

INTERACTIVE E-CUBE GAMES FOR COGNITIVE ASSESSMENT

A Dissertation

by

XIANGYI CHENG

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee, Kiju Lee
Committee Members, M. Cynthia Hipwell
Ranjana K. Mehta
Vinayak R. Krishnamurthy
Head of Department, Bryan P. Rasmussen

March 2022

Major Subject: Mechanical Engineering

Copyright 2022 Xiangyi Cheng

ABSTRACT

This dissertation presents a low-cost adaptive method for cognitive assessment using the e-Cube system. It features six games: Assembly (e-Cube^A), Shape-Matching (e-Cube^S), Sequence-Memory (e-Cube^{M1}), Spatial-Memory (e-Cube^{M2}), Path-Tracking (e-Cube^P), and Maze (e-Cube^Z), designed to measure cognitive skills autonomously. Using the embedded play complexity measures and dynamic item generators, these games are adaptive to an individual's performance. The evaluation focused on testing the play complexity measures, cognitive assessment, and reliability and usability of the system. The six e-Cube games and three subtests (Block Design, Digit Span, and Matrix Reasoning) of the Wechsler Adult Intelligence Scale - Fourth Edition were administered to 78 participants, of which 41 were assigned to the adaptive e-Cube games and the rest to the fixed games. The computed play complexities were negatively and positively correlated with the mean scores and mean completion time of each item, respectively, providing preliminary evidence for validating the play complexity measures. The preliminary validity of the e-Cube system as a cognitive assessment was confirmed by the significant correlations found between the WAIS-IV subtests and the e-Cube games. The false detection rate was only 0.1% and the average System Usability Scale score was 86.1, indicating e-Cube is reliable and acceptable. To further evaluate the user experience of e-Cube, a genetic algorithm - Support Vector Machine - based facial expression recognition algorithm was embedded in the system. While the potential of the algorithm has been demonstrated, more future work is needed to test its performance in e-Cube.

DEDICATION

To my father, my mother, my grandfathers, and my grandmothers.

ACKNOWLEDGMENTS

I would not be where I am today without my family, my advisor Dr. Kiju Lee and other advisory committee members, and my labmates and friends. Thank you to my father, Deying Cheng, and my mother, Shufang Jiang. You are the best parents, guiders, and listeners in the world. You have been always teaching me to be a kind and honest person and guiding me to create a long-term vision for the life. Most importantly, you trust me and respect my thoughts regardless of where I am. I decided to move out from home and lived in dorms since I was 16, then went to Beijing, and afterward came to the United States. You support my decisions all the time and let me know you are always there for me. Thank you to my grandparents. You are the role models for me and the entire family.

For my advisor, Dr. Kiju Lee, I would like to extend my most profound gratitude to you. You are an excellent educator who is willing to devote your valuable time to students. I appreciate the weekly meetings in which you provided tons of suggestions for my research. Your deep insights and creative ideas saved me multiple times from struggling with my project. You also care about my career and personal plan. I will never forget the teaching opportunities and the help you offered during my academic job search. Your optimistic attitude towards both work and life also leads me to a positive mindset, which is my precious life asset. Overall, you let me know how important an advisor is to a person - a wonderful advisor not only mentor students but also significantly impact their lives and help them achieve their dreams. I will try my best to be as responsible and passionate a teacher as you were to me.

I would also like to express my thankfulness to my advisory committee members, Dr. M. Cynthia Hipwel, Dr. Ranjana K. Mehta, and Dr. Vinayak R. Krishnamurthy. You provided lots of informative and constructive feedback to this research. I also deeply appreciate Dr. Grover C. Gilmore and Dr. Alan Lerner at Case Western Reserve University. Your expertise in Psychology or Neurology provided tremendous supports in this multi-disciplinary research.

A special thanks to my labmates in the Adaptive Robotics and Technology (ART) Lab, Chuanqi Zheng, Elisabeth Ford, Ricardo Castillo, Yuan Wei, Kangneoung Lee, Kevin Airis, and Annalisa Tostenson. Because of you, the laboratory is surrounded by warmth and joy. I appreciate that each of you squeezed two hours out of your busy schedule to attend the human subject study in this research. Last but not the least, thank you to all my friends. My life is more relaxing and interesting with you. Especially, thank you to Duanyang (David) Zheng for motivating me when I was frustrated and providing proofreading for this dissertation.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

The research team includes Dr. Kiju Lee of the Departments of Engineering Technology & Industrial Distribution and Mechanical Engineering, Dr. Grover C. Gilmore of the Department of Psychological Sciences at Case Western Reserve University (CWRU), Dr. Alan Lerner of the Department of Neurology at CWRU, Mr. Michael Allan of Technology Transfer Office at CWRU, graduate student Elisabeth S. Ford of the Department of Engineering Technology and Industrial Distribution, and myself. Facial expression recognition in Chapter 6 was a team project with Xiao Liu, a former labmate and currently a doctoral student in Computer Science at Arizona State University. I carried out all other work in this dissertation independently under the supervision of Dr. Lee.

This work was supported by a dissertation committee consisting of Dr. Lee of the Department of Engineering Technology & Industrial Distribution and Mechanical Engineering, Dr. M. Cynthia Hipwell of the Department of Mechanical Engineering, Dr. Vinayak R. Krishnamurthy of the Department of Mechanical Engineering, and Dr. Ranjana K. Mehta of the Department of Industrial & Systems Engineering.

Funding Sources

Graduate study was supported by National Science Foundation - Partnerships for Innovation - Technology Translation (NSF-PFI-TT) under Grant No. 2002721.

NOMENCLATURE

WAIS-IV	Wechsler Adult Intelligence Scale - Fourth Edition
SUS	System Usability Scale
CWRU	Case Western Reserve University
TAMU	Texas A&M University
NSF-PFI-TT	National Science Foundation - Partnerships for Innovation - Technology Translation
MMSE	Mini-Mental State Exam
MoCA	Montreal Cognitive Assessment
MCI	Mild cognitive impairment
AD	Alzheimer's disease
BD	Block Design
DS	Digit Span
MR	Matrix Reasoning
CAT	Computerized adaptive testing
DDA	Dynamic Difficulty Adjustment
TAM	Technology acceptance model
CNN	Convolutional Neural Network
SIG-Blocks	Sensor-Integrated Geometric Blocks
TAG-Games	Tangible Geometric Games
3D	Three-dimensional

2D	Two-dimensional
ROI	Region of interests
HSV	Hue, saturation, value
RDP	Ramer–Douglas–Peucker
DBMS	Database management systems
SQL	Structured Query Language
GUI	Graphical User Interface
GLCM	Gray-level co-occurrence matrix
SD	Standard deviation
FER	Facial expression recognition
GA-SVM	Genetic algorithm - Support Vector Machine
CK+	The Extended Cohn-Kanade Dataset
MUG	Multimedia Understanding Group
JAFFE	Japanese Female Facial Expression
MMI	MMI Facial Expression Database
AUs	Action Units
FACS	Facial Action Coding System
FS	Facial Segment
LC	Landmark curvature
VL	Vectorized landmark
$\tilde{\mathbf{H}}$	Transformation matrix
N_v	The number of vertices
H	Entropy

f	Gray-level co-occurrence matrix of an image
D_t	Total play complexity
D_{compos}	Compositional complexity
D_{config}	Configurational complexity
$w_{i,j}$	Qualitative weight
D_m	Maze pathway complexity
D_s	Solution logarithmic complexity
D_l	Solution length complexity
D	Desired play complexity
d	Play complexity of a generated item
ϵ	Tolerance
O	The number of connected nodes
c	Maze complexity in the Prim's algorithm
e	Maze density in the Prim's algorithm
M	Polynomial interpolation order
e-Cube ^A	Assembly game
e-Cube ^S	Shape-Matching game
e-Cube ^{M1}	Sequence-Memory game
e-Cube ^{M2}	Spatial-Memory game
e-Cube ^P	Path-Tracking game
e-Cube ^Z	Maze game

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGMENTS	iv
CONTRIBUTORS AND FUNDING SOURCES	vi
NOMENCLATURE	vii
TABLE OF CONTENTS	x
LIST OF FIGURES	xiii
LIST OF TABLES.....	xvi
1. INTRODUCTION.....	1
1.1 Background.....	1
1.1.1 Computer-based cognitive assessments	2
1.1.2 Cognitive assessment employing tangible objects	3
1.1.3 Adaptive testing in computer-based cognitive assessments	4
1.1.4 Methods for user acceptance and usability evaluation.....	6
1.2 Previous Works and Limitations.....	8
1.3 Research Objectives.....	9
2. TECHNOLOGY DEVELOPMENT	11
2.1 Hardware	11
2.1.1 Block design and fabrication.....	11
2.1.2 Other hardware components	13
2.1.3 Computing environment	14
2.2 Vision Algorithms.....	14
2.2.1 Camera transformation	14
2.2.2 Identification of top surface images	17
2.2.2.1 Black and white detection.....	19
2.2.2.2 Polygon determination	22
2.2.2.3 Shape classification	22
2.3 Database	26
2.3.1 Database structure	26

2.3.2	Data collection using MySQL	28
3.	GAME DESIGN	29
3.1	Design of Six e-Cube Games	29
3.1.1	Assembly: e-Cube ^A	29
3.1.2	Shape-Matching: e-Cube ^S	30
3.1.3	Sequence-Memory: e-Cube ^{M1}	30
3.1.4	Spatial-Memory: e-Cube ^{M2}	31
3.1.5	Path-Tracking: e-Cube ^P	33
3.1.6	Maze: e-Cube ^Z	34
3.2	Graphical User Interface	35
4.	DEVELOPMENT OF ADAPTIVE GAMES	43
4.1	Computational Measures of Play Complexity	43
4.1.1	Play complexity measure for e-Cube ^A	45
4.1.2	Play complexity measure for e-Cube ^S	49
4.1.3	Play complexity measure for e-Cube ^{M1} and e-Cube ^{M2}	52
4.1.4	Play complexity measure for e-Cube ^P	52
4.1.5	Play complexity measure for e-Cube ^Z	56
4.2	Adaptive Game Generators	58
4.2.1	Game procedure	59
4.2.2	Item generators for e-Cube ^A , e-Cube ^{M1} , and e-Cube ^{M2}	60
4.2.3	Item generator for e-Cube ^S	60
4.2.4	Item generator for e-Cube ^P	64
4.2.5	Item generator for e-Cube ^Z	65
5.	HUMAN SUBJECT EVALUATION	68
5.1	Methods	68
5.2	Protocol	71
5.3	Scoring System	72
5.4	Results	74
5.4.1	Hypothesis 1. Significant correlations between the performance indicators and the play complexity values	75
5.4.2	Hypothesis 2. Preliminary validity of e-Cube games for cognitive assessment	75
5.4.2.1	Mean differences in the two groups – Adaptive and Fixed	75
5.4.2.2	Comparisons between the fixed and adaptive games	76
5.4.2.3	Preliminary evaluations of age and gender differences	78
5.4.2.4	Preliminary behavior analyses	81
5.4.3	Hypothesis 3. Reliability of autonomous data collection and SUS results	83
6.	EMBEDDED FACIAL EXPRESSION RECOGNITION	84
6.1	GA-SVM-based FER Algorithm	84
6.1.1	Dataset preparation	85

6.1.2	Geometric feature extraction	86
6.1.3	Feature type & interpolating order determination	89
6.1.4	GA-based parameter selection and evaluation on prepared datasets	91
6.2	Performance Comparison with CNN	92
6.3	Performance Comparison with Other Existing Methods	95
6.4	FER Application in e-Cube	98
7.	CONCLUSION AND DISCUSSION	99
	REFERENCES	106
	APPENDIX A. COLOR GAMES	123
	APPENDIX B. DATA CONSTRUCTION AND MANAGEMENT	125
B.1	Database Construction	125
B.2	Writing Commands	126
B.3	Data Query	128
	APPENDIX C. DATA ANALYSIS USING WAIS-IV RAW SCORES.....	129

LIST OF FIGURES

FIGURE	Page
2.1 Hardware of the e-Cube games.	11
2.2 (a-f) Geometric images on six surfaces of a cube and (g) the fourteen shapes obtained by rotating the six geometric images by 0, 90, 180, and 270° and their labels.	12
2.3 (a) A cube assembled by the 3D printed white base and seven black geometric parts and (b) nine fully assembled cubes.	13
2.4 Placemat with a brown rectangular region in the center.	14
2.5 Perspective transform in e-Cube: (a) an original camera view; (b) corner detection applied on the original camera view; and (c) the processed camera view with corner labels.	16
2.6 Direction 1, 2, and 3 for camera placements.	17
2.7 (a-c) Square cell outlines displayed in the e-Cube system indicating (d-f) the number of cubes being used in the test item and where the cubes must be located.	18
2.8 Flow chart of the identification of top surface images.	19
2.9 (a) Original video frame; (b) detected white portions; (c) detected black portions; and (d) segmented cubes by superimposing the white and black portions.	22
2.10 Vertical distance of (a) the vertically placed and (b) horizontally placed strip.	24
2.11 Isosceles and right triangle geometries with different orientations.	24
2.12 Shape detection using binary and Otsu thresholding.	25
2.13 Scheme of the e-Cube database structure.	27
3.1 Sample items of e-Cube ^A (adapted from [1]).	30
3.2 (a) Sample items of e-Cube ^S and (b) their answers (adapted from [1]).	31
3.3 Sample items of e-Cube ^{M1}	32
3.4 Sample items of e-Cube ^{M2}	33
3.5 Sample items of e-Cube ^P	34

3.6	(a) Sample items of e-Cube ^Z : entrances in blue and goal points in red and (b) their shortest paths solved by A^* algorithm.	35
3.7	Access page of e-Cube.	37
3.8	Instruction page of e-Cube.	38
3.9	Game page of e-Cube ^A	38
3.10	Game page of e-Cube ^S	39
3.11	Game page of e-Cube ^{M1} - item displaying.	39
3.12	Game page of e-Cube ^{M1} - item recalling.	40
3.13	Game page of e-Cube ^{M2} - item displaying.	40
3.14	Game page of e-Cube ^{M2} - item recalling.	41
3.15	Game page of e-Cube ^P	41
3.16	Game page of e-Cube ^Z	42
4.1	(a) Possible directions for computing a GLCM and (b) an item with a small D_{config} value.	45
4.2	Items with identical compositional complexity.	46
4.3	Assigned indices and IDs to the shapes for computing GLCM.	47
4.4	GLCM computation of an e-Cube ^A item.	47
4.5	Weight examples following the selection criteria shown in Table 4.1.	48
4.6	Play complexity of the e-Cube ^A items shown in Fig. 4.2.	49
4.7	Play complexity of the e-Cube ^A item 1-10 shown in Fig. 3.1.	50
4.8	Play complexity of the e-Cube ^A item 11-20 shown in Fig. 3.1.	51
4.9	Play complexity of the e-Cube ^S items shown in Fig. 3.2.	53
4.10	Play complexity of the e-Cube ^{M1} items shown in Fig. 3.3.	54
4.11	Play complexity of the e-Cube ^{M2} items shown in Fig. 3.4.	55
4.12	Play complexity of the e-Cube ^P items shown in Fig. 3.5.	56

4.13	Maze pathway conversion: (a) original maze; (b) pathway; and (c) pathway represented in 13×17 dots form, similar to e-Cube ^P items.	57
4.14	Play complexity of the e-Cube ^Z items shown in Fig. 3.6a.	58
4.15	Flowchart of the (a) measures of play complexity and (b) item generators.	59
4.16	Flowchart of the item generator for e-Cube ^A	61
4.17	Flowchart of the item generator for e-Cube ^{M1}	62
4.18	Position order of a (a) four-cube or (b) nine-cube e-Cube ^S item.	63
4.19	Examples of Design 6, 7, 8, and 10 of e-Cube ^S	64
4.20	Flowchart of the item generator for e-Cube ^P	66
4.21	Flowchart of the item generator for e-Cube ^Z	67
6.1	Detected landmarks (yellow) and AU groups (multicolors): (a) a sample image from CK+ and (b) one from MUG (reprinted from [2]).	86
6.2	Feature extraction from given landmarks associated with FS ₁ and FS ₂ : LC feature (κ_{21}) at L_{21} , and VL feature (d_{23}, θ_{23}) at L_{23} (reprinted from [2]).	88
6.3	(a) CK+ examples of seven emotions (i.e., anger, disgust, fear, happy, neutral, sadness, and surprise) (top); landmarks detected (middle); and cropped and resized (bottom) and (b) MUG examples organized in the same way as (a) (reprinted from [2]).	89
6.4	Landmarks selected via the GA-based parameter optimization process: (a) CK+ with 8-class; (b) CK+ with 7-class; and (c) MUG with 7-class (reprinted from [2]). .	91
6.5	Proposed CNN architecture for FER: 4 convolutional layers followed by 1 fully connected layer (reprinted from [2]).	94
6.6	Test accuracy comparison between GA-SVM and CNN on 8-class CK+, 7-class CK+ and 7-class MUG (reprinted from [2]).	96
A.1	Color cubes: (a) six colors and (b) the fabricated cubes.	123

LIST OF TABLES

TABLE	Page
2.1 Selection criteria of the results from binary and Otsu thresholding.....	25
3.1 Six e-Cube games and their expected associated cognitive skills.....	29
4.1 Weight selection for computing D_{config}	48
5.1 Six e-Cube games and their expected associations with WAIS subtests.....	69
5.2 Gender distribution, age mean, and age SD of Fixed Group, Adaptive Group, and all participants.	72
5.3 Scoring system for fixed e-Cube ^A	73
5.4 Scoring system for fixed e-Cube ^S , e-Cube ^{M1} , and e-Cube ^{M2}	73
5.5 Scoring system for fixed e-Cube ^P	73
5.6 Scoring system for fixed e-Cube ^Z	73
5.7 Score statistics from the WAIS-IV subtests and e-Cube games.	74
5.8 Correlations between the play complexity values and 1) the mean score and 2) the mean completion time obtained from individual items from the Fixed Group participants.....	76
5.9 Correlations with degrees of freedom 37 between the scores of fixed e-Cube and scaled scores of the WAIS-IV subtests, shown as $r(p)$	77
5.10 Correlations with degrees of freedom 41 between the scores of adaptive e-Cube and scaled scores of the WAIS-IV subtests, shown as $r(p)$	77
5.11 Inter-correlations with degrees of freedom 37 within the scores of fixed e-Cube games, shown as $r(p)$	79
5.12 Inter-correlations with degrees of freedom 41 within the scores of adaptive e-Cube games, shown as $r(p)$	79
5.13 Correlations between the age and WAIS raw scores/e-Cube scores, shown as $r(p)$...	80
5.14 Score statistics of male and female participants in Fixed Group (male: $n = 22$ and female: $n = 15$).	80

5.15	Score statistics of male and female participants in Adaptive Group (male: $n = 24$ and female: $n = 17$).....	81
5.16	Correlations with degrees of freedom 37 between the behavior data and the 1) e-Cube scores; 2) raw scores of the WAIS-IV subtests; 3) scaled scores of the WAIS-IV subtests; or 4) ages from the Fixed Group participants.	82
5.17	Mean and SD of SUS scores in Fixed Group, Adaptive Group, and both groups.	83
6.1	AUs, FSs with description, and associated landmark numbers for each segment (reprinted from [2])......	87
6.2	VL, LC features, and polynomial interpolation order (M) determination (reprinted from [2]).	90
6.3	SVM parameters and their corresponding validation and test accuracy results (reprinted from [2]).	92
6.4	Statistics of the GA-SVM based FER: precision, recall, and F1-score (reprinted from [2]).	93
6.5	Statistics of CNN based FER: precision, recall, and F1-score (reprinted from [2]).	95
6.6	Comparison of recent FER approaches based on frontal face data (reprinted from [2]).	97
A.1	Lower and upper bounds of the six colors in HSV color space.	124
C.1	Correlations with degrees of freedom 37 between the scores of fixed e-Cube and raw scores of the WAIS-IV subtests, shown as $r(p)$	130
C.2	Correlations with degrees of freedom 41 between the scores of adaptive e-Cube and raw scores of the WAIS-IV subtests, shown as $r(p)$	130

1. INTRODUCTION

This dissertation presents the development and evaluation of a new autonomous vision-based system, called *e-Cube*, for assessing multiple cognitive skills, including fine-motor abilities, visual-motor integration, perceptual reasoning, and working memory in an adaptive manner. The e-Cube technology adopts a set of blocks, a camera with a stand, and a host computing and display device as hardware and fast and reliable algorithms for adaptive game generation and autonomous administration. Six e-Cube games have been designed based on the technology, which are expected to measure different cognitive domains. Preliminary validity, reliability, and usability of the six games and the technology were evaluated through a human subject study. This chapter first reviews related literature and previous work and then states the research objectives.

1.1 Background

Cognitive abilities affect a person's learning outcomes, physical and mental health, and social behavior [3, 4, 5]. Cognitive assessments aim to measure multiple domains of cognition, including visuospatial abilities, working memory, language, attention, executive function, fine-motor skills, and orientation [6, 7]. Identifying impairment in any of these domains, diagnosing the cause, specifying the severity, and tracking the progression of the symptom are the common purposes of cognitive assessments in clinical settings [8].

There exist standardized instruments widely used for assessing cognitive skills. The Wechsler Adult Intelligence Scale (WAIS), often referred to as a "gold" standard, offers a comprehensive assessment of cognitive abilities and intelligence using 15 subtests targeting various cognitive domains [9, 10]. The fourth edition (WAIS-IV) is normed on the ages of 16-90. WAIS is adopted in clinical, research, and educational settings with its proven effectiveness, user-friendliness, and developmental appropriateness [10]. It takes about 60-90 minutes to administer by a qualified psychologist or psychiatrist. The administration and scoring processes of WAIS are considered labor-intensive and costly [11]. The Stanford-Binet Intelligence Scale – Fifth Edition (SB5) is

another standardized instrument that consists of 5 verbal and 5 nonverbal subtests. SB5 can be administered for people at the ages of 2 and older, and thus has been widely used in early childhood assessment as well as career assessment and cognitive ability evaluation [12]. Several WAIS and SB5 subtests heavily rely on a person's verbal skills and thus show some limitations when administered to examinees using a non-native language version [13, 14]. Raven's Progressive Matrices (RPM) is a nonverbal test that consists of 60 items, each of which is a visual geometric design with a missing piece. RPM measures the cognitive functioning of individuals from 5-year-olds to the elderly regardless of the differences in culture, language, and reading/writing skills [15, 16]. Another existing test, called the Wonderlic Personnel Test, estimates one's intelligence via 50 multiple choice questions answered within 12 minutes. The questions focus on mathematics, vocabulary, logic, and reasoning [17]. Certain companies and associations have employed this test to select their employees or to predict their performance [18].

While the instruments reviewed above aim for comprehensive cognitive assessments, several brief assessment tools also exist for rapid cognitive screening. The Mini-Mental State Exam (MMSE) [19] and Montreal Cognitive Assessment (MoCA) [20] are two of the commonly accepted instruments widely used for screening mild cognitive impairment (MCI), Alzheimer's disease (AD), and other dementias. Assessments of social anxiety and obsessive-compulsive disorder via questionnaires have also been developed and validated due to the importance of cognitive factors in the treatment of such mental disorders [21, 22].

1.1.1 Computer-based cognitive assessments

The advancement of digital technologies has enabled researchers to explore computer-based methods for cognitive assessments. Computer-based tests can reduce administration burden, compute scores automatically, reduce cheating, and standardize test conditions once the test is successfully validated [23]. One straightforward application is a direct conversion of a paper-and-pencil test into a computerized version while retaining the contents and formats. Q-interactive is a digital system initially developed for use with WAIS-IV. It displays test items, stores data, and computes scores under the operation of the administrator using two iPads – one used by the administrator

and the other by the test taker [24]. However, this digital version of WAIS-IV still requires a professional to be the administrator and manually check the correctness in some subtests, such as Block Design (BD). BD involves a set of physical cubic blocks for examinees to manipulate. The examinee is asked to assemble the blocks to match their top surfaces with a given image displayed on the booklet. Manually checking the assembled blocks remains the same as in administering the original version of WAIS without reducing too much labor.

An increasing body of research focuses on the innovative use of computers and other technologies for cognitive assessment. Interactive and engaging features in technology-enabled methods can make assessments more acceptable and attractive to young children [25]. For example, Test de Evaluación Neuropsicológica Infantil (TENI) is a game-based instrument consisting of 10 subtests developed for measuring the cognitive abilities of Latin American children ages 3 to 9. It provides an enjoyable and non-stressful experience in both education and clinical settings [26]. Several computerized methods have also been developed for the older population. No-Go Discriminant Task is a tablet game similar to the classic Go game [27]. While a strong correlation was found between standard cognitive assessment scores and performance in this game, several usability and ergonomic issues were also identified, such as unintentional screen touching, the lack of challenge due to the game being too easy, and occurrence of required assistance in holding the tablet [28]. A digital whack-a-mole game with nine holes filled with moles – which pop up randomly – was introduced to fill the gap existing in the accurate evaluation of the transitory condition between normal stage and cognition failure [29]. This game is intended for screening cognitive impairment. The overall scoring depends on whacking each mole as it appears and the reaction time.

1.1.2 Cognitive assessment employing tangible objects

Fine-motor and visual-motor integration are important cognitive skills that are often difficult to measure objectively. These skills are closely linked to many neurological diseases and brain injuries [30]. Existing research also suggests that deterioration of fine-motor control and coordination characterizes sensorimotor deficiencies of MCI and AD [26, 27, 28, 31]. To capture the coupled and complex nature of these skills properly, assessments often employ physical objects to

be manipulated by examinees. The BD subtest in WAIS, originated from the Koh Block Design test, was one of the examples [8, 32] directly targeting these cognitive areas. In addition, BD requiring examinees to continuously manipulate blocks to construct answers offers opportunities to study the problem-solving process [33]. This process study can reveal insensible cognitive strategies of individuals. However, the scoring of BD only depends on accuracy and time as they are easy to measure, although how the examinee manipulates the blocks is also a critical factor [33]. Besides using the blocks, the Nine-Hole Peg Test (NHPT) evaluates fine motor skills and finger dexterity by asking the examinee to take the pegs from a container and place them into the holes on the board one by one as quickly as possible [34]. The NHPT is commonly used for patients with various neurological diagnoses.

Despite the importance, assessing cognitive and fine motor skills employing physical objects poses unique challenges. It is labor-intensive and involves possible errors and subjectivity during the assessment. Automating such procedures requires additional technological innovations beyond what is expected for computerized tests. For example, a platform called ETAN employs a tangible user interface and physical objects for evaluating cognition via a well-known neuropsychological test—the Baking Tray Task [35]. Cognitive Cubes are sensor-embedded cubes interfaced with a host computer designed for assessing spatial, constructive ability by asking users to construct 3D shapes with the cubes. A pilot study involving 16 participants demonstrated that this three-dimensional form of assessment was sensitive to differences in individual cognitive abilities [30]. Some tools involving physical objects are designed for specific age groups. For example, a board game consisting of two sensor-augmented tokens and a displaying board has been developed for evaluating children’s fine-motor skills and tested on 30 children aged 7-9 [36]. LogicART is another hardware-software system employing physical materials augmented by wireless communications developed as an assessment and training tool for children [37].

1.1.3 Adaptive testing in computer-based cognitive assessments

Most existing instruments involve a fixed set of items for all examinees. Adaptive testing, on the other hand, involves different items for each individual based on their performance or known

abilities. This is not easy to implement in the traditional paper-based tests, while the adoption of a computer makes it achievable. Computerized adaptive testing (CAT) selects the test items sequentially based on the examinee's previous performance thus can tailor the test to each examinee's latent trait level [38]. Classical CAT is one of the main applications of the item response theory (IRT), which is a probability model estimating the likelihood of different responses to questions/tests items by people with various traits [39]. Therefore, IRT models are useful in evaluating certain traits of an individual through a measurement continuum [40]. CAT in cognitive assessments has illustrated its efficiency in improving the selection of proper items, reducing unnecessary testing time, and reducing cost [41, 42]. Moreover, researchers have combined CAT with cognitive diagnostic models (CDMs) to implement cognitive diagnostic computer adaptive test (CD-CAT) [43]. The essential components of CD-CAT include the item response model, calibrated item bank, entry-level (starting point), item selection rule, scoring method, and termination criteria [43, 44]. This approach selects items according to the item response model and item selection rule from the bank where all the items are stored [44].

The item bank in the CD-CAT must contain a sufficient number of items with enough breadth and depth. However, construction and calibration of a large item bank are expensive and time-consuming [45]. Instead, some computer-based games, especially serious games designed for other primary purposes beyond pure entertainment, use Dynamic Difficulty Adjustment (DDA) to adjust its difficulty by the players' performance [46]. The methodology of DDA can be different depending on the game types. Machine learning techniques, such as support vector machine (SVM), Monte-Carlo tree search, and neural network, were applied to the data collected during the games to evaluate the player's level then adjust the difficulty [47, 48, 49]. Parameter-based approaches are also broadly adopted to achieve DDA because of their high flexibility and efficiency. Some researchers have also established the relationship of player's performance and difficulty parameters [46, 50]. For example, the number of deaths, the number of times hit by an enemy, the game playtime, and the expected completion time were used as parameters to adjust the game difficulty to enable a better user experience in a side-scrolling game [50]. For games involving

a rehabilitation robot, the distance to the target and the speed of the target were obtained by an evolutionary algorithm and adjusted based on users' performance [46].

Since the utility of DDA has been primarily demonstrated for computer games [51], it can be possibly implemented in CAT to adjust the item complexity by tailoring the items to the examinee's skill level without requiring an exhaustive item bank. As for the specific application of this project, there are 38,416 distinctive items for 2×2 items and 20,661,046,784 items for 3×3 items available. Knowing that the 2×2 items require 1MB to store the necessary data, the memory required for storing the 3×3 items is estimated to be more than 500GB. This apparently exceeds the memory limit of most domestic personal computers or tablet devices. DDA eliminates this requirement and therefore is considered as a useful approach to build an adaptive testing environment for cognitive assessment tests.

1.1.4 Methods for user acceptance and usability evaluation

User acceptance and usability are vital components for successful technology adoption [52, 53]. Acceptance evaluation aims for measuring users' subjective opinions of information technology (IT) - the willingness to use it [54, 55]. Among various IT acceptance theories, the technology acceptance model (TAM) proposed by Fred Davis [56] has been widely used [52]. TAM takes perceived usefulness and perceived ease of use as the two main factors to investigate technology development [55]. Researchers have employed TAM in assessing user acceptance of technical applications including but not limited to wireless Internet [57], health information technologies [58], e-learning [59], e-shopping [60], and online banking [61]. TAM was also extended by adding external variables such as facilitating condition [62] and the information behavior model [63] to accommodate different needs. Different than acceptance, usability was defined as "a quality attribute that assesses how easy user interfaces are to use" by the well-known usability expert Jakob Nielsen [64, 65]. Therefore, usability testing determines how well a user can use the product [54]. International Standards Organization 9241-11 defines effectiveness, efficiency, and satisfaction as the three aspects for evaluating usability [66]. There are some conventional usability tests, for instance, Thinking-aloud [67], RITE method [68], heuristic evaluation [69], and Goal Question

Metric (GQM) model [70]. Besides these, some researchers used objective attributes, such as time on task, task success ratio, the number of errors, searching time, and data recorded for the graphical tasks' software to evaluate usability, whereas subjective perceptions from questionnaires have been taken as TAM results [54, 71].

A literature explored the relationship between TAM and usability tests using an eCampus learning system and found that TAM results are not positively correlated with the usability test results [54]. This indicates that TAM and usability tests are somewhat different, at least in certain areas. Jone Brooke proposed a tool, called the System Usability Scale (SUS), to assess both user acceptance and usability of products [52, 55, 72]. SUS is a 10-item questionnaire with demonstrated validity and reliability [52]. It has been strongly recommended as a measure of perceived usability by researchers [73]. SUS has also been used in the usability evaluation of cognitive assessment tools. For example, a novel cognitive assessment based on the social robot "pepper" and IBM cloud AI "Watson" has been evaluated using SUS and found to be reliable and usable [74]. RehabCity, an online game for cognitive assessment and rehabilitation, demonstrated its good effectiveness, efficiency, and satisfaction based on the SUS results collected from the participants [75].

With the rapid technology evolution, usability and user experience become critical for delivering better products and surviving in competitive markets [76]. Therefore, researchers began to develop novel usability evaluation approaches that are expected to be easy, cheap, quick, and accurate [77]. The heuristic evaluation was combined with a gesture-level model (GLM) to test the usability of mobile applications in early design stages, the preliminary results of which demonstrated its potential [77]. To overcome the limitations of retrospective Thinking-aloud, an eye-tracking technique has also been employed as a tool to evaluate usability [76]. Since the retrospective Thinking-aloud approach asks users to recall their thoughts after try-on, there is always a period of silence without any feedback. By combining the examination of eye movement patterns and verbal comments, researchers demonstrated that it is feasible to evaluate usability in or without silence [76]. Facial expression recognition is another advanced method for measuring qualitative user experiences [78]. Since facial expression reflects one's emotional/mental status and stress

level, it can leverage with other quantitative assessments. The recognition usually consists of four steps, image acquisition, pre-processing, feature extraction, and classification (or regression) [79]. There are numerous ways to achieve each of the steps. For example, the feature can be extracted by using Action Units [80], facial texture [81], or gradient features [82]. An algorithm combining brightness and contract filter and edge extraction filter with Convolutional Neural Network (CNN) based learning and Support Vector Machine (SVM) showed 98.19% accuracy for facial expression classification [78]. An extended work based on this facial expression recognition approach further demonstrated its potential in the user experience evaluation [2].

1.2 Previous Works and Limitations

Several standardized instruments for cognitive assessments have been developed, validated, and adopted in clinical practice, research, and education [10, 15]. However, limitations and challenges in the current methods still exist. For example, RPM and the Wonderlic Personnel Test lack the capability of measuring fine motor skills, which is a critical factor in predicting MCI-AD conversion along with the cognitive status [83]. Although WAIS-IV and SB5 provide comprehensive assessments, administrations of these tests are costly, time-consuming, labor-intensive, error-prone, and heavily affected by language and culture [11]. One literature even pointed out that the verbal/non-verbal dichotomy in SB5 is not consistent with its theoretical underpinnings, thus their existences are not clear yet [84].

Preserving the physical manipulation task similar to the WAIS-IV BD subtest and the convenience of technology-enabled administration and data collection, Sensor-Integrated Geometric Blocks (SIG-Blocks) were developed and tested for their potential in measuring cognitive skills [85]. Each SIG-Block, covered with simple black-and-white geometric shapes, can sense physical motions applied to it, detecting adjacent blocks, and sending such data to a local host computer in real-time using embedded sensors and wireless communication. Tangible Geometric Games (TAG-Games) refers to the computerized games that use SIG-Blocks as a means of game control. Three types of TAG-Games, including assembly, shape matching, and memory, were designed for assessing cognitive and fine-motor skills while employing the physical objects, i.e., SIG-Blocks.

A study involving 86 participants illustrated its reliability via split-half and test-retest reliability evaluation and showed that the three games were correlated with the three subtests of WAIS (i.e., BD, Matrix Reasoning (MR), and Digit Span (DS)) [85]. To understand the relationship between the complexity of the items used in the games and the actual performance of the participants, computational complexity measures were defined and correlations between the complexity values and performance metrics (e.g., time and accuracy) were analyzed. The results showed high correlations in all three games [85, 86]. TAG-Games is one of a few systems that can automate the administration and data collection of such tasks involving physical object manipulation. Besides TAG-Games, Music Blocks and the corresponding autonomous games built using SIG-Blocks as platforms introduced aural stimuli into assessment, which successfully demonstrated audio and visual stimuli share the same and highest accuracy in memory [87]. Another tangible memory game, Memorix, allows developers to customize the test items and design dynamically by adopting a set of interactive creation blocks *i*SIG-Blocks and a tabletop platform [88].

Despite the demonstrated potential, several challenges were identified [1, 87]:

- **High cost:** Each block costs about \$150 and the entire system costs about \$1500 without the cost of the host computer.
- **High maintenance:** The system requires frequently recharging and needs a lot of work from technicians to keep it in good condition once the blocks face malfunctions.
- **Technical malfunctions:** Sensors may face difficulties when detecting some of the rotations, errors can occur when the hand grasping the block interfered with reflective optical sensors.

These challenges make this form of technology not ideally suitable for easy and long-term adoption and use.

1.3 Research Objectives

This research is built on the previous work on SIG-Blocks and TAG-Games [85]. For the technology to be broadly used as a cognitive assessment tool, a computer vision-based approach has

been developed and applied to the new system, called e-Cube, using a set of 3D-printed cubes and a webcam instead of the highly instrumented, and thus expensive, SIG-Blocks. This project has been funded by National Science Foundation (NSF) through the Partnerships for Innovation-Technology Translation (PFI-TT) program. Therefore, the research objectives of this dissertation closely align with the program goal of creating a pathway towards commercialization by conducting further research to generate evidence-based knowledge for technology validation and identify potential commercial pathways. The technology conversion from the sensor-instrumented hardware to the vision-based method makes it cheaper, easier to use for a longer time, and more adaptive and suitable for market needs. In particular, the ongoing pandemic due to COVID-19 urged us to create a tool that can be used for self-assessment at individual homes. Considering these market needs and technology readiness, the development of the e-Cube system follows the following design criteria: 1) low cost, 2) ease of use, 3) autonomous administration and data collection, 4) low maintenance, 5) reliable and valid assessment outcomes, and 6) adaptable to individual differences.

Towards the development and evaluation of the e-Cube system, this research project pursues the following three specific aims:

- **Aim 1: Technology development.** Establish the technical groundwork of vision-based methods for evaluating accuracy and behavior by detecting the shapes on the cubes;
- **Aim 2: e-Cube game design.** Design a set of adaptive games based on computational measures of play complexity and adaptive game generators for measuring various cognitive skills.
- **Aim 3: Human subject evaluation.** Evaluate the system and games through human subject studies by testing the following three hypotheses: 1) performance indicators are correlated with the play complexity values of the items in each e-Cube game; 2) the assessment results from the adaptive e-Cube games are correlated with the results from the existing standardized methods; and 3) the e-Cube technology is reliable and usable.

2. TECHNOLOGY DEVELOPMENT

This chapter presents the hardware, vision algorithms, and database of the e-Cube technology. The e-Cube technology is in the form of a low-cost, low-maintenance, easy-to-use instrument for target applications in routine cognitive assessments that can be self-administered at home without requiring a professional.

2.1 Hardware

The hardware of the e-Cube system involves a set of 3D-printed cubes (i.e. geometric blocks), a fabric placemat, a webcam with a custom-designed stand, and a host computing device with a display, as shown in Fig. 2.1.

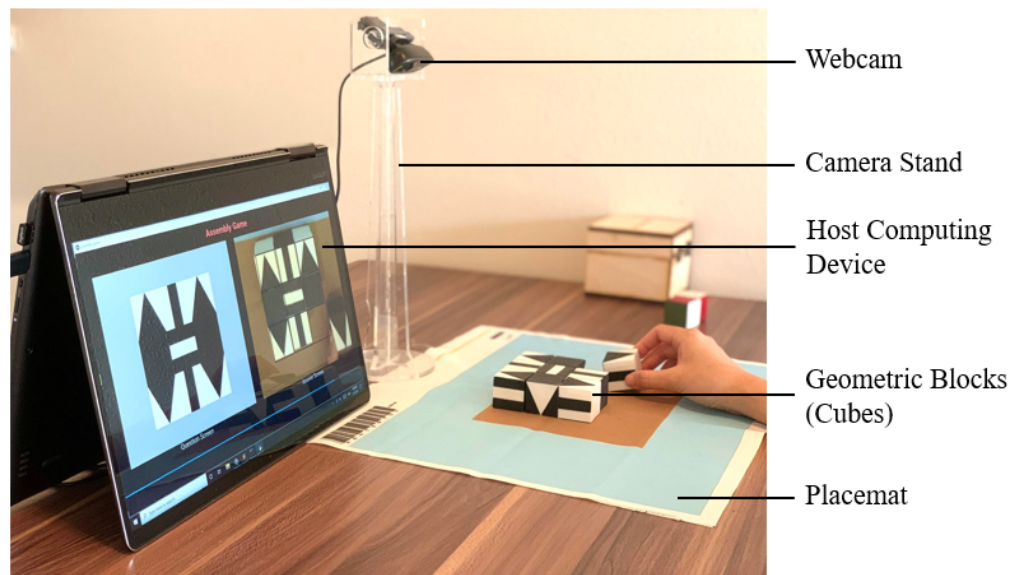


Figure 2.1: Hardware of the e-Cube games.

2.1.1 Block design and fabrication

Each cube is about 1.2-inch long along each side and this size is considered appropriate to be comfortably manipulated by all ages, including children and adults. Six faces of the cube have

the same shapes used in SIG-Blocks (Fig. 2.2) [85]. These shapes contain three distinct pairs of geometric images, including squares (Fig. 2.2a,b), strips (Fig. 2.2c,d), and triangles (Fig. 2.2e,f), representing different rotational symmetries with opposite colors in black and white. A white or black square would remain the same configuration when rotated by 90, 180, or 270° (4-fold rotational symmetry). Each strip shape remains the same configuration when rotated only by 180° (2-fold rotational symmetry), and the triangle has no repeated configuration (1-fold rotational symmetry). For each cube covered with these six images on its surfaces, there exist fourteen distinctive top surface shapes when seen from a fixed viewpoint (Fig. 2.2g).

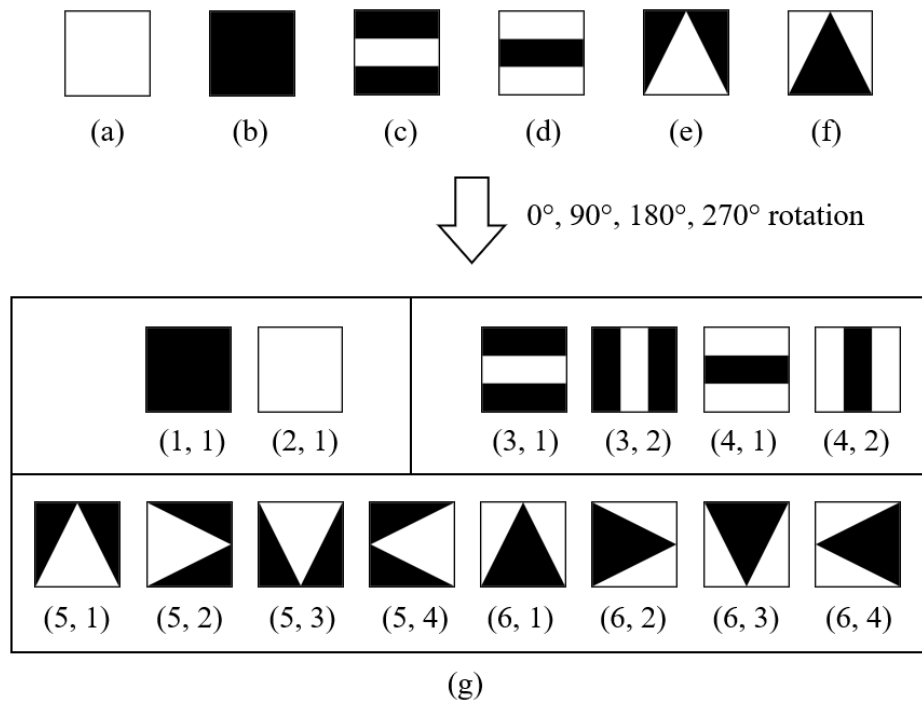


Figure 2.2: (a-f) Geometric images on six surfaces of a cube and (g) the fourteen shapes obtained by rotating the six geometric images by 0, 90, 180, and 270° and their labels.

To fabricate via 3D printing, the cube was decomposed into a white base and seven black geometric parts (Fig. 2.3a). Individual components were 3D printed separately using polylactic acid (PLA) filaments. Fig. 2.3b shows nine fabricated cubes, the black parts of which fit in the corresponding grooves in the white base. Different geometric shapes or solid colors may also

be used. For example, color cubes can replace the geometric cubes for easier test items when necessary (Appendix A).

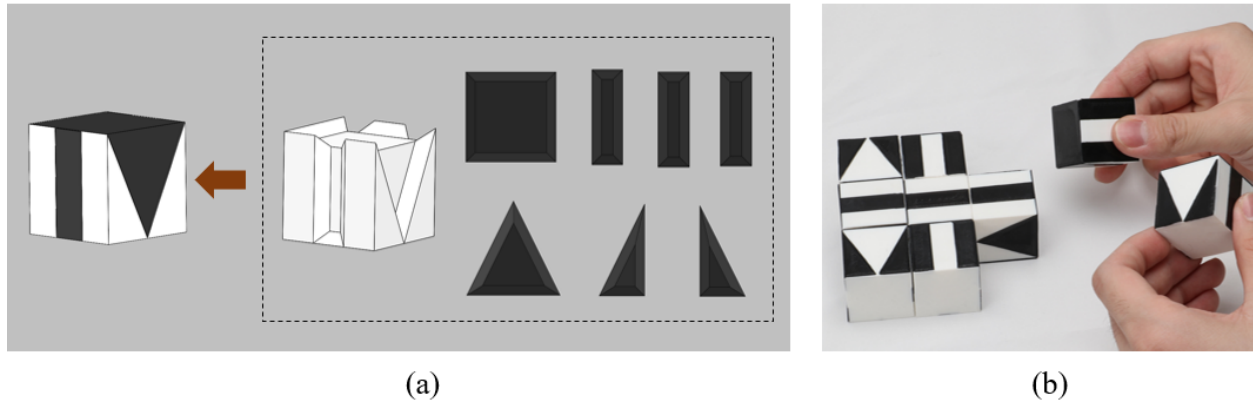
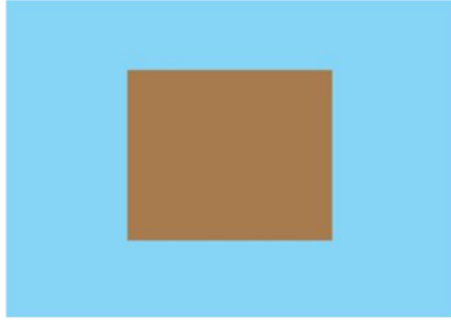


Figure 2.3: (a) A cube assembled by the 3D printed white base and seven black geometric parts and (b) nine fully assembled cubes.

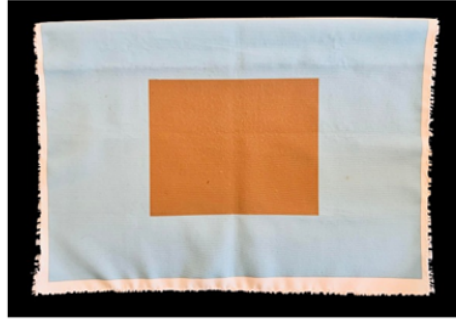
2.1.2 Other hardware components

The placemat is made of 11.75"×16.5" woven canvas with a brown rectangular region in the center (7.5"×6") (Fig. 2.4). When the e-Cube game starts, the camera automatically detects corners of the brown region – the area where all cubes are expected to be placed – and traces the movements of the cubes within this region. Alternatively, the design can be color-printed on paper which will further reduce the cost.

A webcam serves as an "eye" of the e-Cube system capturing real-time activities in this vision-based assessment. An easy-to-use and low-cost device with appropriate specifications, i.e., Logitech C270 with 1280 × 720 pixels resolution, was selected from a variety of available products in the market. This camera has embedded smart light correction and USB-A plug-and-play of simple connection without requiring additional drives and costs less than \$30. A 13-inch high camera stand for this webcam was also designed and built with an adjustable angle joint (Fig. 2.1).



(a) design drawing



(b) manufactured placemat

Figure 2.4: Placemat with a brown rectangular region in the center.

2.1.3 Computing environment

The e-Cube system operates in Lenovo Yoga 710 15.6 inches with Intel® core™ i5-7200U CPU at 2.5 GHz and 8 GB RAM. Lenovo Yoga 2-in-1 laptop with a 360° flipped touch screen has features of both a tablet and a laptop. Combining the computational capacity of a laptop and the convenient user interaction of a tablet, the laptop only cost about \$700 at the time of its purchase in 2019. Alternatively, the system has also been smoothly executed in Dell XPS 15 Laptop with Intel® core™ i7-10750H CPU at 2.6 GHz and 16 GB RAM. In general, computing devices with similar or higher processing capacities would be qualified to fully execute the e-Cube system.

2.2 Vision Algorithms

Vision algorithms in e-Cube aim to identify the top surface images of the cubes, given an arbitrary camera position around the placemat. To achieve this, a universal camera transformation algorithm and vision processing algorithms for geometric image classification are developed and implemented.

2.2.1 Camera transformation

The brown rectangular region of the placement defines the ‘play area’ where all cubes are expected to be placed during the test. The laptop must display and track the real-time view of the play area. To allow flexibility in the camera position, a strategy for the autonomous transformation

of the camera view is needed. Perspective transform, also known as homography [89], is applied to transform the original three-dimensional (3D) object view to a fixed two-dimensional (2D) view without deformation. The camera transformation follows the procedures described below:

1. Detect four corners of the brown area in the original camera frame (Fig. 2.5a) by Harris corner detector [90]. Gradients in larger magnitude along the x -axis and y -axis over small regions may typify edges or corners. However, only the gradients around corners owe significant magnitudes along both axes. Note that the covariance matrix of gradients is

$$M_g = \begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix} = \sum_{p \in P} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \quad (2.1)$$

where $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$. Each element in the M_g matrix is a summation of the products of the derivatives at every pixel p over the entire image P . Hence, two significant eigenvalues appear in the M_g matrix once corners exist. To screen out the irrelevant pixels and only target the "corner candidates", Harris corner detector sets a threshold on value of R , which is defined as

$$R = \det(M_g) - \kappa \text{trace}^2(M_g) \quad (2.2)$$

where $\det(M_g)$ is the determinant of the covariance matrix and $\text{trace}(M_g)$ is the sum of elements on the main diagonal of the covariance matrix. The pixels with large derivatives that fall above the threshold are treated as the "corner candidates."

2. Cluster all the "corner candidates" in four groups using K-means [91] and sort their centroids by their positions and label them as 1 (bottom right), 2 (bottom left), 3 (top right), 4 (top left), respectively (Fig. 2.5). Each centroid is a detected corner and mapped to one of the four pre-defined locations. However, the detected corners via K-means are randomly arranged and this would raise an inaccurate transformation outcome. Therefore, sorting the detected

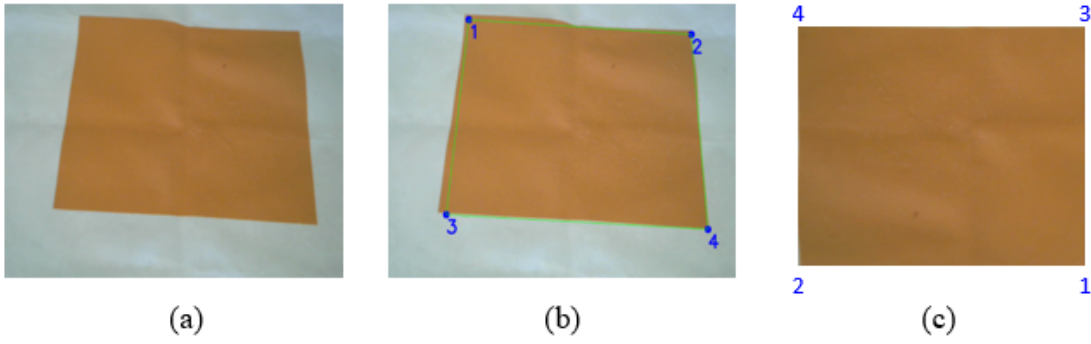


Figure 2.5: Perspective transform in e-Cube: (a) an original camera view; (b) corner detection applied on the original camera view; and (c) the processed camera view with corner labels.

corners are performed based on their locations.

3. Compute the transformation matrix $\tilde{\mathbf{H}}$ based on the four detected corners and the four pre-defined locations (i.e. $(625, 500)$, $(0, 500)$, $(625, 0)$, $(0, 0)$ in e-Cube). Denote the coordinates of a detected corner as

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.3)$$

then the homogeneous vector of \mathbf{x} is

$$\tilde{\mathbf{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.4)$$

The homogeneous vector of the pre-defined coordinate $\tilde{\mathbf{x}}'$ mapped by this point can be expressed as

$$\tilde{\mathbf{x}}' = \begin{bmatrix} \tilde{x}' \\ \tilde{y}' \\ 1 \end{bmatrix} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} \quad (2.5)$$

where $\tilde{\mathbf{H}}$ is a 3×3 matrix that can be obtained by establishing the mapping relationships

between the four detected corners and the four pre-defined locations.

4. Transform the unprocessed camera view to the view that only contains the focus play area (Fig.2.5c) based on $\tilde{\mathbf{H}}$. Suppose the homogeneous vector of a location within the play area is $\tilde{\mathbf{p}}$, its transformed location $\tilde{\mathbf{p}}'$ is computed by

$$\tilde{\mathbf{p}}' = \tilde{\mathbf{H}}\tilde{\mathbf{p}} \quad (2.6)$$

This autonomous transformation offers flexibility in camera placement, allowing users to place the camera at their preferred location around the placemat. Specifically, users can place the camera along direction 1, direction 2, or direction 3 shown in Fig. 2.6 based on their habits and handedness. All the image processing applications mentioned in this dissertation are performed using OpenCV with Python.

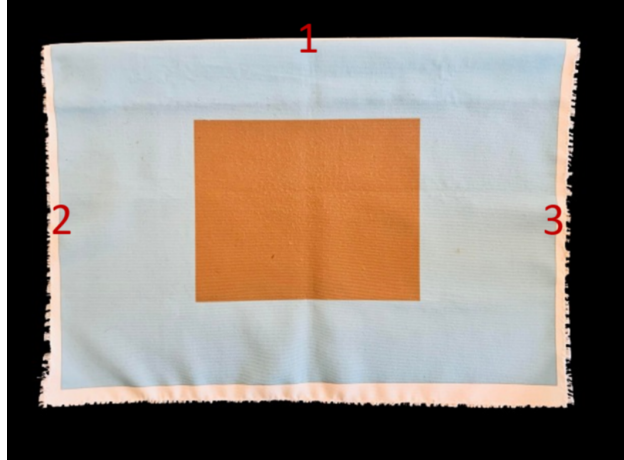


Figure 2.6: Direction 1, 2, and 3 for camera placements.

2.2.2 Identification of top surface images

The e-Cube system requires accurate identifications of top surface images of the cubes placed within the play area to provide accurate assessment outcomes. The detection must be in real-time

to enable adaptive testing based on the examinee’s real-time performance. In addition, considering the potential use at individual homes, the algorithms must run smoothly in low-end computing devices. To do so, it is necessary to develop reliable and accurate algorithms for image processing accommodating the limited processing capabilities. Since the images on the cube surfaces are with 1-, 2-, or 4-fold rotational symmetry, there are 14 distinct shapes to be identified (Fig. 2.2g). Individual square cells are assigned for placing the cubes, wherein a developed geometry and orientation detection algorithm is executed. The number of the assigned cells is determined by the game settings. For example, there may be 1 (Fig. 2.7a), 4 (Fig. 2.7b), or 9 (Fig. 2.7c) cell(s) displayed by their outlines. Fig. 2.8 illustrates the identification process of the top surface images applied to each cell. First, the algorithm detects the black and white regions within each cell to check if there exists a cube through black and white detection. Second, it identifies polygons in the detected white regions if a cube exists. Last, it determines the specific geometry and its orientation using the shape detection algorithm.

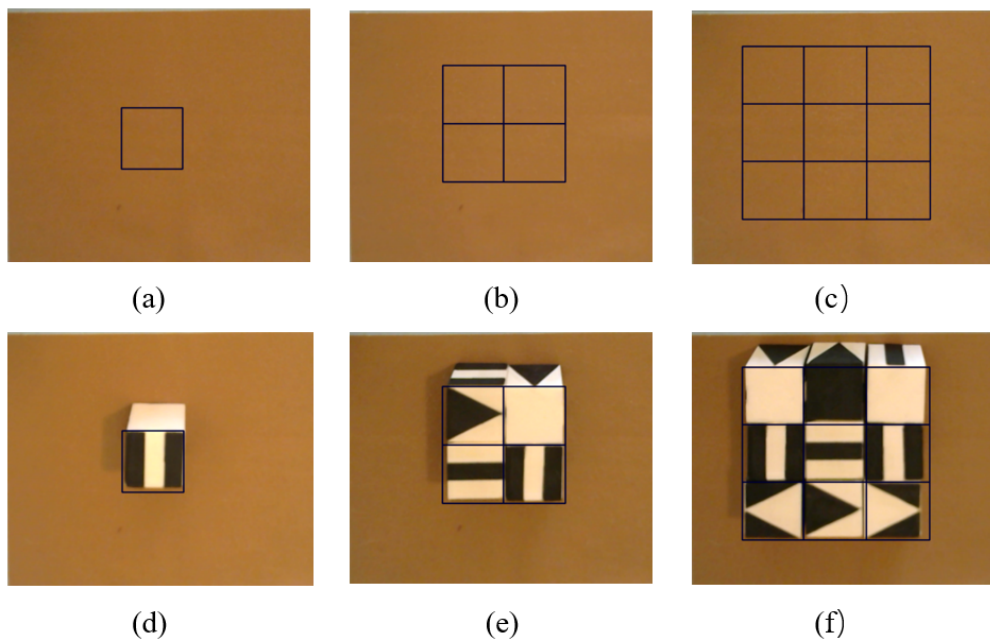


Figure 2.7: (a-c) Square cell outlines displayed in the e-Cube system indicating (d-f) the number of cubes being used in the test item and where the cubes must be located.

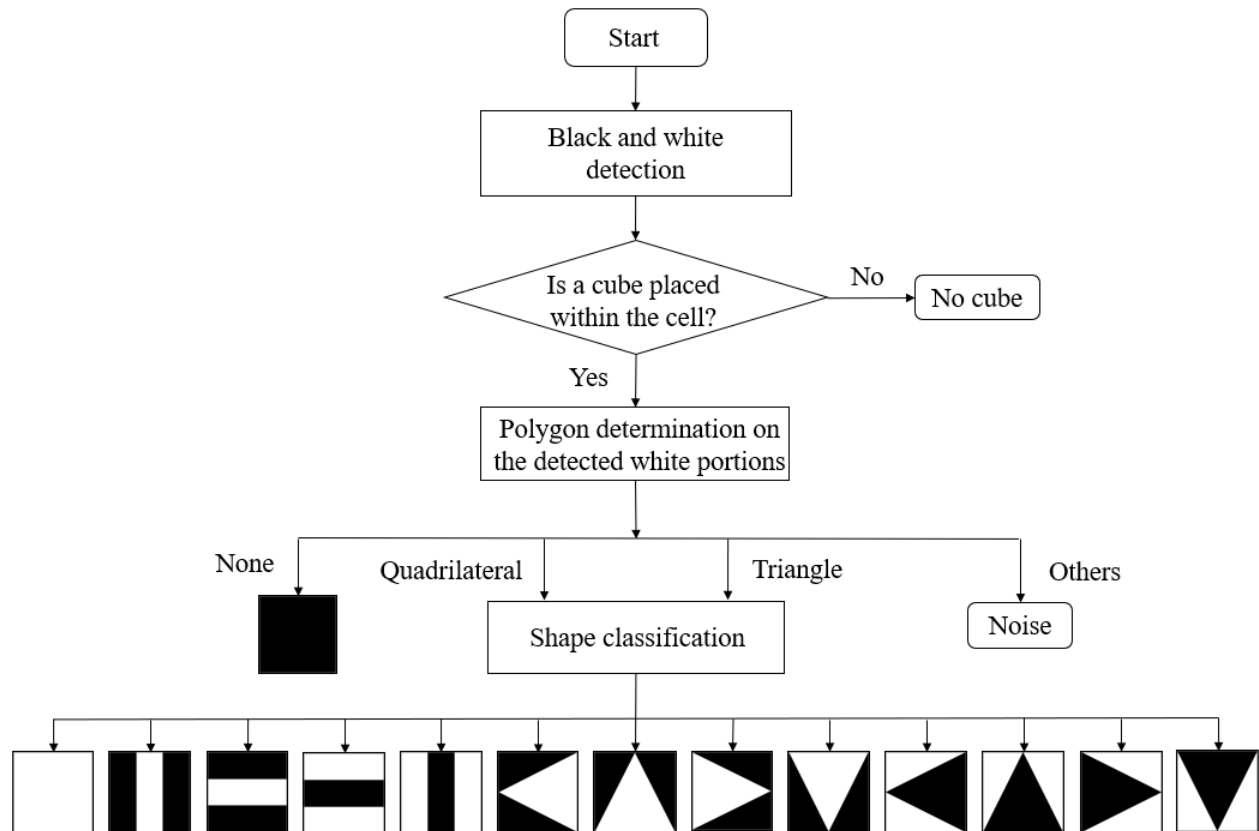


Figure 2.8: Flow chart of the identification of top surface images.

2.2.2.1 Black and white detection

The camera view displays the brown play area where the cubes are placed after transformation. To detect the top face images of the cubes, the system must know if there exists a cube(s). This step was achieved by using black and white detection to identify each of the color portions. It then follows the process to extract the polygons and then classify the shapes. The white detector follows three steps to process the video frames: 1) original video frames are converted to gray-scale; 2) gray-scale frames are denoised and smoothed; and 3) binary thresholding and Otsu thresholding are applied to pre-processed video frames.

Binary threshold replaces every pixel below the constant threshold value with black color; otherwise, flip it to white. However, it may lead to a high false detection rate due to its static threshold value. For example, low illumination results in low intensities so the white portions cannot be

segmented accurately. To make the detection approach accommodate different illuminating conditions, the system also adopted Otsu thresholding to improve the detection performance. Otsu thresholding separates the pixels of a given image into two classes using a computed threshold T that minimizes the weighted sum of within-class variance under the assumption that the image is a bimodal distribution [92]. The weighted sum of the within-class variances in the L gray-level at the threshold T is expressed as:

$$\sigma_W^2 = \omega_0(T)\sigma_0^2(T) + \omega_1(T)\sigma_1^2(T) \quad (2.7)$$

where the weight of the first class $w_0(T)$ and the second class $w_1(T)$ are the total probability of the pixel value in class 0 and class 1 obtained from the gray-level histogram represented as

$$w_0(T) = \sum_{i=1}^T P(i); \quad w_1(T) = \sum_{i=T+1}^L P(i)$$

With-in variances of the two classes are

$$\sigma_0^2(T) = \sum_{i=1}^T [i - \mu_0(T)]^2 \frac{P(i)}{\omega_0(T)}; \quad \sigma_1^2(T) = \sum_{i=T+1}^L [i - \mu_1(T)]^2 \frac{P(i)}{\omega_1(T)}$$

The mean of each class, μ_0 and μ_1 , is simply computed as

$$\mu_0(T) = \sum_{i=1}^T \frac{iP(i)}{\omega_0(T)}; \quad \mu_1(T) = \sum_{i=T+1}^L \frac{iP(i)}{\omega_1(T)}$$

Optimization of (2.7), however, involves second-order statistics (class variance), which is not considered a simple measure. The constant total variance of an image σ is composed of the within-class variance (σ_W) and between-class variance (σ_B), such that

$$\sigma = \sigma_W + \sigma_B$$

Given σ , minimizing the within-class variance σ_W is equivalent to maximizing the between-class

variance σ_B . This is a first statistic problem (class variance), formulated as

$$\sigma_B = \omega_0(T)(1 - \omega_0(T))[\mu_0(T) - \mu_1(T)]^2$$

Although Otsu thresholding theoretically outperforms binary thresholding when two classes (i.e. white and black colors) exist, it fails to cluster pixels into a single class. This method attempts to find a point separating pixels in two classes and thus shows limitations when only one class exists. Therefore, false white detection may occur in the following three scenarios using Otsu thresholding:

- No cube is placed.
- A cube is placed with the white square facing up.
- A cube is placed with the black square facing up.

In the e-Cube system, both binary and Otsu thresholding are used in white detection due to their complementary advantages and limitations. The segmentation outcomes from binary thresholding serve as a “rough detection” identifying if a cube is placed in the designated area with black detection (described below). Fig. 2.9b displays the results of the binary thresholding applied on an original video frame containing some cubes (Fig. 2.9a). In addition, the white portions detected by binary and Otsu thresholding are both treated as a region of interests (ROI) to detect a specific shape (described in Chapter 2.2.2.2).

As for the black detection, using thresholding techniques may lead to false-positive errors caused by the mistakenly captured shadows as well as the brown background and false-negative errors caused by the light reflection that hides the black portions. Therefore, searching a few “black-ish” colors in HSV space ((hue, saturation, value) can simplify the detection because “value” is the only parameter to be adjusted for targeting colors.

With the black and white detection, the system can determine whether a cube is placed into a cell by superimposing the detected portions (Fig. 2.9d). If a cube is placed, then it detects and

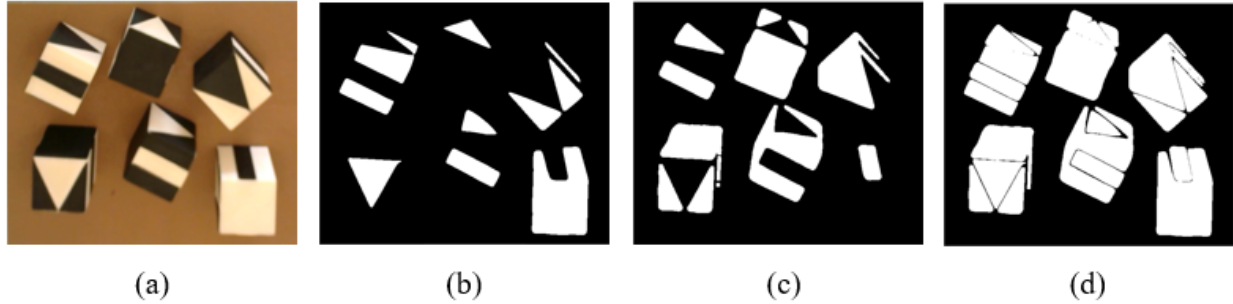


Figure 2.9: (a) Original video frame; (b) detected white portions; (c) detected black portions; and (d) segmented cubes by superimposing the white and black portions.

classifies the shapes.

2.2.2.2 Polygon determination

Taking the ROI (i.e. the segmentation results from binary and Otsu thresholding) as inputs, the system finds a polygon within the ROI based on the number of vertices (N_v), which can be either none ($N_v = 0$), a triangle ($N_v = 3$), a quadrilateral ($N_v = 4$), or others. Accurate detection of a polygon is critical for geometric shape classification. To achieve this, the system follows three steps: 1) convert the ROI images to the one that only contains the shape contour pixels. This was achieved by the algorithm proposed in [93], which can extract both outer and hole borders; 2) apply Ramer–Douglas–Peucker (RDP) algorithm [94] to the contour-extracted images to identify the critical vertices. In the e-Cube system, the number of detected critical vertices can be either 0, 3, 4, or others, which indicates none, a quadrilateral, a triangle, and noises, respectively. A quadrilateral can be either a square or a strip, and a triangle can either be a right-angled or isosceles triangle; none denotes for no polygon; and noises represent the previous three conditions are not satisfied.

2.2.2.3 Shape classification

Shape classification identifies the geometry and orientation of the fourteen shapes from the determined polygon. To do so, each of the shapes is assigned a label (i, j) , as shown in Fig. 2.2g, where $i = 1, \dots, 6$ represents the index of six distinct geometries and $j = 1, \dots, 4$ indicates the

four orientations, i.e., 0, 90, 180, and 270°, considering the rotational symmetry in each shape.

If the polygon in the ROI is either a quadrilateral or triangle, its geometry and orientation are further determined using the geometric features. As for a quadrilateral, the system identifies if it is a square, a strip with a longer vertical side (Fig. 2.10a), or a strip with a longer horizontal side (Fig. 2.10b). Since the aspect ratio is 1 : 1 for the square and 1 : 3 for the strip, the system can distinguish the two by its relative ratio of lengths using a ratio threshold. Unlike a square shape that remains the same configuration, a strip has different configurations when rotated by 90° (Fig. 2.2g). Detecting the orientation of a strip requires further computations. Some level of misalignment (blue dash line in Fig. 2.10 left) must be allowed. Suppose the acceptable angle around the perfect vertical alignment is between $[-\theta, \theta]$, the vertical distance from vertex 0 to 2 (i.e. $y_2 - y_0$) then is $a \cos \theta$ at the angle of $-\theta$ or θ (Fig. 2.10a). Therefore, the strip is believed to be vertically placed if $y_2 - y_0$ is between $[a \cos \theta, a]$. Similarly, the strip is horizontally placed if $y_2 - y_0$ is between $[\frac{a}{3} \cos \theta, \frac{a}{3}]$ (Fig. 2.10b). However, the pixel distance measured from the video frame differs from the theoretical distance, which may lead to inaccurate evaluation. Instead, the intervals of vertical pixel distances (i.e. $y_2 - y_0$) are defined differently for distinctive orientation, thus are used to determine the value of j (i.e. the orientation) when i is either 3 (the shape containing one strip) or 4 (the shape containing two strips). The value of i can be determined by how many white strips are detected.

Similarly, a triangle can be classified if it is one of the shapes in Fig. 2.11 (i.e., a right triangle or an isosceles triangle with its orientation) based on its shape and orientation. The ratio of its shortest length to its largest length is different in the two triangles, thus can be used to determine whether the value of i is 5 or 6. Furthermore, the system can identify the orientation of a triangle by the relative locations of its vertex that is opposite to the shortest side (named vertex V_c) in respective of the other two vertices (Fig. 2.11).

Different from the other thirteen shapes, the black square shape labeled as (1,1) (Fig. 2.2g) is the only one that does not contain white geometries. Therefore, no polygon would be detected by applying the polygon determination on the detected white portions. However, this shape can

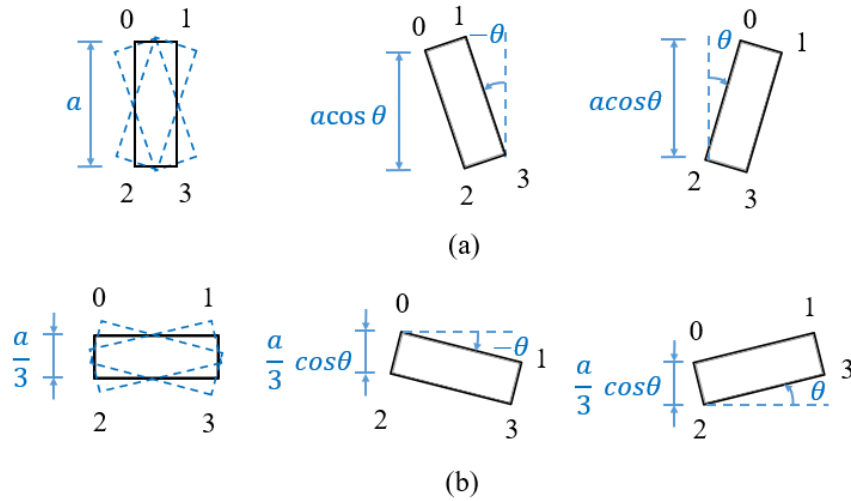


Figure 2.10: Vertical distance of (a) the vertically placed and (b) horizontally placed strip.

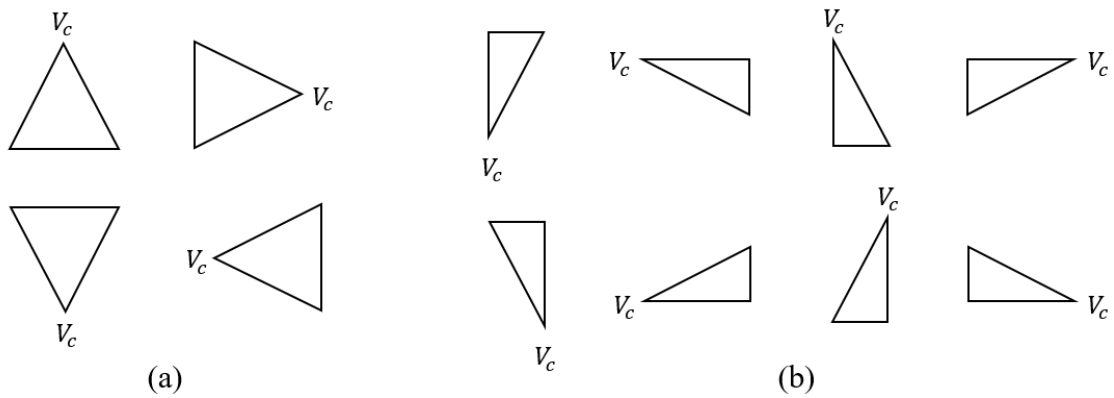


Figure 2.11: Isosceles and right triangle geometries with different orientations.

be identified when the vertex number in the polygon determination is 0 after confirming a cube is placed by the black and white detection.

Note that the system takes the white regions detected from both binary and Otsu thresholding as the ROI for identifying geometry and orientation. As a consequence, the system provides two sets of shape detection results, which are the same in most cases but may be different. For example, Fig. 2.12 shows different scenarios obtained from binary and Otsu thresholding in e-Cube system, in which A, B, C, and D are from binary thresholding and E as well as F are from Otsu's. In general, binary thresholding outperforms in the no cube scenario and the square classification while Otsu's

outperforms in the strip and triangle classification. Based on this conclusion, selection criteria are defined in Table 2.1 to choose a "better" detection result.

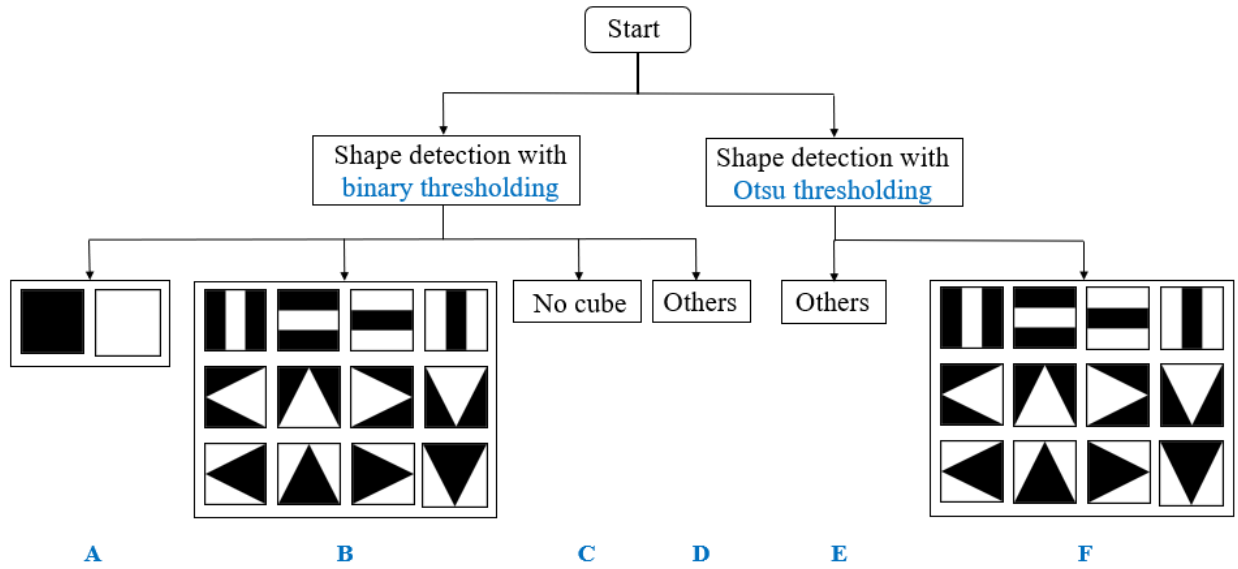


Figure 2.12: Shape detection using binary and Otsu thresholding.

Table 2.1: Selection criteria of the results from binary and Otsu thresholding.

Input		Output
Binary	Otsu	
A	E	A
B	E	E
C	E	C
D	E	E
A	F	A
B	F	F
C	F	C
D	F	F

2.3 Database

Data collected by the e-Cube system must be stored and organized in a way that protects the individually identifiable information while allowing easy ways to access and retrieve the data for analysis. SQLite, MySQL, and PostgreSQL are three popular choices of open-source database management systems (DBMS) that can access, create, store, modify, and query data by Structured Query Language (SQL), a tool to communicate with the database. SQLite requires no server and thus can serve as a portable DBMS in low-memory applications. Security of SQLite, however, is considered to be weaker than the systems that use a server, such as MySQL and PostgreSQL. PostgreSQL faces problems of fewer resources and tools to manage the database compared to the widely used MySQL [95]. Thanks to its better performance on speed, security, and capacity of handling large-scale data, MySQL can provide broad usability in potential commercialization. Therefore, MySQL was selected as the DBMS embedded in e-Cube to store and manage users' data.

2.3.1 Database structure

The e-Cube database consists of four tables - "User", "Visiting", "Behavior", and "Accuracy". The scheme of the entire database structure displayed in Fig. 2.13 shows the associations and attributes. Specifically, as a parent table, "User" stores user information including date of birth, gender, game version (i.e. fixed e-Cube or adaptive e-Cube described in Chapter 5) the examinee played, and a unique and randomly assigned auto-generated ID (i.e. primary key in SQL). This ID represents a user's identity in a confidential way. Note that it is of significant importance not to use users' names to identify the data belonging due to the risk of data leaking, no matter how small the possibility is. "Visiting", "Behavior", and "Accuracy" are the child tables. A foreign key named User ID is linked with the primary key and uses the same number as it in the parent table so that the data in the child tables can be identified to whom it belongs to. "Visiting" stores user visiting information including how many times this user has been played the game (Total Visiting Times), login time of each visit (Login Time), what the game order of each visit is (Game Order), and

whether or not the user finishes all the games for each visit (Finish or Not). Both “Behavior” and “Accuracy” record the Current Visiting Index (i.e. 1st time, 2nd time, etc.), the Game Type, and the Item Number. Besides these shared attributes, “Behavior” also records the real-time detection results (Detected Results) and its detected time (Time), while “Accuracy” stores the answer of each test item (Answer), whether or not the user provides the correct answer (Correctness), item complexity (Current Difficulty), the highest complexity of the item that has been correctly answered (Maximum Difficulty), the lowest complexity of the item that has been wrongly answered (Minimal Difficulty), the number of the Path-Tracking items that have been correctly answered (Correct P-T Item Number), the number of the Path-Tracking items that haven been correctly answered but are not with the shortest path (Multi-path P-T Item Number), and the number of the Path-Tracking items that haven been wrongly answered (Wrong P-T Item Number).

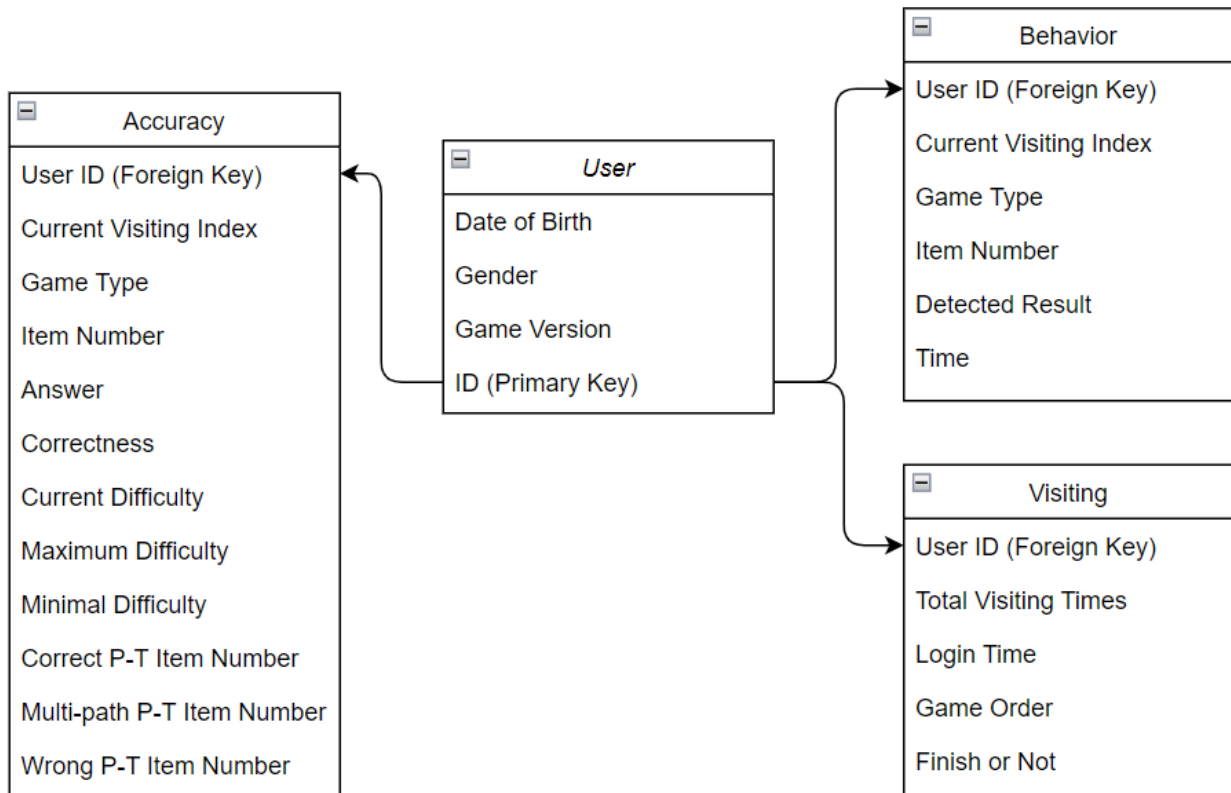


Figure 2.13: Scheme of the e-Cube database structure.

The Python code for constructing the database and the four tables without passwords enclosed are shown in Appendix B.1. The name of the attributes in the code may be different from the one in Fig. 2.13, but the displaying orders are identical.

2.3.2 Data collection using MySQL

This subsection describes 1) how to write/insert data into the four tables using MySQL writing commands; 2) when should they be inserted. The writing commands for the tables are displayed in Appendix B.2. As for "User" and "Visiting" tables, the writing commands are only activated if the "Store Data" checkbox is enabled and the personal information is filled in once users click the "Sign In" button on the access page (described in Chapter 3). Since "Behavior" records the real-time detection of all the manipulations and corresponding time, the writing of this table is performed during the entire game. "Accuracy" contains the final answers provided by the users, so the writing of it is activated once the final answer is provided. Note that the Writing command of "Accuracy" provided in Appendix B.2 only contains the one used for e-Cube^A. The process for other games, including e-Cube^S, e-Cube^{M1}, e-Cube^{M2}, e-Cube^P, and e-Cube^Z, is similar to this command. However, there exist some differences due to their different game settings and game versions. For example, e-Cube^P records three more parameters including Correct P-T Item Number, Multi-path P-T Item Number, Wrong P-T Item. If the items are automatically generated (i.e. in adaptive e-Cube), Current Difficulty, Maximum Difficulty, and Minimal Difficulty should be inserted. The last writing command is executed when users click the "Exit" button on the last page (described in Chapter 3) after all the games are played: flipping the binary attribute "finish" in "Visiting" from 0 to 1 (the last command in Appendix B.2). By doing so, the system can determine whether the user has finished all the games.

The written data can be queried and extracted by MySQL commands for further data management and analysis. Appendix B.3 shows how to run data queries via the assigned ID number for a participant (described in Chapter 5).

3. GAME DESIGN

This chapter presents six games and a graphical user interface (GUI) designed for the e-Cube system based on the developed technology. The six games include Assembly (e-Cube^A), Shape-Matching (e-Cube^S), Sequence-Memory (e-Cube^{M1}), Spatial-Memory (e-Cube^{M2}), Path-Tracking (e-Cube^P), and Maze (e-Cube^Z). The first three games are reconstructed versions of the previously designed games [1] and the latter three are newly added to further explore their potential roles in cognitive assessment.

3.1 Design of Six e-Cube Games

The games aim to autonomously assess an individual's cognitive abilities including fine motor skills, visual-motor integration, perceptual reasoning, and working memory by capturing correctness, completion time, and the number of movements, meanwhile providing an enjoyable experience that may reduce anxiety and pressure. These six types of games and the cognitive skills expected to be associated with are summarized in Table 3.1.

Table 3.1: Six e-Cube games and their expected associated cognitive skills.

	e-Cube ^A	e-Cube ^S	e-Cube ^{M1}	e-Cube ^{M2}	e-Cube ^P	e-Cube ^Z
Fine-motor skills	✓			✓	✓	✓
Visual-motor integration	✓			✓	✓	✓
Perceptual reasoning		✓			✓	✓
Working memory			✓	✓		

3.1.1 Assembly: e-Cube^A

e-Cube^A involves assembly tasks, asking an examinee to match the given assembly figure shown on the screen using four or nine cubes, similar to the BD subtest of WAIS. Besides the final assembly correctness, the performance at each manipulation step, including the provided top

surface images and accuracy with its timestamp, is also recorded by the system autonomously. Twenty sample items of e-Cube^A are displayed in Fig. 3.1.

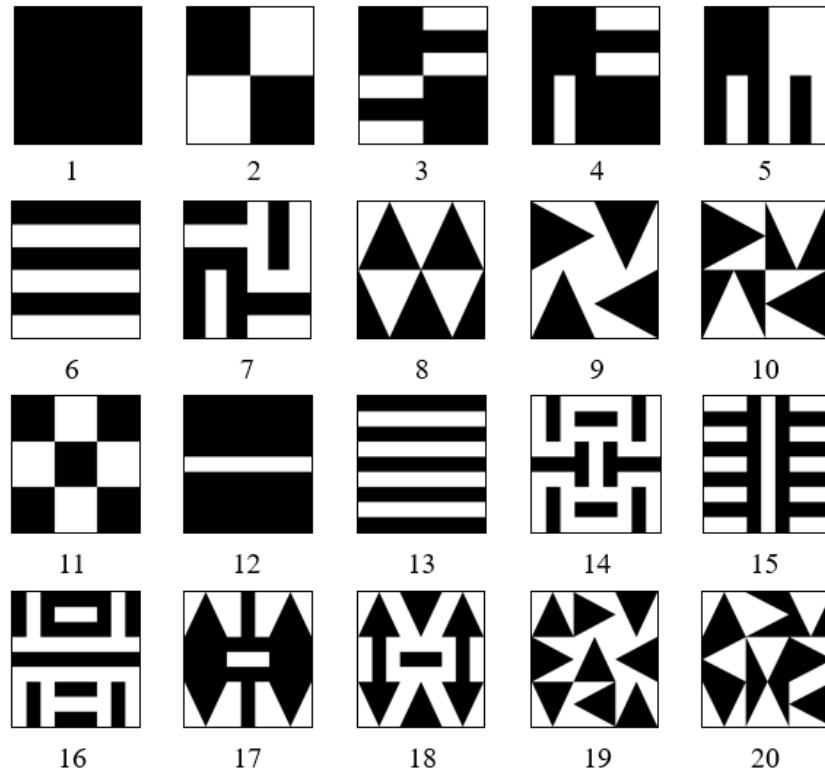


Figure 3.1: Sample items of e-Cube^A (adapted from [1]).

3.1.2 Shape-Matching: e-Cube^S

Each e-Cube^S item is formed by a visual geometric design containing 2×2 or 3×3 shapes with a missing piece. The examinee is asked to find a shape that best completes the pattern by placing a block with the desired image facing up. The system captures the provided answer, correctness, and completion time. Ten sample items of e-Cube^S and their answers are displayed in Fig. 3.2.

3.1.3 Sequence-Memory: e-Cube^{M1}

e-Cube^{M1} is designed to measure working memory, which is an important index to predict cognitive skills [96]. The system displays a sequence of shapes, where each shape appears for

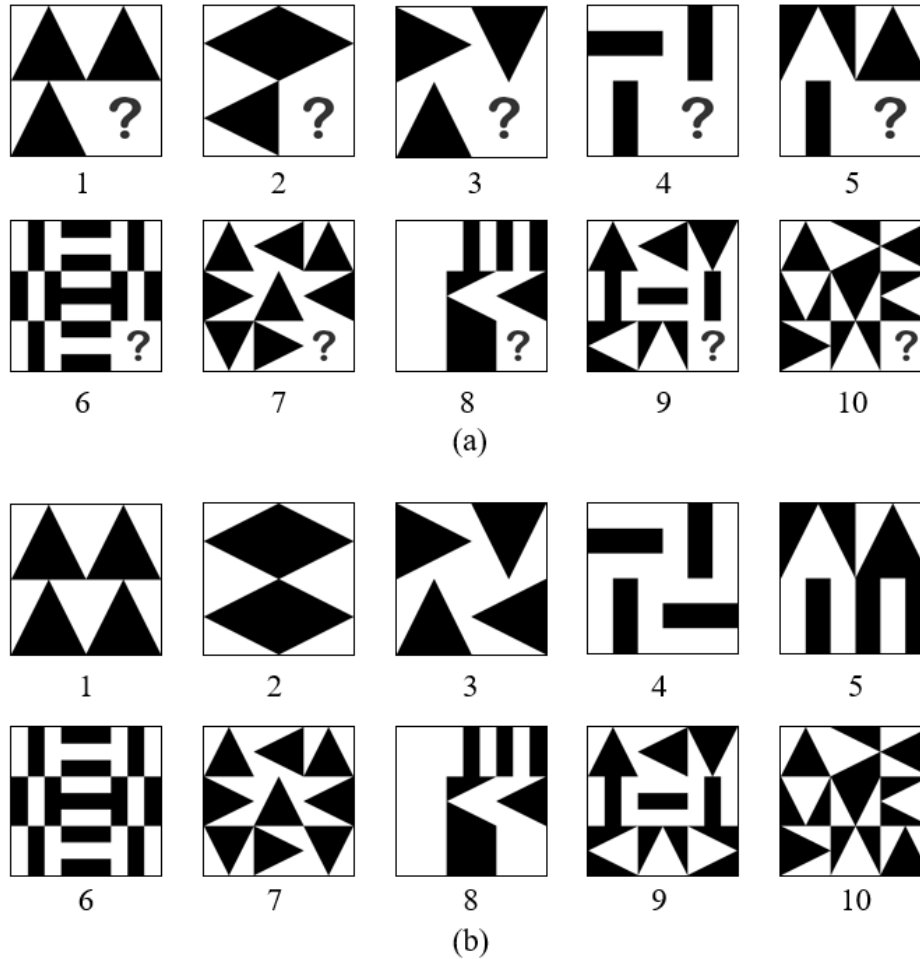


Figure 3.2: (a) Sample items of e-Cube^S and (b) their answers (adapted from [1]).

one second then disappears. The examinee is asked to memorize the sequence of the displayed shapes and provide an answer using a single cube by sequentially placing the shapes in the exact order. Fig. 3.3 shows 16 sample items of e-Cube^{M1}. This game discontinues once two consecutive incorrect answers are obtained to maintain rapport and avoid fatigue. The system records the detected shapes with timestamps in each sequence and the final accuracy.

3.1.4 Spatial-Memory: e-Cube^{M2}

In e-Cube^{M2}, each item contains an assembly pattern of two or more shapes for five seconds, and then they disappear. The examinee is asked to memorize the assembly configurations and provide the answer by assembling the cubes to match the pattern. This game targets measuring

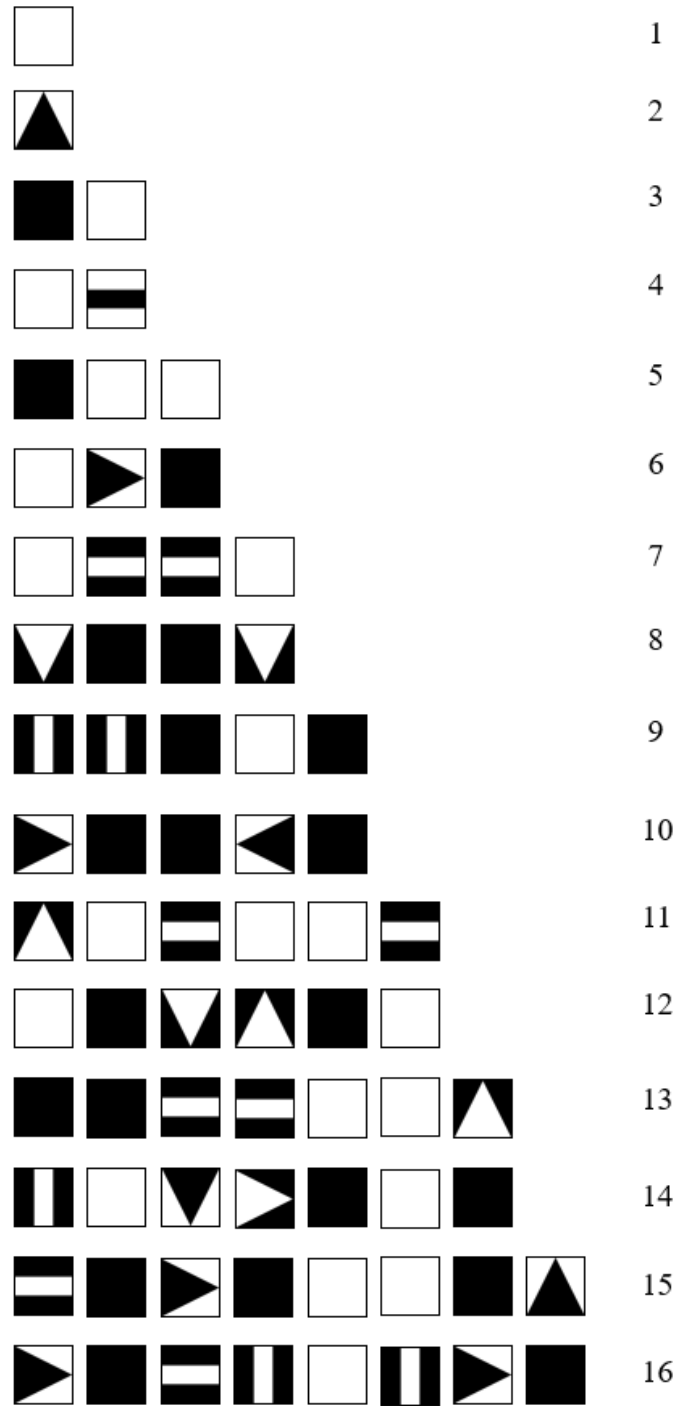


Figure 3.3: Sample items of e-Cube^{M1}.

one's working memory skills and fine-motor ability. While the types of items are similar to those in e-Cube^A, the assembly configurations in e-Cube^A do not display outlines around individual

shapes while e-Cube^{M2} have these lines visible. The detected shapes, completion time, accuracy, and the percent accuracy (e.g., 75% correct if three out of four shapes in a 2×2 assembly item are correctly memorized and reconstructed) are measured for each item in this game. Ten sample items are shown in Fig. 3.4.

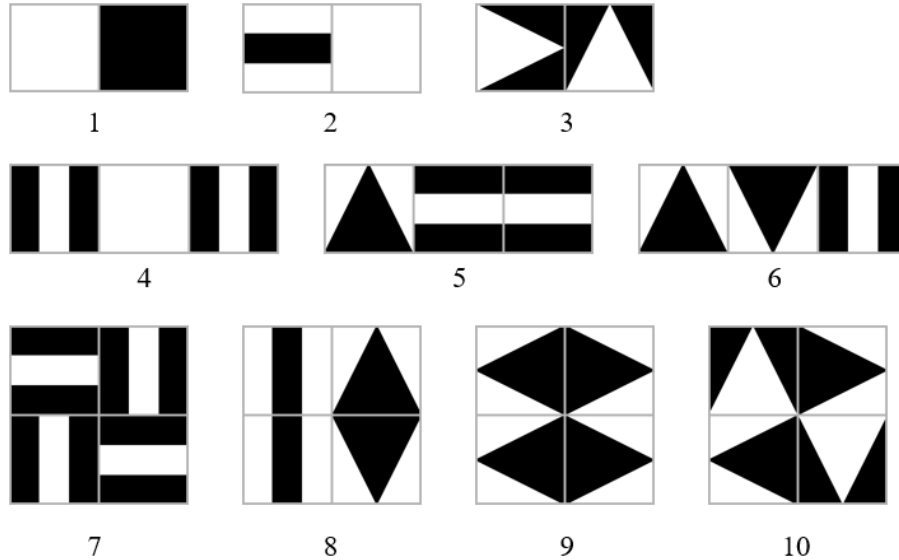


Figure 3.4: Sample items of e-Cube^{M2}.

3.1.5 Path-Tracking: e-Cube^P

Given a connected path, the examinee is asked to recreate the path by moving a cube with the white surface facing up to connect the dots on a 5×5 grid in e-Cube^P. With the known starting point and endpoint, the goal is to trace the path using the minimum distance of movements. The examinee hits one of the blue circles using one cube with the white surface facing up to start the game and turn the black side up when finished. The total number of dot-to-dot connections (dots are also referred to as nodes) and deviation from the displayed path are measured in addition to the correctness and completion time for each item.

Technically, the shape classification algorithm is continuously executed inside each invisible

cell around each dot. Path connection only activates after the white surface hits the start point. If one or more than one cell detects a black surface while no white surfaces are detected, the algorithm ends and goes to the next round. Mathematically, the concept of the Euler path and circuits for undirected graphs forms a foundation for creating the e-Cube^P items. An Euler path is a walk through the undirected graph using every edge only once, while an Euler circuit is defined as an Euler path sharing the same point for starting and stopping [97]. Some examples of e-Cube^P using Euler paths and circuits are displayed in Fig. 3.5.

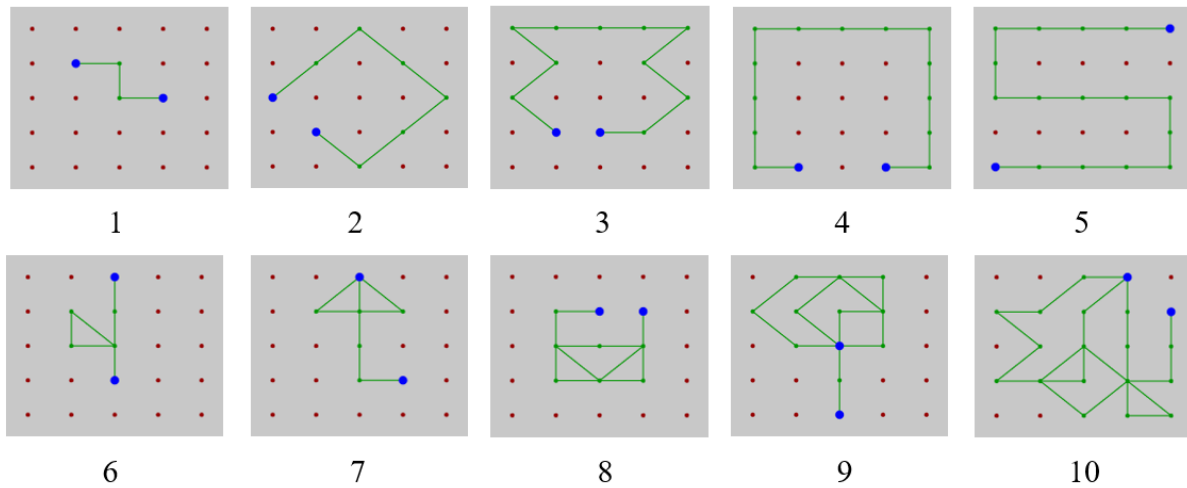


Figure 3.5: Sample items of e-Cube^P.

3.1.6 Maze: e-Cube^Z

e-Cube^Z asks the examinee to find the shortest path from the entrance to the goal point in a maze by moving one cube with the white side facing up. Each maze is generated from a 12×16 binary matrix using the Prim's maze generator [98], where 0 represents an aisle (white) and 1 represents a wall (black). The shortest paths are solved automatically by the A^* algorithm [99]. Similar to e-Cube^P, the total length of the path and deviation from the shortest path solution are measured in addition to the final correctness and completion time for each item. The examinee is asked to hit the entrance using one cube with the white surface facing up to start the game and

place the black surface up to end the game. Ten sample items of e-Cube^Z and their shortest paths solved by A^* are displayed in Fig. 3.6.

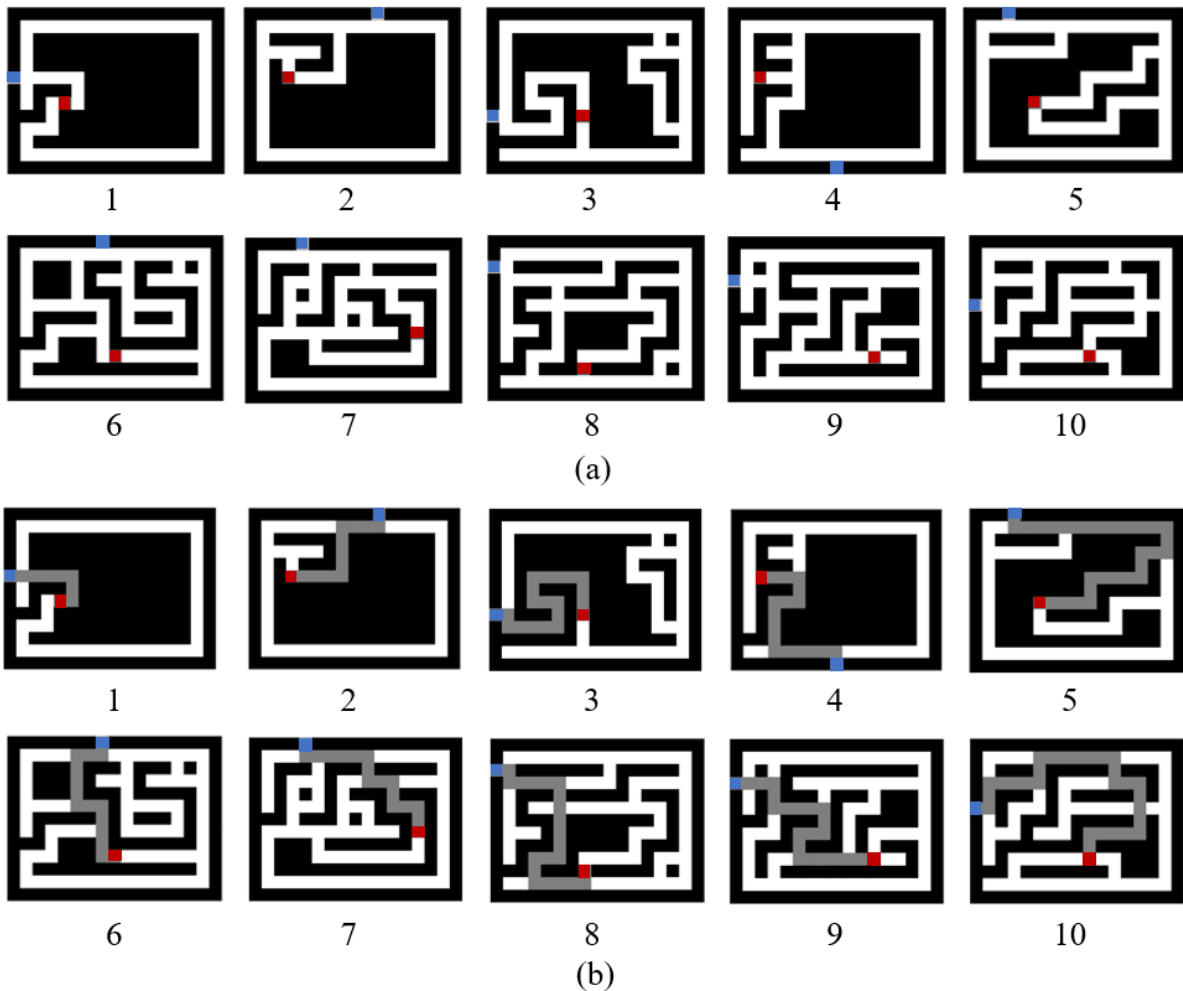


Figure 3.6: (a) Sample items of e-Cube^Z: entrances in blue and goal points in red and (b) their shortest paths solved by A^* algorithm.

3.2 Graphical User Interface

GUIs allow users to achieve human-computer interaction in high efficiency. A well-designed GUI, especially in games, renders text interpretations, graphical/video displays, audio outputs, instructions, user action receptions, and fun features. The current version of the GUI for the e-

Cube games focuses on ease-of-use and technical functionality while future improvements will focus on user-engaging features to make the games interesting and enjoyable. The top 5 popular python GUI frameworks for building GUIs consist of Tkinter, Kivy, PyQt, WxPython, and PySide [100]. Compared to other platforms, Kivy outperforms in multi-touch profitability, compatible system diversity (it can be used in Windows, Linux, OS X, Android, iOS, and Raspberry Pi), and a large number of the available mature toolkits. It is also capable to integrate the algorithms in OpenCV and other Python libraries. These advantages make Kivy the proper GUI platform for e-Cube.

Three types of user interfaces were built in e-Cube using Kivy: an access page, instruction pages, and game pages. The access page (Fig. 3.7) allows the user to input their personal information to register or log in. In addition to the date of birth and gender, the current login page asks which version of the game the user will play or will be assigned for. This selection was added because two versions of the e-Cube games were developed for evaluations (i.e., fixed e-Cube or adaptive e-Cube described in Chapter 5). Users can also permit data storing for game evaluation and video recording for facial expression analysis by clicking the checkbox labeled "Store Data" and "Record Facial Expression" on the access page. The functions only activate once the corresponding checkbox is enabled. The instruction pages deliver essential step-by-step guidance and descriptions to users. Although there are a few instruction pages, the designs of these pages are similar, as shown in Fig. 3.8. It consists of a title, the main content description, a "Mute" checkbox allowing users to mute the audio narratives, a "Close" button to exit the app, a "Back" button returning to the previous page (if have), and a "Continue" button exceeding to the next page (if have). A game page displays a game title on the top, a question screen (if have), an answer screen (i.e., real-time video view from the camera), and a progress bar at the bottom (only in the fixed version of e-Cube). Examples of the game pages for all six games are shown in Fig. 3.9 - 3.16. Each of e-Cube^{M1} and e-Cube^{M2} has two types of game pages, item displaying and item recalling pages, due to their game settings.

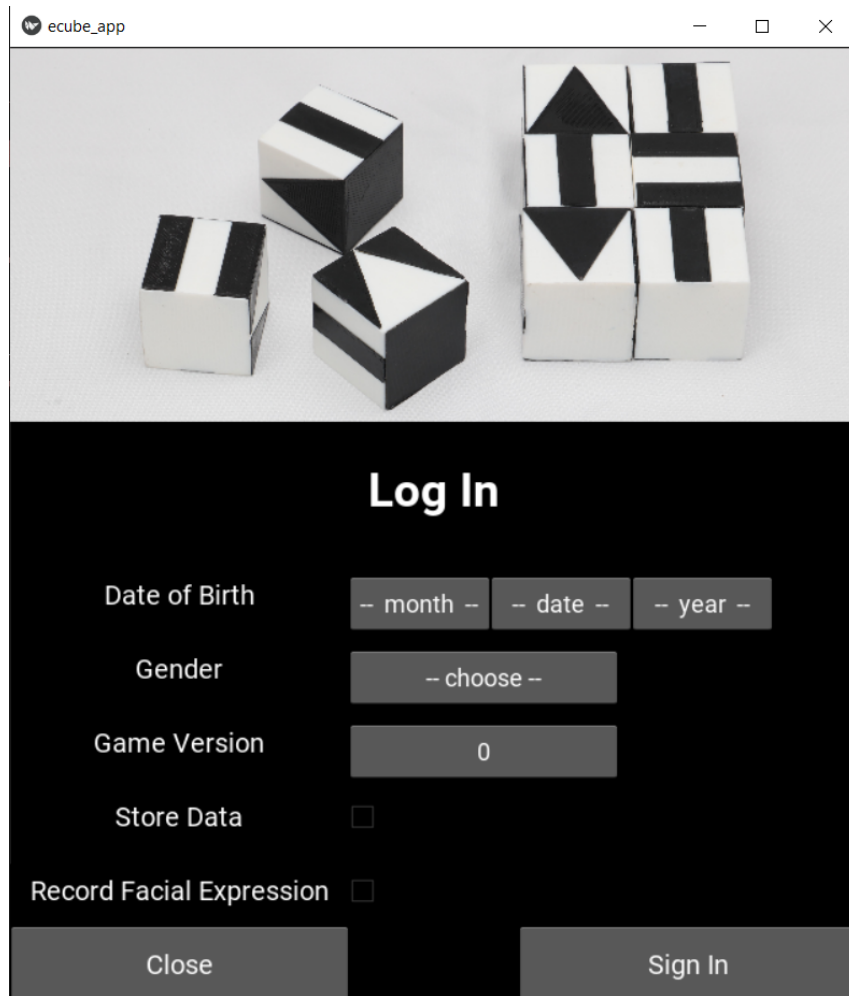


Figure 3.7: Access page of e-Cube.

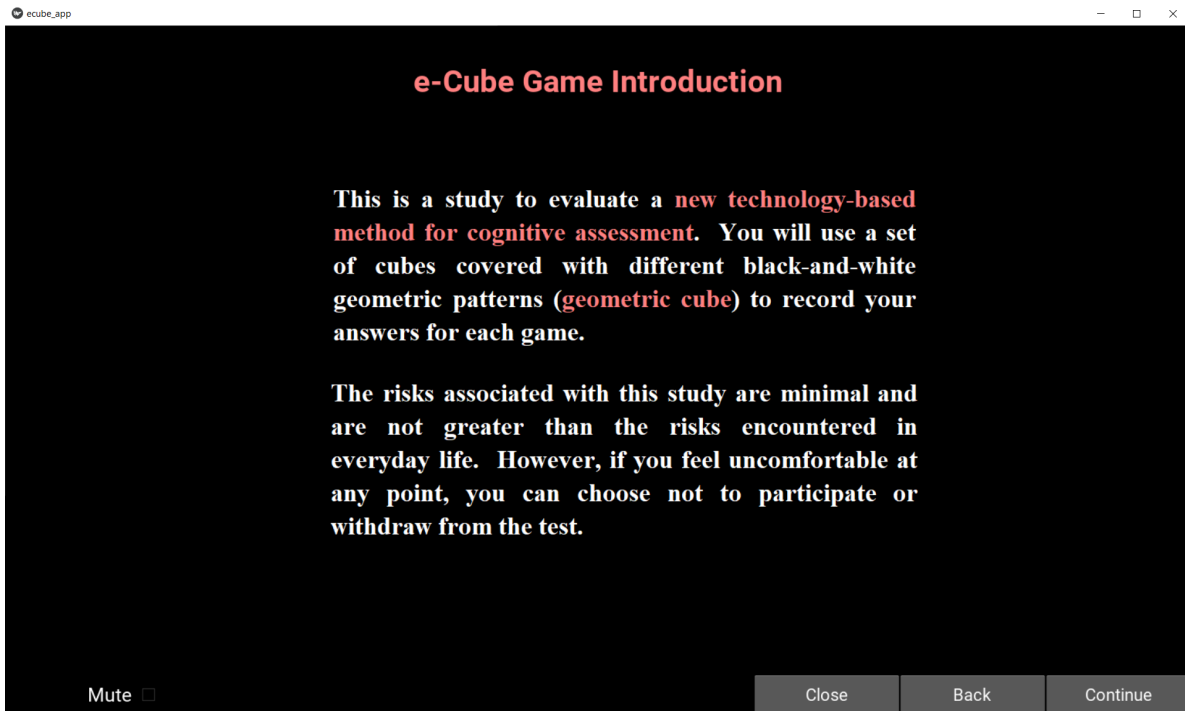


Figure 3.8: Instruction page of e-Cube.

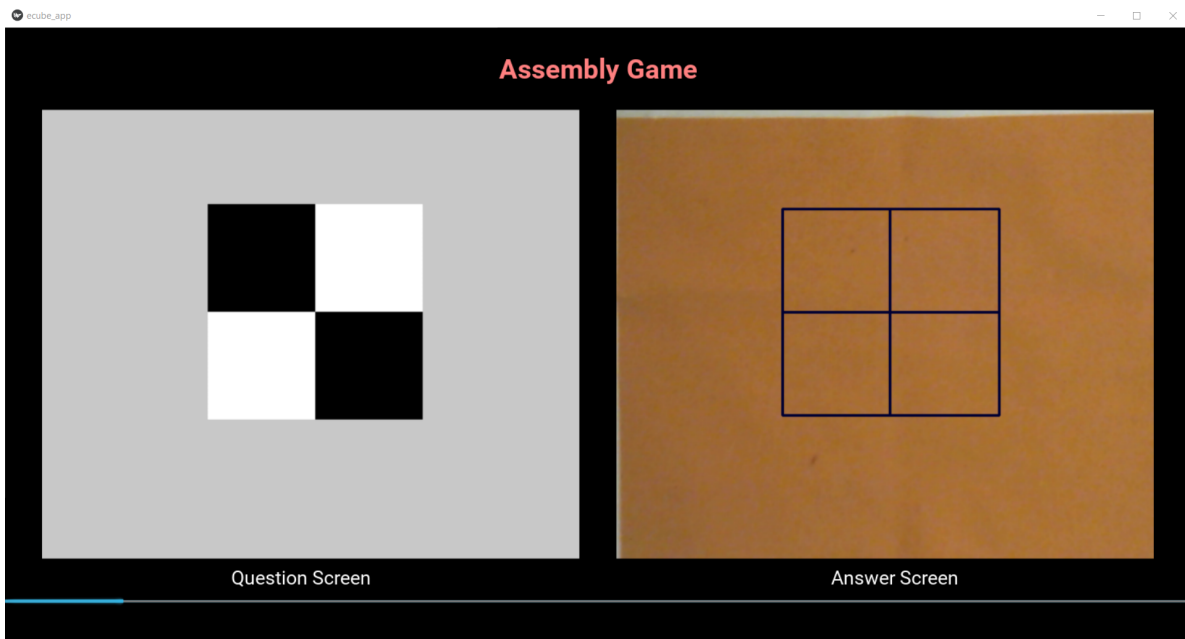


Figure 3.9: Game page of e-Cube^A.

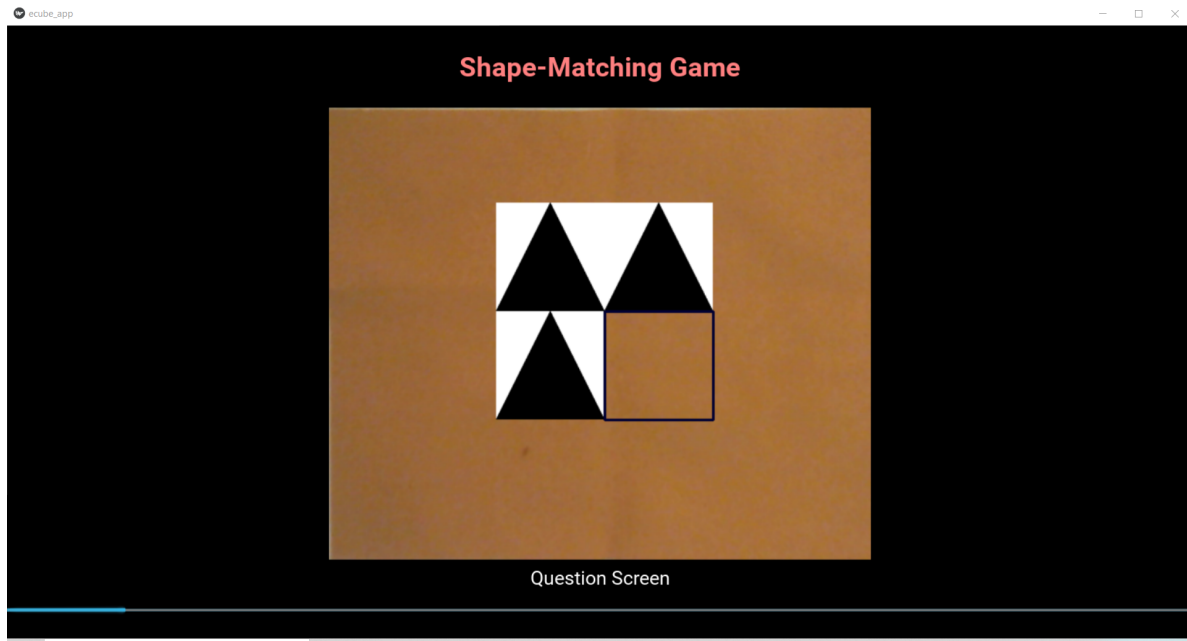


Figure 3.10: Game page of e-Cube^S.

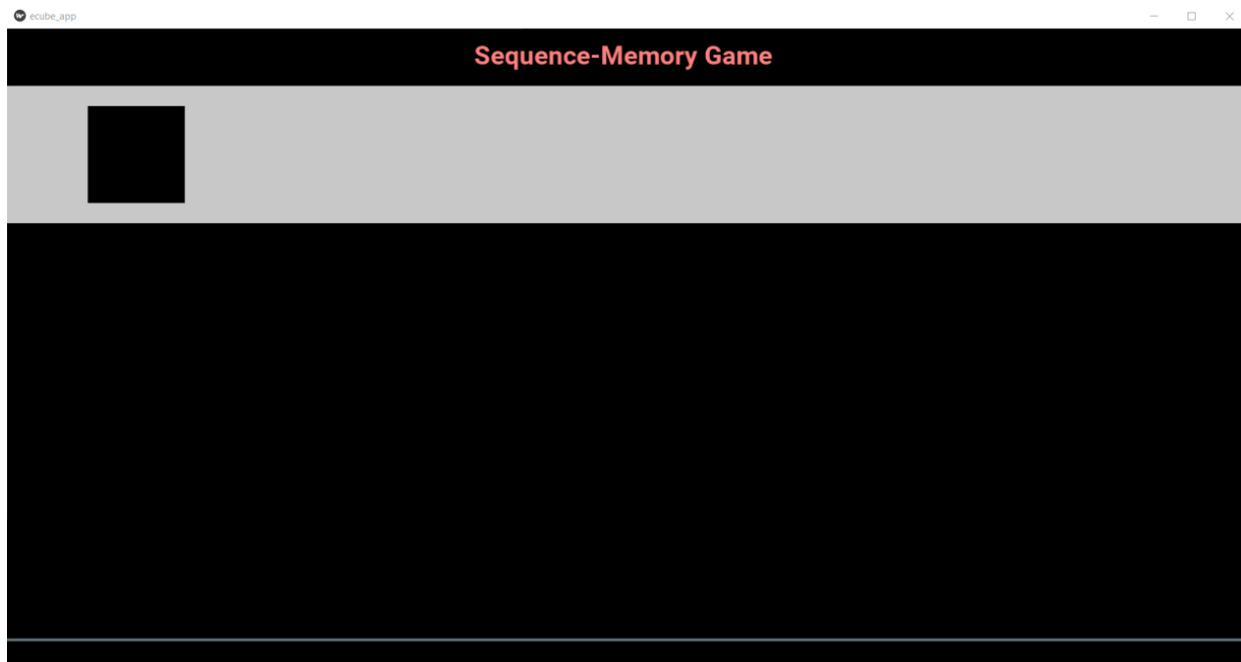


Figure 3.11: Game page of e-Cube^{M1} - item displaying.

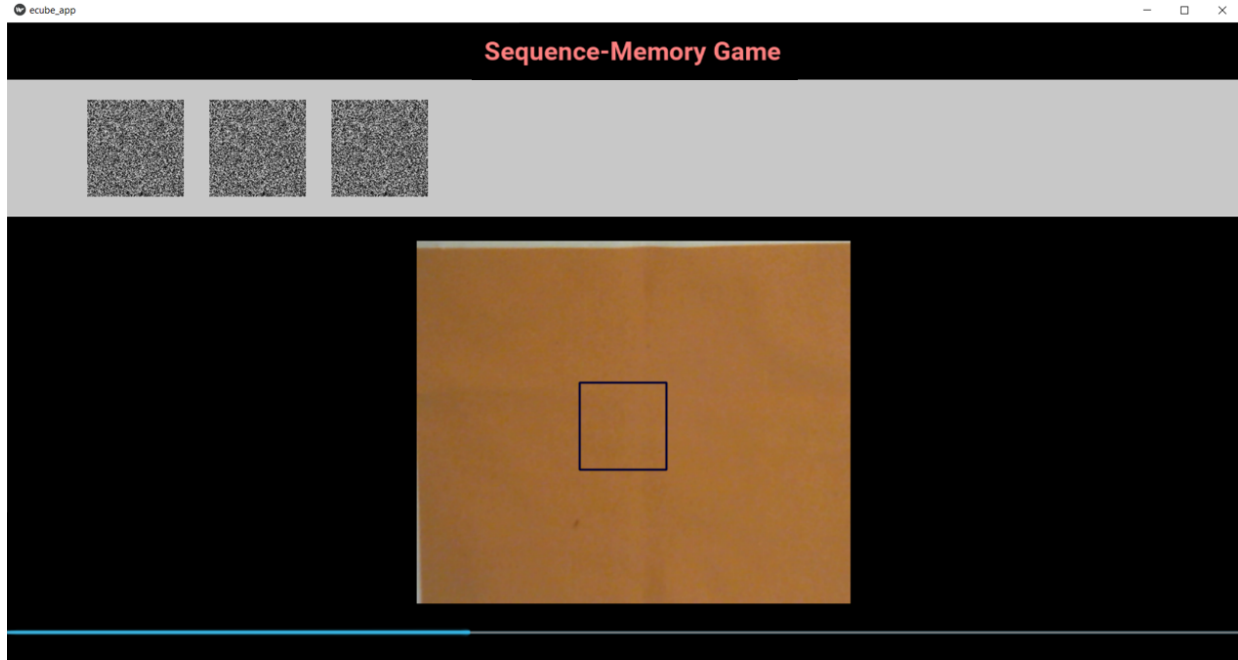


Figure 3.12: Game page of e-Cube^{M1} - item recalling.

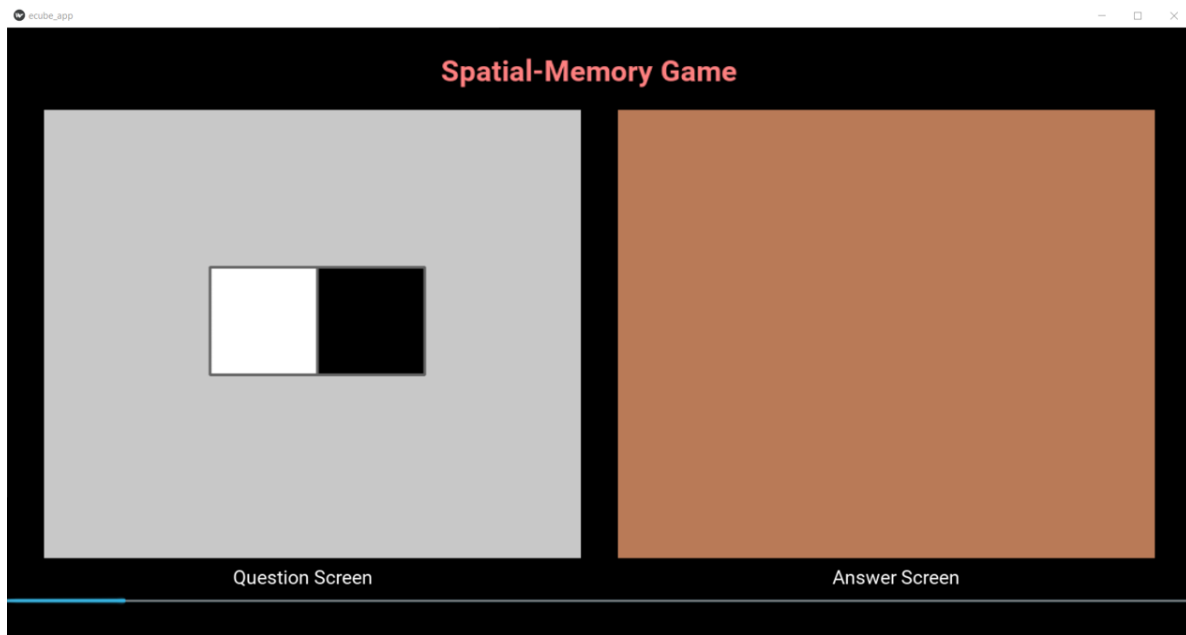


Figure 3.13: Game page of e-Cube^{M2} - item displaying.

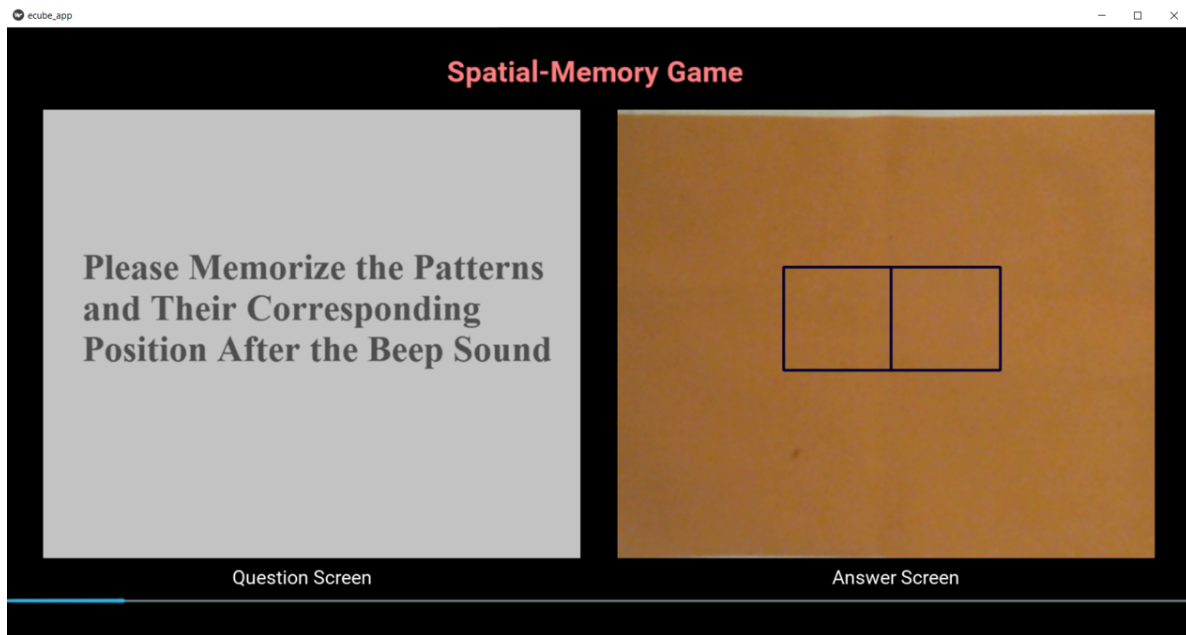


Figure 3.14: Game page of e-Cube^{M2} - item recalling.

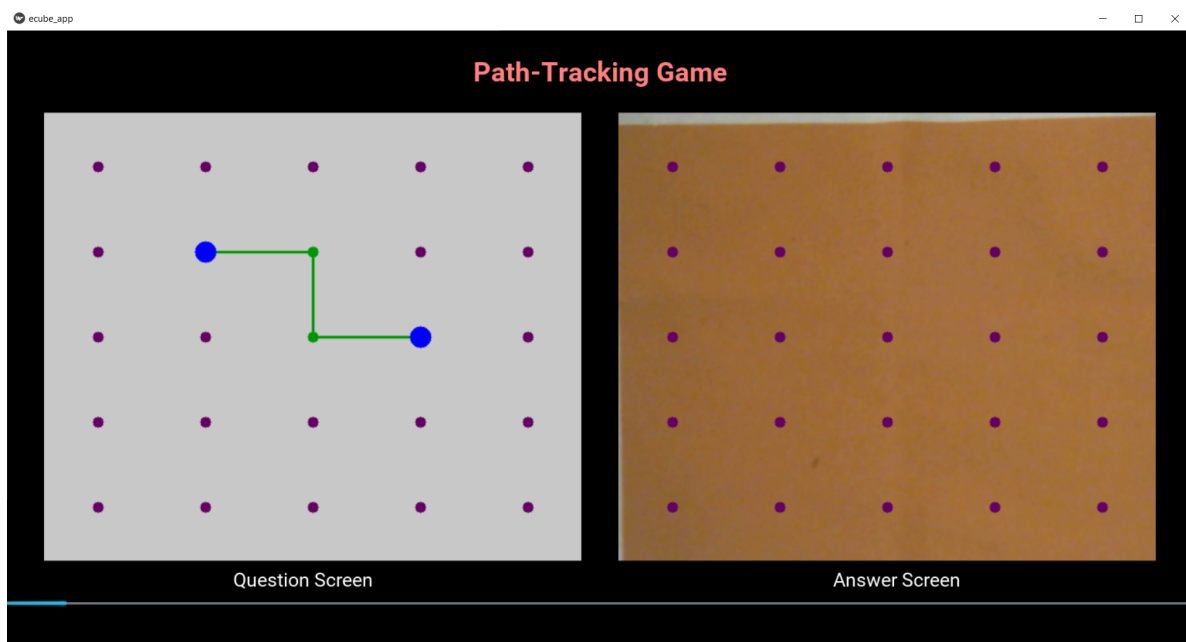


Figure 3.15: Game page of e-Cube^P.

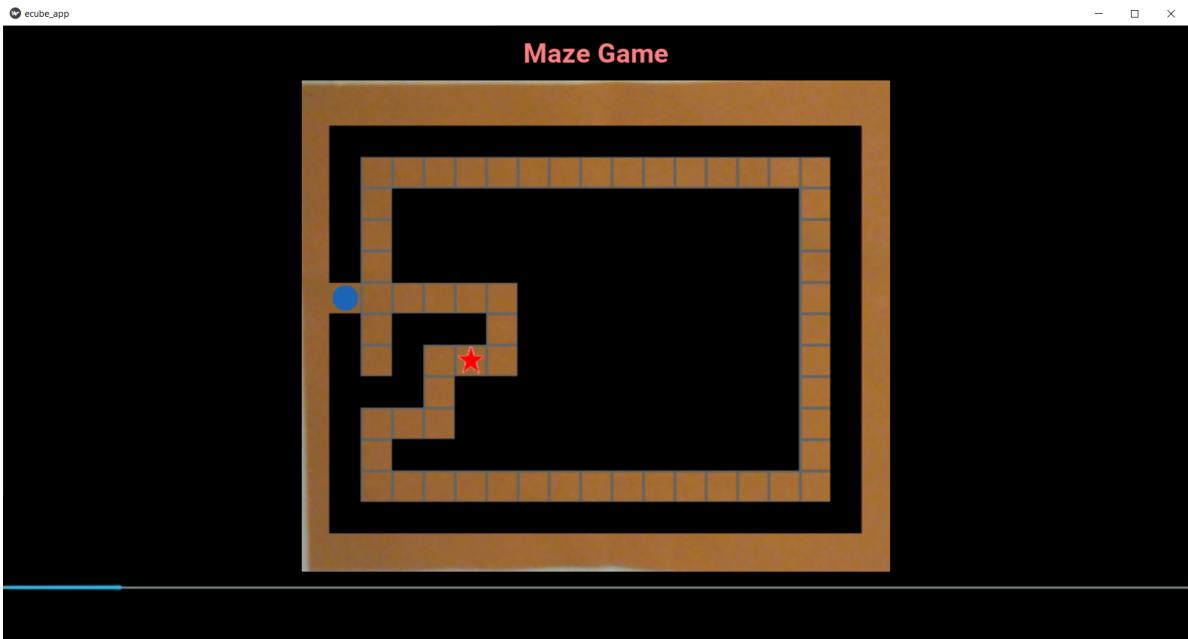


Figure 3.16: Game page of e-Cube^Z.

4. DEVELOPMENT OF ADAPTIVE GAMES

This chapter presents the methods and algorithms developed and adopted for establishing the adaptive e-Cube games, including the computational measures of play complexity and adaptive game generators for individual games. The adaptive e-Cube games allow personalized assessment based on an individual's real-time performance.

4.1 Computational Measures of Play Complexity

The difficulty is a subjective attribute and is affected by one's cognitive skills as well as physical and mental status. The difficulty of correctly finishing an item in e-Cube is called *play complexity*. Computational measures of play complexity aim to quantify the complexity associated with the geometric images and patterns used in the e-Cube games, and thus can be used to guide the system to generate *easier* or *harder* items based on the performance on the previous items. This can also be used to set a pre-defined starting point of the test given the known cognitive status of the examinee. The computational measures of play complexity, therefore, form the technical foundation for generating adaptive test items in e-Cube. Shannon's entropy [101] measuring data uncertainty, randomness, or disorder is adopted in the e-Cube games to quantify the difficulty of each item. The mathematical definition of Shannon entropy is given by

$$H = - \sum_{i=1} P_i \log_2(P_i) \quad (4.1)$$

where P_i is the probability of i th event. If the probability is evenly distributed in one set, i.e. $P_1 = P_2 = \dots = P_n = \frac{1}{n}$, Shannon entropy is simplified as

$$H = - \sum_{i=1}^n \frac{1}{n} \log_2\left(\frac{1}{n}\right) = \log_2(n) \quad (4.2)$$

Shannon entropy, however, can only measure the disorder of compositional information. This is often insufficient to capture difficulties associated with assembly configurations. For example, It

has limitations on disorder measuring for an image that contains position information. To address this, several approaches have been introduced in previous literature to analyze configuration data [102]. Most of these approaches, filtering, neighbor counting, and distance computing between same/different value(s), were developed for processing a larger amount of data, such as images with numerous pixels. They have poor performance and low reliability if the data size is small, such as the items in e-Cube that involve patterns with only 1×3 , 2×2 , or 3×3 grids (e.g. items in e-Cube^A, e-Cube^S, e-Cube^{M1}, and e-Cube^{M2}). However, *Entropy based on the gray-level co-occurrence matrix* can analyze the disorder associated with the matrix-like or grid-like data regardless of the size, so it is suitable and adopted to measure the configurational disorder in e-Cube.

The gray-level co-occurrence matrix (GLCM) was originally defined as the distribution of co-occurring pixel values at a given offset $(\Delta x, \Delta y)$ in a matrix form. This was first proposed to classify image textures in gray-scale [103]. Researchers have used the GLCM in image entropy computations to group the texture of some rocks into seven classes based on their color disorder [104]. For an image with an $m \times n$ dimension and L gray level, its GLCM (f) is represented as an $L \times L$ square matrix, given by

$$f_{\Delta x, \Delta y}(i, j) = \sum_{x=1}^n \sum_{y=1}^m \begin{cases} 1, & I(x, y) = i \text{ and } I(x + \Delta x, y + \Delta y) = j \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

The offset $(\Delta x, \Delta y)$ is typically defined as horizontally, vertically, and/or diagonally adjacent pixels. Horizontally adjacent pixels can be paired along 0° and/or 180° ; vertically adjacent pixels can be paired along 90° and/or 270° , and diagonally adjacent pixels can be paired along 45° , 135° , 225° , and/or 315° (Fig. 4.1a). The offsets formed by different directions can be combined to increase the sensitivity of identifying textures. Based on the obtained f , its entropy can be

computed as:

$$H = - \sum_i \sum_j \left(\frac{f(i,j)}{\sum_i \sum_j f(i,j)} \right) \cdot \log \left(\frac{f(i,j)}{\sum_i \sum_j f(i,j)} \right) \quad (4.4)$$

The play complexity for each e-Cube game combining Shannon entropy that measures the compositional disorder and the entropy based on the GLCM that measures the configurational disorder is defined as follows.

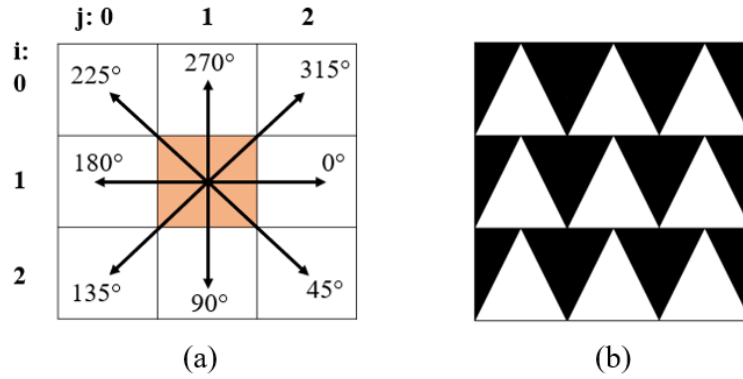


Figure 4.1: (a) Possible directions for computing a GLCM and (b) an item with a small D_{config} value.

4.1.1 Play complexity measure for e-Cube^A

Total play complexity (D_t) of an e-Cube^A item is computed based on the compositional complexity D_{compos} and configurational complexity D_{config} , given by

$$D_t = \eta D_{compos} + D_{config} \quad (4.5)$$

where D_{compos} represents the complexity associated with individual shapes (i.e., the number of shapes and their rotational symmetry); D_{config} captures the orientation and color differences among the shapes in the way they are arranged; and η is a sigmoid function that equals to $\frac{1}{1+e^{-1.5 \times D_{config}}}$. ηD_{compos} is influenced by the value of D_{config} . If an item has a small D_{config} value, i.e. the configuration is simple (e.g. the item shown in Fig. 4.1b), the visual complexity is expected to be lower

than the items formed by the same geometries but with different orientations.

Using Shannon entropy, D_{compos} is defined as

$$D_{compos} = \sum_m [\log_2(n_1) + \log_2(n_2)] \quad (4.6)$$

where m is the total number of the shapes in an item, n_1 is the number of the available distinctive shapes among the six faces of a cube ($n_1 = 6$ for e-Cube) and n_2 is the number of the distinctive orientations by rotating this shape by 0° , 90° , 180° , and 270° ($n_2 = 1$ for squares, $n_2 = 2$ for strips, or $n_2 = 4$ for triangles). Fig. 4.2 shows three examples with the same D_{compos} value, while their visual complexities are different. This shows the insufficiency of only using D_{compos} without measuring the configurational information. To accommodate the above limitation and capture the the configurational disorder, the weighted entropy based on GLCM is used to evaluate D_{config} as a supplement. To do so, a unique index number from 1 to 14 is assigned to each distinctive shape, as shown in Fig. 4.3, for computing f (i.e. the GLCM).

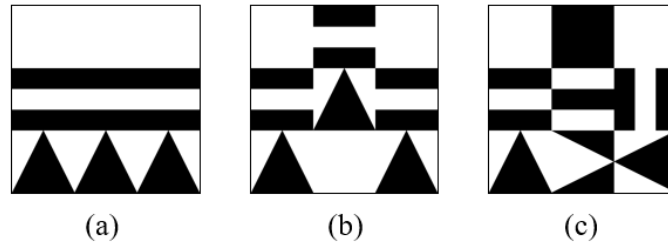


Figure 4.2: Items with identical compositional complexity.

First, all the shapes of an item used in e-Cube^A are represented as J , with the indices corresponding to each shape and its location (Fig. 4.4). Second, f is computed based on the offsets along 0° , 45° , 90° , and 135° directions on J following (4.3). Note that one more pair is extracted along each direction – the pair formed by the last and first element (circulant pair) in the direction. For example, the second row in J is [3 5 4]. The offsets along 0° direction are (3 5) and (5 4)















Index	1	2	3	4	5	6	7
(Geometry, Color, Orientation) IDs	(0,0,0)	(0,1,0)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)	(2,0,0)
Shape							
Index	8	9	10	11	12	13	14
(Geometry, Color, Orientation) IDs	(2,0,1)	(2,0,2)	(2,0,3)	(2,1,0)	(2,1,1)	(2,1,2)	(2,1,3)
Shape							

Figure 4.3: Assigned indices and IDs to the shapes for computing GLCM.

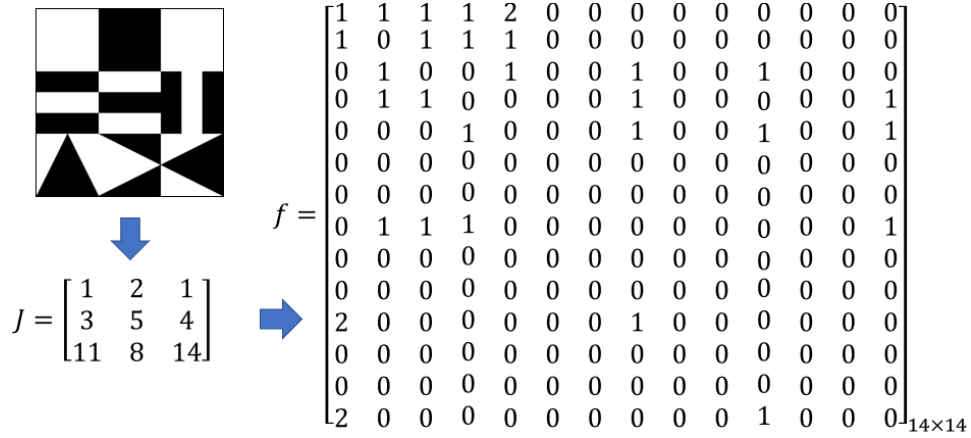


Figure 4.4: GLCM computation of an e-Cube^A item.

while the offsets with the circulant pair along 0° are (3 5), (5 4), and (4 3). By doing so, the configurational disorder is inclusively captured. Lastly, the weighted entropy based on f is calculated following by

$$D_{config} = H_{config} = - \sum_{i=1}^{14} \sum_{j=1}^{14} w_{i,j} p_{i,j} \log_2(p_{i,j}) \quad (4.7)$$

where

$$p_{i,j} = \frac{f(i,j)}{\sum_{i=1}^{14} \sum_{j=1}^{14} f(i,j)} \quad (4.8)$$

Note that $p_{i,j} = 0$ if $f_{i,j} = 0$. The qualitative weight ($w_{i,j}$) associates the subjective factor with the

objective probability [105]. The weight and probability can be independent. The weight estimates the visual complexity of two adjacent elements based on their colors and orientations. To compare the differences among these fourteen distinctive shapes, three IDs are assigned to each shape to categorize its geometry (square, strip, or triangle), color, and orientation, as shown in Fig. 4.3. Referring to the IDs, the criteria for the weight selection is displayed in Table 4.1. Eight weight examples are shown in Fig. 4.5 based on this selection criteria.

Table 4.1: Weight selection for computing D_{config} .

The Fact of the Two Adjacent Rotated Shapes		Weight ($w_{i,j}$)
with the same shape ID		$w_{i,j} = 1$
with different shapes ID	with the same color ID and orientation ID	$w_{i,j} = 1$
	with different color IDs but same orientation ID or in same color ID but different orientation IDs	$w_{i,j} = 1$
	with different color IDs and different orientation IDs	$w_{i,j} = 2$

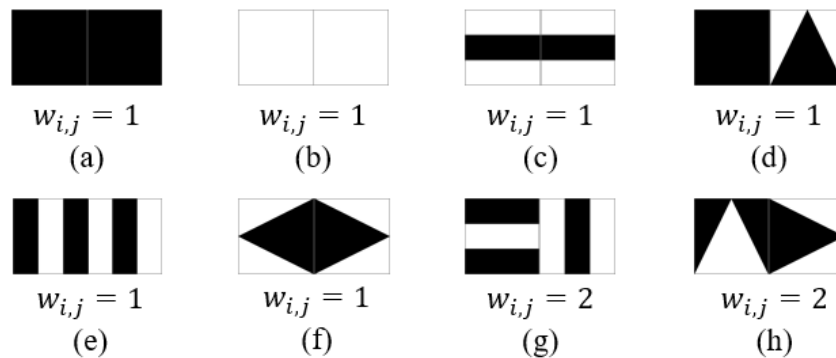


Figure 4.5: Weight examples following the selection criteria shown in Table 4.1.

Based on (4.5), the total complexity (i.e. D_t) with the D_{compos} and D_{config} of the examples shown in Fig. 4.2 are calculated and displayed in Fig. 4.6. Fig. 4.7 and Fig. 4.8 display the play

complexity values of some 2×2 items and 3×3 items, respectively. The complexity computed by the previous approach for TAG-Games [1] is also attached in the last column for comparison. Since the new play complexity measure include both compositional and configurational randomness, it constructs a better frame for the visual complexity estimation.


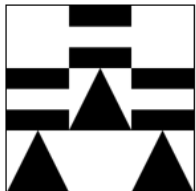

No.	Items	D_{compos}	D_{config}	D_t
a		32.265	2.838	34.652
b		32.265	2.972	34.867
c		32.265	5.281	37.534

Figure 4.6: Play complexity of the e-Cube^A items shown in Fig. 4.2.

4.1.2 Play complexity measure for e-Cube^S

The computation of D_t for e-Cube^S still follows (4.5). However, each shape contributes the same in computing D_{compos} since e-Cube^S asks the examinee to focus on the visual design of the patterns other than assembly. In this case, D_{compos} estimates the complexity caused by the number of the shapes used in one item, the equation of which is shown in (4.9), where n_s is the number of the shapes in an item.











No.	Items	D_{compos}	D_{config}	D_t	Complexity Computed by [1]
1		10.340	0.000	5.170	10.33
2		10.340	1.918	11.707	12.92
3		12.340	1.918	13.601	14.92
4		12.340	3.349	15.608	15.92
5		12.340	3.585	15.868	16.92
6		14.340	1.918	15.494	16.92
7		14.340	4.780	19.109	18.92
8		18.340	2.975	21.106	20.92
9		18.340	3.585	21.840	22.92
10		18.340	5.975	24.312	22.92

Figure 4.7: Play complexity of the e-Cube^A item 1-10 shown in Fig. 3.1.

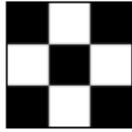


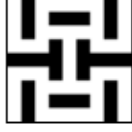




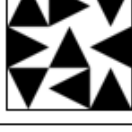

No.	Items	D_{compos}	D_{config}	D_t	Complexity Computed by [1]
11		23.265	1.954	24.042	30.24
12		26.265	1.879	26.665	32.65
13		32.265	1.879	32.327	38.65
14		32.265	3.227	35.239	44.15
15		32.265	3.766	35.918	38.65
16		32.265	4.063	36.255	45.15
17		34.265	4.313	38.524	48.73
18		38.265	3.617	41.714	50.56
19		41.265	3.742	44.856	54.15
20		41.265	7.000	48.264	58.73

Figure 4.8: Play complexity of the e-Cube^A item 11-20 shown in Fig. 3.1.

$$D_{compos} = -n_s \sum_{i=1}^{14} \frac{1}{14} \log_2\left(\frac{1}{14}\right) \approx 3.807 \cdot N \quad (4.9)$$

D_{config} is the summation of $H_{config1}$, $H_{config2}$, and $H_{config3}$, as shown in (4.10), which are the weighted entropy calculated based on f_1 (offsets along 0° and 180° with circulant pairs), f_2 (offsets along 90° and 270° with circulant pairs), and f_3 (offsets along 45° , 135° , 225° , and 315° with circulant pairs).

$$D_{config} = H_{config1} + H_{config2} + H_{config3} \quad (4.10)$$

Note that if an item is in 2×2 , D_{config} is the summation of $H_{config1}$ and $H_{config2}$, the weighted entropies computed based on f_1 (offset along 0°) and f_2 (offset along 90°). Fig. 4.9 shows the computed play complexity of the items shown in Fig. 3.2.

4.1.3 Play complexity measure for e-Cube^{M1} and e-Cube^{M2}

The measure of play complexity for e-Cube^{M1} is the same as the one for e-Cube^A, except the offset directions of f matrix are only along 0° and 180° for sequential items. Fig. 4.10 displays the play complexity of the e-Cube^{M1} items shown in Fig. 3.3.

The items of e-Cube^{M2} are similar with the ones of e-Cube^A and e-Cube^{M1} - both e-Cube^{M2} and e-Cube^A (or e-Cube^{M1}) use 2×2 (or 1×2 , 1×3) patterns. Therefore, if an e-Cube^{M2} item is in 2×2 (or 1×2 , 1×3) dimension, the system uses the play complexity measure for e-Cube^A (or e-Cube^{M1}). Fig. 4.11 displays the computed play complexity of e-Cube^{M2} items shown in Fig. 3.4.

4.1.4 Play complexity measure for e-Cube^P

The play complexity of an e-Cube^P item is equivalent to its network complexity. Shannon entropy has been proved to be reliable and reasonable in the network complexity computation [106]. Suppose the number of the vertices in a path graph is V and a certain degree is a_i , the path complexity is measured by

$$D_t = \sum_{i=1}^V a_i \log_2(a_i) \quad (4.11)$$

Fig. 4.12 displays the play complexity of the e-Cube^P items shown in Fig. 3.5.






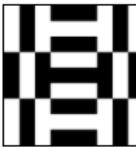
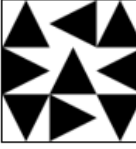



No.	Items	D_{compos}	D_{config}	D_t
1		15.230	0.000	7.615
2		15.230	3.000	18.062
3		15.230	6.000	21.228
4		15.230	3.000	18.062
5		15.230	6.000	21.228
6		34.266	7.755	42.021
7		34.266	8.127	42.393
8		34.266	9.576	43.842
9		34.266	10.337	44.603
10		34.266	16.297	50.564

Figure 4.9: Play complexity of the e-Cube^S items shown in Fig. 3.2.









































































No.	Items			No.	Items		
	D_{compos}	D_{config}	D_t		D_{compos}	D_{config}	D_t
1	2.585	0.000	1.292	2	4.585	0.000	2.292
							
3	5.170	1.000	5.227	4	6.170	1.000	6.044
	 				 		
5	7.755	1.500	8.515	6	9.755	2.000	11.292
	  				  		
7	12.340	1.585	12.877	8	14.340	1.585	14.707
	   				   		
9	14.925	2.250	16.681	10	16.925	2.250	18.615
	    				    		
11	19.510	2.171	20.957	12	19.510	2.922	22.191
	     				     		
13	22.095	3.085	24.966	14	23.095	3.849	26.872
	      				      		
15	25.680	3.093	28.527	16	27.680	3.236	30.701
	       				       		

Figure 4.10: Play complexity of the e-Cube^{M1} items shown in Fig. 3.3.









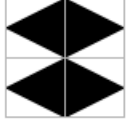

No.	Items	D_{compos}	D_{config}	D_t
1		5.170	1.000	5.227
2		6.170	1.000	6.044
3		9.170	1.000	8.497
4		9.755	1.000	8.975
5		11.755	1.500	12.134
6		12.755	2.000	14.150
7		14.340	1.918	15.494
8		16.340	2.752	18.832
9		18.340	1.918	19.281
10		18.340	5.975	24.312

Figure 4.11: Play complexity of the e-Cube^{M2} items shown in Fig. 3.4.

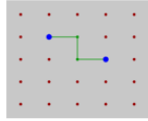
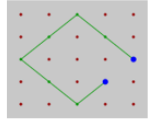
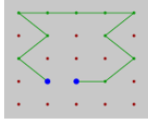
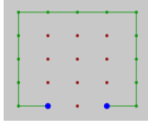
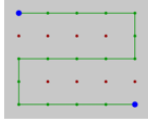
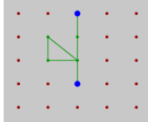
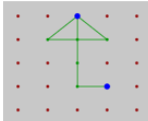
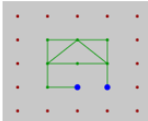
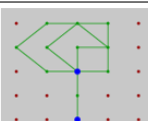
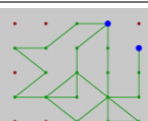
No.	Items	D_t	No.	Items	D_t
1		4.000	2		12.000
3		20.000	4		26.000
5		30.000	6		14.000
7		20.755	8		32.000
9		43.610	10		64.265

Figure 4.12: Play complexity of the e-Cube^P items shown in Fig. 3.5.

4.1.5 Play complexity measure for e-Cube^Z

Although one measure of the maze complexity has been presented [107], which performed well in a maze with both entrance and exit at its edges, it hardly estimates the complexity of a maze that has an exit within it. Hence, (4.11) is adopted to estimate the maze pathway complexity (D_m) by considering the grids and the aisles formed by the grids in the e-Cube^Z items as nodes and paths in the e-Cube^P items (Fig. 4.13). By doing so, D_m is the total degree of the aisles and reflects the complexity of a maze itself regardless of the position of the entrance and exit.

However, the complexity of solving a maze would be different due to the different positions of entrance and exit. Therefore, solution logarithmic complexity (D_s) is defined in (4.12) representing the complexity caused by the vertex degrees, where L is the total length of the shortest path solved

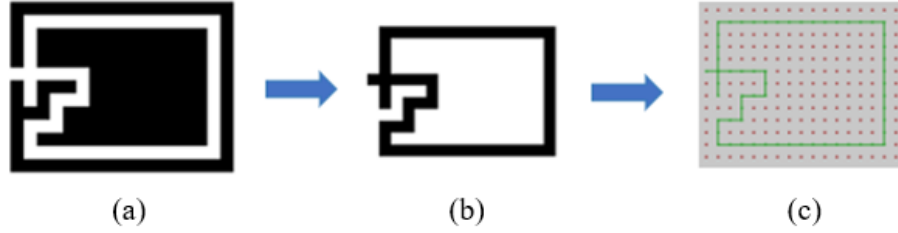


Figure 4.13: Maze pathway conversion: (a) original maze; (b) pathway; and (c) pathway represented in 13×17 dots form, similar to e-Cube^P items.

by A* and s_i is the degree of each grid in this solution.

$$D_s = \sum_{i=1}^L s_i \log_2(s_i) \quad (4.12)$$

Solution length complexity (D_l) computed based on (4.13) measures the complexity caused by the shortest path length.

$$D_l = \log_2(L) \quad (4.13)$$

(4.14) computes the total complexity (D_t) based on (4.12) and (4.13), where 0.4 makes this complexity value comparable with the other e-Cube games. In order to balance the magnitudes with D_m , addition of D_s and D_l is multiplied by 10.

$$D_t = 0.4 \times [D_m + 10 \times (D_s + D_l)] \quad (4.14)$$

Following this method, the play complexity of e-Cube^Z items shown in Fig. 3.6a is computed and displayed in Fig. 4.14, where the complexity computed by the published method [107] is also included. Using this previous approach, the complexity values of the first two items in Fig. 4.14 are 249.195 and 38.526, but their visual complexities are similar, which reveals the inefficiency of the approach in [107]. On the contrary, the newly defined approach shows a more consistent record of measuring the maze complexity. This comparison demonstrates the proposed method performs better in measuring the complexity of a maze with an exit within it.







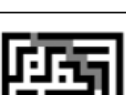



No.	Items	D_m	D_s	D_t	$D_t=0.4[D_m + 10(D_s + D_t)]$	Maze Complexity Computed by [107]
1		58.000	1.585	3.170	42.220	249.195
2		62.340	2.000	3.700	47.738	38.526
3		82.510	1.000	4.322	54.292	114.156
4		63.510	3.000	4.170	54.084	31.145
5		84.925	3.000	4.907	65.598	177.582
6		100.020	4.000	3.907	71.635	158.175
7		104.605	5.000	4.087	78.192	108.824
8		99.435	6.585	4.459	83.951	105.992
9		104.020	7.585	4.460	89.785	87.847
10		104.020	8.585	4.907	95.575	88.610

Figure 4.14: Play complexity of the e-Cube^Z items shown in Fig. 3.6a.

4.2 Adaptive Game Generators

The development of the play complexity measures is a critical step towards building an adaptive game generator, which includes a defined game procedure and six item generators (for six games).

Each item generator can automatically generate an item given a play complexity. Indeed, this item generation is an inverse procedure of the play complexity measuring (Fig. 4.15). The play complexity changes with the examinee’s correctness based on the game procedure. Therefore, the game generator makes adaptive testing achievable in e-Cube and this version is named *adaptive* e-Cube. Adaptive e-Cube can provide test items within a precise cognitive ability range for each examinee.

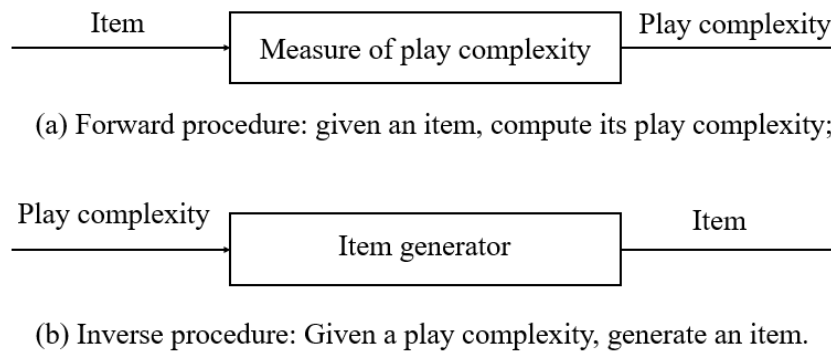


Figure 4.15: Flowchart of the (a) measures of play complexity and (b) item generators.

4.2.1 Game procedure

The game begins at an item with a pre-defined low complexity. If the examinee answers the first item correctly, it proceeds to the next item with a higher complexity; otherwise, a new item at the same complexity is generated. If two consecutive incorrect answers are provided, the complexity reverses to the midpoint of the latest, correctly answered item complexity and the current incorrectly answered one. The difference between the current and the next complexity values - referred to as a step size - can be either a positive, zero, or a negative value. The game ends either at a pre-defined highest complexity level or when the step size becomes smaller than a predefined threshold. The following subsections describe how the items are autonomously generated by item generators.

4.2.2 Item generators for e-Cube^A, e-Cube^{M1}, and e-Cube^{M2}

The flowchart of the item generator for e-Cube^A is displayed in Fig. 4.16. The generator takes inputs of a desired play complexity value D , the item size (i.e. 2×2 or 3×3), and a small tolerance value ϵ and generates a new item with a play complexity d , such that $|D - d| \leq \epsilon$. Since the complexity value is within $[5.170, 24.312]$ for a 2×2 e-Cube^A item and $[11.632, 48.900]$ for a 3×3 e-Cube^A item, the algorithm terminates if D is not in the range. It then activates a loop that begins with randomly picking 4 or 9 shapes based on the size of the item. The proposed measure of play complexity (4.5) calculates d of the current item. Then the system checks whether d satisfies the terminating condition, $|D - d| \leq \epsilon$, where ϵ is set as 0.3. If this condition is satisfied, one new item is generated; otherwise, d can be adjusted by changing one shape. If $D > d$ ($D < d$) stating the current generated item needs to be more complicated (simpler), the algorithm changes square to strip or strip to triangle (triangle to strip or strip to square) with an arbitrary orientation if the current item has any. If not, the generator adjusts the orientation of one shape to be different from (the same as) its neighboring shape. Afterward, the generator computes the play complexity of the updated item to determine the continuation of the loop. Note that time is counted during the loop and ϵ will be enlarged if the generation time is too long (e.g. longer than 3 seconds).

Compared to that for e-Cube^A, slight differences exist in the item generator for e-Cube^{M1} due to its sequential format (Fig. 4.17). Instead of using a 2D size as one of the inputs, length has been adopted representing the number of the desired images in a sequence. Moreover, the default value of ϵ is 0.1. The rest of the algorithm is kept the same.

The item generator for e-Cube^{M2} inherits the property of both e-Cube^A and e-Cube^{M1} – use e-Cube^A generator if the item size is 2D and use e-Cube^{M1} generator if the item is in a sequence.

4.2.3 Item generator for e-Cube^S

Ten types of visual designs are used to generate e-Cube^S items. The designs specify the shape rotation, color change, or symmetry of each item instead of fixing the pattern. The game continues at the first five items regardless of the examinee's correctness. However, if the examinee fails after

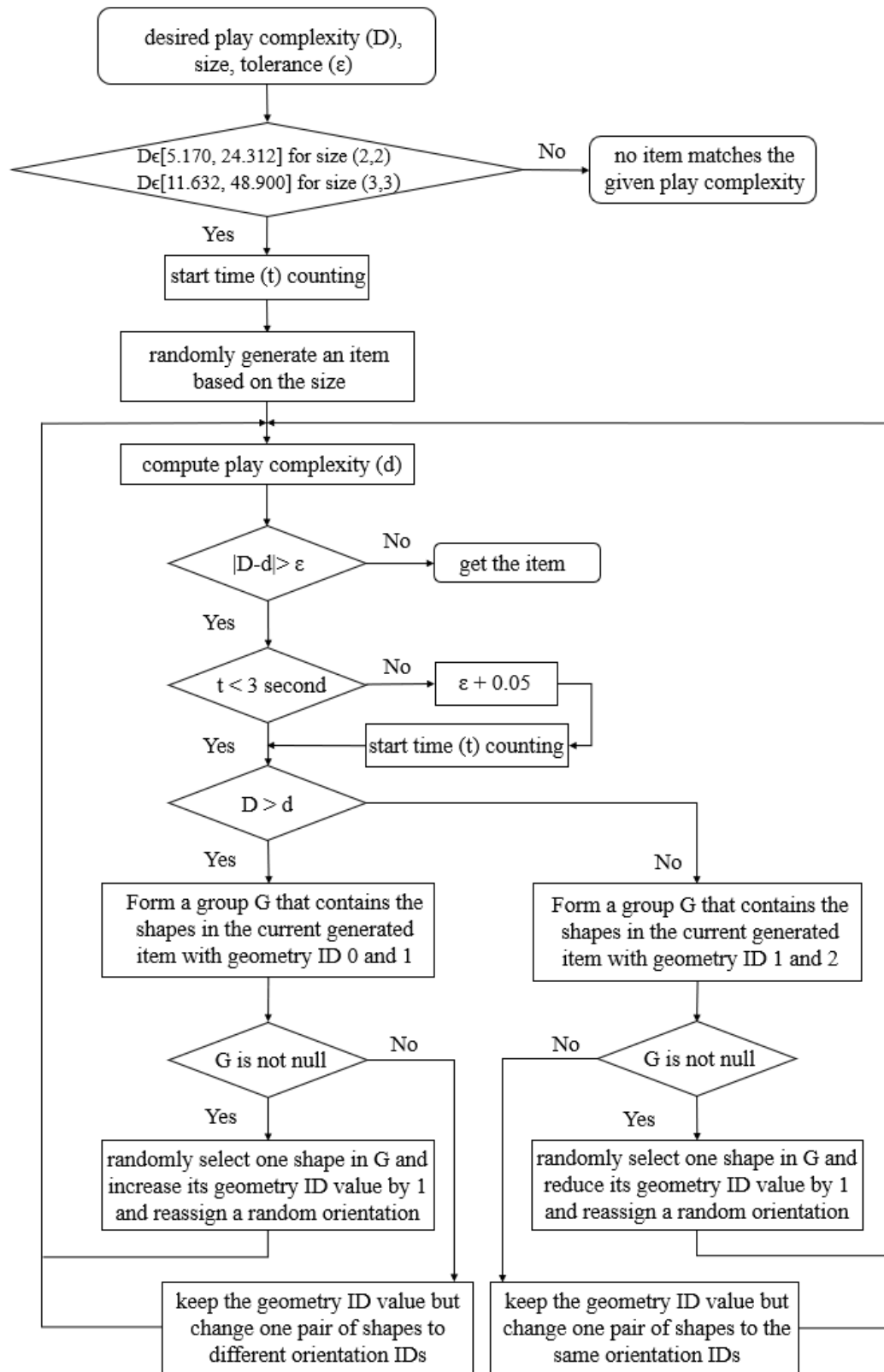


Figure 4.16: Flowchart of the item generator for e-Cube^A.

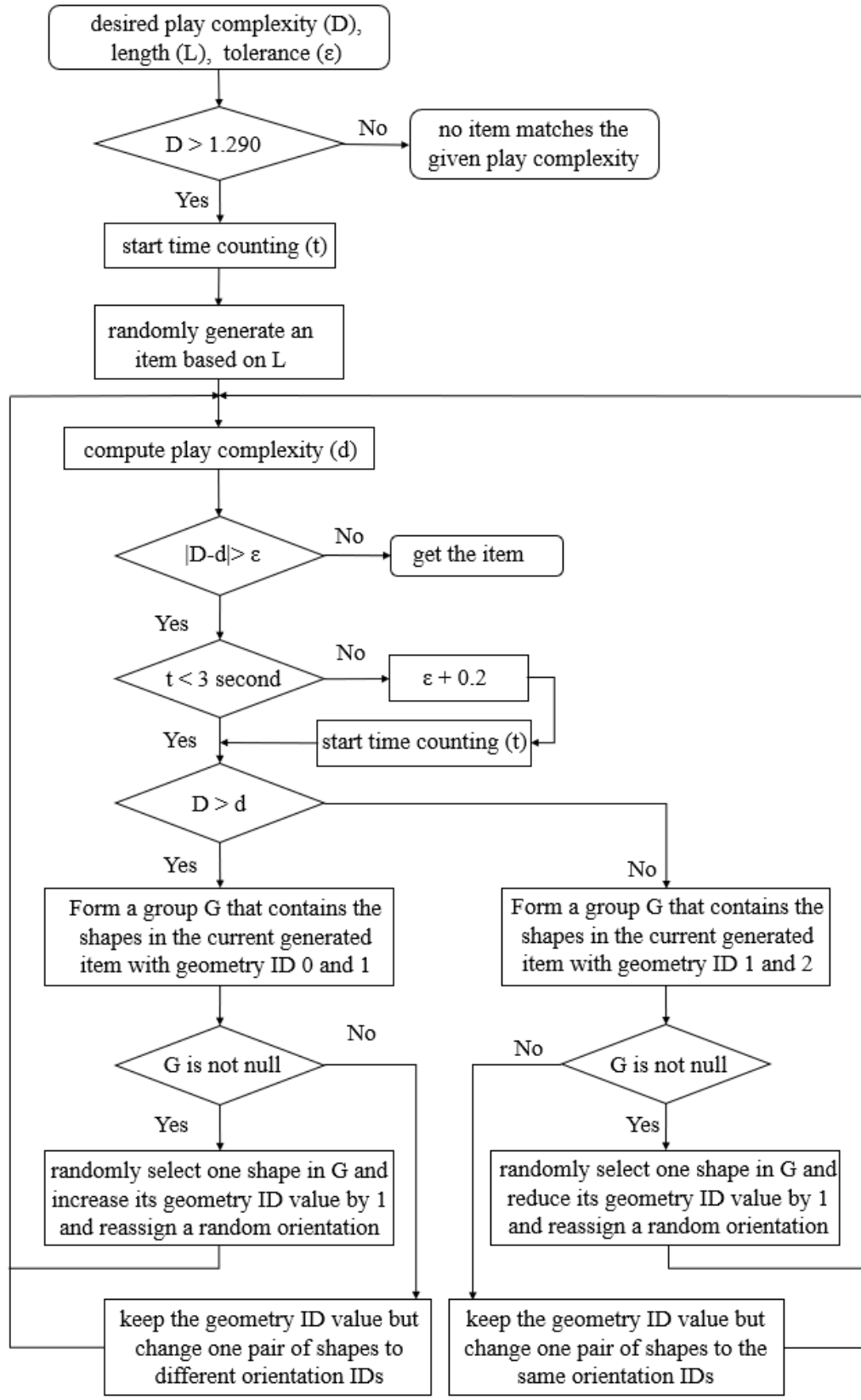


Figure 4.17: Flowchart of the item generator for e-Cube^{M1}.

the first five items, another item with the same relationship will be generated. The game ends after receiving two consecutive wrong answers. Using the position order of a four- or nine-cube e-Cube^S item shown in Fig. 4.18 to describe, the ten designs include:

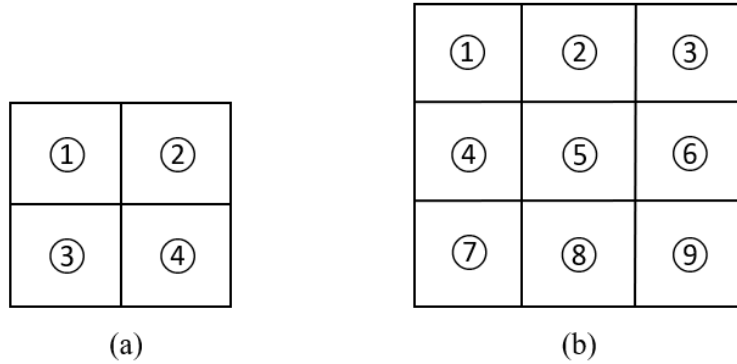


Figure 4.18: Position order of a (a) four-cube or (b) nine-cube e-Cube^S item.

Design 1: four shapes are identical (e.g. item 1 in Fig. 3.2b);

Design 2: four triangle shapes are symmetric about the horizontal and vertical axes (e.g. item 2 in Fig. 3.2b);

Design 3: four (or nine) shapes formed by rotating a triangle shape 90° clockwise/counter-clockwise following the order of ①, ②, ④, and ③ ((①, ②, ③, ⑥, ⑤, ④, ⑦, ⑧, and ⑨)) (e.g. item 3 in Fig. 3.2b);

Design 4: four strips are diagonally symmetric (e.g. item 4 in Fig. 3.2b);

Design 5: two rows/columns formed by triangles and strips are with the opposite colors (e.g. item 5 in Fig. 3.2b);

Design 6: for a nine-cube item, shapes are the same in a principle diagonal and two offset diagonals, which include a square, strip, and triangle shape (e.g. Fig. 4.19a);

Design 7: for a nine-cube item, shapes are rotated 90° clockwise/counter-clockwise along rows/columns that include a square, strip, and triangle shape (e.g. Fig. 4.19b);

Design 8: nine shapes formed by rotating a triangle shape 90° clockwise/counter-clockwise following the order of (1), (2), (3), (6), (5), (4), (7), (8), and (9) (e.g. Fig. 4.19c);

Design 9: nine shapes formed by rotating a triangle shape 90° clockwise/counter-clockwise following the order of (1), (3), (5), (7), and (9) and another triangle shape with an opposite color 90° clockwise/counter-clockwise following the order of (2), (4), (6), and (8) (e.g. item 10 in Fig. 3.2);

Design 10: nine shapes formed by rotating a triangle shape 90° clockwise/counter-clockwise while changing it to the opposite color following the order of (1), (2), (3), (6), (5), (4), (7), (8), and (9) (e.g. Fig. 4.19d);

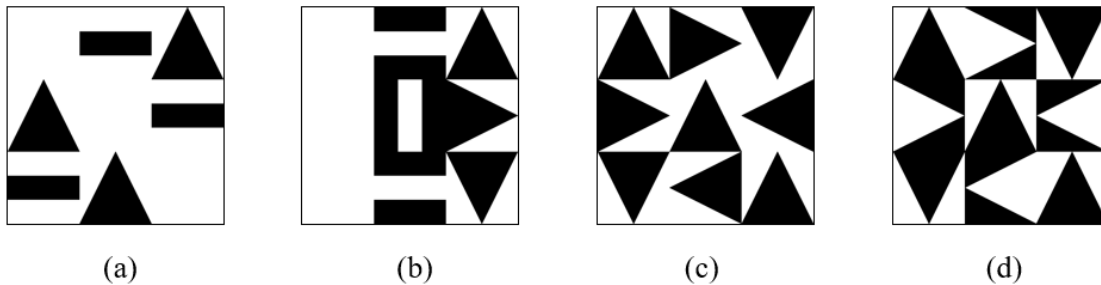


Figure 4.19: Examples of Design 6, 7, 8, and 10 of e-Cube^S.

4.2.4 Item generator for e-Cube^P

The item generator (Fig. 4.20) for e-Cube^P generates a path given a desired play complexity D , the number of connected nodes O , and a tolerance value ϵ . First, it randomly generates a path by selecting one random dot then collecting it with its unconnected neighbor until O nodes are obtained. Then, the system computes its play complexity as d using (4.11). If $|D - d| \leq \epsilon$, the item generator terminates and the output is the current path; if not, the algorithm gets into a loop,

in which the path should be adjusted until the values of d and D are close enough. How to adjust the path? Known that the more intersections the node connections have, the more difficult the path is, based on the play complexity defined in (4.11). Since it is hard to adjust the number of intersections given the number of nodes, a hypothesis is made - the more closed the connected nodes are (measured by distance), the more intersections they might have. Therefore, the item generator for e-Cube^P adjusts the item to be more difficult (or easier) by decreasing (or increasing) the total Euclidean distance among nodes. For example, if $D > d$, the current generated path with the play complexity d should be updated to a harder one so the algorithm forms a pool containing the unconnected neighbor that has the smallest distance for each node. Then it randomly selects one node with its neighbor having the smallest total distance and substitutes the subsequent node with this neighbor. The path following the neighbor is discarded and a random path based on the left node number is regenerated. The algorithm then computes the play complexity of the updated path and continues the loop until $|D - d| \leq \epsilon$ is satisfied.

4.2.5 Item generator for e-Cube^Z

The randomized Prim's algorithm [98] constructs mazes by creating walls and aisles. The number of the walls and aisles is determined by two parameters, the maze complexity c and its density e . However, the maze generated by Prim's does not have a specific entrance or exit while their locations affect the actual difficulty of the maze (4.14). To address this and develop an item generator for e-Cube^Z (Fig. 4.21), an approximated relationship among the desired play complexity D , c , and e are experimentally inferred as $c = \frac{D}{1000}$ and $e = \frac{D}{800}$. The terminating condition is $|D - d| \leq \epsilon$ where ϵ is a tolerance value. The system generates a maze using Prim's with the parameters c and e , then breaks one arbitrary wall at one of the borders and takes it as the entrance. Afterward, it itemizes all the aisles as "exit candidates" and computes their play complexities as $d_1, d_2, \dots, d_i, \dots, d_n$ where n is the number of the exit candidates. If one of the absolute difference values between D and d_i is smaller than ϵ , the algorithm terminates and the output is the current maze with the selected entrance and the i th exit; otherwise, the algorithm generates another maze and continues this procedure.

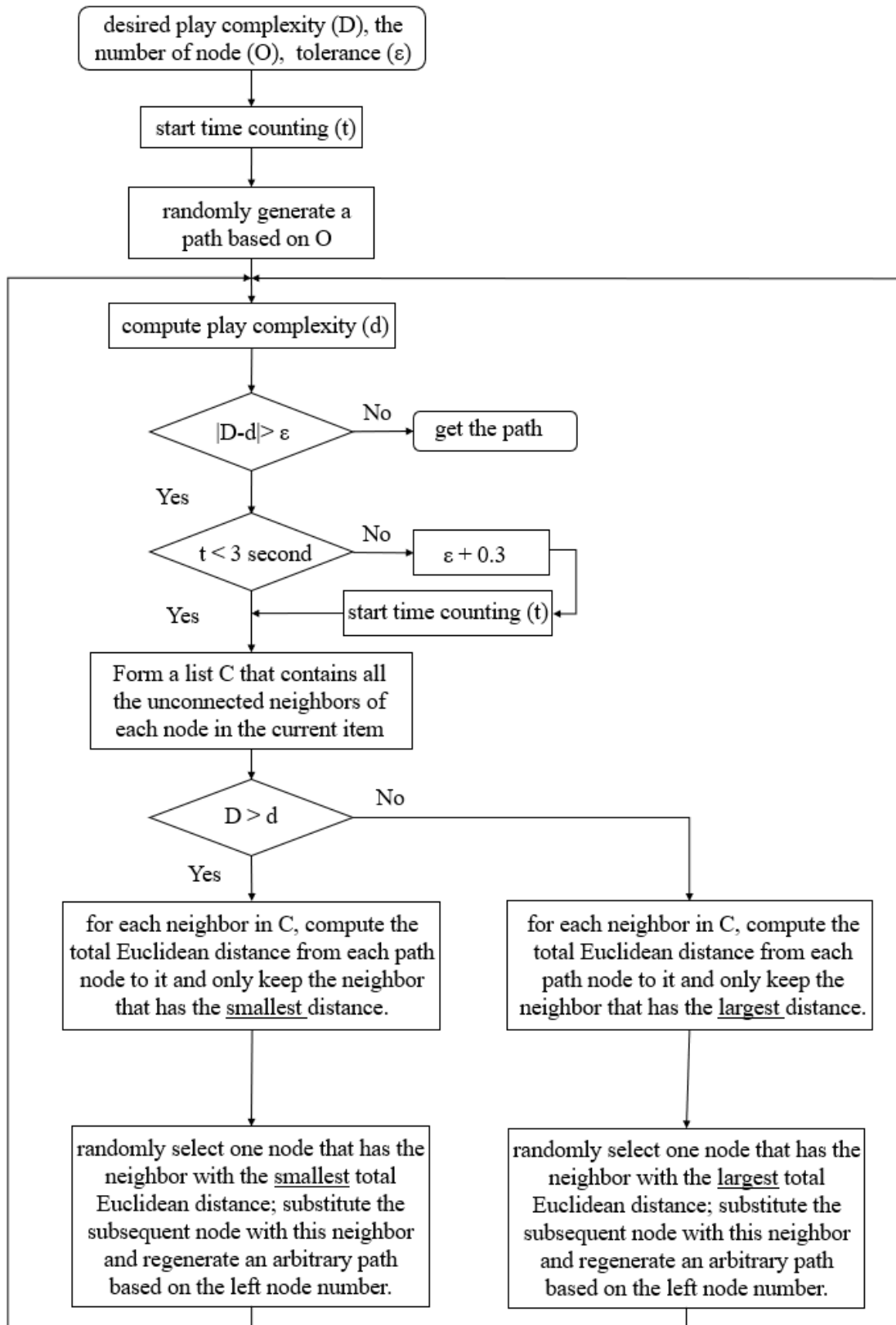


Figure 4.20: Flowchart of the item generator for e-Cube^P.

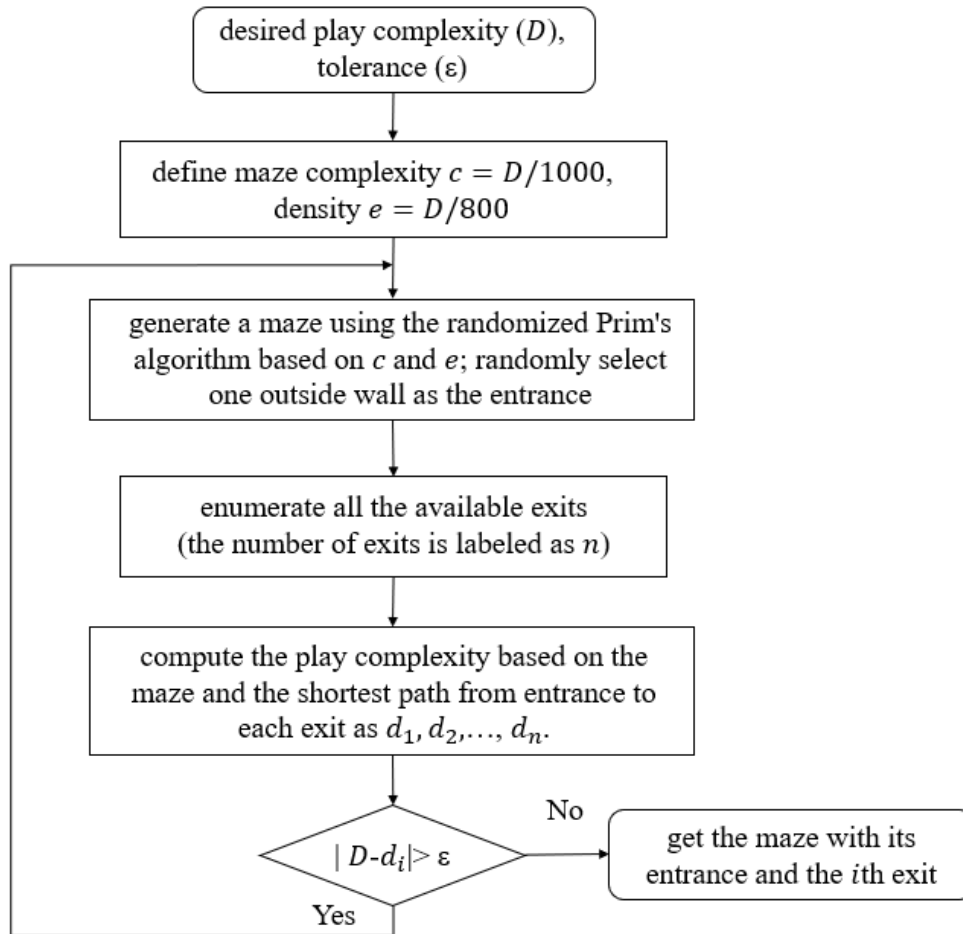


Figure 4.21: Flowchart of the item generator for e-Cube^Z.

5. HUMAN SUBJECT EVALUATION

This chapter presents the methods, protocol, scoring system, and results of the human subject evaluation on e-Cube system aiming to test the **three hypotheses**:

- H1.** Performance indicators – scores and time on individual items – are correlated with the play complexity values of the items in each e-Cube game.
- H2.** The scores from the adaptive e-Cube games are correlated with that from the existing standardized methods.
- H3.** The e-Cube technology is reliable and usable.

5.1 Methods

To test the three hypotheses, the study employed two versions of e-Cube games: e-Cube with fixed items (called *fixed* e-Cube) and e-Cube with adaptive items using the developed item generators (called *adaptive* e-Cube). Each participant is assigned to one of the two groups to experience either the fixed or adaptive e-Cube games (i.e., Fixed Group or Adaptive Group). The fixed e-Cube games provide the same items for each examinee, while the adaptive ones offer different items as well as different numbers of items based on users' real-time performance. The items used in the fixed version of the e-Cube games and their play complexities were previously shown in Fig. 3.1-3.2, 3.3-3.5, and 3.6, where fixed e-Cube^A and e-Cube^S share the same items as in [85].

Hypothesis 1 aims to evaluate if the proposed computational measure of play complexity (i.e. D_t) captures relative difficulties of the items in each e-Cube game. If it is defined properly, participants' performance measured by scores/time would decrease/increase as D_t increases in the items. To evaluate this hypothesis, correlation analyses were performed between the D_t values and the performance indicators of individual items obtained by the participants in Fixed Group. Since these participants experienced the same set of items for each game, the results allow proper comparisons across the items.

Hypothesis 2 evaluates the potential utility of the adaptive e-Cube games for cognitive assessment. Three subtests of WAIS-IV – BD, DS, and MR – were adopted for testing this hypothesis. They are well-established, standardized, and validated to measure fine motor skills, visual-motor integration, perceptual reasoning, and working memory and therefore expected to allow preliminary validation of the e-Cube games. Specifically, e-Cube^A is expected to be related to the WAIS-BD subtest because both require the assembly of blocks to match the top surface with a given pattern. e-Cube^S requires the participant to find a shape that completes a pattern thus is expected to tap the similar cognitive abilities as the MR subtest. Since e-Cube^{M1} and e-Cube^{M2} are designed to capture working memory skills, those games are projected to yield a relationship with the DS subtest. The remaining games, e-Cube^P and e-Cube^Z, are timed games and ask participants to give the shortest trajectory by reasoning, it is conjectured that there may be a relationship between these games and BD and MR. The hypothesized relationships between the e-Cube games and three WAIS subtests are summarized in Table 5.1.

Table 5.1: Six e-Cube games and their expected associations with WAIS subtests.

	Block Design	Digit Span	Matrix Reasoning
e-Cube^A	✓		
e-Cube^S			✓
e-Cube^{M1}		✓	
e-Cube^{M2}		✓	
e-Cube^P	✓		✓
e-Cube^Z	✓		✓

The adaptive e-Cube games were also compared with the fixed e-Cube games, in terms of the overall scores, correlations with the WAIS subtests, and inter-correlations within the individual e-Cube games. Developing a standardized test with a fixed set of items requires large-scale human subject evaluations to fine-tune and select the proper items. Without such process, an arbitrarily selected set of items used for assessment is expected to show limitations, while this comparison

aims to determine whether the adaptive feature of the games leads to better assessment outcomes compared to those with fixed items.

Hypothesis 3 evaluates the reliability and usability of the e-Cube technology. The reliability is evaluated by the false detection rate of the vision system (i.e. the top surface identification), defined by the percentage ratio of the incorrect detections to the total number of detections. For the usability evaluation, SUS is administered to all participants upon completion of the assessment session. SUS consists of 10 questions each to be scored from 1 to 5 representing strongly agree to strongly disagree. The total scores of SUS can be converted into a 100 - scale [108]. The 10 questions in SUS are tailored to evaluate the usability of e-Cube as follows:

1. I think that I would like to use the e-Cube games frequently.
2. I found the system unnecessarily complex.
3. I thought the e-Cube games were easy to use.
4. I think that I would need the support of a technical person to be able to use e-Cube games.
5. I found the various functions in the e-Cube games were well integrated.
6. I thought there was too much inconsistency in the e-Cube games.
7. I would imagine that most people would learn to use the e-Cube games very quickly.
8. I found the e-Cube games very cumbersome to use.
9. I felt very confident using the e-Cube games.
10. I needed to learn a lot of things before I could get going with the e-Cube games.

The SUS score (out of 100) is computed for each participant based on the directly obtained points for the questions and the score converting system [108]. The mean value and standard deviation of the SUS scores are then calculated from each group for analysis and comparison.

5.2 Protocol

The human subject study took place at Texas A&M University (TAMU) (IRB2019-1079D, Approval Date: 12/22/2020). Bulk recruitment emails were sent to TAMU communities and flyers were placed in buildings within the university for recruiting healthy participants at ages from 18 to 64 years old. Once potential participants contact the research team, a pre-screening survey is sent via email to self-identify their eligibility before scheduling a visit. The pre-screening survey consists of four questions that ask for 1) age, 2) date of birth, 3) gender, and 4) health conditions. Individuals who are not within the age range or answered yes for any of the following health conditions were excluded: stroke, other neurological diseases, low vision/blindness with aid, hearing loss or deafness with aid, or difficulties in arm/hand movements for manipulating small objects.

Seventy-eight participants (46 males) were recruited and screened. All 78 were eligible and thus enrolled in the study. Informed consent and background information (i.e., age and gender) were obtained from each participant. Most of the participants were randomly assigned to either Fixed Group or Adaptive Group, while some efforts were made to balance the gender and age distribution when the participants were placed into the groups towards the end. Fixed Group involved 37 participants (22 males) and Adaptive Group involved 41 participants (24 males). Due to the convenience of recruitment and proximity to the study location, most of the participants were students from various departments and programs across the TAMU campus while several faculty and staff members, alumni of the university, and a few residents also participated. Sixty-six out of 78 participants were aged between 18 and 30 and were approaching their college degrees. The gender distribution, age mean, and age standard deviation (SD) are summarized in Table 5.2.

The order between the two sets of the tests – WAIS-IV subtests and e-Cube games was randomized. The administration order of the three subtests of WAIS-IV followed the standardized protocol (BD, DS, and MR), while the order of the six e-Cube games was randomized. Upon completion of both tests, SUS was administered to each participant. The entire session took about 90 minutes, 50 minutes for e-Cube, 25 minutes for WAIS-IV subtests, 5 minutes for SUS, and a 10-minute break between e-Cube and WAIS-IV subtests. Upon completion of the entire session, a \$10 gift card was

provided to each study participant as a token of our appreciation for their time and participation.

Table 5.2: Gender distribution, age mean, and age SD of Fixed Group, Adaptive Group, and all participants.

	Fixed Group	Adaptive Group	Total
Male n (age mean \pm SD)	22 (26.00 \pm 8.50)	24 (26.46 \pm 9.43)	46 (26.24 \pm 8.80)
Female n (age mean \pm SD)	15 (25.93 \pm 8.44)	17 (23.38 \pm 3.40)	32 (24.47 \pm 6.33)
Total n (age mean \pm SD)	37 (25.97 \pm 8.36)	41 (25.10 \pm 7.51)	78 (25.51 \pm 7.88)

5.3 Scoring System

Since e-Cube games are not standardized, optimal scoring methods have not yet been developed. At this evaluation stage, the scoring methods employed for the WAIS-IV subtests and the previous study [85] were benchmarked and modified to be suitable for the e-Cube games.

The scoring methods for the fixed e-Cube games are as follows. Scoring for e-Cube^A (Table 5.3) is similar to the methods used in BD and the assembly game in TAG-Games, which considers correctness, item size, and completion time [85]. A correct answer for a 2×2 item that is completed within 20 seconds yields 3 points and between 20-60 seconds yields 2 points. A correct answer for a 3×3 item completed within 30 seconds, between 30-80 seconds, or between 80-120 seconds results in 4, 3, or 2 points, respectively. An incorrect answer or exceeding the time limit (60 seconds for a 2×2 item and 120 seconds for a 3×3 item) results in 0 points. e-Cube^S, e-Cube^{M1}, and e-Cube^{M2} solely use correctness as the scoring criteria: 2 points for each correct answer and 0 for an incorrect answer (Table 5.4). Scoring methods for e-Cube^P and e-Cube^Z are based on correctness, completion time, and whether the taken path is the shortest one or not. For e-Cube^P, the shortest path finished within 20 seconds, between 20-40 seconds, or between 40-80 seconds yields 4, 2, or 1 point(s); correct path but not the shortest completed within 20 or between 20-40 yields 2 or 1 point(s); incorrect path or exceeding the time limit (80 seconds for the shortest path and 40 seconds for the correct but not the shortest) results in 0 point (Table 5.5). For e-Cube^Z,

the shortest path completed within 10 seconds, between 10-20 seconds, or between 20-40 seconds yields 4, 2, or 1 point(s); correct path but not the shortest completed within 10 seconds or between 10-20 yields 2 or 1 point(s); incorrect path or exceeding the time limit (40 seconds for the shortest path and 20 seconds for the correct but not the shortest) results in 0 points (Table 5.6).

Table 5.3: Scoring system for fixed e-Cube^A.

Items	Up to (seconds) or correctness	Score
2 × 2 item	20	3
	60	2
	> 60 or wrong	0
3 × 3 item	30	4
	80	3
	120	2
	> 120 or wrong	0

Table 5.4: Scoring system for fixed e-Cube^S, e-Cube^{M1}, and e-Cube^{M2}.

Correctness	Score
Correct	2
Wrong	0

Table 5.5: Scoring system for fixed e-Cube^P.

Condition		Up to (seconds)	Score
Correct	Took the shortest path	20	4
		40	2
		80	1
	Not took the shortest path	20	2
		40	1
		Wrong or > 80	

Table 5.6: Scoring system for fixed e-Cube^Z.

Condition		Up to (seconds)	Score
Correct	Took the shortest path	10	4
		20	2
		40	1
	Not took the shortest path	10	2
		20	1
		Wrong or > 40	

Adaptive e-Cube games require some additional considerations for scoring. If an item is generated with the same play complexity as the previous one answered incorrectly, the score for the

correct answer is 1 point less than the score used in the fixed version. Two consecutive incorrect answers result in the system generating an easier item, and in this case, a correct answer for that newly generated item yields only 1 point.

5.4 Results

The mean scores and SD values obtained by participants in each group and total participants for the WAIS-IV subtests and e-Cube games are summarized in Table 5.7. In the WAIS, a raw score is the summation of all the directly obtained points, and a scaled score is converted from the raw score based on the examinee’s age. All enrolled participants completed the entire session without withdrawal.

Table 5.7: Score statistics from the WAIS-IV subtests and e-Cube games.

		Fixed Group	Adaptive Group	Two-tailed T-Test
		(mean \pm SD)	(mean \pm SD)	$\alpha = 0.05$
WAIS-IV Raw Scores	BD	51.32 \pm 11.17	53.56 \pm 8.67	$t(76) = -0.99, p = 0.324$
	DS	28.14 \pm 4.75	30.07 \pm 5.86	$t(76) = -1.59, p = 0.115$
	MR	21.62 \pm 2.44	22.68 \pm 2.48	$t(76) = -1.90, p = 0.061$
WAIS-IV Scaled Scores	BD	12.05 \pm 3.22	12.49 \pm 2.88	$t(76) = -0.63, p = 0.532$
	DS	9.89 \pm 2.46	11.05 \pm 3.39	$t(76) = -1.71, p = 0.092$
	MR	12.43 \pm 2.17	13.54 \pm 2.50	$t(76) = -0.02, p = 0.042$
e-Cube Scores	e-Cube ^A	58.89 \pm 6.87	61.46 \pm 11.18	$t(76) = -1.21, p = 0.231$
	e-Cube ^S	16.54 \pm 1.86	15.12 \pm 2.64	$t(76) = 2.71, p = 0.008$
	e-Cube ^{M1}	20.59 \pm 4.69	17.98 \pm 3.88	$t(76) = 2.70, p = 0.009$
	e-Cube ^{M2}	17.24 \pm 2.18	16.95 \pm 2.93	$t(76) = 0.49, p = 0.662$
	e-Cube ^P	26.84 \pm 5.87	26.29 \pm 7.62	$t(76) = 0.35, p = 0.726$
	e-Cube ^Z	22.51 \pm 6.35	22.24 \pm 5.18	$t(76) = 0.21, p = 0.837$

5.4.1 Hypothesis 1. Significant correlations between the performance indicators and the play complexity values

The correlations between the performance indicators (i.e., score and time) and the play complexity values (i.e. D_t) reflect how well the difficulties of individual items are captured by the computed D_t values. Two performance indicators were used to evaluate the play complexity measures: 1) the mean scores and 2) the mean completion time obtained for each item from the Fixed Group participants.

The correlational analyses were performed between the D_t values and both performance indicators, as shown in Table 5.8. The D_t values were highly positively correlated with the mean completion time in all e-Cube games, indicating that the items with higher D_t yielded a longer time to answer. In addition, high negative correlations between the D_t values and the mean scores were found in all e-Cube games except for e-Cube^{M2}. In these five e-Cube games, the items with higher D_t yielded lower mean scores. In e-Cube^{M2}, no significant correlation between the mean scores and the play complexity values was observed. This was mainly due to items 8 and 9 (Fig. 3.4) whose complexity values were not well captured by the defined complexity measure, which was originally proposed for the assembly tasks. The symmetry of these items made them very easy to memorize. Without these two items, a significant correlation was found, such that $r(8) = -0.67, p = 0.060$. Overall, the high correlations between the play complexity values and the performance indicators show the potential utility of the proposed D_t definitions for the e-Cube games for properly capturing the item difficulties.

5.4.2 Hypothesis 2. Preliminary validity of e-Cube games for cognitive assessment

5.4.2.1 Mean differences in the two groups – Adaptive and Fixed

A two-tailed t-test with equal variance ($\alpha = 0.05$) comparing Fixed Group and Adaptive Group was conducted using the scores of the WAIS subtests (i.e., raw and scaled scores) and e-Cube games to determine whether there exist significant differences in mean scores of the two groups, as summarized in Table 5.7. The t-test showed no significant differences between the means of the

Table 5.8: Correlations between the play complexity values and 1) the mean score and 2) the mean completion time obtained from individual items from the Fixed Group participants.

	e-Cube Game (Degrees of Freedom)	Mean Score r (p)	Mean Time r (p)
The Play Complexity Values (D_t)	e-Cube ^A (20)	−0.82 (< 0.001)	0.86 (< 0.001)
	e-Cube ^S (10)	−0.75 (0.009)	0.95 (< 0.001)
	e-Cube ^{M1} (16)	−0.99 (< 0.001)	0.98 (< 0.001)
	e-Cube ^{M2} (10)	−0.18 (0.622)	0.90 (< 0.001)
	e-Cube ^P (10)	−0.74 (0.011)	0.82 (0.002)
	e-Cube ^Z (10)	−0.81 (0.003)	0.72 (0.015)

raw scores of the three WAIS subtests in the two groups. In the WAIS scaled scores, a significant difference was not found between the means of BD and DS, but there was a difference in the mean scores of MR between the two groups. The t-test for e-Cube^S and e-Cube^{M1} showed significant differences in the mean scores of these two games between the groups, while no significant differences were found in other games. The mean difference analyses between the two groups lay the foundation of the following evaluation for comparing the performance of the fixed and adaptive versions of the e-Cube games.

5.4.2.2 Comparisons between the fixed and adaptive games

Correlation analyses between the e-Cube scores and the scores from the WAIS-IV subtests were performed to understand the relationships between the two in each group. Since the correlation analyses using WAIS raw scores and scaled scores are similar and consistent, only the results using the scaled scores are presented here. The results using the raw scores can also be found in Appendix C.

Tables 5.9 and 5.10 summarize the correlations between the e-Cube scores and WAIS-IV subtests scaled scores in Fixed Group and Adaptive Group, respectively. Interestingly, the two groups show different trends in the results. The adaptive e-Cube games yielded more significant correla-

Table 5.9: Correlations with degrees of freedom 37 between the scores of fixed e-Cube and scaled scores of the WAIS-IV subtests, shown as r (p).

		WAIS Subtests Scaled Scores		
		BD	DS	MR
Scores	e-Cube ^A	0.50 (0.001)	-0.11 (0.529)	0.06 (0.704)
	e-Cube ^S	0.34 (0.041)	-0.32 (0.057)	0.09 (0.830)
	e-Cube ^{M1}	0.30 (0.067)	0.11 (0.516)	0.19 (0.262)
	e-Cube ^{M2}	0.21 (0.223)	0.27 (0.109)	0.11 (0.499)
	e-Cube ^P	0.48 (0.002)	-0.01 (0.976)	0.12 (0.496)
	e-Cube ^Z	0.27 (0.109)	-0.00 (0.980)	0.16 (0.358)

Table 5.10: Correlations with degrees of freedom 41 between the scores of adaptive e-Cube and scaled scores of the WAIS-IV subtests, shown as r (p).

		WAIS Subtests Scaled Scores		
		BD	DS	MR
Scores	e-Cube ^A	0.47 (0.001)	0.23 (0.145)	0.26 (0.101)
	e-Cube ^S	0.17 (0.276)	0.21 (0.194)	0.49 (0.001)
	e-Cube ^{M1}	0.38 (0.015)	0.24 (0.131)	0.11 (0.498)
	e-Cube ^{M2}	0.07 (0.675)	0.48 (0.001)	0.11 (0.495)
	e-Cube ^P	0.44 (0.004)	0.01 (0.967)	0.38 (0.014)
	e-Cube ^Z	0.28 (0.076)	0.06 (0.732)	0.06 (0.732)

tions with the three WAIS subtests and were better matched with the predicted outcomes shown in Table 5.1. In both groups, e-Cube^A was correlated with BD confirming the expected relationship between the two. In Fixed Group, e-Cube^S and e-Cube^P were also correlated with BD, while no other significant correlations were found.

On the other hand, Adaptive Group showed higher and more distinctive correlations with the WAIS-IV subtests and most of the correlations met the design exceptions (Table 5.1). For example, besides the correlation between e-Cube^A and BD, e-Cube^S was correlated with MR; e-Cube^{M2} was

correlated with DS; and e-Cube^P was correlated with both BD and MR. However, although e-Cube^{M1} were designed to measure working memory, it was only correlated with BD in Adaptive Group. No correlation was found between e-Cube^Z and any of the WAIS-IV subtests in both groups.

Compared to the adaptive version generating different items, the fixed e-Cube games using the same set of items show some limitations. Since the selected items were not validated and standardized through large-scale studies, some of them showed inconsistent results. For example, in the fixed e-Cube^S, it is found that most of the participants correctly answered items 1-7 and 9 while only 51% and 24% of participants correctly answered items 8 and 10, respectively. Moreover, participants in Fixed Group may provide unintended correct answers without the discontinuation rule applied in the games. For instance, given a fixed number of items sorted in increasing difficulty, one may fail to answer correctly in the early items but can provide correct answers in the later items unintentionally. These factors are believed to significantly affect the correlation results in Fixed Group.

To further evaluate the e-Cube games, inter-correlations among the games were analyzed for the fixed and adaptive versions (Tables 5.11 and 5.12). The adaptive e-Cube showed much fewer inter-correlations compared to the fixed version, indicating that the adaptive games share fewer cognitive domains. In the fixed version, most of the games were somewhat correlated except for e-Cube^S. In the adaptive one, e-Cube^P and e-Cube^Z were correlated with each other and also with e-Cube^A.

5.4.2.3 Preliminary evaluations of age and gender differences

As described previously, most of the study participants were college students between the ages of 18 and 30, and only 3 out of 78 were over 50. Therefore, no significant age correlation was expected and the results confirmed this. The relationships between ages and scores were analyzed by computing the correlation between the age and WAIS raw scores/e-Cube scores for Fixed Group, Adaptive Group, and all the participants, as shown in Table 5.13. The results indicated that the ages were not correlated with the raw scores of the WAIS subtests in both groups. Among the

Table 5.11: Inter-correlations with degrees of freedom 37 within the scores of fixed e-Cube games, shown as r (p).

e-Cube^A	1 (0)					
e-Cube^S	-0.02 (0.900)	1 (0)				
e-Cube^{M1}	0.36 (0.029)	0.12 (0.486)	1 (0)			
e-Cube^{M2}	0.23 (0.173)	-0.04 (0.828)	0.35 (0.035)	1 (0)		
e-Cube^P	0.43 (0.008)	-0.02 (0.910)	0.35 (0.034)	0.07 (0.684)	1 (0)	
e-Cube^Z	0.48 (0.002)	-0.12 (0.481)	0.45 (0.006)	0.21 (0.219)	0.27 (0.109)	1 (0)
	e-Cube^A	e-Cube^S	e-Cube^{M1}	e-Cube^{M2}	e-Cube^P	e-Cube^Z

Table 5.12: Inter-correlations with degrees of freedom 41 within the scores of adaptive e-Cube games, shown as r (p).

e-Cube^A	1 (0)					
e-Cube^S	0.22 (0.173)	1 (0)				
e-Cube^{M1}	0.11 (0.481)	0.13 (0.424)	1 (0)			
e-Cube^{M2}	0.22 (0.174)	0.06 (0.688)	0.23 (0.156)	1 (0)		
e-Cube^P	0.51 (< 0.001)	0.22 (0.163)	0.20 (0.209)	0.11 (0.502)	1 (0)	
e-Cube^Z	0.49 (0.001)	0.04 (0.799)	0.17 (0.292)	0.17 (0.280)	0.65 (< 0.001)	1 (0)
	e-Cube^A	e-Cube^S	e-Cube^{M1}	e-Cube^{M2}	e-Cube^P	e-Cube^Z

e-Cube games, a negative correlation was found in the fixed e-Cube^{M1} while no other correlations related to the e-Cube games were found.

The mean and SD of the scores of the WAIS subtests and e-Cube games were computed for each gender in each group (i.e., Fixed Group and Adaptive Group). A two-sample two-tailed t-test was used to analyze the score differences between the genders. Tables 5.14 and 5.15 show the score statistics of male and female participants in Fixed Group and Adaptive Group, respectively. There was no evidence of gender differences in the current data.

Table 5.13: Correlations between the age and WAIS raw scores/e-Cube scores, shown as $r(p)$.

Game/subtests		Age	
		Fixed Group (degrees of freedom 37)	Adaptive Group (degrees of freedom 41)
Raw scores of WAIS subtests	BD	-0.25 (0.132)	0.03 (0.840)
	DS	0.17 (0.314)	0.21 (0.183)
	MR	-0.29 (0.086)	0.21 (0.183)
Scores of e-Cube	e-Cube ^A	-0.08 (0.643)	-0.21 (0.177)
	e-Cube ^S	-0.08 (0.643)	0.14 (0.387)
	e-Cube ^{M1}	-0.39 (0.017)	0.04 (0.824)
	e-Cube ^{M2}	-0.25 (0.128)	0.20 (0.220)
	e-Cube ^P	0.12 (0.461)	0.10 (0.548)
	e-Cube ^Z	-0.13 (0.451)	0.16 (0.314)

Table 5.14: Score statistics of male and female participants in Fixed Group (male: $n = 22$ and female: $n = 15$).

		Male (mean \pm SD)	Female (mean \pm SD)	Two-tailed T-Test $\alpha = 0.05$
WAIS-IV Raw Scores	BD	51.00 \pm 12.15	51.80 \pm 9.96	$t(35) = -0.21, p = 0.834$
	DS	28.32 \pm 4.78	27.87 \pm 4.85	$t(35) = 0.28, p = 0.781$
	MR	21.59 \pm 2.46	21.67 \pm 2.48	$t(35) = -0.09, p = 0.928$
WAIS-IV Scaled Scores	BD	12.09 \pm 3.31	12.00 \pm 3.21	$t(35) = 0.08, p = 0.934$
	DS	9.95 \pm 2.46	9.80 \pm 2.54	$t(35) = 0.19, p = 0.854$
	MR	12.45 \pm 2.24	12.40 \pm 2.13	$t(35) = 0.07, p = 0.941$
e-Cube Scores	e-Cube ^A	58.64 \pm 5.71	59.27 \pm 8.50	$t(35) = -0.27, p = 0.788$
	e-Cube ^S	16.36 \pm 2.10	16.80 \pm 1.47	$t(35) = -0.69, p = 0.492$
	e-Cube ^{M1}	20.55 \pm 4.98	20.67 \pm 4.39	$t(35) = -0.08, p = 0.940$
	e-Cube ^{M2}	16.82 \pm 2.44	17.87 \pm 1.60	$t(35) = -1.46, p = 0.153$
	e-Cube ^P	27.09 \pm 5.05	26.47 \pm 7.07	$t(35) = 0.31, p = 0.756$
	e-Cube ^Z	23.09 \pm 6.38	21.67 \pm 6.44	$t(35) = 0.66, p = 0.511$

Table 5.15: Score statistics of male and female participants in Adaptive Group (male: $n = 24$ and female: $n = 17$).

		Male	Female	Two-tailed T-Test
		(mean \pm SD)	(mean \pm SD)	$\alpha = 0.05$
WAIS-IV Raw Scores	BD	54.92 \pm 9.59	51.65 \pm 7.03	$t(39) = 1.20, p = 0.239$
	DS	31.08 \pm 6.22	28.65 \pm 5.15	$t(39) = 1.32, p = 0.193$
	MR	23.25 \pm 2.01	21.88 \pm 2.91	$t(39) = 1.78, p = 0.082$
WAIS-IV Scaled Scores	BD	13.17 \pm 3.06	11.53 \pm 2.37	$t(39) = 1.85, p = 0.073$
	DS	11.71 \pm 3.56	10.12 \pm 3.00	$t(39) = 1.50, p = 0.141$
	MR	14.08 \pm 2.41	12.76 \pm 2.49	$t(39) = 1.70, p = 0.097$
e-Cube Scores	e-Cube ^A	61.08 \pm 13.62	62.00 \pm 6.72	$t(39) = -0.26, p = 0.800$
	e-Cube ^S	15.29 \pm 2.54	14.88 \pm 2.83	$t(39) = 0.48, p = 0.631$
	e-Cube ^{M1}	18.08 \pm 4.06	17.82 \pm 3.71	$t(39) = 0.21, p = 0.836$
	e-Cube ^{M2}	17.33 \pm 2.33	16.41 \pm 3.62	$t(39) = 0.99, p = 0.328$
	e-Cube ^P	27.83 \pm 8.38	24.12 \pm 5.98	$t(39) = 1.57, p = 0.125$
	e-Cube ^Z	22.79 \pm 5.40	21.47 \pm 4.91	$t(39) = 0.80, p = 0.428$

5.4.2.4 Preliminary behavior analyses

The behavior data consists of the normalized manipulation steps obtained from e-Cube^A and e-Cube^{M2} and normalized movements obtained from e-Cube^P and e-Cube^Z. The normalized manipulation steps were computed as the total steps taken for an item divided by the number of the shapes in the item - the average manipulation steps taken for a shape. The normalized movements were the total traversed path in an item divided by the shortest path of the item. The behavior analyses were performed by computing the correlations between the behavior data and the 1) e-Cube scores; 2) raw scores of WAIS; 3) scaled scores of WAIS; or 4) age of the Fixed Group participants, as shown in Fig. 5.16.

Interestingly, both positive and negative correlations were found between the behavior data and scores. There were three pairs showing positive correlations with each other, the behavior of e-Cube^A and the scores of e-Cube^{M2}, the behavior of e-Cube^{M2} and the raw scores of DS, and the

Table 5.16: Correlations with degrees of freedom 37 between the behavior data and the 1) e-Cube scores; 2) raw scores of the WAIS-IV subtests; 3) scaled scores of the WAIS-IV subtests; or 4) ages from the Fixed Group participants.

Games/Subtests/		Normalized Manipulation/Movement Steps			
Age		e-Cube ^A	e-Cube ^{M2}	e-Cube ^P	e-Cube ^Z
e-Cube Scores	e-Cube ^A	-0.17 (0.302)	-0.38 (0.020)	-0.24 (0.145)	0.12 (0.461)
	e-Cube ^S	-0.27 (0.108)	-0.19 (0.252)	-0.23 (0.165)	0.05 (0.766)
	e-Cube ^{M1}	-0.08 (0.637)	-0.22 (0.187)	0.10 (0.561)	-0.07 (0.687)
	e-Cube ^{M2}	0.64 (< 0.001)	-0.04 (0.797)	0.25 (0.141)	0.16 (0.341)
	e-Cube ^P	0.02 (0.909)	-0.33 (0.048)	-0.43 (0.008)	-0.37 (0.023)
	e-Cube ^Z	-0.15 (0.391)	-0.41 (0.010)	0.16 (0.308)	-0.13 (0.440)
Raw Scores of WAIS	BD	-0.37 (0.024)	-0.47 (0.003)	-0.18 (0.299)	0.23 (0.170)
	DS	0.25 (0.142)	0.38 (0.019)	0.26 (0.123)	-0.14 (0.424)
	MR	-0.01 (0.936)	-0.32 (0.053)	-0.08 (0.643)	0.11 (0.518)
Scaled Scores of WAIS	BD	-0.37 (0.023)	-0.48 (0.003)	-0.19 (0.260)	0.23 (0.169)
	DS	0.29 (0.087)	0.42 (0.010)	0.21 (0.206)	-0.14 (0.399)
	MR	0.02 (0.888)	-0.26 (0.117)	-0.11 (0.508)	0.09 (0.589)
Age		0.25 (0.134)	0.30 (0.067)	-0.31 (0.059)	-0.02 (0.901)

behavior of e-Cube^{M2} and the scaled scores of DS, which indicated the higher the scores were, the more steps they took. In contrast, the behavior of e-Cube^A was negatively correlated with the raw scores and scaled scores of BD; the behavior of e-Cube^{M2} was negatively correlated with the scores of e-Cube^A, e-Cube^P, and e-Cube^Z and the raw scores of BD and DS, and the scaled scores of BD; the behavior of e-Cube^P was negatively correlated with its own scores; and the behavior of e-Cube^Z was negatively correlated with the scores of e-Cube^Z. These negative correlations showed that the higher the scores were, the fewer steps they took to finish the items. Among the found correlations, the most expected one was that between the normalized movement steps of e-Cube^P and its scores. Since whether the taken path is the shortest or not is a scoring factor in this game, participants who used fewer steps to recreate the path tended to have higher scores. However, a negative correlation was not found between the behavior data and scores in e-Cube^Z, which employed a similar scoring

method with e-Cube^P.

There are also other findings in this behavior analysis. The results obtained by using WAIS raw scores and scaled scores showed similar relationships with the behavior data. Besides this, the ages were weakly positively correlated with the normalized manipulation steps of e-Cube^{M1}, indicating older participants tended to apply more shakes, rotations, or moves on the cubes; the ages were also negatively correlated with the normalized movement steps of e-Cube^P, showing younger participants took fewer repetitive steps to recreate the path.

5.4.3 Hypothesis 3. Reliability of autonomous data collection and SUS results

The e-Cube technology operated smoothly without any significant technical issues identified during the study. The false detection rate was about 0.1% (6 out of 5990), indicating the system is reliable. Note that all of the analyses and computations mentioned previously are based on the corrected data.

The mean SUS score (out of 100) from the 78 participants was 83.24 with SD = 11.55. To further analyze the difference between the fixed and adaptive e-Cube games, the average scores and SDs for both groups were obtained, shown in Table 5.17. The mean of the SUS scores from the Fixed Group participants who experienced the fixed e-Cube games was 80.34 (SD = 13.11). The mean SUS score from the Adaptive Group participants who used the adaptive e-Cube games was 86.10 (SD = 8.84). A two-sample, two-tailed t-test with $\alpha = 0.05$ showed $t(76) = -2.29, p = 0.025$, so Adaptive Group showed a significantly higher mean SUS score with a smaller SD than Fixed Group. Based on an industry standard [109], the usability of both fixed and adaptive e-Cube games is considered as Grade A, i.e., the games are acceptable.

Table 5.17: Mean and SD of SUS scores in Fixed Group, Adaptive Group, and both groups.

Groups	Fixed Group	Adaptive Group	Two-tailed T-Test
Mean (SD)	80.34 (13.11)	86.10 (8.84)	$t(76) = -2.29, p = 0.025$

6. EMBEDDED FACIAL EXPRESSION RECOGNITION

This chapter presents the technical details of the embedded facial expression recognition (FER) algorithm. The human subject study described in Chapter 5 adopted SUS for a simple survey-based usability evaluation. The facial expression often reflects the emotional status of a person and thus can be used as another qualitative measure of usability. The e-Cube system is equipped with an autonomous facial emotion detection algorithm. Although it has not been included in the human subject study yet, this function can enable interesting future studies for better understanding the user engagement and enjoyableness of the e-Cube games when adding fun features to the game design. It can also be used for other types of technologies and devices.

This work was carried out in collaboration with Xiao Liu under the supervision of my advisor Dr. Lee and published in [2]*. My contribution was on building a classic feed-forward CNN for performance comparison and deploying the GA-SVM FER in the e-Cube system and thus this chapter focuses on this aspect.

6.1 GA-SVM-based FER Algorithm

The presented FER method was achieved by Support Vector Machine combined with a genetic algorithm (GA-SVM) using facial geometric features. The method was compared with a CNN, one of the widely adopted methods for facial emotion recognition. The results showed that the test accuracy of GA-SVM was comparable with that of CNN, while the GA-SVM approach employed less complicated models and thus showed more suitable for real-time machine vision applications in automated systems. The collaborative work included dataset preparation, feature extraction and determination, and algorithm evaluation.

The developed FER method follows three steps: 1) landmark detection and geometric feature extraction; 2) parameter selection using genetic algorithm (GA); and 3) SVM-based classification. It was originally designed for a low-cost educational robot, Woody [110], by computing efficiently

*Reprinted with permission from “GA-SVM-based Facial Emotion Recognition Using Facial Geometric Features” by Xiao Liu, Xiangyi Cheng, and Kiju Lee, 2020. IEEE Sensors Journal, 21, 11532–11542, Copyright 2021 by IEEE.

and saving storage space using Support Vector Machine (SVM). These properties of the algorithm make it practical for being deployed in the e-Cube as a novel user experience evaluation. Moreover, harder questions generally raise anxiety and sadness while easier items produce happiness and neural feelings, so FER may estimate the complexity of items and verify the computational complexity measure from a subjective perspective.

6.1.1 Dataset preparation

Since the objective of this algorithm is to recognize facial expressions from the frontal faces, traditional FER datasets only containing full frontal face images from a single angle view are considered acceptable. Existing datasets include the Extended Cohn-Kanade Dataset (CK+) [111], Multimedia Understanding Group (MUG) [112], Japanese Female Facial Expression (JAFFE) [113], and MMI Facial Expression Database (MMI) [114]. Among them, CK+ and MUG were used to perform and validate the proposed FER algorithm. For each dataset, 10% of the images were randomly selected for testing. Among the rest images, 10% were used for validation and 90% for training.

CK+ consists of 593 image sequences taken from 123 subjects with eight emotions, of which 327 sequences are labeled. Each sequence contains images that show the change from a neutral expression to a certain emotion. We extracted six frames from each labeled sequence including the start frame (neutral) and the last five frames (the labeled emotion). Based on this approach, 1,872 frames from eight classes of emotion, anger (225), disgust (295), contempt (90), fear (125), happiness (345), neutral (327), sadness (140), and surprise (415) separately, were extracted. MUG contains 1,462 color image sequences from 86 subjects with seven facial expressions. 2,658 images were processed representing seven classes of emotion, anger (318), disgust (366), fear (207), happiness (520), neutral (521), sadness (339), and surprise (380).

As mentioned, the difference in the total number of emotions existed in CK+ and MUG, i.e. CK+ (8-class) and MUG (7-class), leads to the fact that a simple combination or comparison of these two datasets cannot be achieved. Hence, the common seven emotions (except for the contempt group included in CK+), can be evaluated as 7-class FER. By performing on CK+ solely,

however, we still can develop the vision for our FER algorithm on the 8-class dataset.

6.1.2 Geometric feature extraction

Geometric features spend less computation compared to Traditional Scale-Invariant Feature Transform (SIFT) [115] and Histogram of Oriented Gradient (HOG) [116]. The presented geometric features are based on landmark information and action units (AUs) of faces. Sixty-eight landmarks including contour points/corners of eyes, eyebrows, nose, and chin were detected by Dlib tool kit, which performed face detection in advance. Fig. 6.1 represents the 68 landmarks extracted from two sample images in yellow dots from the detected faces and AUs classified by the corresponding landmarks in other different colors. Describing facial expressions, 30 AUs proposed in Facial Action Coding System (FACS) is considered to taxonomize human facial movements [117, 118]. Among these 30 AUs, 12 AUs listed in Table 6.1 can be described by 68 detected landmarks. Furthermore, we decomposed and regrouped these 12 AUs into facial segments (FSs) to better construct features. The description and the landmarks corresponding to the construction of each FS have been organized in Table 6.1.

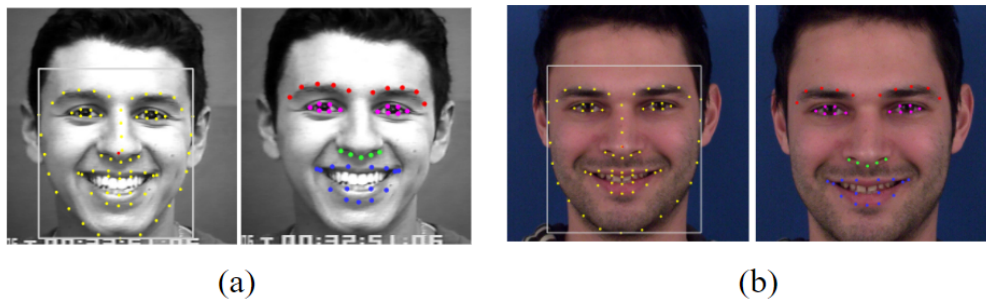


Figure 6.1: Detected landmarks (yellow) and AU groups (multicolors): (a) a sample image from CK+ and (b) one from MUG (reprinted from [2]).

Based on the background of AUs and the construction of FSs that can be described by the facial landmarks, two extracted geometric features are proposed: landmark curvature (LC) and vectorized landmark (VL). Both LC and VL were computed from the landmarks extracted by

Table 6.1: AUs, FSs with description, and associated landmark numbers for each segment (reprinted from [2]).

AU	FS	Description	Index of Landmarks
AU ₁	FS ₁	Left inner brow raiser	20-22
	FS ₂	Right inner brow raiser	23-25
AU ₂	FS ₃	Left outer brow raiser	18-20
	FS ₄	Right outer brow raiser	25-27
AU ₅	FS ₅	Left upper lid raiser	37-40
	FS ₆	Right upper lid raiser	43-46
AU ₇	FS ₇	Left lid tightener	37, 40-42
	FS ₈	Right lid tightener	43, 46-48
AU ₉	FS ₉	Nose wrinkler	32-36
AU ₁₀	FS ₁₀	Upper lid raiser	49-55
AU ₁₂ ; AU ₁₅	FS ₁₁	Left lip corner	49, 61, 68
	FS ₁₂	Right lip corner	55, 65, 66
AU ₂₀ ; AU ₂₃	FS ₁₃	Lip stretched/tightener	49, 55-60
AU ₁₃	FS ₁₄	Left cheek puffer	1-6
	FS ₁₅	Right cheek puffer	12-17
AU ₁₇	FS ₁₆	Chin raiser	7-11

Dlib, the basis of AUs and FSs. The location of an extracted landmark is expressed by the pixel position with respect to the reference frame, the origin of which is located at the upper left corner of the image. With $N = 68$ representing the total number of landmarks, each landmark location is denoted as $L_i = (x_i, y_i)$, for $i = 1, \dots, N$. Therefore, the denotation of each FS constructed by the corresponding subset of landmarks, shown in Table 6.1, can be specified, e.g., $FS_1 = \{L_{20}, L_{21}, L_{22}\}$ and $FS_2 = \{L_{23}, L_{24}, L_{25}\}$.

To create an LC feature using a smoother and more natural curve fitted by a specific FS_n , we applied least squares regression on each FS to achieve curve fitting, the mathematical function of which is expressed as

$$f_{FS_n}(x) = \sum_{k=0}^M a_k x^k \quad (6.1)$$

where M determines the order of the least-squares regression. Higher the order, smoother the fitting function with more expensive computation. Hence, M affects the geometric shape of the curve which leads to different training results. To prevent poor-fitting conditions, linear interpolation constructing new data points was adopted. Given the obtained fitting function based on (6.1), we were able to compute the local curvature at landmark i as:

$$\kappa_i = \frac{|f''_{FS_n}(x_i)|}{\left\{1 + (f'_{FS_n}(x_i))^2\right\}^{\frac{3}{2}}} \quad (6.2)$$

where x_i is the x coordinate of landmark L_i , a group member of FS_n , $f'_{FS_n}(x)$ and $f''_{FS_n}(x)$ are the first and second derivatives with respect to x , respectively. The purple dashed curve shown in Fig. 6.2 is the curve fitting result of FS_1 . Moreover, κ_{21} calculated from L_{21} , a landmark of FS_1 , can be visualized from the dashed white circle below the purple dashed curve.

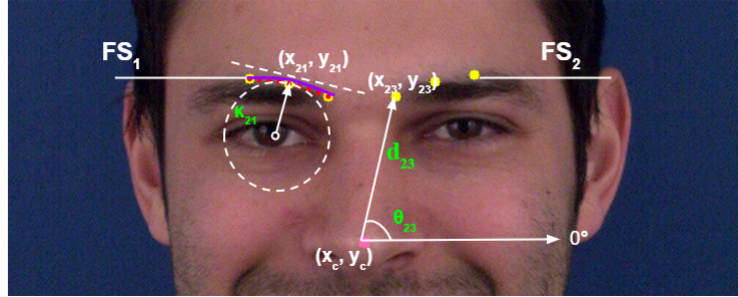


Figure 6.2: Feature extraction from given landmarks associated with FS_1 and FS_2 : LC feature (κ_{21}) at L_{21} , and VL feature (d_{23}, θ_{23}) at L_{23} (reprinted from [2]).

The polar coordinates of the landmarks with respect to the geometric center of all landmarks, the Euclidean distance and angle, are named VL features. For N landmarks, the geometric center is defined as $L_c = (x_c, y_c)$, where $x_c = \frac{1}{N} \sum_{j=1}^N x_j$ and $y_c = \frac{1}{N} \sum_{j=1}^N y_j$. Both distance (d_j) and angle measured from the horizontal axis (θ_j) for L_j are calculated as

$$d_j = \sqrt{(x_j - x_c)^2 + (y_j - y_c)^2}; \quad \theta_j = \tan^{-1} \frac{y_j - y_c}{x_j - x_c}.$$

VL feature d_{23} and θ_{23} at $L_{23} \in FS_2$ are shown in Fig. 6.2.

Applying the feature extraction on CK+ and MUG is one of the most important procedures before feeding the information into the SVM classifier. Fig. 6.3a displays seven images arbitrarily selected from CK+ (top); the detected faces boxed around with 68 landmarks shown in yellow dots for each (middle); and the cropped and resized face images for better visualization of the detected landmarks (bottom). The same operation applies on MUG in Fig. 6.3b.

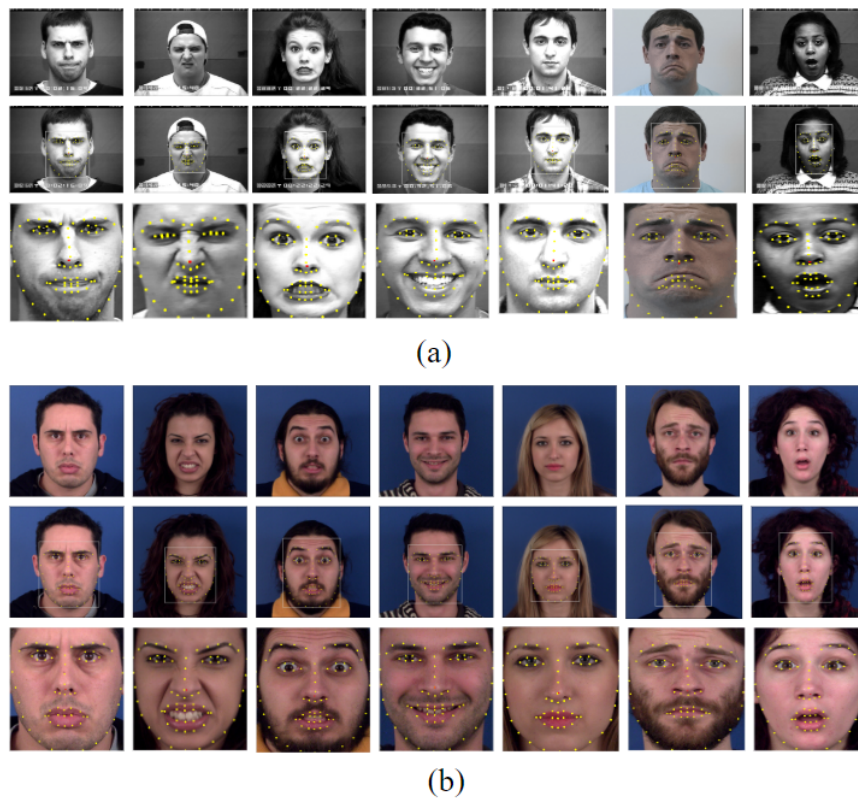


Figure 6.3: (a) CK+ examples of seven emotions (i.e., anger, disgust, fear, happy, neutral, sadness, and surprise) (top); landmarks detected (middle); and cropped and re-sized (bottom) and (b) MUG examples organized in the same way as (a) (reprinted from [2]).

6.1.3 Feature type & interpolating order determination

The efficiency and accuracy of adopting two proposed feature types, VL and LC, requires a verification that the combination of both outperforms either of them. W_1 and W_2 were assigned

to VL and LC as weight parameters when inputting them into SVM. Moreover, recall that an independent parameter M representing the order of the curve fitting function in (6.1) needs to be specified. With initially setting weight parameters W_1 and W_2 to 0.5 and the regularization parameter C in SVM to 1 as default values, efficiency and accuracy of each situation were obtained by feeding the extracted features into SVM on both CK+ and MUG, shown in Table 6.2.

Table 6.2: VL, LC features, and polynomial interpolation order (M) determination (reprinted from [2]).

VL	LC	M	CK+ Dataset		MUG Dataset	
			time (sec)	Accuracy	time (sec)	Accuracy
✓	✗	✗	402.33	76.24%	1387.54	82.20%
✗	✓	2	214.17	72.27%	719.11	78.60%
✗	✓	3	213.38	76.57%	719.36	84.09%
✗	✓	4	215.49	74.82%	708.44	77.84%
✓	✓	2	415.48	84.62%	1421.70	88.02%
✓	✓	3	416.36	88.01%	1399.96	88.83%
✓	✓	4	416.74	85.98%	1406.76	87.48%

Obviously, the validation accuracy of using only one of the features was greatly lower than of adopting both although the computational time was less. Among the cases that were applied by both VL and LC, the performance was the best when M was set to 3, the accuracy of which was the highest while the computational time was still comparable. However, the accuracy dropped dramatically when $M > 3$ due to overfitting. Comparing the spending time of the algorithm on CK+ and MUG, time on MUG was approximately three times larger than time on CK+ since MUG dataset size is bigger. Fortunately, the same trend was perceived in CK+ and MUG, which illuminates the decision of using both VL and LC as well as $M = 3$ is reasonable.

6.1.4 GA-based parameter selection and evaluation on prepared datasets

After confirming the feature type and the order of the curve fitting function, to-be-solved problems still existed including the determination of the subset of the detected 68 landmarks that can extract features and provide better or equal results; the determination of the SVM parameters (W_1 , W_2 , and C). Therefore, GA-based selection that can solve a multi-attribute problem involving non-binary SVM-related parameters and a binary facial landmark selection vector ($\vec{\psi}$) was adopted. By the GA-based parameter selection, the best gene sequence (Γ) that provided the highest average validation accuracy after 300 iterations was obtained and used to get the test accuracy during the evaluation.

After the selection, the number of landmarks of 8-class CK+, 7-class CK+, and 7-class MUG was reduced to 41, 37, and 31, respectively, from the total of 68. Fig. 6.4 displays the selected landmarks (red) for these three experimental cases. Moreover, Table 6.3 shows the selected W_1 and C values for each case as well as the corresponding validation accuracy and test accuracy. The fact that the test accuracy values were slightly higher than the validation accuracy in all three cases implied that the SVM models were not underfitting/overfitting by GA-SVM optimization.

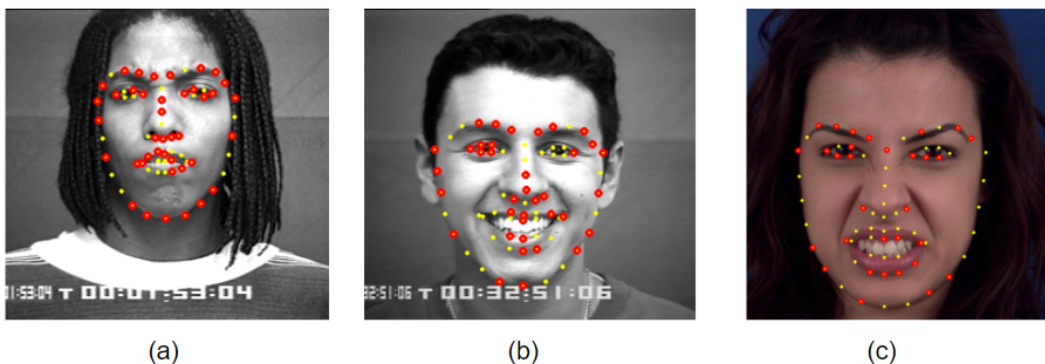


Figure 6.4: Landmarks selected via the GA-based parameter optimization process: (a) CK+ with 8-class; (b) CK+ with 7-class; and (c) MUG with 7-class (reprinted from [2]).

In addition to the validation and test accuracy results, overall classification performance includ-

Table 6.3: SVM parameters and their corresponding validation and test accuracy results (reprinted from [2]).

Dataset	W_1	C	Validation Accuracy	Test Accuracy
CK+ (8)	0.85	$10^{2.5}$	93.57%	95.85%
CK+ (7)	0.75	10^2	95.58%	97.59%
MUG (7)	0.85	10^3	96.29%	96.56%

ing precision, recall, and F1-score was also examined to determine whether or not the SVM results in a balanced recurrence for each emotion class [119]. Table 6.4 represents the statistical results of 8-class CK+, 7-class CK+, and 7-class MUG. After analyzing the data, the average precision of all cases was above 0.97, implying that the percentage of true positives was high and the classification result was balanced. The recall value was about 0.95 overall, indicating that the number of false negative classifications was low and thus the model was sensitive. The F1-score conveyed the balance between the precision and recall values. The average of the F1-score was about 0.96, implying that the classifier was competitive in terms of precision and recall.

6.2 Performance Comparison with CNN

CNN is known as a highly flexible and commonly used classifier with excellent performance. However, CNN is a relatively complex model compared to SVM, which implies the computation is also larger. If our results are comparable with the outcome using CNN, the presented method will be more appropriate in terms of a real-time application, such as the usage in e-Cube games.

One classic feed-forward CNN was constructed to compare our results. As same as the data preparation used for SVM training and test sets, we randomly selected 10% images from each dataset as the test set and randomly separated 90% and 10% of the rest images as the training set and validation set. The architecture of CNN included four convolutional layers, one fully connected layer, and one softmax output layer (Fig. 6.5). The input images were in 80×80 pixels and normalized. The first convolutional layer convolved images with 16 kernels in 3×3 (3-pixel

Table 6.4: Statistics of the GA-SVM based FER: precision, recall, and F1-score (reprinted from [2]).

Dataset	Emotion	Precision	Recall	F1-score
CK+ (8-class)	Anger	1.00	0.95	0.98
	Disgust	1.00	0.93	0.96
	Fear	1.00	0.92	0.96
	Happy	1.00	1.00	1.00
	Sad	0.93	1.00	0.97
	Surprise	0.97	0.95	0.96
	Neutral	0.84	0.97	0.90
	Contempt	1.00	0.89	0.94
CK+ (7-class)	Anger	1.00	0.96	0.97
	Disgust	1.00	0.97	0.98
	Fear	1.00	1.00	1.00
	Happy	1.00	1.00	1.00
	Sad	1.00	0.86	0.92
	Surprise	0.97	1.00	0.98
	Neutral	0.90	0.88	0.89
MUG (7-class)	Anger	0.97	0.97	0.97
	Disgust	0.97	0.97	0.97
	Fear	0.90	0.90	0.90
	Happy	1.00	0.98	0.99
	Sad	0.97	0.94	0.95
	Surprise	0.92	0.95	0.94
	Neutral	0.98	1.00	0.99

width and 3-pixel height) patch. Its weight tensor had a volume size of $3 \times 3 \times 1$ and 16 output channels to the subsequent convolutional layer. The output features of the first convolution layer then followed a process of convolving training patterns with weight tensors and adding bias. Then, max pooling in size of 2×2 was performed. The image size was reduced to 40×40 before being fed into the next hidden layer. The second convolution layer had 32 kernels for each 3×3 patch. Therefore, the weight tensor had a volume size of $3 \times 3 \times 32$ with 32 output channels. After following the same convolution and max pooling processes, the image size was further reduced to 20×20 . The convolutional layer III and IV had 64 and 128 kernels, respectively, for each 3×3

patch.

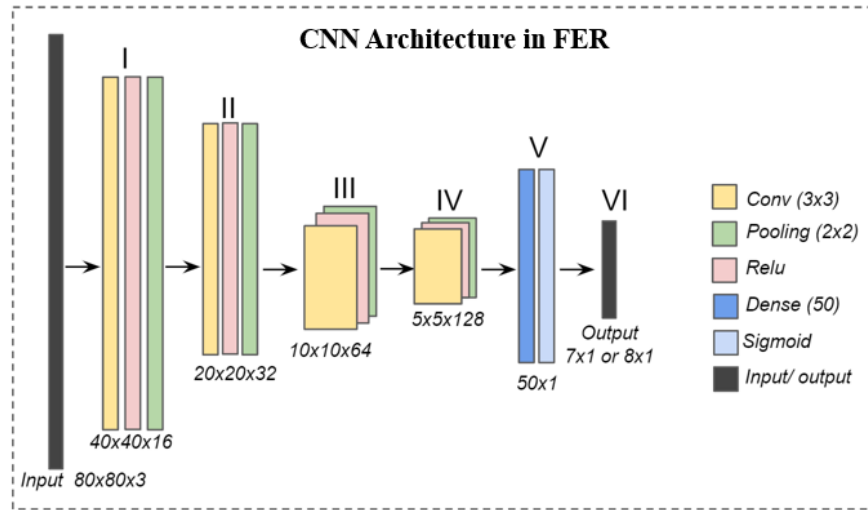


Figure 6.5: Proposed CNN architecture for FER: 4 convolutional layers followed by 1 fully connected layer (reprinted from [2]).

After four layers of convolution, one fully-connected (local V in Fig. 6.5) and one softmax output layer (local VI in Fig. 6.5) were followed. Local V layer contained 50 neurons and local VI layer had 7 or 8 neurons – which was the number of emotion classes. The input of local V was multiplied by a weight tensor, added by a bias, and then applied to a sigmoid function. To avoid overfitting, a dropout was implemented prior to local VI. Cross entropy was used as a loss function and Adam was chosen as an optimizer. Fifty epochs were performed as the result converged within this range. Table 6.5 shows the statistical results using this CNN model. Comparisons between the presented GA-SVM and this CNN method in terms of test accuracy on 8-class CK+, 7-class CK+, and 7-class MUG are shown in Fig. 6.6. GA-SVM achieved 95.85%, 97.59%, and 96.56% in test accuracy on 8-class CK+, 7-class CK+, and 7-class MUG, respectively. The CNN-based method resulted in 95.43%, 97.34%, and 99.62% on 8-class CK+, 7-class CK+, and 7-class MUG. The presented GA-SVM method resulted in slightly higher test accuracy values for CK+ while CNN outperformed for MUG.

Table 6.5: Statistics of CNN based FER: precision, recall, and F1-score (reprinted from [2]).

Dataset	Emotion	Precision	Recall	F1-score
CK+ (8-class)	Anger	0.84	1.00	0.91
	Disgust	0.94	0.97	0.96
	Fear	0.95	1.00	0.97
	Happy	1.00	1.00	1.00
	Sad	1.00	1.00	1.00
	Surprise	0.98	0.98	0.98
	Neutral	0.96	0.76	0.85
	Contempt	0.90	1.00	0.91
CK+ (7-class)	Anger	0.96	1.00	0.98
	Disgust	0.96	0.96	0.96
	Fear	1.00	1.00	1.00
	Happy	1.00	1.00	1.00
	Sad	1.00	0.94	0.97
	Surprise	1.00	0.97	0.99
	Neutral	0.90	0.93	0.91
MUG (7-class)	Anger	1.00	1.00	1.00
	Disgust	1.00	1.00	1.00
	Fear	1.00	0.96	0.98
	Happy	1.00	1.00	1.00
	Sad	1.00	1.00	1.00
	Surprise	0.97	1.00	0.98
	Neutral	1.00	1.00	1.00

6.3 Performance Comparison with Other Existing Methods

Table 6.6 lists relatively recent work in the field of vision-based FER using the traditional datasets, such as CK+, MUG, JAFFE, and MMI, and reports FER performance results in comparison with the presented approach. Existing literature reporting only the cross validation accuracy is not included in this table because the results – regardless if it is K-fold or leave-N-out approach – are likely biased. As shown in Table 6.6, the Viola-Jones algorithm is commonly used for face detection [120, 121, 122, 123]. Some papers did not report the information about the specific face detection algorithm; and there also exist FER algorithms that did not require

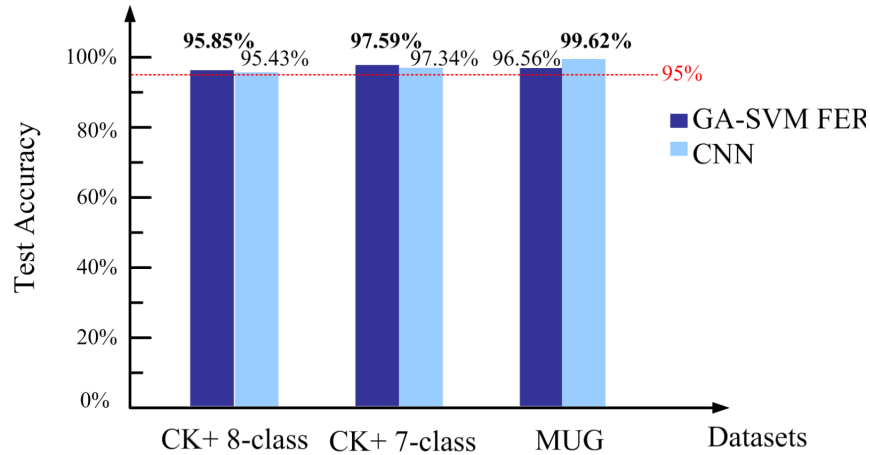


Figure 6.6: Test accuracy comparison between GA-SVM and CNN on 8-class CK+, 7-class CK+ and 7-class MUG (reprinted from [2]).

this process exclusively [78, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133]. Applications using SVM, LDA, and some network methods required extracted input data fed into the classifiers [78, 121, 122, 123, 127, 128, 130, 133], while other methods took the images in the datasets directly as the input [120, 125, 126, 129, 131, 132]. The training accuracy for 6-class CK+ varied between 85.42% and 99.33%; 7-class CK+ varied between 95.6% and 96.46%; 8-class CK+ around 95.51%; 6-class MUG varied between 87.65% and 99.3%; 7-class MUG around 99.3%; 6-class JAFFE varied between 84% and 98.8%; 7-class JAFFE around 98.43%; and 6-class MMI varied between 81.5% and 97.55%. One literature also reported 76% accuracy on the combined dataset of MUG, CK+, and JAFFE. The presented method resulted in 95.85% on 8-class CK+ and 97.59% on 7-class CK+, which are higher than these previously reported results. On the other hand, CNN showed a slightly better performance than our method on 7-class MUG, consistent with the result obtained using our algorithm.

Unlike using the traditional datasets, FER using wild datasets involves significant inconsistency and uncertainties. Therefore, deep neural network (DNN) is more broadly adopted for FER with wild datasets because it shows better performance in terms of the feature extraction and classification, compared to other traditional machine learning methods, such as SVM and LDA [137]. In

Table 6.6: Comparison of recent FER approaches based on frontal face data (reprinted from [2]).

Dataset	#. Class	Face Detection	Feature Exaction	Classifier	Accuracy (%)	Ref.
MMI CK+	6	Viola-Jones algorithm	Geometric feature and LBP texture feature fused by autoencoder	SVM	93.19 95.01	[123]
MMI CK+	6	Viola-Jones algorithm	Geometric feature and LBP texture feature fused by autoencoder	Self-organzing map (SOM)	97.55 98.95	[123]
FER-2013 [134]	7	N/A	CNN parameters	SVM	97.85	[78]
RaFD [135]	6	Viola-Jones algorithm	Pyramid HOG (PHOG)	GA-LDA	97.33	[121]
CK+	8	N/A	Difference center-symmetric local directional pattern (DCS-LDP)	SVM	95.51	[127]
CK+	6	N/A	Haar Wavelet Transform and Gabor wavelets	SVM	98	[130]
JAFFE CK+ MUG	6	Viola-Jones algorithm	Geometric facial landmarks	CNN	84.00 85.42 89.19	[122]
CK+ MUG JAFFE MMI	6	N/A	Normalized distance signature	Multilayer Perceptron (MLP)	96.3 96.7 94 81.5	[124]
CK+ MUG MMI JAFFE	6	N/A	Grids, triangles on face images and texture features found by salient landmarks	Deep belief network (DBN)	98 99.1 93.1 98.8	[133]
CK+ MUG MMI JAFFE	6	N/A	Distance and texture characteristics among the landmark points	Radial basis function (RBF) network	98.6 99.3 90.9 97.6	[128]
Combination of MUG, CK+ and JAFFE	7	Used, but not specified	Face alignment by landmarks with CNN		76	[125]
JAFFE CK+	7	Viola-Jones algorithm	CNN with histogram equalization (HE) and discrete wavelet transform (DWT)		98.43 96.46	[120]
MUG CK+	7	N/A	CNN-base and hybrid inherited lightweight network HiNet		99.3 95.7	[126]
RaFD CK+ MUG	6	N/A	CNN with 4 convolutional layers, 3 max-pooling, 1 fully-connected, and 1 softmax output layer.		93.33 99.33 87.65	[131]
CK+ FER-2013 FERG [136] JAFFE	6	N/A	End-to-end deep learning framework based on attentional convolutional network		98.0 70.02 99.3 92.8	[132]
FER-2013	7	N/A	CNN		88.9	[129]
CK+	8				95.85	
CK+	7	Viola-Jones algorithm	Presented geometric features with GA-SVM optimization		97.59	-
MUG	7				96.56	

addition, the features extracted from a deep convolutional neural network (DCNN) fed into other traditional machine learning techniques (i.e., SVM, LDA, or KNN) can further improve the FER performance compared to pure CNN in the wild [138]. Most of these algorithms, however, used DNN and its extension for both feature extraction and classification. DCNN with pre-processed images (i.e., face crop, gray scale, and contrast normalization) showed the test accuracy of 76.79% and 77.08% on AffectNet and the Karolinska Directed Emotional Faces (KDEF) dataset [139], respectively [140]. DenseCANet and DenseSANet adopted an attention mechanism based module with DenseNet, which is one of the DCNN architectures [141]. The accuracy of both DenseCANet and DenseSANet was over 60% for the AffectNet dataset.

6.4 FER Application in e-Cube

High efficiency and accuracy make the presented GA-SVM algorithm appropriate to be embedded into the real-time-based e-Cube games by using the laptop's built-in camera. Once the checkbox "Record Facial Expression" on the access page (Fig. 3.7) is enabled by the user, each frame containing the frontal face can be captured by the built-in camera and the facial expressions can be analyzed using the proposed algorithm. This analysis can help understand user experience and provide subjective evidence for further supporting the use of the proposed computational measures of play complexity. However, more human subject studies are needed to validate the role of this novel approach in the e-Cube system.

7. CONCLUSION AND DISCUSSION

This dissertation presented the design, development, and evaluation of the e-Cube system as a new form of technology for automated cognitive assessment and the facial expression algorithm for future user experience evaluation. Six types of generative, adaptive games including e-Cube^A, e-Cube^S, e-Cube^{M1}, e-Cube^{M2}, e-Cube^P, and e-Cube^Z were designed to measure the cognitive skills of individuals including fine motor skills, visual-motor integration, perceptual reasoning, and working memory. Computational measures of play complexity were proposed and adopted for autonomous item generators in the adaptive game system. The e-Cube technology and the adaptive games were evaluated by testing the three hypotheses: 1) the performance indicators (i.e. scores and time on individual items) are correlated with the play complexity values of the items in each e-Cube game, 2) the scores from the adaptive e-Cube games are correlated with that from the existing standardized methods, and 3) the e-Cube technology is reliable and usable. The human subject study for evaluating the hypotheses was conducted on the TAMU campus, and thus most of our study participants were students aged between 18 and 30. Therefore, the results must be interpreted considering the skewed age distribution.

Hypothesis 1 was tested by the correlational analyses performed between the play complexity values and the performance indicators – the mean scores and mean completion time obtained from Fixed Group. Each of them was used to compute the correlation with the play complexity values. The play complexity values were found to be highly correlated with the mean completion time in all the games and with the mean scores, except for e-Cube^{M2}. As discussed previously, two items in e-Cube^{M2} involved symmetric arrangements of the geometric shapes – which make it easier to memorize regardless of the geometric complexity of the shapes. The same play complexity measure was used for both e-Cube^{M2} and e-Cube^A, which appear to be not ideal for reflecting the difficulty associated with memory tasks, despite high correlation was found using the mean completion time. This problem can be avoided at the software level by adjusting the algorithm for the item generator. Moreover, such symmetric images are rarely created in the adaptive version and

thus not expected to affect the assessment outcome. Specifically, the adaptive e-Cube^{M2} showed significant correlation with DS implying its potential validity for assessing one's working memory skills while the fixed e-Cube^{M2} did not show any correlation with DS.

To test the preliminary assessment validity of the adaptive e-Cube games (**Hypothesis 2**), correlational analyses were performed between the scores of the WAIS-IV subtests (i.e. raw scores and scaled scores) and e-Cube scores. The correlations had similar trends using the raw scores and scaled scores. Overall, the adaptive e-Cube version, compared to the fixed version, yielded more significant correlations with the WAIS-IV subtests. The adaptive e-Cube games employed a common discontinuation rule (i.e., the significantly small step size leading to termination of the game), which possibly reduces the number of items in each game, fatigue, and unintended correct answers. The automatic item generator in the adaptive games adjusts item complexity based on real-time performance, enabling the system to generate a more appropriate assessment for everyone. Inter-correlational analyses also showed that the games in the adaptive version are more independent from one another implying that they tap cognitive domains differently than the fixed version. Besides this, the age effect and gender difference were also investigated. The correlation between ages and scores was only observed in e-Cube^{M1}, where ages increase as their e-Cube^{M1} scores decrease. The score differences between gender were not found in this study. In addition, the preliminary behavior analyses were conducted to determine the relationships between the normalized manipulation/movement steps and 1) the scores and 2) ages. Most of the negative correlations were related to assembly or path-planing tasks, which showed that the participants receiving higher scores in these games/subtests tended to avoid repetitive or redundant steps. However, the positive correlations were related to either e-Cube^A, e-Cube^{M2}, or DS. Both e-Cube^A and e-Cube^{M2} involved assembly tasks, while both e-Cube^{M2} and DS were designed for assessing working memory skill. Due to the currently limited sample size, the reasons that cause the positive correlations are not clear yet, thus more further studies are needed.

There were some other notable findings from the Hypothesis 2 evaluation study. The mean scores in MR and e-Cube^S between Fixed Group and Adaptive Group were tested to be different

in the opposite way. The mean score of Fixed Group was higher in e-Cube^S and lower in MR than that of Adaptive Group. In fixed e-Cube^S, the correctness of each item was found to be inconsistent (i.e., most of the participants correctly answered items 1-7 and 9 while only 51% and 24% of the participants correctly answered items 8 and 10, respectively). This inconsistency resulted in a higher mean score in Fixed Group, but the results were not correlated with MR. On the other hand, Adaptive Group showed a significant correlation between the e-Cube^S scores and the MR results. e-Cube^{M1} was intended to measure one's working memory by displaying a sequence of images in an increased length. It was initially hypothesized that the results from e-Cube^{M1} would be correlated with DS, but no significant correlation was found between the two. DS relies on auditory recall of numbers, sequence, and order, while e-Cube^{M1} is performed by the visual recall of geometric images. The difference between the auditory and visual recalls, language and cultural differences in participants, and how these affect and differently reflect one's working memory skills are yet to be investigated. In addition, many participants experienced more fatigue during DS and e-Cube^{M1} than in any other subtests and games due to the higher, continuous attention level. Compared to e-Cube^{M1}, another memory task e-Cube^{M2}, requiring the examinees to memorize an assembled configuration, did result in a significant correlation with DS in the adaptive version. Several improvements have also been identified to be made in e-Cube^{M2}. This game asks the examinee to memorize an item displayed on a screen for five seconds and once it disappears with a beeping sound the examinee is asked to reconstruct the item using the geometric cubes. Some participants attempted to move the cubes before the displayed item disappears. In this case, the administrator reminded him/her not to start before the beep. This occasionally interrupted proper memorizing tasks in those participants. Clearer instructions embedded in the game system and periodic reminding can help reduce such cases. As for e-Cube^Z, it was not correlated with any WAIS-IV subtests, while being correlated with other e-Cube games. In addition, e-Cube^Z and e-Cube^P were correlated with e-Cube^A, implying that their game settings/measured cognitive outcomes are similar to e-Cube^A and thus may be removed from the e-Cube games.

For testing **Hypothesis 3**, the technical functionality and usability of the system were examined

by checking the false detection rate and using SUS. The low false detection rate (0.1%) proves the reliability and technical functionality of the e-Cube system. The average SUS scores of both fixed and adaptive e-Cube games were acceptable based on the industry standard [109], while the adaptive games resulted in a significantly higher mean SUS score than the fixed one. The SD of SUS scores in the adaptive games was also less than that of the fixed version showing less variability in the scores.

To further investigate the usability of e-Cube, a facial emotion recognition algorithm using two newly defined geometric features, landmark curvature and vectorized landmark, developed in collaboration with Xiao Liu and under the supervision of my advisor Dr. Lee [2] was embedded in the system. These features were extracted from facial landmarks associated with individual components of facial muscle movements. The algorithm combined SVM-based classification with a GA for a multi-attribute optimization problem of feature and parameter selection. Experimental evaluations were conducted on the extended CK+ and the MUG dataset. For 8-class CK+, 7-class CK+, and 7-class MUG, the validation accuracy was 93.57, 95.58, and 96.29%; and the test accuracy resulted in 95.85, 97.59, and 96.56%, respectively. Overall precision, recall, and F1-score were about 0.97, 0.95, and 0.96. The technique was also compared with a CNN, one of the widely adopted methods for facial emotion recognition. The presented method showed slightly higher test accuracy than CNN for 8-class CK+ (95.85% (SVM) vs. 95.43% (CNN)) and 7-class CK+ (97.59 vs. 97.34), while the CNN slightly outperformed on the 7-class MUG dataset (96.56 vs. 99.62). Compared to CNN-based approaches, this method employs less complicated models and thus shows potential for real-time machine vision applications in automated systems, e.g. e-Cube system. However, further studies are needed to demonstrate its use in e-Cube.

Future Work

The e-Cube system is designed to be low-cost, low-maintenance, and easy-to-use with the features of autonomous administration and data collection. The results from the adaptive e-Cube games have been demonstrated to be correlated with that from the existing standardized methods. These properties and evidence make it have great potential in serving as a tool for screening

MCI routinely, targeting older adults aged 65 years old and older. Therefore, future research may address its utility as a routine cognitive assessment for patients with MCI.

Patients with MCI are indeed at significantly higher risk of developing AD and other dementias than those without MCI [142]. Early diagnosis at the MCI stage can bring early and targeted intervention, and thus results in a better overall health outcome and well-being of a patient. Such efforts would lessen the symptoms within a limited time, prioritize patients' health such as planning the future and changing the lifestyle, and save costs by applying the corresponding interventions that can prevent or delay the onset of AD and other dementias [143, 144]. Cognitive declines in MCI including memory deficits, attention/executive dysfunction, and visuospatial impairment are believed to be greater compared to the normal performance given the age and educational level [145], thus can be distinguished by screening tools before performing neuroimaging and cerebrospinalfluid-CSF for diagnosis via observed abnormalities in these biomarkers [146, 147].

While several screening instruments for MCI and AD (i.e. MMSE and MoCA) have been developed, validated, and even adopted in clinical practice, limitations and challenges in identifying patients with mild symptoms still exist. There is no denying that the informant questionnaires involved in the tests improved the screening accuracy [148]. However, inaccurate and unsure answers provided by families or friends may lead to false detection and cause negative even severe impacts. In addition, one-time screening barely offers a dynamic cognitive change over time, which can hardly detect a long preclinical phase where the brain changes without measurable symptoms prior to the occurrence of MCI. Therefore, routine cognitive assessment providing a baseline functioning and its changes over time is in demand, which can potentially expose a great number of patients with MCI [149, 150]. However, using MMSE or MoCA as a routine cognitive assessment potentially brings bias due to the learning effect since it requires users to screen their cognitive ability on a regular basis. Moreover, both MMSE and MoCA fail to eliminate the interfering impacts associated with age, education, cultural background, and language [151, 152]. To extend the usage of these screening instruments in diverse populations, derivative versions in different languages are also promoted. However, studies pointed out that result generalization from one population

to another should be treated cautiously although equivalence between instruments are maximized [153].

The adaptive feature allows the e-Cube technology to have great potential in being a routine cognitive assessment because it can generate and offer different items every time a user plays it. The human subject study testing the e-Cube as a routine cognitive assessment on older adults has been initialized in Peach Creek Assisted Living in College Station, Texas. Several older adults have experienced the games but the study was delayed due to the COVID-19 pandemic. In spite of that, more evaluations targeting older adults can be done in the future. The games have been tailored for older adults based on the results and observations obtained from the previous study. For example, the 20 items in e-Cube^A have been refined to 10 for reducing difficulty and time, thus lessening fatigue; e-Cube^P and e-Cube^Z have been removed from the system; and the discontinuation rule of the games has been changed from the small step size to two consecutive wrong answers leading to the termination. The color games shown in Appendix A may also be used for older adults because they are easier.

Further human subject studies can also evaluate the proposed FER algorithm as a novel user experience evaluation approach. The hypothesis would be "items with high play complexity result in negative emotions, while items with low play complexity result in positive emotions". The real-time detected emotions can be analyzed with the scores obtained from each item as well as the SUS results.

The adaptive game procedure can be optimized after simulations and further analyses. The current item generators always offer items from the lowest to highest play complexity following the procedure, which might be inefficient in assessing one's cognitive skills. The procedure can adopt the optimization with boundaries strategy, which reaches the range where the examinee's cognitive ability falls in by using the minimum steps/items. This new procedure can be evaluated by adjusting the order of the items and then simulating users' performance based on the adjusted order using the score of each item obtained from the previous study.

As for the reliability of the system, there is still room for improvements to further eliminate

false detection. The majority of the incorrect detection was caused by the player's hands blocking the view or interrupting proper vision processing. An algorithm to detect hands may be added to prevent hand motions from interfering with block detection.

REFERENCES

- [1] D. Jeong, K. Endri, and K. Lee, “TaG-Games: tangible geometric games for assessing cognitive problem-solving skills and fine motor proficiency,” in *2010 IEEE Conference on Multisensor Fusion and Integration*, pp. 32–37, IEEE, 2010.
- [2] X. Liu, X. Cheng, and K. Lee, “GA-SVM based Facial Emotion Recognition using Facial Geometric Features,” *IEEE Sensors Journal**, 2020.
- [3] A. P. Mackey, A. T. Park, S. T. Robinson, and J. D. Gabrieli, “A pilot study of classroom-based cognitive skill instruction: effects on cognition and academic performance,” *Mind, Brain, and Education*, vol. 11, no. 2, pp. 85–95, 2017.
- [4] D. M. Lipnicki, S. R. Makkar, J. D. Crawford, A. Thalamuthu, N. A. Kochan, M. F. Lima-Costa, E. Castro-Costa, C. P. Ferri, C. Brayne, B. Stephan, *et al.*, “Determinants of cognitive performance and decline in 20 diverse ethno-regional groups: A COSMIC collaboration cohort study,” *PLoS medicine*, vol. 16, no. 7, p. e1002853, 2019.
- [5] M. Cauchoix, P. Chow, J. Van Horik, C. Atance, E. Barbeau, G. Barragan-Jason, P. Bize, A. Boussard, S. D. Buechel, A. Cabirol, *et al.*, “The repeatability of cognitive performance: a meta-analysis,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 373, no. 1756, p. 20170281, 2018.
- [6] A. Iizuka, H. Suzuki, S. Ogawa, T. Takahashi, S. Murayama, M. Kobayashi, and Y. Fujiwara, “Association between the frequency of daily intellectual activities and cognitive domains: A cross-sectional study in older adults with complaints of forgetfulness,” *Brain and Behavior*, vol. 11, no. 1, p. e01923, 2021.
- [7] J. A. Holdnack, “The development, expansion, and future of the WAIS-IV as a cornerstone in comprehensive cognitive assessments,” in *Handbook of psychological assessment*,

*©2021 IEEE. Reprinted, with permission, from Xiao Liu, Xiangyi Cheng, and Kiju Lee, GA-SVM based Facial Emotion Recognition using Facial Geometric Features, IEEE Sensors Journal, 2020

pp. 103–139, Elsevier, 2019.

- [8] H. Woodford and J. George, “Cognitive assessment in the elderly: a review of clinical methods,” *QJM: An International Journal of Medicine*, vol. 100, no. 8, pp. 469–484, 2007.
- [9] A. S. Kaufman and E. O. Lichtenberger, *Assessing adolescent and adult intelligence*. John Wiley & Sons, 2005.
- [10] D. E. Hartman, “Wechsler Adult Intelligence Scale IV (WAIS IV): return of the gold standard,” *Applied neuropsychology*, vol. 16, no. 1, pp. 85–87, 2009.
- [11] S. Stirk, B. Field, and J. Black, “An independent investigation of the utility of the learning disability screening questionnaire (LDSQ) within a community learning disability team,” *Journal of Applied Research in Intellectual Disabilities*, vol. 31, no. 2, pp. e223–e228, 2018.
- [12] G. H. Roid and M. Pomplun, *The stanford-binet intelligence scales*. The Guilford Press, 2012.
- [13] E. C. Duggan, L. M. Awakon, C. C. Loaiza, and M. A. Garcia-Barrera, “Contributing towards a cultural neuropsychology assessment decision-making framework: Comparison of WAIS-IV norms from Colombia, Chile, Mexico, Spain, United States, and Canada,” *Archives of Clinical Neuropsychology*, vol. 34, no. 5, pp. 657–681, 2019.
- [14] K. Wilson and L. Gilmore, “Assessing intellectual functioning in young adolescents: How do the WISC-IV and SB5 compare?,” *Journal of Psychologists and Counsellors in Schools*, vol. 22, no. 1, pp. 1–14, 2012.
- [15] J. Raven, “The Raven’s progressive matrices: change and stability over culture and time,” *Cognitive psychology*, vol. 41, no. 1, pp. 1–48, 2000.
- [16] T. Zhuo and M. Kankanhalli, “Solving Raven’s Progressive Matrices with Neural Networks,” *arXiv preprint arXiv:2002.01646*, 2020.
- [17] T. D. Matthews and K. S. Lassiter, “What does the wonderlic personnel test measure?,” *Psychological reports*, vol. 100, no. 3, pp. 707–712, 2007.

- [18] B. D. Lyons, B. J. Hoffman, and J. W. Michel, “Not much more than g? An examination of the impact of intelligence on NFL performance,” *Human Performance*, vol. 22, no. 3, pp. 225–245, 2009.
- [19] M. F. Folstein, S. E. Folstein, and P. R. McHugh, ““mini-mental state”: a practical method for grading the cognitive state of patients for the clinician,” *Journal of psychiatric research*, vol. 12, no. 3, pp. 189–198, 1975.
- [20] Z. S. Nasreddine, N. A. Phillips, V. Bédirian, S. Charbonneau, V. Whitehead, I. Collin, J. L. Cummings, and H. Chertkow, “The Montreal Cognitive Assessment, MoCA: a brief screening tool for mild cognitive impairment,” *Journal of the American Geriatrics Society*, vol. 53, no. 4, pp. 695–699, 2005.
- [21] C. R. Glass, T. V. Merluzzi, J. L. Biever, and K. H. Larsen, “Cognitive assessment of social anxiety: Development and validation of a self-statement questionnaire,” *Cognitive Therapy and Research*, vol. 6, no. 1, pp. 37–55, 1982.
- [22] O. C. C. W. Group *et al.*, “Cognitive assessment of obsessive-compulsive disorder,” *Behaviour Research and Therapy*, vol. 35, no. 7, pp. 667–681, 1997.
- [23] S. M. Bodmann and D. H. Robinson, “Speed and performance differences among computer-based and paper-pencil tests,” *Journal of Educational Computing Research*, vol. 31, no. 1, pp. 51–60, 2004.
- [24] R. M. Noland, “Intelligence testing using a tablet computer: Experiences with using Q-interactive,” *Training and Education in Professional Psychology*, vol. 11, no. 3, p. 156, 2017.
- [25] C. Singleton, K. Thomas, and J. Horne, “Computer-based cognitive assessment and the development of reading,” *Journal of Research in Reading*, vol. 23, no. 2, pp. 158–180, 2000.

- [26] M. Tenorio Delgado, P. Arango Uribe, A. Aparicio Alonso, and R. Rosas Diaz, “TENI: A comprehensive battery for cognitive assessment based on games and technology,” *Child Neuropsychology*, vol. 22, no. 3, pp. 276–291, 2016.
- [27] T. Tong, J. Yeung, J. Sandrakumar, M. Chignell, M. C. Tierney, and J. Lee, “Improving the ergonomics of cognitive assessment with serious games,” in *Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care*, vol. 4, pp. 1–5, SAGE Publications Sage India: New Delhi, India, 2015.
- [28] T. Tong, M. Chignell, M. C. Tierney, and J. Lee, “A serious game for clinical assessment of cognitive status: validation study,” *JMIR serious games*, vol. 4, no. 1, p. e7, 2016.
- [29] T. Tong, M. Chignell, P. Lam, M. C. Tierney, and J. Lee, “Designing serious games for cognitive assessment of the elderly,” in *Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care*, vol. 3, pp. 28–35, SAGE Publications Sage CA: Los Angeles, CA, 2014.
- [30] E. Sharlin, Y. Itoh, B. Watson, Y. Kitamura, S. Sutphen, and L. Liu, “Cognitive cubes: a tangible user interface for cognitive assessment,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 347–354, 2002.
- [31] H. Jimison, M. Pavel, K. Wild, D. Williams, J. McKanna, and P. Bissel, “Embedded Assessment of Cognitive Performance with Elders’ Use of Computer Games in a Residential Environment,” in *Proceedings of the Workshop on the Cognitive Science of Games and Gaming*, 2006.
- [32] J. M. Reid, “Testing nonverbal intelligence of working-age visually impaired adults: Evaluation of the Adapted Kohs Block Design Test,” *Journal of Visual Impairment & Blindness*, vol. 96, no. 8, pp. 585–595, 2002.
- [33] A. C. Dunn, A. Qiao, M. R. Johnson, and M. Kunda, “Measuring More to Learn More From the Block Design Test: A Literature Review,” in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 43, 2021.

- [34] S. Mendoza-Sánchez, F. Molina-Rueda, L. L. Florencio, M. Carratalá-Tejada, and A. Cuesta-Gómez, “Reliability and agreement of the Nine Hole Peg Test in patients with unilateral spastic cerebral palsy,” *European Journal of Pediatrics*, pp. 1–8, 2022.
- [35] A. Cerrato, M. Ponticorvo, O. Gigliotta, P. Bartolomeo, and O. Miglino, “The assessment of visuospatial abilities with tangible interfaces and machine learning,” in *International work-conference on the interplay between natural and artificial computation*, pp. 78–87, Springer, 2019.
- [36] S. Mironcika, A. de Schipper, A. Brons, H. Toussaint, B. Kröse, and B. Schouten, “Smart toys design opportunities for measuring children’s fine motor skills development,” in *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, pp. 349–356, 2018.
- [37] F. Ferrara, M. Ponticorvo, A. D. Ferdinando, and O. Miglino, “Tangible interfaces for cognitive assessment and training in children: LogicART,” in *Smart Education and e-Learning 2016*, pp. 329–338, Springer, 2016.
- [38] Y. Cheng, “Improving Cognitive Diagnostic Computerized Adaptive Testing by Balancing Attribute Coverage: The Modified Maximum Global Discrimination Index Method,” *Educational and Psychological Measurement*, vol. 70, no. 6, pp. 902–913, 2010.
- [39] L. Benedetto, A. Cappelli, R. Turrin, and P. Cremonesi, “R2DE: a NLP approach to estimating IRT parameters of newly generated questions,” in *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pp. 412–421, 2020.
- [40] J. Kean, E. F. Bisson, D. S. Brodke, J. Biber, and P. H. Gross, “An introduction to item response theory and Rasch analysis: Application using the eating assessment tool (EAT-10),” *Brain Impairment*, vol. 19, no. 1, pp. 91–102, 2018.
- [41] H. Wouters, A. H. Zwinderman, W. A. van Gool, B. Schmand, and R. Lindeboom, “Adaptive cognitive testing in dementia,” *International journal of methods in psychiatric research*, vol. 18, no. 2, pp. 118–127, 2009.

- [42] H. Wouters, J. van Campen, B. Appels, R. Lindeboom, M. Buitter, R. J. de Haan, A. H. Zwinderman, W. A. van Gool, and B. Schmand, “Does adaptive cognitive testing combine efficiency with precision? Prospective findings,” *Journal of Alzheimer’s Disease*, vol. 25, no. 4, pp. 595–603, 2011.
- [43] A. Huebner, “An overview of recent developments in cognitive diagnostic computer adaptive assessments,” *Practical Assessment, Research, and Evaluation*, vol. 15, no. 1, p. 3, 2010.
- [44] N. A. Thompson, “A practitioner’s guide for variable-length computerized classification testing,” *Practical Assessment, Research, and Evaluation*, vol. 12, no. 1, p. 1, 2007.
- [45] B. A. Bergstrom, M. E. Lunz, and R. C. Gershon, “Altering the level of difficulty in computer adaptive testing,” *Applied Measurement in Education*, vol. 5, no. 2, pp. 137–149, 1992.
- [46] K. d. O. Andrade, T. B. Pasqual, G. A. Caurin, and M. K. Crocomo, “Dynamic difficulty adjustment with Evolutionary Algorithm in games for rehabilitation robotics,” in *2016 IEEE International Conference on Serious Games and Applications for Health (SeGAH)*, pp. 1–8, IEEE, 2016.
- [47] O. Missura and T. Gärtner, “Player modeling for intelligent difficulty adjustment,” in *International Conference on Discovery Science*, pp. 197–211, Springer, 2009.
- [48] M. Jennings-Teats, G. Smith, and N. Wardrip-Fruin, “Polymorph: dynamic difficulty adjustment through level generation,” in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, pp. 1–4, 2010.
- [49] L. Sha, S. He, J. Wang, J. Yang, Y. Gao, Y. Zhang, and X. Yu, “Creating appropriate challenge level game opponent by the use of dynamic difficulty adjustment,” in *2010 Sixth International Conference on Natural Computation*, vol. 8, pp. 3897–3901, IEEE, 2010.
- [50] H. Fernández, K. Mikami, and K. Kondo, “Difficulty Adjustment Using Player’s Performance and Electroencephalographic Data,” *The Journal of the Society for Art and Science*, vol. 18, pp. 143–155, 2019.

- [51] J. F. Pinto, H. R. Carvalho, G. R. Chambel, J. Ramiro, and A. Goncalves, “Adaptive game-play and difficulty adjustment in a gamified upper-limb rehabilitation,” in *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*, pp. 1–8, IEEE, 2018.
- [52] P. Vlachogianni and N. Tselios, “Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review,” *Journal of Research on Technology in Education*, pp. 1–18, 2021.
- [53] E. Park, “User acceptance of smart wearable devices: An expectation-confirmation model approach,” *Telematics and Informatics*, vol. 47, p. 101318, 2020.
- [54] C.-C. Lin, “Exploring the relationship between technology acceptance model and usability test,” *Information Technology and Management*, vol. 14, no. 3, pp. 243–255, 2013.
- [55] A. Revythi and N. Tselios, “Extension of technology acceptance model by using system usability scale to assess behavioral intention to use e-learning,” *Education and Information technologies*, vol. 24, no. 4, pp. 2341–2355, 2019.
- [56] F. D. Davis, *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. PhD thesis, Massachusetts Institute of Technology, 1985.
- [57] J. Lu, C.-S. Yu, C. Liu, and J. E. Yao, “Technology acceptance model for wireless Internet,” *Internet research*, 2003.
- [58] D. Tao, T. Wang, T. Wang, T. Zhang, X. Zhang, and X. Qu, “A systematic review and meta-analysis of user acceptance of consumer-oriented health information technologies,” *Computers in Human Behavior*, vol. 104, p. 106147, 2020.
- [59] A. Granić and N. Marangunić, “Technology acceptance model in educational context: A systematic literature review,” *British Journal of Educational Technology*, vol. 50, no. 5, pp. 2572–2593, 2019.
- [60] W. M. Lim and D. H. Ting, “E-shopping: an Analysis of the Technology Acceptance Model,” *Modern Applied Science*, vol. 6, no. 4, p. 49, 2012.

- [61] T. Pikkarainen, K. Pikkarainen, H. Karjaluoto, and S. Pahnla, "Consumer acceptance of online banking: an extension of the technology acceptance model," *Internet research*, 2004.
- [62] S. Sukendro, A. Habibi, K. Khaeruddin, B. Indrayana, S. Syahrudin, F. A. Makadada, and H. Hakim, "Using an extended Technology Acceptance Model to understand students' use of e-learning during Covid-19: Indonesian sport science education context," *Heliyon*, vol. 6, no. 11, p. e05410, 2020.
- [63] H.-P. Shih, "Extended technology acceptance model of Internet utilization behavior," *Information & management*, vol. 41, no. 6, pp. 719–729, 2004.
- [64] J. Nielsen, *Usability engineering*. Morgan Kaufmann, 1994.
- [65] A. Lodhi, "Usability Heuristics as an assessment parameter: For performing Usability Testing," in *2010 2nd International Conference on Software Technology and Engineering*, vol. 2, pp. V2–256, IEEE, 2010.
- [66] E. Din, "9241-11. ergonomic requirements for office work with visual display terminals (vdts)–part 11: Guidance on usability," *International Organization for Standardization*, 1998.
- [67] T. Boren and J. Ramey, "Thinking aloud: Reconciling theory and practice," *IEEE transactions on professional communication*, vol. 43, no. 3, pp. 261–278, 2000.
- [68] M. C. Medlock, D. Wixon, M. Terrano, R. Romero, and B. Fulton, "Using the RITE method to improve products: A definition and a case study," *Usability Professionals Association*, vol. 51, pp. 1963813932–1562338474, 2002.
- [69] H. M. Salman, W. F. W. Ahmad, and S. Sulaiman, "Usability evaluation of the smartphone user interface in supporting elderly users from experts' perspective," *Ieee Access*, vol. 6, pp. 22578–22591, 2018.
- [70] P. B. Bokingkito Jr and L. T. Caparida, "Usability evaluation of a real-time water quality monitoring mobile application," *Procedia Computer Science*, vol. 197, pp. 642–649, 2022.

- [71] M. L. Lourenço, F. Lanhoso, and D. A. Coelho, “Usability Evaluation of Slanted Computer Mice,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 8, p. 3854, 2021.
- [72] J. Brooke, “SUS: a retrospective,” *Journal of usability studies*, vol. 8, no. 2, pp. 29–40, 2013.
- [73] J. R. Lewis, “The system usability scale: past, present, and future,” *International Journal of Human–Computer Interaction*, vol. 34, no. 7, pp. 577–590, 2018.
- [74] A. Di Nuovo, S. Varrasi, D. Conti, J. Bamsforth, A. Lucas, A. Soranzo, and J. McNamara, “Usability evaluation of a robotic system for cognitive testing,” in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 588–589, IEEE, 2019.
- [75] A. Vourvopoulos, A. L. Faria, K. Ponnampalani, and S. Bermudez i Badia, “RehabCity: design and validation of a cognitive assessment and rehabilitation tool through gamified simulations of activities of daily living,” in *Proceedings of the 11th conference on advances in computer entertainment technology*, pp. 1–8, 2014.
- [76] H. Cho, D. Powell, A. Pichon, L. M. Kuhns, R. Garofalo, and R. Schnall, “Eye-tracking retrospective think-aloud as a novel approach for a usability evaluation,” *International journal of medical informatics*, vol. 129, pp. 366–373, 2019.
- [77] J. Park and M. Zahabi, “A novel approach for usability evaluation of mobile applications,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 65, pp. 437–441, SAGE Publications Sage CA: Los Angeles, CA, 2021.
- [78] X. Liu and K. Lee, “Optimized Facial Emotion Recognition Technique for Assessing User Experience,” in *2018 IEEE Games, Entertainment, Media Conference (GEM)*, pp. 1–9, IEEE, 2018.
- [79] D. Canedo and A. J. Neves, “Facial expression recognition using computer vision: a systematic review,” *Applied Sciences*, vol. 9, no. 21, p. 4678, 2019.

- [80] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 97–115, 2001.
- [81] F. Kong, "Facial expression recognition method based on deep convolutional neural network combined with improved LBP features," *Personal and Ubiquitous Computing*, vol. 23, no. 3, pp. 531–539, 2019.
- [82] S. Nigam, R. Singh, and A. Misra, "Efficient facial expression recognition using histogram of oriented gradients in wavelet domain," *Multimedia tools and applications*, vol. 77, no. 21, pp. 28725–28747, 2018.
- [83] J. J. de Paula, M. R. Albuquerque, G. M. Lage, M. A. Bicalho, M. A. Romano-Silva, and L. F. Malloy-Diniz, "Impairment of fine motor dexterity in mild cognitive impairment and Alzheimer's disease dementia: association with activities of daily living," *Brazilian Journal of Psychiatry*, vol. 38, no. 3, pp. 235–238, 2016.
- [84] E. O. Lichtenberger, "General measures of cognition for the preschool child," *Mental retardation and developmental disabilities research reviews*, vol. 11, no. 3, pp. 197–208, 2005.
- [85] K. Lee, D. Jeong, R. C. Schindler, and E. J. Short, "SIG-Blocks: Tangible game technology for automated cognitive assessment," *Computers in Human Behavior*, vol. 65, pp. 163–175, 2016.
- [86] D. H. Jeong, *Distributed Wireless Sensor Network Systems: Theoretical Framework, Algorithms, and Applications*. PhD thesis, Case Western Reserve University, 2015.
- [87] D. Miranda and K. Lee, "Music Blocks: Audio-augmented Block Games for Play-Based Cognitive Assessment," in *2018 IEEE Games, Entertainment, Media Conference (GEM)*, pp. 375–381, IEEE, 2018.
- [88] K. Lee and D. Jeong, "Memorix: A tangible memory game using iSIG-blocks," in *2014 IEEE Games Media Entertainment*, pp. 1–8, IEEE, 2014.

- [89] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [90] C. G. Harris, M. Stephens, *et al.*, “A combined corner and edge detector,” in *Alvey vision conference*, vol. 15, pp. 10–5244, Citeseer, 1988.
- [91] S. Lloyd, “Least squares quantization in PCM,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [92] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [93] S. Suzuki *et al.*, “Topological structural analysis of digitized binary images by border following,” *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.
- [94] U. Ramer, “An iterative procedure for the polygonal approximation of plane curves,” *Computer graphics and image processing*, vol. 1, no. 3, pp. 244–256, 1972.
- [95] “Database Selection.” <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>.
- [96] A. Miyake and P. Shah, *Models of working memory: Mechanisms of active maintenance and executive control*. Cambridge University Press, 1999.
- [97] Y. Nakamura and P. Rudnicki, “Euler circuits and paths,” *Formalized Mathematics*, vol. 6, no. 3, pp. 417–425, 1997.
- [98] R. C. Prim, “Shortest connection networks and some generalizations,” *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.
- [99] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

- [100] “GUI Selection.” <https://www.askpython.com/python-modules/top-best-python-gui-libraries>.
- [101] C. E. Shannon, “A mathematical theory of communication,” *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.
- [102] P. Gao, Z. Li, and H. Zhang, “Thermodynamics-based evaluation of various improved Shannon entropies for configurational information of gray-level images,” *Entropy*, vol. 20, no. 1, p. 19, 2018.
- [103] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, “Textural features for image classification,” *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.
- [104] M. Partio, B. Cramariuc, M. Gabbouj, and A. Visa, “Rock texture retrieval using gray level co-occurrence matrix,” in *Proc. of 5th Nordic Signal Processing Symposium*, vol. 75, Cite-seer, 2002.
- [105] S. Guiaşu, “Weighted entropy,” *Reports on Mathematical Physics*, vol. 2, no. 3, pp. 165–179, 1971.
- [106] D. Bonchev and G. A. Buck, “Quantitative measures of network complexity,” in *Complexity in chemistry, biology, and ecology*, pp. 191–235, Springer, 2005.
- [107] M. S. McClendon *et al.*, “The complexity and difficulty of a maze,” in *Bridges: Mathematical Connections in Art, Music, and Science*, pp. 213–222, Bridges Conference, 2001.
- [108] “Easily calculate SUS Score.” <https://uxplanet.org/easily-calculate-sus-score-a464d753e5aa>.
- [109] “5 Ways to Interpret a SUS Score.” <https://measuringu.com/interpret-sus-score/>.
- [110] D. Hayosh, X. Liu, and K. Lee, “Woody: Low-Cost, Open-Source Humanoid Torso Robot,” in *2020 17th International Conference on Ubiquitous Robots (UR)*, pp. 247–252, IEEE, 2020.

- [111] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 94–101, IEEE, 2010.
- [112] N. Aifanti, C. Papachristou, and A. Delopoulos, “The MUG facial expression database,” in *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pp. 1–4, IEEE, 2010.
- [113] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, and J. Budynek, “The Japanese female facial expression (JAFFE) database,” in *Proceedings of third international conference on automatic face and gesture recognition*, pp. 14–16, 1998.
- [114] M. Valstar and M. Pantic, “Induced disgust, happiness and surprise: an addition to the mmi facial expression database,” in *Proc. 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect*, p. 65, Paris, France., 2010.
- [115] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.
- [116] H. Fleyeh and J. Roch, *Benchmark evaluation of HOG descriptors as features for classification of traffic signs*. Höskolan Dalarna, 2013.
- [117] E. Friesen and P. Ekman, “Facial action coding system: a technique for the measurement of facial movement,” *Palo Alto*, vol. 3, 1978.
- [118] J. Hamm, C. G. Kohler, R. C. Gur, and R. Verma, “Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders,” *Journal of neuroscience methods*, vol. 200, no. 2, pp. 237–256, 2011.
- [119] R. Baeza-Yates, B. Ribeiro-Neto, *et al.*, *Modern information retrieval*, vol. 463. ACM press New York, 1999.

- [120] B. R. Ilyas, B. Mohammed, M. Khaled, A. T. Ahmed, and A. Ihsen, "Facial expression recognition based on dwt feature for deep cnn," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 344–348, IEEE, 2019.
- [121] H. Boubenna and D. Lee, "Feature selection for facial emotion recognition based on genetic algorithm," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 511–517, IEEE, 2016.
- [122] N. Gopalan, S. Bellamkonda, and V. S. Chaitanya, "Facial Expression Recognition Using Geometric Landmark Points and Convolutional Neural Networks," in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1149–1153, IEEE, 2018.
- [123] A. Majumder, L. Behera, and V. K. Subramanian, "Automatic facial expression recognition system using deep network-based data fusion," *IEEE transactions on cybernetics*, vol. 48, no. 1, pp. 103–114, 2016.
- [124] A. Barman and P. Dutta, "Facial expression recognition using distance signature feature," in *Advanced Computational and Communication Paradigms*, pp. 155–163, Springer, 2018.
- [125] S. Xu, Y. Cheng, Q. Lin, and J. Allebach, "Emotion Recognition Using Convolutional Neural Networks," *Electronic Imaging*, vol. 2019, no. 8, pp. 402–1, 2019.
- [126] M. Verma, S. K. Vipparthi, and G. Singh, "Hinet: Hybrid inherited feature learning network for facial expression recognition," *IEEE Letters of the Computer Society*, vol. 2, no. 4, pp. 36–39, 2019.
- [127] Y. Jiao, X. Jia, and J. Zhao, "Facial Expression Recognition Method Based on Difference Center-Symmetric Local Directional Pattern," in *2019 International Conference on Computer, Network, Communication and Information Systems (CNCI 2019)*, Atlantis Press, 2019.
- [128] A. Barman and P. Dutta, "Facial expression recognition using distance and texture signature relevant features," *Applied Soft Computing*, vol. 77, pp. 88–105, 2019.

- [129] N. Meeki, A. Amine, M. A. Boudia, and R. M. Hamou, “Deep Learning for Non Verbal Sentiment Analysis: Facial Emotional Expressions,” tech. rep., EasyChair, 2020.
- [130] C. V. R. Reddy, U. S. Reddy, and K. V. K. Kishore, “Facial Emotion Recognition Using NLPCA and SVM Facial Emotion Recognition Using NLPCA and SVM,” *Traitement du Signal*, vol. 36, no. 1, pp. 13–22, 2019.
- [131] A. Fathallah, L. Abdi, and A. Douik, “Facial expression recognition via deep learning,” in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 745–750, IEEE, 2017.
- [132] S. Minaee and A. Abdolrashidi, “Deep-emotion: Facial expression recognition using attentional convolutional network,” *arXiv preprint arXiv:1902.01019*, 2019.
- [133] A. Barman and P. Dutta, “Influence of shape and texture features on facial expression recognition,” *IET Image Processing*, vol. 13, no. 8, pp. 1349–1363, 2019.
- [134] P.-L. Carrier, A. Courville, I. J. Goodfellow, M. Mirza, and Y. Bengio, “FER-2013 face database,” *Universit de Montral*, 2013.
- [135] J.-Y. Son, J.-H. Lee, J.-Y. Kim, J.-H. Park, and Y.-H. Lee, “RAFD: Resource-aware fault diagnosis system for home environment with smart devices,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1185–1193, 2012.
- [136] D. Aneja, A. Colburn, G. Faigin, L. Shapiro, and B. Mones, “Modeling stylized character expressions via deep learning,” in *Asian conference on computer vision*, pp. 136–153, Springer, 2016.
- [137] S. Kim and H. Kim, “Deep Explanation Model for Facial Expression Recognition Through Facial Action Coding Unit,” in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 1–4, IEEE, 2019.
- [138] Z. Fei, E. Yang, D. D.-U. Li, S. Butler, W. Ijomah, X. Li, and H. Zhou, “Deep convolution network based emotion analysis towards mental health care,” *Neurocomputing*, vol. 388, pp. 212–227, 2020.

- [139] D. Lundqvist, A. Flykt, and A. Öhman, “The Karolinska directed emotional faces (KDEF),” *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, vol. 91, no. 630, pp. 2–2, 1998.
- [140] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, “Deep Convolutional Neural Network for Robust Facial Emotion Recognition,” in *2019 IEEE International Symposium on INnovations in Intelligent Systems and Applications (INISTA)*, pp. 1–6, IEEE, 2019.
- [141] C. Wang, K. Lu, J. Xue, and Y. Yan, “Dense Attention Network for Facial Expression Recognition in the Wild,” in *Proceedings of the ACM Multimedia Asia*, pp. 1–6, MMA-sia, 2019.
- [142] T. Doi, H. Makizako, K. Tsutsumimoto, R. Hotta, S. Nakakubo, K. Makino, T. Suzuki, and H. Shimada, “Combined effects of mild cognitive impairment and slow gait on risk of dementia,” *Experimental Gerontology*, vol. 110, pp. 146–150, 2018.
- [143] A. Association, “2019 Alzheimer’s disease facts and figures,” *Alzheimer’s & Dementia*, vol. 15, no. 3, pp. 321–387, 2019.
- [144] J. Rasmussen and H. Langerman, “Alzheimer’s disease—why we need early diagnosis,” *Degenerative Neurological and Neuromuscular Disease*, vol. 9, p. 123, 2019.
- [145] A. Nahar, A. Lukose, R. G. Ramesh, S. Purokayastha, S. H. Kadiveti, S. P. Thangaraju, and N. P. Rao, “Alzheimer’s Dementia: An Overview,” *Journal of the Indian Institute of Science*, vol. 97, no. 4, pp. 591–602, 2017.
- [146] C. Ledig, A. Schuh, R. Guerrero, R. A. Heckemann, and D. Rueckert, “Structural brain imaging in Alzheimer’s disease and mild cognitive impairment: biomarker analysis and shared morphometry database,” *Scientific reports*, vol. 8, no. 1, pp. 1–16, 2018.
- [147] S. P. Caminiti, T. Ballarini, A. Sala, C. Cerami, L. Presotto, R. Santangelo, F. Fallanca, E. G. Vanoli, L. Gianolli, S. Iannaccone, *et al.*, “FDG-PET and CSF biomarker accuracy in prediction of conversion to different dementias in a large multicentre MCI cohort,” *NeuroImage: Clinical*, vol. 18, pp. 167–177, 2018.

- [148] D. R. Roalf, P. J. Moberg, S. X. Xie, D. A. Wolk, S. T. Moelter, and S. E. Arnold, “Comparative accuracies of two common screening instruments for classification of Alzheimer’s disease, mild cognitive impairment, and healthy aging,” *Alzheimer’s & Dementia*, vol. 9, no. 5, pp. 529–537, 2013.
- [149] S. Pendlebury, S. Klaus, M. Mather, M. De Brito, and R. Wharton, “Routine cognitive screening in older patients admitted to acute medicine: abbreviated mental test score (AMTS) and subjective memory complaint versus Montreal Cognitive Assessment and IQ-CODE,” *Age and ageing*, vol. 44, no. 6, pp. 1000–1005, 2015.
- [150] J. B. Sherman, A. Chatterjee, R. D. Urman, D. J. Culley, G. J. Crosby, Z. Cooper, H. Javedan, D. L. Hepner, and A. M. Bader, “Implementation of routine cognitive screening in the pre-operative assessment clinic,” *A&A Practice*, vol. 12, no. 4, pp. 125–127, 2019.
- [151] T. N. Tombaugh and N. J. McIntyre, “The mini-mental state examination: a comprehensive review,” *Journal of the American Geriatrics Society*, vol. 40, no. 9, pp. 922–935, 1992.
- [152] E. C. Rosca, L. Albarqouni, and M. Simu, “Montreal Cognitive Assessment (MoCA) for HIV-Associated Neurocognitive Disorders,” *Neuropsychology review*, pp. 1–15, 2019.
- [153] S. Freitas, S. Batista, A. C. Afonso, M. R. Simões, L. de Sousa, L. Cunha, and I. Santana, “The Montreal Cognitive Assessment (MoCA) as a screening test for cognitive dysfunction in multiple sclerosis,” *Applied Neuropsychology: Adult*, vol. 25, no. 1, pp. 57–70, 2018.

APPENDIX A

COLOR GAMES

This appendix presents the design and fabrication of the color cubes. The color cubes can serve as another type of tangible object used in e-Cube to form easier games compared to the geometric cubes. The color detection approach for identifying these colors was also briefly described.

Fig. A.1a displays the six distinctive colors, red, yellow, blue, green, gray, and white painted on the surfaces of a color cube and Fig. A.1b displays the nine fabricated color cubes. To identify the six colors regardless of the illuminating condition, the system 1) converts all the pixels inside the area into HSV color space; 2) extracts the pixels that are within the six desired color ranges specifying the low and upper bounds of the colors defined in A.1; and 3) determines whether or not the color is presented based on the ratio of the extracted pixels and the entire designated region. This detection is similar to the black detection used for the geometric cubes.

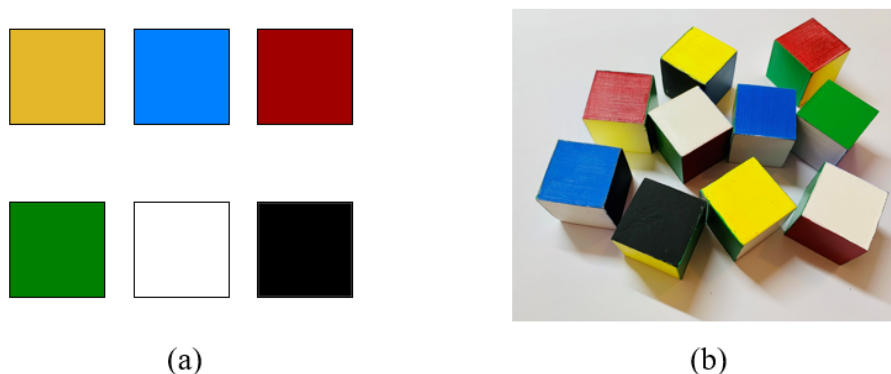


Figure A.1: Color cubes: (a) six colors and (b) the fabricated cubes.

Table A.1: Lower and upper bounds of the six colors in HSV color space.

Color	Lower Bound (H, S, V)	Upper Bound (H, S, V)
Red	(0, 160, 100) (170, 160, 160)	(30, 130, 150) (180,255,255)
Yellow	(25, 160, 150)	(32, 255, 255)
Blue	(100, 100, 150)	(120, 180, 255)
Green	(30, 90, 80)	(70, 200, 165)
Black	(0, 0, 0)	(50, 200, 120)
White	(15, 0, 230)	(35, 100, 255)

APPENDIX B

DATA CONSTRUCTION AND MANAGEMENT

This appendix presents the Python code for building the e-Cube database, inserting data into the database, and querying the data.

B.1 Database Construction

The database for the e-Cube system can be built using the following code:

```
import mysql.connector

db = mysql.connector.connect (
host = 'localhost',
user = 'root',
password = '',
database = 'ecube_preliminary')

mycursor = db.cursor()

mycursor.execute("CREATE DATABASE ecube_preliminary")

mycursor.execute("CREATE TABLE User (username VARCHAR(20)
NOT NULL, date_of_birth VARCHAR(20) NOT NULL, gender
ENUM('M', 'F', 'O') NOT NULL, game_version ENUM('0', '1', '2')
NOT NULL, id int PRIMARY KEY NOT NULL)")

mycursor.execute("CREATE TABLE Behavior (userId int NOT NULL,
FOREIGN KEY(userId) REFERENCES User(id) ON DELETE CASCADE,
```

```
current_visiting_index smallint UNSIGNED NOT NULL, game_type
VARCHAR(30) NOT NULL, item_number smallint UNSIGNED NOT NULL,
detected_result VARCHAR(2000) NOT NULL, time_s float(3)
NOT NULL) ")
```

```
mycursor.execute("CREATE TABLE Accuracy (userId int NOT NULL,
FOREIGN KEY(userId) REFERENCES User(id) ON DELETE CASCADE,
current_visiting_index smallint UNSIGNED NOT NULL, game_type
VARCHAR(30) NOT NULL, item_number smallint UNSIGNED NOT NULL,
answer VARCHAR(2000) NOT NULL, correctness BIT(1) NOT NULL,
current_difficulty float(3), maximal_scored_difficulty float(3),
ceiling_difficulty float(3), correct_pt_item_number smallint
UNSIGNED, multi_pt_item_number smallint UNSIGNED,
wrong_pt_item_number smallint UNSIGNED) ")
```

```
mycursor.execute("CREATE TABLE Visiting (userId int NOT NULL,
FOREIGN KEY(userId) REFERENCES User(id) ON DELETE CASCADE,
total_visiting_times smallint UNSIGNED NOT NULL, login_time
datetime NOT NULL, game_order VARCHAR(1000) NOT NULL, finish
BIT(1) NOT NULL) ")
```

```
db.commit()
```

B.2 Writing Commands

The commands shown in this section present how the data is written into the constructed database during the games.

```
from datetime import datetime
```

```
mycursor.execute("INSERT INTO User (date_of_birth, gender,  
game_version, id) VALUES (%s, %s, %s, %s)", (user_birth,  
user_gender, user_version, unique_id))
```

```
mycursor.execute("INSERT INTO Visiting (userId,  
total_visiting_times, login_time, game_order, finish) VALUES  
(%s, %s, %s, %s, %s)", (unique_id, 1, datetime.now(), str(order),  
0))
```

```
mycursor.execute("INSERT INTO Behavior(userId,  
current_visiting_index, game_type, item_number, detected_result,  
time_s) VALUES (%s, %s, %s, %s, %s, %s)", (user_id, visit,  
'assembly', str(self.assem_test_i-2), str(self.result),  
str(self.current_time)))
```

```
db.commit()
```

```
mycursor.execute("INSERT INTO Accuracy (userId,  
current_visiting_index, game_type, item_number, answer,  
correctness) VALUES (%s, %s, %s, %s, %s, %s)", (user_id, visit,  
'assembly', str(self.assem_test_i-2),  
str(game_info.answer_a[self.assem_test_i]), correct))
```

```
db.commit()
```

```
mycursor.execute("UPDATE Visiting SET finish = 1 where  
userId = '" + user_id + "' ORDER BY total_visiting_times DESC")
```



```
db.commit()
```

B.3 Data Query

Data can be queried by knowing the assigned ID number of a participant. For example, if the ID number is 5609, the process of data query can be:

```
mycursor.execute("SELECT id FROM User where id = '" + '5609' +  
" ' ")
```

```
data = mycursor.fetchone()
```

```
db.commit()
```

```
if data is not None:
```

```
    mycursor.execute("SELECT id FROM User where id = '" +  
    user_UIN + " ' ")
```

```
    for x in mycursor:
```

```
        user_id = str(x[0])
```

```
    mycursor.execute("SELECT MAX(visit_n) FROM Visit where  
    userId = '" + user_id + " ' ")
```

```
    for v in mycursor:
```

```
        visit = str(v[0])
```

```
    mycursor.execute("SELECT * FROM Behavior where userId = '"  
    + user_id + " ' and visit = '" + visit + " ' ")
```

```
    mycursor.execute("SELECT * FROM Accuracy where userId = '"  
    + user_id + " ' and visit = '" + visit + " ' ")
```

```
    for x in mycursor:
```

```
        print (x)
```

APPENDIX C

DATA ANALYSIS USING WAIS-IV RAW SCORES

Section 5.4.2.2 in this dissertation investigated the relationships between the e-Cube games and WAIS-IV subtests by analyzing their score correlations, where the correlations were computed using the WAIS-IV scaled scores. This appendix presents the correlations between the e-Cube scores and WAIS-IV raw scores as well as the differences in correlations using raw and scaled scores.

Tables C.1 and C.2 show the correlations between the e-Cube scores and WAIS-IV raw scores for Fixed Group and Adaptive Group, respectively. Compared to Tables 5.9 and 5.10 using the scaled scores, the two results using the raw scores show similar correlation trends. However, some differences still exist and are described in the following. The coefficient of the correlation between e-Cube^A and BD in Fixed Group was slightly higher than in Adaptive Group using the scaled scores, while they were the same using the raw scores. This further supported that e-Cube^A was related to the performance of BD in both groups. Besides this, significant correlations were also found between e-Cube^{M1} and BD in Fixed Group as well as e-Cube^A and MR in Adaptive Group using the raw scores. However, the two relationships did not match the design expectations (Table 5.1).

Table C.1: Correlations with degrees of freedom 37 between the scores of fixed e-Cube and raw scores of the WAIS-IV subtests, shown as $r(p)$.

		WAIS Subtests Raw Scores		
		BD	DS	MR
Scores	Fixed Games			
	e-Cube ^A	0.51 (0.001)	-0.11 (0.501)	0.06 (0.719)
	e-Cube ^S	0.33 (0.047)	-0.29 (0.079)	0.09 (0.602)
	e-Cube ^{M1}	0.33 (0.049)	0.13 (0.441)	0.28 (0.099)
	e-Cube ^{M2}	0.23 (0.170)	0.28 (0.094)	0.17 (0.320)
	e-Cube ^P	0.48 (0.002)	0.00 (0.993)	0.08 (0.659)
	e-Cube ^Z	0.28 (0.090)	-0.02 (0.888)	0.14 (0.395)

Table C.2: Correlations with degrees of freedom 41 between the scores of adaptive e-Cube and raw scores of the WAIS-IV subtests, shown as $r(p)$.

		WAIS Subtests Raw Scores		
		BD	DS	MR
Scores	Adaptive Games			
	e-Cube ^A	0.51 (< 0.001)	0.19 (0.228)	0.33 (0.037)
	e-Cube ^S	0.14 (0.383)	0.21 (0.189)	0.51 (< 0.001)
	e-Cube ^{M1}	0.36 (0.019)	0.21 (0.180)	0.11 (0.507)
	e-Cube ^{M2}	0.02 (0.878)	0.50 (< 0.001)	0.12 (0.448)
	e-Cube ^P	0.44 (0.004)	-0.04 (0.821)	0.43 (0.005)
	e-Cube ^Z	0.27 (0.092)	0.06 (0.703)	0.08 (0.637)