, and the target width W as noise (Fig. 7).

---

[4] Recall Shannon's Theorem 17: $C = B\log_2\left(1+\frac{S}{N}\right)$ (Equation 5)

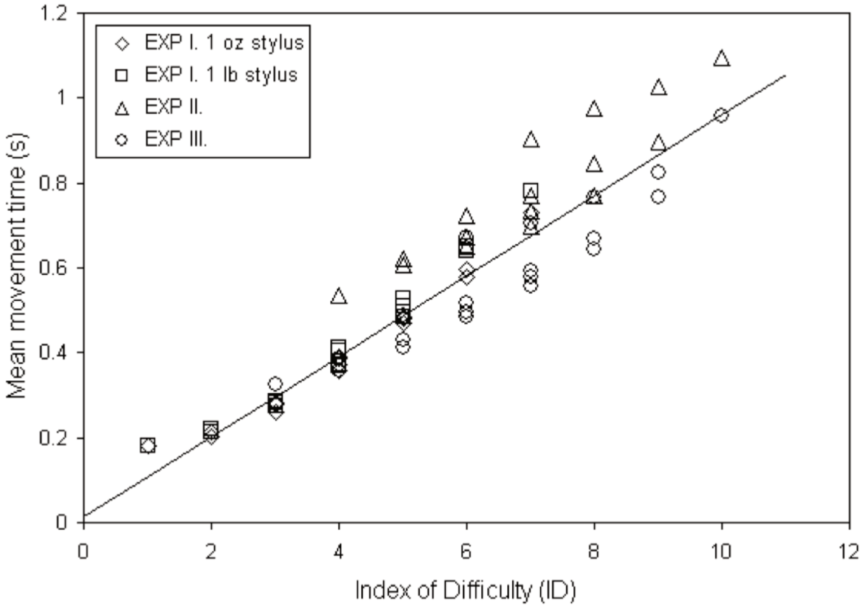


Figure 8: Results of Fitts' 4 experiments in the 1954 paper. Taken from [187].

Fitts ran 4 experiments using reciprocal tapping, disk transfer, and pin transfer tasks and combined various levels of *A* and *W*. In the reciprocal tapping task in Experiment 1, participants used a metal-tipped stylus (1 oz. version on the first day; 1 lb. version on the second day) to tap two stationary strips of metallic targets. The width of the plates (*W*) varied from 0.25 to 2 inches, and the distance between them varied from 2 to 16 inches. Participants were instructed to strike the targets alternately to score as many hits as possible. In other words, accuracy was encouraged. In the disk transfer task in Experiment 2, participants were instructed to transfer and stack round plastic discs (with holes drilled through the middle) from one pin to another. Holes of different sizes and pins of different diameters were used. In the pin transfer task in Experiment 3, participants were instructed to transfer pins of different diameters from one set of holes to another set of holes. Participants were instructed to work at their maximum rate.

The results of these 4 experiments can be found in Fig. 8. Note that the figure comes from a reanalysis of Fitts' data by Mackenzie [138]. In the 1954 paper, Fitts did not elaborate on the relationship between movement time and index of difficulty. He proposed two concepts: Index of Difficulty (ID) and Index of Performance (the rate of performance IP).

Index of Difficulty (ID) states that the minimum amount of information required to produce a movement having a particular average amplitude plus or minus a specified tolerance (variable error) is proportional to the logarithm of the ratio of the tolerance to the possible amplitude range:

$$\mathrm{ID} = \log_2\left(\frac{2A}{W}\right). \tag{9}$$

Index of Performance (IP) shows the capacity of the human motor system. It is measured in bits per unit time and is homologous to the rate of gain of information in Hick's paradigm [98] and analogous to Shannon's channel capacity:

$$\mathrm{IP} = \frac{\mathrm{ID}}{\mathrm{MT}}. \tag{10}$$

where MT is the empirically determined movement time.

Fitts reported that IP ranged from 10.3 to 11.5 bits/s in Experiment 1; 7.5 to 10.4 bits/s in Experiment 2; and 8.9 to 12.6 bits/s in Experiment 3. He concluded that the rate of performance (IP) in a given type of task is approximately constant over a considerable range of movement amplitudes and tolerance limits, but falls off outside this optimum range. The level of optimum performance was found to vary slightly among the three tasks in the range between about 10 to 12 bits/s.

It was not until 1964, by examining the effects of response amplitude and terminal accuracy on 2-choice reaction time (RT) and on movement time (MT) that Fitts found the correlation between ID and MT was found to be above 0.99 over the ID range from 2.6 to 7.6 bits per response [65]. Therefore, the Fitts' law that we know today is written as:

$$\mathrm{MT} = a + b \times \mathrm{ID}. \tag{11}$$

where a and b are empirically determined constants.

All above-mentioned studies except Fitts (1964) were done during the 1950s and most of these applications were summarized in a book by Attneave [7].

2.2 FROM CRITICISM TO ABANDONMENT

While psychologists were still enthusiastic about applying information theory, Shannon and the information theory community started to challenge the use of information theory outside the sphere of communication engineering.

Shannon himself was among the skeptics as he is quoted as saying "Information theory has perhaps ballooned to an importance beyond its actual accomplishments" (cited in [110]). He insisted that "the use of a few exciting words like information, entropy, redundancy, do not solve all our problems" [190]. Elias [47], an important figure of the information theory community, urged authors to stop writing papers using information theory outside of its intended scope.

Reflecting on its applications in psychology, McGill [149] also stated that "The somewhat fortuitous marriage of the information measures and information theory may, in the long run, prove to have confused psychologists as much as it has stimulated them." He mentioned that perhaps the most important reason is that Shannon's information measure is not the sort of information with which we are familiar, and psychologists have made very little use of the performance criteria and of the basic theorems of information theory apart from the notion of channel capacity.

Garner [77] not only summarized the ideas and experiments of information-theoretic applications in psychology, but also expanded the interest to the relation between mean response times and the uncertainty of the stimuli to which participants were responding. In early experiments, mean response time appeared to grow linearly with uncertainty, but glitches soon became evident. Laming [119] in the late 1960s also commented on the choice-response paradigm that "This idea does not work... there are further unpublished results that show it to be hopeless". Substantial sequential effects exist between a stimulus and at least the immediately preceding stimulus-response pair, but with the magnitude of the correlation dropping from close to one for small signal separation in either decibels or frequency to about zero for large separations [81]. Similarly, Bertelson [18] expressed that the paradigm could be explained as a sequential effect independently of stimulus entropy.

Gradually, as the importance of this reality began to set in at the end of the 1960s, one saw fewer – although still a few – attempts to understand global psychological phenomena in simple information theory terms. When Shannon died on February 24, 2001, at age 84, several psychologists paid homage to this creator of information theory by looking back at history.

The same year, Laming [121] provided a detailed critique. He mentioned that Shannon's way of defining capacity requires that not individual signals be transmitted but rather very long strings of them so as to get rid of redundancies. This is rarely possible within psychological experiments, e. g. a choice-reaction experiment involves the transmission of single stimuli, one at a time, a condition that affords no opportunity for the sophisticated coding on which Shannon's theorem depends.

Furthermore, under the influence of Shannon's theory, psychologists are inclined to suppose that information is absolute. The truth is that it is not. Data is absolute, but information is always relative to the two hypotheses between which it distinguishes. Criticizing the human observer as a physical system, Laming [120] also put forward the difference between information available to the observer and the partitioning of values of that information between the available responses (the choice of criteria). As he said "Looking solely at information throughput, and disregarding the criteria, it can be shown that the information available to the observer is derived from a sensory process that is differentially coupled to the physical stimulus, because the component of information derived from the stimulus mean is entirely absent from the information implicit in the observer's performance".

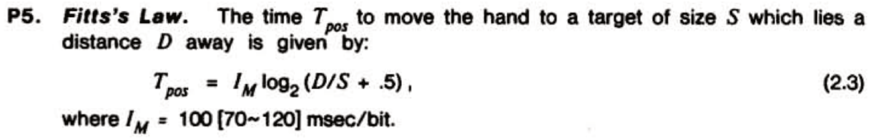
Luce [136] in 2003 echoed this statement by further elaborating on the incompatibility between information theory and psychology. He argued that the elements of choice in information theory are absolutely neutral and lack any internal structure; the probabilities are on a pure, unstructured set whose elements are functionally interchangeable. That is fine for a communication engineer who is totally unconcerned with the signals communicated over a transmission link; interchanging the encoding does not matter at all. By and large, however, the stimuli in psychological experiments are to some degree structured, and so, in a fundamental way, they are not in any sense interchangeable. If one is doing an absolute judgment experiment of pure tones that vary in intensity or frequency, the stimuli have a powerful and relevant metric structure, namely, differences or ratios of intensity and frequency measures between pairs of stimuli. Similarly, if one does a memory test, one has to go to very great pains to avoid associations among the stimuli. Stimulus similarity, although still ill understood and under active investigation, is a powerful structural aspect of psychology.

In summary, the word *information* has been almost seamlessly transformed into the concept of "information-processing models" in which information theory per se plays no role. The idea of the mind being an information-processing network with capacity limitations has stayed with us, but in far more complex ways than pure information theory.

3

PSYCHOLOGY AND HCI – FITTS' LAW

Newell, Card and colleagues [161, 162] first articulated the prospective role of psychology in HCI in the early 1980s. Particularly, they presented laws as design principles regarding the perceptual system, the motor system, and the cognitive system for developers to maximize usability in the design of human-computer interfaces. Chapter 2 *The Human Information-Processor* listed 9 principles of operation to model human information processing [161, p.27], which included the two information-theoretic attempts from psychology: Fitts' law (Fig. 9) and the Uncertainty principle (Hick's law).



P5. ***Fitts's Law.*** The time T_{pos} to move the hand to a target of size S which lies a distance D away is given by:

$$T_{pos} = I_M \log_2 (D/S + .5) , \qquad (2.3)$$

where I_M = 100 [70~120] msec/bit.

Figure 9: Card, Moran and Newell's description of Fitts' law. Taken from [161, p.27].

Since the 1980s, Fitts' law has welcomed a large number of applications and research efforts in the HCI community are still ongoing [1]. It is recognized as "the law of pointing", and is regularly used for, e. g. device evaluation or interface design. Being successfully applicable to all sorts of conditions including restricted visual feedback [220], different types of participants (e. g. elders [10] and children [103]), and different environments such as under water [113], the law has proven to be largely robust from the *empirical* point of view. Its *theoretical* foundation, however, has been challenged many times in many frameworks.

This chapter reviews the theoretical studies on Fitts' law. Fitts described his interpretation of human motor performance as follows:

> *"If the amplitude and tolerance limits of a task are controlled by E (the experimenter), and S (the participant) is instructed to work at* ***his maximum rate****, then* ***the average time per response*** *will be directly proportional to* ***the minimum amount of information*** *per response demanded by the particular conditions of amplitude and tolerance."* [63, p.2]

---

[1] The fundamental topic of human motor performance has invited much research interest in psychology, most of which was summarized in the book chapter by Meyer and colleagues [152].

3.1 WHICH MINIMUM AMOUNT OF INFORMATION?

Fig. 9 shows Card, Moran and Newell's description of Fitts law. Note that the version shown in their book was not Fitts' original formula. This formula was in fact proposed by Welford [213], who argued that the formulation $\log_2\left(\frac{D}{W}+0.5\right)$ makes movement time dependent on a kind of Weber fraction [2] in that the subject is called upon to distinguish between the distances to the far and the near edges of the target.

Fitts originally denoted $\mathrm{ID} = \log_2 \frac{2D}{W}$ to express the *minimum amount of information*. This expression is still being used in psychology today [156, 170]. The HCI community, on the other hand, has unanimously adopted Mackenzie's formulation to describe task difficulty, written as:

$$\mathrm{ID} = \log_2(1 + \frac{D}{W}).$$

This improvement, according to Mackenzie [138], was more consistent with Shannon's Theorem 17 and the available empirical data. Gori et al. [80] recently used the notion of "geometrically uniformly distributed targets" and proved that this version of ID is indeed equivalent to Shannon's Capacity.

Mackenzie [138] also proposed to measure movement endpoints from the center of the target and, assuming that the distributions of these measures is normal, to compute an *effective* index of difficulty ID_e, written as:

$$\mathrm{ID}_e = \log_2\left(1 + \frac{\bar{D}}{W_e}\right). \tag{12}$$

where $\bar{D}$ corresponds to the average covered distance and W_e is the effective width. The computation of W_e is detailed in [197]. Let σ denote the standard deviation of the endpoint distribution, and ε the error rate, e. g. the proportion of target misses:

$$\text{If } \sigma \text{ is available:} \quad W_e = 4.133\sigma$$

$$\text{If not:} \quad W_e = \begin{cases} W \times \frac{2.066}{z(1-\varepsilon/2)} & \text{if } \varepsilon > 0.0049\% \\ 0.5089 \times W & \text{otherwise.} \end{cases} \tag{13}$$

---

[2] The Difference Threshold (or "Just Noticeable Difference") is the minimum amount by which stimulus intensity must be changed in order to produce a noticeable variation in sensory experience. It was first proposed by Ernst Weber, who observed that the size of the difference threshold appeared to be lawfully related to initial stimulus magnitude [52]. This relationship, known as Weber's law: $\frac{\Delta I}{I} = k$ where ΔI represents the difference threshold, I represents the initial stimulus intensity and k signifies that the proportion on the left side of the equation remains constant despite variations in the I term.

Equation 13 is based on Crossman [36], who was the first to try to incorporate the error rate information into his ID measure, leveraging the standard Gaussian distribution model. Fitts, on the other hand, did not measure actual amplitudes, but classified the movements in a dichotomous way as hits and misses. In 2002, the ISO 9241 standard [199] was published, providing standards for human-computer interface testing, including Mackenzie's effective index of difficulty.

Recently, Gori et al. [80] challenged the standard for three reasons: (1) The justification of a Gaussian distribution of endpoints is empirically questionable; (2) Information theory provides no justification for the relation $W_e = 4.133\sigma$ and (3) The threshold of error rate placed at 0.0049% (Equation 13) is arbitrary. Instead, they proposed a compliant Index of Difficulty $\mathrm{ID}(\varepsilon) = (1-\varepsilon)\log_2(1+\frac{D}{W})$ where ε accounts for error rate.

So far, we have identified the following 5 different indices of difficulty:

- Fitts [63]: $\mathrm{ID} = \log_2 \frac{2D}{W}$;
- Welford [213]: $\mathrm{ID}_w = \log_2 \left(\frac{1}{2} + \frac{D}{W}\right)$;
- Mackenzie [138]: $\mathrm{ID} = \log_2(1 + \frac{D}{W})$;
- Mackenzie adjusted [197, 199]: $\mathrm{ID}_e = \log_2 \left(1 + \frac{\bar{D}}{W_e}\right)$;
- Gori [80]: $\mathrm{ID}(\varepsilon) = (1-\varepsilon)\log_2(1 + \frac{D}{W})$.

According to Gori et al. [80], Mackenzie's ID conforms to the notion of Shannon's channel capacity. For empirical studies, the behavior of the standardized ID_e for vanishing error rates is problematic as $\lim_{\varepsilon \to 0} \mathrm{ID}_e = \infty$ (Fig 10). In contrast, $\mathrm{ID}(\varepsilon)$ is more consistent with the fundamentals of information theory, and is continuous towards the zero error rate region, thus allowing the researcher to dispense with an arbitrary treatment of the 0 percent miss case, as well as is much simpler to derive than the ISO-recommended ID_e. However, the comparison of ID_e and $\mathrm{ID}(\varepsilon)$ seems to provide essentially the same results (Fig. 10). More research is needed to evaluate the implications of such result.

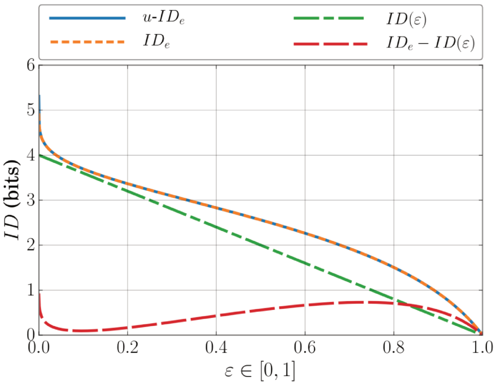


Figure 10: Figure 8 in Gori et al. [80] comparing $\mathrm{ID}(\varepsilon)$ and ID_e for erasure rate in [0,1], for $D = 15$.

3.2 THE AVERAGE OF MINIMUM OR THE AVERAGE OF AVERAGE?

Fitts needed the participant (S) to work at his maximum rate, so that the resulting movement times MT reflected S's full commitment to the pointing task. Therefore, the *average time per response* in Fitts' term corresponds to "average minimum", which seemed to have confused many. Authors in Fitts' law research use different wordings, which suggest other interpretations of MT: Soukoreff and Mackenzie [197] considered "movement time performance for rapid aimed movements", Hoffman [99] "movement time", and Drewes [44] "mean time". Consequently, Fitts' law has always been considered as a law of average performance.

Yet historically speaking, Fitts' information-theoretic rationale for aiming movements considers the transmission of information about the target through a human motor channel. Fitts' law can be derived by computing the capacity of this channel, which is a theoretical upper bound – the maximum amount of information that can be transmitted reliably – and which is accordingly calculated as an extreme through the Channel Coding Theorem – the maximum of mutual information over all input distributions. Hence, only movements that maximize transmitted information should be relevant for the derivation of Fitts' law. But can one always reach his or her maximum performance? Participants can be instructed to perform as fast as they could in a controlled experiment. In the real world, however, one rarely tries to point as fast and as accurately as possible. Even in a controlled experiment, participants' attention fluctuates.

Building on Guiard and colleagues [88, 89], Gori et al. [80] recently argued that Fitts' law should be considered as a law of performance limit. They introduce this concept by reanalyzing the data from a pointing study run "in the wild" [31]. For several months Chapuis et al. [31] unobtrusively logged cursor motion from several participants using their own computer. The authors were able to identify offline the start and end of pointing movements as well as the target information, for several hundreds of thousands of click-terminated movements. With this information, one can then represent the movements in a MT versus ID graph, as normally done in a controlled Fitts' law study. To compute task difficulty in the 2D space of computer screens, they followed the suggestion of Mackenzie and Buxton [140] and chose $\mathrm{ID} = \log_2\left(1 + \frac{D}{\min(H,W)}\right)$ where H and W are the height and width of the target, respectively. Whenever an item was clicked, it was considered the target, meaning the rate of target misses was zero percent.

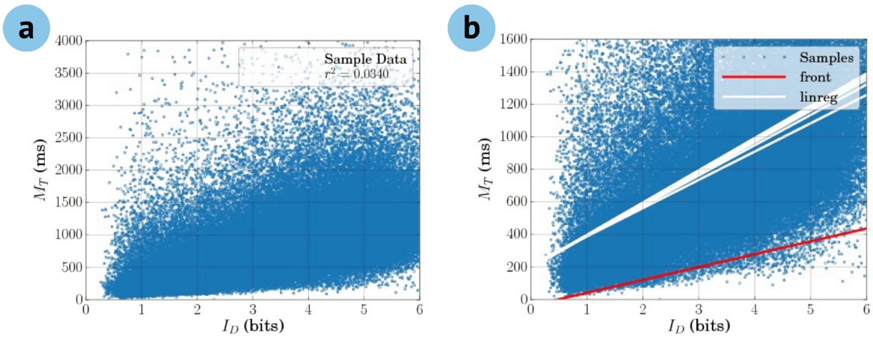


Figure 11: Movement data as a function of ID from participant 3 of Chapuis et al. [31]. (a) MT up to 4 seconds and (b) MT up to 1.6 seconds: linreg (white) shows the linear fits from the usual linear regression and front (red) shows an estimate of the front of performance. Figures are taken from [80].

Fig. 11 illustrates data from one representative participant (P3) of Chapuis et al. [31]. We can see that the data, obtained with no speeding instructions (and no experimenter to recall them), exhibits a huge amount of stochastic variability along both dimensions of the plot. Judging by linear regression on this raw data, they find that movement time and the index of difficulty are essentially uncorrelated since the r-squared coefficient is very close to 0 ($r^2 = 0.0340$). In contrast, Fig. 11 (b) reveals that the bottom edge of the scatter plot is approximately linear: this linear edge is what justifies Fitts' law. In other words, the empirical regularity in Fitts' law is, in essence, a *front of performance*, a lower bound that cannot be passed by human performance [80]. This is in line with the concept of Fitts' 1954 paper, titled *The information capacity of the human motor system in controlling the amplitude of movement*.

Even though the front of performance has a solid theoretical background, the average law, the Fitts' law that most of us are familiar with, claims more than 60 years of empirical validation. Both of them characterize certain aspects of motor performance: the front Fitts' law gives robust and consistent results as it relates to human motor capacity, which provides a reference to HCI researchers; while the average Fitts' law reflects the task environment, e. g. device, user ability, etc., which helps us better understand the elements that are not directly related to the human motor system.

3.3 DOES THROUGHPUT EQUAL CHANNEL CAPACITY?

From the minimum amount of information and the average time per response, Fitts also derived the *Index of Performance* IP, which is computed by dividing ID by the empirically determined movement time MT: $\mathrm{IP} = \frac{\mathrm{ID}}{\mathrm{MT}}$ (Equation 10), representing the subject's maximum rate.

Fitts [63] concluded that the rate of performance IP in a given type of task is approximately constant over a considerable range of movement amplitudes and tolerance limits, expressing the capacity of the human motor system. Measured in bits per second, however, this index of performance is different from throughput, which is claimed to be constant as well.

In engineering, throughput is widely used to measure an effective speed of data transmission, equivalent to the transmission rate R in Shannon's terms [34]. In Fitts' paradigm, we also see two definitions of throughput.

The first one is from Mackenzie [141], and is defined as follows:

$$TP = \frac{ID_e}{MT} = \frac{\log_2(\frac{A_e}{4.133 \times SD_x} + 1)}{MT}. \tag{14}$$

where TP is the throughput, in bits per second, combining speed and accuracy in a single measure, A_e is the distance or amplitude of movements and SD_x is the standard deviation of movement along the x axis. Mackenzie concluded that throughput remains constant and is independent of the speed-accuracy tradeoff.

Mackenzie [141] conducted a controlled experiment with 18 participants varying 3 instructions; a balanced strategy, an emphasis on speed and an emphasis on accuracy respectively, and concluded that "This work provides empirical evidence in support of an important but difficult-to-test tenet of Fitts' law: that throughput is independent of the speed-accuracy tradeoff", supported by Fig. 12.

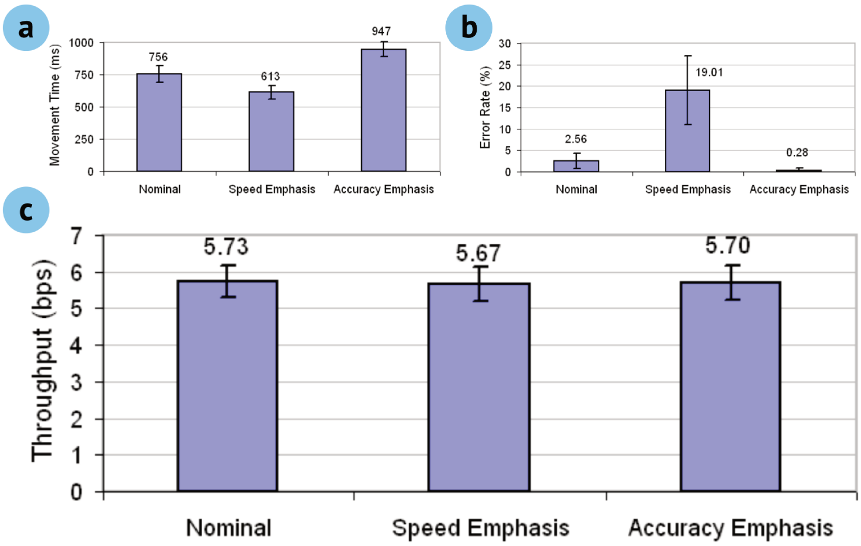


Figure 12: (a) Movement time; (b) Accuracy rate and (c) Throughput accross 3 conditions with a balanced strategy, an emphasis on speed and an emphasis on accuracy respectively. Taken from Mackenzie [141].

Building on Card et al. [26], Zhai [222] came to a similar conclusion that "throughput is conceptually a true constant, describing the index of performance": if we move fast, we ought to make more errors; if we aim for accuracy or precision, we ought to slow down. However, he argued that Mackenzie's TP "is an ill-defined concept that may change its value with the set of ID values used for the same input device and cannot be generalized beyond specific experimental target distances and sizes". Instead, he described the two constants in Fitts' expression $MT = a + b \times ID$ as capturing different aspects: a reflects the non-informational aspect and constant b the informational aspect of input performance. Therefore, Zhai called the inverse of the slope b, $1/b$, throughput. Note that when the intercept $a = 0$, Zhai's throughput equals Mackenzie's.

Guiard and colleagues also studied the notion of speed-accuracy tradeoff of aimed movement [88]. They proposed a mathematical description [89]: The Weighted Homographic (WHo) Model, defined as:

$$(y–y_0)^{1-\alpha}(x–x_0)^{\alpha} = k_{\alpha}. \tag{15}$$

where x and y represent the accuracy dimension and speed dimension respectively, x_0 and y_0 represent the theoretical minima of axioms accuracy and speed, $k > 0$ is an adjustable constant, and weighting exponent α is also an adjustable coefficient ($0 < \alpha < 1$). The role of α is to allow some degree of asymmetry, and it should not approach 0 or 1.

They demonstrated the model using 4 datasets: Fitts' reciprocal tapping data [63], two datasets from Schmidt et al. [182] and Guiard [88] (Fig. 13). They concluded that "human aimed movements are indeed governed by a single speed/accuracy trade-off, and that that trade-off is what it is regardless of the experimental technique with which it is demonstrated".

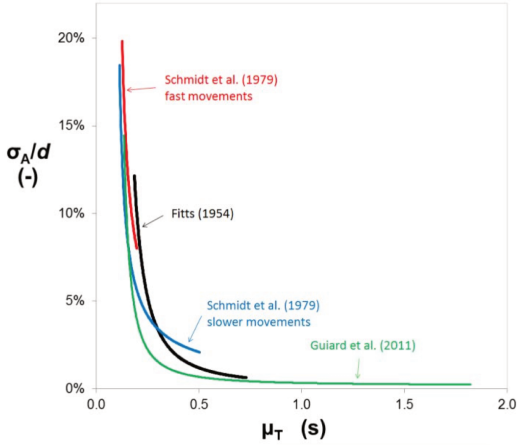


Figure 13: Guiard and Rioul's WHo model describing speed-accuracy tradeoff of aimed movement. μ_T represents the average movement time; σ_A represents the standard deviation of movement amplitude; and d represents the distance to target center. Taken from [89].

It seems that Mackenzie, Zhai and Guiard all agreed that this speed-accuracy tradeoff paradigm, namely throughput, is described and supported by Fitts' law.

- Mackenzie [141]: "We describe an experiment to test the hypothesis that Fitts' throughput is independent of the speed-accuracy tradeoff";
- Zhai [222]: "Throughput (TP), also known as index of performance or bandwidth in Fitts' law tasks, has been a fundamental metric in quantifying input system performance";
- Guiard [89]: "The trade-off is described by the Fitts' law".

However, from previous analysis we know that what Fitts [63, 65] called the index of performance represents the maximum rate and channel capacity (as demonstrated by Gori et al. [80]), which, in engineering terms, is the *maximum* rate at which data could be potentially transmitted. Yet throughput captures the *actually* transmitted information rate, which is bounded by the channel capacity [3]. Even though Fitts' law has firm empirical validation, we invite the HCI community to revisit these three questions: What is the minimum amount of information? Should we take the average of the minimum or the average of the average? What are we measuring – throughput or capacity?

---

[3] Recall the notions of channel capacity C and information transmitted at a rate R in Chapter 1 Section 1.3.

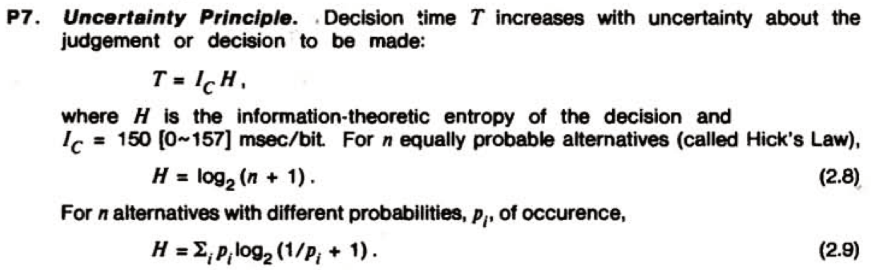
4

PSYCHOLOGY AND HCI – HICK'S LAW

The work presented in this chapter is submitted to ACM CHI 2019: How Relevant is Hick's Law for HCI. Wanyu Liu (WL), Julien Gori (JG), Michel Beaudouin-Lafon (MBL), Olivier Rioul (OR) and Yves Guiard (YG).

Conceptualization: WL, JG
Data reanalysis: WL
First draft: WL
Paper writing: WL, JG, MBL, OR, YG
Supervision: OR, MBL, YG

Together with Fitts' law [63], Hick's law (or the Hick-Hyman law) [98, 104] was also first introduced to HCI by Newell, Card and colleagues [161, 162] as the Uncertainty principle. They described the law as the time a person takes to make a decision as a result of the possible choices she has: increasing the uncertainty/information will increase the decision time. This uncertainty is captured by the information-theoretic notion of entropy and is described by a logarithmic function by Equation 2.8 when n choices are equally probable or Equation 2.9 when the n choices have different probabilities of occurrence p_i (Fig. 14).



P7. ***Uncertainty Principle.*** Decision time T increases with uncertainty about the judgement or decision to be made:

$$T = I_C H,$$

where H is the information-theoretic entropy of the decision and I_C = 150 [0~157] msec/bit. For n equally probable alternatives (called Hick's Law),

$$H = \log_2 (n + 1). \tag{2.8}$$

For n alternatives with different probabilities, p_i, of occurence,

$$H = \Sigma_i p_i \log_2 (1/p_i + 1). \tag{2.9}$$

Figure 14: Card, Moran and Newell's description of Uncertainty Principle. Taken from [161, p.27].

Since its introduction, Hick's law has been used in a number of HCI contexts but its applications have been controversial: some researchers found that the law did not apply to HCI tasks while others regard it as a fundamental law of interface design.

- The law is ignored in many HCI textbooks [180, 194], but it is taught in HCI classes as one of the few quantitative laws in psychology. We interviewed a few HCI professors on their understanding of Hick's law and received replies such as *"it's about response time", "decision making time", "I have two slides about the law, adapted from the slides by someone else", "I teach it but I don't feel 100% comfortable about talking about it"*.
- Conceptually, it is regarded as a fundamental law of interface design. It appears in design books, e. g. [130], and numerous online articles discussing how understanding Hick's law could improve interface design [1].

---

[1] Examples can be seen at `https://measuringu.com/hci-laws/` and `https://uxplanet.org/design-principles-hicks-law-quick-decision-making-3dcc1b1a0632`.

It has been claimed to apply to a large number of contexts, including menu design, device settings and road signs. Essentially, when faced with a set of choices, this "Hick-based" design principle guides interface design with the concept *less is more* (or rather, *fewer is better*).

- In practice, it has not seen many successful applications. Only a few HCI publications incorporate Hick's law, e. g. Soukoreff & Mackenzie [216]. In 2005, Seow [187] compared Hick's law and Fitts' law, the two information-theoretic principles, and examined the possible reasons for the lack of uptake of Hick's law to gain momentum in the field. Nevertheless, few studies have incorporated Hick's law into their work since then, e. g. Cockburn et al. [33].

The controversial aspect of Hick's law and the lack of comprehensive understanding may explain why many HCI researchers have not ventured to apply it to interaction tasks. Furthermore, there seems to be different definitions of Hick's law. While for psychologists the law has exclusively to do with the context of the choice-reaction paradigm, HCI researchers seem to apply it whenever choices are presented to the user, including for visual search time, e. g. [123, 143, 218], decision time, e. g. [33] or reaction time, e. g. [181]. Does the law really apply to these settings?

This chapter strives to clarify some misunderstanding about Hick's law so as to provide a clearer picture of the choice-reaction paradigm in HCI studies. First, I re-examine HCI studies that have used Hick's law and revisit the historical context of the choice-reaction paradigm in psychology. I then demonstrate that a number of logarithmic phenomena observed in HCI do not justify the law; conversely, I show that the choice-reaction paradigm does not always scale logarithmically with the number of choices. I conclude with the practical implications of this new look at Hick's law for HCI.

4.1 HCI APPLICATIONS OF HICK'S LAW

I first review HCI studies that have used Hick's law. I describe their respective contexts and tasks, and highlight the inconsistencies that emerge.

4.1.1 *Modeling Menu Performance*

In a menu selection task, Landauer & Nachbar [123] asked participants to select a target item (a number in the ordered list from 1 to 4096, or a word in a list of 4096 alphabetically ordered words) by a series of touch menu choices among sequentially subdivided ranges.

The number of alternatives at each step was 2, 4, 8, or 16. The authors found that a logarithmic function fits the mean response time (T) well, as implied by Hick's law (reaction time RT) and Fitts' law (movement time MT):

$$T = \underbrace{c_1 + k_1 \log b}_{\mathrm{RT}} + \underbrace{c_2 + k_2 \log \frac{d}{w}}_{\mathrm{MT}} = c + k \log b. \tag{16}$$

where c and k are empirically determined constants, b is the number of alternatives at each step, d is the distance moved and w the width of the target that must be hit. Since the target width w was proportional to $1/b$ in their experiment, the Fitts' law term $\log \frac{d}{w}$ reduces to $\log b$.

Landauer & Nachbar also observed that *"in the most extreme case (words), total selection time varied from 23.4 down to 12.5 seconds for branching factors of 2 to 16"*. They consequently concluded that broader, shallower menu trees yield faster search time than narrower, deeper ones (p. 76).

Cockburn et al. [33] present a predictive model for linear menu performance that uses the "Hick-Hyman law" to model decision time for expert users. They state that the decision time is dependent on the entropy of each item $H_i = \log_2(1/p_i)$. Therefore, decision time for each item is given by $T_{hhi} = b_{hh} \times H_i + a_{hh}$.

The authors conducted a calibration study with 8 participants, varying menu length (2, 4, 8 and 12 items) and block (1 to 7 for two static menu conditions and 1 to 3 for a random condition) and using a uniform distribution of target occurrences. Removing data from block 1 where participants were mostly doing visual search, they found that the decision time could be modeled as $T_{hhi} = 0.08 \log_2 n + 0.24$ with $R^2 = 0.98$. They then used the values of a_{hh} and b_{hh} for the real experiment, which used a Zipfian distribution [2] to account for target frequencies and found that decision time followed Hick's law. It is unclear how they applied the law though, namely whether H_i in the real experiment was computed as $b_{hh} \times \sum \log_2(1/p_i) + a_{hh}$ or as $b_{hh} \times \sum(p_i \times \log_2(1/p_i)) + a_{hh}$. Either way seems problematic as the constants were derived from a uniformly distributed menu.

---

[2] A Zipfian distribution is defined as: $f(k;s,N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)}$ where N is the number of elements, $k \in [1,N]$ the rank of the considered element (with $k = 1$ is the element with highest frequency) and s the value of the exponent characterizing the distribution.

4.1.2 *Modeling Text Entry*

Hick's law has also been used for modeling text entry. Investigating the theoretical upper and lower bounds of typing speed using a stylus on a soft keyboard, Soukoreff & Mackenzie [216] argue that Hick's law can be used to account for the visual scan time (RT) of each entry, in order to compute the lower bound of the typing rate:

$$RT = a + b \log n. \tag{17}$$

where n is the number of choices and a and b are empirically determined constants.

Borrowing from Welford [214], they set $a = 0$ and $b = 1/5 = 0.2$ as Welford states that for subjects in their twenties using key presses to signal choices, the reciprocal of the slope of Hick's law lies is between 5 and 7 bps. Therefore, with a 27-character alphabet (26 letters plus space), $n = 27$, the lower bound of the visual scan time for novices is:

$$RT = 0.2 \log_2 27 = 0.951 \text{ seconds.} \tag{18}$$

This model was rejected by empirical investigations by Mackenzie et al. [142, 143] who observed about twice the time expected of 0.951 seconds for visual scanning. Mackenzie et al. concluded that *"Although the Hick-Hyman metric may still be valid in general, clearly as applied here it is confounded with the complex movement behavior we observed"*.

Sears et al. [185] also illustrate that it is inappropriate to use Hick's law for a simple visual search component task, such as the one introduced by Soukoreff & Mackenzie [216]. Sears et al. argue that using Hick's law implies that only the number of keys is important when determining which key to press. In contrast, they provide evidence that both the keyboard layout, e. g. QWERTY or Dvorak, and the number of letters represented by each key, e. g.three letters per key on a telephone keypad, must be considered.

Wobbrock & Myers [218] introduce a stroke-based word completion technique for trackball text entry and include Equation 18 in their model. This term is added after the entry of every letter and represents the time it takes for a user to find their word among n choices, where n is the number of completions offered for the current prefix ($0 \leqslant n \leqslant 4$). The authors show that this new stroke-based word prediction and completion technique outperforms a major commercial on-screen keyboard. They do not, however, demonstrate nor analyze if and how reaction time plays a role.

4.1.3 *Interface Design Guideline*

In the design community, Hick's law is interpreted as a general design guideline, which we refer to as the "Hick-based design principle" in this chapter. In the book *Universal Principles of Design* [130], Lidwell et al. state that *"Designers can improve the efficiency of design by understanding the implications of Hick's law"* (p. 120). Similarly, in a Web entry titled "Hick's law: Making the choice easier for users" [3], Soegaard writes that *"Understanding Hick's law means you can design so that more users will visit and stay on your website"*.

Wang [208] states that *"Essentially, Hick's law provides a general guideline for the design and use of hierarchical menu structures. This is consistent with the study [123] showing that users do not consider each choice one by one. What they normally do is to subdivide the choices into categories, and choices in each category are further divided. The resulted structure will be a tree, which can help users to make a quicker decision."*

Ali & Liem [3] claim that *"Within the context of design, Hick's law promotes the use of design methods to simplify decision-making in situations where designers are presented with multiple options. In practice, it has fundamentally proven to be effective in the design of menus, control display, way finding layout."*

Hick's law is also invoked in guidelines for designing applications for mobile devices [159], visualizations [92] and spreadsheets [30]. It seems that Hick's law is a magical formula in the design community and is widely used to rationalize two principles: (a) Minimize the number of choices; and (b) Categorize choices, instead of overwhelm users with all the choices at once.

4.1.4 *Summary: Inconsistencies*

From the literature above, we can see several inconsistencies in the use of Hick's law.

First, we notice that the formulation used in HCI studies is different from the one introduced by Card et al. [161], especially the question of n vs. $n+1$. Nearly all authors use $\log n$, with the notable exception of Cockburn et al. [33], although in that case it is unclear how the stimulus information was computed. This raises a first question: **Which formula for Hick's Law?**

---

[3] https://www.interaction-design.org/literature/article/hick-s-law-making-the-choice-easier-for-users

Second, Hick's law is used to model expert users' decision time by Cockburn et al. [33] and novice users' visual search time by Soukoreff & Mackenzie [216]. Even though Sears et al. [185] showed the incompatibility of the law with visual search, Wobbrock & Myers [218] use it to model visual search time. In the design community, on the other hand, the law seems to work universally. This raises a second question: **When does the law apply?**

Third, Landauer & Nachbar [123] conclude, based on their empirical data, that broader, shallower menu trees yield faster search than narrower, deeper ones. This contradicts the common belief in the design community that a tree structure helps users make a quicker decision [208]. This raises the third question: **What does the law really say?**

To answer the questions above, in the next section I review the choice-reaction paradigm in psychology.

4.2 CHOICE-REACTION TIME IN PSYCHOLOGY

While HCI researchers associate reaction time with Hick's law, there is a long tradition in psychology in studying choice-reaction paradigm. In this section, I review these studies in psychology and attempt to give a more precise definition of Hick's law.

4.2.1 *Before Information Theory*

Several studies have been conducted before Hick's experiment, which are briefly mentioned in Chapter 2 Section 2.1. The first results on reaction time (RT) are due to Helmholtz [95], the famous physician & physicist of the nineteenth century. He determined that signals travel the nervous system at about 60 m/s. A comparison between typical reaction times observed in common tasks and the calculated propagation times revealed that the propagation time for signals could not account for the reaction time, implying that humans were not simply hard-wired to respond to certain stimuli but that time was required *"in the brain for the processes of perceiving and willing"*. By the end of the nineteenth century and early twentieth century, three other important results were known:

- Donders [42] introduced the three-class taxonomy of reaction time that is still in use today: simple RT (*a*-reaction time) is the time it takes to react, with a predetermined response, to the onset of a stimulus whose identity is known in advance but whose time of occurrence is uncertain; choice RT (*b*-reaction time) is the time it takes to react to the onset of one of several possible stimuli, following a given stimulus-response mapping rule;

and go-no-go RT (*c*-reaction time) is the time it takes to respond to a stimulus that may or may not occur at a predetermined point in time. Donders showed that simple RT was the shortest and choice RT the longest.

- Merkel [151] performed an experiment using an identification-choice reaction task, i. e. measured *b*-reaction time. It showed that it takes longer to respond to a stimulus when it belongs to a larger set of stimuli.

- Many psychologists have attributed the "reaction-time era" to such causes as the division of attention or a reduction in the effective intensity of the stimulus without providing a quantitative theory. Blank [20] was the first to postulate a logarithmic relationship between reaction time and the number of alternatives but did not give further explanation.

4.2.2 *The Information Analogy: Hick & Hyman*

Hick's work was strongly inspired by Merkel's results. In fact one could say that Hick did nothing but replicate Merkel's experiment and used a logarithmic scale for the x-axis rather than Merkel's linear scale. This misses an important point: Hick's contribution is conceptual rather than experimental. Using the information-theoretical rationale that was popular at the time, Hick interpreted this logarithmic curve by considering the human as a channel of information transmission [98]. Accordingly, reaction time is seen as resulting from the uncertainty of the stimulus, which can only be processed at some maximum rate. The information rationale had an immediate effect: If the "information" – in layman's sense – mattered, then all the ways in which information could be varied mattered. This introduced new ways of testing the relationship between reaction time and the "information" provided by the stimulus.

Hyman [104] varied the entropy of the stimuli in 3 ways:

- By changing the total number n of stimuli (Hick [98], Merkel [151]). In this case, uncertainty increases with the number of stimuli.

- By changing the probability that each stimulus is indeed activated. The more similar the probabilities, the higher the uncertainty. In the limit case, when all the stimuli are activated with equal probability, uncertainty reaches its maximum value of $\log n$.

- By establishing "grammar rules", i. e. introducing conditional probabilities between successive stimuli. For example, if stimuli B is activated, then it is certain that stimuli D will be activated next.

Their respective experiments are detailed in Chapter 2 Section 2.1.

4.2.3 *Choice Reaction Time: Results*

After Hick's and Hyman's respective experiments, a number of studies measuring reaction time in a choice-reaction task were reported. Here I summarize the main results.

4.2.3.1 *Reaction Time: Context*

Most choice reaction time studies were conducted with the goal of measuring *b*-reaction time on very simple tasks, such as responding to bulbs lighting up by pressing keys (Hick [98]) or through speech (Hyman [104]). It is not clear how well the logarithmic relationship between time and information holds for more complex tasks that potentially require a lot of decision making. One exception is given by Crossman [37], who investigated card sorting and found results consistent with the rest of the literature.

4.2.3.2 *Measuring Information: Entropy or Mutual Information?*

It has consistently been found that for a range between 1 and 3 to 4 bits, reaction time increases linearly with entropy, irrespective of which of the three variables described above is being manipulated. However, whenever the number of choices becomes larger, it seems that reaction time is consistently over-estimated. In fact, Fitts & Posner [67] indicated that whatever the number of possible stimuli, reaction time will seldom exceed 1s. Seibel [186] reported that there is almost no difference in reaction time between responses to 31 (5 bits) or 1023 (10 bits) stimuli. Fitts et al. [66] showed that the response to very low probability alternatives is faster than predicted by the law. On the other hand, Pollack [174] found that the linear relationship extends to about 10 bits in a task where words had to be named. The actual range where the relationship holds is thus very dependent on the actual task.

Modulating the speed-accuracy characteristic to modify the values of mutual information leads to gross underestimates of reaction time, especially for very small values of mutual information I, i. e. in conditions where many mistakes are made. Fitts [64] reported that beyond 0.6 bits of equivocation ($H(X|Y)$, see Equation 2), the loss of information resulting from errors increases faster than the gain from increased response speed. We will therefore now consistently use the term *stimulus uncertainty* rather than the vague term of "information" to characterize the stimuli.

4.2.3.3 *N or N+1?*

Hick [98] found that if the number of possible signals is n and reaction time is plotted against $\log(n+1)$, the observed reaction times for different numbers of signals lie on a straight line which also passes through the origin.

The reason why the fit is better for $(n+1)$ instead of n is that if the subject is uncertain about when a signal will appear, then when it does appear, he or she needs to not only decide which it is, but also decide that a signal has occurred at all. In fact, the $+1$ in Hick's formulation has not always been easy to understand and several other alternative equations have often been preferred.

Immediately after Hick, Crossman [35] conducted a card-sorting experiment: The subjects held a well-shuffled pack, turned up the cards one by one and sorted them into various classes. The results were plotted against $\log n$ as the pack was always available, hence there was no uncertainty about when a fresh signal would appear. Hyman [104] extended Hick's concept by manipulating uncertainty with unequal probabilities: $-\sum_{i=1}^{n} p_i \log_2 p_i$. Even when he replicated Hick's experiment with equal probability, he proposed $\log n$, not $\log(n+1)$. While Suci et al. [200] fitted the data equally well with n and $n+1$, other researchers such as Griew [84] and Brown [23] found that $n+1$ fitted data slightly better. Crossman [37] also plotted data against $\log(n+0.45)$.

As Welford [214] pointed out, the proposed mathematical formulations provide merely a summary statement of a complex process of observation, identification, choice and reaction which highly depends on the specific task environment. Hence one possible formula to account for this complexity is $\log(n+n_0)$ where n_0 describes the effect of temporal uncertainty expressed in terms of n. n_0 varies from zero if the subject is able to estimate exactly when the next signal will appear, to more than 1 if she does not have any idea of when the stimulus will show up. When the time at which the stimulus appears is reasonable but not completely predictable, $n_0 \in [0,1]$.

4.2.3.4 *Effect of Stimulus-Response Compatibility*

S-R (Stimulus-Response) compatibility was introduced in psychology to characterize the fact that it is easier to respond to a stimulus using certain responses than others. If the stimulus is coded in terms of digits appearing on a screen, it is for example much easier for someone to call the corresponding digit out than, say, to call a letter or another digit out. It has repeatedly been found that the better the SR compatibility, the shallower the slope relating reaction time to stimulus uncertainty.

In fact, whereas Hick's light and key experiment reports rates of about 200 ms/bit, there are many cases where the slope can approach 0 ms/bit (Fig. 15), such as Leonard's experiment [126] where the subject rested his fingers upon vibrators and touched the vibrator that was activated (Fig. 15 Curve J), or Mowbray's experiment [157] where subjects gave a voice reaction to Arabic numerals (Fig. 15 Curve G).

In those cases, corresponding to extremely good S-R compatibilities, there is virtually no effect of stimulus uncertainty on reaction time.

Stimulus-Response (S-R) compatibility was introduced in psychology to characterize the fact that it is easier to respond to a stimulus using certain responses than others. If the stimulus is coded in terms of digits appearing on a screen, it is for example much easier for someone to call out the corresponding digit than, say, to call a letter or another digit out. It has repeatedly been found that the better the S-R compatibility, the shallower the slope relating reaction time to stimulus uncertainty. In fact, whereas Hick's light and key experiment reports rates of about 200 ms/bit, there are many cases where the slope approaches 0 ms/bit (Figure 32 in Fitts & Posner [67, p.105]), such as Leonard's experiment [126] in which the subject rested his fingers upon vibrators and touched the vibrator which was activated, or Mowbray's experiment [157] where subjects gave a voice reaction to arabic numerals. In those cases, corresponding to extremely good S-R compatibility, there is virtually no effect of stimulus uncertainty on reaction time.

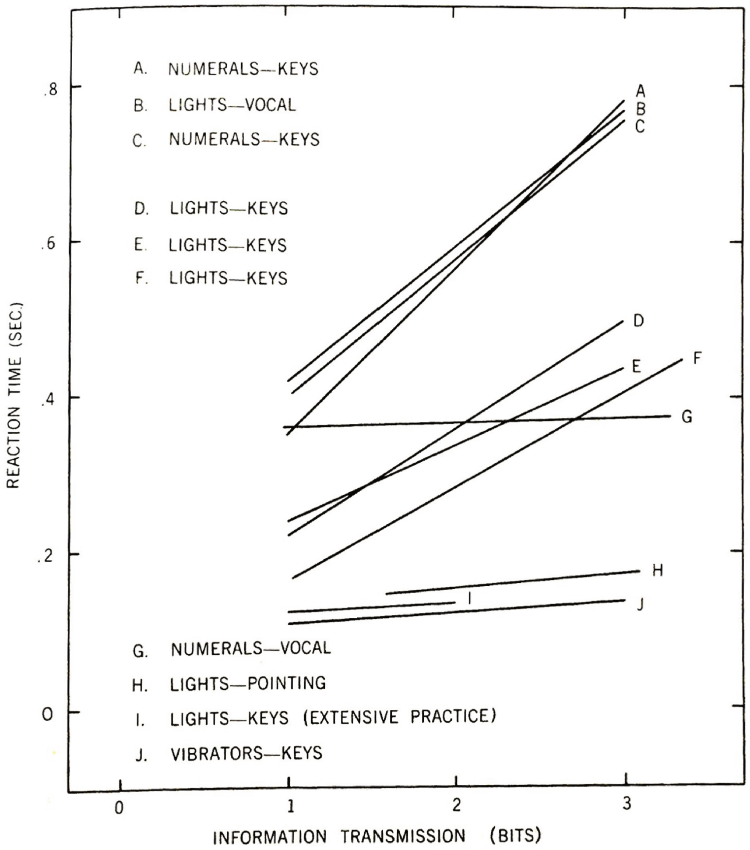


Figure 15: Reaction time as a function of stimulus "information" in different tasks. Constant reaction time: J [126], G [157], I [158]; Larger slope: F [98], B [22]. Taken from Fitts & Posner [67, p.105].

Indeed, as Fitts & Posner pointed out [67], anything that decreases the spatial or energy correspondence between input and output, therefore reducing compatibility, increases the slope. This principle can probably be beneficial to interface design.

4.2.3.5 *Effect of Learning*

The effect of learning is very similar to that of S-R compatibility. When participants are heavily trained, the effects of the uncertainty of the stimulus and even of the S-R compatibility can be reduced so that reaction time is almost constant, regardless of the number of items. Mowbray's experiment [158] showed that reaction time for choices among up to 10 possibilities could be reduced to that of a two choices alternative when a subject practiced a key-press task for a period of 6 months. Although this is somewhat questioned by Welford [214], it is clear that practice will significantly reduce the slope: Knight & Dagnall [114] reported slopes dropping from 73 ms/bit to 23 ms/bit after two months of practice.

4.2.4 *Clarifying Hick's law*

It is clear at this point that Hick's law is much more complicated than Card et al.'s description [161] would suggest. Indeed, learning, S-R compatibility and stimulus uncertainty all affect reaction time. Furthermore, these effects are dependent on each other. For example, if one wishes to modify uncertainty by changing the probabilities of activation of each stimulus, then the subject has to go through an extensive learning phase, as discussed by Hyman [104]. Yet, she will inevitably improve her skill in the matter of the experiment, leading to a reduction of the slope.

Similarly, good S-R compatibility is usually desirable, otherwise the experiment will appear poorly designed. However, this makes the effect of Hick's law much harder to grasp, as the influence of stimulus uncertainty is then highly reduced.

Finally, learning does not affect all experiments in the same way. The highest rates are usually found with experiments using words. This is not necessarily because the S-R compatibility is particularly good, but rather because reading and remembering words is a highly over-learned task, which we train daily. What can we say about Hick's light-key association task? Is this a completely new task, or are we somewhat familiar with it?

We therefore propose the following, a clarification of Hick's law:

The choice reaction time for users performing a simple task grows linearly with the stimulus uncertainty, measured by entropy, in the range of 1 to 4 bits. The better the S-R compatibility and the better the training, the shallower the slope. With appropriate learning, the effects of S-R compatibility and stimulus uncertainty can be reduced to almost zero.

4.3 THE CHOICE-REACTION PARADIGM AND HCI

In this section, I revisit the HCI applications of Section 4.1, comparing them with the choice-reaction paradigm in psychology and outlining the discrepancies in the use of the law. Seow [187] offered three plausible reasons for the failure of Hick's law in HCI: (1) The complexity of computing information measures; (2) The complexity of stimuli including font sizes, colors, etc. and (3) The unpredictability of stimuli changes over time with practice. Therefore Hick's law appears to be optimal only in predicting novice performance. Here we offer two additional explanations for why it is not trivial to use Hick's law in HCI studies.

4.3.1 *Decomposition of Time Measures*

Table 1 summarizes and demonstrates the differences between how Hick and Hyman introduced the paradigm and how the law has been used in HCI studies. All the studies assume a stimulus-response (S-R) paradigm. From a measurement perspective we typically face three time marks: Stimulus-onset time T1, Response-onset time T2, and Response-termination time T3, allowing the calculation of three relevant time durations: movement time MT = T3 - T2, task completion time TCT = T3 - T1 and a third time, say xT = T2 - T1 (Fig. 16 (a)). xT can stand for RT as in "reaction time" (e. g. Hick [98], Hyman [104]), or in more complex tasks, can include, e. g. visual search [123, 216], decision [33]. Importantly, xT and movement time MT are, by definition, non-overlapping intervals.

From the previous section we know that Hick along with other psychologists measured the choice reaction time (b-RT) in response to a particular stimulus: the time it takes to press the key, hence, reaction time dominates task completion time (Fig. 16 (b)). In contrast, all the studies in Table 1 fall into the paradigm of Fig. 16 (c) where movement time MT contributes a relatively large portion of task completion time TCT. Despite the fact that xT involves different mental processes in these studies (e. g. S identification, visual search, memory search, decision to respond), the authors of these studies attributed these phenomena to Hick's law.

Some of these phenomena can indeed be explained by a logarithmic function, but attributing them to Hick's law is a very unfortunate conflation between the formula and the law. For example, visual search in a hierarchical structure is logarithmic, as shown by Landauer and Nachbar [123]. In fact, anything that involves a *divide and conquer* strategy can be logarithmic. In many cases, this has nothing to do with Hick's law. In this case, a good practice can be seen from Cockburn and Gutwin [32], who state that scrolling an alphabetically organized menu can be predicted by a logarithmic function, but do not relate it to Hick's law. Here xT is mostly dominated by visual search and can indeed be modeled by a logarithmic function. The time to make a decision also has nothing to do with Hick's law, despite its frequent use as an argument in the design community. Section 4.4 gives a mathematically sound proof showing that a logarithmic function of task difficulty contradicts the commonly believed Hick-based design principle.

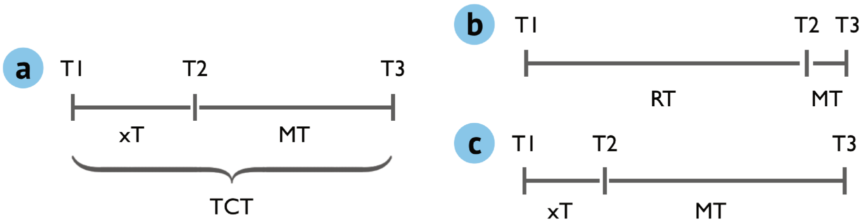


Figure 16: (a) Three time marks in a stimulus-response (S-R) paradigm: S-onset time T1, R-onset time T2, and R-termination time T3. Task completion time (TCT) = T3 - T1, movement time MT = T3 - T2 and xT = T2 - T1, which describes the psychological processes (reaction, visual search, decision, etc.) before movement. (b) xT = RT choice reaction time in Hick's experiment [98], while (c) xT involves different mental tasks in the studies listed in Table 1.

4.3.2 *The Differences between Novices and Experts*

Another important issue is whether the law applies to novice users, e. g. [143], expert users, e. g. [33], or both, e. g. [218]. Rather than stating that Hick's law appears to be optimal in predicting novice performance [187], we believe that, once again, it depends on the task, S-R compatibility and practice.

In his experiments, Hick trained the participants with more than 8000 practice trials whereas in Hyman's experiments, more then 15,000 trials were registered, yet there was still a slope in choice reaction time. In a way, the participants in Hick's and Hyman's experiments were trained enough to recognize the mapping between the lamp and the key (Hick) and between the lamp and the word (Hyman), but not trained enough to completely wear out the reaction time.

	Hick [98]	*Hyman [104]*	*Landauer & Nachbar [123]*	*Cockburn et al. [33]*	*Soukoreff & Mackenzie [216]*	*Mackenzie et al. [142, 143]*	*Wobbrock & Myers [218]*
Task	Reaction	Reaction	VS*	Decision	VS	VS	VS
Stimuli	Random	Random	Ordered	Random	Keyboard	Keyboard	Random
Participants	Well-trained	Well-trained	All users	Users starting from block 2	Novice users	Novice users	All users
Distribution	Uniform	Non-uniform	Uniform	Zipfian	Uniform	Uniform	Uniform
Information	Up to 3.32 bits	Up to 2.81 bits	Up to 4 bits	Up to 3.58 bits	4.75 bits	4.75 bits	Up to 2 bits
Measure	Mutual Information	Entropy	Entropy	Entropy	Entropy	Entropy	Entropy
Formula	$\log(n+1)$ or $\log(n_e+1)$	$\log n$ or $-\sum_{i=1}^{n} p_i \log_2 p_i$	$\log n$	$\log n$	$\log n$	$\log n$	$\log n$
Results	Logarithmic	Logarithmic	Logarithmic	Logarithmic	Logarithmic	Not Logarithmic	Logarithmic

Table 1: A comparison of Hick 1952, Hyman 1953 and HCI studies that used Hick's law (*VS**: Visual Search).

Indeed, as shown in the previous section, later studies confirmed that an extremely well-trained participant can react in almost constant time despite stimulus uncertainty [158]. There are also tasks that we are naturally experts at, such as resting fingers on vibrators and pressing the corresponding key when it vibrates [126], or giving a voice reaction to Arabic numerals [157].

In HCI studies, it is difficult to judge novice vs. expert users in the stimulus-response context. We are daily computer users, and we are all semi-experts in responding to a visual stimulus with a mouse pointer. Unless the interaction technique uses a device that participants have never seen before or a mapping that is completely novel, we should rethink what is their expertise in this choice-reaction paradigm.

4.3.3 *Effect Size of Hick's Law*

Then what is the effect size of choice reaction time in HCI studies? As stated in Section 4.2, the slope in Hick's law depends on learning and S-R compatibility. Two questions then naturally emerge:

- How familiar are the participants in HCI experiments with the tasks we have them perform?
- What is the S-R compatibility of the tasks we usually ask participants to perform?

Arguably, a successful interaction should be easily learned by participants, or, even better, exploit already over-learned tasks, and should have a good S-R compatibility. Therefore, it can be expected that for a successful technique, the slope of Hick's law is already quite shallow.

To illustrate this point, we reanalyzed data from Roy et al. [181] and Liu et al. [134] where a simple command selection task was carried out. In Roy et al. [181], participants needed to select a highlighted command by touching the screen of a tablet with a predetermined finger in the *Glass* condition vs. one of several fingers in the *Glass+Skin* condition. In Liu et al. [134], participants were instructed to move the mouse pointer to hit a highlighted stimulus among several commands. The finger or the pointer was rested on a designated area at the beginning of each trial. In both cases, reaction time was measured between the start of the trial and the first move of the finger or pointer. Fig. 17 shows the plot of reaction time as a function of stimulus uncertainty.

We ran a repeated-measures full factorial ANOVA for 3 conditions on these reaction times (Table 5). In conditions *Glass+Skin* [181] and command selection [134], the effect of stimulus uncertainty on reaction time is not significant.

In the *Glass* condition, even though $p < 0.05$, the effect size is very small. In fact, the effect size for participant is 0.49, compared to 0.10 for stimulus uncertainty. This means that having a different participant incurs a higher variance than switching the stimulus uncertainty condition. The slope of Hick's law is very small: 32ms/bit in the *Glass* condition, 8ms/bit in the *Glass+Skin* condition and 4ms/bit for command selection [134]. This means that the reaction time can be treated as a constant.

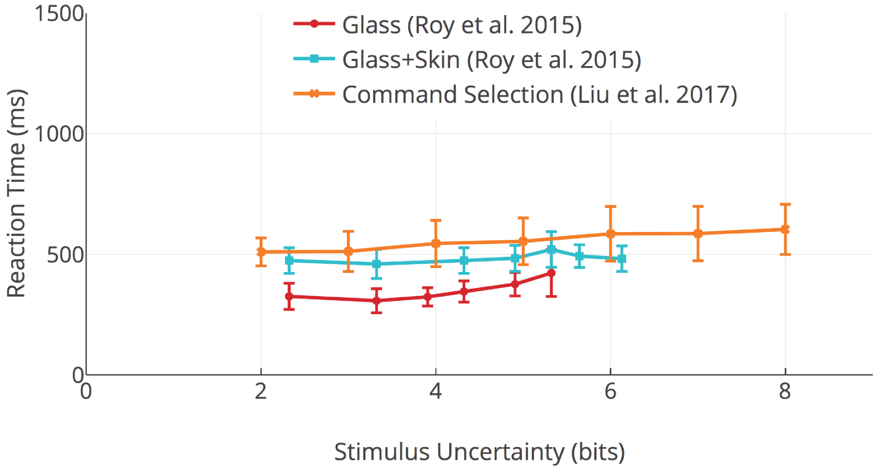


Figure 17: Reanalysis of data from Roy et al. [181] and Liu et al. [134]: reaction time as a function of stimulus uncertainty.

Factors	*df, den*	*F*	*p*	*Effect Size*
Glass	5, 65	3.39	0.0087	0.10
Glass+Skin	6, 78	4.53	0.053	0.02
CS	6, 66	7.27	0.816	0.06

Table 2: Full-factorial ANOVA for 3 conditions on their respective reaction times: Glass and Glass+Skin from [181] and CS (Command Selection) from [134].

Indeed, for GUI tasks using the mouse or fingers and a task similar to Hick's light-key pressing paradigm, stimulus uncertainty shows little effect on reaction time: Inter-subject variability induces more variation in reaction time than stimulus uncertainty. In fact, given the relative magnitude of movement time, Hick's choice reaction time can be considered constant.

4.4 IMPLICATIONS FOR HCI

In this section, based on a mathematical analysis, I first show that the Hick-based design principle praised in the design community is unduly justified by Hick's law. Then, I build on the previous discussions and suggest that, in spite of the complexity of the psychological process behind xT (Fig. 16), we can advocate a simple design strategy, on the premise that xT is either convex or concave.

4.4.1 *The Concavity of the Logarithm Contradicts the Hick-based Design Principle*

Hick's law is usually used as an argument by the design community to justify the need to display as few items as possible (Section 4.2). Consider a scenario where a designer has to display N items, to see how the Hick-based design principle holds.

CAR WEBSITE SCENARIO We consider as a practical example a car website which has $N = 512$ cars to display. We list three display strategies and evaluate Hick's law in each case:

- Display all the cars on the same page. Hick's law states that reaction time RT is given by

  $$RT = a + b \log_2(512) = a + 9 \times b. \quad (19)$$

- Split the 512 cars into 4 pages of 128 uncategorized cars. We apply Hick's law to each page and sum the reaction times. If we consider that the user will go through all the pages, total reaction RT time is given by

  $$RT = 4 \times (a + b \log_2(128)) = 4a + 28 \times b. \quad (20)$$

  Over time, there is one chance out of four that the item she is looking for is in each page, so that the average RT is given by

  $$RT = 1/4\Big(\sum_{i=1}^{4} a\ i + b\ i\ \log_2(128)\Big) = 10/4\ a + 70/4\ b. \quad (21)$$

- Split the 512 cars in 4 categories of 128 cars. The participant selects one item among 4 to select a category, and then selects one item among 128. This is the so called tree strategy, or divide and conquer strategy. The total reaction time RT is given by

  $$RT = a + b \log_2(4) + a + b \log_2(128) = 2a + 9 \times b. \quad (22)$$

For this example, the optimal strategy according to Hick's law consists of displaying all the cars at once on the same page.

In fact the following general result holds. When there are N items to be displayed that can be separated into k subgroups, applying Hick's law leads to the following:

- It is never advantageous to split elements into uncategorized subgroups of equal sizes. Indeed, for $k \in \mathbb{N}; \quad 2 \leqslant k \leqslant N$, we have that:

  $$\left(N/k\right)^{k} \geqslant N; \qquad \text{if } N \geqslant k^{k/(k-1)}. \tag{23}$$

  Then, taking the logarithms on both sides, we get:

  $$k \log(N/k) \geqslant \log N, \tag{24}$$

  which generalizes[4] Equation (20).

- It is not advantageous, or at best useless, to split elements into categorized subgroups of equal size. Indeed, since $N = k \times \frac{N}{k}$, $\log N = \log k + \log N/k$, so that

  $$\begin{aligned} RT &= a + b\log(k) + a + b\log(N/k) && (25) \\ &= a + (a + b\log N). && (26) \end{aligned}$$

  This generalizes Equation (22). Assuming there are m nested categories, we must ultimately pay the price of an additional $(m-1)a$ seconds on RT if we use a tree strategy.

The conclusion of this small example is that, contrary to common sense, Hick's law actually suggests displaying as many items as possible on the car rental site. We do not claim that the Hick-based design principle is flawed, but rather that it cannot be justified by Hick's law. There are many different phenomena taking place when a user is investigating the car rental web page, which are far more complex than simple stimulus response and therefore Hick's model is far too simple.

In serial visual search, selection time increases linearly with size [214], i. e. $RT = a + b\ N$. If one were to choose a joint pair among a set of size N, we would expect a quadratic increase in RT as there are $N(N-1)/2$ possible pairs. This suggests a variety of non logarithmic models.

---

[4] The right-hand side condition of Equation (23) implies that N must always be greater than the number of categories. When we increase k, N should increase a little faster. Practical cases in HCI usually verify that condition, e. g. in the example above with $k = 4$, N should be greater or equal to 7.

4.4.2 *xT Design Principle: A Matter of Convexity*

What matters for design is whether the function relating xT and N is convex or concave. As in the previous example, we consider two different situations: one in which items can be categorized, the other in which they cannot. This leads to two different convexity results. Let f be the function that relates xT to N

$$xT = f(N), \tag{27}$$

CASE 1: ITEMS CANNOT BE CATEGORIZED If the items are displayed in two sets of sizes x and y ($x+y=N$), then $xT = f(x)+f(y)$, whereas if the items are displayed all at once, $xT = f(x+y)$. Therefore, determining whether or not we should split the items boils down to whether $f(x+y)$ is greater or lower than $f(x)+f(y)$

$$f(x+y) \geqslant f(x)+f(y), \tag{28}$$

If Equation (28) holds, then f is said to be superadditive; else f is subadditive. It can be shown [25] that if $f(0)=0$:

$$\begin{aligned} f \text{ convex } &\text{ implies } f \text{ superadditive;} & (29)\\ f \text{ concave } &\text{ implies } f \text{ subadditive.} & (30) \end{aligned}$$

The important information is thus whether f is a convex or concave function. This leads to our first xT design principle:

> If items are not categorizable and f is convex, e. g. quadratic, then it is useful to group the items, even arbitrarily. If f is concave, e. g. logarithmic , then it is better to display all items at once[5].

CASE 2: ITEMS CAN BE CATEGORIZED For this case, we assume for simplicity that the items are categorized in x categories with y items each. Therefore $N = xy$. From the example of the previous subsection, it is clear that the relevant question is whether $f(x)+f(y)$ is greater or not than $f(xy)$. It is easily shown through the previous result that

$$\begin{aligned} f(\exp(.)) \text{ convex } &\text{ implies } f(x)+f(y) \leqslant f(xy); & (31)\\ f(\exp(.)) \text{ concave } &\text{ implies } f(x)+f(y) \geqslant f(xy). & (32) \end{aligned}$$

In Hick's paradigm reaction time grows logarithmically, in which case $f(\exp(x))$ is linear; this is the limit case where $f(x)+f(y) = f(xy)$, i. e. the two strategies are equivalent. As all functions relating selection time to the number of items realistically grow faster than the logarithm (any practical HCI setup is likely more time consuming than Hick's task), we can state the second xT design principle as follows:

---

[5] Note that we use the convention that a choice from a null set takes 0 seconds, which explains the extra conditions needed in (23) as $\log(0) \to -\infty$. Also note that using the formulation of (7) conveniently solves this issue by adding 1 to N.

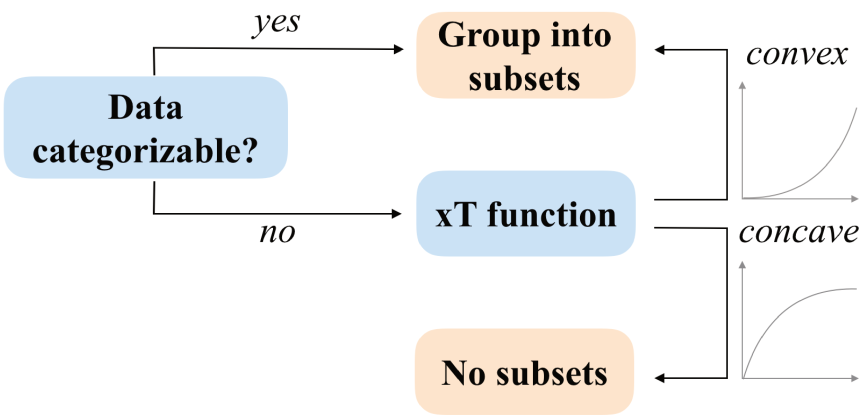


Figure 18: xT design guideline based on data categorization and xT function.

> If items can be categorized, then they should always be split.

Fig. 18 summarizes the two xT design principles.

4.5 CONCLUSION

I have shown that Hick's law is not always relevant to the tasks being studied. Psychologists have successfully investigated the limits and conditions of application of the law, but this knowledge is rarely applied in HCI. To summarize:

- Hick's law may not be relevant to a given context. In fact, we doubt that Hick's law has much to offer in many HCI tasks. In addition, not all reaction times are expected to scale logarithmically with n.
- Conversely, an empirical logarithmic reaction time does not justify Hick's law. Many mechanisms can lead to a logarithmic time. Visual search on a hierarchical menu is a good example [123].

Probably much more can be learned from reaction time than the binary observation of whether or not Hick's law holds. Psychologists have found that reaction time highly depends on the task, S-R compatibility, and practice. These notions are directly applicable to HCI. One of the designer's goals is certainly to maximize S-R compatibility, e. g. by making the interface more intuitive, which also makes it easier to learn. This suggests that in simple GUI tasks, reaction time can be treated as constant; this is indeed supported by the empirical analysis in the previous section.

Looking forward, creating an empirical taxonomy for reaction times analogous to the more theoretical computational complexity might be useful for HCI researchers who strive to model human behavior. Since sub-additivity is the main issue, an index that quantifies this property would probably be useful. $S = \frac{f(x)+f(y)}{f(x+y)}$ is a natural candidate. More empirical and theoretical work should determine whether S is indeed useful.

5

OTHER INFORMATION THEORY ATTEMPTS IN HCI

Apart from Fitts' law and Hick's law, information theory has found few other uses in HCI. This chapter examines several attempts: statistical language modeling for text entry, human information capacity and some applications inspired by control theory [107]. My goal is to explore how information theory has been used in these domains and what can be done for future work.

5.1 STATISTICAL LANGUAGE PROCESSING FOR TEXT ENTRY

Statistical language processing models text entry as communication over a noisy channel and calculates the bitrate of a text entry method. It is based on Shannon's estimation of entropy [189] and exploits the inherent redundancies for language modeling and prediction. Intelligent text entry systems that use this approach result in a lower error rate and potentially a higher entry rate [116].

The basic idea is in line with fundamental information-theoretic concepts. Assume a source alphabet Ω follows a probability distribution. The entropy of such alphabet is then $H(\Omega)$ (Equation 1) and the perplexity. $PP(\Omega) = 2^{H(\Omega)}$, measures how well a probability model is at prediction [1]. The lower the perplexity, the easier the prediction. If the random variable $\mathbb{I}$ is a distribution over the set of words the user is intending to write and the random variable $\mathbb{O}$ is a distribution over the set of words the user is actually writing, then the rate R (in bits per time unit) is:

$$R = \frac{I(\mathbb{I};\mathbb{O})}{t}. \tag{33}$$

where $I(\mathbb{I};\mathbb{O})$ is the mutual information (Equation 2) and and t is the average time it takes to write a word in $\mathbb{O}$. If the probability of error is zero, that is, all words in $\mathbb{I}$ can always be inferred from $\mathbb{O}$, then $R = \frac{H(\mathbb{I})}{t}$ (Equation 6).

---

[1] The perplexity is the exponentiation of the entropy, which is a more clearcut quantity. The lowest perplexity that has been published on the Brown Corpus (1 million words of American English of varying topics and genres) as of 1992 is about $2^{7.95} = 247$ per word, corresponding to a cross-entropy of $\log_2 247 = 7.95$ bits per word or 1.75 bits per letter [24] using a trigram model. Recall that Shannon [188] estimated the word-entropy of printed English as 11.82 bits per word and Grignetti [85] estimated 9.83 bits per word (Chapter 2 Section 2.1). It is often possible to achieve lower perplexity on more specialized corpora, as they are more predictable.

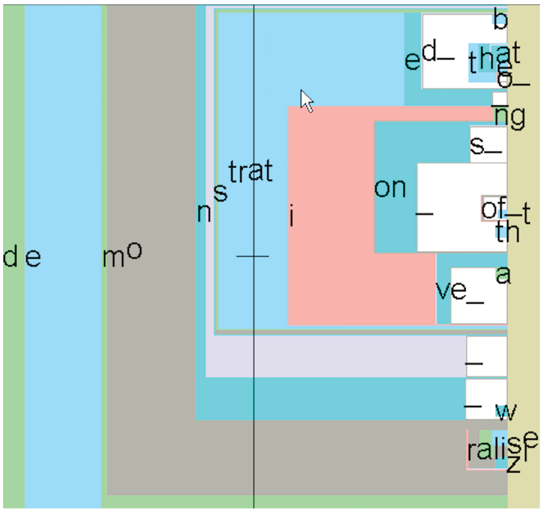


Figure 19: A user is writing "demonstration" in Dasher. The nearby alternatives such as "demonstrated that", "demonstrative", and "demoralise" can be selected by moving the cursor up or down.

Dasher [137, 209, 210] is one of the few information-efficient text entry methods. Unlike most other text entry methods, in Dasher a user's gesture does not map to a particular letter or word. Instead, text entry involves continuous visually-guided control of gestures via, for example, a mouse or eye gaze. The prediction of the next word is expressed in a colored piece of text. The more probable the text, the more space it is given so that it is easier to select (Fig. 19). A user writes text by continuously navigating to a desired sequence of letters in a graphical user interface laid out according to a language model. If the user writes at a rate R_D (in bits) then Dasher attempts to zoom in to the region containing the user's intended text by a factor of 2^{R_D}. If the language model generates text at an exchange rate of R_{LM} then the user will be able to reach an entry rate of R_D/R_{LM} characters per second [2].

Other intelligent text entry methods incorporate models such as Gaussian Process to improve text entry rate. Building on Kristensson et al. [117, 118], Weir et al. [212] combined a language model with a touch model to account for the inherent uncertainty of the touching process on mobile devices. The statistical decoder is often achieved by employing a Markov chain model [148] and token passing strategies [221] (Fig. 20). Weir et al. added a Gaussian Process (GP) touch distribution model (the likelihood) to capture individual users' physical uncertainty and demonstrated that such a method reduced the character error rate by 1.0% when participants are standing still when typing and by 1.3% when participants are walking when typing.

---

[2] http://www.inference.org.uk/dasher.

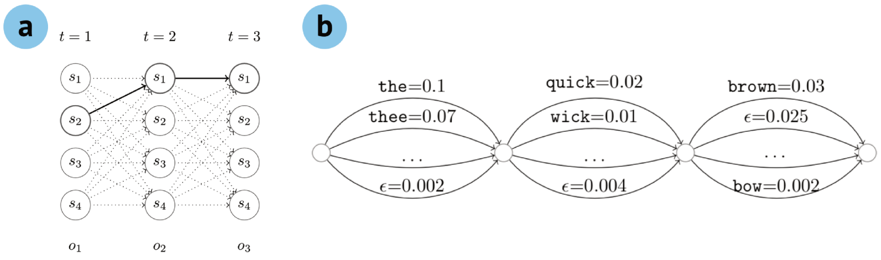


Figure 20: (a) Trellis of an observation sequence $O = \{o_1, o_2, o_3\}$. The thick arrows indicate an example of a highest probability path. (b) An example of a word confusion network [147]. ϵ represents a ϵ-transition state to each observation index that allows a token to propagate to the next observation index without generating a corresponding letter. Figure taken from [165].

These are two examples among many intelligent text entry methods [116]. Fundamentally, such methods aim to amplify users' ability to communicate as quickly and as accurately as possible by exploiting redundancies in natural languages. While the information-theoretic concept of bitrate (Equation 33) provides a natural and intuitive measure of text entry rate, however, such a method has not been widely adopted by the HCI community. To describe the speed of typing, researchers tend to use *words per minute* (WPM), which is measured in words entered in a minute and the definition of each word is often standardized to be 5 characters. Describing accuracy is more problematic due to the nature of mistakes: there are at least 4 basic types of errors, including entering an incorrect character (substitution), omitting a character (omission), adding an extra character (insertion), or swapping neighboring characters (transposition).

Soukoreff and Mackenzie [196] outlined the recent developments in text-entry error rate measurement including total error rate, not-corrected error rate, corrected error rate, KSPC (keystrokes per character) [139], and Levenshtein string distance statistic [195]. KSPC represents the number of keystrokes required, on average, to generate a character of text for a given text entry technique in a given language. For instance, the Qwerty keyboard has KSPC = 1 as each letter has a dedicated key. If a user perfectly types a sentence without making any mistakes, KSPC also equals 1. However, if she mistypes a letter, erases it and inputs the correct letter (2 more steps), $\text{KSPC} > 1$. In a way, KSPC describes whether there are extra actions involved in typing, whether they are due to error or not. The Levenshtein string distance (LD) [40, 127] is a measure of the similarity between two strings. For instance, if string s is "test" and string t is "test", then $\text{LD}(s, t) = 0$, because no transformations are needed. The strings are already identical. If s is "test" and t is "tent", then $\text{LD}(s, t) = 1$, because one substitution (change "s" to "n") is sufficient to transform s into t. The greater the Levenshtein distance, the more different the strings are.

A typical use example of these measures is described as follows:

Intended sentence: *I am fine if you ask.*
Transcribed sentence: *I a**n** fin**w** ← **e** if you ask.*

If we assume that a rather expert user [3] transcribed the sentence in 4 seconds, then we can compute the measures:

$$\text{Words per Minute(WPM)} = 21 \div 5 \div 4 \times 60 = 63$$
$$\text{Not Corrected Error Rate} = 1 \div 22 \times 100\% = 4.5\%$$
$$\text{Corrected Error Rate} = 1 \div 22 \times 100\% = 4.5\%$$
$$\text{Total Error Rate} = 2 \div 22 \times 100\% = 9.0\%$$
$$\text{Keystrokes per Character (KSPC)} = 23 \div 21 = 1.09$$
$$\text{Levenshtein String Distance (LD)} = 1$$

Keystrokes per character (KSPC) corresponds to corrected error rate while Levenshtein string distance to not-corrected error rate. We can see that even with such a complicated error measurement approach, the 4 types of errors are still not fully taken into account. Indeed, measuring errors has proven to be difficult in text entry. But one can hardly make meaningful observations about speed in the absence of accuracy. As a result, in controlled experiment settings, participants were often instructed to consciously limit errors within a reasonable range, such as 4% (e. g. [53]), so that only the speed dimension is of concern.

It is surprising that so few researchers have taken advantage of information theory to measure performance in text entry when it has already been used for statistical language modeling. The notion of equivocation $H(X|Y)$ (Equation 2 and 3) naturally provides measures for errors and how one can recover the source messages from the received ones. I will demonstrate in Part iii that we can use information-theoretic measures for evaluating text entry performance.

5.2 HUMAN INFORMATION CAPACITY

The notion of human information capacity, or rather the notion of throughput, has been mostly used in aimed movement, for instance, selecting targets with the mouse yields throughput of 3.7-4.9 bits per second [197]. It has also recently been applied to full-body movements [164] [4].

---

[3] An average professional typist types usually at speeds of 50 to 80 wpm.

[4] Like many other researchers (e. g. Mackenzie [141], Zhai [222] and Guiard [89]), Oulasvirta et al. [164] was in fact measuring throughput, not capacity in their study. The notions of channel capacity C and information transmitted at a rate R are defined in Chapter 1 Section 1.3.

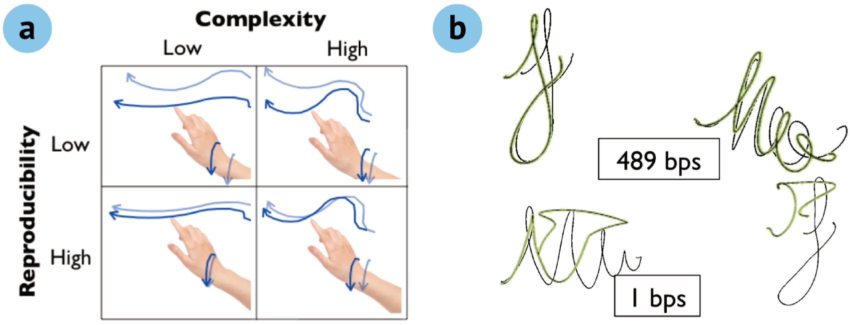


Figure 21: (a) Mutual information of repeated movements (here: blue and light-blue trajectories), determined by their complexity and the precision of reproduction. Highly complex and precisely reenacted movements yield the highest throughputs. (b) Study 3: Bimanual in-air gesturing: the best (top) and the worst (bottom) performance respectively. Figure adapted from [164].

Oulasvirta et al. [164] extended the notion of throughput to study human control of continuous sensors. Their method takes input motion data with any number of movement features (observation points on the human body) and calculates throughput from mutual information of two or more deliberately repeated movement sequences. The definition of mutual information captures the intuition that a skilled actor can produce complex (surprising) movements and reenact them precisely at will (Fig. 21 (a)). With these measures, Oulasvirta et al. [164, 191] investigated this question in the context of 3D motion capture interfaces and touchscreens, targeting the related application of gesture recognition. In the first study with ballet teachers, they estimated throughput as high as 1307 bits per second. In the second study of cyclical tapping with a mouse, the estimated throughput is from 24 to 37 bits per second on average. In study 3 of bimanual in-air gestures, throughput was 182.7 bits per second with dominant hand removed, 217.8 bits per second with non-dominant hand removed, and 322.1 bits per second with both hands. The best and the worst performance can be seen in Fig. 21 (b).

Even though this method provides a quantitative measurement of throughput in the continuous space, one limitation is that all measurements are assumed to follow a joint Gaussian distribution so that the mutual information computation boils down to estimating a correlation coefficient rho with an additional bias due to the estimation of rho. Indeed, as Berdahl et al. [14] put it, "the authors appear to be overestimating the channel capacity... The reason for this is that they do not have a way of discounting certain input signals that may be impossible for users to perform – for example, it is not possible to put the human fingertips in any randomly selected arbitrary position in Cartesian space. However, since they are counting such orientations, the estimated channel capacity is greatly inflated and no longer directly comparable with traditional throughput measurements".

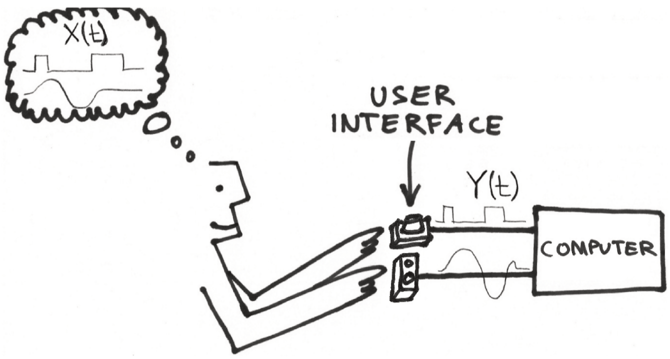


Figure 22: Model of a user controlling sound using continuous analog sensors. Taken from [14].

Instead, Berdahl et al. [14] proposed a model to account for human subject controlling a single, continuous sensor. Rather than estimating the joint probability function, the goal was to estimate channel capacity/throughput by asking human subjects to "perform" gestures that match idealized, band-limited Gaussian "target gestures". Then, the signal-to-noise ratio of the recorded gestures determines channel capacity/throughput. They find that the channel capacity for control of a single, continuous sensor is as high as 4 or 5 bits per second.

It is crucial to note, however, that test subjects in Berdahl et al.'s study were sufficiently trained in accurately operating the sensors. Although the work was only focused on users controlling sound using continuous analog sensors and sonic interaction design, the communication channel in their model was represented by a human together with a user interface, which could be potentially generalized and extended to other scenarios (Fig. 22). In Part iii, I will demonstrate such user-to-computer information transmission scheme.

5.3 POINTING WITHOUT A POINTER

Inspired by control theory [107], Williamson and Murray-Smith [217] in their work *Pointing without a Pointer* also proposed a multimodal communication channel between the human and the system that is bandwidth-limited both perceptually and physiologically [5].

---

[5] *Pointing without a Pointer* is an example of an interface built on methods from manual control theory - the study of how humans control dynamic systems, and close to the methods used in perceptual control theory [175]. It suggests that many kinds of behavior can be described as continuous control problems, and provides an empirical method for the estimation of a subject's intention.

They considered interface components as independent agents competing for user attention. Each agent is associated with an action in the user interface. Agents try to determine whether the user is interested in them by designing and running “experiments” to look for correlated responses in the actions of user. The experiments can take any form where the agent changes its state and tests for correlated responses in its inputs (movements of the mouse, for example). As the user’s attention is a scarce resource, the agents must compete to optimally ascertain user intentions with limited attentional resources.

One application of such a method is a selection task without requiring an explicit pointer as shown in Fig. 23 (a). Each object is considered to be an agent whose experiment on the user is the disturbance; intention is detected by looking for controlling behavior. The agents produce a continuous probability of selecting agent i, p_i.

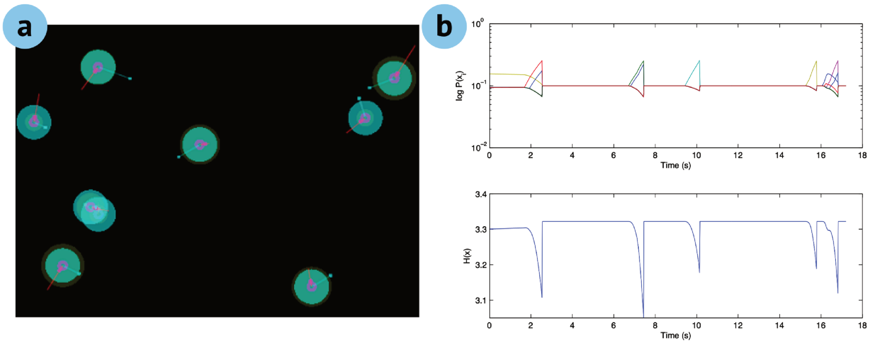


Figure 23: (a) The prototype of [217]. Each agent is shown as a circle with radius proportional to selection probability. The velocity of the objects is shown as an aid to users. (b) Probability time series for 10 objects with a number of correct selection events. Entropy is shown as the blue line below. Figures adapted from [217].

A selection event is considered to have happened when p_i exceeds some threshold, captured by the drop in entropy (Fig. 23 (b)), representing the accumulated information gradually gained by the system. By adding degrees of freedom using higher dimensional controllers and changing the presentation of the disturbances, such framework for probabilistic selection interfaces in continuous environments without a pointer can be adapted and extended. In Part ii, I will introduce a similar entropy-based method where the computer “runs experiments” and gains information from the user.

Coincidently, Williamson and Murray-Smith [217] were the first to describe excitation of displayed objects as a strategy for selection by making corresponding motions with an input device.

This work inspired Fekete et al. [54] to explore motion correlation as a selection technique in conventional graphical user interfaces. Rather than using pseudo-random movement, their idea was to associate objects with oscillatory movement, drawing on user's natural ability for harmonic motion with their hands. In their design of the motion-pointing technique, the graphical objects of interest retain their static presence in the interface but are augmented with a moving dot describing a small elliptical movement. These works represent milestones in establishing the motion correlation principle [207].

In summary, these studies demonstrate the potential of using information theory and information-theoretic notions to study the user-to-computer communication process and to design interaction with improved communication rate.

6

SUMMARY

I have presented various research endeavors applying information theory in psychology and in human-computer interaction. Some involve the communication channel with human users while others do not; some are more intuitive than others; some are more successful than others. I particularly want to summarize and highlight the following four aspects that motivate the rest of the thesis:

- Chapter 2: Indeed, there are a number of difficulties in applying information theory, notably the fact that the notion of *information* in information-theoretic terms has absolutely no semantic meaning; it is entirely described by a probability distribution. Measuring information content (entropy) in printed English text or multimedia content is absolute as long as the probability distribution of words and multimedia content (e. g. video frames) is considered objectively and as a matter of fact. On the other hand, if we were to derive information content from the stimuli based on human users' reaction, as in the stimulus-response paradigm, we need to make sure that the experimental setting corresponds to the information transmission process, rather than the information processing phase.
- Chapter 3 and 4: We have seen that the understanding and applications of the two main laws in HCI – Fitts' law and Hick's law – are problematic and I have provided an in-depth discussion of Hick's law in Chapter 4. It is necessary, therefore, to clarify what they are, how they should be used and when they do not apply. I believe that the HCI community at large can benefit from theoretically justified methods.
- Chapter 5: We have also seen the potential of investigating the information transmission process from the user to the computer using the tools of information theory. Particularly, how to quantify the information is an interesting question. In Part ii, I will introduce a Bayesian Information Gain framework that is based on Bayesian Experimental Design using the criterion of mutual information from information theory. This approach quantifies the information sent by the user to the computer to express her intention. By having the computer demanding more information at each time, I will show that the interaction & communication efficiency can be improved.

- Chapter 5: Since information theory has already been used in measuring entropy of English and measuring human performance, can we further extend it to describe interaction tasks at large? This generic communication scheme enables us to examine the communication process between the user and the computer with theoretically valid tools and provides several useful measures that have not been taken advantage of. In Part iii, I will introduce these information-theoretic measures for characterizing interaction tasks and demonstrate that it offers a richer picture of a given interaction scenario in comparison to the existing measurements.

We cannot foresee how information theory is going to affect and inspire future interaction design, nor can we guess which theory will be the next trend. Information theory was adopted and then dropped by psychologists, yet it still has much potential for understanding and designing the human-computer communication process. I hope this part has fulfilled its purpose: understanding the past as well as the present, and taking a glimpse into the future.

"The farther backward you can look,
the farther forward you are likely to see."
- Winston S. Churchill

Part II

A BAYESIAN INFORMATION GAIN (BIG) FRAMEWORK FOR QUANTIFYING INFORMATION

The goal of this part is to introduce Bayesian Information Gain (BIG), an information-theoretic framework based on Bayesian Experimental Design to quantify the information sent by users to computers in the interaction loop. BIG (a) allows the measurement of information in bits and (b) improves the efficiency of interaction & communication by maximizing or leveraging the expected information gain from the user's subsequent input.

I first introduce the BIG framework and then demonstrate two use cases: BIGnav, for multiscale navigation and BIGFile, for hierarchical file retrieval, both of which show a new way of interacting with improved communication efficiency, suggesting other possible "BIG" opportunities.

7

BAYESIAN INFORMATION GAIN FRAMEWORK

7.1 MOTIVATION

In this information-abundant world, a large amount of information is exchanged between users and computers: we obtain information from the computer to increase our knowledge and to complete tasks, and we send information to the computer to express our ideas and intentions. Several studies have investigated the information that users obtain *from* computers. For instance, Pirolli and Card [169] have introduced the concept of *information foraging*, describing the phenomenon that people adapt their strategies to increase information gain in an online information seeking task.

However, there is little understanding of the information sent by the user *to* the computer. We are familiar with the notion that we give inputs (or commands) to the computer, not information. Yet these inputs reflect the user's intentions, letting the computer know what is the user's goal, therefore, they represent information. This leads to a number of questions: how much information is there in these input commands? Can we quantify this information? If we can, what can we do with it and what does it imply?

In this part, we introduce a Bayesian Information Gain framework (BIG), based on Bayesian Experimental Design [132]. It uses the criterion of information gain, also known as mutual information in information theory [188], to quantify the information sent by the user to the computer in the interaction loop. Information is defined in terms of the computer's knowledge about what the user wants. At the beginning of the interaction, the user has certain goals, e. g. looking for a particular item on a website or typing a particular word on the keyboard. The computer has some uncertainty about the user's goal. This uncertainty is represented by the computer's prior knowledge, expressed in a probabilistic model. When taking input from the user, the computer updates its knowledge about what the user is looking for. Therefore, the information carried by the user input is the knowledge gained by the computer to know the user's goal.

One can simply use BIG to measure the information sent by the user to the computer. Furthermore, by maximizing or leveraging the expected information gain from the user's subsequent input through manipulation of the feedback, the computer can increase the information gain from the user, improving interaction efficiency.

7.2 BIG FRAMEWORK

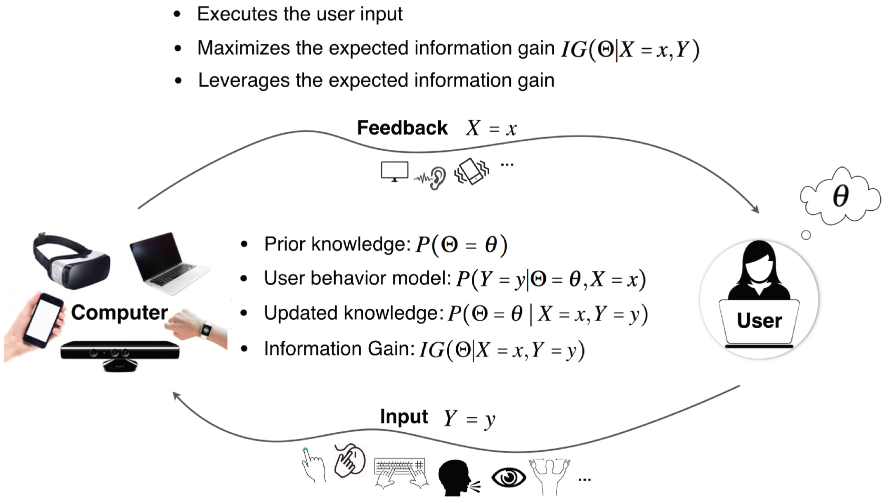

Executes the user input
Maximizes the expected information gain $IG(\Theta|X=x,Y)$
Leverages the expected information gain
Feedback $X=x$
...
θ
Prior knowledge: $P(\Theta=\theta)$
User behavior model: $P(Y=y|\Theta=\theta,X=x)$
Updated knowledge: $P(\Theta=\theta\mid X=x,Y=y)$
Information Gain: $IG(\Theta|X=x,Y=y)$
Computer
User
Input $Y=y$
...

Figure 24: The BIG framework: there are three key random variables: the potential targets Θ, system feedback X and user input Y. The computer also has some prior knowledge about the user's intended target $p(\Theta=\theta)$ and a user behavior function expressing what the user would do $p(Y=y|\Theta=\theta,X=x)$. After sending the feedback $X=x$ and receiving the user input $Y=y$, the computer updates its knowledge about the user's goal and calculates the information gain from the user input. In order to play a more active role, the computer can try to maximize the expected information gain or leverage it for better interaction by manipulating the feedback.

In this chapter, we introduce the Bayesian Information Gain (BIG) framework, which is a general approach that can be applied to a wide range of interaction tasks (Fig. 24).

BIG is inspired by Bayesian Experimental Design [132], which provides a framework to optimize the choice of an experiment x that will provide an observation y by maximizing an expected utility, commonly defined in terms of the information gained about the parameter θ by the experiment x. The utility may also involve factors such as the financial (or other) cost of performing the experiment.

In the BIG framework, the computer "runs experiment" on the user by sending the feedback $X=x$ (experiment) and receives user's subsequent input $Y=y$ (observation) to update its knowledge about the user's goal (parameter).

BIG uses the following notations that are common for Bayesian Experimental Design [132]:

1. Θ represents the possible intended targets in the user's mind.

2. $p(\Theta = \theta)$ for all values of θ is the prior probability distribution of target, which expresses the computer's prior knowledge about the random variable Θ. $p(\Theta = \theta)$ can be uniform if no data about the user's interests is available, or can be based on external data sources or interaction history.

3. X represents any possible feedback provided by the computer and $X = x$ is a particular feedback sent to the user.

4. Y represents any particular command y issued by the user.

5. $p(Y = y|\Theta = \theta, X = x)$ is the probability of the user giving an input command $Y = y$ when she wants $\Theta = \theta$ and sees $X = x$. This can be modeled from the interaction history, or by user calibration, and can be user-independent.

6. $p(\Theta|X = x, Y = y)$ is the computer's updated knowledge about the user's goal after showing the user $X = x$ and receiving the input $Y = y$ from the user. It is calculated through Bayes' theorem:

   $$p(\Theta = \theta \mid X = x, Y = y) = \frac{p(Y = y \mid \Theta = \theta, X = x)p(\Theta = \theta)}{p(Y = y|X = x)}. \tag{34}$$

   where $p(Y = y \mid X = x) = \sum_{\theta'} p(Y = y \mid \Theta = \theta', X = x)p(\Theta = \theta')$.

7. $I(\Theta; Y|X = x)$ is the mutual information between what the user wants and what she provides as input when seeing $X = x$. It is the difference between two uncertainties[1]:

   $$I(\Theta; Y|X = x) = H(\Theta) - H(\Theta|X = x, Y). \tag{35}$$

   This can also be interpreted as the *expected* information gain, which is always positive. To calculate this, we use Bayes' theorem for entropy to convert Equation 35 to:

   $$I(\Theta; Y|X = x) = H(Y|X = x) - H(Y|\Theta, X = x). \tag{36}$$

   where the first term is given by:

   $$\sum_{y} p(Y = y|X = x) \log_2 p(Y = y|X = x).$$

   and the second one by:

   $$\sum_{y,\theta} p(\Theta = \theta)p(Y = y|\Theta = \theta, X = x) \log_2 p(Y = y|\Theta = \theta, X = x).$$

---

[1] For a given X, knowing Y decreases uncertainty (increases knowledge) about Θ, by a quantity which is precisely the mutual information $I(\Theta; Y|X = x)$.

8. $IG(\Theta|X=x, Y=y)$ is the difference between the computer's previous knowledge $H(\Theta)$ and current knowledge $H(\Theta|X=x, Y=y)$ about the user's goal, representing the *actual* information carried by the user input:

$$IG(\Theta|X=x, Y=y) = H(\Theta) - H(\Theta|X=x, Y=y). \tag{37}$$

Information gain might be negative if the user, e. g. makes an error, but is positive on average[2].

Table 3 summarizes the notations in Bayesian Experimental Design and Bayesian Information Gain respectively.

	BED	*BIG*
θ	parameter to be determined	intended target in the user's mind
y	observation	user input
x	experimental design	system feedback
$p(y\|\theta, x)$	model for making observation y, given θ and x	model for user providing input y, given θ and x
$p(\theta)$	prior	system's prior knowledge about the user's goal
$p(\theta\|y, x)$	posterior	updated knowledge
$I(\Theta; Y\|X=x)$	utility of the design x	utility of the feedback x
$IG(\Theta\|X=x, Y=y)$	utility of the experiment outcome after observation y with design x	utility of the outcome after user input y with system feedback x

Table 3: Notations in Bayesian Experimental Design (BED) and in Bayesian Information Gain (BIG) respectively.

One can always calculate the actual information gain, or the information carried by the user input informing the computer what she wants with Equation 37 – "Running a normal experiment". By manipulating the feedback with Equation 35, e. g. finding the $X = x$ that maximizes or leverages the expected information gain, the system "runs a better experiment" on the user in order to gain more information about the user's goal, i. e. the intended target. The computer then plays a more active role and therefore increases interaction & communication efficiency.

---

[2] IG is an "instantaneous" quantity that is positive on average: $I = E_y(IG) \geqslant 0$.

In the next 4 chapters, I illustrate two applications of this BIG approach: maximizing the expected information gain in multiscale navigation (Chapter 9) and leveraging the expected information gain in hierarchical file retrieval (Chapter 11). In both cases, BIG is used in a different manner regarding the types of intended target Θ, system feedback X and user input Y and receives different subjective experience by the participants in the respective controlled experiments. Chapter 8 and Chapter 10 provide the context of multiscale navigation and hierarchical file retrieval respectively. In Chapter 12, I discuss how BIG is related to other conceptual frameworks, and outline opportunities for future work.

8

MULTISCALE NAVIGATION

Multiscale interfaces are a powerful way to represent large datasets such as maps, documents and high-resolution images. The canonical navigation commands in this type of interfaces are pan and zoom (as seen in most of the applications such as Google Maps [1]): panning lets users change the position of the view while zooming lets them modify the magnification of the viewport [71, 166]. Other representations include focus+context [211], overview+detail [183], treemap [193], hyperbolic tree [122], etc. (Fig. 25). In all cases they leave the user in complete control of navigation, leading to frustrating situations such as getting "lost in desert fog" where there is no information available to aid decision making [111].

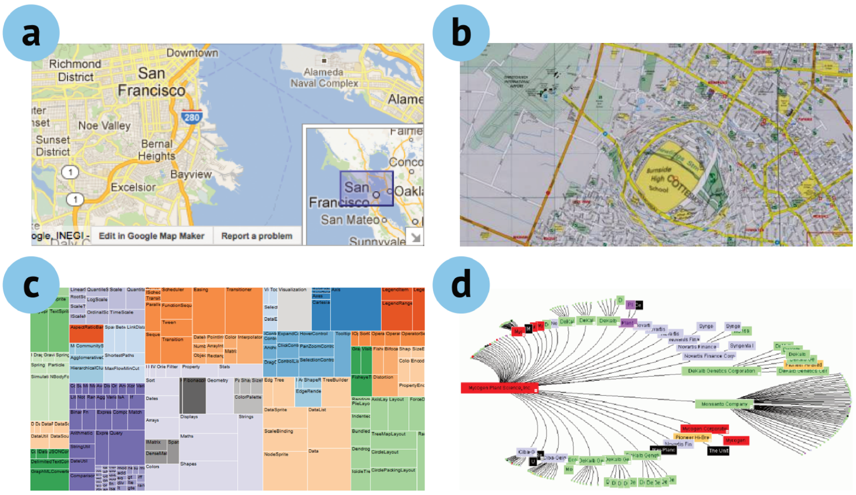

a
b
c
d
Richmond District
San Francisco
Downtown
Alameda Naval Complex
Noe Valley
Sunset District
Bernal Heights
Bayview
Excelsior
280
San Francisco
Oakl
San Mateo
Edit in Google Map Maker
Report a problem

Figure 25: Multiscale representations and interactions: (a) Overview+detail; (b) Focus+context; (c) Treemap and (d) Hyperbolic tree.

In this chapter, I review some assisted multiscale navigation techniques before introducing a guided multiscale navigation technique, BIGnav, in the next chapter.

Since pan and zoom are the most commonly used commands for multiscale navigation, much work has explored effective pan-and-zoom navigation. One can group these studies into 3 categories:

- Better understanding multiscale navigation;
- Assisting navigation by exploiting information space features;
- Interpreting user intention for guiding navigation.

---

[1] https://www.google.com/maps

8.1 BETTER UNDERSTANDING MULTISCALE NAVIGATION

Furnas and Bederson introduced space-scale diagrams, which provide an analytic and principled framework to examine multiscale navigation [70, 71] (Fig. 26). By representing both a spatial world and its different magnifications, the diagrams allow the direct visualization and analysis of important scale-related issues for interfaces.

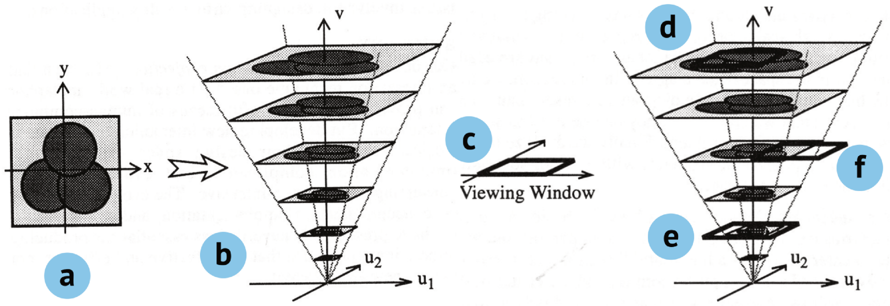


Figure 26: Space-scale diagrams: a 2D picture at its original scale (a) and at each possible magnification along the vertical scale axis (b); The viewing window keeps a fixed size and (c) is shifted along the space-scale diagram to (d) a zoomed-in view; (e) a zoomed-out view seeing the entire original picture and (f) a view of part of the picture. Taken from [71].

Navigation tasks consist in acquiring or pointing a specific target, characterized by a position and size. It generally involves view navigation, whereby the user must first bring the target into view, at a scale where it can be selected. Despite being quite different from traditional pointing, Guiard & Beaudouin-Lafon [86] used theoretical rationales and empirical studies to show that Fitts' law [63] applies to multiscale pointing for indices of difficulty (ID) beyond 30 bits, in contrast to the classical Fitts' law studies where ID has been confined in the 2-10 bit range (Fig. 27). In particular, Fig. 27 (b) illustrates the reduction of ID over time. In contrast to this progressive decrease with continuous panning and zooming, we will see in the next chapter that BIGnav works differently.

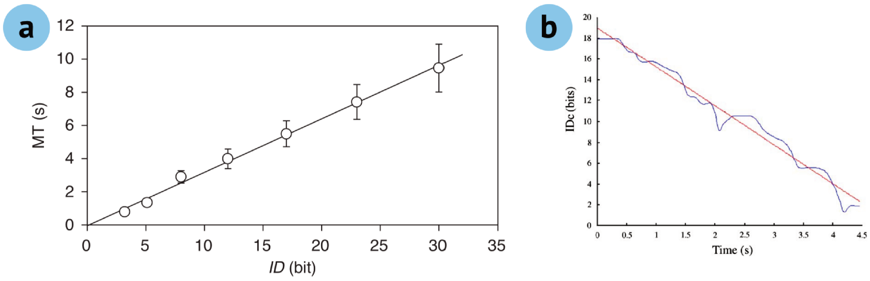


Figure 27: (a) Fitts' law assessed over a selection of IDs ranging beyond 10 bits and (b) Evolution of the current level of ID over a representative target-reaching movement. Taken from [86].

8.2 EXPLOITING INFORMATION SPACE FEATURES

Techniques in this category assist navigation by taking advantage of the system's knowledge of the information space. While users might not be fully aware of the features in the multiscale world, the system can facilitate navigation by exploiting such information and therefore help the user interact with it more effectively, such as avoiding getting "lost in desert fog".

Javed et al. introduced GravNav [108], a gravity-inspired navigation that uses the underlying visual space to assist navigation. GravNav calculates an attention vector depending on the viewport position in the space and the surrounding objects of interest. This vector is then used to modulate the speed or even the direction of the user's interactions to facilitate navigation to objects of interest (Fig. 28 (a)).

Ishak and Feiner [106] introduced content-aware scrolling, a method that varies the direction, speed, and zoom during scrolling, based on document content properties. For instance, in Fig 28 (b), the scrolling speed is varied based on the locations of important regions (solid black parts of path) and unimportant regions (dotted black parts of path) in a reading task (left) and in a text search task for "people" in the same page (right).

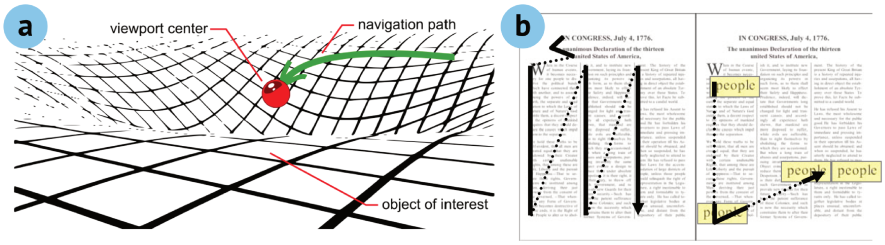

a
viewport center
navigation path
object of interest
b
IN CONGRESS, July 4, 1776.
The unanimous Declaration of the thirteen united States of America,
people
people
people
people

Figure 28: (a) GravNav: illustration of an attention gravity well for an object and (b) Content-aware scrolling. Taken from [108] and [106] respectively.

Galyean [75] used "The River Analogy" to assist 3D navigation in a virtual reality environment: it pulls users along a pre-computed path, while still letting them deviate slightly from it. As shown in Fig. 29, there are a number of different parts: the anchor moving along the path, a spring attaching the user position to the anchor, a thrust imparted by the user dictated by the direction the user is looking, and a general viscous damping to prevent the user from oscillating around the anchor position.

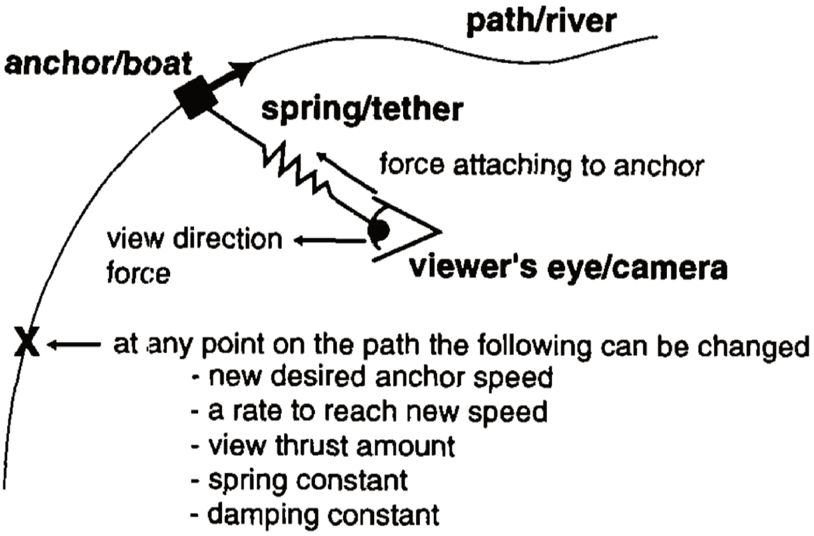

path/river
anchor/boat
spring/tether
force attaching to anchor
view direction force
viewer's eye/camera
X
at any point on the path the following can be changed
- new desired anchor speed
- a rate to reach new speed
- view thrust amount
- spring constant
- damping constant

Figure 29: "The River Analogy". Taken from [75].

In the same vein but first introduced as pointing techniques, object pointing [87] and semantic pointing [19] also "steer" users towards potential targets, therefore reducing the risk of getting lost. Object pointing [87] introduces a special screen cursor that skips empty spaces, and allows jumping directly to the target. Semantic pointing [19] uses two independent sizes for each potential target presented to the user: one size in motor space adapted to its importance for the manipulation, and one size in visual space adapted to the amount of information it conveys. The disentanglement of motor space from visual space facilitates pointing movement and can be easily extended to multiscale navigation.

8.3 INTERPRETING USER INTENTION

Another family of techniques guides navigation from the user's side by interpreting user intention.

Igarashi & Hinckley proposed Speed-Dependent Automatic Zooming (SDAZ), where the zooming factor depends on the user-controlled velocity so that users do not need to directly control zooming [105]. The basic idea is to automatically shrink the information space (e. g. a large document) when the user scrolls fast, thus maintaining constant perceptual scrolling speed and presentation of the global overview of the information space. As shown in Fig. 30 (a), the document automatically zooms out when the user scrolls fast. The speed of visual flow across the screen is held constant. Section headings and images become salient in the zoomed-out view to guide navigation.

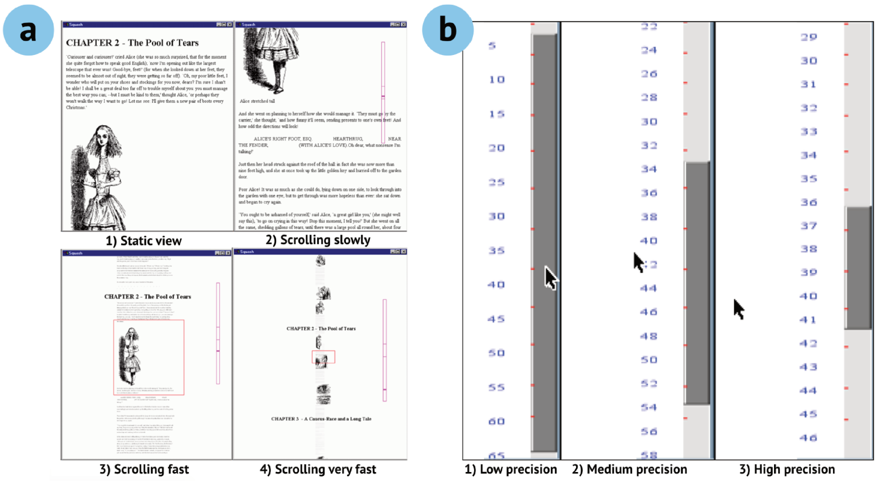


Figure 30: (a) Scrolling a long document using SDAZ and (b) The Orthozoom scroller. Taken from [105] and [5] respectively.

By contrast, Appert & Fekete created Orthozoom, a 1D scroller where users control zooming by moving along the scroller's orthogonal dimension, improving navigation in very large documents [5]. Orthozoom behaves like a traditional slider when the mouse is moved within the bounds of the slider. When dragging the mouse outside the bounds of the slider, it continuously changes the granularity of the slider and the scale of the document (Fig. 30 (b)). The granularity and scale decrease as the mouse cursor goes farther away from the slider bounds. In a controlled experiment, Appert & Fekete showed that Orthozoom performed twice as fast as SDAZ (Fig. 31) with an ID of up to 30 bits.

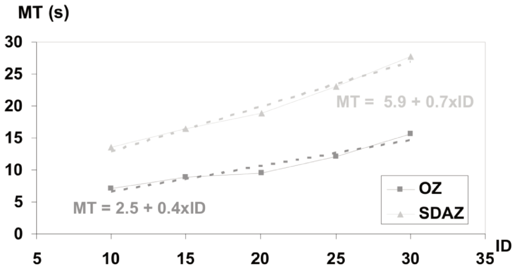


Figure 31: Controlled experiment comparing Orthozoom (OZ) with SDAZ regarding mean movement time as a function of ID. Taken from [5].

While these approaches have proven effective, they benefit from only taking advantage of the information from one side: either the information in the large spaces, or the information in user intention. In the next chapter, I introduce a BIG-inspired multiscale navigation technique, BIGnav, which uses both the information space features and interprets user intention to constantly update the system knowledge and guide navigation.

9

MULTISCALE NAVIGATION: BIGNAV

In this chapter, I introduce BIGnav, a guided multiscale navigation technique that is based on the BIG framework where the system tries to gain maximal information at each interaction step. I first describe each variable in BIGnav, demonstrate the 1D and 2D implementations, report a controlled experiment, and illustrate a more realistic application application: BIGMap.

The work presented in this chapter is published at ACM CHI 2017: BIGnav: Bayesian Information Gain for Guiding Multiscale Navigation. Wanyu Liu (WL), Rafael Lucas D'Oliveira (RD), Michel Beaudouin-Lafon (MBL), and Olivier Rioul (OR). (Best of CHI Paper Award)

Conceptualization: WL, RD
Implementation: WL
Experimental design: WL, RD
Evaluation: WL
Data analysis: WL
First draft: WL
Paper writing: WL, RD, MBL, OR
Supervision: OR, MBL

9.1 BIGNAV

BIGnav is a guided multiscale navigation technique based on the BIG framework (Fig. 32). The three key random variables Θ, X and Y represent the following, respectively:

- Θ represents any point of interest in the multiscale space. For each target θ, the probability that it is the actual intended target is $P(\Theta = \theta)$. These probabilities constitute the a priori knowledge that the system has about the user's interest, and is updated as the user navigates.
- X represents any possible view provided by the system. $X = x$ is a particular view shown to the user. Note that the number of possible views is potentially very large.
- Y represents any particular command y issued by the user. The possible input commands are: move towards a direction, zoom in or click on the target when it is big enough to be clickable. Note that zooming out is not required in this framework: if the target is out of view, the user should indicate in which direction it is rather than zooming out.

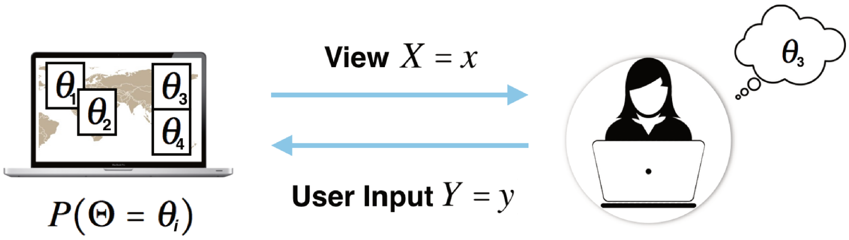


Figure 32: Illustration of BIGnav: the user is looking for a particular city θ_3 from all the potential targets Θ and she receives the system feedback X, which is a different view at each step, and provides input Y.

BIGnav guides navigation through 3 steps:

(1) Interpreting user input: Given the view x shown to the user and the user's intended target θ, $p(Y = y|\Theta = \theta, X = x)$ is the probability that the user provides an input command $Y = y$ given θ and x. This probability distribution is the system's interpretation of the user's intention when giving this command. For instance, if city A is to the left of the user, what is the probability of the user giving the left command when knowing that city A is located to her left, provided she can only go left or right? $p($*go left* | *city A is the intended target, city A is located to the left of the current view*$) = 1$ if the user is completely confident about what she is doing. But maybe the user is not accurate all the time. Say she is only correct 95% of time, then we need to consider that she makes errors. For instance, $p($*go left* | *city A is the intended target, city A is located to the left of the current view*$) = 0.95$ and $p($*go right* | *city A is the intended target, city A is located to the left of the current view*$) = 0.05$. $p(Y|\Theta = \theta, X = x)$ is a priori knowledge that must be given to the system. In the implementation section, we describe how we define it in 1D and 2D situations respectively.

(2) Updating system's knowledge: Given the view x shown to the user and the user reaction y to that view, the system can update its estimate $p(\Theta|X = x, Y = y)$ of the user's interest with Equation 34. If the system has no prior knowledge about the user's intended target, e. g. at the beginning, each θ has the same probability of being the target and $p(\Theta)$ is uniform. As the user issues commands, the system gains knowledge about the likelihood that each point of interest be the target, reflected by the changes to the probability distribution. This is done, for each point of interest, by taking its previous probability, multiplying by the above user input function $p(Y = y|\Theta = \theta, X = x)$, and normalizing it so that the sum of the new probabilities over all the points of interest equals one.

(3) Navigating to a new view: With the new probability distribution after receiving user input, BIGnav then goes over each view $x \in X$, calculates its expected information gain with Equation 36 and picks the view for which it is maximal. To maximize Equation 36, BIGnav looks for a trade-off between two entropies. To maximize the first term, the view should be such that all user commands given that view are equally probable (for the system). To minimize the second term, the view should provide the user with meaningful information about the points of interest. Maximizing a difference does not necessarily mean to maximize the first term and minimize the second, so the maximum information gain is a trade-off between these two goals. For example, showing only ocean will increase the first term but will also increase the second term. After locating the view with maximal information gain, BIGnav navigates there and waits for the user's next input.

9.2 IMPLEMENTATION: BIGNAV IN 1D

We first use a 1D example to walk through how BIGnav works step by step (Fig. 33).

Fifty cities are the points of interest, therefore $\Theta = \{1, 2, \ldots, 50\}$. The system does not have prior knowledge about the user's intended target city, so the initial distribution is $p(\Theta_1 = i) = \frac{1}{50}$. The view presented to the user at each step is defined by $X = \{[a, b] \subseteq [1, 50]\}$. The maximum zoom factor is such that a view cannot be smaller than two blocks ($b - a \leqslant 2$). Since it is a 1D map, the user can go to the left, go to the right, zoom in or select the target if the view is at the maximum scale. We note these commands $Y = \{\leftarrow, \rightarrow, +\ (\text{zoom in}), \bullet\ (\text{click target } i)\}$.

We start by modeling the user's behavior. We consider that the user makes some mistakes when panning and zooming, but will not miss the target when it is shown in the view and clickable:

$$p(Y = \rightarrow \mid \Theta = \theta, X = [a, b]) = \begin{cases} 0.9 & b < \theta \\ 0.05 & a < \theta \\ 0.05 & a \leqslant \theta \leqslant b \text{ and } b - a > 2 \\ 0 & a \leqslant \theta \leqslant b \text{ and } b - a \leqslant 2 \end{cases}$$

$$p(Y = \leftarrow \mid \Theta = \theta, X = [a, b]) = \begin{cases} 0.05 & b < \theta \\ 0.9 & a < \theta \\ 0.05 & a \leqslant \theta \leqslant b \text{ and } b - a > 2 \\ 0 & a \leqslant \theta \leqslant b \text{ and } b - a \leqslant 2 \end{cases}$$

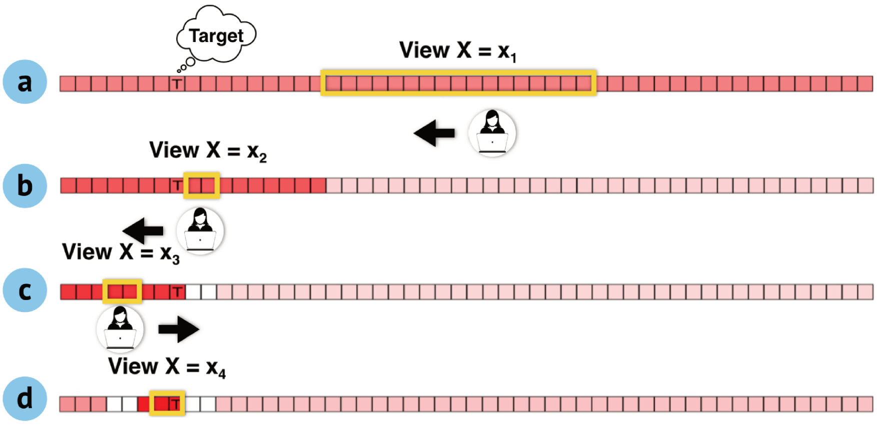


Figure 33: The user navigates to a particular city (T) among 50 others with BIGnav in 4 steps from (a) to (d). The color gradient shows the probability of each city being the user's target. The redder, the higher the probability. The yellow rectangle is the view that the system sends to the user.

$$p(Y = + \mid \Theta = \theta, X = [a, b]) = \begin{cases} 0.05 & b < \theta \\ 0.05 & a < \theta \\ 0.9 & a \leqslant \theta \leqslant b \text{ and } b - a > 2 \\ 0 & a \leqslant \theta \leqslant b \text{ and } b - a \leqslant 2 \end{cases}$$

$$p(Y = \bullet \mid \Theta = \theta, X = [a, b]) = \begin{cases} 1 & a \leqslant \theta \leqslant b \text{ and } b - a \leqslant 2 \\ 0 & \text{otherwise.} \end{cases}$$

In Fig. 33, the cities are represented by square boxes and colored in shades of red indicating the degrees to which the system believes the city is the target, i. e. city i is darker than j if $p(\Theta = i) > p(\Theta = j)$. City 8 has a **T** indicating that it is the target. The yellow rectangle is the view that the system shows to the user. After seeing the view, the user provides an input command y to the system.

We can now show BIGnav in action.

Step 1 (Fig. 33 (a)): Since the initial distribution is uniform, the system's uncertainty about the user's target is $H_1 = H(\Theta_1) = \log_2 50 = 5.64$ bits.

The system then goes over every image $[a, b]$, finds that $[18, 34]$ maximizes the expected information gain and displays the corresponding initial view to the user. In this case the expected information gain from the user's next action is $IG(\Theta_1 \mid X = [18, 34], Y) = 1.08$ bits.

The user inputs ← after seeing $[18, 34]$. The system then updates its knowledge with Equation 34 and ends up with a new distribution Θ_2 given by $p(\Theta_2) = p(\Theta_1 \mid X = [18, 34], Y = \leftarrow)$. Using Bayes' theorem we have:

$$p(\Theta_2 = i) = \begin{cases} 0.05 & i < 18 \\ 0.002 & i \geqslant 18. \end{cases}$$

The updated uncertainty is $H_2 = H(\Theta_2) = 4.65$ bits, resulting in an actual information gain $H_1 - H_2 = 0.99$ bits, very close to the expected information gain of 1.08 bits.

Step 2 (Fig. 33 (b)): The system now searches for the best view using the new distribution $p(\Theta_2)$, finds that it is $[9, 10]$ with an expected information gain of $IG(\Theta_2 \mid X = [9, 10], Y) = 1.24$ bits and displays it to the user. The user then inputs ← after seeing $[9, 10]$. The system then updates Θ_2 to Θ_3 as follows:

$$p(\Theta_3 = i) = \begin{cases} 0.12 & i < 9 \\ 0 & 9 \leqslant i \leqslant 10 \\ 0.006 & 10 < i < 18 \\ 0.0003 & i \geqslant 18. \end{cases}$$

The entropy of Θ_3 is $H_3 = 3.36$ bits, so the actual information gain for this step is $H_2 - H_3 = 1.29$ bits, higher than the expected information gain of 1.24 bits.

Step 3 (Fig. 33 (c)): With the same process, the best view is now $[4, 5]$ with an expected information gain of $IG(\Theta_3 \mid X = [4, 5], Y) = 1.58$ bits. The user inputs →, leading to the updated distribution:

$$p(\Theta_4 = i) = \begin{cases} 0.01 & i < 4 \\ 0 & 4 \leqslant i \leqslant 5 \\ 0.28 & 5 < i < 9 \\ 0 & 9 \leqslant i \leqslant 10 \\ 0.015 & 10 < i < 18 \\ 0.0007 & i \geqslant 18. \end{cases}$$

The entropy of Θ_4 is $H_4 = 2.70$ bits, so the actual information gain is $H_3 - H_4 = 0.66$ bits, compared to the expected information gain of 1.58 bits.

Step 4 (Fig. 33 (d)): The best view is now $[7, 8]$ with an expected information gain of $IG(\Theta_4 \mid X = [7, 8], Y) = 1.84$ bits. The user sees that the target city is in the view and clicks on it. The updated distribution is updated to:

$$p(\Theta_5 = i) = \begin{cases} 1 & i = 8 \\ 0 & \text{otherwise.} \end{cases}$$

The entropy of Θ_5 is $H_5 = 0$ bits since there is no more uncertainty about the target. The actual information gain is $H_4 - H_5 = 2.7$ bits, while the expected gain was 1.84 bits.

In this way, the user finds her target city in only 4 steps. At step 1, BIGnav divides the map in 3 so that the three commands (left, right and zoom in) have equal probability. It does not consider a click as the view is still far from being fully zoomed-in to select the target. At step 2, one would expect it to divide the left third of the map in 3 again so that the view would be about 5 boxes wide. However, since it is close to the maximum scale, and it knows that the user never misses her target when it is in the view and is clickable, showing a 2-box zoomed-in view will give BIGnav extra information: if this is the target, the user will click on it; if it is not and the user moves away, the probabilities of these two boxes become 0. Step 3 and step 4 work similarly.

We ran 200 simulations with 50 cities and a uniform initial distribution and found that it required 3.3 steps on average.

9.3 IMPLEMENTATION: BIGNAV IN 2D

I implemented BIGnav in a 2D application using Java 1.8 and the open source ZVTM toolkit [167]. As for the 1D case, we need to define Θ, X and Y:

Θ represents points of interest in the multiscale information space. Each point θ_i is defined by a triplet (x_i, y_i, p_i) where (x_i, y_i) is the coordinate of point i and p_i is the dynamic probability that point i is the user's intended target.

X represents views that the system can show to the user. A view is defined by a triplet (vx, vy, z) where (vx, vy) is the center of the view and the zoom level z determines the view size. A view is fully zoomed in when $z = 1$. In traditional multiscale navigation, the system can pan and zoom continuously, leading to a huge number of possible views. With BIGnav, we need to calculate the information gain corresponding to every single view $X = x$, which would incur an enormous computational cost if views could be centered at any pixel and have any size. We therefore discretize the set of views by using tiles and discrete zoom factors. The tiles are 200×150 pixels each, and each tile can contain at most one point of interest. When $z = 1$, the view is composed of 4×4 tiles. Each successive value of z increases the number of tiles (5×5, 6×6, etc.).

Y represents input commands that the user can provide. In many pan-and-zoom applications, users can pan in any direction by a range of distances, and zoom in and out by fixed amounts. As for the views, we reduce this set of commands to make computation tractable in our prototype. We slice the view into nine regions representing eight panning directions and a central zooming region (Fig. 34). The eight panning regions have a 45° angle, and the zooming region is half the size of the view. A single movement of the mouse wheel triggers a zoom while a drag action triggers a pan. The angle between the mouse-down and mouse-up points of the drag determines the panning direction. The last input is a click on the target, available only when zoom level $z = 1$.

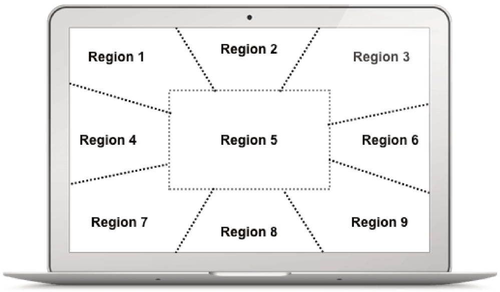


Figure 34: Nine regions representing user input, delimited by dotted lines. Panning regions also include the space outside the current view.

Command	*Main Region*	*Adjacent Regions*	*Other Regions*
Pan	0.90	0.04	0.0033
Zoom	0.95	0.00625	0.00625
Click	1	0	0

Table 4: Calibration results used as prior knowledge about the user behavior $P(Y = y | \Theta = \theta, X = x)$.

We now describe our implementation of the navigation steps.

(1) Interpreting user input: To interpret user input, we need to define $p(Y = y \mid \Theta = \theta, X = x)$, e. g. the probability of each command given a target and a view, e. g. $p($*pan East* $\mid$ *target (5, 7), view (4, 4, 2)*$)$. If the user were perfectly reliable, we could assign a probability of 1 to the correct command for each target θ and each view x, and 0 to the others. But we know that users make errors. To model the error rate, we collected data during a calibration session. The goal was to determine how confident users were when issuing commands. The task was to indicate in which direction the target was in a set of views. A set of concentric circles, identical to those used in the experiment below, showed the direction of the target when it was not within the view (Fig. 35 (a)).

We tested all ten input commands Y (8 pan operations, zoom in and click on the target) with 5 repetitions each, resulting in 50 trials per participant ($N = 16$). The results (Table 4) show that 90% of panning commands are correct and 4% are in one of the adjacent directions (Fig. 34). For zooming commands, 95% of the commands are correct while for clicking on the target, 100% of the commands are correct.

(2) Updating system's knowledge: We use Equation 34 to update the probabilities p_i of each point of interest being the target given the current view x:

For all points of interest θ_i*, the new* p'_i *is the previous* p_i *multiplied by the user expected behavior* $p(Y = y \mid \Theta = \theta_i, X = x)$ *divided by the normalization over all points of interest.*

(3) Navigating to a new view with maximum expected information gain: For each view x and each user input y, the expected information gain is the difference between two uncertainties:

Uncertainty before user input y *minus the sum of* $p_i \times \log_2 p_i$
over all points of interest
Uncertainty after user input y *minus the sum of* $p'_i \times \log_2 p'_i$
over all points of interest

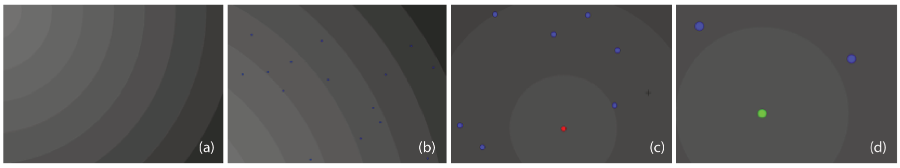


Figure 35: (a) Calibration: participants were asked to give a direction that indicates where they think the target is located (here: north-west). (b-c) Experimental condition with the target indicated in red. (d) Last step in the experimental condition: the target becomes green when it is clickable.

We then calculate the new view:

For all possible views x*, calculate expected information gain with Equation 36. Return the view* $(vx_{max}, vy_{max}, z_{max})$ *with maximum information gain and display it.*

The code is available on Github: `https://github.com/wanyuliu/BIGnav`.

9.4 CONTROLLED EXPERIMENT

Our goal is to study the performance of BIGnav compared with the standard pan & zoom navigation (STDnav) in what Javed et al. call *micro-level navigation*, when the user has decided on a destination and needs to navigate to it [108]. This is different from searching [168] or wayfinding [51] tasks where the user does not know where the target is located.

We conducted a controlled experiment where participants have to navigate towards a known target. Based on the theoretic analysis, we formulate four hypotheses:

H1: BIGnav is faster than STDnav for distant targets;
H2: BIGnav performs better in non-uniform information spaces, i. e. when the system has prior knowledge of the user's interest;
H3: BIGnav outperforms STDnav in terms of number of commands and rate of decreasing uncertainty;
H4: STDnav is preferred by users, more comfortable and more intuitive.

9.4.1 *Participants*

Sixteen participants (3 female), age 24 to 30 (mean = 25.9, σ = 1.7), were recruited from our institution and received a handful of candies for their participation. All of them were right-handed, had normal or corrected-to-normal vision, and were familiar with WIMP interfaces.

Participants were instructed to navigate to the target as fast as they could but were not informed of the condition they used.

9.4.2 *Apparatus*

The experiment was conducted on a MacBook Air with a 1.4 GHz processor and 4 GB RAM. The software was implemented in Java with the ZVTM toolkit [167]. The window was 800×600 pixels, centered on a 13-inch screen set to 1440×900 resolution. A standard mouse was used with the same sensitivity for all participants.

9.4.3 *Procedure*

We use a full-factorial within-participants design with a main factor: navigation technique (**Tech**); and two secondary factors: distribution of targets (**DISTR**), index of difficulty (**ID**).

9.4.3.1 *Navigation Technique (**Tech**)*

We compare BIGnav with standard pan-and-zoom:

- BIGnav: our guided navigation technique. The ten user commands (eight pan, zoom in, select target) and error rates are as described in the previous section.
- STDnav: the standard pan and zoom technique, used as baseline. A left mouse drag pans the view in world space proportional to the number of pixels dragged in screen space, and the mouse wheel zooms around the center of the view.

In order to compare information gains between the two conditions, we make the same computations as for BIGnav in the STDnav condition, except for the display of the new view.

9.4.3.2 *Distribution (**DISTR**)*

In order to compare different types of information spaces, we compared 6 distributions of points of interest by combining three spatial distributions (*Grid*, *Random* and *Cluster*) with three probability distributions (*Uniform*, *Random*, *Cluster*) of the a priori likelihood of each target. Since not all combinations are meaningful, we selected 6 of them. The first 3 have a uniform probability distribution, e. g. all points of interest have equal probability of being the target, and different spatial distributions:

- *Grid+Uniform*: Points of interest are arranged in a grid, providing a strong visual pattern.
- *Random+Uniform*: Points of interest are placed randomly.

- *Cluster+Uniform*: Points of interest are organized in clusters that are typical of geographical maps [28], where a central city is surrounded by smaller towns. We used 5 clusters of 10 targets.

The other 3 distributions use a non-uniform probability distribution of being the target:

- *Grid+Random*: Points of interest are on a grid with random probabilities of being the target. These probabilities are bounded by *Uniform* and *Cluster*.
- *Random+Random*: Points of interest are randomly distributed and have random probabilities.
- *Cluster+Cluster*: Points of interest are clustered and the probability of the center of each cluster is ten times higher than that of the surrounding points of interest.

The first 3 configurations are meant to demonstrate that BIGnav works well even without prior knowledge about potential targets. The other 3 configurations are meant to assess the added advantage, if any, of using such prior knowledge. In particular, the last distribution is typical of, e. g. maps.

9.4.3.3 *Index of Difficulty (**ID**)*

The ID is related to the distance between the initial position of the view and the target to navigate to. Using Fitts' definition of the ID [63], the distance D to travel is $D = 2^{ID} \times W$, where W is the (constant) target width. We adopted the same large IDs as in other multiscale navigation studies [5, 108]: 10, 15, 20, 25 and 30 bits.

We used a [2×6×5] within-subject design: we tested 2 **Tech** for 6 **DISTR** and 5 **ID** conditions. Each condition was replicated 5 times, so that each participant performed 300 trials. We blocked the conditions by **Tech**. Half the participants started with STDnav and the other half with BIGnav. Within each block, we systematically varied the order of **DISTR** and **ID** combinations across participants using a Latin square so as to reduce the influence of learning effects. For each condition, the targets were drawn randomly according to the probability distribution of the **DISTR** condition. All participants used the same target in the same **DISTR**× **ID**× Replication condition.

9.4.4 *Task*

The task is a multiscale pointing task: starting from a fully zoomed-out view, the participant must navigate towards the target until it is fully zoomed in and click on it. The target is surrounded by concentric circles so that it is always possible to tell in which direction and how far it is (Fig. 35).

The information space contains 50 points of interest: 49 are distractors and displayed in blue, one is the target and displayed in red. The ID is used to compute the scale of the initial view so that it contains all the points of interest. The target becomes green and clickable only when the view is fully zoomed in.

Participants first receive general instructions about the session and performed several practice trials with each technique. After the session, they answer a questionnaire asking them for feedback and comments on the experiment and the techniques. A typical session lasts 60 minutes, including training.

9.4.5 *Data Collection*

For each trial, the program collects the task completion time (TCT), the commands that the participants issued, the uncertainty and position of the view at each step and the information gain after each command. We collected 2 **Tech** × 6 **DISTR** × 5 **ID** × 5 Replications × 16 Participants = 4800 trials in total.

9.5 RESULTS

For our analyses, we first removed 23 missed trials (about 0.5%) and then 54 outliers (about 1.1%) in which TCT was 3 standard deviations larger than the mean. We verified that misses and outliers were randomly distributed across participants, techniques and conditions.

9.5.1 *Task Completion Time*

Table 5 shows the results of a repeated-measures full factorial ANOVA on TCT. All main effects are significant, as well as two interaction effects: **Tech** × **DISTR** and **Tech** × **ID**.

Figure 36 shows the interaction effect between **Tech** and **ID** for task completion time (TCT). On average, BIGnav is 24.1% faster than STDnav across all **ID**. A post-hoc Tukey HSD test reveals a robust interaction effect: BIGnav is significantly faster than STDnav when **ID** > 15 ($p < 0.0001$), significantly slower when **ID** $= 10$ ($p < 0.0001$) and not significantly different for **ID** $= 15$ ($p = 0.99$). These results support **H1**: BIGnav is 22.3% faster than STDnav for ID $= 25$ and 35.8% faster for ID $= 30$.

Factors	*df, den*	*F*	*p*
Tech	1, 15	4948.94	< 0.0001
DISTR	5, 75	38.32	< 0.0001
ID	4, 60	5363.35	< 0.0001
Tech × **DISTR**	5, 75	47.23	< 0.0001
Tech × **ID**	4, 60	955.19	< 0.0001
DISTR × **ID**	20, 300	1.26	$= 0.2$
Tech × **DISTR** × **ID**	20, 300	0.72	$= 0.8$

Table 5: Full-factorial ANOVA on TCT.

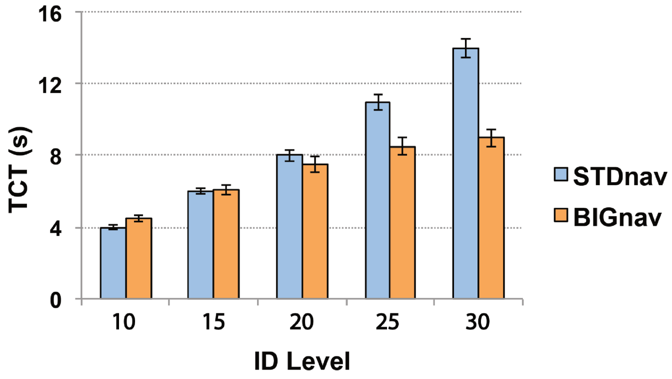


Figure 36: Means and confidence intervals of TCT by **ID**.

The ANOVA also reveals an interaction effect between **Tech** and **DISTR**. A post-hoc Tukey analysis shows that for STDnav, **DISTR** does not affect TCT. BIGnav, however, shows a larger advantage in non-uniform information spaces (13.7% faster, $p < 0.01$) than in uniform ones, with *Cluster+Cluster* being the fastest (Fig. 37), supporting **H2**. However, there is no significant difference among probability distribution conditions. For non-uniform distributions: *Cluster+Cluster* (6.51 ± 1.71s), and *Random+Random* (6.74 ± 1.55s), *Grid+Random* (6.62 ± 1.67s). For uniform distributions: *Grid+Uniform* (7.70 ± 1.60), *Random +Uniform* (7.83 ± 1.58s), and *Cluster+Uniform* (7.70 ± 1.62s).

We then further compare BIGnav and STDnav when **ID** > 15 in all **DISTR** conditions with a post-hoc Tukey HSD test. The results indicate that BIGnav is significantly faster than STDnav for distant targets especially in non-uniform information spaces. Particularly, when ID $= 25$, BIGnav is 16.9% faster than STDnav in uniform distributions ($p < 0.001$) and is 27.6% faster in non-uniform ones ($p < 0.0001$). When ID $= 30$, BIGnav is 31.7% faster than STDnav in uniform distributions ($p < 0.0001$) and is 40.0% faster in non-uniform ones ($p < 0.0001$).

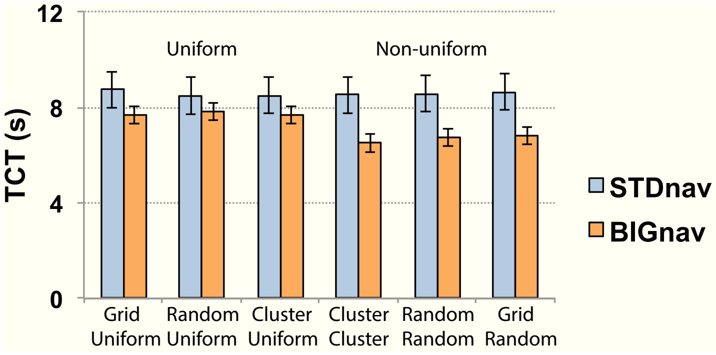


Figure 37: Means and confidence intervals for TCT by **DISTR**.

In summary, these results support hypotheses **H1** and **H2**: BIGnav is faster than STDnav for distant targets, especially in non-uniform information spaces. BIGnav is also not significantly different from STDnav for close targets (**ID** = 15).

9.5.2 *Number of Commands*

In order to get a sense of the differences in control strategies across conditions, we compare the number of user commands issued by the participants. Because of the continuous control in the STDnav conditions, we aggregate the mouse and wheel events as follows: we count one panning command per sequence from a mouse down to a mouse up, and one zooming command per series of mouse wheel with less than 300ms between them.

We perform a **Tech** × **DISTR** × **ID** full-factorial ANOVA on the number of commands issued (Table 6) and find that while **Tech** and **ID** significantly affect the number of commands used, **DISTR** has a non-significant effect. The ANOVA also indicates that the **Tech** × **ID** interaction effect is significant. A post-hoc Tukey HSD confirms that while the number of commands progressively increases with ID in STDnav, it is barely affected by ID in BIGnav (Fig. 38).

Factors	*df, den*	*F*	*p*
Tech	1, 15	364818.2	< 0.0001
DISTR	5, 75	0.99	$= 0.4$
ID	4, 60	10636.43	< 0.0001
Tech × **DISTR**	5, 75	0.23	$= 0.9$
Tech × **ID**	4, 60	11783.96	< 0.0001
DISTR × **ID**	20, 300	0.65	$= 0.9$
Tech × **DISTR** × **ID**	20, 300	0.71	$= 0.8$

Table 6: Full-factorial ANOVA on the number of commands.

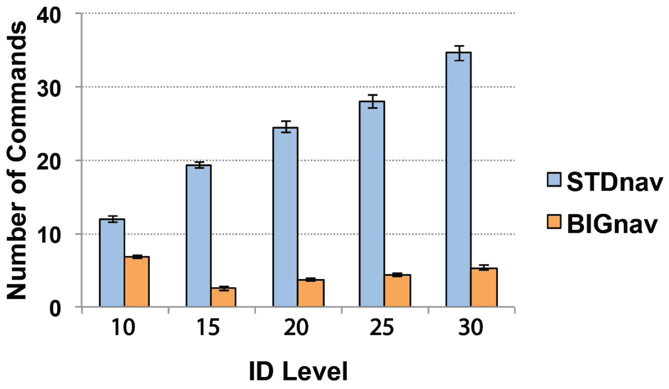


Figure 38: Means and confidence intervals for the number of commands by **Tech**.

For instance, when **ID** = 25, STDnav requires 28.1 commands on average whereas BIGnav requires 4.4 commands. When **ID** grows to 30, STDnav requires 34.5 commands on average whereas BIGnav requires only 5.8 commands.

Interestingly, BIGnav results in more commands for **ID** = 10 than for larger distances. Although it still outperforms STDnav, it requires significantly more commands for **ID** = 10 than for the other ID except = 30. The reason may be that the target falls into the view very quickly but BIGnav tends to move it away to gain more information, because it does not know it is the target. Some participants got frustrated by this behavior and started to issue arbitrary commands.

Regarding the ratio between pan and zoom commands, we find that in STDnav, $79.33 \pm 2.03\%$ of the commands are zooming commands v.s. $20.67 \pm 2.51\%$ for pans, but the proportions are reversed for BIGnav: $26.74 \pm 4.46\%$ for zooming v.s. $73.26 \pm 3.87\%$ for panning. This is because with BIGnav, zooming is only needed when the target is within the view, and panning commands most often result in a new view with a different level of zoom.

9.5.3 *Uncertainty and Information Gain*

Since the essence of BIGnav is to maximize the expected information gain at each command, we compare the actual information gain between STDnav and BIGnav. In both cases, the uncertainty that the system has about the users' intended target drops to zero gradually.

In STDnav, sometimes a command does not make a difference in uncertainty, i. e. the information gain is null. This is typically the case when the system is certain of what the user is going to do.

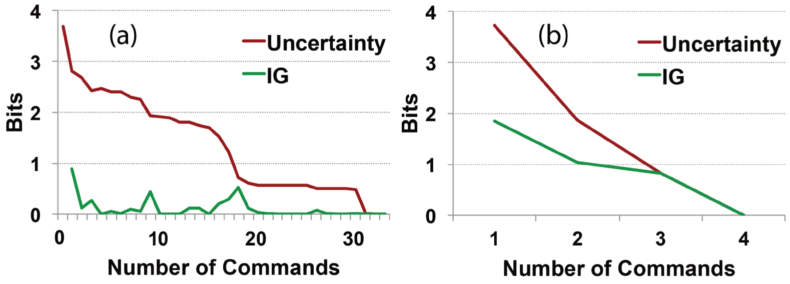


Figure 39: Uncertainty decrease and information gain for each successive command in (a) STDnav and (b) BIGnav.

For example, when completely zoomed out, users must zoom in. Similarly, if a view contains 99% of the probability distribution, users will almost certainly zoom in.

By contrast, with BIGnav, the system gains information at each step, therefore uncertainty drops to zero much faster and with many fewer commands. In our data, 40.4% of the commands in STDnav do not reduce uncertainty. The rest of the commands reduce uncertainty by 0.26 bits on average. In BIGnav, all commands reduce uncertainty by 0.88 bits on average. Figure 39 shows typical plots of uncertainty reduction for the two techniques and the same other conditions (*Random+Uniform* and ID $= 30$). These results support **H3**: BIGnav outperforms STDnav with much lower command usage and much higher rate of decreasing uncertainty.

9.5.4 *Trajectory in Multiscale Worlds*

Another way to look at the navigation strategies is to plot the reduction in **ID** over time. As participants pan and zoom, they get closer (most of the time) to the target and therefore the **ID** progressively decreases from the initial level to 0. Figure 40 shows typical plots for two trials by the same participant in the same condition (*Random+Uniform* and ID $= 30$). With STDnav, the reduction of **ID** is globally steady while with BIGnav we see sudden drops and long plateaus as well as occasional increases of the **ID**.

ID increases may occur for example when the view is close to the target and there is a cluster of points of interest further away in that direction. To maximize the expected information gain, BIGnav may choose to move towards the cluster and end up further away from the target. Another cause for ID increase is when the user makes a mistake.

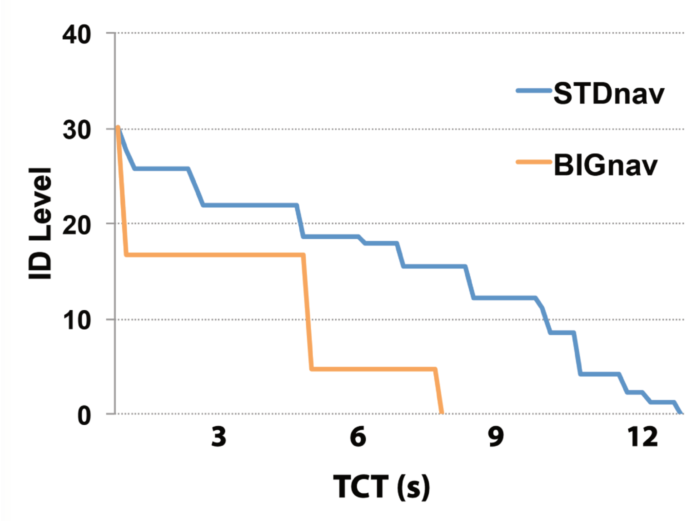


Figure 40: Time plot of the decrease in ID in the STDnav and BIGnav conditions, for two trials with the same other conditions.

The long plateaus represent waiting time and confirm the qualitative results reported below: that BIGnav incurs a higher cognitive load. This is probably due in part to our implementation of BIGnav, which skips to the new view after each input command rather than transition to it with an animation. But it is also probably the case that the user has to interpret the new location to plan the next move, whereas with STDnav the user can anticipate the system response.

9.5.5 *Qualitative Results*

The post-hoc questionnaire provides self-evaluation of performance and comfort level as well as subjective preference for the two techniques. Regarding performance and comfort level, assessed on a five-point Likert scale, we find no significant differences.

While we expected participants to dislike BIGnav because of its unusual and possibly counter-intuitive mode of operation despite its efficiency (**H4**), we were surprised that half the participants liked it better than STDnav: *"with one direction, it combines zoom and pan, which was faster than doing it by hand"*, *"The way it navigates is quite interesting. For most of the cases, only 2,3 actions are needed to find the target"*, and *"I like the interaction part. Somebody is guessing what I'm doing"*. The eight participants who preferred STDnav found it *"more comfortable, doesn't require that much attention"*, *"more intuitive as I can anticipate what I would see next"* and *"I'm already used to it"*. These results indicate that BIGnav can be a practical technique for efficient navigation.

9.6 APPLICATION: BIGMAP

To demonstrate a realistic application of BIGnav, I implemented a map application with a 80000×60000 pixel high-resolution map of Europe using the ZVTM toolkit [167] (Fig. 41). It features the top 50 largest urban areas[1] and uses their population as probability distribution. This corresponds to the *Random+Random* distribution of the controlled experiment.

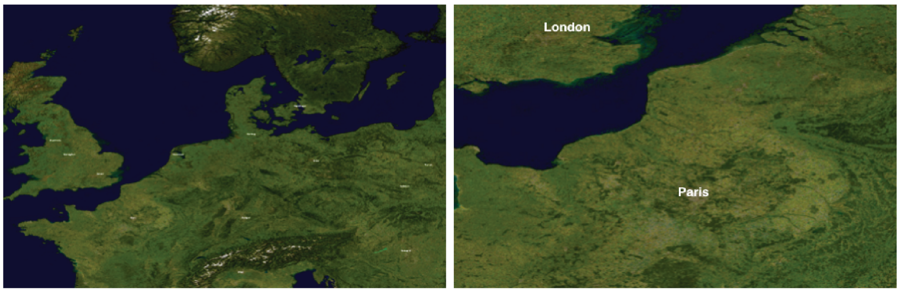


Figure 41: (a) Part of the map of Europe used by BIGMap. (b) Navigating towards Paris from the previous view.

We conducted several pilot studies where participants had to navigate to specific cities from a completely zoomed-out view down to the maximum scale, where the city labels are readable ($ID = 25$). Since this task relies on cognitive skills such as the participants' geographical knowledge, we concentrated on observing users and collecting subjective evaluation feedback. A similar version, implemented in Javascript using the OpenLayers API [2] can be found at: `https://perso.telecom-paristech.fr/wliu/BIGMap.html`.

Most participants could navigate to the target city very quickly, in a few steps, especially for cities with large population, hence higher probability, such as London and Paris. One of the participants referred to BIGnav as *"3 steps to Paris"*. For smaller cities such as Helsinki (rank 50), participants could still navigate efficiently. Most of them felt comfortable with BIGnav as they were familiar with the map of Europe and could reorient themselves rapidly. However, they got frustrated when the target was already in the view but BIGnav moved away from it in order to gain information. One participant mentioned that *"it would be nice if we could change between pan and zoom and this one [BIGnav] freely so that it can help us get through all the zooming at the beginning, but once I see the target, maybe I'll switch back to pan and zoom for the last few steps"*.

---

[1] `https://en.wikipedia.org/wiki/Larger_urban_zone`

[2] `https://openlayers.org`

BIGnav illustrates how to derive a probability distribution from external data, here the population of the cities. More generally, the distribution should reflect the targets' degree of interest, which is typically application-dependent. The distribution can also integrate usage data, such as the most popular cities. Finally the results of a search can be turned into a distribution according to the ranking of the results, therefore integrating searching and navigation into a single paradigm.

9.7 DISCUSSION

We have shown that BIGnav is an effective technique, especially for distant targets and non-uniform information spaces. The most efficient distribution condition in the experiment was *Cluster+Cluster*, which corresponds to the small-world structures found in a large number of datasets, showing that BIGnav is a promising approach for real-world applications.

However, both the experiment and the BIGnav prototype exhibit some shortcomings, especially for small-ID tasks. We now discuss how to make BIGnav more comfortable to use.

In standard pan-and-zoom interfaces, users can navigate the space in a continuous manner and constantly anticipate the system response. This gives them a sense of control and makes for a smooth user experience. By contrast, BIGnav uses discrete steps and the system's response can be difficult to anticipate and even frustrating, in particular when getting close to the target. This results in long idle times between commands (Fig. 40) and a higher cognitive load as users reorient themselves and decide on their next move. In a sense, this proves the success of the technique, since it is designed to maximally challenge the user at each step.

Yet there must be a way to improve user experience and make navigation smoother. First, we could use animations to smooth transitions and help users stay oriented [3]. Research has shown that one-second animations are sufficient and do not slow down expert users [12]. Second, we could combine BIGnav with standard pan-and-zoom according to user input: large panning and zooming movements would use BIGnav, smaller ones traditional pan and zoom. Finally we could reduce the size of the grid and increase the number of panning directions to provide finer control, however this requires heuristics or optimizations of the computational cost.

---

[3] This is implemented in BIGMap, but not in the original experiment.

In the next two chapters, I demonstrate another use of the BIG framework where improved efficiency and user experience are both met in the context of hierarchical file retrieval. I will return to more discussions of these two applications as well as the framework itself in Chapter 12.

10

HIERARCHICAL FILE RETRIEVAL

Retrieving files is an extremely common task for all computer users. We all have a large number of files and folders in our personal information systems and many interfaces support different ways of accessing them. One of the most common methods is navigating through a file hierarchical. Yet this process can be slow and repetitive: Bergman et al. found that Mac users spent 12 seconds and Windows users spent more than 17 seconds per retrieval on average [17]. Multiple views are available to represent the files, e. g. Fig. 42 shows the (a) Icon view, (b) list view, and (c) column view in the Finder on Mac OS X.

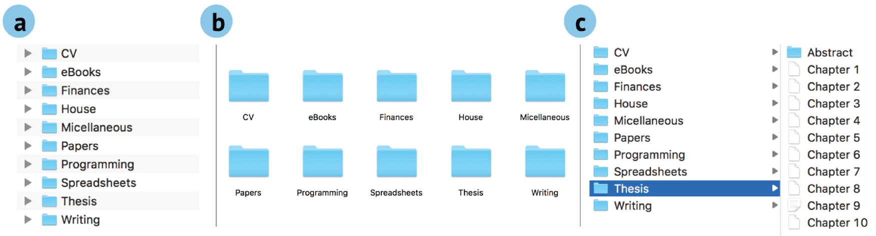

a
CV
eBooks
Finances
House
Micellaneous
Papers
Programming
Spreadsheets
Thesis
Writing
b
CV
eBooks
Finances
House
Micellaneous
Papers
Programming
Spreadsheets
Thesis
Writing
c
CV
eBooks
Finances
House
Micellaneous
Papers
Programming
Spreadsheets
Thesis
Writing
Abstract
Chapter 1
Chapter 2
Chapter 3
Chapter 4
Chapter 5
Chapter 6
Chapter 7
Chapter 8
Chapter 9
Chapter 10

Figure 42: Three common views to represent files & structures: (a) List view; (b) Icon view and (c) Column view.

Another common approach is to search by, e. g. typing one or more keywords. Despite continuous improvements in search algorithms, previous studies [11, 15, 21, 60] have shown users' continued preference for hierarchy-based file navigation over alternative methods. One of the most important reasons is that the locations and mechanisms of navigation-based retrieval remain consistent and reliable, whereas the organization and content of search results, which are extracted without context, can vary from one retrieval to the next, resulting in extra cognitive effort [15, 60, 124]. In addition to using search to potentially improve file access, e. g. [38], researchers have also looked at different aspects of file retrieval: how users organize personal information [17] and how visualization [109] and prediction algorithms [55, 59, 125] can improve efficiency.

We introduce BIGFile, a fast file retrieval technique based on the BIG framework where the computer tries to gain information from the user to determine which file or folder she is looking for. Unlike BIGnav, BIGFile features a split adaptive interface that combines a traditional file navigation interface with an adaptive part that displays the shortcuts selected by BIG so that users can navigate the list as usual, or select any part of the paths in the adaptive area.

I first review related work on personal file systems, file retrieval techniques, and adaptive user interfaces in this chapter. In the next chapter, I describe BIGFile's interface as well as its underlying algorithm and report on two studies.

10.1 PERSONAL FILE SYSTEMS

Many prior studies have investigated how people manage and retrieve information from their personal file systems.

10.1.1 *File Management*

File hierarchies are the predominant way to organize files: files and folders nested inside other folders. Previous studies that investigated how people structure their file hierarchies have found that these hierarchies are broad, shallow, and often unbalanced.

Gonçalves and Jorge [79] analyzed the structure of the file hierarchies of 11 participants, examining only portions containing user documents. Users averaged about 8000 files, however there was considerable variation. Folders contained an average of 13 files, had a branching factor (the average number of subfolders at a given tree level) of 1.84, and the hierarchies were fairly well balanced. The hierarchies had an average depth of 8.45. In an analysis of filenames, they found that 60% of filenames contained numbers, but only 0.33% contained dates. Filename lengths averaged 12.6 characters. However they are likely longer in modern systems as the study was conducted in 2003 and file systems no longer impose tight constraints on filename lengths.

Henderson and Srinivasan [96] ran a similar but larger scale study of Windows XP users in 2009, again analyzing portions of hierarchies that contained user documents. They found similar results to Gonçalves and Jorge: 5850 documents per user, an average tree depth of 9.65, folders containing an average of 11.1 files and a branching factor of 1.93. They also found that 74% of folders did not contain any subfolders, but the folders that did averaged 4.1 subfolders each. 7.9% of folders were completely empty. When performing name comparisons, 21.8% of filenames were duplicates, as well as 23.5% of folder names. Although the average maximum tree depth was 9.65, average depths within the trees were considerably smaller – 3.4.

Futhermore, users tend to have different habits for building these structures. Malone [145] described two types of users based on their document management strategies for paper documents, later referred to as *filers* and *pilers* [204]. *Filers* are more organized, quickly classifying new documents and placing them in an appropriate location. *Pilers* spend less effort organizing their documents, and their collections may appear to be less orderly. This reduced level of organization means that it can be harder to remember document locations. Whittaker and Hirschberg [215] also found that people often forgot the categories they had already created, leading to duplicate categories that meant that files were often overlooked when attempting to retrieve all the information on a topic.

10.1.2 *File Retrieval*

Similar to various file management practices, users retrieve files in different manners.

10.1.2.1 *How Fast Do Users Retrieve Files?*

In a large-scale study with 289 participants, Bergman et al. [17] examined how various factors affected file navigation (retrieving a file by traversing through the hierarchy using a file browser). Their method involved statically recording the state of participants' 'recent documents' list, then asking them to navigate to each of those files using a file browser, with video capturing their actions. By analyzing the video they found that Mac and Windows users structured their files in different ways, with Windows users keeping more files but fewer subfolders in each folder than Mac users. As a result, retrieved files were deeper in the file hierarchy on Windows (2.9 levels deep, compared to 2.4 levels on Mac OS X) and file retrieval times were slower (17.3 seconds on Windows, 12.6 seconds on Mac OS X).

Fitchett and Cockburn [60] conducted a 4-week empirical study to characterize 26 participants' actual file retrievals on their personal Mac computers. They found that the mean time to retrieve files using file browser navigation was 10.2 seconds, vs. 5.7 seconds when using Spotlight searches, and 16.5 seconds when using Finder searches. Their explanation for the high mean value with Finder search is that it was used for files that were harder to find. Since each navigation retrieval can be decomposed into a series of individual steps at each level of the hierarchy, they also analyzed 'step time', where each step descended to the next level of the hierarchy within a single window by opening a folder. They found that the mean step time was 3.6 seconds and that the step times were shorter at deeper levels, possibly because deeper locations contain fewer items [17].

10.1.2.2 *How Often Files Are Accessed?*

Strong patterns of revisitation in various types of computer use have been discussed in prior studies such as command usage [83] and email messages [48]. Zipf's law [223] is often used to characterize such patterns of behavior, with retrieval counts to distinct items following a power-law according to their rank:

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)}.$$

Where N is the number of elements, $k \in [1, N]$ is the rank of the considered element (with $k = 0$ for the element with the highest frequency) and s is the value of the exponent characterizing the distribution.

Fitchett [60] found that the frequencies of file retrievals can be approximated by Zipf's Law, suggesting that people's patterns of file retrieval are strongly repetitive, with a small number of frequently revisited files, and a large number of infrequently visited ones. As shown in Fig. 43 (a), the cross-participant mean proportion of file retrievals by ranked count of file retrievals (for the 10 most frequently visited files) for the 17 participants who retrieved more than 100 different files approximates a Zipf's law. Fig. 43 (b) shows that 60.8% of files were retrieved only once, and that 98.0% were retrieved 10 or fewer times. Half of the participants accessed at least one file more than 20 times and the maximum number of revisits was 280 times.

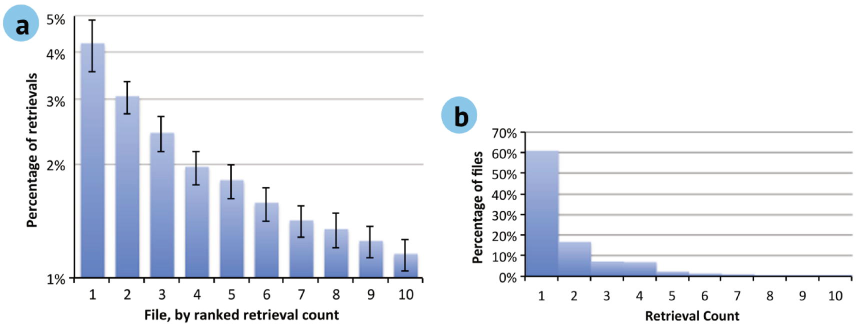


Figure 43: (a) Mean proportion of file retrievals by ranked count of file retrievals. Error bars show standard error. Y-axis uses a log scale. (b) Distribution of files' retrieval counts, across all participants. Taken from [60].

10.1.2.3 *Navigation vs. Search*

Navigation- and search-based methods are the predominant retrieval methods that can be used to retrieve a file. While search offers important benefits in certain situations, a number of prior studies have shown that most users prefer navigation to search, with search being used only as a method of last resort [15, 60, 124].

Bergman et al. offered an explanation by the relative cognitive requirements of the two approaches [15]. They pointed out that users prefer orienteering (that is, taking small steps towards a target using partial information and contextual knowledge) to teleporting (that is, jumping directly to the target) [176, 204]. Navigation uses an orienteering approach, with users able to use recognition at each step of a retrieval to identify the next folder [15]. Orienteering offers several advantages over keyword search, including decreased cognitive load, a sense of location, and a better understanding of the result [204]. Bergman et al. [15] also note that, with navigation, "users can continue to think of the project they are working on at the time", even if search might be faster.

Search interfaces, on the other hand, typically use a teleporting approach that shows an immediate list of results with little or no context [204]. Search also relies on users recalling attributes of a target file in order to devise a search query [15], which is more cognitively demanding than recognition [206]. Furthermore, search offers no reminding feature. This means that users are unlikely to encounter an item through search if they have forgotten they have it or how it is described in the file system, resulting in a lower sense of control [11]. A final potential limitation of search-based file access is that it provides minimal support for learning and rehearsing the location-based retrieval mechanics that users are likely to use for future accesses.

10.1.2.4 *Folder Uncertainty*

If users are unsure of their navigation to files, they are likely to open more folders than necessary, i. e. opening an incorrect folder and then backtracking. Elsweiler et al. [48] introduced the Folder Uncertainty Ratio (FUR) to account for this effect in email folders. FUR is defined as "the number of folders opened with respect to the number of unique folders opened".

Fitchett [60] reported that in their study, the mean FUR value for navigation retrievals was low at 1.02. The percentage of retrievals with a FUR > 1 was 5.2%, and the percentage of retrievals with FUR $\geqslant 2$ was 0.3%. These values contrast with Elsweiler et al. [48] who found high levels of uncertainty when navigating to email messages, with 29.5% of retrievals having a FUR > 1, and 8.67% having FUR $\geqslant 2$. Fitchett [60] offered a plausible explanation that users invest more effort in crafting effective file hierarchies than they do with email.

These practices and characteristics might evolve over time with the continuous improvements in search algorithms and the introduction of new interfaces and novel interaction techniques. In the next chapter, I report on a pilot study to capture real users' file structures and understand their file navigation practices, informing our simulations (Study 1) and experiment (Study 2).

10.2 ENHANCED FILE RETRIEVAL TECHNIQUES

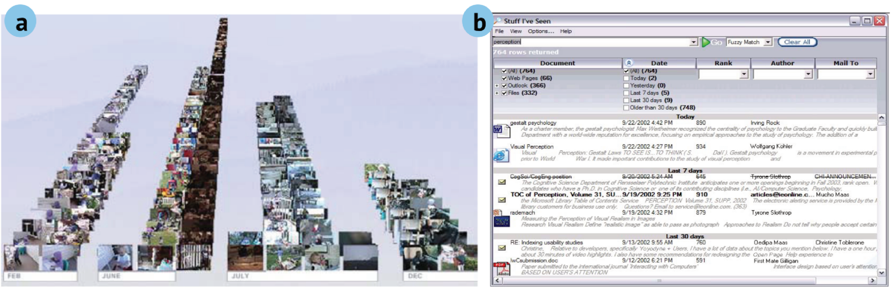
Previous studies have explored how to use visualization, search and prediction to enhance file retrieval.

10.2.1 *Visualization and Search*

Visualization techniques such as treemaps [193] and hyperbolic trees [122] (Fig. 25) have been introduced to depict hierarchical structures. However, this type of technique has not been widely adopted for file systems.

Many search-based systems have also been proposed as alternatives to hierarchical file systems, typically to address deficiencies such as requirements to maintain a folder hierarchy and to store each file in a unique location. For example, *MyLifeBits* [78] is primarily based on time attributes and is intended as a replacement for a file hierarchy. It focuses mainly on multimedia files and supports rich annotations. It uses a range of visualizations, particularly those based on time (Fig. 44 (a)).

Stuff I've Seen [45] is a search system focused on retrieving previously accessed documents. It gives particular emphasis to filtering and sorting by the date documents were modified, and integrates files, email messages and web pages into its results (Fig. 44 (b)). A prior study to the *Stuff I've Seen* system found that the inclusion of landmarks on a timeline reduced retrieval times compared to a normal timeline that only provides dates [179].



perception

764 rows returned

Document	Date	Rank	Author	Mail To
☑ (All) (764) ☑ Web Pages (66) + ☑ Outlook (366) + ☑ Files (332)	☑ (All) (764) ☐ Today (2) ☐ Yesterday (0) ☐ Last 7 days (5) ☐ Last 30 days (9) ☐ Older than 30 days (748)			
Today				
gestalt psychology	9/22/2002 4:42 PM	890	Irving Rock	
Visual Perception	9/22/2002 4:27 PM	934	Wolfgang Köhler	
Last 7 days				
~~CogSci/CogEng position~~	~~9/20/2002 5:24 AM~~	~~645~~	~~Tyrone Slothrop~~	~~CHI-ANNOUNCEMEN...~~
TOC of Perception, Volume 31, SU...	**9/19/2002 9:25 PM**	**910**	**articles@ieonline.c...**	Mucho Maas
rademach	9/19/2002 4:32 PM	879	Tyrone Slothrop	
Last 30 days				
RE: Indexing usability studies	9/13/2002 9:55 AM	760	Oedipa Maas	Christine Toblerone
IwCsubmission.doc	9/12/2002 6:21 PM	591	First Mate Gilligan	

As a charter member, the gestalt psychologist Max Wertheimer recognized the centrality of psychology to the Graduate Faculty and quickly bui
Department with a world-wide reputation for excellence, focusing on empirical approaches to the study of psychology. The addition of a

Visual Perception: Gestalt Laws TO SEE IS...TO THINK (S. Dalí). Gestalt psychology is a movement in experimental p
prior to World War I. It made important contributions to the study of visual perception and

The Cognitive Science Department of Rensselaer Polytechnic Institute anticipates one or more openings beginning in Fall 2003, rank open. W
candidates who have a Ph.D. in Cognitive Science or one of its contributing disciplines (i.e., AI/Computer Science, Psychology,

the Microsoft Library Table of Contents Service PERCEPTION Volume 31, SUPP. 2002 The electronic alerting service is provided by the
library customers for business use only. Questions? Email to service@ieonline.com. (363)

Measuring the Perception of Visual Realism in Images
Research Visual Realism Define "realistic image" as able to pass as photograph Approaches to Realism Do not tell why people accept certain

Christine, Relative to developers, specifically Yoyodyne + Users, I have a lot of data about the topics you mention below. I have a one hour
about 30 minutes of video highlights. I also have some recommendations for redesigning the Open Page Help experience to

Paper submitted to the international journal 'Interacting with Computers' Interface design based on user's attention
BASED ON USER'S ATTENTION

Figure 44: (a) Clustered-time view of query results in MyLifeBits [78] and (b) Stuff I've Seen interface with top view where the lower portion of the display shows the search results, here documents including the search query keyword *perception* [45].

Similar research systems based on search include: Lifestreams [69], TimeSpace [115], Placeless Documents [43] and Phlat [39]. All of these systems are heavily based on time attributes.

10.2.2 *Prediction*

Several prediction algorithms have been proposed to account for users' repetitive behavior and to improve the efficiency of accessing previously used items. Algorithms that predict upcoming actions based on previous actions include:

1. Most Recently Used (MRU) calculates ranks based solely on recency.

2. Most Frequently Used (MFU) calculates ranks based solely on frequency.

3. Split Recency and Frequency (SR&F) [55] selects n items with MRU, then the rest with MFU.

4. Combined Recency and Frequency (CRF) [125], used originally for cache management, considers every past access of an item. It is calculated by Equation 38, where w_f is the item's weight, n is the number of past accesses, t is the current time and t_i is the time of access i (where time is counted in terms of discrete events).

   $$w_f = \sum_{i=1}^{n} \frac{1}{p}^{\lambda(t-t_i)}. \tag{38}$$

5. The *Adaptive* algorithm filters menus in software such as Microsoft Office 2000 [6]. Item counts are incremented when selected and decremented after multiple sessions of disuse.

6. The Places Frecency algorithm (PF) is used in Firefox to order URL suggestions when typing a web address [29]. The last 10 accesses of each item are placed in time-based buckets with different weights based on recency. Other factors, such as the method of website access, are also incorporated but can be stripped out for general purpose use.

7. A Markov chain [148] can be used to make predictions:

   $$P(X_{n+1} = x | X_n = x_n) = \frac{|x_n \to x|}{|x_n|}.$$

   where $|x_n|$ is the number of previous occurrences of state x_n, and $|x_n|$ is the number of previous transitions from state x_n to x. X_i represents the state at time i. Given the most recent access x_n, the calculated probabilities provide a ranking, and MRU can be used to break ties.

8. AccessRank uses a score that blends Markov chains and CRF, a time weighting component as well as a switching threshold to predict what users will do next and to maximize list stability [59].

The AccessRank score w_n is defined as:

$$w_n = {w_{m_n}}^{\alpha} {w_{crf_n}}^{\frac{1}{\alpha}} w_{t_n}. \tag{39}$$

where w_{m_n} is the Markov weight and w_{crf_n} is the combined recency and frequency (CRF) weight with $p = 2$ and $\lambda = 0.1$ (Equation 38). The Markov weight is rewritten to always give non-zero weights:

$$w_{m_n} = \frac{|x_n \to x| + 1}{|x_n| + 1}.$$

time weight w_{t_n} gives higher weight to items that have historically been more frequently accessed at the current time of day or day of week, and is defined as:

$$w_{t_n} = \max(0.8, \min(1.25, hd))^{0.25}.$$

where h and d are defined as follows: let c_h be the current hour of the day. For item n, let h be the ratio of the number of previous accesses of n in hours in the range $[c_h - 1, c_h + 1]$ compared to the average number of previous accesses of n for a three hour slot. Similarly, let d be the ratio of the number of previous accesses of n on the current day of the week to the average across all days of the week. h and d are set to 1 if fewer than 10 accesses in total have occurred in the corresponding slot.

To improve prediction list stability, Fitchett and Cockburn [59] also defined a switching threshold: item A and item B are compared and their new weights w_A and w_B are such that $w_B > w_A$, then B will only be ranked higher than A if $w_B > w_A + \delta$ where $\delta \geqslant 0$ is an AccessRank parameter. An item C not in the previous list is assumed to have ranking $r_C = \infty$.

Using 3 log datasets from previously published studies (window switching [201], web browsing [203], and command line use [82]), Fitchett and Cockburn [59] demonstrated that AccessRank more accurately predicts upcoming accesses than Markov, CRF and MRU. Moreover, the prediction lists generated by AccessRank are more stable than the three other algorithms (Fig. 45).

Based on the results, they also recommended to use (λ, δ) values of (1.65, 0.2) to give the best compromise between accuracy and stability [59]. When stability is unimportant, values of (1.65, 0) give the best top prediction accuracy, while (2.5, 0) may be better if the average rank is the primary goal. When stability is particularly important, high values for both parameters can be used, e. g. (2.5, 0.5).

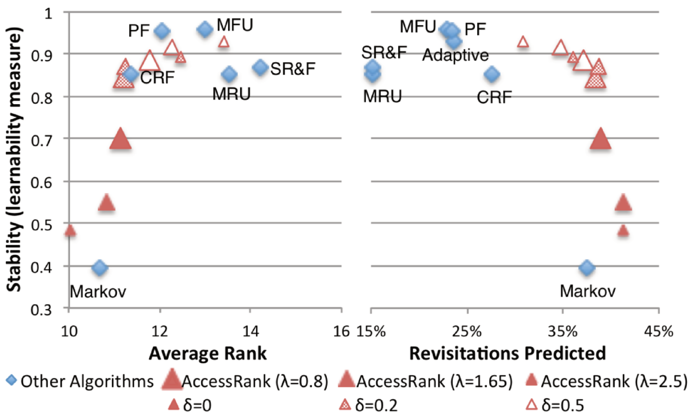


Figure 45: Prediction list stability v.s. Average Rank and Percentage Revisitations predicted over three datasets. Lower average ranks are better, while a higher percentage of revisitations predicted is better. Taken from [59].

Having shown that AccessRank is the best-of-breed prediction algorithm, Fitchett et al. further used it to improve navigation-based file retrieval [61, 62]. First, they proposed three augmented interfaces to facilitate file access [61]: Icon Highlights, Hover Menu and Search Directed Navigation (Fig. 46). Icon Highlights and Hover Menu used AccessRank with parameters $\lambda = 0.8$ and $\delta = 0.5$, and Search Directed Navigation used character-based filtering to determine candidates. For the short duration of the controlled experiment, the time weighting component of AccessRank was not used.

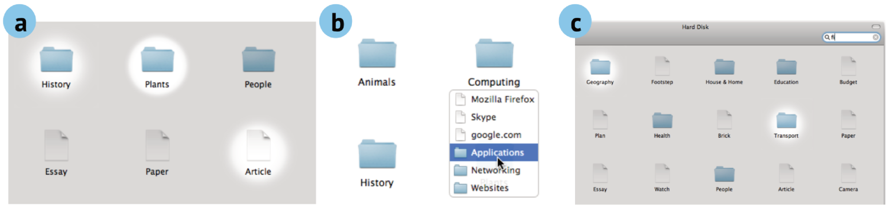
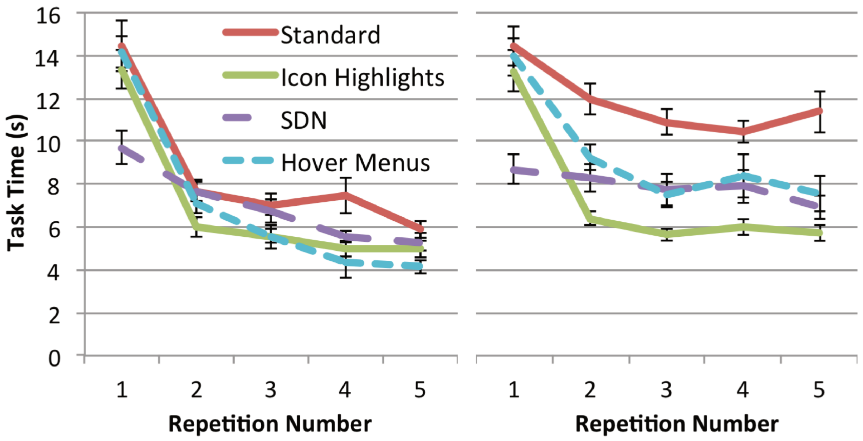

a
History
Plants
People
Essay
Paper
Article
b
Animals
Computing
Mozilla Firefox
Skype
google.com
Applications
Networking
Websites
History
c
Hard Disk
fi
Geography
Footstep
House & Home
Education
Budget
Plan
Health
Brick
Transport
Paper
Essay
Watch
People
Article
Camera

Figure 46: (a) Icon Highlights assists users in visually identifying likely targets at each level of the hierarchy; (b) Hover Menus facilitates shortcuts to likely folders and files across levels of the hierarchy; and (c) Search Directed Navigation highlights items that match search query terms. Taken from [61].

The controlled experiment indicated that these three augmented interfaces allowed faster task completion than the standard file browser in both spatially stable icons condition (Fig. 47 (a)) and spatially unstable icons condition (Fig. 47 (b)). Participants also favored the augmented interfaces. However, Hover Menus, while preferred to the standard browser, was rated as the most mentally demanding of the three augmentations. Therefore, in a later longitudinal field evaluation of Finder Highlights [62], only Icon Highlights and Search Directed Navigation were included.



(a) Spatially stable icons (b) Maximally unstable icons

Figure 47: Task times by repetition number in [61]. The task was to navigate within a three-level semantically organized hierarchy to select a target file that was cued by displaying its name in a small window at the side of the file browser interface. Repetition number indicates how many times the same item has been accessed.

Since Fitchett and Cockburn [59] have shown that AccessRank is the best-of-breed prediction algorithm, in the next chapter, I compare the BIGFile algorithm with AccessRank. In our controlled experiment to investigate BIGFile's performance, we replicate and extend the methodology used by Fitchett et al. [61]. BIGFile also uses a split adaptive interface, therefore I now review prior studies on this topic.

10.3 ADAPTIVE USER INTERFACES

Adaptive user interfaces (AUIs) are a class of interfaces that adapt the presentation of functionality automatically, in response to individual user behavior or context. Research results on adaptive user interfaces are mixed. On the one hand, AUIs can lead to better user performance and satisfaction:

- Findlater et al. introduced *Ephemeral Adaptation* where the predicted items appear first, while remaining items gradually fade in to reduce menu visual search time while maintaining spatial consistency [58] (Fig. 48). With two controlled laboratory studies, they showed that ephemeral adaptation results in both performance and user satisfaction benefits over a static control condition when adaptive accuracy is high.

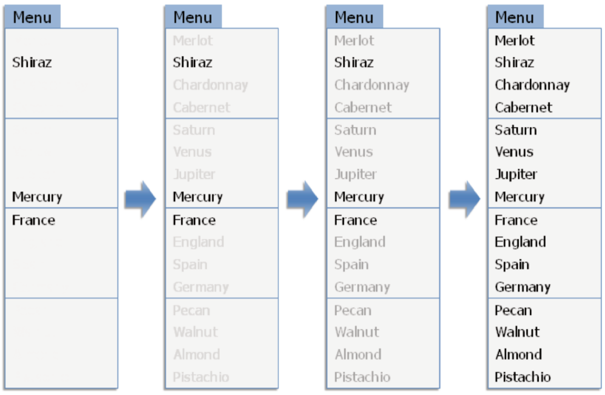
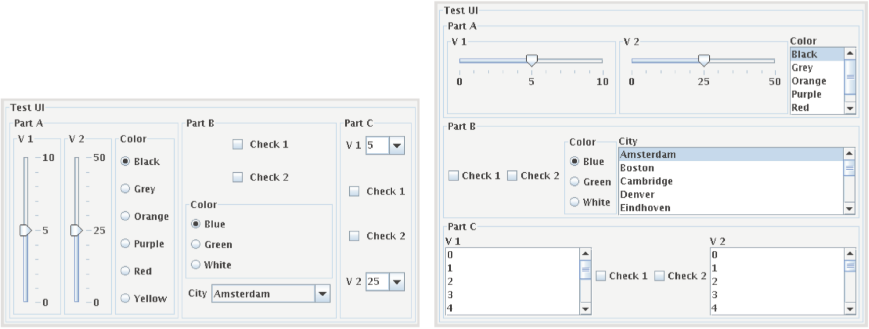

Menu
Merlot
Shiraz
Chardonnay
Cabernet
Saturn
Venus
Jupiter
Mercury
France
England
Spain
Germany
Pecan
Walnut
Almond
Pistachio

Figure 48: Ephemeral adaptation applied to menus: predicted items appear immediately, while remaining items gradually fade in. Taken from [58].

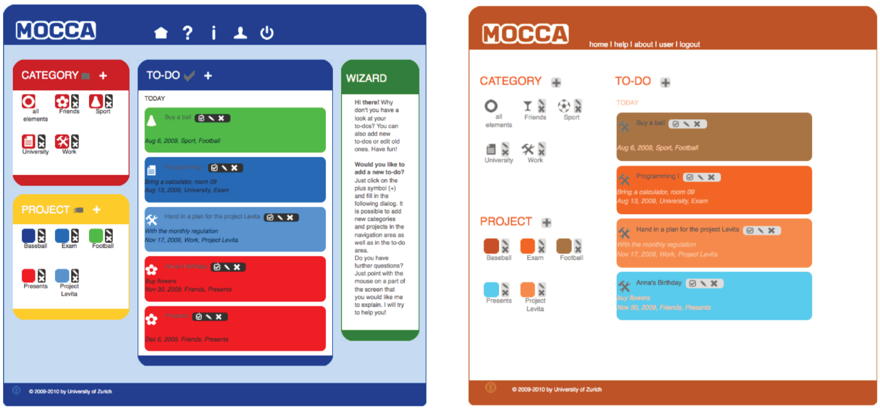
- Gajos et al. introduced two systems for automatically generating personalized interfaces adapted to the individual motor capabilities of users with motor impairments [73]. The first system, *Supple* uses a model of the user's preferences and the second system, *Supple*++, models a user's motor abilities directly from a set of one-time motor performance tests. Both systems were shown to be faster than the traditional interface and were preferred by the participants (Fig. 49).


Test UI
Part A
V 1
V 2
Color
Black
Grey
Orange
Purple
Red
Yellow
Part B
Check 1
Check 2
Blue
Green
White
City
Amsterdam
Boston
Cambridge
Denver
Eindhoven
Part C

baseline AB03

Figure 49: The baseline interface is shown in comparison to interfaces generated automatically by *SUPPLE* based on two participants' preferences. Participant AB03 preferred lists to combo boxes but preferred them to be short. Taken from [73].

- Reinecke and Bernstein proposed to design culturally adaptive systems, which automatically generate personalized interfaces that correspond to cultural preferences [177]. By incorporating participants' cultural background, they found that a majority of international participants preferred their personalized versions over a non-adapted interface of the same web site MOCCA (Fig. 50).


MOCCA
CATEGORY +
all elements
Friends
Sport
University
Work
PROJECT +
Baseball
Exam
Football
Presents
Project Levita
TO-DO +
TODAY
WIZARD
© 2009-2010 by University of Zurich
MOCCA
home I help I about I user I logout
CATEGORY
all elements
Friends
Sport
University
Work
PROJECT
Baseball
Exam
Football
Presents
Project Levita
TO-DO
TODAY
© 2009-2010 by University of Zurich

Buy a ball
Aug 6, 2009, Sport, Football

Programming I
Bring a calculator, room 09
Aug 13, 2009, University, Exam

Hand in a plan for the project Levita
With the monthly regulation
Nov 17, 2009, Work, Project Levita

Anna's Birthday
Buy flowers
Nov 30, 2009, Friends, Presents

Presents
Dec 6, 2009, Friends, Presents

Hi there! Why don't you have a look at your to-dos? You can also add new to-dos or edit old ones. Have fun!

Would you like to add a new to-do? Just click on the plus symbol (+) and fill in the following dialog. It is possible to add new categories and projects in the navigation area as well as in the to-do area.
Do you have further questions? Just point with the mouse on a part of the screen that you would like me to explain. I will try to help you!

Buy a ball
Aug 6, 2009, Sport, Football

Programming I
Bring a calculator, room 09
Aug 13, 2009, University, Exam

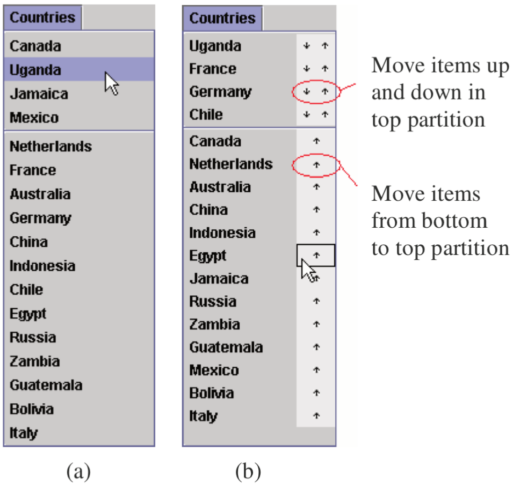
Hand in a plan for the project Levita
With the monthly regulation
Nov 17, 2009, Work, Project Levita

Anna's Birthday
buy flowers
Nov 30, 2009, Friends, Presents

Figure 50: Left: adapted interface for a participant who was from Mexico, and had shortly lived in Bulgaria before coming to Switzerland; Right: the interface was personalized for an Indian participant, who had lived in France and the US for several years. Taken from [177].

On the other hand, when poorly designed, AUIs can also lead to the user being confused and feeling a loss of control over the interface:

- Findlater and McGrenere [55] compared three interfaces: *Static*, where the interface does not change during the course of use; *Adaptive*, where the system controls changes and *Adaptable*, where the user controls changes. Applied to split menus [184] (Fig. 51), they found that optimal static split menus are significantly faster than adaptive menus and users preferred the adaptable interface to the other two because they can customize the interface according to their own needs and they do not need to adapt to the system whenever it makes changes.


Countries
Canada
Uganda
Jamaica
Mexico
Netherlands
France
Australia
Germany
China
Indonesia
Chile
Egypt
Russia
Zambia
Guatemala
Bolivia
Italy
Countries
Uganda
France
Germany
Chile
Canada
Netherlands
Australia
China
Indonesia
Egypt
Jamaica
Russia
Zambia
Guatemala
Mexico
Bolivia
Italy
Move items up and down in top partition
Move items from bottom to top partition

(a) (b)

Figure 51: (a) Static split menu and (b) Adaptable split menu. Taken from [55].

- Gajos et al. [74] designed and implemented 3 adaptive graphical interfaces: *Split Interface*, which provides an additional toolbar that copies important functions onto this toolbar in a spatially stable manner; *Moving Interface*, which moves promoted functionality from inside popup panes onto the main toolbar; and *Visual Popout Interface*, which highlights promoted buttons in magenta (Fig. 52). With two experiments comparing them with non-adaptive interface, Gajos et al. found that while *Split Interface* has perceived high benefit and low cost, *Moving Interface* and *Visual Popout Interface*, where the perceived cost exceeded the benefit, were rejected by the participants.

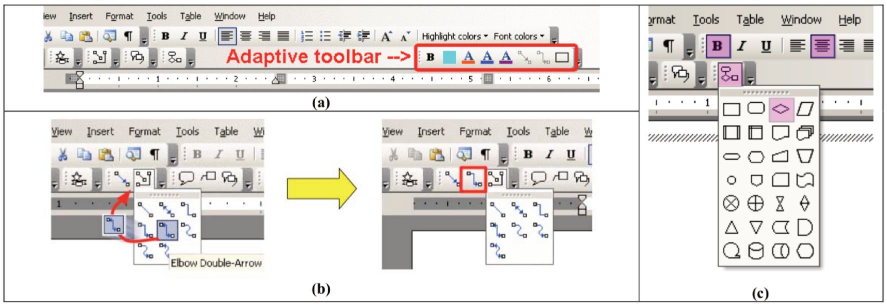


Figure 52: (a) The Split Interface; (b) The Moving Interface and (c) The Visual Popout Interface. Taken from [74].

Split interfaces, which are based on Split Menus [184], are a type of AUI that are particularly effective [56, 57, 74]. A split interface typically has two parts: a static part that represents the "status quo" original interface, and an adaptive part that augments the static part. The adaptive part changes its contents based on what the system believes the user needs. Users have the choice between interacting with the static part or the adaptive part. Thus, the learnability of the original interface is not hindered, and user performance can be enhanced if users take advantage of the adaptive part. Most split interfaces to date have been designed for menu selection [58], but other interface elements have also been split, such as the toolbar [74], emails [27], and relevant documents on Google Drive [202]. BIGFile introduces split interfaces to hierarchical file systems.

11

HIERARCHICAL FILE RETRIEVAL: BIGFILE

In this chapter, I introduce BIGFile, a fast file retrieval technique based on the BIG framework. BIGFile provides interface shortcuts to assist the user in navigating to a desired target (file or folder). BIGFile's split interface combines a traditional lisst view with an adaptive area that displays shortcuts to the set of file paths estimated by a computationally efficient algorithm based on the Bayesian Information Gain framework. Users can navigate the list as usual, or select any part of the paths in the adaptive area. I first describe each variable in BIGFile, demonstrate the BIGFileOpt and BIGFileFast algorithms, and report on two studies: a simulation study comparing BIGFileFast with AccessRank, and a controlled experiment comparing BIGFile with ARFile (AccessRank instantiated in a split interface) and with a Finder-like list view as baseline.

The work presented in this chapter is published at ACM CHI 2018: BIGFile: Bayesian Information Gain for Fast File Retrieval. Wanyu Liu (WL), Olivier Rioul (OR), Joanna McGrenere (JM), Wendy E. Mackay (WM), and Michel Beaudouin-Lafon (MBL). (Best Paper Honorable Mention Award)

Conceptualization: WL
Design: WL, JM, WM, MBL
Implementation: WL
Experimental design: WL, JM, WM, MBL
Evaluation: WL
Data analysis: WL
First draft: WL
Paper writing: WL, JM, MBL, OR
Supervision: OR, MBL

11.1 BIGFILE INTERFACE

BIGFile improves navigation-based file retrieval efficiency by providing *shortcuts* that can reduce the number of steps (user inputs) to reach the target file or folder: the user can skip levels in the hierarchy by selecting one of the shortcuts, similar to Hover Menu [61]. I first present the BIGFile interface before describing the algorithm that determines the shortcuts.

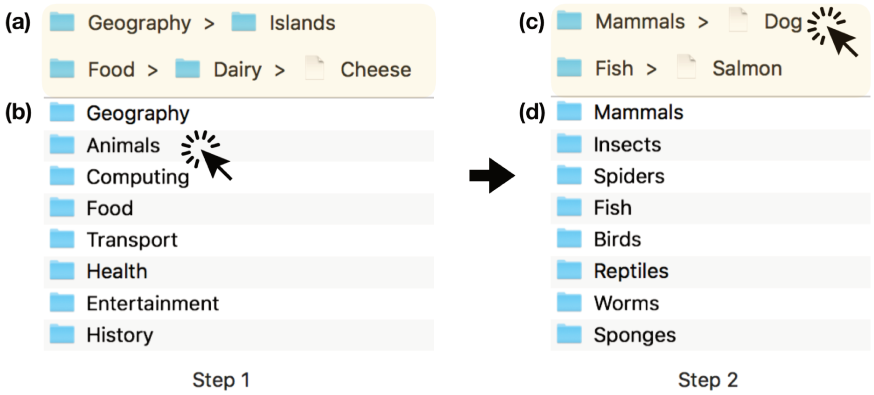


Figure 53: The BIGFile interface as the user navigates to "Dog" in a file retrieval task. (a) and (c) show the adaptive part with two shortcuts, (b) and (d) the static part. In Step 1, the shortcuts do not help and the user selects "Animals" in the static part, leading to Step 2 where the user directly selects "Dog" in the first shortcut, saving one navigation step.

BIGFile features a split adaptive interface (Fig. 53): the shortcuts are presented in the *adaptive area* at the top, while the *static area* at the bottom is a traditional list view of the current folder. The shortcuts in the adaptive area are the paths to the items selected by the BIGFile algorithm, relative to the current folder. Displaying the relative paths, rather than just the items, offers users contextual information that helps them determine if they correspond to the target they are looking for. It also lets users navigate directly to any folder in the path by clicking on it, typically when the target is not in the shortcuts, but a partial path to it is. Finally a back button (visible in Fig. 57) lets users go back to the previous state of the interface.

Both the shortcuts in the adaptive area and the items in the static area are updated after each user input. Similar to many other split adaptive interfaces [55, 184], if the system correctly estimates the user's target item, the user can select the shortcut, or navigate the hierarchy using the static part as usual. If none of the system's estimates are correct, the impact for the user is minimal since the items remain at their usual locations in the static part of the interface.

For example, in Fig. 53 (left), "Islands" and "Cheese" are the estimated items, presented along with their paths in the designated adaptive area (a). The static area (b) presents the usual hierarchy. A user could, for example, click on "Dairy" to access dairy products other than "Cheese" inside the folder (not shown in the figure). If the user clicks on "Animals", the static area is updated, showing the items inside the "Animals" folder (Fig. 53 (d)). The adaptive area is also updated with a new set of estimated targets ("Dog" and "Salmon", Fig. 53 (c)). If the user is looking for the item "Dog", she can save one step ("Mammals") by clicking the shortcut in the adaptive area. The number of shortcuts is user-customizable.

We created and considered a number of alternative designs for the interface, including an integrated view where each shortcut is displayed, together with its path, next to the corresponding root folder in the list view. However, we found that this integrated view makes it difficult to display shortcuts of arbitrary depth. Moreover, scrolling the view often hides shortcuts, which partially defeats their purpose. In addition, this design only works for the list view, while the split interface can work with any of the traditional views in the static area, e. g. the icon and column views of the Mac OS Finder. Therefore, we chose what seemed to be the simplest and most obvious option for our first implementation and comparison. Note that the split interface design is not specific to BIGFile and can be used with any algorithm that predicts potential targets. For example, we used it with the AccessRank algorithm in Study two, described later this chapter.

11.2 BIGFILE ALGORITHM

I introduce two algorithms for BIGFile: BIGFileOpt, an optimal but computationally costly algorithm and BIGFileFast, a suboptimal but very efficient version, which is used for both the simulations and experiment.

11.2.1 *BIGFileOpt: Optimal Algorithm*

In order to apply the Bayesian Information Gain (BIG) framework to file retrieval, let us consider a regular hierarchical file system. Without loss of generality, we consider a single window, with a current folder F. We define the following:

- Θ represents all the folders and files that a user might be interested in. Θ may include all the files and folders in the file system, but is more likely to be narrowed to a subset based on user preference or the task at hand. For example, it can be reduced to a subset of the user's home folder and/or to a category of files such as documents of a certain type. In the simulations and the experiment, we used only the files as potential targets and excluded the folders.

- For each potential target $\Theta = \theta$, the initial probability, at the beginning of a retrieval task, that it is the actual intended target is $P(\Theta = \theta)$. This probability distribution is calculated using the Combined Recency and Frequency (CRF) algorithm (Equation 38) using $\{p = 2, \lambda = 0.1\}$, in AccessRank [59]. The probability that a file θ is the target is calculated by normalizing its weight: $P(\Theta = \theta) = w_\theta / \sum w_\theta$ and is updated after each retrieval of a target by the user, to reflect interaction history. At each step of the retrieval task, i.e. after each user input, $P(\Theta = \theta)$ is updated using Bayes' rule, as described in Algorithm 1.

- X represents the view generated by the system when first opening a window and after receiving each user input in that window. This view is composed of the *static part* S, which shows the folders and files of the current folder F, and the *adaptive part* A, which shows the N folders and files that are produced by the BIGFile algorithm to serve as shortcuts at this step. A view $X = x$ is therefore represented by $S \bigcup A$. The number of shortcuts N is user customizable.

- Y represents any user input. At each step, the user issues an input $Y = y$ to the system: the user can select any of the items in the static and adaptive parts, or go back to the previous view with the back button in case of an error.

Algorithmus 1 : BIGFileOpt
Search the optimal set of N shortcuts.

```
Data : $\Theta, X, Y, P(Y = y|\Theta = \theta, X = x), IG_{max} = 0$
Result : Return set $A$ that, together with set $S$, has the maximal
         expected information gain (IG).
Receive user input $Y = y$
Update the probability distribution of $\Theta$ (Bayes rule):
  $P(\Theta = \theta|X = x, Y = y) = \frac{P(Y=y|\Theta=\theta,X=x)P(\Theta=\theta)}{P(Y=y|X=x)}$
for all the combinations $A$ of $N$ nodes that are below the current
  folder $F$ in the hierarchy do
    Compute $IG(S \bigcup A) = I(\Theta; Y|X(S \bigcup A))$
                             $= H(\Theta) - H(\Theta|X(S \bigcup A), Y)$
    // I is mutual information and H is entropy
    if $IG(S \bigcup A) > IG_{max}$ then
        $IG_{max} = IG(S \bigcup A)$
        $A_{max} = A$
return $A_{max}$
```

- $P(Y = y|\Theta = \theta, X = x)$ represents prior knowledge about user behavior: given view x and target θ, what is the probability that user input is y at this step. For simplicity, one can assume that the user does not make mistakes and therefore that this probability is 1 if the user is issuing the correct input, 0 otherwise. Alternatively, one can use a calibration session, as in the BIGnav experiment (Chapter 9). Note that since the user may select items that are not in Θ during the steps that lead to a selection, user behavior must be known for any item in the file system.

At each step, i. e. after each user input, the static part S of the interface is updated if the current folder has changed, i. e. if the user has clicked on a folder to navigate to it. Then the adaptive part A of the interface is updated to display the N items selected by the BIGFile algorithm.

Algorithm 1 presents BIGFileOpt, an optimal algorithm that finds the N items that, together with S, maximize the expected information gain from the user's next input. This slight modification of the original BIG framework lets us calculate an optimal view $S \bigcup A$. However, considering the sizes of typical personal file systems, this algorithm is not practical: the number of sets to test grows like f^N, where f is the number of files and folders and N the number of items in the adaptive part.

I now present a suboptimal but computationally efficient algorithm to address this problem.

11.2.2 *BIGFileFast: Efficient Algorithm*

We take advantage of the hierarchical structure of the file system to select a set A' of targets that, together with S, has one of the highest, if not the highest, expected information gain. At each step, BIGFileFast creates a tree whose leaves are the n targets ($n > N$) with highest probability as follows:

First, the tree contains the n targets and their parent nodes up to the root (Fig. 54 (b)); Then BIGFileFast compresses this tree by replacing nodes that have a single child with that child (Fig. 54 (c)). This much smaller tree vastly reduces the number of sets of N targets to try. For $N = 4$, we found that $n = 6$ gave good results; $n = 10$ gave slightly better results, but was too slow for large hierarchies. Note that since the set is recomputed after each user input, it adapts dynamically to the user's navigation.

We further optimize the search as follows: Consider a candidate set $A = a_0, a_1, ...a_N$ and an item a_i of A with a child a'_i. Let A' be the set where a_i is replaced by its child a'_i: $A_1 = a_0, ..., a_{i-1}, a'_i, a_{i+1}, ..., a_N$. If the expected information gain for the set A' is lower than that of A, then we do not consider any set with an item in the subtree of a'_i, effectively pruning that subtree. Fig. 54 and Algorithm 2 describe the implementation of BIGFileFast used in the simulations and the experiment later in this chapter.

BIGFileFast dramatically reduces search time, making it real time, and selects sets of targets with near-optimal expected information gain. We ran simulations comparing it to BIGFileOpt and found that, for example, for 1000 targets and a 12-level hierarchy, BIGFileOpt takes roughly three minutes while BIGFileFast responds in interactive time. Also, on average, the expected information gain of BIGFileFast was 84.7% that of BIGFileOpt.

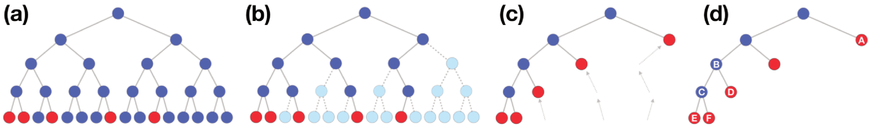


Figure 54: BIGFileFast with a binary tree: (a) Find the $n = 5$ most probable targets (red nodes); (b) Find their parents up to the root (dark blue nodes) and prune all the non-parent nodes (light blue nodes); (c) recursively replace the parent of a node by its child if it is the only child; (d) if the expected information gain of (A,B) is greater than that of (A,C), prune branch C and move directly to the next branch (A,D), skipping E and F.

Algorithmus 2 : BIGFileFast
Efficiently search a suboptimal set of shortcuts.

```
Result : Return set $A$ that, together with set $S$, has a suboptimal
         maximal expected information gain.
// Create the compressed tree, i.e.
The minimal subtree of the current folder $F$ that contains the $n$
  most probable targets:
for each element $t$ of the tree do
    if $t$ is the only child of its parent then
        Replace the parent by $t$ and remove the parent
// Search this simplified tree
$A$ = a set of $N$ nodes such that no node is in the subtree of
  another node
$IG_{max} = IG(S \bigcup A)$
while there are more sets to explore do
    if Node $a_i \in A$ has a child $a_i'$ not yet explored then
        $A' = A - a_i + a_i'$
        Compute $IG'(S \bigcup A')$
        if $IG(S \bigcup A) > IG'(S \bigcup A')$ then
            Skip the subtree rooted at $a_i'$
        else
            $IG(S \bigcup A) = IG'(S \bigcup A')$
            $A = A'$
    else
        $A = A - a_i +$ the root of the next branch
        Compute $IG(S \bigcup A)$
    if $IG(S \bigcup A) > IG_{max}$ then
        $IG_{max} = IG(S \bigcup A)$
        $A_{max} = A$
return $A_{max}$
```

11.3 PILOT STUDY

We conducted a pilot study to capture real users' file structures and understand their file navigation practices, informing our simulations (Study 1) and experiment (Study 2). We wanted to see if and how the structures and practices reported in the literature [16, 96] have changed.

We recruited 15 participants from our institution, including faculty members, post-docs and students, all in technical areas. 13 were MacOS users, 2 were Windows users. We wanted to know the depth and breadth of their file systems, their navigation strategies, their preferred view for retrieving files, and the problems they run into.

Participants filled out a questionnaire, ran a script to get summary data of their file structures on their primary computer, and then reflected on their own file retrieval behavior. To run the script, participants identified the folder or folders that contain(s) the files that they navigate routinely with the Finder or File Explorer, such as the Documents and Desktop folders, but not the Music folder. The script counts the number of files and folders and returns a table with the file structure information and a graph visualizing the hierarchy (Fig. 55).

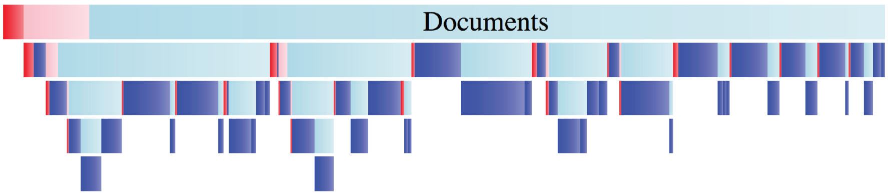


Figure 55: Visualization of P12's Documents folder, a 5-level tree. Each folder is represented by four bars: number of subfolders (branching factor) in red; number of files (folder size) in dark blue; total number of folders in subtree in pink; total number of files in subtree in pale blue.

File Structures. Our findings differ somewhat from previous studies that found that people's file hierarchies tend to be shallow and broad (small depth and large branching factor), and have small and well defined folders [16, 96]. We found the average depth to be 7.7 ($\min = 5, \max = 10, \sigma = 1.18$), and interviews with participants confirmed that they do regularly navigate to deeper levels to access a file or folder. The average branching factor was 5.62 ($\min = 2.8, \max = 10.7, \sigma = 1.95$) and the average folder size 8.2 ($\min = 3.8, \max = 14.6, \sigma = 3.37$), which are relatively smaller than the findings in [16, 96], which found an average branching factor of 10 and an average folder size of 11. We also found that, in general, relatively fewer folders and files are nested at deep levels, suggesting that people do not build extremely complex file structures.

Navigation Strategy. Ten participants reported that they first use Finder (or equivalent file navigator) to locate a file. Five reported using Spotlight (or equivalent search tool) first. This supports previous studies [15, 62] reporting that people usually navigate the hierarchical structure to locate a folder or a file, and use search as a "last resort" if navigation does not work.

Most-used View. Six participants preferred List view, four preferred Column view, three preferred both List view and Column view and regularly switch between them. Only two participants preferred Icon view, although many participants mentioned that they use Icon view to preview images.

Navigation Difficulty. Participants reported two main challenges when retrieving a file: (1) files/folders with repeated names, and (2) having partial knowledge about the location or the name of the target. This echoes previous studies [96, 145] reporting that people face retrieval difficulties stemming from semantic ambiguity. Hence, contextual information is crucial for a successful file retrieval. We used the findings from this pilot study to inform the designs of the simulations and experiment.

11.4 STUDY 1: SIMULATIONS

I ran simulations to investigate BIGFileFast's performance in estimating the target in a given hierarchical structure. The goal was to know how well the algorithm performs with respect to the following factors:

1. *Depth and width*: Both previous studies [79, 96] and our pilot study show that users have different file structures. Combining the results from [79, 96] and our pilot study, we used **DEPTH** $= \{4, 6, 8, 10, 12\}$ and **BRANCHING FACTOR** $= \{2, 4, 6, 8\}$ for the simulations.

2. *Initial distribution*: We did not log participants' use of their file system in the pilot study, but previous work indicates that file system use approximately follows a Zipf distribution [60]. To simulate different types of use history, we used two **DISTRIBUTION**s: $Z(s = 1)$ and $Z(s = 2)$. The latter is a more skewed distribution describing cases where users focus primarily on a small set of targets.

3. *Size of target set*: Both previous work [79, 96] and the pilot study suggest that users have different numbers of files and folders in their file system. Therefore, we used different target set sizes to see how BIGFileFast would perform. In our simulations, **TARGET SET SIZE** $= \{10, 100, 1000\}$.

We used AccessRank as baseline as it outperforms existing prediction algorithms [59]. In the case of navigation-based file retrieval, AccessRank predicts the target by assuming that a subfolder is likely to be selected if its parent folder is selected, captured by the Markov chain model. Similarly, BIGFileFast also assumes that the target is within the subtree of the current folder, and renormalizes the probabilities at each step. The key differences between AccessRank and BIGFileFast are as follows:

- AccessRank assigns a score to all folders and files while BIGFileFast only considers the set of potential targets.

- AccessRank updates the score of an item (file or folder) once it has been clicked while BIGFileFast updates the probability of all potential targets after each user input.
- AccessRank identifies the N items with highest scores while BIGFileFast identifies N items that provide the maximally informative view.
- AccessRank has a parameter δ to control the stability of the prediction list; BIGFileFast does not.

11.4.1 *Simulation Settings*

We generated a number of symmetric hierarchical structures crossing **DEPTH** with **BRANCHING FACTOR** = 2 and **BRANCHING FACTOR** with **DEPTH** = 4. When needed, extra targets were added at the deepest level so that there would be 100 and 1000 targets respectively. Depending on the target set sizes, we constructed a series of selections following the Zipf distributions. We randomized the mapping between the Zipf distribution and the targets, as well as the order of the selections.

We logged the number of steps needed to locate the target, the information gain and the accuracy rate for both algorithms. Note that we consider the folders on the path to the final target to be partially correct. For example, if the target is at level L_2 but the shortcut is only correct up to the folder at level $L_1 < L_2$, we consider the accuracy rate to be L_1/L_2, no matter how many steps it takes to get to the target level L_2.

We used $\{\alpha = 0.8, \delta = 0.5\}$ for AccessRank as in [61] and $\{p = 2, \lambda = 0.1\}$ in CRF for both AccessRank and BIGFileFast. We also assumed 100% correct user behavior for all simulations, i. e. that users would be as efficient as possible, always selecting an item from the adaptive area if it would get them to the target in fewer steps. Each condition [**DEPTH** × **BRANCHING FACTOR** × **TARGET SET SIZE** × **DISTRIBUTION**] was run 100 times, and the average taken.

11.4.2 *Simulation Results*

Fig. 56 shows the number of steps and the accuracy rate for the two algorithms using a $Z(s = 1)$ distribution. The results for $Z(s = 2)$ distribution are very similar; both BIGFileFast and AccessRank performed slightly better than they did with the $Z(s = 1)$ distribution. This is intuitive since both algorithms are based on frequency and recency of the file system use, and $Z(s = 2)$ focuses on a small set of very frequent items. In information-theoretic terms, the computer starts with more knowledge (less uncertainty) about the user's goal.

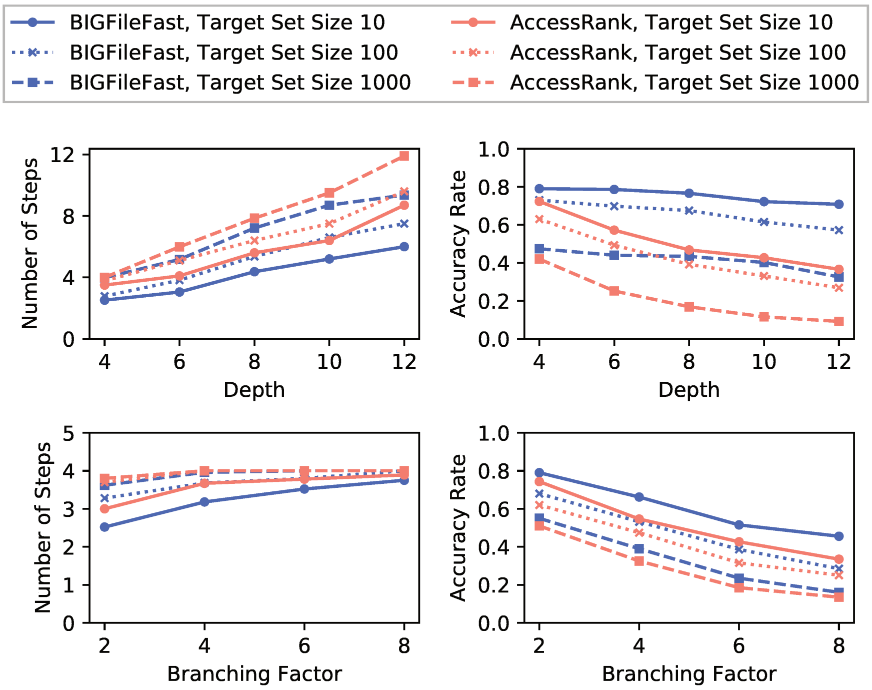


Figure 56: Simulations of BIGFileFast v.s. AccessRank for a Zipf distribution ($s = 1$) and three target set sizes (10, 100, 1000). The plots show the number of steps (left, lower is better) and accuracy rates (right, higher is better) as a function of depth (top) and branching factor (bottom).

In general, BIGFileFast takes fewer steps to locate the target than AccessRank (Fig. 56, top left). In particular, the deeper the target is located, the better BIGFileFast performs: when **DEPTH** = 4, BIGFileFast averages 3.1 steps v.s. 3.8 for AccessRank; when **DEPTH** = 12, BIGFileFast averages 7.6 steps v.s. 10.1 for AccessRank. Increasing depth decreases the accuracy rate for both BIGFileFast and AccessRank, but the effect is more pronounced for AccessRank as shown in Fig. 56, top right: when **DEPTH** = 4, BIGFileFast is 66.5% accurate on average v.s. 62.4% for AccessRank; when **DEPTH** = 12, BIGFileFast is 53.5% accurate v.s. 24.2% for AccessRank.

This can be explained by the fact that AccessRank assigns a score to all files and folders; hence, the more levels that are traversed to get to a target, the more folders that are not targets themselves (but are on the way to the targets) have their scores increased. Another factor is that AccessRank takes user input into account for the next retrieval, but not within a retrieval task. If a node is shown but is not chosen, it will show up again at the next step as a prediction if it has a relatively high score and the final target is in the same parent folder. This results in wasting a prediction slot and not gaining information from the user.

By comparison, BIGFileFast considers each user input within a retrieval, and since it assumes correct user behavior, if a node is shown and not chosen, all potential targets inside that node will be assigned a probability of 0. Therefore, the whole branch starting from that node will be discarded, i. e. it will not show up in the prediction slots at the next step.

Increasing the branching factor negatively affects both BIGFileFast and AccessRank (Fig. 56, bottom). The accuracy rate of BIGFileFast drops from 66.5% to 30.1% while the accuracy of AccessRank drops from 62.4% to 24.3%. This is not surprising as there is not much information from the user input for a wide but shallow (**DEPTH** $= 4$) hierarchy. Increasing target set size also negatively affects the performance of both BIGFileFast and AccessRank.

Averaged across all simulations, BIGFileFast is 15.5% more accurate and takes 23.1% fewer steps than AccessRank. The results can be summarized as follows: The deeper the target is located, the better BIGFileFast is than AccessRank; Increasing either target set size or branching factor negatively affect the performance of both BIGFileFast and AccessRank; and BIGFileFast performs better on a deep hierarchy than on a broad hierarchy. This echoes the results of BIGnav, which exhibits better results on navigation tasks with higher IDs, i. e. on harder tasks. We next compare BIGFile (which uses BIGFileFast) with a split interface using AccessRank in an experiment with real users.

11.5 STUDY 2: EXPERIMENT

I conducted an experiment to investigate the effectiveness of BIGFile with users. The goal was to replicate and extend the methodology used by Fitchett et al. [61]. We used their implementation of the algorithm with the exception of one improvement which is noted below. We also used their hierarchical structure, which is a 3-level semantically organized hierarchy.

Since the pilot study showed that people do navigate to deep levels, we extended their structure to 6 levels using the branching factors and folder sizes from Bergman [16]: 10, 5, and 4 folders, and 11, 8 and 7 files at levels 4, 5 and 6 respectively. Example targets for level 3 include ‘Dog’ with the path “Animals > Mammals > Dog” and ‘Darwin’ with the path “People > Inventors/Scientists > Darwin”. Example targets for level 6 include ‘Hawaii’ with the path “Geography > Islands > Tropical > Touristic > Large > Hawaii”, and ‘Brie’ with the path “Food > Dairy > Cheese > France > Creamy > Brie”. As in [61], only the folders containing the final target are populated. In total, the hierarchy contains 958 folders and 1068 files, of which 30 files are chosen as targets for each level-3 and level-6 condition.

11.5.1 *Method*

We used a [3×2] within-subject design with 3 **INTERFACE** conditions: BIGFile with the BIGFileFast algorithm, ARFile, a split interface using AccessRank for prediction, and a standard Finder interface as baseline; and 2 target **LEVEL**s: 3 and 6.

We made a slight modification to AccessRank in order to make ARFile as effective as possible for users. In AccessRank, each folder and file is assigned a score. If users constantly go to the same item (file or folder), the algorithm's set of top predictions might include both the item and its parent folder. Since we are showing the full paths to the predicted items (not just the items themselves), this would result in an overlap between the shortcuts. Therefore we only show the deepest path if one shortcut is a prefix of another.

To model the user behavior, we used the notion of Folder Uncertainty Ratio [48], which was used by Fitchett & Cockburn [60] to illustrate users' uncertainty when navigating to files. If users are uncertain that they are going down the correct path, they are likely to select incorrect folders by mistake. Fitchett and Cockburn [60] found that users were accurate about 94% of the time, while the other 6% of the time, they clicked on the wrong folder. Thus, we set the rate of correct user input to 94% and divided the remaining 6% among the other user inputs. These rates were used in the user behavior function in BIGFileFast and for calculating information gain in ARFile and in Finder. Furthermore, as in our simulations, we used $\{\alpha = 0.8, \delta = 0.5\}$ for AccessRank as in [61] and $\{p = 2, \lambda = 0.1\}$ for CRF for both AccessRank and BIGFileFast. A list view was used for the static part in all interface conditions because it was preferred in our pilot study.

11.5.2 *Participants*

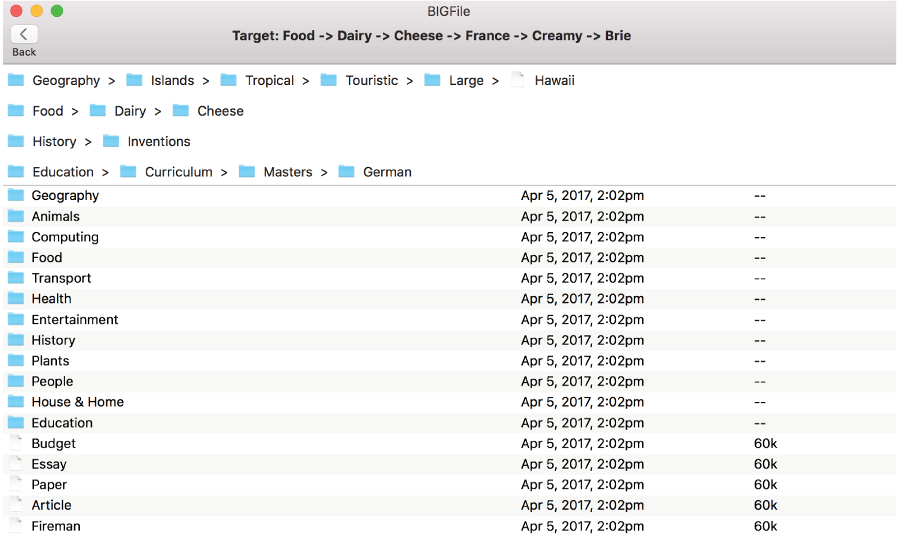
Eighteen participants (7 women), aged 21 to 39 (mean = 28.5, σ = 5.1), all right-handed and with normal or corrected-to-normal vision, volunteered to participate in the experiment. Ten were MacOS users, eight were Windows users but were familiar with list view.

11.5.3 *Apparatus*

The experiment was conducted on a Macbook Pro with a 2.7 GHz processor, 8 GB RAM with resolution of 2560×1600. The file browser window was 880 × 631 pixels, as in [61]. One row on the list view takes 20 pixels. The software was implemented in Swift 3.0. The code can be found at `https://github.com/wanyuliu/BIGFile`.

11.5.4 *Procedure*

The experiment consisted of two parts: practice, where participants familiarized themselves with the split interface using a training file hierarchy, and retrieval, where participants completed a series of file retrievals following a stimulus, which was presented as a path to the final target, e. g. "Food > Dairy > Cheese > France > Creamy > Brie", as shown in Fig. 57.



Target: Food -> Dairy -> Cheese -> France -> Creamy -> Brie

Geography > Islands > Tropical > Touristic > Large > Hawaii
Food > Dairy > Cheese
History > Inventions
Education > Curriculum > Masters > German

Geography	Apr 5, 2017, 2:02pm	--
Animals	Apr 5, 2017, 2:02pm	--
Computing	Apr 5, 2017, 2:02pm	--
Food	Apr 5, 2017, 2:02pm	--
Transport	Apr 5, 2017, 2:02pm	--
Health	Apr 5, 2017, 2:02pm	--
Entertainment	Apr 5, 2017, 2:02pm	--
History	Apr 5, 2017, 2:02pm	--
Plants	Apr 5, 2017, 2:02pm	--
People	Apr 5, 2017, 2:02pm	--
House & Home	Apr 5, 2017, 2:02pm	--
Education	Apr 5, 2017, 2:02pm	--
Budget	Apr 5, 2017, 2:02pm	60k
Essay	Apr 5, 2017, 2:02pm	60k
Paper	Apr 5, 2017, 2:02pm	60k
Article	Apr 5, 2017, 2:02pm	60k
Fireman	Apr 5, 2017, 2:02pm	60k

Figure 57: BIGFile experimental condition: the stimulus (full path to the target) is first presented in a modal window (not shown), and the participant must click "start" to begin the trial. The stimulus is also displayed at the top of the BIGFile browser throughout the trial. The image is cropped to save space: 11 additional files were visible below 'Fireman'.

During the retrieval phase, participants always started with level 3, and then proceeded to level 6 using the same interface. At each level, they completed two sessions. Session 1 consisted of 20 file retrievals, which comprised 10 different target files following a near-Zipf distribution (frequencies 5, 3, 2, 2, 2, 2, 1, 1, 1, 1), as in [61]. Unlike [61], where the experiment started with a uniform probability distribution, we started with the above-mentioned Zipf distribution so that the item that was assigned a certain frequency would appear the corresponding number of times. For instance, if an item was assigned a frequency of 5, it would appear as the target stimulus 5 times during the session. The mapping between frequency distribution and targets was counterbalanced across all participants and all conditions.

Each trial started by displaying the stimulus inside a popup window hiding the file browser. Participants were instructed to take as much time as they needed to understand the stimulus. When they were ready, they hit a start button to initiate the trial, at which point the content appeared inside the file browser (in both the adaptive and static parts, for the two conditions with split interfaces) and they were instructed to retrieve the file as fast and accurately as possible. When the popup window disappeared, the stimulus was shown in the toolbar at the top of the file browser, as in Fig. 57. When the participant successfully clicked the target, a popup window appeared with the stimulus for the next trial. If they clicked a wrong target, a popup window let them know that they had made an error and asked them to try again. After clicking a folder or a file, the score for this item was updated in ARFile. Similarly, after each user input, the probability of each potential target being the actual target was updated, and after each retrieval, the initial distribution for the potential targets was updated in BIGFile.

Session 2 repeated Session 1 with the same initial distribution and randomized selection order. The goal was to see whether and how participants would use the split interfaces once they were more familiar with the file hierarchy and had some expectations about the targets, which is more representative of real use. Participants could take a break between sessions and between interface conditions.

For each level, we categorized the 30 targets into 3 non-overlapping groups of 10. To reduce learning effects stemming from familiarity with the hierarchy, within each group, the targets came from different top-level folders for level 3, and from different second-level folders for level 6. The order of interface and group of targets were counterbalanced using Latin Square across all participants. Thus, the target group, the order in which each target group is seen, the ordering of targets within a group, and the order in which each interface is seen all serve as control variables. After Session 2, for each interface, participants completed the NASA Task Load Index (TLX) worksheets [93] and provided comments on the interface. After all three conditions, participants were asked for their preferences among the three interfaces. The experiment lasted about 90 minutes.

11.5.5 *Data Collection*

For each trial, the program collects the task completion time (**TCT**), the number of steps a participant takes to locate the target (the number of items clicked, including the final target), the amount of time spent at each step, the uncertainty the computer has about the final target, the calculated shortcuts, the participant's input at each step, and the information gain after each input. We collected 3 **INTERFACE** × 2 **LEVEL** × 2 **Session** (20 Selections each) × 18 Participants = 4320 trials.

11.6 RESULTS

For the analyses, we first removed 60 outliers (about 1.3%) in which **TCT** was larger than 3 standard deviations from the mean. We verified that outliers were randomly distributed across participants, interfaces and conditions. We also checked for outliers for all our other dependent variables, but none were found. Note that the results are the same if we include the outliers in the analyses. Except where noted, we ran a repeated-measures **INTERFACE** × **LEVEL** × **Session** factorial ANOVA on the dependent measures[1].

11.6.1 *Task Completion Time and Step Time*

Table 7 shows the results of a repeated measures ANOVA on **TCT**. All main effects are significant, as well as two interaction effects: **INTERFACE** × **LEVEL** and **LEVEL** × **Session**.

Factors	*df, den*	*F*	*p*
INTERFACE	2, 34	452.47	< 0.0001
LEVEL	1, 17	895.61	< 0.0001
Session	1, 34	32.12	< 0.0001
INTERFACE × **LEVEL**	2, 34	211.89	< 0.0001
LEVEL × **Session**	1, 17	14.69	$= 0.0242$

Table 7: Significant effects in the full-factorial ANOVA on **TCT**.

On average, BIGFile is 39.3% faster than ARFile, and 59.0% faster than Finder, across all levels and sessions. The significant interaction effect between **INTERFACE** and textbfLEVEL is shown in Fig. 58 (a) and (b). A post-hoc Tukey HSD test reveals that all differences are significant: BIGFile is 44.5% faster than ARFile and 63.8% faster than Finder at level 6, while BIGFile is 27.8% faster than ARFile and 47.6% faster than Finder at level 3. These findings are consistent with our simulation results: the deeper the target is located, the better BIGFile-Fast is compared to AccessRank.

---

[1] All analyses are performed with SAS JMP, using the REML procedure to account for repeated measures.

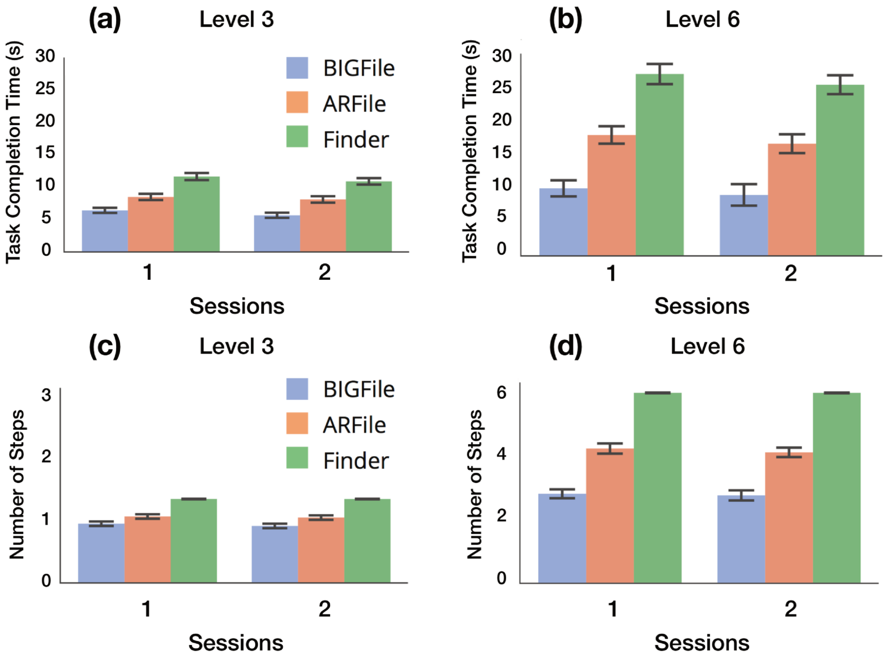


Figure 58: Task Completion Time (a, b) and number of steps (c, d) for the 3 interfaces, in 2 sessions, at levels 3 & 6, with 95% confidence intervals.

The repeated measures ANOVA on step time (the time of a single step) shows only two significant main effects: **INTERFACE** ($F_{2,34} = 114.48, p < 0.0001$) and **LEVEL** ($F_{1,17} = 142.27, p < 0.0001$). A post-hoc Tukey HSD test indicates that BIGFile averages 3.29s per step, which is significantly faster than ARFile (3.78s), which is significantly faster than Finder (4.05s). In terms of levels, the average step time is 3.35s for level 3 v.s. 3.94s for level 6, despite the fact that there are fewer files and folders at levels 5 and 6. Although not significant, the average step time is 3.78s in Session 1 v.s. 3.52s in Session 2. The **LEVEL** × **Session** interaction indicates that the difference in performance between Session 1 and Session 2 was generally smaller at Level 6 than at Level 3, probably because Level 3 trials provided some training for Level 6 trials.

11.6.2 *Number of Steps and Information Gain*

Table 8 shows the results of a repeated measures ANOVA on the number of steps (user inputs) required to locate the target. Both **INTERFACE**, **LEVEL** and their interaction significantly affect the number of steps. A post-hoc Tukey HSD shows the following significant differences:

- For level 3, BIGFile takes 2.09 steps to locate the target, compared with ARFile's 2.36 steps and Finder's 3.08 steps;

- For level 6, BIGFile takes 2.8 steps v.s. ARFile's 4.22 and Finder's 6.05, as shown in Fig. 58 (c) and (d).

The interaction effect indicates that the impact of the algorithm is more pronounced at level 6 than level 3. This also reflects the results of the simulations: BIGFile has a greater impact on a deep hierarchy.

Regarding the average information gain, BIGFile gains 1.27 bits per user input on average, while ARFile gains 0.94 bits and Finder gains 0.69 bits. A typical plot of the 3 interfaces under the same condition at level 6 is shown in Fig. 59. It shows that BIGFile gains more information from each user input, therefore the uncertainty drops to zero much faster at each step than with ARFile and Finder. By providing shortcuts, ARFile also gains more information from each user input than Finder.

Factors	*df, den*	*F*	*p*
INTERFACE	2, 34	3178.13	< 0.0001
LEVEL	1, 17	7267.81	< 0.0001
INTERFACE × LEVEL	2, 34	739.30	< 0.0001

Table 8: Full-factorial ANOVA on the number of steps required to locate the target. Only significant effects are shown.

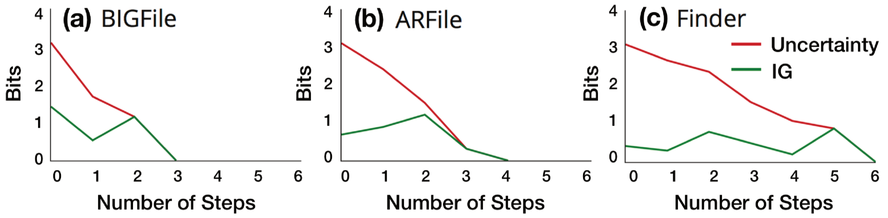


Figure 59: Uncertainty and information gain after each step in (a) BIGFile, (b) ARFile and (c) Finder.

11.6.3 *Subjective Feedback*

Subjective responses in all categories of the NASA TLX worksheets favored the split interfaces over Finder (Fig. 60). We note significant effects between BIGFile and ARFile as well as between ARFile and Finder in terms of mental demand, physical demand, performance, effort and frustration. In the Finder conditions, four participants asked *"where is the section above [adaptive area]? Can you make it come back?"* and mentioned that *"this is really slow..."*. However we found no significant differences between BIGFile and ARFile.

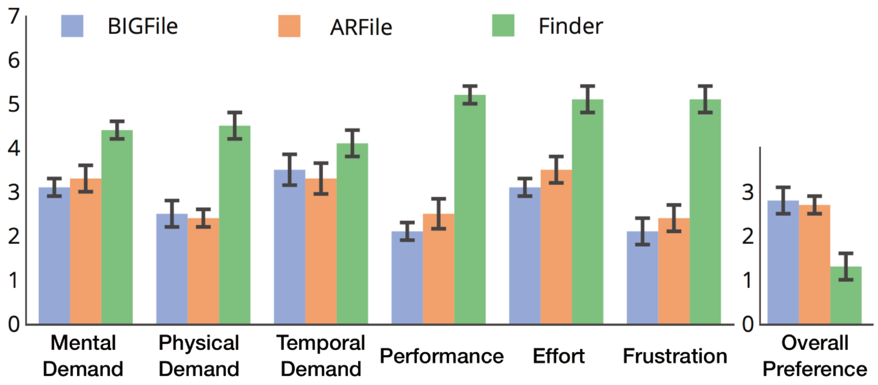


Figure 60: NASA TLX scores (from mental demand to frustration, lower is better) and overall preference (higher is better).

In terms of overall preference, participants ranked the interfaces (1 to 3). Both BIGFile and ARFile were preferred by participants over Finder, but there was no difference in preference between BIGFile and ARFile. When asked about their preference, 11 participants mentioned that they did not see any difference between BIGFile and ARFile and thought those interface conditions were just a different set of test words.

Three participants asked if they could search and whether the results in the adaptive area would correspond to their queries. Five participants mentioned that they would like to see "it" on their own file systems: "*I'm curious to see how it'd work on my own Finder. I have so many files everywhere with super long names... Wonder if this one will still work*". Another participant asked "*Can you reorder the list somehow as you like? Can you change the number [of shortcuts]?*" All these comments provide opportunities to further improve BIGFile.

11.7 DISCUSSION

We have seen that BIGFile is an effective technique, saving time and steps to retrieve a file in a hierarchical structure. We now discuss the benefits of split adaptive interfaces for file retrieval, provide a deeper comparison of BIGFileFast and AccessRank and outline some limitations of this work.

11.7.1 *Split File Interface*

To the best of our knowledge, BIGFile is the first attempt at introducing split adaptive interface to hierarchical file retrieval. Regardless of the underlying algorithm, BIGFileFast or AccessRank, the split interface outperformed the traditional Finder with unanimous preference from participants. This has not always been the case with adaptive interfaces, even split ones.

For example, Gajos & Chauncey [72] have demonstrated systematic individual differences in the use of adaptive features, correlated with users' personality traits. Hence this approach does not benefit all users equally. In our case, the preference might be due to the nature of the task: retrieving a file in a 3-level or 6-level hierarchy is much more difficult than selecting menu items, which is the task used in most split adaptive interface studies. Therefore, split adaptive interfaces may be more beneficial for difficult tasks where users need to "work hard" to reach their goal.

One possible issue with split interfaces is screen real estate. The more shortcuts are shown in the adaptive area, the better the underlying algorithm will work. But more shortcuts use more space and may result in higher cognitive demand and more occurrences of scrolling. Future work should therefore study the effects of the number of shortcuts on performance, preference and cognitive load.

11.7.2 *Comparisons with AccessRank*

Even though BIGFileFast can locate the final target more accurately than AccessRank in our simulations and experiment, unlike AccessRank, it does not account for repeated user behavior and repetitive access at the same time of day or day of the week. It also does not have a parameter to control the stability of estimated shortcuts across successive steps. These features are likely to benefit users in real settings. Future work should study their effect in BIGFile. More generally, AccessRank needs to be compared with BIGFileFast in more realistic settings.

We were surprised that users did not express a preference between BIGFile and ARFile, attributing the differences to the set of targets rather than the underlying algorithm. This may be due to the fact that interface differences are more obvious to users than the inner workings of a system. Indeed, Fitchett et al. [61] found *Icon Highlights* and *Search Directed Navigation* to be more effective than *Hover Menus*, even though the latter predicts targets several levels down the hierarchy. In that respect, our split interface is an alternative to *Hover Menus* that shows to be effective for both BIGFileFast and AccessRank. Further work should therefore tease apart the respective roles of the interface and the prediction algorithm in file retrieval tasks.

11.7.3 *Limitations*

Despite BIGFile's strong performance benefits, we want to emphasize some limitations of our experiment.

File Hierarchies: Our pilot study was designed to inform the design of our simulations and experiment in terms of the depth and width of the hierarchy we should evaluate. Even though we combined the results from our pilot study with those in the literature, it is still possible that the hierarchies we used are not fully representative. A larger scale study is needed to capture user file structures and retrieval practices.

Potential Target Set Size: The simulations showed that BIGFileFast performs much better on a 10-item potential target set than on a 1000-item potential target set. The latter is more realistic since users have thousands files and folders in their file systems. Larger target sets should therefore be tested to produce more robust findings.

Task Instruction: The task was initiated by showing a full path to the final target, which allowed users to compare the paths shown in the adaptive area with the instruction. In real life, recall of either the full path or the name of the final target is imperfect. Therefore, it is important to study how BIGFileFast performs in a more realistic setting, where navigation is combined with exploration.

In the next chapter, I discuss BIGnav and BIGFile as two applications of the Bayesian Information Gain (BIG) framework. I analyze their similarities and differences, discuss how BIG is relevant to other conceptual frameworks, and outline the opportunities for future work.

12

DICUSSION AND FUTURE PERSPECTIVES

In this chapter, I discuss: (a) The possible explanations for why BIGFile was preferred by the participants but not BIGnav, despite improved efficiency in both cases and (b) How BIG is conceptually related to other frameworks in the vein of human-computer partnership.

12.1 BIGNAV VS. BIGFILE

Both BIGnav and BIGFile significantly improved interaction efficiency: up to 40% for BIGnav and up to 64% for BIGFile compared to their respective baseline. Yet the subjective experience in these two cases differ. In BIGnav, half the participants did not prefer BIGnav in comparison to the standard pan and zoom, but in BIGFile, all participants unanimously preferred BIGFile and ARFile, a split adaptive interface using AccessRank [59] for prediction, to the Finder-like list view, which was used as baseline.

The first plausible explanation is the way in which the BIG framework is used. In BIGnav, the goal was to illustrate the best case scenario. The feedback, which is a view in multiscale navigation, is searched from the entire information space, therefore providing absolutely maximal information. It means that the system can jump far away from the current view, causing confusion for the users. In BIGFile, however, the feedback, which includes the static area presenting the usual hierarchy and the adaptive area presenting the estimated shortcuts, is maximally informative only respective to in this given situation. If BIG were used at its best, the view in BIGFile would comprise folders and files from anywhere in the hierarchical structure. This requires users to have perfect knowledge of their file system, which is fairly unrealistic. Therefore, BIGnav maximizes the absolute expected information gain and BIGFile maximizes the expected information gain relative to the current context.

The second plausible explanation is that in BIGnav, since two consecutive views might be far apart, the users cannot anticipate what they will see next. This results in higher cognitive load as they need to interpret what the system has just done and reorient themselves before inputting the next command (see Fig. 40).

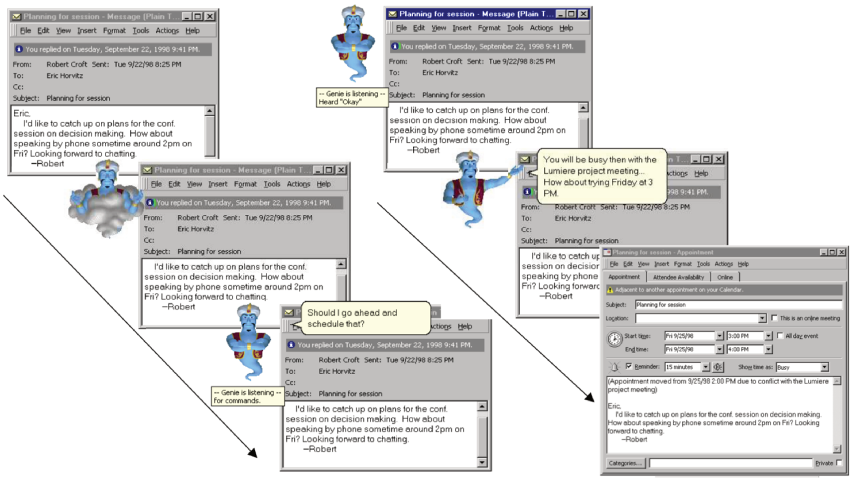
In contrast, in BIGFile, users have access to both the estimated shortcuts and the usual hierarchy so that they can choose which one to use: if the estimated items are not correct, they can always navigate the hierarchy in the usual way. Also, since contextual information is provided, users can navigate to a parent folder if the estimated path is only partially correct.

In both BIGnav and BIGFile, the system is actively probing users for more information. However, it seems that providing users with choices rather than making decisions for them is important. At a higher level, BIG fosters a form of collaborative interaction whereby system and user work together to achieve a common objective [171, 205]. It combines information from the system's side with user's intentions in order to optimize decision under uncertainty. This concept is related to several human-computer collaboration and adaptation notions that I discuss in the next section.

12.2 COLLABORATION AND ADAPTATION

While the notion of cooperative interaction between humans and computers is quite ancient [128], the concept of human-computer collaboration has emerged in HCI only recently [205]. In human-computer collaboration, two agents, a human and a computer, work together to achieve shared goals. A key aspect of collaboration is communication, for example to define goals, negotiate how to proceed and determine who will do what. Similar ideas can be found in *mixed-initiative design* [100], *human-computer partnership* [171] and *co-adaptation* [144]. All these approaches suggest that great opportunities lie between systems that provide automatic services [102] and systems where users are in full control [192].

1. **Mixed-Initiative Design**. Mixed-initiative interaction is inspired by human-human interaction where two humans communicate to negotiate in a dialogue based conversation [4]. The observations are that we use a range of different actions in the course of a conversation such as evaluating and comparing options and suggesting courses of action. In mixed-initiative interaction, a human and an intelligent system, use a flexible interaction strategy where each agent can contribute to the task what it does best [4]. Early works such as Lookout [101] illustrates how the system and users collaborate to perform complex tasks. The system has a utility function where it evaluates the costs and benefits of offering the service, and takes control when the benefits are significant but leaves the users in control when the costs are high (Fig. 61).



Planning for session - Message (Plain T...

You replied on Tuesday, September 22, 1998 9:41 PM.

From: Robert Croft Sent: Tue 9/22/98 8:25 PM
To: Eric Horvitz
Cc:
Subject: Planning for session

Eric,
I'd like to catch up on plans for the conf. session on decision making. How about speaking by phone sometime around 2pm on Fri? Looking forward to chatting.
–Robert

-- Genie is listening -- Heard "Okay"

You will be busy then with the Lumiere project meeting... How about trying Friday at 3 PM.

Should I go ahead and schedule that?

-- Genie is listening -- for commands.

Planning for session - Appointment

Adjacent to another appointment on your Calendar.

Subject: Planning for session
Location:
☐ This is an online meeting
Start time: Fri 9/25/98 3:00 PM ☐ All day event
End time: Fri 9/25/98 4:00 PM
☑ Reminder: 15 minutes Show time as: Busy

(Appointment moved from 9/25/98 2:00 PM due to conflict with the Lumiere project meeting)

Eric,
I'd like to catch up on plans for the conf. session on decision making. How about speaking by phone sometime around 2pm on Fri? Looking forward to chatting.
–Robert

Categories... Private ☐

Figure 61: Lookout: procedures that harness probability and utility guide Lookout's actions to assist users with accessing their calendar and composing appointments, based on a background analysis of email being reviewed. Taken from [101].

Like mixed initiative systems, BIG optimizes decision under uncertainty. However, the use of Bayesian Experimental Design [132] turns these systems on their head: rather than finding the action that best responds to the user, BIG challenges the user to give it useful information.

2. **Human-Computer Partnership**. Pohl's discussion of human-computer partnership in decision-support systems [171] is based on *human-computer integration*, one of the early fantansies envisioning a future where computers will not only be used as a tool when they are needed, but be integrated into every part of our lives. A famous example is Licklider [129] who wrote in 1960 of a coming "symbiosis" of people and machines that "will involve very close coupling between the human and the electronic members of the partnership". Other 1960's visions of a symbiotic future include Engelbart's "augmented human intellect" [49], Ted Nelson's dynamic hypermedia [160], and Alan Kay's Dynabook [112]. Emphasizing computer assistance rather than replacing human decision making, Pohl [171] suggests a complementary partnership between human users and computers. Similarly, BIG takes advantage of machine power and human intelligence: the "stimulus-response" relationship is replaced by a more collaborative one so that computers are no longer waiting for users to give input, but are probing users to facilitate interaction.

3. **Co-Adaptation**. Mackay [144] has proposed the concept of co-adaptation between people and technology to describe how users both *adapt* to the interactive technology they use, and also *appropriate* that technology for their own purposes.

Adaptation and appropriation occur together. In co-adaptive systems, users simultaneously adapt to the constraints imposed by the system and appropriate it in their own personal way. Rather than assuming that the designer is fully responsible for the 'user experience', co-adaptation suggests that users will always place their personal stamp on their use of the system. BIG leads to a reverse form of co-adaptation: while co-adaptation is about users adapting to new technology and also adapting it to their own needs, BIG adapts to users through its prior knowledge, which can change over time, and adapts users to its needs by prompting them constantly for information.

4. **Recommender Systems**. Another type of interactive systems that are worth mentioning is recommender systems. Recommender systems provide filtered information and seek to predict the "preference" that users would give to an item. They are used in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general [178]. Many of them use machine learning on massive data collected from users' history usage. Despite the progressively improved algorithms, recommender systems are still facing problems such as lack of accuracy [150], lack of trust [163], and being "annoying" for not being aware of the real situation [2].

   At first glance, BIGFile resembles a recommender system: it uses the user's history as priors and presents estimated shortcuts that users might be interested in. However, unlike recommender systems, in the BIG framework, the system is not trying to predict the most relevant items at the time, but tries to maximize the expected information gain. It might not necessarily show the items with the highest likelihood of being the target, but provides the approximately equiprobable items to estimate the final target. The simulations and controlled experiment in Chapter 11 have illustrated the effectiveness of the BIG approach in comparison to the best-of-breed prediction algorithm [59].

12.3 FUTURE PERSPECTIVES

Despite the effectiveness of the BIG approach demonstrated by BIGnav for multiscale navigation and BIGFile for hierarchical file retrieval, a number of improvements for these two particular cases as well as several opportunities for the framework at large arise.

BIGnav: In addition to the potential improvements that are mentioned in the discussion section of Chapter 9, namely using animation and combining BIGnav with standard pan and zoom, we can also regularize the search to compute the *locally* maximal expected information gain. As discussed earlier, BIGnav searches the entire information space, which can result in long-distance jumps between views. Regularizing the search area should therefore reduce the increased cognitive load. Moreover, we could allow users to stop the transition to the next view if they notice that BIGnav is going in the wrong direction, through more dynamic control.

BIGFile: As discussed in Chapter 11, even though BIGFileFast outperforms AccessRank [59], we would like to evaluate BIGFileFast with AccessRank with the stability parameter and potentially repeated user behavior. As BIGFileFast is a general algorithm that can be applied to navigate any hierarchical tree structure, its potentials should be explored in other applications. In addition, BIGFile should be evaluated in a longitudinal study to see whether and how it scales in realistic settings.

BIG: BIG is a general framework that can be utilized in many interaction tasks (Fig. 24). Once the potential targets Θ and their probability distribution $p(\Theta)$, system feedback X, user input Y, and user behavior $p(Y = y|\Theta = \theta, X = x)$ are modeled, one can always compute the actual information gain, or the information carried by the user input informing the computer of what she wants. In addition to maximizing the expected information gain (BIGnav) and leveraging the expected information gain (BIGFile), we can also combine the information criterion with other utility functions. Furthermore, the user behavior model can be personalized to account for individual behavior and can change over time.

BIG offers an operationalized framework that applies to human-computer communication in general. The information-theoretic notion of information (entropy) is a value capable of measuring a startling array of things – from the flip of a coin, to a telephone call and to human-computer interaction. It helps us better understand how information is exchanged and how technology can play a more active and "intelligent" role in the interaction, leading to more effective human-computer partnerships.

"Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?"
- T.S. Eliot

Part III

INFORMATION-THEORETIC MEASURES FOR CHARACTERIZING INTERACTION

This part builds on and extends the two previous parts. Users send information to the computer, therefore, an interaction task can be described using information-theoretic terms: how much information can be transmitted (entropy); how much information is successfully transmitted (mutual information); and what is the rate of successfully transmitted information (throughput).

I first introduce the information-theoretic measures to characterize an interaction task and demonstrate them in the context of command selection and text entry, comparing the information-theoretic notion of throughput with two existing definitions of throughput. By doing so, I further elaborate the benefits of using this framework to quantify the information transmission process from a user to a computer via an input device and an interface.

13

MOTIVATION

The work presented in this part is under preparation for a journal submission: Information-theoretic Measures for Characterizing Interaction. Wanyu Liu, Antti Oulasvirta, Olivier Rioul, Michel Beaudouin-Lafon and Yves Guiard.

Part i and part ii have shown a number of similar ideas from the literature that describe an interaction task involving a user, a computer, an interface and an input device with information-theoretic terms:

- **Pointing:** Two different throughputs [26, 138, 222] have been proposed to characterize the performance rate in aimed movements in bits per second (Part i).
- **Text Entry:** Text entry techniques, such as Dasher [137], which features an average text entry rate of 4.8 bits per second (Part i).
- **Full-body information capacity:** Oulasvirta et al. [164, 191] studied human control of continuous sensors and estimated throughput from 24 to 37 bits per second in a cyclical tapping experiment with a mouse (Part i).
- **Sonic interaction:** Berdahl et al. [14] focused on users controlling sound using continuous analog sensors and found that channel capacity for controlling a single, continuous sensor as high as 4 or 5 bits per second (Part i).
- **Multiscale navigation & hierarchical file retrieval:** BIGnav [134] reduces uncertainty by 0.88 bits on average in the controlled experiment while BIGFile [135] gains 1.27 bits per user input on average (Part ii).

Additionally, Roy et al. [181] compared two command selection techniques on touchscreen (Fig. 62) and analyzed data in information-theoretic terms [1]. In the communication channel considered in their study, a user serves as the source of information with her hand as the information emitter, and transmits information to the system with the touch screen as the receiver of the coded message. The code shared by the source (the user) and the destination (the system) is the mapping of a set of touch events to a set of commands. Roy et al. hypothesized that the transmitted information levels off, as in absolute judgment tasks [153], and that throughput as a function of command entropy is bell-shaped (Fig. 63).

---

[1] See Part i Chapter 4.

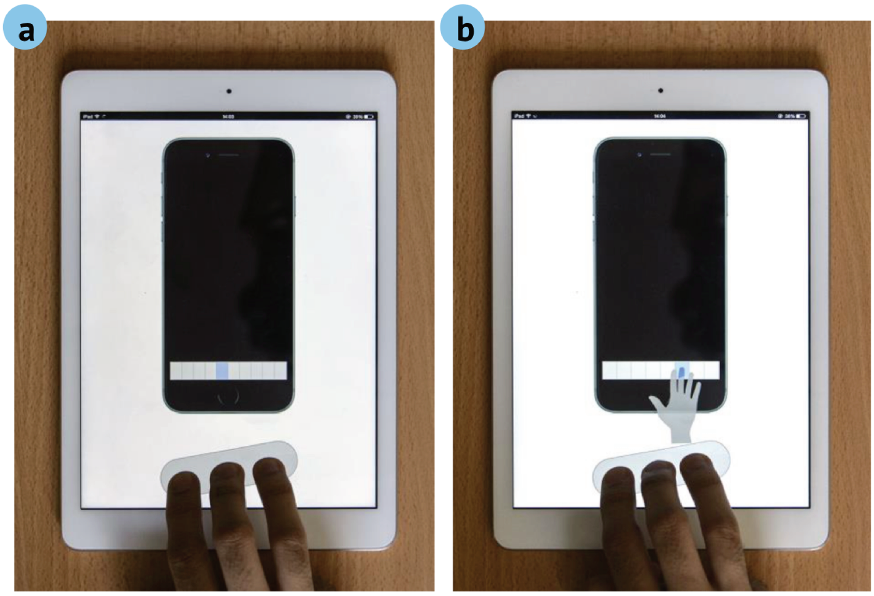


Figure 62: (a) Technique *Glass* and (b) Technique *Glass+Skin* with finger identification. Taken from [181].

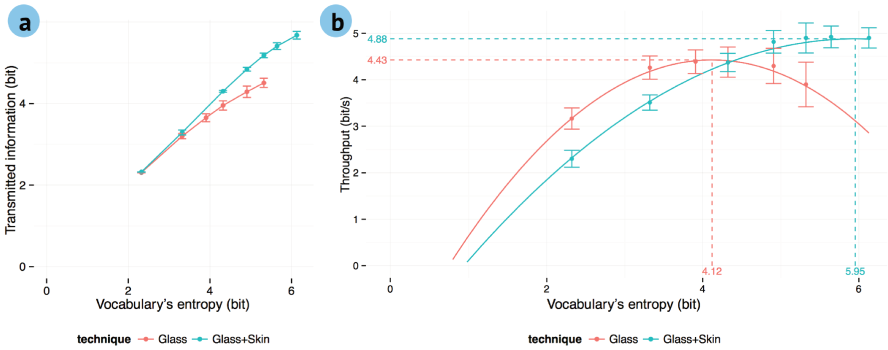


Figure 63: (a) Transmitted information and (b) Throughput as a function of vocabulary entropy. Taken from [181].

More recently, Guiard et al. [90] suggested using information theory as a general tool to evaluate input techniques. This is the most relevant literature to the present study as it outlines the shortcomings of conventional performance measurements and suggests that information-theoretic measures can help to address these problems.

Traditionally, input techniques are evaluated both objectively and subjectively. Subjective assessment includes usability, attitude or emotion through metrics relevant to experience (e. g. [93]), as well as physiological factors such as heart rate (HR) and Electromyography (EMG) [146]. Very often, Likert scales [131] are used to determine users' assessment of a given technique.

Subjective preference is an indisputably important factor to indicate how good a given input technique is. Objective evaluation, however, provides a matter-of-fact quantification of performance.

Since the beginnings of HCI [26] the objective evaluation of input techniques has primarily relied on speed and accuracy measurements, keeping in line with the psychological tradition. The separation of speed from accuracy entails a dilemma when comparing input techniques: if one technique is faster but less accurate than the other, which one is better? As a solution, in many cases such as pointing [138] and text entry [53], researchers usually deliberately control the error rate to some reasonable limit, typically 2% – 4%, so that only the speed dimension is of concern. But this approach hinders us from understanding the full speed-accuracy trade-off. One correct solution is to compare trade-off functions, rather than times and/or error rates, but this is a costly option and experimenters have persistently ignored it.

Another drawback of traditional measurement is the treatment of errors. Error rate is a rather coarse measure of accuracy. Insensitive to possible patterns, error rates convey no information on the sorts of errors users are prone to.

Guiard et al. [90] present the key information-theoretic terms (Fig. 64) and discuss how they can be applied to the input techniques evaluation problems. Namely, throughput offers a holistic picture of the speed-accuracy tradeoff while equivocation $H(X|Y)$ provides more information about how errors are made [2]. They also outline two concrete methodologies, one focused on the discrete timing and the other on the continuous time course of information gain by the computer, which is similar to the BIG approach (Part ii).

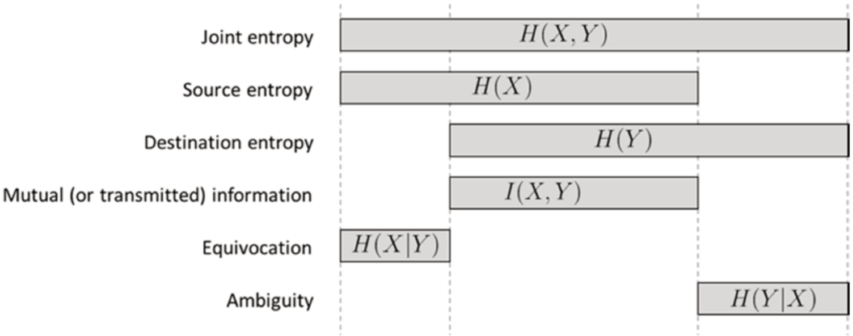


Figure 64: The decomposition of Shannon's entropy. Taken from [90].

---

[2] $H(X|Y) = H(X) - I(X;Y)$. See Chapter 1.

However, Guiard et al. [90] only provide theoretical discussions, which need to be complemented by experimental evidence. They also do not illustrate how such theoretical measures can be applied to input methods beyond pointing.

Building on previous studies and extending Guiard et al. [90], I advocate for the use of information-theoretic measures to characterize an interaction task. I first introduce each element in this framework and illustrate how these measures can be used in tasks beyond pointing: command selection and text entry, comparing the information-theoretic notion of throughput with two existing definitions of throughput. I demonstrate the benefits of using the information-theoretic measures to quantify the information transmission process from a user to a computer via an interface and an input device, which provides a coherent description of the task and allows conducting controlled experiment without deliberately controlling error rate. I also outline other possibilities for using the theoretically justified measurements to investigate and design interaction.

The contribution of this part lies in illustrating the advantages of these measures and promoting the use of information theory as a unified tool to characterize interaction.

14

INFORMATION-THEORETIC MEASURES

In this chapter, I demonstrate how information-theoretic measures can be used to characterize an interaction task. I consider the user as the information emitter, the computer as the information receiver, and the interface as the information channel as shown in Fig. 65.

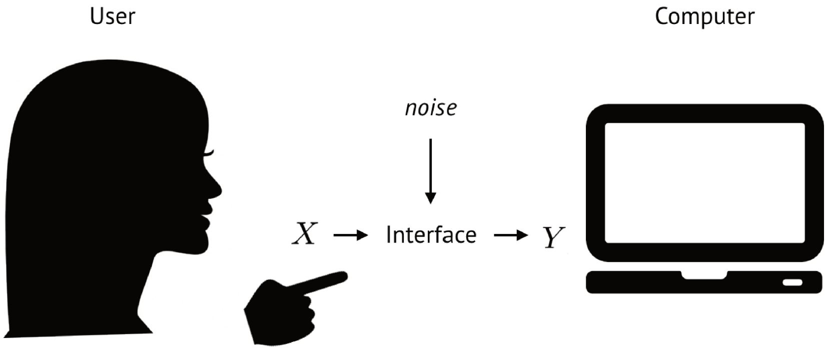


Figure 65: Information-theoretic measures for examining human-computer interaction as information transmission from the user to the computer through the channel interface. X is the intended input from the user and Y is the actual input received by the computer.

Note that I do not tease apart all the elements in Fig. 2. I do not distinguish how the information from the user is encoded nor how the information received by the computer is decoded.

Also, similar to Guiard et al. [90], I acknowledge that interaction sequences and feedback loops are often involved in a human-computer dialogue [68]. However, I assume that the fundamental concern of the human-to-computer communication process is the efficiency with which messages can be transmitted from the emitter (the user) to the receiver (the computer).

For a particular interaction task, I consider the messages (a set of intended inputs) sent by the user as random variable X. For instance, for a n-item menu, X takes n possible values in $\{x_1, x_2, ..., x_n\}$. X follows a probability distribution $p(x) = p(X = x)$ describing how each input is used. This can be computed from the user's history of use in real life, or from data collected in controlled studies.

The input received by the computer is modeled as random variable Y, which also takes values in $\{x_1, x_2, ..., x_n\}$. Since there is noise Z, which can come from the user's head, motor movement, or computer decoding, the input Y received by the computer does not always equal the input X sent by the user, therefore, there are errors, represented by random variable E:

$$E = \begin{cases} 0 & \text{if } X = Y; \\ 1 & \text{if } X \neq Y. \end{cases} \tag{40}$$

The probability of error $P_e = p(X \neq Y)$ representing the error rate has binary entropy:

$$H(E) = -P_e \log_2 P_e - (1 - P_e) \log_2 (1 - P_e). \tag{41}$$

14.1 INPUT ENTROPY $H(X)$

Using Equation 1, input entropy $H(X)$ captures how much information there is for the user to transmit. It is the maximum amount of information that could be transmitted in a given interaction scenario and it corresponds to the **input size** and the **probability distribution** of the inputs. The bigger the input size, or the more non-uniform the probability distribution, the higher the input entropy, and the more information can be sent.

A simple example considers a 4-item menu {"copy", "paste", "edit", "share"}. If the user uses all items equally, then the amount of information is maximum: $H(X) = \log_2 4 = 2$ bits. If $p(X = x)$ corresponds to {"copy" $= \frac{1}{2}$, "paste" $= \frac{1}{4}$, "edit" $= \frac{1}{8}$, "share" $= \frac{1}{8}$}, then the information in this context is $H(X) = -\sum_{i=1}^{4} p_i \log_2 p_i = \frac{1}{2}(-1) + \frac{1}{4}(-2) + \frac{1}{8}(-3) + \frac{1}{8}(-3) = 1.75$ bits. If there are more items, e. g. 8 equiprobable items, then the information is $H(X) = \log_2 8 = 3$ bits. This information expresses how uncertain the computer is about the user's input. An extreme case would be {"copy" $= 1$, "paste" $= 0$, "edit" $= 0$, "share" $= 0$}, meaning that the user only accesses the item "copy", then the information transmitted from the user to the computer is $H(X) = 0$: the computer has zero uncertainty about the user's input and knows exactly what the user would like to do.

14.2 TRANSMITTED INFORMATION $I(X; Y)$

Since there is noise Z in the channel, the received input Y by the computer does not always equal the intended input X. The successfully transmitted information is captured by mutual information (Equation 2).

For instance, in the aboved-mentioned case where all 4 items are equiprobable {“copy” $= \frac{1}{4}$, “paste” $= \frac{1}{4}$, “edit” $= \frac{1}{4}$, “share” $= \frac{1}{4}$}, the information the user can transmit is 2 bits. In the following sequence of inputs [“copy”, “paste”, “copy”, “paste”, “edit”, “share”, “edit”, “share”], the user made an error for the 5th input (“edit”): she wanted to input“share”. Then we can compute mutual information $I(X;Y) = 1.656$ bits [1]. The transmitted information percentage is therefore $I(X;Y)/H(X) = 1.656/2 = 82.8\%$. Roughly 82% of information gets successfully transmitted from the user to the computer. This measure of how much information the user actually transmits is similar to, but different from, the accuracy measure, which in this case is $7/8 = 87.5\%$.

14.3 TRANSMITTED INFORMATION RATE TP

Throughput captures the successfully transmitted information rate. It is computed by dividing the amount of successfully transmitted information $I(X;Y)$ by the average time T required to transmit such information and is measured in bits per second: $TP = \frac{I(X;Y)}{T}$ (Equation 6).

For instance, if the user takes 1.5 seconds to complete the series of above-mentioned inputs on average, then we can compute the throughput $TP = I(X;Y)/T = 1.656/1.5 = 1.104$ bits per second. The transmitted information rate from the user is then 1.104 bits per second. Throughput quantifies the information transmission efficiency for an interaction task and combines speed and accuracy into one single measure. In addition to the individual speed and accuracy dimensions, throughput accounts for the speed-accuracy tradeoff. For instance, if we have one technique that transmits more information than the other but takes longer, we can compare the information transmission rates to decide which technique is better in terms of speed-accuracy tradeoff, or information transmission efficiency.

14.4 EQUIVOCATION H(X|Y)

Equivocation $H(X|Y)$ captures the information loss in the transmission process and describes the uncertainty the computer has about the user's intended target given what is the actual input. It is computed by the difference between how much information the user could have transmitted (input entropy) and how much information the user actually transmitted (transmitted information), as indicated by Equation 2. In the above-mentioned case, the equivocation $H(X|Y) = H(X) - I(X;Y) = 2 - 1.656 = 0.344$ bits.

---

[1] `https://github.com/wanyuliu/Information-Theoretic-Metrics/` includes python code for computing all metrics of the examples in this part.

Equivocation gives much more information about how the user makes errors in comparison to the commonly used error rate P_e. Imagine that the user wants to select the item "edit" 4 times in a row and consider the following 2 scenarios:

- She successfully selects "edit" twice and selects the left neighboring item "paste" and the right neighboring item "share" once each due to hand jitter.
- She successfully selects "edit" twice and hits two random items far away from the intended one.

The error rate P_e is 0.5 in both cases but it is relatively easier to recover the intended input X from the actual input Y in the first case, therefore equivocation is lower in the first case than in the second. Equivocation not only illustrates the fact that information gets lost in the transmission process but also how it gets lost, therefore potentially helping to recover the true message. We will discuss the use of equivocation in intelligent text entry in Chapter 16.

14.5 ADVANTAGES OF THE INFORMATION-THEORETIC MEASURES

Using these information-theoretic measures has several advantages:

- They provide a standard language to characterize an interaction task. Input entropy $H(X)$ captures how much information can be transmitted; Transmitted information $I(X;Y)$ captures how much information the user actually transmits; Equivocation $H(X|Y)$ captures how much information gets lost in the transmission process and it is related to how the user makes errors; Throughput TP quantifies the information transmission efficiency and characterizes the speed-accuracy tradeoff.
- They are rooted in information theory with solid theoretical foundations. They (a) capture the distribution aspect of an interaction task, which has not been taken into account before; (b) they provide a more consistent description of the task when conditions are changed; and (c) they can be reasoned about using mathematical tools.
- In particular, throughput provides a way of avoiding the pitfall of the classic methodology of performance evaluation: to deliberately control experimental conditions in order to have error rates close to a reasonable minimum, usually below 4%, resulting in performance measures characterized only by speed. Such method is widely used in pointing (e. g. [138]) and text entry (e. g. [53]). Throughput offers an accurate and principled way of combining the speed and accuracy dimensions of performance.

- Equivocation provides information about how users make errors. The more errors, the more random the errors, the higher the equivocation. On the other hand, the familiar error rate is a rather coarse measure of accuracy. Insensitive to possible patterns, error rates convey no information on the sorts of errors users are prone to make [7]. Only by knowing how users make errors, can we learn from user behavior and design interaction with intelligent error correction approaches.

These measures can be applied and generalized as long as the following are known:

- X: A set of all possible messages (the intended inputs) that a user can transmit;
- $p(X)$: The probability distribution of the intended inputs;
- Y: The actual inputs, which take values in X;
- T: The time required for a user to transmit the messages.

In the next two chapters, I demonstrate how to use these measures as well as their benefits in the context of command selection (Chapter 15) and text entry (Chapter 16). Then I discuss other opportunities for future work in Chapter 17.

15

COMMAND SELECTION

Selecting commands is one of the most common interactions in graphical user interfaces and menus are widely used for exploring and selecting them. Given their prevalence and importance, menus have motivated many studies in HCI, including more than 60 new menu techniques during the last two decades (see Bailly et al. [8]). Conceptually speaking, menus offer a set of options where selecting and executing one (or more) of the options results in a change in the state of the interface [94]. They provide a straightforward approach to apply the information-theoretic measures as we know the input size X: The items in a menu; $p(X)$: The distribution of how each item is used; Y: The actual input on the menu and T: The time it takes to select a menu item.

This chapter demonstrates how the information-theoretic measures are used in the context of command selection. It includes:

- **Simulations** to explore how conditions affect the measures, comparing the information-theoretic notion of throughput with two existing definitions;
- **Data reanalysis** of existing command selection datasets to demonstrate the coherence of the information-theoretic notion of throughput in responding to the interaction task;
- **A controlled study** that investigates the information-theoretic measures with real users and reasons about their characteristics using information-theoretic concepts.

15.1 SIMULATIONS

The goal of simulations is to explore how changing conditions affects the information-theoretic measures. Particularly, we compare the information-theoretic notion of throughput with two existing definitions of throughput, discussed in Part i Chapter 3, both of which only apply to pointing:

- Equation 14 from Mackenzie [141].
- $1/b$ from Card et al. [26] and Zhai [222].

15.1.1 *Simulation Settings*

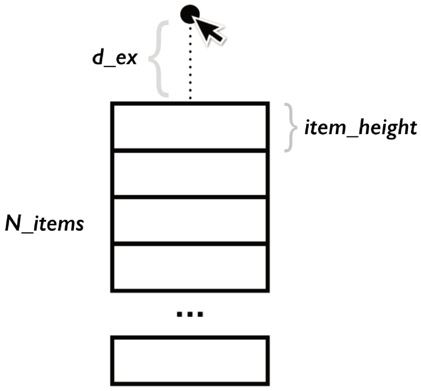


Figure 66: Simulation settings with three variables: *N_items* represents N items in the menu; *items_height* represents the height of the items and *d_ex* represents the distance between the starting point and the first item. We also control the probability distribution of the items as the fourth variable.

We simulate *pure pointing* [1] on a linear menu with N items of a certain height placed at a certain starting point (Fig. 66). The set of intended inputs X takes values in the menu items $\{x_1, x_2, \ldots, x_N\}$ and follows a probability distribution $p(x_i) = p(X = x_i)$ describing its usage. Y is the actual input, which also takes values in $\{x_1, x_2, \ldots, x_N\}$. The three throughput measures are defined as follows:

1. Information-theoretic throughput $TP_i = \frac{I(X;Y)}{MT} = \frac{H(X)-I(X;Y)}{MT}$.
2. Mackenzie's throughput $TP_m = \sum_{i=1}^{N} p(i) \times \frac{ID_i}{a+b\times ID_i}$.
3. Zhai's throughput $TP_z = \sum_{i=1}^{N} p(i) \times \frac{ID_i}{b\times ID_i} = 1/b$.

where $I(X;Y)$ represents transmitted information; $H(X)$ represents input entropy; index of difficulty for item i is $ID_i = \log_2(1 + \frac{D_i}{4.133\sigma})$; D_i is the average distance to select item i and σ is the standard deviation of the endpoint distribution; movement time for item i is therefore $MT_i = a + b \times ID_i$ where a and b are Fitts' law constants; $TP_m = TP_z$ if $a = 0$.

Each condition includes changing one controlled variable while keeping the others constant and comprises a series of 40 randomized selections according to the distribution condition. Errors are modeled by a Gaussian distribution $X \sim \mathcal{N}(0, \sigma^2)$ by adjusting the standard deviation σ of the endpoint distribution. Based on the errors, the actual user input Y is computed, as well as the error rate.

---

[1] *Pure pointing* refers to a task where movement time alone contributes to task completion time and can be therefore predicted by Fitts' law [63].

The movement time MT_i to item i is computed according to the actual user input using the Fitts' law constants $a = 0.37$ and $b = 0.13$ from [33]. Each condition is run 100 times and averages are taken.

The intended inputs X, actual inputs Y and movement time MT are logged, so as to compute input entropy $H(X)$, transmitted information $I(X;Y)$, error rate, and each of the three definitions of throughput.

15.1.2 *Simulation Results*

Simulation results are categorized into two groups: changing the values of the controlled variable does not change input entropy; and input entropy might change due to increased number of items or change of distribution.

15.1.2.1 *Constant Input Entropy*

Varying Error Rate. Since I assume a Gaussian distribution, we can model different error rates by adjusting the variance σ^2. In the simulations, σ^2 takes values in {0.1, 0.3, 0.5, 1, 3, 10}, which corresponds to error rates {0.0075, 0.0166, 0.054, 0.2766, 0.6784, 0.8480}. Other variables are kept constant: the number of items $N = 8$, starting distance *d_ex* = 4 and item height = 2. The input distribution is uniform, meaning that each item appears 5 times (randomized) out of 40 selections.

Fig. 67 (a) illustrates the differences among the three definitions of throughput. Since $TP_z = 1/b$, it stays constant for a given b across conditions. TP_m, on the other hand, accounts for errors, therefore, the higher the error rate, the lower the throughput. The information-theoretic notion of throughput TP_i provides a similar result and this can be explained by Fig. 67 (b): the higher the error rate, the higher the equivocation $H(X|Y)$, therefore the lower the transmitted information $I(X;Y)$. Since movement time is constant, the transmitted information rate decreases.

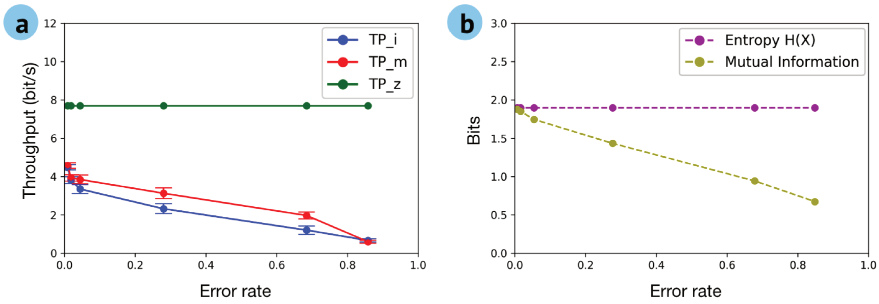


Figure 67: (a) Comparison among three measures of throughput and (b) Pattern of input entropy as well as transmitted information when error rate increases.

Varying Item Height and Starting Distance. Similarly, I modeled different item heights and starting distances to investigate how the three measures of throughput are affected. Conceptually speaking, changing item height or starting distance does not affect input entropy but affects movement time: the bigger the item height, or the larger the starting distance, the longer time it takes to reach the items. The other variables are kept constant in both cases respectively: the number of items $N = 8$, starting distance $d\_ex = 4$ when item heights are varied, item height $= 2$ when starting distances are varied, variance $\sigma^2 = 0.5$, corresponding to an error rate of 5.4%. A uniform distribution was also used.

Fig. 68 shows the differences among the three measures of throughput when varying (a) item height and (b) starting distance. Since the input entropy does not change in both cases, nor does the transmitted information due to the constant error rate, what affects throughput is movement time. When increasing item height or starting distance, the average movement time to reach the items increases, therefore the information-theoretic notion of throughput decreases, as shown in both Fig. 68 (a) and (b). However, we observe that TP_m increases, which does not conform to the information-theoretic rationale.

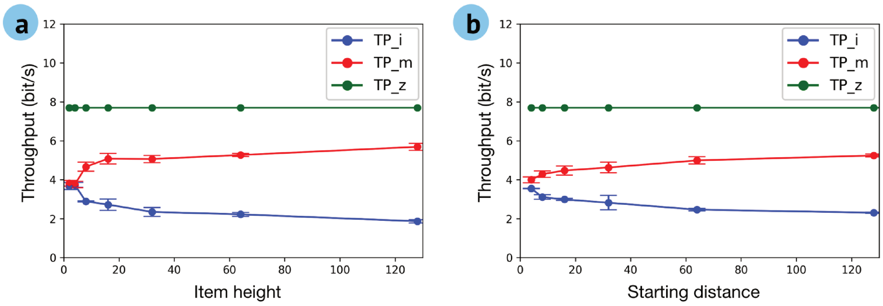


Figure 68: Comparison among the three measures of throughput when (a) item height and (b) starting distance are changed.

15.1.2.2 *Changing Input Entropy*

Varying the Number of Items. We can also change the input entropy by varying the number of items in the menu. Similar to previous simulations, the number of items takes values in {4, 8, 16, 32, 64, 128, 256} while other variables stay constant: starting distance $d\_ex = 4$, item height $= 2$, variance $\sigma^2 = 0.5$, corresponding to an error rate of 5.4% and a uniform distribution.

Fig. 69 illustrates (a) the comparison among the three measures of throughput and (b) the pattern of input entropy as well as transmitted information when the number of items increases.

More items indicate more input entropy and since we kept the error rate constant, the transmitted information increases with input entropy (Fig. 69 (b)). Moreover, average movement time increases with the number of items, but more slowly than input entropy. Therefore, the transmitted information rate increases (Fig. 69 (a)). TP_m shows a similar increasing pattern but with a much shallower slope.

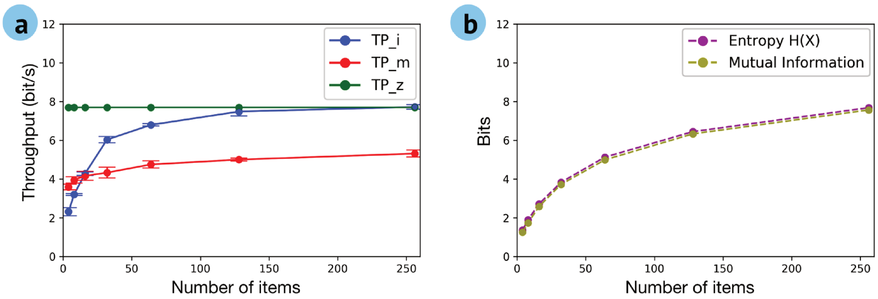


Figure 69: (a) Comparison among the three measures of throughput and (b) Pattern of input entropy as well as transmitted information when the number of items increases.

Varying Distribution. Finally, we can also change the input distribution to change the input entropy. I used 3 distributions: Uniform, Zipfian with $s = 1$ and Zipfian with $s = 2$ [2]. As before, other variables are kept constant: the number of items $N = 8$, starting distance *d_ex* = 4, item height = 2 and variance $\sigma^2 = 0.5$. While a uniform distribution incurs the highest uncertainty (entropy), a Zipfian distribution indicates a skewed distribution with some elements more frequently selected than the others.

Fig. 70 illustrates (a) the comparison among the three measures of throughput and (b) the pattern of input entropy as well as transmitted information when distribution changes. It can be seen that changing the distribution from a uniform one to a non-uniform one decreases input entropy. Since the error rate is kept constant across conditions, the transmitted information decreases with input entropy: there is less information that can be transmitted in a skewed distribution (Fig. 70 (b)). As movement time is the same across conditions, the transmitted information rate is lower with a skewed distribution than with a uniform one, as observed in the information-theoretic notion of throughput (Fig. 70 (a)). Neither TP_m nor TP_z account for a change in distribution.

---

[2] The definition of a Zipfian distribution is given in Part i Chapter 4.

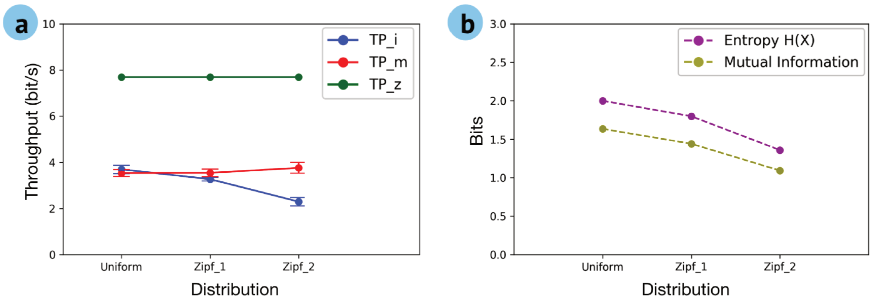


Figure 70: (a) Comparison among the three measures of throughput and (b) Pattern of input entropy as well as transmitted information when distribution is changed.

15.1.3 *Summary*

The simulations help us explore the characteristics of the information-theoretic notion of throughput in comparison to the two other known definitions of throughput in the context of command selection with pure pointing. We summarize the results in Table 9.

	$H(X)$	$I(X;Y)$	$H(X\|Y)$	TP_i	TP_m	TP_z
NofX $\nearrow$	$\nearrow$	$\nearrow$	–	$\nearrow$	$\nearrow$	–
$p(X)$ $\nearrow$	$\searrow$	$\searrow$	–	$\searrow$	–	–
P_e $\nearrow$	–	$\searrow$	$\nearrow$	$\searrow$	$\searrow$	–
T $\nearrow$	–	–	–	$\searrow$	$\nearrow$	–

Table 9: A summary of how the three measures of throughput change when different conditions change. $\mathrm{NofX}\nearrow$ represents increasing the number of items; $p(X)\nearrow$ represents skewing the input distribution; $P_e\nearrow$ represents increasing error rate and $T\nearrow$ represents increasing movement time. – represents no change.

We can see that the information-theoretic notion of throughput is consistent with the change of experimental condition. Furthermore, since it is rooted in information theory, we can foresee how the changes affect throughput by theoretical reasoning. On the contrary, $TP_z = 1/b$ is insensitive to the changes while TP_m contradicts the information-theoretic notions when probability distribution or movement time changes. By definition, both TP_m and TP_z heavily depend on Fitts' law constants, which are derived from linear regression. When a task involves more than pure pointing, which is usually the case in HCI, they are prone to errors. Moreover, neither of them considers distribution aspect, which proves to be important in empirical studies [133], as demonstrated in the next section using previous collected data.

15.2 DATA REANALYSIS

In this section, I reanalyze data collected from two studies comparing menu performance: Bailly et al. [9] and Liu et al. [133]. In the former, a linear menu with 3 different menu organizations (unordered, alphabetic and semantic) and 3 different lengths (8, 12 and 16) and a uniform distribution was used; in the latter, an unordered linear menu with 12 items using 3 distributions (uniform, Zipfian with $s = 1$ and Zipfian with $s = 2$) was considered. The information-theoretic measures are defined as in the previous section: X denotes the menu items; $p(X)$ captures the distribution of the intended inputs and Y describes the actual inputs. Unlike the simulations, these two studies capture real user behavior performing a menu selection task in a controlled setting: they first respond to the stimulus, and then search for the corresponding item before moving the mouse pointer to reach it. I compare the information-theoretic notion of throughput TP_i with TP_m and TP_z to explore how they evolve.

15.2.1 *Changing the Number of Items and Menu Organization*

In order to validate a mathematical model of menu performance, Bailly et al. [9] considered three menu organizations: *unordered*, where menu items are randomly organized, *alphabetic*, where menu items are alphabetically organized, and *semantic*, where items that convey similar semantic meaning are grouped together and separated from other groups visually. They also considered 3 menu lengths: 8, 12 and 16 items, with the same menu width.

Participants first hover above a single button on the screen to get the stimulus and press the button to start the trial, then click on the correct target as fast as they can. Bailly et al. [9] removed the wrong selections, therefore speed alone can provide metrics for how organization or length affects menu performance. The authors graciously provided their data so that I could analyze it and compute three definitions of throughput *including* errors.

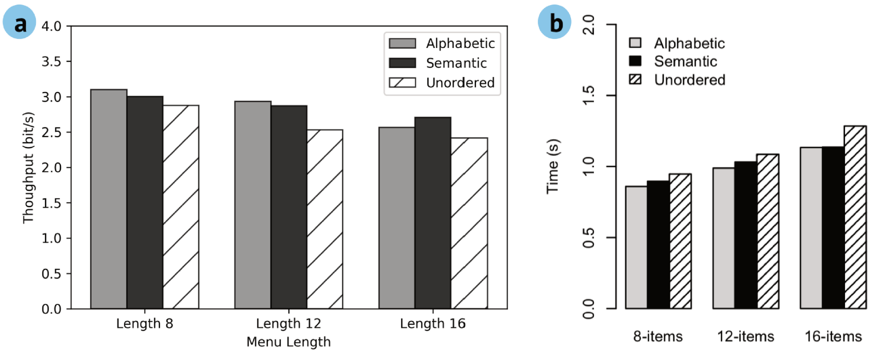


Figure 71: (a) Information-theoretic notion of throughput and (b) selection time (taken from [9]) for 3 organizations and menu lengths.

Fig. 71 (a) shows how the information-theoretic notion of throughput responds to different menu organizations and menu lengths. We can see that, for instance, an alphabetically organized linear menu with 8 items has roughly a 3 bit/s information transmission rate, which is the highest across all conditions. When menu length increases, the transmitted information rate decreases. This contradicts Table 9 in the previous section where increasing the number of items increases input entropy and therefore increases information transmission rate. The reason is that, in this experiment, increasing the number of items from 8 to 12 increases input entropy by $\log_2 12 - \log_2 8 = 0.58$ bits but increases selection time more probably due to increased visual search time. Therefore, having more items in the menu increases input entropy, but also the overall time, resulting in a reduced throughput.

Nevertheless, the information-theoretic notion of throughput provides consistent results compared with Fig. 71 (b). For a menu with 8 items, alphabetic organization has the lowest selection time, followed by semantic organization and then unordered organization. In Fig. 71 (a), we observe the same pattern: alphabetic organization has the highest throughput, then semantic organization and then unordered organization. This also holds true for a menu with 12 items. However, for a menu with 16 items, alphabetic organization results in roughly the same selection time as semantic organization in Fig. 71 (b). In Fig. 71 (a), semantic organization has a higher throughput than alphabetic organization. This implies that semantic organization has a better speed-accuracy tradeoff, or a higher information transmission rate than alphabetic organization when there are more items in the menu.

Comparing the information-theoretic notion of throughput TP_i with TP_m and TP_z (Fig. 72), we can see that while TP_m demonstrates a similar pattern as TP_i, TP_z, on the other hand, since it is defined as $1/b$, heavily depends on the linear regression of Fitts' law. Consequently, it is difficult to predict or justify its behavior when changing experimental conditions.

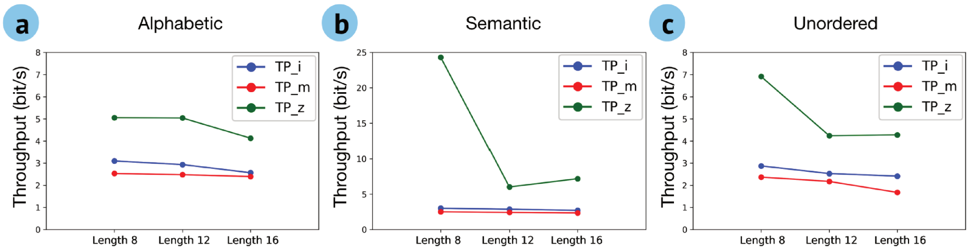


Figure 72: Three definitions of throughput in (a) alphabetic; (b) semantic and (c) unordered menu. Data from [9].

15.2.2 *Changing Distribution*

Liu et al. [133] used one menu organization *Unordered* and one menu length *12 items* but varied the distribution. Their goal was to see how sensitive users are to different distributions as menu selection is known to follow a Zipfian distribution in real life [33, 55] but is usually tested with a uniform distribution in lab studies. In order to compare menu items with the same item frequency but that belong to different distributions, they considered three distributions that result in 45 selections (Fig. 73).

Frequency distribution	*Selection count for 12 items (decreasing order)*	*Distance (R^2) ∣ d*
Uniform	**6**, **4**, 4, 4, 4, 4, 4, 4, **3**, 3, 3, **2**	n/a ∣ 0.26
Zipfian s=1	15, **6**, 5, **4**, **3**, 3, **2**, 2, 2, 1, 1, 1	>0.99 ∣ 0.14
Zipfian s=2	28, **6**, **4**, **3**, **2**, 1, 1, 0, 0, 0, 0, 0	>0.99 ∣ 0.19

Figure 73: Distributions used in [133]: names, multiset of item frequencies, the r-squared and pairwise distance. These distributions have 4 item frequencies in common marked in bold.

Liu et al. [133] also followed the same procedure as Bailly et al. [9]: participants need to hover above a button to get the stimulus and click on it to start the trial. They are instructed to select the target as fast and accurately as they can. Wrong selections were removed in the original analysis, but I reanalyze them here *including* errors.

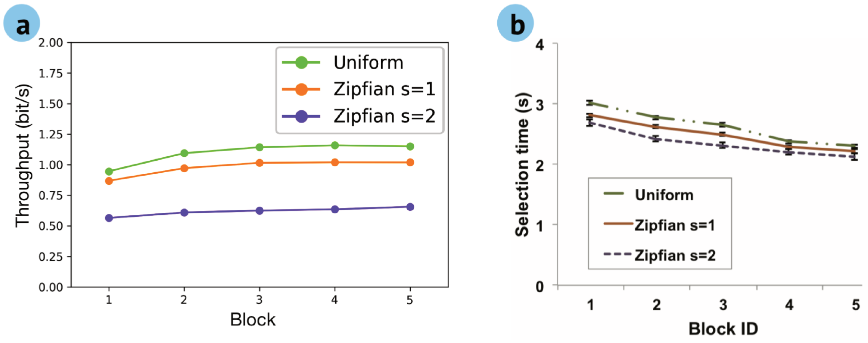


Figure 74: (a) Information-theoretic notion of throughput and (b) selection time (taken from [133]) for 3 distributions across 5 blocks.

Fig. 74 shows how the information-theoretic notion of throughput responds to different distributions. In the experiment, the input entropy for 3 distributions differs: 3.54 bits for the uniform distribution, 3.06 bits for the Zipfian ($s = 1$) distribution and 1.83 bits for the Zipfian ($s = 2$) one. Even though Zipfian distributions are faster to select, as seen in Fig. 74 (b), the uniform distribution has a higher information transmission rate (Fig. 74 (a)). This is in line with Table 9 where skewed distributions have lower throughput than the uniform distribution.

Comparing the information-theoretic notion of throughput TP_i with TP_m and TP_z (Fig. 75), we can see that while TP_m demonstrates a similar pattern as in the simulations (Fig. 70), showing the opposite trend than the information-theoretic notion of throughput TP_i, TP_z, which is computed from the linear regression of Fitts' law and takes the inverse of the slope b goes below 0. This means that in the condition Zipfian ($s = 2$), pointing is no longer the dominating component of the total selection time. Indeed, as stated in [133], users select only a small portion of the commands very frequently and leave most of the other items in the menu aside. This creates an unbalanced familiarity with the menu as a whole: when an infrequent item appears, participants spend more time visually searching for it. Therefore, replying on Fitts' paradigm no longer holds in this case.

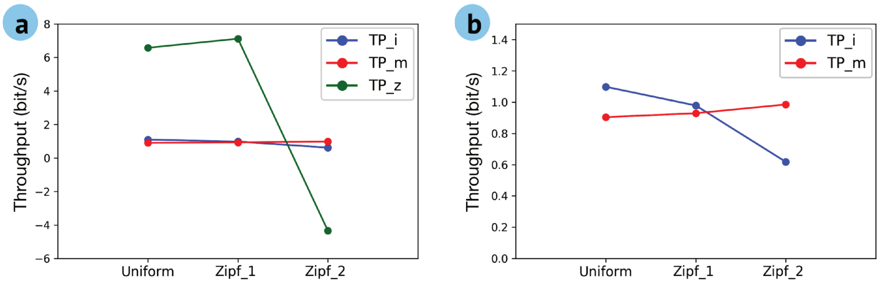


Figure 75: (a) Three definitions of throughput in 3 distributions and (b) a zoomed-in comparison between TP_i and TP_m.

15.2.3 *Summary*

The reanalysis of empirical data helps us better understand how the information-theoretic measures play out in real studies. The results can be summarized as follows:

1. The two existing definitions of throughput are difficult to interpret when beyond pure pointing tasks, especially $\mathrm{TP}_z = 1/b$;
2. The information-theoretic measures respond consistently with the experimental conditions and provide the additional benefits of investigating the combined dimension of speed and accuracy;
3. The empirical computation of the information-theoretic notion of throughput matches simulations, except in the case of an increased number of items. In the simulations, TP_i increases with input entropy; whereas in empirical data, TP_i decreases when input entropy increases. In the next section, I report on a controlled study that varies the number of items (input entropy) and I use information-theoretic concepts to reason about the properties of transmitted information as well as its rate.

15.3 CONTROLLED STUDY

The work presented in this section is published at IFIP INTERACT 2017: Information-Theoretic Analysis of Human Performance for Command Selection. Wanyu Liu (WL), Olivier Rioul (OR), Michel Beaudouin-Lafon (MBL) and Yves Guiard (YG).

Since the notion of user-to-computer information transmission process is rooted in information theory, we can use mathematical tools to reason about the properties of the measures. This work is inspired by the study by Roy et al. [181] where the information-theoretic measures are used to illustrate the differences between two interaction techniques when the number of inputs, which they call *input vocabulary* and *command's entropy*, increases. As Roy et al. [181] focused on comparing these two techniques (Fig. 62), they hypothesized that the leveling-off effect of the transmitted information is similar to absolute judgment tasks [3] [153], and that throughput as a function of command's entropy is bell-shaped (Fig. 63).

Conceptualization: OR, WL
Implementation: WL
Evaluation: WL
Data analysis: WL
First draft: WL
Paper writing: WL, OR, MBL, YG
Supervision: OR, MBL, YG

To investigate and analyze these phenomena theoretically, I designed and conducted an ad-hoc command selection experiment. This section reports on this experiment and provides a theoretical analysis for the above hypotheses.

15.3.1 *Data Collection*

15.3.1.1 *Participants and Apparatus*

Twelve volunteers (1 female), age 23 to 31 (mean = 26.6, $\sigma = 1.9$), were recruited from our institution. All of them were right-handed and interacted with WIMP interfaces regularly.

The experiment was conducted on a Macbook Pro with a 2.7 GHz processor, 8 GB RAM and resolution at 227 pixels per inch. The software was implemented in Java and the experiment window was 600 × 400 pixels. The targets representing the commands were displayed at the top of the window as a row of adjacent rectangles. The total area covered by the targets was 256 pixels wide and 30 pixels high. The width of the targets depended on the experimental condition. A circle positioned 150 pixels down below the target area was used to reset the cursor position of each trial. A standard mouse was used with the same sensitivity for all participants.

15.3.1.2 *Task, Stimulus and Design*

In response to a visual stimulus, participants were instructed to click on the highlighted target command (Fig. 76 (a)) as fast and accurately as they could. If they correctly hit the target command, it turned green (Fig. 76 (b)). Clicking on a non-target command would turn it red. In both cases the trial was complete after a single selection. The cursor was reset at the same position at the start of each trial.

---

[3] See Part i Chapter 2.

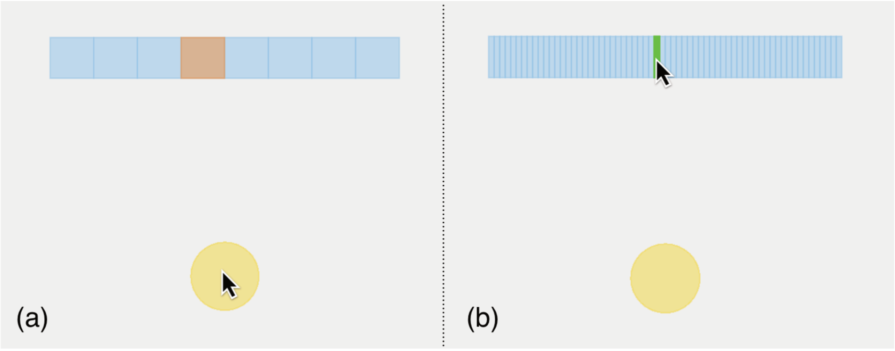


Figure 76: Experimental condition: (a) The cursor is reset at the center of the circle when the trial starts in condition 8; (b) correctly selected target command turns green in condition 64.

Based on a pilot study, we used 4, 8, 16, 32, 64, 128 and 256 commands in the experiment with a uniform distribution, corresponding to 2 to 8 bits of information as input entropy. Note that more than 7 bits of information is relatively high for normal users to process, but we wanted to push the limits of the participants. The size of the target representing each command was inversely proportional to the number of commands in the set, so that the set of target commands always occupied the same overall space.

We used a within-participant design and counter-balanced the order of the number of commands across participants with a Latin square. There were 3 replications for each block. A block consisted of presenting all targets in random order. Since the distribution is uniform, each command should appear the same number of times. However, this would result in a very long and tiring selection in conditions 128 and 256. In order to keep the duration of the experiment manageable, each participant had to select only 64 targets in conditions 128 and 256, but the full range was covered across all participants.

For each trial, the program collected user input and movement time, which started from moving the cursor to clicking on the target. The time that participants needed to respond to the stimulus before moving the cursor was logged but separated from movement time. The total duration of the experiment was around 20 minutes per participant. In summary, the design was: 12 Participants × (4+8+16+32 +64+64+64 Commands) × 3 Replications = 9,072 trials.

15.3.2 *Results*

Given the measured movement times and error, we can analyze transmitted information $I(X;Y)$, throughput TP and error rate P_e as a function of the input entropy $H(X)$, and how equivocation $H(X|Y)$ is related to error rate P_e.

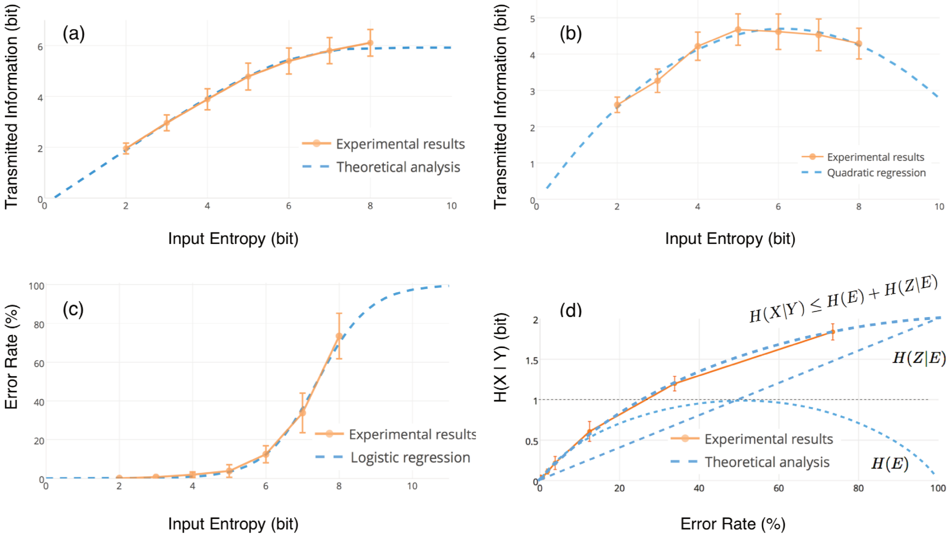


Figure 77: Experimental results with 95% confidence intervals.

15.3.2.1 *Transmitted Information*

Fig. 77 (a) shows that transmitted information $I(X;Y)$ increases gradually with the input entropy $H(X)$. Similar to the findings of Roy et al. [181], transmitted information tends to reach an upper bound, confirming the limited capacity. Their reason for the leveling off is that it is similar to absolute judgment tasks [153]. We offer another explanation based on information theory later in this section.

15.3.2.2 *Throughput*

Similar to Roy et al.'s study [181], throughput (TP) also shows a bell-shaped behavior: it increases with input entropy, reaches a maximum around 5 bits and then decreases as shown in Fig. 77 (b).

15.3.2.3 *Error Rate*

Fig. 77 (c) demonstrates that when the input entropy $H(X)$ is small and the task is easy, users do not tend to make mistakes, hence P_e is very small. When the input entropy increases, users make more and more errors, up to 73.5% when entropy equals 8 bits. It is obvious that when input entropy gets very high, the error rate would level off at 100%.

15.3.2.4 *Equivocation*

Fig. 77 (d) illustrates the relationship between equivocation $H(X|Y)$ and error rate P_e: equivocation increases with error rate in this case. We offer a theoretical rationale for this phenomenon next.

15.3.3 *Information-Theoretic Analysis*

We provide an information-theoretic analysis for (a) why mutual information should level off; and (b) why throughput should be a bell-shaped function of the input entropy in the given scenario. As in Chapter 14, X is the intended input, Y is the actual input, Z is the noise in the information channel and E is the errors.

As Equation 2 indicates, mutual information is the difference between the input entropy $H(X)$ and equivocation $H(X|Y)$ of the input given the output: $I(X;Y) = H(X) - H(X|Y)$. The conditional entropy equivocation is a measure of the uncertainty about X knowing Y; but if we know Y, the uncertainty on the noise Z is the same as that on X, so we can rewrite Equation 2 as:

$$I(X;Y) = H(X) - H(Z|Y). \tag{42}$$

Here $H(X) = \log_2 M$. We now would like to bound the penalty term – equivocation – $H(Z|Y)$ in the transmitted information. Since the knowledge of the output Y reduces the uncertainty on the noise Z (conditioning reduces entropy [34, Theorem 2.6.5]), we have:

$$H(Z|Y) \leqslant H(Z). \tag{43}$$

In other words, equivocation does not exceed the entropy of the noise. Thus it is the noise's entropy that penalizes the transmitted information.

In our experiment, users make errors as defined in Equation 40, and we can use the chain rule [34, Theorem 2.2.1]: $H(Z) = H(Z, E) = H(E) + H(Z|E)$ where:

$$H(Z|E) = P_e \times H(Z|E=1) + (1 - Pe) \times H(Z|E=0) = P_e \times H(Z|E=1). \tag{44}$$

since there remains no uncertainty on the noise Z if there is no error ($E = 0$). Combining the above, we find that the equivocation is bounded by:

$$H(X|Y) \leqslant H(E) + P_e \times H(Z|E=1). \tag{45}$$

This is known in information theory as Fano's inequality [34, Theorem 2.10.1].

Here $H(E)$ is given by Equation 41 and is at most one bit (when $P_e = 0.5$). Hence making errors penalizes the amount of transmitted information by at most one bit. However, considering the second term of Equation 45, the uncertainty on "wrong selections" $H(Z|E = 1)$ incurs an additional penalty on the amount of transmitted information: how users make errors, not just the fact that they make errors, affects the amount of transmitted information.

In our case, errors are clustered near the actual target, hence the entropy of the noise is lower than if they were evenly distributed. The relationship between error rate P_e and $H(X|Y)$ observed from empirical data matches exactly the above illustration as shown in Fig. 77 (d). We can now reason as follows:

- **For small** M**:** users do not tend to make errors, $H(E) \approx 0$ and $P_e \approx 0$, therefore $H(X|Y)$ is close to zero or remains very small when the error rate is low. So $I(X;Y)$ increases with $H(X) = \log_2 M$;
- **For large** M**:** we tend to have $P_e = 1$, $H(E) = 0$, users cannot make a correct selection, but the errors are clustered around the target as in pointing tasks [219]. Doubling the number of commands from M to $2M$ adds 1 bit to the input entropy, but since the error area around the correct target is approximately the same physical size, the number of possible errors is also doubled. Hence the equivocation is also increased by 1 bit. In our data, the possible errors in condition 128 are 1-3 around the target while in condition 256 they are 1-5 around the target, which corresponds approximately to the same physical area. As a result, the amount of transmitted information $I(X;Y) = H(X) - H(Z|Y)$ is not increasing any more and levels off as illustrated in Fig. 77 (a).

Combining this analysis with movement time, we can now turn to the theoretical analysis of the throughput TP:

- **For small** M**:** $\log_2 M$ is also small, and movement time is dominated by the intercept, hence can be considered as approximately constant. TP increases slowly with the input entropy;
- **For large** M**:** movement time grows linearly with $\log_2 M$, and transmitted information $I(X;Y)$ levels off. Hence TP gradually decreases as demonstrated in Fig. 77 (b).

However, we should distinguish the ceiling effect of transmitted information in our case from that in absolute judgment tasks [153]. Roy et al. claimed that they have the same characteristics but in our case, the errors made by users are around the target since they can see where it is, and therefore $H(Z)$ is only a few bits. In absolute judgment tasks, since the key phenomenon is that human short-term memory has a limited capacity, one would expect that when the number of randomly ordered stimuli increases, $H(Z)$ gets close to $\log_2(M-1)$ as Y can take any value in $\{1, 2, \ldots, M\}$. If this were the case, mutual information $I(X;Y)$ should go down, instead of leveling off as $I(X;Y) \approx \log_2 M - \log_2(M-1)$ at first order when M is very large. Since input entropy never gets very large in this type of tasks, the phenomenon is thus never observed. This would require further investigation in the context of absolute judgment tasks.

15.3.4 *Summary*

In summary, in command selection tasks, the amount of transmitted information gradually increases with input entropy until it reaches its capacity, and then levels off. Correspondingly, TP demonstrates a bell-shaped behavior, increasing to reach a maximum and then decreasing. This maximum (corresponding to an entropy of 5 bits, i. e. 32 commands in our experiment) provides the optimal input size for the given selection technique.

Following Soukoreff and MacKenzie [198], who argue that people are imperfect information processors, this experiment demonstrates that when input entropy increases, users tend to make more errors, which leads to the above-mentioned behavior of transmitted information and throughput. However, we should be aware that this is just one particular case because of the specific conditions of most HCI experiments including the one reported in this section:

- Participants are instructed to move as fast or as accurately as they can, sometimes both. We can imagine that if they could take their time to complete the task, the error rate would be always low, therefore the mutual information $I(X;Y)$ would always increase with $H(X)$.
- Since the stimulus is often visual, the errors are around the real target, which causes the leveling off effect of transmitted information. If errors were different, equivocation would be different, and the transmitted information would be different.

The theoretical formulations have shown that it is not just the fact that users make errors, but *how* they make errors, that affects the transmitted information, which is in turn tightly related to both the experimental design and the instructions to the users. In this case, equivocation $H(X|Y)$ can help us better understand the nature of errors, which is not captured by the commonly used error rate P_e, and potentially inform the design of error correction schemes. In the next chapter, I apply the information-theoretic measures to text entry, demonstrate the computation of conditional input entropy and discuss the effects of auto-correction.

16

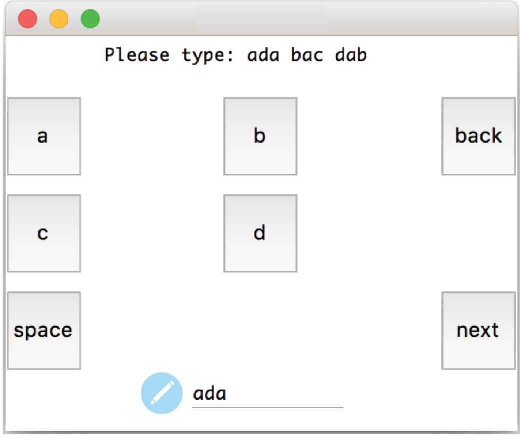
TEXT ENTRY

I have outlined the statistical language processing concept in Part i, Chapter 5. Although text entry is modeled as communication over a noisy channel, the information-theoretic measures have not been widely adopted for describing text entry behavior. Instead, text entry is often characterized by words per minute (WPM), keystrokes per character (KSPC), Levenshtein string distance (LD), etc. (See Part i, Chapter 5).

This chapter attempts to apply the information-theoretic measures to text entry. It first introduces how to estimate these measures using conditional entropy and demonstrates the implementation of an intelligent statistical decoder for auto-correction. Then it presents a set of simulations where the information-theoretic measures are compared with the conventional text entry measures and the information-theoretic notion of throughput is compared with the two other definitions of throughput. I also report on a pilot study with real users to explore the measures in a more realistic setting, as well as the effect of auto-correction. Note that the study in this chapter is still preliminary. It calls for further investigation.

16.1 CONDITIONAL INPUT ENTROPY

Similar to command selection, we need to model the intended input X, the probability distribution $p(X)$ describing X, the actual input Y and the time it takes to transmit the messages. However, unlike command selection, where one command changes the status of the interface, text entry involves typing a sentence that includes a string of letters and special characters. I therefore define a series of random variables X_i that represent the character that a user is supposed to type at time i. Instead of considering all 26 letters and special characters, here I use a simplified example (Fig. 78): X_i takes values in $\{a, b, c, d, \text{`` ''}\}$ where " " represents a space. Random variable $Y_i$ represents the character that a user actually types at time $i$. It also takes values in $\{a, b, c, d, \text{`` ''}\}$. In Fig. 78, one can also use *back* to erase the incorrect characters or use *next* to transcribe the next sentence.



Please type: ada bac dab

ada

Figure 78: A toy keyboard with 7 keys: *a, b, c, d, space* (corresponding to character " "), *back* represents backspace to erase the incorrect characters, *next* goes to the next sentence. The intended input X is shown above the keys and the currently transcribed sentence Y is indicated at the bottom.

To estimate the entropy of a sentence, a probabilistic language model is needed. A language model captures probable sequences of letters and words by counting their occurrence in some large representative dataset – a corpus, and assigns probabilities to sequences of letters and words [165]. Since natural languages are highly redundant, these probability could be based on all previous characters $p(X_m|X_{m-1}, X_{m-2}, \dots, X_1)$. However, this does not scale as the number of characters m increases. Therefore language models make a Markov assumption and restrict the number of previous characters to $n-1$. This results in an n-gram language model, which estimates $p(X_m|X_{m-1}, X_{m-2}, \dots, X_{m-n})$. Commonly used models are unigram ($n = 1$), bigram ($n = 2$) and trigram ($n = 3$). The higher the value n, the more accurate the estimation.

In this chapter, I use a unigram model. Therefore, the input entropy of a sentence including m symbols can be estimated as:

$$H(X_1, X_2, \dots, X_m) = \sum_{i=1}^{m} H(X_i|X_{i-1}). \tag{46}$$

Using the chain rule [34, Theorem 2.2.1], Equation 46 can be rewritten as:

$$\begin{aligned} H(X_1, X_2, \dots, X_m) &= H(X_1) + H(X_2|X_1) + \dots + H(X_m|X_{m-1}) \\ &= -\sum p(X_1) \log p(X_1) - \sum p(X_2|X_1) \log p(X_2|X_1) \\ &\quad - \dots - \sum p(X_m|X_{m-1}) \log p(X_m|X_{m-1}). \end{aligned}$$

Once the user has transcribed the intended sentence, we can estimate the equivocation as follows:

$$\begin{aligned} H(X_1, ..., X_m | Y_1, ... Y_m) &= \sum_{i=1}^{m} H(X_i | X_{i-1}, Y_i, Y_{i-1}) \\ = H(X_1|Y_1) + H(X_2|X_1, Y_1, Y_2) &+ ... + H(X_m|X_{m-1}, Y_{m-1}, Y_m). \end{aligned} \tag{47}$$

With equivocation, we can estimate transmitted information, which is the difference between Equation 46 and Equation 47:

$$\begin{aligned} &I(X_1, X_2, ..., X_m | Y_1, Y_2, ..., Y_m) \\ &= H(X_1, X_2, ..., X_m) - H(X_1, X_2, ..., X_m | Y_1, Y_2, ..., Y_m). \end{aligned} \tag{48}$$

and compute transmitted information as a percentage $I(X;Y)/H(X)$.

For instance, let us assume that we have a language model L as follows: {bab cc dad d dd ddd dc dc dac dab d c bab abad cab dab ada da ab ba ad abba caaba add dd d dddd d dd dd} (Fig. 79).

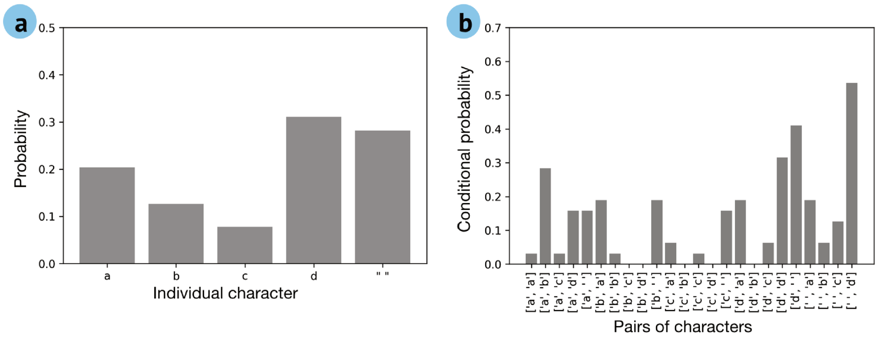


Figure 79: (a) Probability of individual character and (b) Conditional probability of pairs of characters in language model L.

Then the input entropy of a sentence, e. g. "ada bac dab" (Fig. 78), using Equation 46 is 40.4 bits. The average entropy per symbol is 3.67 bits [1]. If a user transcribes this sentence several times over time {"ada bac dab", "ada bac dab", "ada bac dab", "ada bac dab", "ada bac dab", "ada bac dab"}, with some minor errors as indicated in red, the transmitted information using Equation 48 is 33.9 bits. 84% information gets successfully transmitted, in comparison to the conventional accuracy rate of 97%. If an expert user types this sentence at a rate of 63 WPM (see Part i Chapter 5), which means that she types this sentence in 2.1 seconds on average, the information transmission efficiency TP_i is therefore 16.1 bit/s.

---

[1] Shannon's average entropy per letter using 26 letters and conditioning on 1 previous letter is 4.14 bits [189].

16.2 INTELLIGENT STATISTICAL DECODER

As discussed in Part i, Chapter 5, text entry is an integral activity in our daily life, therefore there has been a number of intelligent text entry methods that strive to improve text entry rate [165]. These methods infer or predict the user's intended text by exploiting redundancies in natural languages to increase users' ability to communicate as quickly and as accurately as possible.

Auto-correction is one of those methods. Its principal purpose is to correct common spelling or typing errors. If it corrects user input correctly, it saves time compared to doing it manually. On the other hand, if the algorithm corrects it in a wrong way, it is very costly as the user needs to correct the auto-correction before moving on [2].

The information-theoretic measures can be used to investigate the effect of auto-correction. To do so, I implemented a substitution-only statistical decoder using the token passing method [165, 221] on the 7-key keyboard (Fig. 78). Since text entry is inherently *noisy* and *uncertain*, text entry decoding approach aims to find the most probable message Y given the intended message X. This conditional probability $p(y|x)$ is computed from Bayes' rule:

$$p(Y = y|X = x) = \frac{p(X = x|Y = y)p(Y = y)}{p(X = x)}.$$

The most probable message Y is then $\arg\max_Y [\frac{p(X=x|Y=y)p(Y=y)}{p(X=x)}]$. Since the decoder is only attempting to identify the message that maximizes the posterior probability, the denominator is just a normalization constant and will be invariant under the search. It can therefore be ignored: $\arg\max_Y [p(X = x|Y = y)p(Y = y)]$ where $p(X = x|Y = y)$ is the likelihood of the input message $X = x$ given a particular hypothesis for $Y = y$ and $p(Y = y)$ is the prior probability of the message $Y = y$ without taking the input message into account. The objective of the decoder is therefore to identify the most probable hypothesis for the observation sequence $Y = \{Y_1, Y_2, ...Y_i\}$.

Similar to Fig. 20 in Part i Chapter 5, we decode an observation sequence of length i by starting with a single initial token, which predicts the empty string ϵ with 1.0 probability for observing zero observations (Fig. 80).

---

[2] The growing use of auto-correction on smartphones has also led to the creation of several websites, e. g. `http://www.damnyouautocorrect.com`, where people post and share humorous or embarrassing cases of improper auto-corrections.

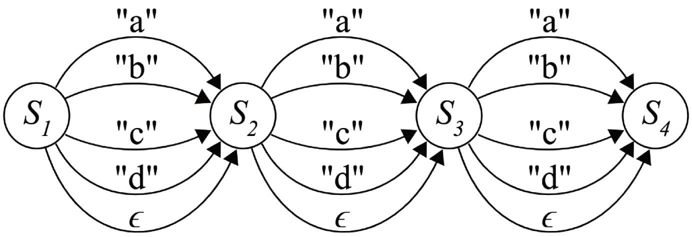

"a"
"b"
"c"
"d"
ϵ
S_1
S_2
S_3
S_4

Figure 80: A substitution-only decoder: S_i represents a state and several paths with different symbols are possible from the start state to the end state. The subsitution-only decoder will search all such possible paths and substitute an observation for a symbol when transitioning from one state to another.

We then take any token at observation index $j \in [0, i)$ and propagate k tokens to observation index $j+1$, where k is for instance the number of keys on the keyboard (possible output symbols). This is repeated for all tokens until a token is in the last observation and can no longer propagate any tokens as there are no more observations. These final tokens represent all the complete hypotheses that explain the entire observation sequence. The final token with the highest probability contains the hypothesis that best explains the observation sequence. A simple example can be seen in Fig. 81.

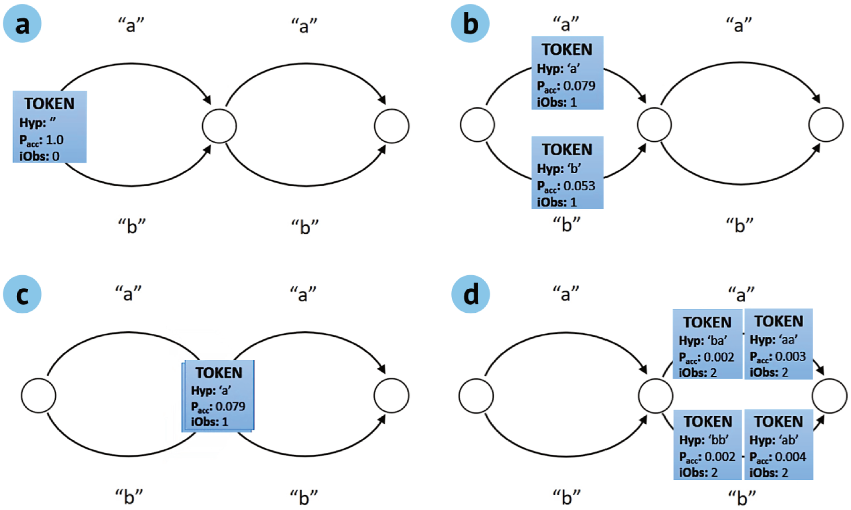

a
"a"
"a"
TOKEN
Hyp: ''
Pacc: 1.0
iObs: 0
"b"
"b"
b
"a"
"a"
TOKEN
Hyp: 'a'
Pacc: 0.079
iObs: 1
TOKEN
Hyp: 'b'
Pacc: 0.053
iObs: 1
"b"
"b"
c
"a"
"a"
TOKEN
Hyp: 'a'
Pacc: 0.079
iObs: 1
"b"
"b"
d
"a"
"a"
TOKEN
Hyp: 'ba'
Pacc: 0.002
iObs: 2
TOKEN
Hyp: 'aa'
Pacc: 0.003
iObs: 2
TOKEN
Hyp: 'bb'
Pacc: 0.002
iObs: 2
TOKEN
Hyp: 'ab'
Pacc: 0.004
iObs: 2
"b"
"b"

Figure 81: (a) A token starts without hypothesis *Hyp* and has accumulated probability $P_{acc} = 1$ at observation index $iObs = 0$; (b) it passes through potential characters "a" and "b" to the state (c) where the most likely token is 'a' and then it continues to the next observation (d).

In my implementation, I computed $p(X = x|Y = y)$ from the pre-transcribed sentences and enabled token pass when the space key is hit: a word will be auto-corrected when it is complete.

By adjusting the pre-transcribed sentences, we can simulate two scenarios: (a) a wrong input is auto-corrected to a correct one and (b) a correct input is auto-corrected to an incorrect one. Both cases will be illustrated next with simulations.

16.3 KEYBOARD SIMULATIONS

In this section, I use the 7-key keyboard example (Fig. 78) to explore the characteristics of the information-theoretic measures. The height and the width of each key are adjustable and the language model L is used. Transmitted information before starting the simulation is set to 1. Conventional text entry measures including words per minute (WPM), error rate, and keystrokes per character (KSPC) are computed the same way as in Part i Chapter 5. Since Fitts' law is also used to model pointing behavior in text entry, we can compare the information-theoretic notion of throughput with the two definitions of throughput from Chapter 15: $TP_m = ID_e/MT$ where ID_e is the effective index of difficulty and MT is movement time (Equation 14) and throughput $TP_z = 1/b$ where b is the slope of Fitts' law. As in Chapter 15, Fitts' law constants are $a = 0.37$ and $b = 0.13$ (from [33]).

The pointer is set at the center of the *next* key at the beginning of the simulation.

Three settings are investigated:

1. The size of the keys is varied and there are no errors.
2. There are uncorrected errors.
3. An intelligent decoder is in the loop.

16.3.1 *No Errors*

In the first simulation, I assume that users perfectly transcribe the intended sentence, but use different key sizes, therefore I compare movement times. Three cases are considered: (a) small keys ($\text{width} \times \text{height} = 4 \times 2$); (b) regular keys ($\text{width} \times \text{height} = 5 \times 5$) and (c) large keys ($\text{width} \times \text{height} = 10 \times 10$) (Fig. 82). Obviously, it is more difficult to type with small keys (a) than with large keys (c). When no errors occur, the information-theoretic notion of throughput should be higher with large keys (c) than with small keys (a). The intended sentence is "ad cb aac" as shown in Fig. 82. The simulation ran 10 perfectly transcribed sentences for each of the three cases along with the time for each key press.

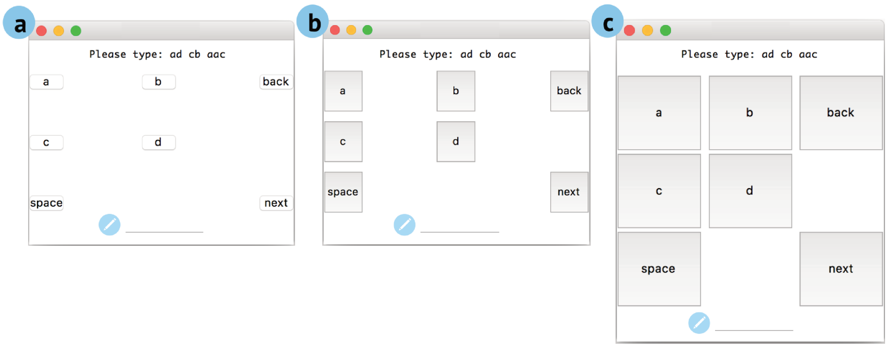


Figure 82: Three keyboards with keys in different sizes: (a) small; (b) regular and (c) large.

Table 10 shows the results of different measures on the 3 keyboards. We can see that the larger the keys, the easier to access them, therefore the less time to transcribe the same sentence. Since I only consider perfectly transcribed sentences, the transmitted information equals entropy. The information-theoretic notion of throughput is computed simply by dividing the transmitted information by the time for transmitting such information. As expected, the throughput is much higher when keys are large than when keys are small: a keyboard with larger keys has higher information transmission efficiency.

The conventional measurements provide similar results. Since the keyboard with large keys is easier to type, the words per minute rate is higher. Since there are no errors (neither uncorrected errors nor corrected errors), Keystroke Per Character (KSPC) equals 1.

Compared to the information-theoretic notion of throughput, TP_m is increasing with the key size but is much smaller while TP_z remains constant across conditions (Fig. 83 (a)).

	Small Keys	*Regular Keys*	*Large Keys*
Total Time (s)	4	3.5	3.2
Input Entropy (bits)	32.8	32.8	32.8
Transmitted Information (bits)	32.8	32.8	32.8
Transmitted Information (%)	1.0	1.0	1.0
Throughput (bit/s)	8.2	9.4	10.3
Words Per Minute	24	27.4	30
KSPC	1.0	1.0	1.0
TP_m (bit/s)	1.3	2.6	4.0
TP_z (bit/s)	7.7	7.7	7.7

Table 10: A summary of different measures on the simulations of 3 keyboards.

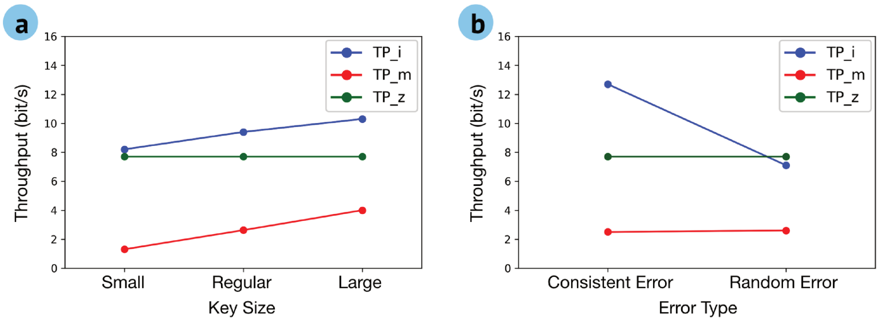


Figure 83: Comparison among three measures of throughput when (a) key size changes and (b) error type changes.

16.3.2 *Uncorrected Errors*

To explore how the information-theoretic measures account for errors, I simulated the scenario where users mistype a character and leaving it in the sentence so that there are still errors at the end. Only the keyboard with regular keys was tested (Fig. 82 (b)).

For the same sentence "ad cb aac", the simulation ran 10 times each in two conditions. In the first condition, the last 3 characters "aac" were registered correctly 7 times and incorrectly 3 times with consistent errors as "abc". In the second condition, the last characters were registered correctly 7 times as well but the errors were "abc", "acc", "adc". Table 11 shows the measures.

	Errors {abc, abc, abc}	*Errors {abc, acc, adc}*
Total Time (s)	3.5	3.5
Input Entropy (bits)	32.8	32.8
Transmitted Information (bits)	31.8	24.8
Transmitted Information (%)	0.98	0.76
Equivocation (bits)	1.0	8.0
Throughput (bit/s)	12.7	7.1
Words Per Minute	27.4	27.4
KSPC	1.0	1.0
Error rate (%)	0.11	0.11
TP_m (bit/s)	2.5	2.6
TP_z (bit/s)	7.7	7.7

Table 11: A summary of different measures on the simulations of 2 error types.

As seen earlier in Chapter 15, it is not just the fact that users make errors but also how they make errors that affects transmitted information. In both cases here, the error rate is 11% but the equivocation & transmitted information differ.

In the first case, since all 3 errors are the same, the equivocation is low and there is a certain probability for receiving "b" when it is supposed to receive "a". In contrast, in the second case, "b", "c" and "d" are received once each when "a" is expected, so it is difficult to recover the intended input from the received characters. The equivocation is high in the second case, and the transmitted information is low compared to the first case. On the other hand, how the errors are made is not reflected in the classical measures (Table 11). Fig. 83 (b) also shows the comparison among the three measures of throughput and we can see that neither TP_m nor TP_z account for the randomness of errors.

16.3.3 *Auto-correction*

We can also use the information-theoretic measures to investigate the effect of auto-correction on text entry. I simulated 3 scenarios using the keyboard with regular-sized keys (Fig. 82 (b)):

1. There is no intelligent decoder in the loop: users mistype a character, use *back* key to erase it and retype the correct one;
2. There is an intelligent decoder in the loop: users mistype a character, which is then auto-corrected correctly by the auto-correction method;
3. There is also an intelligent decoder in the loop: users correctly type a character, which is then auto-corrected incorrectly so that users need to correct the auto-correction and then retype the correct character.

Table 12 summarizes the measures in three scenarios.

	No Auto-correction	*Correct Auto-correction*	*Incorrect Auto-correction*
Total Time (s)	4.4	3.5	4.8
Input Entropy (bits)	32.8	32.8	32.8
Transmitted Information (bits)	32.8	32.8	32.8
Transmitted Information (%)	1.0	1.0	1.0
Throughput (bit/s)	7.5	9.4	6.8
Words Per Minute	21.8	27.4	20
KSPC	1.25	1.0	1.38
TP_m (bit/s)	2.3	2.6	2.4
TP_z (bit/s)	7.7	7.7	7.7

Table 12: A summary of different measures on the simulations with or without auto-correction.

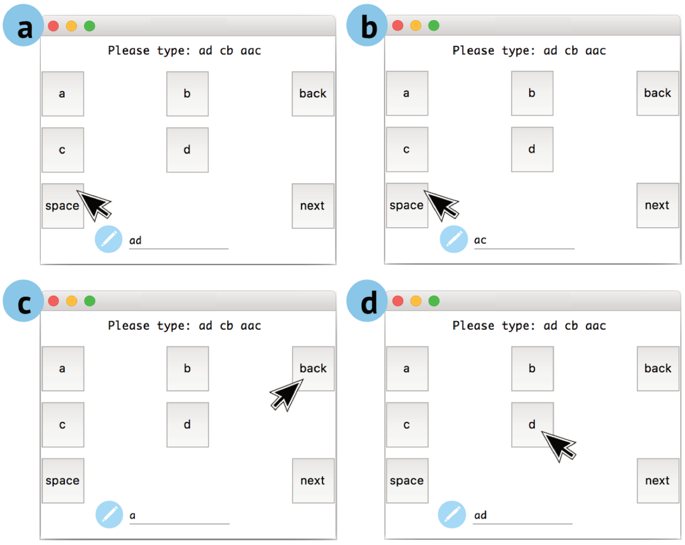

a
Please type: ad cb aac
a
b
back
c
d
space
next
ad
b
Please type: ad cb aac
a
b
back
c
d
space
next
ac
c
Please type: ad cb aac
a
b
back
c
d
space
next
a
d
Please type: ad cb aac
a
b
back
c
d
space
next
ad

Figure 84: After the user hits the space button (a), auto-correction corrects the input to “ac” (b), the user erases the incorrect correction (c) and re-inputs the right character (d).

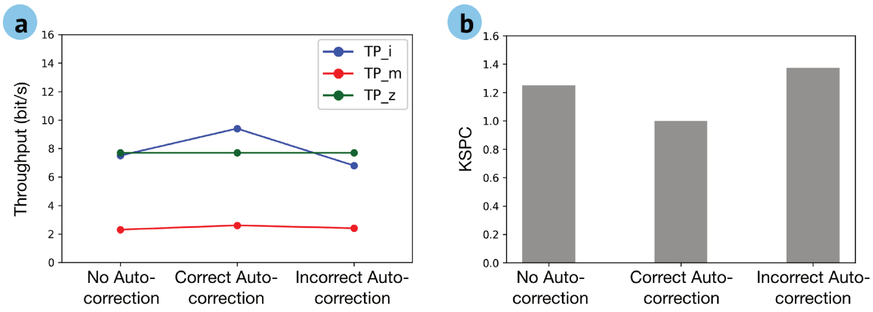


Figure 85: (a) Comparison among three measures of throughput and (b) KSCP with or without auto-correction.

From Table 12 we can see that it is very costly to correct incorrect auto-correction (4.8 seconds vs. 4.4 when correcting errors manually). This is also reflected in the information-theoretic notion of throughput: 6.8 bit/s when correcting an auto-correction vs. 7.5 bit/s when correcting errors manually. In contrast, neither TP_m nor TP_z account for this aspect (Fig. 85 (a)). The reason can be seen in KSPC as when auto-correction provides incorrect correction, extra steps are needed to roll back to the previous state (Fig. 85 (b)). On the other hand, when auto-correction correctly corrects errors, it is beneficial for users. In Fig 84, if a user types “ab” and hits the space key, the input “ab” is automatically transformed into “ad”. Throughput stays the same (9.4 bit/s as in Table 83), liberating users from correcting errors themselves and therefore keeping the information transmission efficiency at a constant rate.

16.3.4 *Summary*

These simulations demonstrate that the information-theoretic measures provide consistent results compared with the conventional text entry measurements. When there are no errors, the information-theoretic notion of throughput TP_i illustrates the speed dimension, similar to words per minute (WPM). When errors are involved, TP_i accounts for the randomness of errors, which is not captured by the classical error rate. The information-theoretic measures also provide a way to examine the effect of auto-correction, namely that having an incorrect auto-correction is more costly than correcting errors manually while having a good decoder keeps the text entry rate constant. By contrast, the other two definitions of throughput do not account for the randomness of errors, nor auto-correction in the loop, nor the speed-accuracy tradeoff, as will be demonstrated in the next section.

16.4 PILOT STUDY

To investigate the information-theoretic measures on text entry in a more realistic setting, I recruited nine participants, age between 24 and 33 (mean = 27.2, σ = 2.1) from our institution. All of them were right-handed, interacted with WIMP interfaces regularly and were familiar with auto-correction methods. The pilot study was conducted on a Macbook Pro with a 2.7 GHz processor and 8 GB RAM. The program (Fig. 78) was implemented in Python 3.6 using the Tkinter GUI toolkit [3]. The same language model L and the keyboard with regular-sized keys (Fig. 82 (b)) were used to allow the comparison between the results from this study and those of the simulations. A standard mouse was used with the same sensitivity for all participants.

Three conditions were considered:

1. No auto-correction. Intended sentence: “ad cb aac” (as in Fig. 84);
2. A good decoder that corrects errors correctly. Intended sentence: “cba dda b” where all three ‘words’ will be correctly corrected if wrongly transcribed;
3. A bad decoder that corrects the right characters wrongly. Intended sentence: “b cd aadc” where the last ‘word’ will be incorrectly corrected even if correctly transcribed.

Before each condition, participants spent some time familiarizing themselves with the interface. They were then instructed to transcribe as fast and as accurately as possible. Note that unlike most text entry experiments, e. g. Feit et al. [53], we do not control error rate, so as to assess the speed-accuracy tradeoff.

---

[3] `https://docs.python.org/3/library/tk.html`

Similar to the simulations, we logged the time for each key press and the cursor coordinates to compute the conventional text entry measures as well as the Fitts' law constants.

16.4.1 *Results*

Table 13 shows the results of the pilot study. Overall, participants took more time transcribing sentences (9.1 seconds in the pilot study vs. 4.4 in the simulations when auto-correction is not involved), most likely due to visual search. In terms of total time, similar to the simulations, a good decoder saves time (8.6 seconds) and a bad decoder costs more time (11.2 seconds) than having no auto-correction (9.1 seconds).

	No Auto-correction	*Good Decoder*	*Bad Decoder*
Total Time (s)	9.1	8.6	11.2
Input Entropy (bits)	32.8	32.8	32.8
Transmitted Information (bits)	32.8	32.8	29.1
Transmitted Information (%)	1.0	1.0	0.88
Throughput (bit/s)	3.6	3.8	2.6
Words Per Minute	10.5	11.1	8.6
KSPC	1.12	1.06	1.54
Error rate (%)	0.05	0	0.14
TP_m (bit/s)	2.3	2.3	3.2
TP_z (bit/s)	1.8	2.8	4.0

Table 13: A summary of different measures of the pilot study.

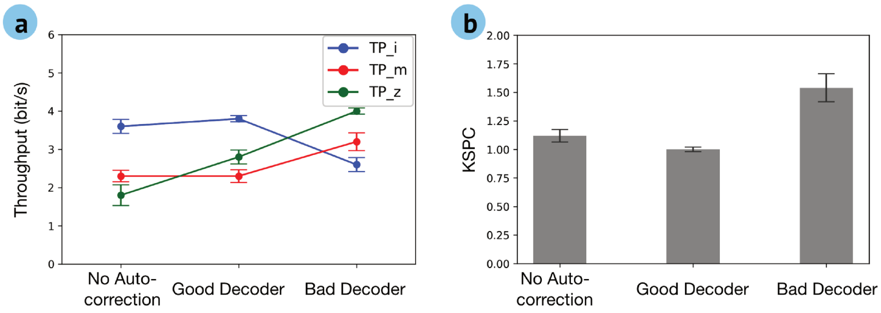


Figure 86: (a) Comparison among three measures of throughput and (b) KSCP in the pilot study.

Regarding errors, participants made few errors in all three conditions, but these errors were treated differently.

In the first condition, when auto-correction is not involved, participants corrected incorrectly transcribed characters (corrected error rate 5%), resulting in transmitted information equal to input entropy. Since there are corrections, which usually involves correcting the incorrect character only, KSPC is greater than but close to 1. In the second condition, when a good decoder is in the loop, participants did not need to correct errors themselves. However one participant typed a wrong character, which was automatically corrected, but he then deleted the correction and retyped it (KSPC averages 1.06). In the third condition, when a bad decoder is used, the error rate is the highest (14%). This is due to both uncorrected errors and corrected errors. Seven participants saw the incorrect corrections, corrected them and then retyped the correct character. Two participants even deleted and retyped the whole word (KSPC averages 1.54) (Fig. 86 (b)). Another two participants did not see the incorrect auto-correction, so the errors were left in the final transcribed sentence, leading to uncorrected errors.

This is also reflected in the information-theoretic measures. When a good decoder or no decoder is involved, the transmitted information equals input entropy as all errors have been corrected. This is in contrast to the condition when a bad decoder is involved: only 88% information is successfully transmitted. A good decoder also saves error correction time, therefore has the highest throughput than no auto-correction condition, which has higher information transmission efficiency than when a bad decoder is in the loop. These results echo the previous simulations where correcting auto-correction is very costly. On the other hand, the other two definitions of throughput exhibit opposite behaviors (Fig. 86 (a)). As in Chapter 15, this is due to the fact that pointing is no longer a dominating element in this task, therefore the two pointing-oriented definitions of throughput can no longer capture the information transmission efficiency.

16.5 SUMMARY

This is but a first step to demonstrate how the information-theoretic measures can be applied in the context of text entry, which is one of the most common but complex tasks that we encounter on a regular basis. I showed how to compute each measure using conditional entropy based on a language model, compared these measures with conventional text entry measures as well as the two definitions of throughput, and discussed the effect of auto-correction on text entry. Results from simulations as well as the pilot study show several benefits of using the information-theoretic measures to characterize text entry:

- The information-theoretic measures show consistent results compared to the conventional text entry measurements. Also, the computation of these measures is easy due to the availability of statistical language processing;
- Using the information-theoretic notion of throughput, researchers do not need to deliberately control error rate when running experiments, which offers a more holistic picture of the speed-accuracy tradeoff;
- Using equivocation, we can also better exploit the distribution of errors, and by understanding it, we can improve the text entry rate by designing better auto-correction and other decoding approaches.

The studies reported in this chapter are preliminary and have several limitations. More work is needed, for instance, to investigate the role of the language model in estimating input entropy and how to design more intelligent decoders by incorporating equivocation. In the next chapter, I discuss these limitations and outline future opportunities using the information-theoretic measures.

17

DICUSSION AND FUTURE PERSPECTIVES

In this chapter, I discuss the two use cases in Chapter 15 and Chapter 16 respectively, summarizing their advantages as well as limitations. I then outline opportunities for future work using information-theoretic measures to characterize human-computer interaction.

17.1 COMMAND SELECTION & TEXT ENTRY

In order to compute the information-theoretic measures, we need to have a set of intended inputs X associated with a probability distribution $p(X)$ describing the use of each input, as well as the actual inputs Y received by the computer and the time T to transmit these inputs (messages). As discussed in Chapter 15, command selection provides a straightforward scenario to examine these measures. It does not require measuring more than what we normally measure: a series of experimental stimuli, the actual user inputs and the time to complete the task. However, information theory lets us explore the interaction task and interpret the results in terms of communication efficiency. For speed, we use the traditional measure; for error, we look at equivocation & information transmission percentage instead of error rate; for speed-accuracy tradeoff, we look at throughput (information transmission efficiency). These measures had not been systematically considered before.

As shown in Chapter 15, equivocation tells us not only that there are errors in the communication channel, but also how users make errors, which is tightly related to the experimental setting and the instruction given to the participants. Throughput provides information about the rate at which information can be transmitted using a certain input technique and/or interface. For instance, in Fig. 71 (a), the information-theoretic notion of throughput shows that the semantic organization of a 16-item menu leads to a higher information transmission rate than an alphabetic organization, whereas in Fig. 71 (b), the traditional time measure shows the same effect for both organizations. In this case, throughput can help interaction designers make more informed decisions. Additionally, throughput provides a bigger picture of the speed-accuracy tradeoff, helping us better understand human behavior in various conditions.

Text entry in Chapter 16 is another scenario that benefits from applying the information-theoretic measures. First, Shannon himself measured the entropy of the English language right after the introduction of information theory; Second, it is a tradition to consider text entry as communication and language as information (bits); Third, the redundancy in languages naturally provides a probability distribution; Fourth, given the complexity of measuring text entry, a common practice for text entry experiments is to control error rate under, e. g. 4%, therefore only the speed dimension is considered when evaluating new text entry techniques or intelligent decoding methods. In this respect, information theory seems appropriate to measure text entry performance. Chapter 16 demonstrates how to use conditional entropy extracted from a language model to measure input entry, as well as the transmitted information to compute throughput. With the 7-key keyboard, I explored various characteristics of these measures, demonstrating their coherence compared to conventional text entry measures and their benefits over the other two definitions of throughput.

There are certainly many other aspects of text entry that I did not cover. First it should be applied to real text entry experiments comparing the information-theoretic measures with conventional metrics without controlling error rate. To do so, we need to use a real language model and a more sophisticated language modeling scheme. For instance, Weir et al. used a language model trained from Twitter, which has 94.6 M sentences, 626 M words, and 2.56 G characters, as well as a character-based 7-gram language modeling scheme [212]. Furthermore, as mentioned in Part i, Chapter 5, there are at least 4 types of errors in text entry. I only considered the case of substitution and have not yet explored how omission, insertion or transposition affect the information-theoretic measures. Finally and perhaps most importantly, since equivocation provides information about how errors are made, it can certainly be used to improve intelligent language decoding such as auto-correction and auto-completion.

17.2 FUTURE WORK

Command selection and text entry are merely two examples among many where we can apply the information-theoretic measures to characterize the interaction task. I demonstrated its advantages over conventional metrics and I hope that it will inspire researchers to study other tasks. Having a standardized language to examine interaction across devices, platforms, user groups and domain would indeed be most beneficial to the community at large.

We can consider any type of interaction is a way for the user to transmit information to the computer. Similar to all communications, the goal is to improve the communication rate, to have more information delivered and to reduce the other party's (here the computer) uncertainty about what we want to express. Information theory provides a way to systematically investigate and characterize interaction [1]. There is still a lot we can explore, e. g. how to model continuous inputs such as gestures and how to model intended inputs at all if they are not so straightforward to model. For instance, how do we know what users want to write in daily life? Additionally, what is the relationship between transmitted information percentage $I(X;Y)/H(X)$ and the commonly defined accuracy rate $1 - P_e$? Can they be rationalized using other information-theoretic concepts in addition to Fano's inequality? To have closed-loop interaction, rather than the current one-directional scheme, how can we involve feedback, which plays a key role in determining the user's subsequent input? The full spectrum of conceptual and practical benefits of Shannon's information theory seems to have a lot to offer to HCI. I hope the communication standpoint supported by ACM SIGCHI Curriculum can truly become human-computer interaction guidelines.

"If you just communicate, you can get by.
But if you communicate skillfully, you can work miracles."
- Jim Rohn

---

[1] We provide the package to compute these information-theoretic measures at `https://github.com/wanyuliu/Information-Theoretic-Metrics/`.

CONCLUSION

This thesis strives to bridge the gap between information theory and HCI by taking the stance that human-computer interaction can be considered as a communication process and therefore be characterized using information-theoretic concepts. While the notions that humans are capable of transmitting and exchanging information, that we are bound by some information capacity, etc. have been around even before information theory was introduced, to the best of my knowledge, this thesis is the first attempt at clarifying the scientific position of information theory in HCI.

In Part i, I introduced some basic information theory concepts and a historical walkthrough of how information theory influenced experimental psychology and HCI. It is rather interesting today to see how experimental psychologists were swept by information theory back in the 1950s. Indeed, information is everywhere and is vital for many aspects of our lives. One can find many quotes about how information is the key, everything is achievable when we are armed with information. So think about what can be done when information is quantifiable. In the 1950s, psychologists were fascinated by the ability to quantitatively measure how we transmit/process information. However, information, or rather entropy, which is a measure of randomness in information theory, has confused many brilliant minds at the time. Information in information theory has absolutely no semantic meaning and is entirely computed by the probability distribution of a random variable.

Among many applications from psychology, Fitts' law and Hick's law are some of the most important and relevant ones to human-computer interaction. Fitts' law has been used to model pointing behavior, one of the most prevalent interactions, and Hick's law has been used to model reaction time in response to a number of stimuli. While these two laws are the closest link between information theory and HCI, both are subject to several misunderstandings. In Part i, I showed that there are still a number of remaining questions related to Fitts' law after 60 years of empirical validation and it is necessary to rigorously examine them from an information-theoretic perspective. Moreover, the contradictory position of Hick's law is probably not due to the fact that the law is wrong, but to how we understand and implement it. I showed that it has been used in many different contexts where it is not supposed to be at work. More importantly, rather than stamping a phenomenon with Hick's name, we can better benefit from having a well-defined taxonomy where different types of measured time are described using correctly formulated mathematical representations.

Chapters 1, 2, 3 and 4 in Part i explained the elements of information theory and how they were understood and applied in experimental psychology, with the goal of understanding information theory and its applications from a historical perspective. By doing so, we can learn from prior studies and avoid making the same mistakes. By examining how relevant Hick's law is to HCI in Chapter 4, I illustrated that it is important to understand what we are measuring when we measure time in controlled experiments. I showed that several time measures often overlap and it is difficult to tear them apart. Therefore, we should be careful when describing time measures in experiments. Chapter 5 examined the recent studies where information theory plays a role with the goal of inspiring future work. Finally, I believe that there is a great deal to learn from psychology, namely the paradigm of stimulus-response compatibility. Since psychology is tightly related to human-computer interaction and we cannot design or study new interfaces or interaction techniques without understanding users, we should probably take better advantage of what has been learned by Psychology and how it can be applied in HCI.

After clarifying the *indirect* link between information theory and HCI in Part i, Part ii and Part iii strived to provide novel perspectives on the *direct* link between information theory and HCI. Part ii introduces a Bayesian Information Gain framework based on Bayesian Experimental Design using the criterion of mutual information to quantify the information sent by the user to the computer and reduce the computer's uncertainty about the user's goal. Information is defined in terms of the computer's knowledge about what the user wants. At the beginning of the interaction, the user has a goal and the computer has some uncertainty about that goal. This uncertainty is represented by the computer's prior knowledge, expressed in a probabilistic model. When providing feedback to the user, the computer takes user input and updates its knowledge about what the user is looking for. Therefore, the information carried by the user input is defined as the knowledge gained by the computer about the user's goal.

I showed two applications of BIG to multiscale navigation (BIGnav, Chapter 9) and hierarchical file retrieval (BIGFile, Chapter 11). In both cases, one can compute the information gained by the computer even when using conventional interaction techniques. I showed that often times the input sent by the user does not carry information to reduce the computer's uncertainty. By presenting the user with a view where the expected information gain is maximized (in the case of BIGnav) or leveraged (in the case of BIGFile), the computer can gain much more information from each user input, therefore locating what the user is looking for with a higher rate of information gain.

Bayesian Information Gain is a general framework that can be applied to many interaction tasks. In order to compute *information*, one needs to model Θ, which represents the potential intended targets; $p(\Theta)$, which represents the computer's prior knowledge about the user's goal; X, which represents the possible system feedback and Y, which represents the possible user input. Often times these three random variables take values in a very large set, therefore the cost of computing the optimal feedback is potentially very high. I showed that discretizing the system feedback Y as well as user input Y reduced the sets, and I introduced a suboptimal algorithm to make the computation tractable. I also proposed several ideas to make interaction more comfortable and intuitive so as to increase both communication efficiency and usability.

Conceptually speaking, BIG is related to several other frameworks such as mixed initiative systems, human-computer partnerships, and co-adaptation. BIG enables collaboration between the user and the computer to achieve shared goals and this collaboration is not possible without contributions from both parties. In line with all these approaches that suggest that great opportunities lie between systems that provide automatic services and systems where users are in full control, BIG combines human intelligence with machine power to reinvent interaction.

BIG is also an instance of probabilistic interface. Similar to other probabilistic interface architectures that treat user input as an uncertain process, BIG uses a user behavior model to represent the ambiguity of user input for the computer. In conventional settings, there is no information-theoretic uncertainty for the user regarding the computer's behavior. However, when presenting a feedback to the user, the computer is uncertain about what the user will do. BIG offers non-deterministic system feedback and user interfaces that leverage probabilistic models to better infer user intent by maximizing/leveraging the expected information gain. It opens the door to a wide range of "BIG" applications and also a new era of probabilistic interaction.

Part iii extends these concepts and applies information theory to characterize an interaction task at large. It is based on the idea that since the user sends information to the computer, we can use information-theoretic concepts to describe the maximum amount of information that can be transmitted, the actual amount of information that gets transmitted and the transmission efficiency. The information-theoretic measures provide several advantages over the conventional speed and accuracy metrics:

1. They allow HCI researchers to investigate the full spectrum of speed and accuracy without deliberately controlling one dimension due to the lack of tool to combine the two (throughput);
2. The information-theoretic notion of equivocation provides much more information about how errors are made in the interaction, which can help design intelligent error correction approaches.

I demonstrated the advantages of using the information-theoretic measures in command selection (Chapter 15) and text entry (Chapter 16). Command selection provides a straightforward perspective on how to compute X: The intended input, which is the possible commands; $p(X)$: The probability distribution of these commands describing their history of use; Y: The actual input, which is the actual commands selected by the user; T: The time to select these commands. I illustrated that the information-theoretic notion of throughput is more consistent than the two existing definitions of throughput and can be rationalized using mathematical tools. Since it provides a rigorous way to combine speed with accuracy, it can help make more informed decisions in terms of speed-accuracy tradeoff for interaction techniques and interfaces.

Text entry is another case where it is possible to apply the information-theoretic measures for the reasons that (a) the redundancy in natural languages provides the probability distribution for the input and (b) Text entry behavior is complex and difficult to capture therefore error rate is usually deliberately controlled by the experimenter. With the information-theoretic measures, one does not need to control the error rate but instead can fully explore the interaction between speed and accuracy as well as take further advantage of how errors are made to devise intelligent text entry methods.

Using these information-theoretic measures to characterize interaction also provides a *direct* link between information theory and HCI. The communication between the user and the computer is truly modeled as a channel and we characterize how much information can be transmitted and how much information is actually transmitted. To compute these measures, one does not need to collect more data than is already done. Information transmission efficiency (throughput) provides a standardized way to assess human-computer interaction. Overall, these measures provide a standard language to characterize interaction.

LOOKING FORWARD

This thesis is a first step towards using information theory as a unified tool for understanding and designing human-computer interaction & communication. It is just the tip of the iceberg: there is still a great deal to explore and I hope to have demonstrated that the community can benefit from such theoretically sound methods.

I have demonstrated how we can learn from the past, presented a Bayesian Information Gain framework to quantify the information sent by the user to the computer and illustrated the advantages of using information-theoretic measures to characterize interaction. Several avenues of future work present themselves: applying BIG to the design or redesign of other interactions, refining the framework by incorporating user behavior in real time, and combining BIG with other frameworks to construct the utility function. More generally, I see great potential in exploring and applying the information-theoretic measures to other interaction tasks, particularly those with continuous inputs and taking further advantage of the notion of equivocation to devise more intelligent systems. In short, there is a vast space of open research, and I hope this human-computer communication standpoint can truly become a standard to guide future interaction design.

"My greatest concern was what to call it. I thought of calling it 'information', but the word was overly used, so I decided to call it 'uncertainty'. When I discussed it with John von Neumann, he had a better idea. Von Neumann told me, 'You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, no one really knows what entropy really is, so in a debate you will always have the advantage'."

- Claude Shannon

Part IV

APPENDIX

A

APPENDIX

A.1 BIGNAV EXPERIMENT CONSENT

Information Theory in Understanding and Guiding Multiscale Navigation Experiment

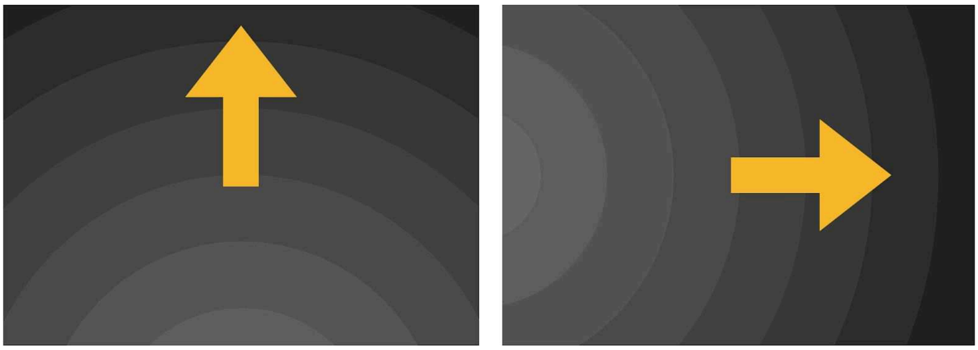
First of all, we express our sincere gratitude to you for participating in this experiment. You can find cookies and candies on the table. Please feel free to serve yourself.

The goal of this study is to gain some insight into the role of information theory in understanding and guiding multiscale navigation. It applies in the scenario where ***users have already decided a destination and navigate to it.*** For instance, navigating to Paris from somewhere else. Different from traditional zooming and panning, it utilizes both human inputs and multiscale world characteristics to enhance navigation.

There are two sessions in this experiment -- *Calibration* and *Controlled Study*.

Calibration:

The goal of this session is to better understand human behavior when zooming and panning in multiscale world. We are particularly interested in ***your perception regarding direction***. For example, if you are aware that ***your destination*** -- **a red dot**, indicated by concentric rings, is on the right, what are the chances that you will go to right whereas what are the chances you will go to other directions due to misjudge, motor error, etc. We will then use the result to tune parameters in the real experiment. Sample trials are as below:



Controlled Study:

In this controlled study, your task is to navigate to the ***target that is in red*** from certain distance to completely zoomed-in level and ***right click*** on it. Concentric circles are placed around the target to always indicate the direction. You will encounter two conditions that represent conventional zooming and panning, and information theoretic navigation respectively. At the end of the experiment, you will be asked for the preference toward these two techniques. There will be 300 trials in total, taking approximately 60 minutes. **PLEASE THINK LOUD** during the experiment. You can terminate it if you do not feel comfortable at any moment. Do not hesitate to ask questions to experimenter if you have any. If you are ready, please sign the consent form below and we can start the experiment.

Consent Form

I agree to participate in the user study conducted by the Human-Computer Interaction group at Telecom ParisTech.

I understand that participation in this study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Participate ID: ____

Participant Signature: ____

Date: ____

A.2 BIGNAV POST-EXPERIMENT QUESTIONNAIRE

Post Experiment Questionnaire

Thank you again for participating in the experiment. Please take a few minutes to fill out the survey. We appreciate your feedback.

* 1. Participant ID: ____

* 2. How would you evaluate your performance in two conditions?

- Standard navigation

Bad				Good
1	2	3	4	5
☐	☐	☐	☐	☐

- Information theoretic navigation

Bad				Good
1	2	3	4	5
☐	☐	☐	☐	☐

* 3. How would you evaluate your comfort level in two conditions?

- Standard navigation

Uncomfortable				Comfortable
1	2	3	4	5
☐	☐	☐	☐	☐

- Information theoretic navigation

Uncomfortable				Comfortable
1	2	3	4	5
☐	☐	☐	☐	☐

* 4. Which technique would you prefer?

- Standard navigation ☐
- Information theoretic navigation ☐

* 5. Why do you prefer this technique?

____

* 6. How would you like to have the information theoretic navigation differently? Do you have any other comments and thoughts?

____

A.3 BIGFILE EXPERIMENT CONSENT

Study: ***BIGfile: Designing Adaptive File Interfaces*** **Participant ID:** *P-*

1. Purpose of the study: We want to compare three different interfaces for file retrieval and see which one is faster and which one is liked the best.

2. Procedures to be followed: If you agree to participate, we will introduce the three interfaces to you and then ask you to perform a series of tasks. You will begin by filling out a short questionnaire about your experience with navigation-based file retrieval. Next, we will ask you to sit in front of a laptop and find a comfortable position to perform the task. We will show you how the interfaces look like and let you try them out. For each interface, we will display a stimulus which indicates a folder or a file and ask you to navigate to this target accordingly. At the end, we will ask you to fill out another short questionnaire and ask you your opinions about the interfaces. We will record the time you spend to find each target, as well as your cursor movements and eye movements.

3. Risks and Discomforts: We do not believe you will experience any risks by participating in this study, beyond those you encounter in daily life. However, you may find the tasks to be repetitive, slightly tiring, uncomfortable or boring.

4. Benefits: You may find the interfaces useful for accessing your files and folders.

5. Duration: The complete study, including pre- and post-questionnaires will take about 60 minutes.

6. Confidentiality: Your participation in this research is confidential and your data will be anonymized. This paper form is the only link between your name and your participant ID, and will be stored in a locked office. The researchers listed below will be able to review only the anonymized data, which will not include any personally identifiable information.

7. Right to ask questions: Please contact *Wanyu Liu (wanyu.liu@telecom-paristech.fr)* with questions, complaints or concerns about this research. Questions about research procedures and future publications will be answered by the research team. You may also contact the Inria COERLE ethics board if you feel this study has harmed you in any way: *http://www.inria.fr/institut/organisation/instances/coerle/composition*

8. Voluntary Participation: Your decision to be in this research is entirely voluntary. You do not have to answer the questions, and you may ask us to withdraw your data from the analysis, up to two months after the study. You may stop at any time, without giving a reason, and there is no penalty for withdrawal. You will not be paid for taking part in this study.

9. CONSENT: You must be at least 18 years to participate in the study. If you agree with the above, please sign your name and indicate the date. You may ask for a copy of this consent form.

Contact:

Principal investigator:	Michel Beaudouin-Lafon - *mbl@lri.fr*	Professor
Co-investigator:	Wendy Mackay - *mackay@lri.fr*	Professor
Co-investigator:	Joanna McGrenere - joanna@cs.ubc.ca	Professor
Key Contact:	Wanyu Liu - *wanyu.liu@telecom-paristech.fr*	PhD Student

The nature and purpose of this research have been sufficiently explained and I agree to participate in this study. I understand that I am free to withdraw at any time without incurring any penalty.

Participant's name *(please print)* Signature Date

Researcher's name *(please print)* Signature Date

A.4 BIGFILE POST-EXPERIMENT QUESTIONNAIRE

BIGFile Questionnaire

* Required

1. Participant ID (Ask the study coordinator) *

____

2. Age *

____

3. Gender *

○ Female
○ Male

4. I use a computer… *

○ Never
○ Monthly
○ Weekly
○ Daily
○ Multiple times per day

5. I use Finder… *

○ Never
○ Monthly
○ Weekly
○ Daily
○ Multiple times per day

Interface 1

For the following questions, please choose a number from 1 to 7 to describe your experience with interface 1.

For each question, we would appreciate any additional comments you have in the “Comments” section.

6. Mental Demand *

How mentally demanding was the task?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

7. Physical Demand *

How physically demanding was the task?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

8. Temporal Demand *

How hurried or rushed was the pace of the task?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

9. Performance *

How successful were you in accomplishing what you were asked to do?

1 (Perfect)	2	3	4	5	6	7 (Failure)
○	○	○	○	○	○	○

10. Effort *

How hard did you have to work to accomplish your level of performance?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

11. Frustration *

How insecure, discouraged, irritated, stressed and annoyed were you?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

12. Please provide any additional comments about or reactions to Interface 1: *

____

____

____

____

____

____

Interface 2

For the following questions, please choose a number from 1 to 7 to describe your experience with interface 2.

For each question, we would appreciate any additional comments you have in the “Comments” section.

13. Mental Demand *
How mentally demanding was the task?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

14. Physical Demand *
How physically demanding was the task?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

15. Temporal Demand *
How hurried or rushed was the pace of the task?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

16. Performance *
How successful were you in accomplishing what you were asked to do?

1 (Perfect)	2	3	4	5	6	7 (Failure)
○	○	○	○	○	○	○

17. Effort *

How hard did you have to work to accomplish your level of performance?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

18. Frustration *

How insecure, discouraged, irritated, stressed and annoyed were you?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

19. Please provide any additional comments about or reactions to Interface 2: *

Interface 3

For the following questions, please choose a number from 1 to 7 to describe your experience with interface 3.

For each question, we would appreciate any additional comments you have in the “Comments” section.

20. Mental Demand *
How mentally demanding was the task?

	1 (Very Low)	2	3	4	5	6	7 (Very High)
	○	○	○	○	○	○	○

21. Physical Demand *
How physically demanding was the task?

	1 (Very Low)	2	3	4	5	6	7 (Very High)
	○	○	○	○	○	○	○

22. Temporal Demand *
How hurried or rushed was the pace of the task?

	1 (Very Low)	2	3	4	5	6	7 (Very High)
	○	○	○	○	○	○	○

23. Performance *
How successful were you in accomplishing what you were asked to do?

	1 (Perfect)	2	3	4	5	6	7 (Failure)
	○	○	○	○	○	○	○

24. Effort *

How hard did you have to work to accomplish your level of performance?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

25. Frustration *

How insecure, discouraged, irritated, stressed and annoyed were you?

1 (Very Low)	2	3	4	5	6	7 (Very High)
○	○	○	○	○	○	○

26. Please provide any additional comments about or reactions to Interface 3: *

Overall Preference

27. Rank the interfaces from 1 (Least Preferred) to 3 (Most preferred) *

	1 (Least Preferred)	2	3 (Most Preferred)
Interface 1:	○	○	○
Interface 2:	○	○	○
Interface 3:	○	○	○

28. Please provide any additional comments below.

____

____

____

____

____

____

BIBLIOGRAPHY

[1] *ACM SIGCHI Curricula for Human-Computer Interaction*. 2009. URL: `http://old.sigchi.org/cdg/cdg2.html#2_1`.

[2] Gediminas Adomavicius and Alexander Tuzhilin. "Context-aware recommender systems." In: *Recommender systems handbook*. Springer, 2011, pp. 217–253.

[3] Abu Ali and Andre Liem. "The use of formal aesthetic principles as a tool for design conceptualisation and detailing." In: *DS 81: Proceedings of NordDesign 2014, Espoo, Finland 27-29th August 2014* (2014).

[4] JE Allen, Curry I Guinn, and E Horvtz. "Mixed-initiative interaction." In: *IEEE Intelligent Systems and their Applications* 14.5 (1999), pp. 14–23.

[5] Caroline Appert and Jean-Daniel Fekete. "OrthoZoom Scroller: 1D Multi-scale Navigation." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montreal, Quebec, Canada: ACM, 2006, pp. 21–30. ISBN: 1-59593-372-7. DOI: `10.1145/1124772.1124776`. URL: `http://doi.acm.org/10.1145/1124772.1124776`.

[6] Michael P Arcuri, Thomas Scott Coon, Jeffrey J Johnson, Alexis Warren Jacob Manning, and Martijn Eldert Van Tilburg. *Adaptive menus*. US Patent 6,121,968. 2000.

[7] Fred Attneave. "Applications of information theory to psychology: A summary of basic concepts, methods, and results." In: (1959).

[8] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. "Visual Menu Techniques." In: *ACM Comput. Surv.* 49.4 (Dec. 2016), 60:1–60:41. ISSN: 0360-0300. DOI: `10.1145/3002171`. URL: `http://doi.acm.org/10.1145/3002171`.

[9] Gilles Bailly, Antti Oulasvirta, Duncan P. Brumby, and Andrew Howes. "Model of Visual Search and Selection Time in Linear Menus." In: *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 3865–3874. ISBN: 978-1-4503-2473-1. DOI: `10.1145/2556288.2557093`. URL: `http://doi.acm.org/10.1145/2556288.2557093`.

[10] Maxim Bakaev. "Fitts' law for older adults: considering a factor of age." In: *Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems*. Sociedade Brasileira de Computação. 2008, pp. 260–263.

[11] Deborah Barreau and Bonnie A. Nardi. “Finding and Reminding: File Organization from the Desktop.” In: *SIGCHI Bull.* 27.3 (July 1995), pp. 39–43. ISSN: 0736-6906. DOI: `10.1145/221296.221307`. URL: `http://doi.acm.org/10.1145/221296.221307`.

[12] Benjamin B Bederson and Angela Boltman. “Does animation help users build mental maps of spatial information?” In: *Information Visualization, 1999.(Info Vis’ 99) Proceedings. 1999 IEEE Symposium on*. IEEE. 1999, pp. 28–35.

[13] John Gilbert Beebe-Center, MS Rogers, and DN O’connell. “Transmission of information about sucrose and saline solutions through the sense of taste.” In: *The Journal of Psychology* 39.1 (1955), pp. 157–160.

[14] Edgar Berdahl, Michael Blandino, David Baker, and Daniel Shanahan. “An Approach for Using Information Theory to Investigate Continuous Control of Analog Sensors by Humans.” In: *Proceedings of the Audio Mostly 2016*. AM ’16. Norrkoping, Sweden: ACM, 2016, pp. 85–90. ISBN: 978-1-4503-4822-5. DOI: `10.1145/2986416.2986450`. URL: `http://doi.acm.org/10.1145/2986416.2986450`.

[15] Ofer Bergman, Ruth Beyth-Marom, Rafi Nachmias, Noa Gradovitch, and Steve Whittaker. “Improved Search Engines and Navigation Preference in Personal Information Management.” In: *ACM Trans. Inf. Syst.* 26.4 (Oct. 2008), 20:1–20:24. ISSN: 1046-8188. DOI: `10.1145/1402256.1402259`. URL: `http://doi.acm.org/10.1145/1402256.1402259`.

[16] Ofer Bergman, Steve Whittaker, Mark Sanderson, Rafi Nachmias, and Anand Ramamoorthy. “The Effect of Folder Structure on Personal File Navigation.” In: *J. Am. Soc. Inf. Sci. Technol.* 61.12 (Dec. 2010), pp. 2426–2441. ISSN: 1532-2882. DOI: `10.1002/asi.v61:12`. URL: `http://dx.doi.org/10.1002/asi.v61:12`.

[17] Ofer Bergman, Steve Whittaker, Mark Sanderson, Rafi Nachmias, and Anand Ramamoorthy. “How Do We Find Personal Files?: The Effect of OS, Presentation & Depth on File Navigation.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’12. Austin, Texas, USA: ACM, 2012, pp. 2977–2980. ISBN: 978-1-4503-1015-4. DOI: `10.1145/2207676.2208707`. URL: `http://doi.acm.org/10.1145/2207676.2208707`.

[18] Paul Bertelson. “Sequential redundancy and speed in a serial two-choice responding task.” In: *Quarterly Journal of Experimental Psychology* 13.2 (1961), pp. 90–102.

[19] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. "Semantic Pointing: Improving Target Acquisition with Control-display Ratio Adaptation." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: ACM, 2004, pp. 519–526. ISBN: 1-58113-702-8. DOI: 10.1145/985692.985758. URL: http://doi.acm.org/10.1145/985692.985758.

[20] G Blank. "Brauchbarkeit optischer Reaktionsmessungen (Usability of optical reaction measurements)." In: *Indust. Psychotech.* 11 (1934), pp. 140–150.

[21] Richard Boardman and M. Angela Sasse. ""Stuff Goes into the Computer and Doesn'T Come out": A Cross-tool Study of Personal Information Management." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: ACM, 2004, pp. 583–590. ISBN: 1-58113-702-8. DOI: 10.1145/985692.985766. URL: http://doi.acm.org/10.1145/985692.985766.

[22] Robert W Brainard, Thomas S Irby, Paul M Fitts, and Earl A Alluisi. "Some variables influencing the rate of gain of information." In: *Journal of experimental psychology* 63.2 (1962), p. 105.

[23] ID Brown. "Many messages from few sources." In: *Ergonomics* 3.2 (1960), pp. 159–168.

[24] Peter F Brown, Vincent J Della Pietra, Robert L Mercer, Stephen A Della Pietra, and Jennifer C Lai. "An estimate of an upper bound for the entropy of English." In: *Computational Linguistics* 18.1 (1992), pp. 31–40.

[25] Andrew M Bruckner, E Ostrow, et al. "Some function classes related to the class of convex functions." In: *Pacific Journal of Mathematics* 12.4 (1962), pp. 1203–1215.

[26] Stuart K Card, William K English, and Betty J Burr. "Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT." In: *Ergonomics* 21.8 (1978), pp. 601–613.

[27] David Carmel, Liane Lewin-Eytan, Alex Libov, Yoelle Maarek, and Ariel Raviv. "Promoting Relevant Results in Time-Ranked Mail Search." In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 1551–1559. ISBN: 978-1-4503-4913-0. DOI: 10.1145/3038912.3052659. URL: https://doi.org/10.1145/3038912.3052659.

[28] Stephen J Carver. "Integrating multi-criteria evaluation with geographical information systems." In: *International Journal of Geographical Information System* 5.3 (1991), pp. 321–339.

[29] Mozilla Developer Center. *The Places frecency algorithm*. 2008.

[30] Hock Chuan Chan, Suparna Goswami, and Hee-Woong Kim. "An alternative fit through problem representation in cognitive fit theory." In: *Journal of Database Management (JDM)* 23.2 (2012), pp. 22–43.

[31] Olivier Chapuis, Renaud Blanch, and Michel Beaudouin-Lafon. "Fitts' law in the wild: A field study of aimed movements." In: (2007). URL: `http://insitu.lri.fr/~chapuis/publications/RR1480.pdf`.

[32] Andy Cockburn and Carl Gutwin. "A predictive model of human performance with scrolling and hierarchical lists." In: *Human–Computer Interaction* 24.3 (2009), pp. 273–314.

[33] Andy Cockburn, Carl Gutwin, and Saul Greenberg. "A Predictive Model of Menu Performance." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: ACM, 2007, pp. 627–636. ISBN: 978-1-59593-593-9. DOI: `10.1145/1240624.1240723`. URL: `http://doi.acm.org/10.1145/1240624.1240723`.

[34] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[35] ERFW Crossman. "Entropy and choice time: The effect of frequency unbalance on choice-response." In: *Quarterly Journal of Experimental Psychology* 5.2 (1953), pp. 41–51.

[36] ERFW Crossman. "The speed and accuracy of simple hand movements." In: *The nature and acquisition of industrial skills* (1957).

[37] Edward RFW Crossman. "The measurement of discriminability." In: *Quarterly Journal of Experimental Psychology* 7.4 (1955), pp. 176–195.

[38] Edward Cutrell, Susan T. Dumais, and Jaime Teevan. "Searching to Eliminate Personal Information Management." In: *Commun. ACM* 49.1 (Jan. 2006), pp. 58–64. ISSN: 0001-0782. DOI: `10.1145/1107458.1107492`. URL: `http://doi.acm.org/10.1145/1107458.1107492`.

[39] Edward Cutrell, Daniel Robbins, Susan Dumais, and Raman Sarin. "Fast, Flexible Filtering with Phlat." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montreal, Quebec, Canada: ACM, 2006, pp. 261–270. ISBN: 1-59593-372-7. DOI: `10.1145/1124772.1124812`. URL: `http://doi.acm.org/10.1145/1124772.1124812`.

[40] Fred J Damerau. "A technique for computer detection and correction of spelling errors." In: *Communications of the ACM* 7.3 (1964), pp. 171–176.

[41] Keith Devlin. “Claude Shannon, 1916–2001.” In: *Focus: The Newsletter of the Mathematical Association of America* 21.5 (2001), pp. 20–21.

[42] Franciscus Cornelis Donders. “Die schnelligkeit psychischer processe (On the speed of mental processes).” In: (1868).

[43] Paul Dourish, W. Keith Edwards, Anthony LaMarca, John Lamping, Karin Petersen, Michael Salisbury, Douglas B. Terry, and James Thornton. “Extending Document Management Systems with User-specific Active Properties.” In: *ACM Trans. Inf. Syst.* 18.2 (Apr. 2000), pp. 140–170. ISSN: 1046-8188. DOI: 10.1145/348751.348758. URL: http://doi.acm.org/10.1145/348751.348758.

[44] Heiko Drewes. “Only One Fitts’ Law Formula Please!” In: *CHI ’10 Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’10. Atlanta, Georgia, USA: ACM, 2010, pp. 2813–2822. ISBN: 978-1-60558-930-5. DOI: 10.1145/1753846.1753867. URL: http://doi.acm.org/10.1145/1753846.1753867.

[45] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. “Stuff I’Ve Seen: A System for Personal Information Retrieval and Re-use.” In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. SIGIR ’03. Toronto, Canada: ACM, 2003, pp. 72–79. ISBN: 1-58113-646-3. DOI: 10.1145/860435.860451. URL: http://doi.acm.org/10.1145/860435.860451.

[46] Connie C Duncan-Johnson and Emanuel Donchin. “On quantifying surprise: The variation of event-related potentials with subjective probability.” In: *Psychophysiology* 14.5 (1977), pp. 456–467.

[47] Peter Elias. “Two famous papers.” In: *IRE Transactions on Information Theory* 4.3 (1958), p. 99.

[48] David Elsweiler, Morgan Harvey, and Martin Hacker. “Understanding Re-finding Behavior in Naturalistic Email Interaction Logs.” In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’11. Beijing, China: ACM, 2011, pp. 35–44. ISBN: 978-1-4503-0757-4. DOI: 10.1145/2009916.2009925. URL: http://doi.acm.org/10.1145/2009916.2009925.

[49] Douglas C Engelbart. “Augmenting human intellect: A Conceptual Framework.” In: (1962).

[50] Charles W Eriksen and Harold W Hake. “Absolute judgments as a function of stimulus range and number of stimulus and response categories.” In: *Journal of Experimental Psychology* 49.5 (1955), p. 323.

[51] Mark Evers, Mariët Theune, and Joyce Karreman. "Which Way to Turn?: Guide Orientation in Virtual Way Finding." In: *Proceedings of the Workshop on Embodied Language Processing*. EmbodiedNLP '07. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 25–32. URL: `http://dl.acm.org/citation.cfm?id=1610065.1610069`.

[52] Gustav Theodor Fechner. "Elements of Psychophysics (translated by Adler HE; edited by Howes DH, Boring EG)." In: *New York, Holt, Rinehart and Winston* (1966).

[53] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. "How We Type: Movement Strategies and Performance in Everyday Typing." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, 2016, pp. 4262–4273. ISBN: 978-1-4503-3362-7. DOI: `10.1145/2858036.2858233`. URL: `http://doi.acm.org/10.1145/2858036.2858233`.

[54] Jean-Daniel Fekete, Niklas Elmqvist, and Yves Guiard. "Motion-pointing: Target Selection Using Elliptical Motions." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: ACM, 2009, pp. 289–298. ISBN: 978-1-60558-246-7. DOI: `10.1145/1518701.1518748`. URL: `http://doi.acm.org/10.1145/1518701.1518748`.

[55] Leah Findlater and Joanna McGrenere. "A Comparison of Static, Adaptive, and Adaptable Menus." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: ACM, 2004, pp. 89–96. ISBN: 1-58113-702-8. DOI: `10.1145/985692.985704`. URL: `http://doi.acm.org/10.1145/985692.985704`.

[56] Leah Findlater and Joanna McGrenere. "Impact of Screen Size on Performance, Awareness, and User Satisfaction with Adaptive Graphical User Interfaces." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, 2008, pp. 1247–1256. ISBN: 978-1-60558-011-1. DOI: `10.1145/1357054.1357249`. URL: `http://doi.acm.org/10.1145/1357054.1357249`.

[57] Leah Findlater and Joanna McGrenere. "Beyond Performance: Feature Awareness in Personalized Interfaces." In: *Int. J. Hum.-Comput. Stud.* 68.3 (Mar. 2010), pp. 121–137. ISSN: 1071-5819. DOI: `10.1016/j.ijhcs.2009.10.002`. URL: `http://dx.doi.org/10.1016/j.ijhcs.2009.10.002`.

[58] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. "Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

CHI ’09. Boston, MA, USA: ACM, 2009, pp. 1655–1664. ISBN: 978-1-60558-246-7. DOI: 10.1145/1518701.1518956. URL: http://doi.acm.org/10.1145/1518701.1518956.

[59] Stephen Fitchett and Andy Cockburn. “AccessRank: Predicting What Users Will Do Next.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’12. Austin, Texas, USA: ACM, 2012, pp. 2239–2242. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208380. URL: http://doi.acm.org/10.1145/2207676.2208380.

[60] Stephen Fitchett and Andy Cockburn. “An empirical characterisation of file retrieval.” In: *International Journal of Human-Computer Studies* 74.Supplement C (2015), pp. 1 –13. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2014.10.002. URL: http://www.sciencedirect.com/science/article/pii/S107158191400127X.

[61] Stephen Fitchett, Andy Cockburn, and Carl Gutwin. “Improving Navigation-based File Retrieval.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’13. Paris, France: ACM, 2013, pp. 2329–2338. ISBN: 978-1-4503-1899-0. DOI: 10.1145/2470654.2481323. URL: http://doi.acm.org/10.1145/2470654.2481323.

[62] Stephen Fitchett, Andy Cockburn, and Carl Gutwin. “Finder Highlights: Field Evaluation and Design of an Augmented File Browser.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 3685–3694. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557014. URL: http://doi.acm.org/10.1145/2556288.2557014.

[63] Paul M Fitts. “The information capacity of the human motor system in controlling the amplitude of movement.” In: *Journal of experimental psychology* 47.6 (1954), p. 381.

[64] Paul M Fitts. “Cognitive aspects of information processing: III. Set for speed versus accuracy.” In: *Journal of experimental psychology* 71.6 (1966), p. 849.

[65] Paul M Fitts and James R Peterson. “Information capacity of discrete motor responses.” In: *Journal of experimental psychology* 67.2 (1964), p. 103.

[66] Paul M Fitts, James R Peterson, and Gerson Wolpe. “Cognitive aspects of information processing: II. Adjustments to stimulus redundancy.” In: *Journal of Experimental Psychology* 65.5 (1963), p. 423.

[67] Paul M Fitts and Michael I Posner. “Human performance.” In: (1967).

[68] James D Foley. "A. vanDam, SK Feiner, and JF Hughes." In: *Computer Graphics: Principles and Practice* (1990).

[69] Eric Freeman and David Gelernter. "Lifestreams: A Storage Model for Personal Data." In: *SIGMOD Rec.* 25.1 (Mar. 1996), pp. 80–86. ISSN: 0163-5808. DOI: 10.1145/381854.381893. URL: http://doi.acm.org/10.1145/381854.381893.

[70] George W. Furnas. "Effective View Navigation." In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. CHI '97. Atlanta, Georgia, USA: ACM, 1997, pp. 367–374. ISBN: 0-89791-802-9. DOI: 10.1145/258549.258800. URL: http://doi.acm.org/10.1145/258549.258800.

[71] George W. Furnas and Benjamin B. Bederson. "Space-scale Diagrams: Understanding Multiscale Interfaces." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 234–241. ISBN: 0-201-84705-1. DOI: 10.1145/223904.223934. URL: http://dx.doi.org/10.1145/223904.223934.

[72] Krzysztof Z. Gajos and Krysta Chauncey. "The Influence of Personality Traits and Cognitive Load on the Use of Adaptive User Interfaces." In: *Proceedings of the 22Nd International Conference on Intelligent User Interfaces*. IUI '17. Limassol, Cyprus: ACM, 2017, pp. 301–306. ISBN: 978-1-4503-4348-0. DOI: 10.1145/3025171.3025192. URL: http://doi.acm.org/10.1145/3025171.3025192.

[73] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. "Improving the Performance of Motor-impaired Users with Automatically-generated, Ability-based Interfaces." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, 2008, pp. 1257–1266. ISBN: 978-1-60558-011-1. DOI: 10.1145/1357054.1357250. URL: http://doi.acm.org/10.1145/1357054.1357250.

[74] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. "Exploring the Design Space for Adaptive Graphical User Interfaces." In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '06. Venezia, Italy: ACM, 2006, pp. 201–208. ISBN: 1-59593-353-0. DOI: 10.1145/1133265.1133306. URL: http://doi.acm.org/10.1145/1133265.1133306.

[75] Tinsley A. Galyean. "Guided Navigation of Virtual Environments." In: *Proceedings of the 1995 Symposium on Interactive 3D Graphics*. I3D '95. Monterey, California, USA: ACM, 1995, 103–ff. ISBN: 0-89791-736-7. DOI: 10.1145/199404.199421. URL: http://doi.acm.org/10.1145/199404.199421.

[76] WR Garner. "An informational analysis of absolute judgments of loudness." In: *Journal of experimental psychology* 46.5 (1953), p. 373.

[77] Wendell R Garner. "Uncertainty and structure as psychological concepts." In: (1962).

[78] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. "MyLifeBits: Fulfilling the Memex Vision." In: *Proceedings of the Tenth ACM International Conference on Multimedia*. MULTIMEDIA '02. Juan-les-Pins, France: ACM, 2002, pp. 235–238. ISBN: 1-58113-620-X. DOI: `10.1145/641007.641053`. URL: `http://doi.acm.org/10.1145/641007.641053`.

[79] Daniel J Gonçalves and Joaquim A Jorge. "An empirical study of personal document spaces." In: *DSV-IS*. Springer. 2003, pp. 46–60. DOI: `10.1007/978-3-540-39929-2_4`. URL: `https://doi.org/10.1007/978-3-540-39929-2_4`.

[80] Julien Gori, Olivier Rioul, and Yves Guiard. "Speed-Accuracy Tradeoff: A Formal Information-Theoretic Transmission Scheme (FITTS)." In: (2018).

[81] David M Green, R Duncan Luce, and Joseph E Duncan. "Variability and sequential effects in magnitude production and estimation of auditory intensity." In: *Perception & Psychophysics* 22.5 (1977), pp. 450–456.

[82] Saul Greenberg. "Using unix: Collected traces of 168 users." In: (1988).

[83] Saul Greenberg and Ian H Witten. "Supporting command reuse: empirical foundations and principles." In: *International Journal of Man-Machine Studies* 39.3 (1993), pp. 353–390. DOI: `10.1006/imms.1993.1065`. URL: `https://doi.org/10.1006/imms.1993.1065`.

[84] Stephen Griew. "Information gain in tasks involving different stimulus-response relationships." In: *Nature* 182.4652 (1958), p. 1819.

[85] Mario C Grignetti. "A note on the entropy of words in printed English." In: *Information and Control* 7.3 (1964), pp. 304–306.

[86] Yves Guiard and Michel Beaudouin-Lafon. "Target Acquisition in Multiscale Electronic Worlds." In: *Int. J. Hum.-Comput. Stud.* 61.6 (Dec. 2004), pp. 875–905. ISSN: 1071-5819. DOI: `10.1016/j.ijhcs.2004.09.005`. URL: `http://dx.doi.org/10.1016/j.ijhcs.2004.09.005`.

[87] Yves Guiard, Renaud Blanch, and Michel Beaudouin-Lafon. "Object Pointing: A Complement to Bitmap Pointing in GUIs." In: *Proceedings of Graphics Interface 2004*. GI '04. London, Ontario, Canada: Canadian Human-Computer Communications Society, 2004, pp. 9–16. ISBN: 1-56881-227-2. URL: `http://dl.acm.org/citation.cfm?id=1006058.1006060`.

[88] Yves Guiard and Halla B Olafsdottir. "On the measurement of movement difficulty in the standard approach to Fitts' law." In: *PLoS one* 6.10 (2011), e24389.

[89] Yves Guiard and Olivier Rioul. "A Mathematical Description of the Speed/Accuracy Trade-off of Aimed Movement." In: *Proceedings of the 2015 British HCI Conference*. British HCI '15. Lincoln, Lincolnshire, United Kingdom: ACM, 2015, pp. 91–100. ISBN: 978-1-4503-3643-7. DOI: `10.1145/2783446.2783574`. URL: `http://doi.acm.org/10.1145/2783446.2783574`.

[90] Yves Guiard, Julien Gori, Quentin Roy, and Olivier Rioul. "Not Just Pointing: Shannon's Information Theory as a General Tool for Performance Evaluation of Input Techniques." In: (2018).

[91] Harold W Hake and WR Garner. "The effect of presenting various numbers of discrete steps on scale reading accuracy." In: *Journal of Experimental Psychology* 42.5 (1951), p. 358.

[92] Peter Hall, Claude Heath, and Lizzie Coles-Kemp. "Critical visualization: a case for rethinking how we visualize risk and security." In: *Journal of cybersecurity* 1.1 (2015), pp. 93–108.

[93] Sandra G Hart and Lowell E Staveland. "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research." In: *Advances in psychology* 52 (1988), pp. 139–183. DOI: `10.1016/S0166-4115(08)62386-9`. URL: `https://doi.org/10.1016/S0166-4115(08)62386-9`.

[94] Martin G Helander. *Handbook of human-computer interaction*. Elsevier, 2014.

[95] Hermann Helmholtz. "Ueber die Fortpflanzungsgeschwindigkeit der Nervenreizung." In: *Annalen der Physik* 155.2 (1850), pp. 329–330.

[96] Sarah Henderson and Ananth Srinivasan. "An empirical analysis of personal digital document structures." In: *Human Interface and the Management of Information. Designing Information Environments* (2009), pp. 394–403. DOI: `10.1007/978-3-642-02556-3_45`. URL: `https://doi.org/10.1007/978-3-642-02556-3_45`.

[97] WE Hick. "A simple stimulus generator." In: *Quarterly Journal of Experimental Psychology* 3.2 (1951), pp. 94–95.

[98] William E Hick. "On the rate of gain of information." In: *Quarterly Journal of Experimental Psychology* 4.1 (1952), pp. 11–26.

[99] Errol R Hoffmann. "Which version/variation of Fitts' law? A critique of information-theory models." In: *Journal of motor behavior* 45.3 (2013), pp. 205–215.

[100] Eric Horvitz. "Principles of Mixed-initiative User Interfaces." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '99. Pittsburgh, Pennsylvania, USA: ACM, 1999, pp. 159–166. ISBN: 0-201-48559-1. DOI: `10.1145/302979.303030`. URL: `http://doi.acm.org/10.1145/302979.303030`.

[101] Eric Horvitz. "Uncertainty, action, and interaction: In pursuit of mixed-initiative computing." In: *IEEE Intelligent Systems* 14.5 (1999), pp. 17–20.

[102] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. "The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users." In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1998, pp. 256–265.

[103] Juan Pablo Hourcade, Benjamin B. Bederson, Allison Druin, and François Guimbretière. "Differences in Pointing Task Performance Between Preschool Children and Adults Using Mice." In: *ACM Trans. Comput.-Hum. Interact.* 11.4 (Dec. 2004), pp. 357–386. ISSN: 1073-0516. DOI: `10.1145/1035575.1035577`. URL: `http://doi.acm.org/10.1145/1035575.1035577`.

[104] Ray Hyman. "Stimulus information as a determinant of reaction time." In: *Journal of experimental psychology* 45.3 (1953), p. 188.

[105] Takeo Igarashi and Ken Hinckley. "Speed-dependent Automatic Zooming for Browsing Large Documents." In: *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*. UIST '00. San Diego, California, USA: ACM, 2000, pp. 139–148. ISBN: 1-58113-212-3. DOI: `10.1145/354401.354435`. URL: `http://doi.acm.org/10.1145/354401.354435`.

[106] Edward W. Ishak and Steven K. Feiner. "Content-aware Scrolling." In: *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*. UIST '06. Montreux, Switzerland: ACM, 2006, pp. 155–158. ISBN: 1-59593-313-1. DOI: `10.1145/1166253.1166277`. URL: `http://doi.acm.org/10.1145/1166253.1166277`.

[107] Richard J Jagacinski and John M Flach. *Control theory for humans: Quantitative approaches to modeling performance*. CRC Press, 2003.

[108] Waqas Javed, Sohaib Ghani, and Niklas Elmqvist. "GravNav: Using a Gravity Model for Multi-scale Navigation." In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '12. Capri Island, Italy: ACM, 2012, pp. 217–224. ISBN: 978-1-4503-1287-5. DOI: `10.1145/2254556.2254597`. URL: `http://doi.acm.org/10.1145/2254556.2254597`.

[109] Brian Johnson and Ben Shneiderman. "Tree-Maps: A Space-filling Approach to the Visualization of Hierarchical Information Structures." In: *Proceedings of the 2Nd Conference on Visualization '91*. VIS '91. San Diego, California: IEEE Computer Society Press, 1991, pp. 284–291. ISBN: 0-8186-2245-8. DOI: `10.1109/VISUAL.1991.175815`.

[110] George Johnson. "Claude Shannon, mathematician, dies at 84." In: *New York Times* 27 (2001), B7.

[111] Susanne Jul and George W. Furnas. "Critical Zones in Desert Fog: Aids to Multiscale Navigation." In: *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*. UIST '98. San Francisco, California, USA: ACM, 1998, pp. 97–106. ISBN: 1-58113-034-1. DOI: `10.1145/288392.288578`. URL: `http://doi.acm.org/10.1145/288392.288578`.

[112] Alan C. Kay. "A Personal Computer for Children of All Ages." In: *Proceedings of the ACM Annual Conference - Volume 1*. ACM '72. Boston, Massachusetts, USA: ACM, 1972. DOI: `10.1145/800193.1971922`. URL: `http://doi.acm.org/10.1145/800193.1971922`.

[113] Robert Kerr. "Movement time in an underwater environment." In: *Journal of Motor Behavior* 5.3 (1973), pp. 175–178.

[114] AA Knight and PR Dagnall. "Precision in movements." In: *Ergonomics* 10.3 (1967), pp. 321–330.

[115] Aparna Krishnan and Steve Jones. "TimeSpace: Activity-based Temporal Visualisation of Personal Information Spaces." In: *Personal Ubiquitous Comput.* 9.1 (Jan. 2005), pp. 46–65. ISSN: 1617-4909. DOI: `10.1007/s00779-004-0291-x`. URL: `http://dx.doi.org/10.1007/s00779-004-0291-x`.

[116] Per Ola Kristensson. "Five challenges for intelligent text entry methods." In: *AI Magazine* 30.4 (2009), p. 85.

[117] Per Ola Kristensson and Keith Vertanen. "Asynchronous multimodal text entry using speech and gesture keyboards." In: *Twelfth Annual Conference of the International Speech Communication Association*. 2011.

[118] Per-Ola Kristensson and Shumin Zhai. “Relaxing Stylus Typing Precision by Geometric Pattern Matching.” In: *Proceedings of the 10th International Conference on Intelligent User Interfaces*. IUI ’05. San Diego, California, USA: ACM, 2005, pp. 151–158. ISBN: 1-58113-894-6. DOI: 10.1145/1040830.1040867. URL: http://doi.acm.org/10.1145/1040830.1040867.

[119] Donald Richard John Laming. “Information theory of choice-reaction times.” In: (1968).

[120] Donald Richard John Laming. *Mathematical psychology*. Academic Press, 1973.

[121] Donald Laming. “Statistical information, uncertainty, and Bayes’ theorem: Some applications in experimental psychology.” In: *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*. Springer. 2001, pp. 635–646.

[122] John Lamping, Ramana Rao, and Peter Pirolli. “A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 401–408. ISBN: 0-201-84705-1. DOI: 10.1145/223904.223956. URL: http://dx.doi.org/10.1145/223904.223956.

[123] T. K. Landauer and D. W. Nachbar. “Selection from Alphabetic and Numeric Menu Trees Using a Touch Screen: Breadth, Depth, and Width.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’85. San Francisco, California, USA: ACM, 1985, pp. 73–78. ISBN: 0-89791-149-0. DOI: 10.1145/317456.317470. URL: http://doi.acm.org/10.1145/317456.317470.

[124] Mark W Lansdale. “The psychology of personal information management.” In: *Applied ergonomics* 19.1 (1988), pp. 55–66. DOI: 10.1016/0003-6870(88)90199-8. URL: https://doi.org/10.1016/0003-6870(88)90199-8.

[125] Donghee Lee, Jongmoo Choi, Jong-Hun Kim, Sam H Noh, Sang Lyul Min, Yookun Cho, and Chong Sang Kim. “On the existence of a spectrum of policies that subsumes the least recently used (LRU) and least frequently used (LFU) policies.” In: *ACM SIGMETRICS Performance Evaluation Review*. Vol. 27. 1. ACM. 1999, pp. 134–143. DOI: 10.1145/301453.301487. URL: http://doi.acm.org/10.1145/301453.301487.

[126] J Alfred Leonard. “Tactual choice reactions: I.” In: *Quarterly Journal of Experimental Psychology* 11.2 (1959), pp. 76–83.

[127] Vladimir I Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals." In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710.

[128] J. C. R. Licklider. "Man-Computer Symbiosis." In: *IEEE Ann. Hist. Comput.* 14.1 (Jan. 1992), pp. 24–. ISSN: 1058-6180. URL: http://dl.acm.org/citation.cfm?id=612400.612433.

[129] Joseph CR Licklider. "Man-computer symbiosis." In: *IRE transactions on human factors in electronics* 1 (1960), pp. 4–11.

[130] William Lidwell, Kritina Holden, and Jill Butler. *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Rockport Pub, 2010.

[131] Rensis Likert. "A technique for the measurement of attitudes." In: *Archives of psychology* (1932).

[132] Dennis V Lindley. "On a measure of the information provided by an experiment." In: *The Annals of Mathematical Statistics* (1956), pp. 986–1005.

[133] Wanyu Liu, Gilles Bailly, and Andrew Howes. "Effects of Frequency Distribution on Linear Menu Performance." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, 2017, pp. 1307–1312. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025707. URL: http://doi.acm.org/10.1145/3025453.3025707.

[134] Wanyu Liu, Rafael Lucas D'Oliveira, Michel Beaudouin-Lafon, and Olivier Rioul. "BIGnav: Bayesian Information Gain for Guiding Multiscale Navigation." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, 2017, pp. 5869–5880. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025524. URL: http://doi.acm.org/10.1145/3025453.3025524.

[135] Wanyu Liu, Olivier Rioul, Joanna McGrenere, Wendy E. Mackay, and Michel Beaudouin-Lafon. "BIGFile: Bayesian Information Gain for Fast File Retrieval." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 385:1–385:13. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173959. URL: http://doi.acm.org/10.1145/3173574.3173959.

[136] R Duncan Luce. "Whatever happened to information theory in psychology?" In: *Review of general psychology* 7.2 (2003), p. 183.

[137] David JC MacKay, Chris J Ball, and Mick Donegan. "Efficient communication with one or two buttons." In: *AIP Conference Proceedings*. Vol. 735. 1. AIP. 2004, pp. 207–218.

[138] Scott MacKenzie. “Fitts’ law as a research and design tool in human-computer interaction.” In: *Human-computer interaction* 7.1 (1992), pp. 91–139.

[139] Scott MacKenzie. “KSPC (keystrokes per character) as a characteristic of text entry techniques.” In: *International Conference on Mobile Human-Computer Interaction*. Springer. 2002, pp. 195–210.

[140] Scott MacKenzie and William Buxton. “Extending Fitts’ Law to Two-dimensional Tasks.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’92. Monterey, California, USA: ACM, 1992, pp. 219–226. ISBN: 0-89791-513-5. DOI: 10.1145/142750.142794. URL: http://doi.acm.org/10.1145/142750.142794.

[141] Scott MacKenzie and Poika Isokoski. “Fitts’ Throughput and the Speed-accuracy Tradeoff.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’08. Florence, Italy: ACM, 2008, pp. 1633–1636. ISBN: 978-1-60558-011-1. DOI: 10.1145/1357054.1357308. URL: http://doi.acm.org/10.1145/1357054.1357308.

[142] Scott MacKenzie and Shawn X Zhang. “An empirical investigation of the novice experience with soft keyboards.” In: *Behaviour & Information Technology* 20.6 (2001), pp. 411–418.

[143] Scott MacKenzie, Shawn X Zhang, and R William Soukoreff. “Text entry using soft keyboards.” In: *Behaviour & information technology* 18.4 (1999), pp. 235–244.

[144] Wendy E Mackay. “Responding to cognitive overload: Co-adaptation between users and technology.” In: *Intellectica* 30.1 (2000), pp. 177–193.

[145] Thomas W. Malone. “How Do People Organize Their Desks?: Implications for the Design of Office Information Systems.” In: *ACM Trans. Inf. Syst.* 1.1 (Jan. 1983), pp. 99–112. ISSN: 1046-8188. DOI: 10.1145/357423.357430. URL: http://doi.acm.org/10.1145/357423.357430.

[146] Regan L. Mandryk, M. Stella Atkins, and Kori M. Inkpen. “A Continuous and Objective Evaluation of Emotional Experience with Interactive Play Environments.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’06. Montreal, Quebec, Canada: ACM, 2006, pp. 1027–1036. ISBN: 1-59593-372-7. DOI: 10.1145/1124772.1124926. URL: http://doi.acm.org/10.1145/1124772.1124926.

[147] Lidia Mangu, Eric Brill, and Andreas Stolcke. “Finding consensus in speech recognition: word error minimization and other applications of confusion networks.” In: *Computer Speech & Language* 14.4 (2000), pp. 373–400.

[148] Andrei Andreyevich Markov. "The theory of algorithms." In: *Am. Math. Soc. Transl.* 15 (1960), pp. 1–14.

[149] William J McGill. "Applications of information theory in experimental psychology." In: *Transactions of the New York Academy of Sciences* 19.4 Series II (1957), pp. 343–351.

[150] Sean M. McNee, John Riedl, and Joseph A. Konstan. "Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems." In: *CHI '06 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '06. Montr&#233;al, Qu&#233;bec, Canada: ACM, 2006, pp. 1097–1101. ISBN: 1-59593-298-4. DOI: 10.1145/1125451.1125659. URL: http://doi.acm.org/10.1145/1125451.1125659.

[151] Julius Merkel. *Die zeitlichen verhältnisse der willensthätigkeit (The Temporal Relations of the Actions of Will, or The Timing of Voluntary Action)*. W. Engelmann, 1883.

[152] David E Meyer, JE Keith-Smith, Sylvan Kornblum, Richard A Abrams, and Charles E Wright. "Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action." In: (1990).

[153] George A Miller. "The magical number seven, plus or minus two: Some limits on our capacity for processing information." In: *Psychological review* 63.2 (1956), p. 81.

[154] George A Miller, George A Heise, and William Lichten. "The intelligibility of speech as a function of the context of the test materials." In: *Journal of experimental psychology* 41.5 (1951), p. 329.

[155] George A Miller and Jennifer A Selfridge. "Verbal context and the recall of meaningful material." In: *The American journal of psychology* 63.2 (1950), pp. 176–185.

[156] Denis Mottet, Liesjet Elisabeth Henriette van Dokkum, Jérôme Froger, Abdelkader Gouaïch, and Isabelle Laffont. "Trajectory formation principles are the same after mild or moderate stroke." In: *PloS one* 12.3 (2017), e0173674.

[157] GH Mowbray. "Choice reaction times for skilled responses." In: *Quarterly Journal of Experimental Psychology* 12.4 (1960), pp. 193–202.

[158] GH Mowbray and MV Rhoades. "On the reduction of choice reaction times with practice." In: *Quarterly Journal of Experimental Psychology* 11.1 (1959), pp. 16–23.

[159] Fatih Nayebi, Jean-Marc Desharnais, and Alain Abran. "An expert-based framework for evaluating ios application usability." In: *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*. IEEE. 2013, pp. 147–155.

[160] Ted Nelson. "Hypertext notes." In: *ten brief essays on hypertext forms circulated in manuscript circa* (1966).

[161] Allen Newell. *The psychology of human-computer interaction*. 1983.

[162] Allen Newell and Stuart K Card. "The prospects for psychological science in human-computer interaction." In: *Human-computer interaction* 1.3 (1985), pp. 209–242.

[163] John O'Donovan and Barry Smyth. "Trust in Recommender Systems." In: *Proceedings of the 10th International Conference on Intelligent User Interfaces*. IUI '05. San Diego, California, USA: ACM, 2005, pp. 167–174. ISBN: 1-58113-894-6. DOI: 10.1145/1040830.1040870. URL: http://doi.acm.org/10.1145/1040830.1040870.

[164] Antti Oulasvirta, Teemu Roos, Arttu Modig, and Laura Leppänen. "Information Capacity of Full-body Movements." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: ACM, 2013, pp. 1289–1298. ISBN: 978-1-4503-1899-0. DOI: 10.1145/2470654.2466169. URL: http://doi.acm.org/10.1145/2470654.2466169.

[165] Antti Oulasvirta, Per Ola Kristensson, Xiao Jun Bi, and Andrew Howes. *Computational interaction*. Oxford University Press Oxford, 2018.

[166] Ken Perlin and David Fox. "Pad: An Alternative Approach to the Computer Interface." In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '93. Anaheim, CA: ACM, 1993, pp. 57–64. ISBN: 0-89791-601-8. DOI: 10.1145/166117.166125. URL: http://doi.acm.org/10.1145/166117.166125.

[167] Emmanuel Pietriga. "A toolkit for addressing hci issues in visual language environments." In: *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*. IEEE. 2005, pp. 145–152.

[168] Emmanuel Pietriga, Caroline Appert, and Michel Beaudouin-Lafon. "Pointing and Beyond: An Operationalization and Preliminary Evaluation of Multi-scale Searching." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: ACM, 2007, pp. 1215–

1224. ISBN: 978-1-59593-593-9. DOI: 10.1145/1240624.1240808. URL: http://doi.acm.org/10.1145/1240624.1240808.

[169] Peter Pirolli and Stuart Card. “Information foraging.” In: *Psychological review* 106.4 (1999), p. 643.

[170] Réjean Plamondon and Adel M Alimi. “Speed/accuracy tradeoffs in target-directed movements.” In: *Behavioral and brain sciences* 20.2 (1997), pp. 279–303.

[171] Jens G Pohl. “Human-Computer Partnership in Decision-Support Systems: Some Design Guidelines.” In: *Collaborative Agent Design (CAD) Research Center* (1997), p. 36.

[172] Irwin Pollack. “The information of elementary auditory displays.” In: *The Journal of the Acoustical Society of America* 24.6 (1952), pp. 745–749.

[173] Irwin Pollack. “The information of elementary auditory displays. II.” In: *The Journal of the Acoustical Society of America* 25.4 (1953), pp. 765–769.

[174] Irwin Pollack and JM Pickett. “The intelligibility of excerpts from conversation.” In: *Language and Speech* 6.3 (1963), pp. 165–171.

[175] WT Powers. *Living control systems: Selected papers of William T. Powers. Gravel Switch, Ky: The Control Systems Group*. 1989.

[176] Pamela Ravasio, Sissel Guttormsen Schär, and Helmut Krueger. “In Pursuit of Desktop Evolution: User Problems and Practices with Modern Desktop Systems.” In: *ACM Trans. Comput.-Hum. Interact.* 11.2 (June 2004), pp. 156–180. ISSN: 1073-0516. DOI: 10.1145/1005361.1005363. URL: http://doi.acm.org/10.1145/1005361.1005363.

[177] Katharina Reinecke and Abraham Bernstein. “Improving Performance, Perceived Usability, and Aesthetics with Culturally Adaptive User Interfaces.” In: *ACM Trans. Comput.-Hum. Interact.* 18.2 (July 2011), 8:1–8:29. ISSN: 1073-0516. DOI: 10.1145/1970378.1970382. URL: http://doi.acm.org/10.1145/1970378.1970382.

[178] Paul Resnick and Hal R Varian. “Recommender systems.” In: *Communications of the ACM* 40.3 (1997), pp. 56–58.

[179] Meredith Ringel, Edward Cutrell, Susan Dumais, Eric Horvitz, et al. “Milestones in time: The value of landmarks in retrieving information from personal stores.” In: *Proc. Interact*. Vol. 2003. 2003, pp. 184–191.

[180] Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction Design, 4th edition*. Wiley, 2015.

[181] Quentin Roy, Yves Guiard, Gilles Bailly, Éric Lecolinet, and Olivier Rioul. “Glass+ skin: an empirical evaluation of the added value of finger identification to basic single-touch interaction on touch screens.” In: *Human-Computer Interaction*. Springer. 2015, pp. 55–71.

[182] Richard A Schmidt, Howard Zelaznik, Brian Hawkins, James S Frank, and John T Quinn Jr. “Motor-output variability: a theory for the accuracy of rapid motor acts.” In: *Psychological review* 86.5 (1979), p. 415.

[183] SCJMB Schneiderman. *Information Visualization: Using Vision to Think*. 1999.

[184] Andrew Sears and Ben Shneiderman. “Split Menus: Effectively Using Selection Frequency to Organize Menus.” In: *ACM Trans. Comput.-Hum. Interact.* 1.1 (Mar. 1994), pp. 27–51. ISSN: 1073-0516. DOI: `10.1145/174630.174632`. URL: `http://doi.acm.org/10.1145/174630.174632`.

[185] Andrew Sears, Julie A Jacko, Josey Chu, and Francisco Moro. “The role of visual search in the design of effective soft keyboards.” In: *Behaviour & Information Technology* 20.3 (2001), pp. 159–166.

[186] Robert Seibel. “Discrimination reaction time for a 1,023-alternative task.” In: *Journal of Experimental Psychology* 66.3 (1963), p. 215.

[187] Steven C Seow. “Information theoretic models of HCI: a comparison of the Hick-Hyman law and Fitts’ law.” In: *Human-Computer Interaction* 20.3 (2005), pp. 315–352.

[188] Claude E Shannon. “A mathematical theory of communication.” In: 27 (1948), pp. 379–423, 623–656.

[189] Claude E Shannon. “Prediction and entropy of printed English.” In: *Bell Labs Technical Journal* 30.1 (1951), pp. 50–64.

[190] Claude E Shannon. “The bandwagon.” In: *IRE Transactions on Information Theory* 2.1 (1956), p. 3.

[191] Michael Sherman, Gradeigh Clark, Yulong Yang, Shridatt Sugrim, Arttu Modig, Janne Lindqvist, Antti Oulasvirta, and Teemu Roos. “User-generated Free-form Gestures for Authentication: Security and Memorability.” In: *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys ’14. Bretton Woods, New Hampshire, USA: ACM, 2014, pp. 176–189. ISBN: 978-1-4503-2793-0. DOI: `10.1145/2594368.2594375`. URL: `http://doi.acm.org/10.1145/2594368.2594375`.

[192] Ben Shneiderman. "Designing the User Interface Strategies for Effective Human-computer Interaction." In: *SIGBIO Newsl.* 9.1 (Mar. 1987), pp. 6–. ISSN: 0163-5697. URL: http://dl.acm.org/citation.cfm?id=25065.950626.

[193] Ben Shneiderman. "Tree Visualization with Tree-maps: 2-d Space-filling Approach." In: *ACM Trans. Graph.* 11.1 (Jan. 1992), pp. 92–99. ISSN: 0730-0301. DOI: 10.1145/102377.115768. URL: http://doi.acm.org/10.1145/102377.115768.

[194] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, and Nicholas Diakopoulos. *Designing the User Interface, 6th Edition*. Pearson, 2017.

[195] R. William Soukoreff and Scott MacKenzie. "Measuring Errors in Text Entry Tasks: An Application of the Levenshtein String Distance Statistic." In: *CHI '01 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '01. Seattle, Washington: ACM, 2001, pp. 319–320. ISBN: 1-58113-340-5. DOI: 10.1145/634067.634256. URL: http://doi.acm.org/10.1145/634067.634256.

[196] R. William Soukoreff and Scott MacKenzie. "Recent Developments in Text-entry Error Rate Measurement." In: *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '04. Vienna, Austria: ACM, 2004, pp. 1425–1428. ISBN: 1-58113-703-6. DOI: 10.1145/985921.986081. URL: http://doi.acm.org/10.1145/985921.986081.

[197] R William Soukoreff and Scott MacKenzie. "Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI." In: *International journal of human-computer studies* 61.6 (2004), pp. 751–789.

[198] R William Soukoreff and Scott MacKenzie. "An informatic rationale for the speed-accuracy trade-off." In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. IEEE. 2009, pp. 2890–2896.

[199] I Standard. "Ergonomic requirements for office work with visual display terminals (vdts)–part 9: Requirements for non-keyboard input devices. ISO Standard 9241-9: 2000." In: *International Organization for Standardization* (2000).

[200] George J Suci, Melvin D Davidoff, and Walter W Surwillo. "Reaction time as a function of stimulus information and age." In: *Journal of Experimental Psychology* 60.4 (1960), p. 242.

[201] Susanne Tak. "Understanding and Supporting Window Switching." In: (2011).

[202] Sandeep Tata et al. "Quick Access: Building a Smart Experience for Google Drive." In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: ACM, 2017, pp. 1643–1651. ISBN: 978-1-4503-4887-4. DOI: 10.1145/3097983.3098048. URL: http://doi.acm.org/10.1145/3097983.3098048.

[203] Linda Tauscher and Saul Greenberg. "How people revisit web pages: empirical findings and implications for the design of history systems." In: *International Journal of Human-Computer Studies* 47.1 (1997), pp. 97–137.

[204] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. "The Perfect Search Engine is Not Enough: A Study of Orienteering Behavior in Directed Search." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: ACM, 2004, pp. 415–422. ISBN: 1-58113-702-8. DOI: 10.1145/985692.985745. URL: http://doi.acm.org/10.1145/985692.985745.

[205] Loren G Terveen. "Overview of human-computer collaboration." In: *Knowledge-Based Systems* 8.2 (1995), pp. 67–81.

[206] Anne Treisman. "Perceptual grouping and attention in visual search for features and for objects." In: *Journal of experimental psychology: human perception and performance* 8.2 (1982), p. 194.

[207] Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke, and Hans Gellersen. "Motion Correlation: Selecting Objects by Matching Their Movement." In: *ACM Trans. Comput.-Hum. Interact.* 24.3 (Apr. 2017), 22:1–22:35. ISSN: 1073-0516. DOI: 10.1145/3064937. URL: http://doi.acm.org/10.1145/3064937.

[208] Harris Wang. "A guide to assistive technology for teachers in special education." In: *Assistive technologies: Concepts, methodologies, tools, and applications*. IGI Global, 2014, pp. 12–25.

[209] David J. Ward, Alan F. Blackwell, and David J. C. MacKay. "Dasher&Mdash;a Data Entry Interface Using Continuous Gestures and Language Models." In: *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*. UIST '00. San Diego, California, USA: ACM, 2000, pp. 129–137. ISBN: 1-58113-212-3. DOI: 10.1145/354401.354427. URL: http://doi.acm.org/10.1145/354401.354427.

[210] David J Ward and David JC MacKay. "Artificial intelligence: fast hands-free writing by gaze direction." In: *Nature* 418.6900 (2002), p. 838.

[211] Colin Ware and Marlon Lewis. "The DragMag Image Magnifier." In: *Conference Companion on Human Factors in Computing Systems*. CHI '95. Denver, Colorado, USA: ACM, 1995, pp. 407–408. ISBN: 0-89791-755-3. DOI: 10.1145/223355.223749. URL: http://doi.acm.org/10.1145/223355.223749.

[212] Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. "Uncertain Text Entry on Mobile Devices." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 2307–2316. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557412. URL: http://doi.acm.org/10.1145/2556288.2557412.

[213] Alan Traviss Welford. "The measurement of sensory-motor performance: survey and reappraisal of twelve years' progress." In: *Ergonomics* 3.3 (1960), pp. 189–230.

[214] Alan Traviss Welford. "Fundamentals of skill." In: (1968).

[215] Steve Whittaker and Candace Sidner. "Email Overload: Exploring Personal Information Management of Email." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '96. Vancouver, British Columbia, Canada: ACM, 1996, pp. 276–283. ISBN: 0-89791-777-4. DOI: 10.1145/238386.238530. URL: http://doi.acm.org/10.1145/238386.238530.

[216] R William Soukoreff and Mackenzie Scott. "Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard." In: *Behaviour & Information Technology* 14.6 (1995), pp. 370–379.

[217] John Williamson and Roderick Murray-Smith. "Pointing Without a Pointer." In: *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '04. Vienna, Austria: ACM, 2004, pp. 1407–1410. ISBN: 1-58113-703-6. DOI: 10.1145/985921.986076. URL: http://doi.acm.org/10.1145/985921.986076.

[218] Jacob O. Wobbrock and Brad A. Myers. "From Letters to Words: Efficient Stroke-based Word Completion for Trackball Text Entry." In: *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*. Assets '06. Portland, Oregon, USA: ACM, 2006, pp. 2–9. ISBN: 1-59593-290-9. DOI: 10.1145/1168987.1168990. URL: http://doi.acm.org/10.1145/1168987.1168990.

[219] Jacob O. Wobbrock, Edward Cutrell, Susumu Harada, and Scott MacKenzie. "An Error Model for Pointing Based on Fitts' Law." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, 2008, pp. 1613–1622. ISBN: 978-1-60558-011-1. DOI: 10.1145/1357054.

1357306. URL: `http://doi.acm.org/10.1145/1357054.1357306`.

[220] Jinglong Wu, Jiajia Yang, and Taichi Honda. “Fitts’ law holds for pointing movements under conditions of restricted visual feedback.” In: *Human movement science* 29.6 (2010), pp. 882–892.

[221] Stephen John Young, NH Russell, and JHS Thornton. *Token passing: a simple conceptual model for connected speech recognition systems*. Cambridge University Engineering Department Cambridge, UK, 1989.

[222] Shumin Zhai. “Characterizing computer input with Fitts’ law parameters - the information and non-information aspects of pointing.” In: *International Journal of Human-Computer Studies* 61.6 (2004), pp. 791–809.

[223] GK Zipf. *Human Behaviour and the Principle of Least-Effort, patterns*. 1949.