

**ROBOT ASSISTED 3D BLOCK BUILDING TO AUGMENT SPATIAL
VISUALIZATION SKILLS IN CHILDREN – AN EXPLORATORY
STUDY**

by
Amama Mahmood

A thesis submitted to The Johns Hopkins University in conformity
with the requirements for the degree of MSE Robotics

Baltimore, Maryland
May, 2020

© 2020 Amama Mahmood
All rights reserved

Abstract

The unique social presence of robots can be leveraged in learning situations to increase comfortability and engagement of kids, while still providing instructional guidance. When and how to interfere to provide feedback on their mistakes is still not fully clear. One effective feedback strategy used by human tutors is to implicitly inform the students of their errors rather than explicitly providing corrective feedback. This essay explores if and how a social robot can be utilized to provide implicit feedback to a user who is performing spatial visualization tasks. We explore impact of implicit and explicit feedback strategies on user's learning gains, self-regulation and perception of robot during 3D block building tasks in one-on-one child-robot tutoring.

We demonstrate a realtime system that tracks the assembly of a 3D block structure using a RealSense[®] RGB-D camera. The system allows three control actions: *Add*, *Remove* and *Adjust* on 2×4 and 2×4 blocks (similar to Lego[®]) of four basic colors (red, blue, green and yellow) to manipulate the 3D structure in the play area. 3D structures can be authored in the *Learning mode* for system to record, and tracking enables the robot to provide selected feedback in the *Teaching mode* depending on the type of mistake made by the user. Proposed perception system is capable of detecting five types of mistakes i.e., mistake in: shape, color, orientation, level from base and position of the block.

The feedback provided by the robot is based on mistake made by the user. Either implicit or explicit feedback, chosen randomly, is narrated by the robot. Various feedback statements are designed to implicitly inform the user of the mistake made.

Two robot behaviours have been designed to support the effective delivery of feedback statements. Nodding is used in conjunction with continuers i.e., *Go on, Continue, Good, Hmm and Right* when the user action results in a correct outcome. Whereas, referential gaze is employed along with the requisite feedback statement whenever the user commits a mistake during the assembly task.

We conducted an exploratory study to evaluate our robot assisted 3D block building system to augment spatial visualization skills with one participant. We found that the system was easy to use. The robot was perceived as trustworthy, fun and interesting. Intentions of the robot are communicated through feedback statements and its behaviour i.e., nodding and referential gaze. Our goal is to explore that the suggestion of mistakes in implicit ways can help the users self-regulate and scaffold their learning processes. However, we do not have enough evidence to support this from our exploratory study.

Furthermore, we discuss shortcomings of our system, compare it to few existing systems, discuss design implications and ways to improve it as future work. The observations from the study with one elementary-aged study are contribution towards our future endeavours in the field of social robotics and education.

Thesis Readers

Dr. Chien-Ming Huang (Primary Advisor)
Assistant Professor
Department of Computer Science
Johns Hopkins University

*Dedicated to my parents, siblings,
my aunt, my cousins and my pets.*

*It would not have been possible without their love,
support and prayers.*

*And finally to all the people who are doing their part
to keep everyone safe in the COVID-19 crisis.*

Acknowledgements

I wish to express my sincere appreciation to my advisor, Dr. Chien-Ming Huang for his guidance, time and encouragement throughout the process. I would also like to acknowledge Jindan Huang and Claudia Juliana Moncaliano for their assistance in implementation of the system. I wish to give due credit to my young cousin without whom I would not have been able to do this exploratory study. In the end, I would like to thank Fulbright Program for funding my masters degree and making it possible for me to be at the Johns Hopkins University.

Contents

Abstract	ii
Dedication	iv
Acknowledgements	v
Contents	vi
List of Figures	ix
Chapter 1 Introduction	1
Chapter 2 Background	4
Spatial Reasoning	4
Definitions and Categorization	4
Development of Spatial Reasoning in Early Years	6
Role of Spatial Reasoning in Mathematics	8
Ways to Augment Spatial Reasoning	10
Child-Robot Tutoring	11
Child-Robot Tutoring and Spatial Reasoning	12
Feedback Strategies	13

Chapter 3	System Overview	18
	Perception	18
	Setup	19
	Different Modes	20
	Representation of Play Area and Blocks	20
	Processing Pipeline	25
	Scanning Control Boxes	25
	Scanning Play Area	27
	Types of Errors Detected	28
	Evaluation and Limitations	32
	Feedback	35
	Robot Behaviour	36
Chapter 4	Exploratory Study	38
	Participant	38
	Task	38
	Procedure	39
	Results and Discussion	40
	Pre and Post-test	40
	Video Analysis	40
	Post-study Interview	43
	Design Implications and Suggested Improvements	45
	Conclusions and Future Work	48
	References	50

Curriculum vitae 55

List of Figures

Figure 1-1	System Setup	3
Figure 2-1	2x2 topology of spatial reasoning categories	6
Figure 3-1	Attributes of <i>Assembly_Graph</i> and <i>Block</i> explained	24
Figure 3-2	Processing pipeline	26
Figure 3-3	Simple 3D structures	30
Figure 3-4	Complex 3D structures	31
Figure 3-5	Example of occlusion	33
Figure 3-6	Example of confusion	34

Chapter 1

Introduction

Good feedback practices by tutors in one-on-one sessions are essential for self-regulated learning. For designing a good feedback practice, one needs to consider many factors. One of many principles to think about design and evaluation of self-created feedback procedures, is that feedback can be provided by a teacher, peer or tutor [1]. There is a discussion that a good feedback strategy would inspire development of self-assessment. Feedback comes in various forms and modes [2] i.e., immediate vs. delayed feedback timing, single-try vs. multiple-try, adaptive vs. non-adaptive and implicit vs. explicit feedback. In human-human interactions, experienced tutors tend to provide indirect feedback to bring the attention of students to an error rather than giving corrective explicit feedback [3]. Expert tutors also employ subtle techniques like delaying affirmation or expressing short hesitation when a student inquires if he is doing alright, thus, hinting him about a mistake in the current step [4]. There are other strategies like asking a leading question or subtly suggesting the student to redirect. On the other hand, it is argued that providing only implicit feedback can be detrimental. If a student spends too much time guessing for a solution or redirecting himself, it would be difficult for him to trace the path he took to solution [5]. Therefore, the tutoring strategy should consider a trade-off between implicit and explicit feedback to allow room for the student to explore and learn from his mistakes, yet not be confused or stuck at an impasse.

The argument has frequently been made that one way to achieve the effectiveness of human tutoring in educational contexts is by using intelligent tutoring systems, embodied agents and robots [6]. A recent survey of long-term human-robot interaction (HRI) highlighted the increasing popularity of using social robots in educational environments [7]. Studies recording attitude of children towards robot show positive outcomes [8]. Although, there are a lot of unexplored topics in this field. While prior research has attempted and succeeded to develop emotionally intelligent and personalized robot tutors, most of these employ explicit feedback methods such as informing the student that answer is wrong, or some step is wrong or giving hints on how to solve the problem [9][10]. To the best of our knowledge, no robot tutoring system has been used to study the effectiveness of implicit strategies to provide feedback to students in one-on-one tutoring sessions. Moreover, there is little discussion on spatial reasoning skills [11] in robot tutoring while it has been established that such skills augment geometric and mathematical understanding of students [12]. Most of the studies are limited to teaching languages (e.g., [9]) and solving mathematical problems (e.g., [10]).

We contextualized our investigation into a one-on-one tutoring interaction. The setup is shown in Figure 1-1. The setup consists of the following:

1. Perception system: To track actions of the participant.
2. Intelligent Robot Tutor: Acts as a tutor to provide feedback based on actions of the participant.
3. 3D blocks: Serves as the playground for the participant to solve designed spatial reasoning task.

We collected data from one participant to explore the impact of implicit and explicit feedback on self-efficacy and learning gain, and her perception of such robot tutor while working on a spatial reasoning task. Analysis of data has resulted in useful

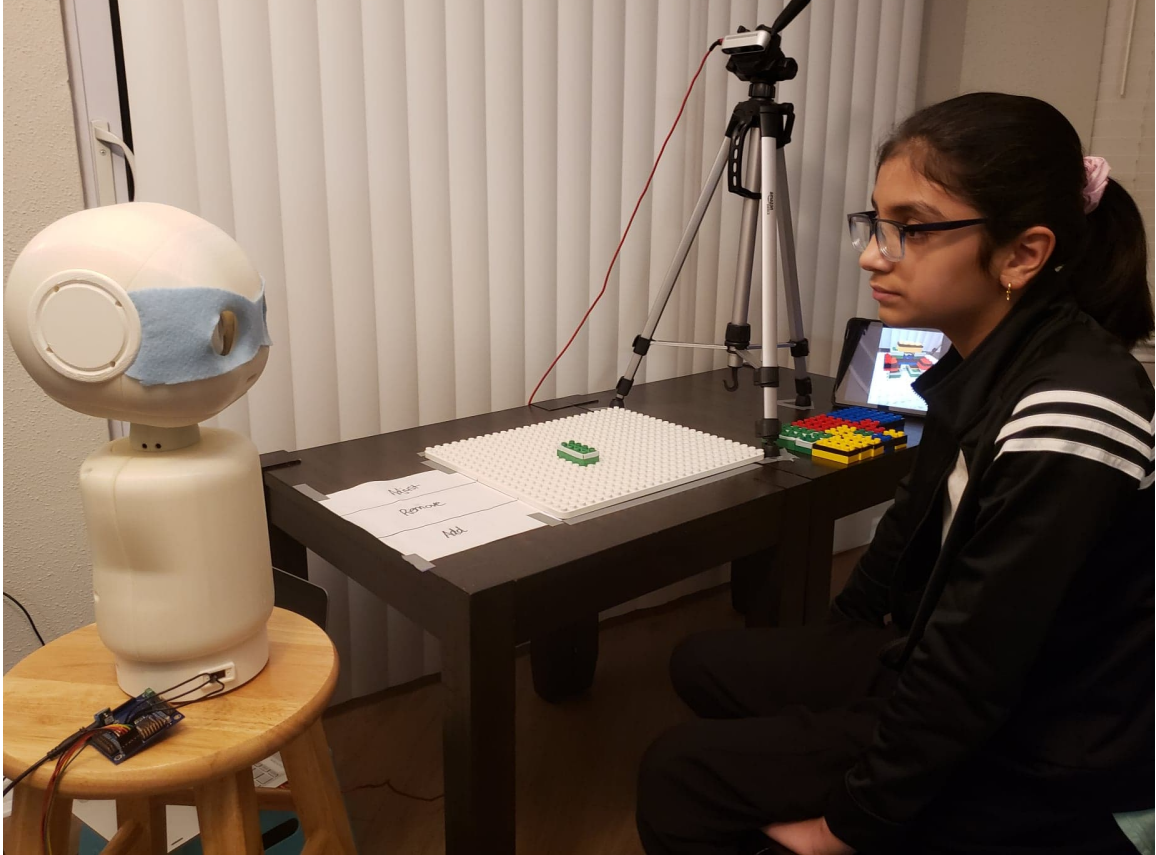


Figure 1-1. System Setup

design implications for future research. We believe that this study opens an interesting research space in robot tutoring. Future studies about tradeoff between implicit and explicit feedback, effectiveness of various types of implicit feedback, its personalization and right timing for intervention etc. will follow.

In the next chapter, we review prior work in field of social robotics for education, feedback practices in tutoring, and development, importance and augmentation of spatial reasoning in early ages. We give background to our study. This review is followed by a description of our system and how the exploratory study is designed along with evaluation. We then discuss our findings and their implications for future work, concluding with a summary of our contributions.

Chapter 2

Background

Spatial Reasoning

Definitions and Categorization

Spatial reasoning has always been of paramount importance for human action and thought. Various verbs e.g., locating, orienting, transforming, visualizing and navigating. are used to indicate spatial reasoning. Thus, from navigating in 3D world to mental manipulation of objects in our surroundings before physically moving them around falls under umbrella of spatial reasoning. There is a considerable debate on the relationships among “visualization”, “visual-spatial reasoning”, and “spatial reasoning”. These terms are used interchangeably by some and some present differences. In the light of these uncertainties, Linn and Peterson (1985) [13] suggested an approach to divide spatial skills into three broad categories:

1. Spatial Perception: Spatial perception is the ability to determine spatial relationships with respect to the orientation one's own body while ignoring distractions. An example test is water level task that requires participants to draw a horizontal line in titled water bottle [14].
2. Mental Rotation: Mental rotation is the ability to mentally rotate a two or three-dimensional objects rapidly and accurately. Shepard and his colleagues

[15] [16] administered tasks to measure the speed of mental rotation.

3. Spatial Visualization: Spatial visualization is the ability to perform complicated, multi-step manipulations of spatially presented information to complete a task. Such tasks may involve the similar underlying processes as in spatial perception and mental rotation but may have multiple possible correct solutions. Three dimensional block building is an example of such tasks.

McGee [17] explains that there are two main factors of spatial ability:

1. Spatial Visualization: Spatial visualization is the ability to imagine manipulation, rotation, twisting or inverting objects without reference to one's self.
2. Spatial Orientation: Spatial orientation is associated with one's ability to imagine the view of an object from different perspectives and directions.

In his book, Davis [18] uses the topological framework of spatial skills proposed by Uttal [19]. The 2×2 categorization scheme is based on two key dimensions of spatial reasoning: static versus dynamic and intrinsic versus extrinsic skills as summarized in Figure 2-1. As per his analysis, the skills and tasks did not always fit into one category or the other. Skills shifted from one category to another depending on the interpretation of the task at hand or ways in which a single task is performed specially when young children's spatial reasoning is concerned.

Regardless of these categorization, some sort of mental imagery and manipulation is involved in spatial reasoning skills. For the scope of this study, spatial visualization component of spatial reasoning is considered which is the ability to imagine an object from one's perspective, mentally rotate it and imagine the rotated image from the same perspective correctly to complete a complicated multi-step task [13].

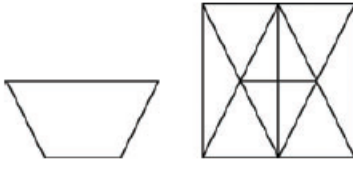



	Intrinsic Specification of object	Extrinsic Relation among objects or relation of object to a frame of reference
Static Object/frame of reference remains stationary	<p><i>Intrinsic-Static</i> Perceive objects while ignoring distractors.</p>  <p>Sample task: Embedded Figures Test <i>Where does the shape on the left appear in the image on the right?</i></p>	<p><i>Extrinsic-Static</i> Describe the spatial position in reference to frame.</p>  <p>Sample task: Water-Level Test <i>The glass jar has some water in it. The jar has been tilted. Draw a line to show how the water would look.</i></p>
Dynamic Object or perspective is transformed	<p><i>Intrinsic-Dynamic</i> Manipulate or mentally transform an object.</p>  <p>Sample task: Mental Rotation Test <i>Which of the three images on the right is the same as the one on the left if rotated?</i></p>	<p><i>Extrinsic-Dynamic</i> Visualize the relation among moving objects or from a different vantage point.</p>  <p>Sample task: Three Mountains Task <i>What does the teddy bear see?</i></p>

Figure 2-1. 2x2 topology of spatial reasoning categories [18].

Development of Spatial Reasoning in Early Years

Development of spatial reasoning starts in early years of human life. Different researchers have presented their theories about timeline of development of spatial reasoning skills in children [11]. Piaget and Inhelder [20] indicated that children go through three stages during this development of their cognitive spatial ability: preoperational stage, concrete operational stage and formal operational stage. They indicated that children under six years of age can locate objects with respect to themselves and are aware of topological spatial relationships. They are in preoperational stage since they understand concepts like separation, proximity and open/close. In concrete operational stage, children of ages 7-9 years develop a cognitive map with

a fixed frame of reference helping them with external view and orientation such as right/left and before/behind. At age of 11, as children enter the last stage i.e., formal operational stage, they understand concept of fixed positions relative to each other that assist them in understanding concepts such as proportional reduction of scale, and estimating straight-line relative distances.

However, later around 2000, Huttenlocher and Newcombe [21] suggest that spatial reasoning develops even earlier, as early as at age of 6 months. Stages presented by them are summarized as follows [11]:

1. Infants at age of 6 months are able to use dead reckoning skills to understand location of objects around. This helps them in keeping track of moving object's direction.
2. At age of 12 months, babies can find hidden stimuli with help of understanding of distance.
3. At age of 18 months, they can navigate easy routes.
4. Using the distance information from landmarks, kids can define locations at age of 2 years. Piaget had suggested that does not happen until they are nine or ten.
5. Kids at age of 3 are able to use simple maps and models.
6. If encouraged to play with maps and tools, kids can fully develop their spatial reasoning skills by the age of nine or ten.

Frick and Wang [22] experiments suggest that 14 month olds with some prior experience can activate mental rotation ability. Younger kids manifest the spatial reasoning skills which can be channeled by mental rotation exercises at early age to further strengthen this ability. However, at this young age, children can only continue an already started rotation.

Research on spatial transformations has indicated complex tasks like mental paper folding task emerges at age of 5.5 years and continue to improve through early elementary school [23]. Object based spatial transformations require children to possess spatial manipulation ability of mental imagery. This occurs for kids aged between 7 and 8 years. Egocentric perspective transformations develop later at age of 8 years rather than 7, than object-based transformations [24].

Role of Spatial Reasoning in Mathematics

Over 60 years ago, a report of a National Science Foundation (NSF) advisory panel, Scientific Careers, was published [25] which emphasized on critical role of spatial ability. It was characterized as an individual difference that affected learning the advanced scientific-technical concepts for outstanding performance in STEM (science, technology, engineering and mathematics). Despite that this fact was established as early as 1957, spatial reasoning skills are not included in curriculum and instruction in educational settings even in STEM domains. Various studies and projects followed further solidifying the positive effect of spatial reasoning skills on success in STEM fields [26].

There are several theories of quantitative reasoning on how spatial reasoning skills may support success in mathematics. Dehaene and his colleagues suggests that quantitative reasoning comprises of two core systems of numbers which tap different neural networks in the human brain: one being approximate and non-symbolic and the other is precise and symbolic. It is suggested that the first core shares neural activity with spatial reasoning skills and the second core is more associated with exact counting and symbolic mathematical operations [27]. As early schooling results in the two core systems of numbers merging, spatial skills' impact on mathematics achievement via the second symbolic system of number will increase significantly. Symbolic and

non-symbolic numerical thinking will mutually enhance one and other as these are taught over time [28]. Existing evidence suggests that the spatial skills, if polished in early elementary school curriculum, will provide a strong foundation for mathematics achievement in elementary school and beyond.

Various studies imply that in elementary school students, spatial skills are known to support understanding of geometry [29], word problem solving and application of more complex and complicated computation strategies [30], especially in girls [31]. Longitudinal studies have also shown that early spatial skills predict later success in mathematics. For instance, spatial reasoning skills measured in first-grade girls are crucial predictors of fifth-grade analytical math reasoning. First-grade assessments included spatial skills, verbal skills, addition/subtraction skills, and frequency of choice of a decomposition or retrieval strategy on the addition/subtraction problems. In fifth grade, girls were given an arithmetic fluency test, a mental rotation spatial task, and a numeric and algebra math reasoning test [32]. Similarly, spatial visualization predicted as early as in kindergarten reflects arithmetic skills in third graders. Linguistic and spatial skills can improve arithmetic development by enhancing children's number-related knowledge [33]. Even at early ages, both mental rotation and spatial visualization skills have depicted associations with numeric operations and addition and subtraction computational skills in grade 1,2,3 and even in kindergarten. A 32-week-teacher-led spatial reasoning intervention in K-2 classrooms highlighted the importance of assisting development of spatial visualization in young kids as part of early mathematics instruction [34]. Another longitudinal study investigated the development of spatial reasoning skills in 304 elementary school children as they progressed from grade 2 to 4 which outlines effects of socioeconomic status, verbal working memory and gender [35].

To summarize, spatial reasoning skills contribute positively to development of mathematical understanding at early ages and are pertinent to achievements in STEM fields

later on.

Ways to Augment Spatial Reasoning

Spatial reasoning can be augmented using various activities and games. Following are some ways suggested to improve spatial skills:

1. Use spatial language and gestures in everyday instructions: Using spatial language such as inside, outside, left, right, behind and on top of, assists babies in learning spatial relations better [36]. For example, sentences to describe scenes: The car is across the street under the tree. Asking kids to repeat after you helps further. Hand-gestures along with spatial language can help kids learn better [37].
2. Teach how to visualize using mind's eye: Visualization is an essential tool for spatial learning. If kids are instructed to visualize a problem while solving it, they would perform much better. For instance, in an experiment, when a ball is dropped even through a twisted tube, preschooler would tend to think it falls right under. But when young kids are distracted from their gravity bias by instructing them to visualize the path of the ball, more students get the right answer [38].
3. Two-dimensional Puzzles and other activities: Various puzzles and matching games can be used to enhance spatial reasoning skills of kids. For instance, tangram, an ancient Chinese puzzle that consists of 7 pieces and can be rearranged into many different shape can help increase spatial skills of students [39]. Similarly, jigsaw puzzle, origami, paper folding, and other open-ended puzzles can help augment spatial visualization skills of kids as these require mental manipulation of various shapes.
4. Three-dimensional block play: Building structures and objects with 3D blocks

e.g., Lego[®] [40] and wooden blocks can significantly increase children’s ability to manipulate things mentally. Giving them a structure to build will expand their knowledge of spatial visualization. Block play is important in early years in helping kids understand spatial reasoning skills which are important for later STEM learning. When kids are prompted and asked questions about their block play structures by teachers or helpers, it grows their use of spatial language as well which is related to increase in spatial reasoning [41] and can play an important role in scaffolding children’s building of more complex structures [42]. A planned block play program similar to Carol Stephenson can help greatly in developing spatial reasoning skills through building complex block structures [43]. A study examining children’s puzzle play [44] shows that it predicts children’s later performance on a spatial transformation task. It also discusses how early puzzle play varies across children and engagement is associated with demographic variables as well as levels of parent language input.

For the sake of this research, we explored 3D block building as the way to augment spatial visualization that we experiment with in our system since 3D block building tasks is a multi-step complex process that involves spatial visualization, mental rotation and perspective taking skills.

Child-Robot Tutoring

Social robots have made their way in educational setting for children in recent years. Research community has been focused on various aspects of social robots in education. Benefits of social robots, technical challenges for building robot tutors, their efficacy, impact of their appearance, role and their behaviour has been vastly studied [45]. Application of social robotics in education has become popular because of availability of robust platforms, mature technology to the point that meaningful interactions

using language and nonverbal behaviour are possible, and the established fact that embodiment results in higher learning gains in educational settings. In various educational settings embodiment of a robot plays an important role on improving the children's performance, engagement and motivation [46] [47]. Many aspects of child-robot tutoring has been investigated such as building models of student knowledge [48], evaluating different teaching and instruction paradigms [49], personalizing content for each child [9] and determining when and how to provide specific kind of assistance and hints [50]. Much of work in which robot is used as a tutor focuses on one-on-one interactions because these offer the greatest potential for personalization [45] [10]. Most of the research has focused on teaching languages [9] and solving mathematical problems [10].

Child-Robot Tutoring and Spatial Reasoning

Spatial reasoning is rather unexplored in one-on-one child-robot tutoring. Reasons can be complexity of interaction, lack of ways to track performance and measure learning gains during complex tasks and difficulty in providing appropriate feedback. However, over the past two decades computer-supported educational tools are being used in learning of spatial skills particularly reflected in geometry e.g., Edwards created a computer environment that enabled kids to learn introductory geometric transformations [51]. GeoCAL [52], a multimedia learning software based on van Hiele's theory of geometric thinking [52] comprising of several games such as jigsaw puzzles, shape tracer and stamping, has notable learning effects on geometric thinking (visual association, description and abstraction).

However, models designed to improve spatial ability in 2D world by using Web-based virtual environment [53] may not work at all in 3D world because of over-simplification of the environment. Therefore, Keren et al. [12] suggested that spatial cognition models should at least be partially embodied and assistive robots can serve the purpose

in developing children's visual-spatial and motor perception. They use assistive robots for groups (2-3 kids) of kindergarten students to promote their geometrical thinking which is one aspect of spatial cognition. The activity for this study is modification of their previous research i.e., Nao (the robot) singing the Hebrew version of song, "Head, Shoulders, Knees and Toes" while demonstrating the movements and invited kids to do some movements [54]. In this study, robot asks to recognize basic shapes on screen and distinguish between upper and lower parts of its body and are asked to find and push relevant buttons.

To summarize, development of spatial reasoning specially spatial visualization skills in 3D environment using educational social robotics is limited and rather unexplored. Thus, in this exploratory study we focus on providing a learning platform comprising of a robot tutor to help learn spatial visualization skills using 3D block building. We track the performance of children and assist them in completing a structure building task in one-on-one child-robot tutoring session.

Feedback Strategies

Good feedback practices by tutors in one-on-one sessions are essential for self-regulated learning. Good feedback practice is broadly defined as something that reinforces the student's capability to self-regulate their own performance. Self-efficacy in turn has positive effects on not only achievement, but also motivational variables such as engagement, effort and persistence [55]. For designing a good feedback practice, one needs to consider many factors. One of many principles to think about design and evaluation of self-created feedback procedures, is that feedback can be provided by a teacher, peer or tutor. Seven principles to think about design and evaluation of self-created feedback procedures are described below [1]: Good feedback practice:

1. helps clarify what good performance is (goals, criteria, expected standards)

2. facilitates the development of self-assessment (reflection) in learning
3. delivers high quality information to students about their learning
4. encourages teacher and peer dialogue around learning;
5. encourages positive motivational beliefs and self-esteem
6. provides opportunities to close the gap between current and desired performance
7. provides information to teachers that can be used to help shape teaching

Informative tutoring feedback provides useful information strategically for task completion rather than offering complete solution. Interactive-Tutoring-Feedback (ITF) refers to feedback components to guide the student toward successful task completion. Narciss has proposed the ITF model that encapsulated the state of the art in developing feedback strategies for interactive learning tasks [56] [2] [57]. As per this ITF model, a feedback strategy is defined as a coordinated plan integrating clear and decisive statements specifying at least the following aspects of a learning process with feedback [58]:

1. *scope and function*: what goals or purposes the feedback serves.
2. *content*: what information is given through the feedback.
3. *presentation*: in which form and modes the feedback content is given.
4. *conditions*: under which situational or individual conditions the feedback is provided.
5. *timing and schedule*: which event within the learning process trigger feedback statements.

This definition enables a wide range of possibilities when it comes to designing feedback strategies. Feedback comes in various forms and modes i.e., immediate vs. delayed feedback timing, single-try vs multiple-try, adaptive vs non-adaptive and implicit vs explicit feedback [2]. In human-human interactions, experienced tutors tend to provide indirect feedback to bring the attention of students to an error rather than giving corrective explicit feedback [3]. Indirect corrective feedback is argued to address some of weakness of direct corrective feedback (CF) (e.g., no explanation to why the correction is needed), as indirect CF engages learners to analyze their own work and figure out and correct their mistakes Expert tutors also employ subtle techniques like delaying affirmation or expressing short hesitation when student inquires if he is doing alright, thus, hinting him about a mistake in the current step [4]. There are other strategies like asking a leading question or subtly suggesting the student to redirect. Such strategies are enlisted in Table 2-I. On the other hand, it is argued that providing only implicit feedback can be detrimental. If a student spends too much time guessing for a solution or redirecting himself, it would be difficult for him to trace the path he took to solution [5]. Therefore, the tutoring strategy might be considered a trade-off between implicit and explicit feedback to allow room for the student to explore and learn from his mistakes, yet not be confused or stuck at an impasse.

Kleij et al. [59] present a meta-analysis of effects of feedback on student's learning outcomes in computer-based learning environments. They deduce that elaborated feedback is particularly more effective than feedback regarding correctness of answer and providing correct answer, specially in mathematics. The results suggested that immediate feedback is better for lower order learning than delayed feedback, whereas delayed feedback is more effective for high order learning. In the literature, there are contradicting results about effectiveness of immediate and delayed feedback. It seems to be task dependent. Thus, in this exploratory study we focus on providing direct/explicit or indirect/implicit feedback as soon as a mistake is made. We also

provide confirmations on correct actions of user.

Feedback Strategy	Reference
1. Draw student's attention to an error and provide second chance	[3]
2. Frequent feedback: Brief agreement with each step (e.g., Mhm and Right) when student is thinking out loud looking for agreement) or a short hesitation (often less than 1 sec) in responding with an "okay" indicating to students to assume that something was amiss with the current step	page 9,10 of [4]
3. Asking a leading question/ giving clues	[4]
4. Asking a leading question/ giving clues. Wait for student to complete your utterance, if student fails, complete it. Both work to avert a wrong solution	[3]
5. Procedural hints and explanations: procedural hints point out appropriate strategy or mention key aspect of underlying concepts (e.g., <i>When expanding a fraction, one alters the numerator and denominator equally</i>). While procedural explanations provide more details about how to put this procedure into practice. (e.g., <i>When expanding a fraction, one alters the numerator and denominator equally. To do this, multiply the numerator and denominator by the same number.</i>).	[60]
6. Conceptual hints and explanations: focus on conceptual knowledge important for solving the target problem e.g., conceptual hint: <i>When expanding a fraction, its value must not change.</i> and conceptual explanation: <i>When expanding a fraction, its value must not change. While expanding, the denominator increases, that means the partitioning becomes more fine-grained. But since the value of the whole fraction does not change, the numerator has to be altered in the same way.</i> Conceptual explanations provide additional information about relevancy of the concepts to the given problem.	[60]
7. Feedback consists of following components: a. Definition: verbally introduce definitions of domain concepts e.g., <i>Attributes are characteristics of an object that persist through the life of the object.</i> b. Example: give an illustration of given concept e.g., <i>Attributes of a car might be its color, model and make</i> c. Question: ask a question for clarity of learner e.g., <i>Why did you set the data type for money to string?</i> d. Scaffold: prompts the learner who might be lost towards a correct solution by pointing in right direction e.g., <i>Use the tutorial to learn about datatypes.</i> e. Picture: contains images, animation or video to visually explain a concept. f. Relationships: provides information to help learner understand how a concept fits into the overall problem solving activity.	[61]

<p>g. Application: shows application of a concept e.g., a learner might know definition of a constructor but might not know that a class could have multiple constructors.</p> <p>h. Exercise: supports active learning through hands-on activities or by applying a concept. It is more like a tutorial mode.</p>	
<p>8. Corrective feedback have been classified into two groups for second language teaching:</p> <p>a. Giving-Answer Strategies (GAS): when teacher directly gives the target form or shows the location of student’s error. These includes: repetition, recast, explicit correction and giving answer.</p> <p>b. Prompting-Answer Strategies (PAS): when teacher pushes students to notice their error and correct it themselves. These include: Meta-linguistic cues (teacher provides information or asks questions regarding the correctness of the student’s utterance), clarification requests and elicitation(allowing student to complete teacher’s utterance and by asking student to reformulate the utterance).</p>	[62]

Table 2-I. Feedback strategies from the literature.

Chapter 3

System Overview

The system is set up as a play area where a user tries to replicate a 3D structure in presence of a robot tutor which provides requisite feedback towards completion of the task. The system, as shown in Figure 1-1, consists of three modules:

1. Perception: Tracks the play area using color + depth (RGB-D) camera stream to detect errors.
2. Feedback: Robot provides selected feedback based on errors detected.
3. Robot Behaviour: Behavioural actions displayed by the robot that makes it life-like.

Perception

Perception module tracks three task actions while building the given 3D structure in the play area. Output of this module is mistakes made by users. This module has various aspects such as detection of RGB-D data, detection of different types of blocks, actions taken by user to manipulate the structure, creating and storing a model of structure and comparing it to target structure to look for mistakes made by user. These aspects are explained below in detail:

Setup

The system consists of a RealSense[®] d435i RGB-D camera mounted on a stand to the right of table surface looking at the playground from right side while the user sits in front of table as shown in Figure 1-1. The camera looks obliquely down on the table surface. On the left side of the table, there is a robot named Maki who acts as a tutor. A set of 2×2 and 2×4 blocks (similar to Lego[®] and Duplo[®]) in 4 basic colors (red, blue, green and yellow) are placed on the table within the reach of user. The table surface has four demarcated regions – *Play area*, *Add box*, *Remove box* and *Adjust box*. The last three areas are used as control boxes for the users to define their actions. This setup is inspired from research by Gupta and colleagues [63]. A base plate for blocks is placed in play area to fix the assembly of structure. The function of each control box is defined as under:

1. Add box: If user intends to add a block to assembly, he would place the block in the *add box* for a couple of seconds before moving it to play area which will indicate the tracking system to look for a new block of given shape and color in the play area.
2. Remove box: If user intends to remove a block from assembly, he would remove the block from play area and place it in the *remove box* before placing it on side of a table, which will indicate the tracking system to look for a block of given shape and color that has been removed from the play area to update the model.
3. Adjust box: If user intends to adjust a block (change it's position) in the assembly, he would remove the block from play area and place it in the *adjust box* and then move it back to play area as per his desire. This will indicate the tracking system to look for and remove the block of given shape and color from the assembly once a block is detected in *adjust box* and then start looking for the same block in play area to add it to a new location.

Different Modes

The main goal of system is to track user's performance towards completion of a target 3D structure and flag the mistakes made by user during the process. To achieve that system must compare the assembly to the target structure during the task. To do so, the tracking system must record the target structure first. Thus, to comply with the requirements, we propose that the system operates in two modes:

1. Learning Mode: In the learning mode, the experimenter (or teacher, supervisor, parent or guardian of the user) builds the required 3D structure using add, remove and adjust commands. The tracking system stores the constructed 3D structure as a target structure in a file. The representation of play area, block models and 3D structure is explained in section below. In short, the system records the structure from demonstration by human expert in the learning mode.
2. Teaching Mode: In the teaching mode, the system tracks the assembly of 3D structure by the user (a child) and compares each action taken by user with the target structure and flags any mistake made. It compares the current structure with the target structure on every user action and evaluate the deviations from each possible correct action since there can be multiple correct actions at every step. If the user action does not coincide fully with any of the possible correct actions, an error is flagged. In short, the system provides feedback to the user based on the previously recorded structure in this mode.

Representation of Play Area and Blocks

For tracking the model and inferring updates, the system needs some sort of representation for the assembly model. We assume that the model resides in a voxelized space [63] where each voxel is equal to 1×1 block. Thus, the model is made of 2×2 block that occupies 4 voxels and 2×4 blocks that occupy 8 voxels. Following information is

maintained for the model which is coded as a class (**Assembly_Graph**) in python. For learning and teaching modes, there are slight changes in the class attributes and functions which will be discussed as needed.

1. **.block_dict**: A dictionary of structure type *Block* (class created in python) where each block has following attributes:

(a) **.xy**: list of voxels that the block is occupying. x and y coordinates represent the voxel as (x,y). The first block, called the reference block, is fixed in the play area and its top left corner (from camera's viewpoint) is assigned the value of (0,0) where x increases towards right and y increases downwards. Voxel (0,0) is circled in Figure 3-1 as the green rectangular block is the reference block and it has

.xy = [(0, 0), (1, 0), (2, 0), (3, 0), (0, 1), (1, 1), (2, 1), (3, 1)]. The workspace is represented as a grid of voxels with (0,0) at the top left corner of the reference block. **.xy** of all other blocks is calculated with reference to (0,0) voxel.

(b) **.z**: **.z** is level from base. Base is at zero level so first layer of blocks will be at level 1 i.e., **.z** = 1 e.g., in Figure 3-1 green and red blocks are at **.z** = 1 and yellow block is at **.z** = 2. **.z** is obtained from depth data corresponding to the center pixel of the block. The value is converted into integer levels e.g., 1,2,3, and so on. after gathering some data points in the play area. Error correction in depth values may be needed since the camera is mounted at an angle and depth values for same level varies across the y-axis. Thus, a linear line is fitted on the gathered data points where error in depth varies with y position of the center pixel. This reduces the error in measurement of **.z** to great extent but sometimes in higher levels the error still exists and it always mistakenly return higher level than required e.g., at level 6, system

assigns value of 7 or 8 instead etc. To further reduce this error, we use the information we already have in the model. Consider the above example: if we get level 7 instead of level 6 for new block, while evaluating **.overlap** and **.overlap_dict** (see below), if we get empty dictionary (means no block under the new block), we decrease **.z** by 1 since it is impossible for a new block to not be connected to any block underneath it if **.z** > 1. It checks **.overlap_dict** again and continues subtracting 1 from **.z** till there is at least one block or base of play area underneath the new block.

- (c) **.angle**: **.angle** can have only two values: 0 or 90 degrees since the base of play area is fixed and blocks can only be placed at two angles. Also, it only matters if it is a 2×4 block. So, **.angle** is assigned a value of 0 when 4 voxels are along y-axis and 2 voxels are along x-axis and a value of 90 (90 degrees rotation) otherwise. For a 2×2 block it is always 0. As shown in Figure 3-1, the green and red blocks are at an angle of 90 degrees while yellow has angle of 0 degrees.
- (d) **.shape**: **.shape** can have one of two values, either 1 or 2: 1 for 2×2 block and 2 for 2×4 block.
- (e) **.color**: **.color** can have one of four values: 1 for red color, 2 for blue, 3 for green and 4 for yellow.
- (f) **.idx**: **.idx** is a unique ID assigned to each block that is a part of assembly.
- (g) **.overlap**: **.overlap** is a Boolean variable which is true when the block is connected to another block below it, in the assembly apart from base of play area.
- (h) **.overlap_dict**: **.overlap_dict** is a python dictionary that contains information of all the blocks connected under the new block. Keys in this dictionary are the **.idxs** of the connected blocks and associated with those

keys are **.xy** voxels of the respective blocks that overlap with the new block. For instance, as shown in Figure 3-1, the yellow block(**.idx = 3**) will have **.overlap_dict = {1 : [(1, 1), (2, 1)], 2 : [(1, 4), (2, 4)]}** since it overlaps with blocks of **.idxs** 1(green block) and 2(red block) whereas blocks 1 and 2 will have empty **.overlap_dict** since there is only base underneath those.

2. **.graph_dict**: It is a dictionary that stores connections of each block with other blocks directly connected on top of it using their **.idx**. For instance, if the target structure has only three blocks and block with **.idx 3** is connected on top of blocks with **.idx 1** and **2** (see Figure 3-1), **graph_dict** will be: $\{1 : [3], 2 : [3], 3 : []\}$. Just by looking at this structure we can tell that only block 3 can be removed in the next step and not 1 and 2 since removing those would require removing block 3 first. This dictionary helps us keep track of removable blocks at any step.

3. **.graph_dict2**: It is another dictionary that stores connections of each block with other blocks directly connected below it using their **idx**. This dictionary is useful when we need to compare structures in the teaching mode. At each step, it is used to find out all possible correct actions given the target structure. For instance, if the target structure has only three blocks and block with **.idx 3** is connected on top of blocks with **.idx 1** and **2** (see Figure 3-1, **.graph_dict** will be: $\{1 : [], 2 : [], 3 : [1, 2]\}$. If in the current structure, during the teaching mode, the user has only placed block 1 correctly. Looking at the example dictionary we can tell, the only next correct action is to add a block similar to block 2 since it does not have any block under it, which is not already added. Just by looking at this dictionary, we can tell that 3 can only be added after both 1 and 2 have been added in the assembly since 3 has to be on top of 1 and 2. Thus, this dictionary helps us identify all the possible correct actions at any step.

4. **.pub**: **.pub** is only used in the teaching mode. It is a ROS publisher for message type *String* that publishes the feedback statement for robot to provide to the user when a mistake is made.

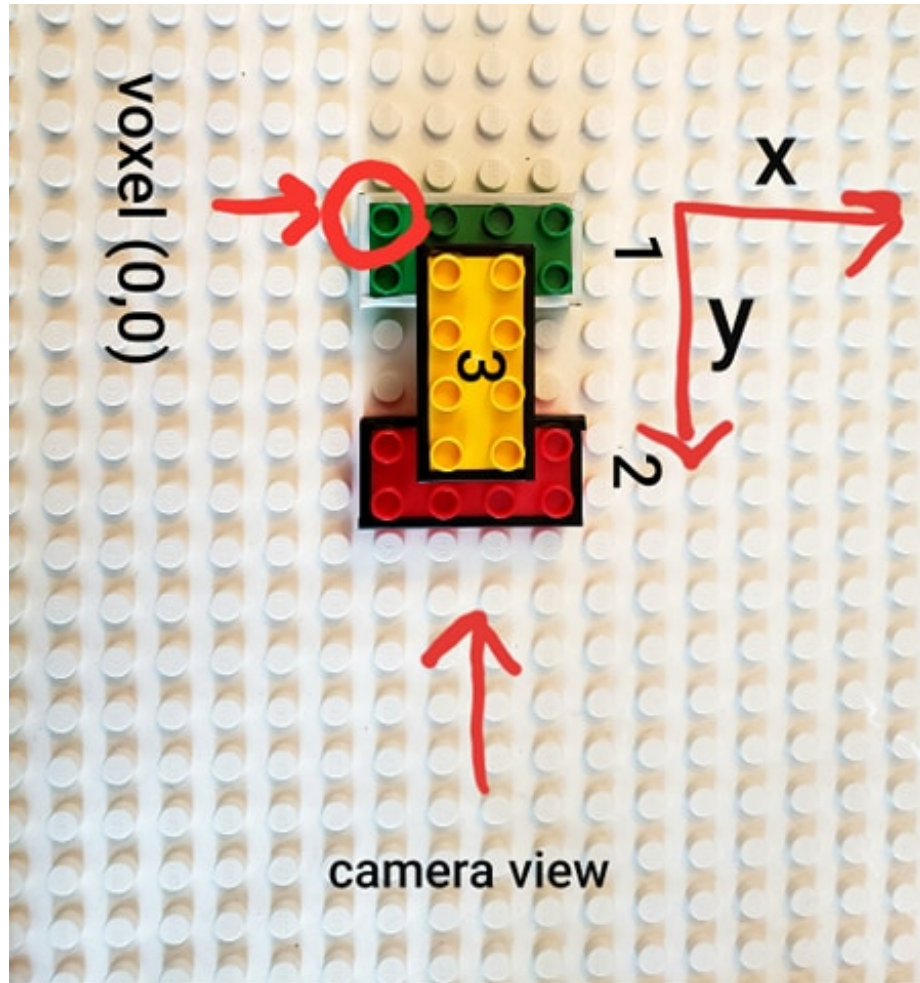


Figure 3-1. Attributes of *Assembly_Graph* and *Block* explained: The reference voxel (0,0) is circled in red. Green block has `.idx = 1`, red block has `.idx = 2` and yellow block has `.idx = 3`. Green and red blocks are at 90 degrees angle and yellow is at 0 degrees. Green and red blocks need to be placed before yellow block. Yellow block is the only block that can be removed in the next action.

For teaching mode, class **Assembly_Graph**, has an extra attribute of type **Assembly_Graph** to pass the target dictionary that is saved in the learning mode. Using these attributes various functions are defined within the class to add and remove blocks, save and load the model, display the model and compare the models to flag

errors. Feedback selection and passing the feedback statement to robot is done using functions within this class as well.

Processing Pipeline

Figure 3-2 shows the work flow of the system for both modes. The RealSense® d435i camera captures and provides a continuous RGB-D images stream. First of all, the reference block is added to the model giving (0,0) coordinates for **.xy** of class *Block* as described above. Once that is done successfully, the system scans for control command in three control boxes: add, remove and adjust. As soon as a block is detected in one of the control boxes, the system processes the play area for requisite action and updates the model accordingly. If system is in the learning mode, on completion of required action, the system again scans for new action and the cycle continues till program is exited. If system is in teaching mode, on completion of requested action, the system compares the current action with all possible correct actions (comparison is made on add and adjust commands and not on remove command) and forwards the errors with respect to every possible correct action to the feedback module. The feedback module chooses a message based on errors and feedback strategy which is published for the robot to convey to the user. Once this action is completed, system starts scanning the control boxes again for next action request. The system exists when the current model is same as target model. In the following sections, we describe details of how each step in the workflow is accomplished.

Scanning Control Boxes

The continuous RGB image stream is divided into 4 fixed regions: add box, remove box, adjust box and play area. The add, remove and adjust boxes are scanned for next command after reference block is added. The RGB image from each box is segmented using Hue, Saturation, and Value (HSV) based color segmentation

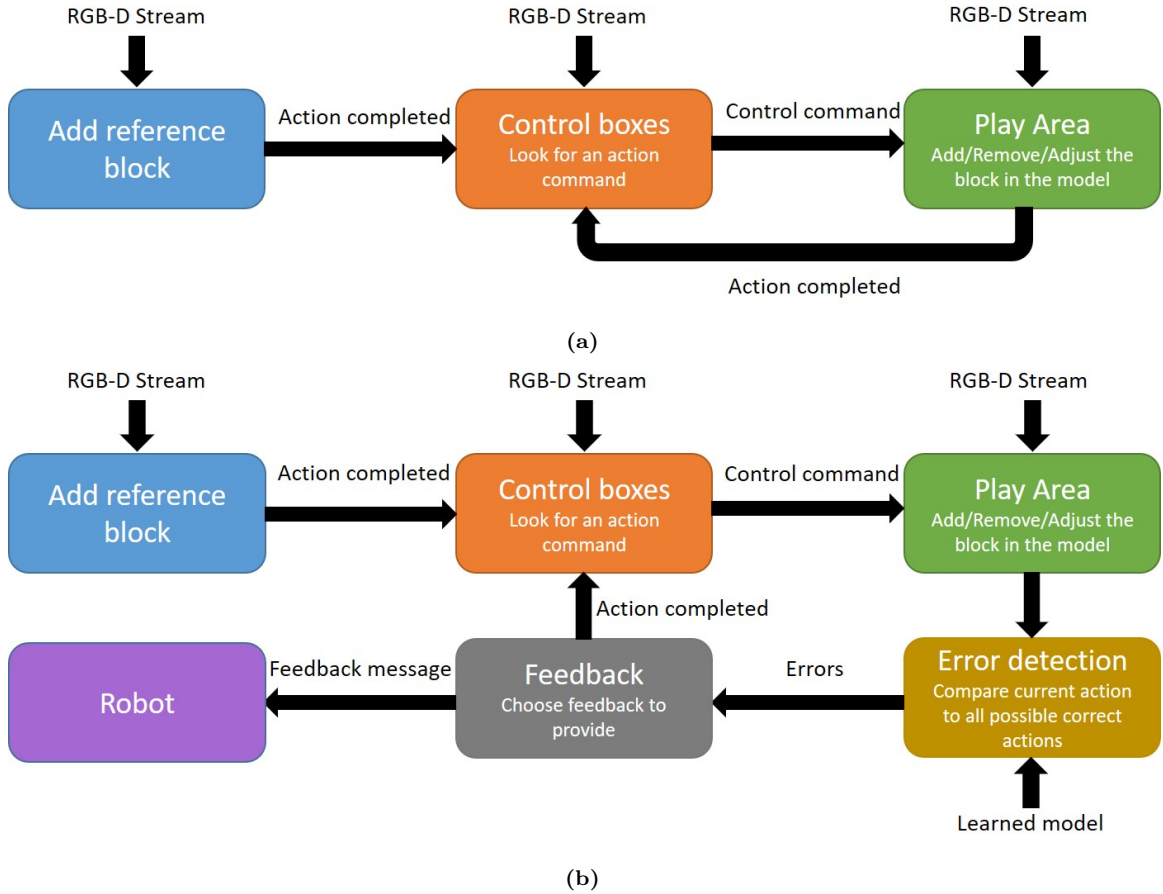


Figure 3-2. Processing pipeline for perception module (a) for learning mode (b) for teaching mode.

to create a binary mask for each of the four colors (red, green, blue and yellow). Bounding rectangle is found for largest contour in the mask by using builtin functions of openCV (`findContours(arguments)`, `max(arguments)` and `boundingRect(arguments)`). If a bounding rectangle is found for same color in same control box for 50 consecutive frames, it is considered a control command. The color and shape (assigned by looking at width to height ratio of the bounding rectangle) of the new block is noted, the respective command (add, remove or adjust) is activated and system starts scanning play area.

Scanning Play Area

RGB-D continuous stream is scanned for new action as per the control command. For add action, first of all, the RGB image is segmented using HSV based color segmentation to create binary mask of color of new block (color is known from control command). Now, this mask is compared with masks from previous frames and if there is no change in the mask for n of consecutive frames ($n= 50$ for our system), all contours whose bounding boxes have width of more than m pixels ($m = 15$ for our system) and have the same shape as indicated by control box are considered as *possible additions*. The x and y pixel coordinates of the center of each bounding box is used to get depth (z) from the corresponding depth data. If the block is 2×4 block, angle is calculated for each possible addition using ratio of width to height of the bounding box. Next, all the *possible additions* are compared with the existing blocks in the model (x,y,z , shape, color and angle). The contours that coincide with existing blocks in the model are discarded thus leaving us with the contour of new block. This block is then added to the model by passing its attributes (x,y,z ,shape, color, angle). The system keeps scanning the play area for new block till a block is added in the model. Once the block is added, the system generates a flag indicating that the add action is completed. For learning mode, each time new block is added **.idx** is counted up. For teaching mode, if the new block added in the current model matches one of the blocks in target model, **.idx** of the new block in the current model is assigned the same value as in target model otherwise it is assigned a much higher value e.g., 21, 22, 23. This makes comparison of two models efficient and easy.

For remove action, mask of known color is created using HSV based color segmentation and is compared with previous frames till there is no change for n consecutive frames ($n = 50$ for our system). Contours with bounding boxes are calculated. From the model, using a function **.removable_blocks()**, we can get the **.idx** of all the blocks that can be removed from the current dictionary. We can also get the **.idx** of blocks in

the model of the shape and color associated with the remove command. Intersection of these two lists, will give us *possible removable blocks* i.e., the **.idx** of the blocks that are removable and of given specifications from remove command. If it is only one block that lies at intersection of these two lists, we remove that block from the model. If there are multiple blocks that can be removed, we look at the bounding boxes of contours of known shape and check which blocks from the model are still in the play area by comparing x, y, z and angle of the contours to the blocks in the model. Discard the **.idx** of these blocks from *possible removable blocks* as these are still present in the play area. There should only be one block left behind in list of *possible removable blocks*, which is the one that has been removed from the play area and placed in the remove box. System generates a flag indicating that the action is completed successfully when the block is removed from the model. Remove action is same for both learning and teaching mode.

For adjust action, remove action is completed followed by add action for the block detected in adjust box. System generates a flag on completion of action and starts scanning for next action.

Types of Errors Detected

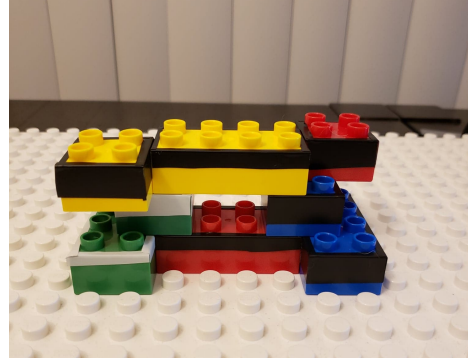
Following types of errors are detected for new block to be added in comparison to all possible correct blocks and passed to feedback module:

1. Shape: The shape of new block is different from the possible correct block.
2. Color: The color of new block is different from the possible correct block.
3. Orientation: The orientation of new block is different from the possible correct block.
4. Level: The **.z** (z-level) of new block is different from the possible correct block.

5. Position: The center of new block is different from the possible correct block.



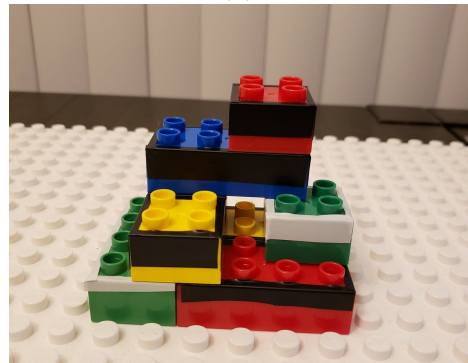
(a)



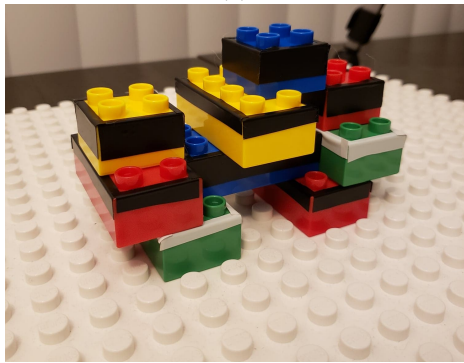
(b)



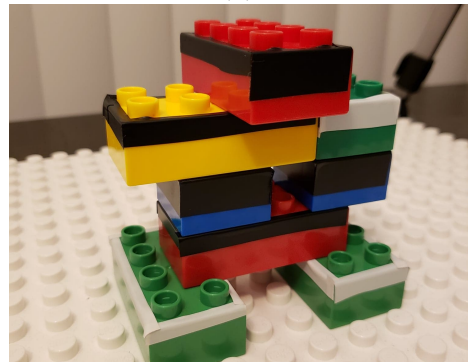
(c)



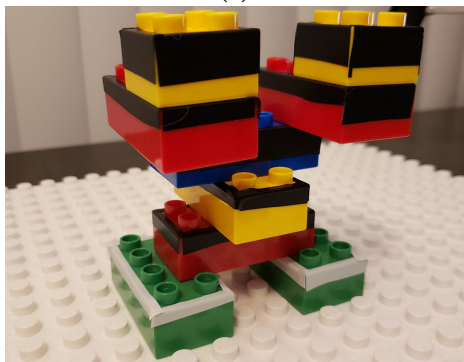
(d)



(e)

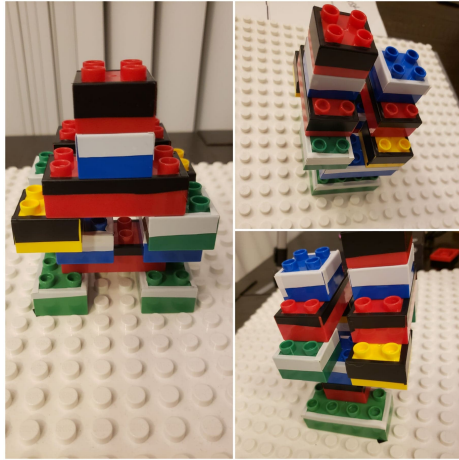


(f)

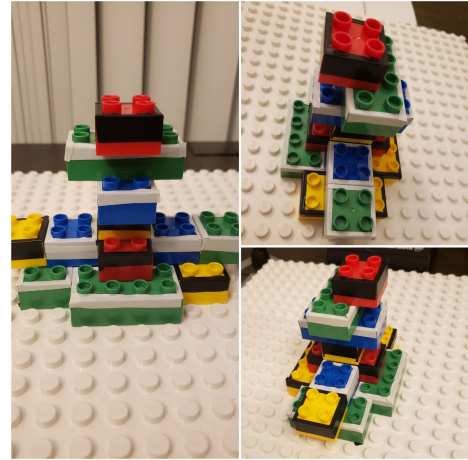


(g)

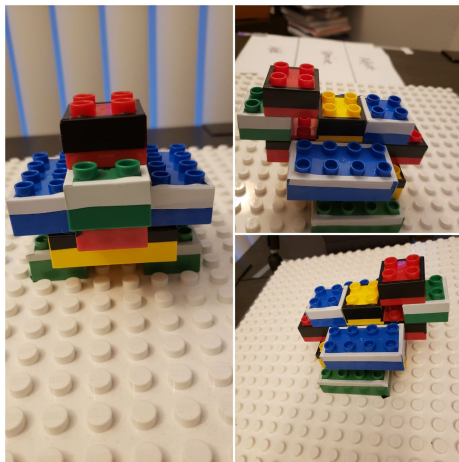
Figure 3-3. Simple 3D structures successfully completed in both learning and teaching modes.



(a)



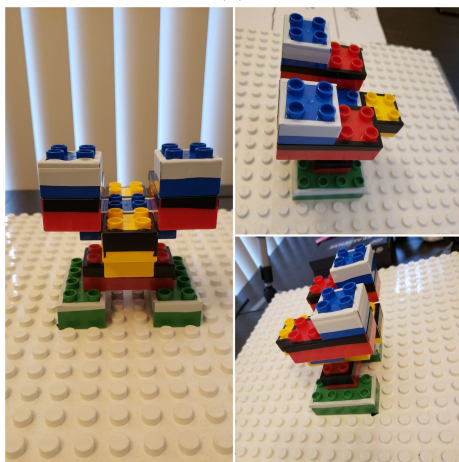
(b)



(c)



(d)



(e)

Figure 3-4. Complex structures successfully completed in both learning and teaching modes. Picture on left is front view and picture(s) on right are side view(s).

Evaluation and Limitations

The system runs on Ubuntu 16.04.6 LTS (in dual boot with windows) in real-time on a laptop PC with 5-core 2.3 GHz Intel[®] processor. It takes about 2-5 seconds to update the model when required. Mostly, there is noticeable lag in the system. The system works fairly well in most of the cases. Some simple structures, which can be recreated by looking at only one view, were successfully completed by us in the learning mode and tested in the teaching mode as shown in Figure 3-3. Some complex structures were also tested by the system are shown in Figure 3-4. These complex structures require at least pictures of 2 views to recreate. These structures were successfully detected and stored by our system. Various random structures were also tried to look for possible errors and limitations of the system. Some of the errors and limitations observed are:

1. Occlusion: Since only one camera is used to track the workspace, if the block to be added or removed is not fully seen by the camera, it will go undetected and cause errors as shown in Figure 3-5. Using multiple RGB-D streams from various cameras can eradicate this issue.
2. Confusion: As explained above, the error in $.z$ is fixed by intuitively lowering the level until it overlaps at least one block underneath it. This fixes the $.z$ for time being for that block but when new block of same shape (or appearing to be of same shape) is requested to be added, the overestimated z would stop at a level higher than before thus re-adding that object thinking that it is a new object at upper level as shown in Figure 3-6. Using better method for estimating error in depth because of oblique camera or using multiple RGB-D cameras can help reduce this error.
3. Due to changes in light source, sometimes, HSV based color detection is faulty but it can be fixed by fixing the light source.

4. The whole 3D structure is grounded on a base, thus not allowing the user to move it around in space.
5. The blocks have been taped on edges (white or black tape depending on which color contrasts better), as seen in figures, to assist color segmentation and enable differentiation of same colored blocks. The boundaries allow different shapes and multiple blocks of similar characteristics to be used for block building.

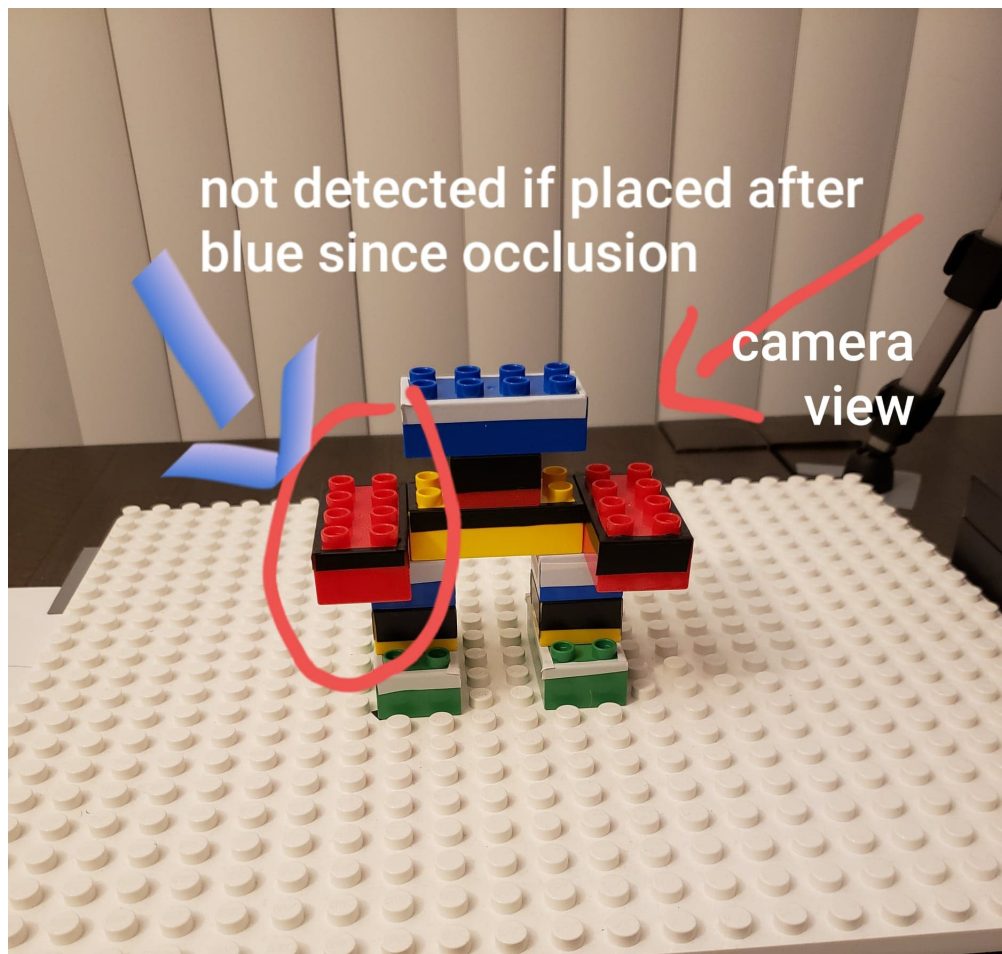


Figure 3-5. Example of occlusion: The circled 2×4 red block on level 4 will not be detected if placed after the 2×4 blue block on level 6 because the camera cannot see it fully.

Despite these limitations, the proposed perception module has some advantages over similar systems. It is able to use two different shapes unlike in Gupta and

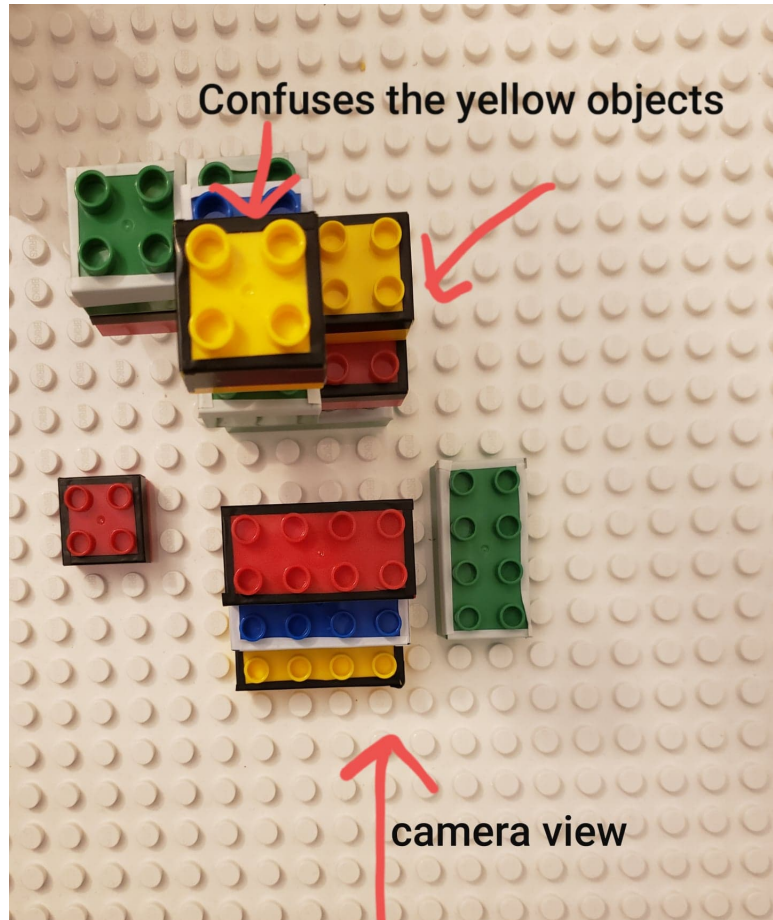


Figure 3-6. Example of confusion: The 2×4 yellow block at lower level is mistakenly added as a 2 yellow block because it appears to be a 2×2 , even before the user placed the top level 2×2 yellow block because the rectangle at $.z = 4$ originally had been assigned $.z > 4$ and it was corrected to 4. Thus, now the visible portion of lower level yellow block appears to be a new block added at level 5 ($.z = 5$).

colleagues' system [63] which uses only 2×4 blocks. It can be extended to other shapes e.g., 1×2 blocks easily. Also, it is able to use multiple blocks of 4 colors of 2 shapes unlike Jones and colleagues' system [64] which uses only one block of each shape (2×4 and 2×2) and 4 colors (total of 8 blocks). Our model representation is robust and efficient. Such simple representation enables us to track an assembly which has multiple paths towards correct solution. We are able to get all possible correct next actions and all possible removable items at any step. It is not a step by step process strictly since there can be multiple correct actions at any step unlike step by

step guided system proposed by Gupta and colleagues [63]. Also, we use a reference block for defining (0,0) for xy-coordinates, thus, the recorded graphs stay valid even if the reference block is moved at the start of each 3D building task.

Feedback

Second module of the system is providing feedback in the teaching mode to help the user finish the 3D block building task at hand. Feedback system receives various types of errors in new block from each possible correct block. Based on the number of possible correct blocks, number of errors from each possible correct block and type of errors, the error on which the robot must give feedback is decided. Following rules are followed:

1. If there is a possible correct block with 0 errors from new block, a brief statement confirming the correctness of the action and suggesting to continue is provided. We call these statements *continuers*.
2. If there is only one possible correct block and there is only one type of error, that error is selected by default.
3. If there are two or more errors with respect to the only possible correct block, following priority is considered while choosing the error to respond to:
Error in: Shape > Color > Orientation > Level > Position
4. If there are two or more errors with respect to two or more possible correct blocks. The possible correct block with which there are least errors is selected and the error priority is decided as following:
Error in: Shape > Color > Orientation > Level > Position

Two main questions to answer are: When to give feedback and what would the feedback statement say. As far as *When* is concerned, for this exploratory study we provide

feedback to user every time a mistake is made. Although due to delays in execution of robot behaviour continuers might be skipped every now and then. To answer the second question, we describe our framework for feedback in Table 3-I. We refer back to various feedback practices described in literature (Table 2-I) while designing these feedback statements. These statements are published to a rostopic for robot to convey to the user.

Robot Behaviour

The third and last module of the system is introducing somewhat life-like movements in the robot. The robot we used is Maki, a 3D printed robot with 7 servos and controlled using arbotix_m controller as seen in the setup (Figure 1-1). Maki provides verbal feedback when a mistake is made by the user. Microsoft® Azure text-to-speech service is used to convert the feedback statement into voice. Apart from speech, we programmed Maki to nod slightly to acknowledge the correct actions of user along with continuers such as *go on*, *continue* and *good* etc. This is to remind user of its presence and keep the user engaged. We have implemented another robot behaviour known as referential gaze. In idle state, Maki seeks to make eye contact with user. When the user makes a mistake, the robot looks towards the play area to redirect user's attention to the mistake (referential gaze), narrates feedback to the user and then looks back towards the user to take requisite action to correct the mistake made.

Error type	Feedback Type (refer to Table 2-1)	Feedback Statement
Shape or Color	8a.	The shape/color should be __ instead of __
	1. 7c. 8b.	Hmm, are you sure you need a __ here?
	3. 4. 7c.	So, if this is a __ , what should be the next step?
	7c. 8b.	What if we try a __ instead?
Orientation	8a.	The orientation is wrong. You need to rotate the block
	1. 8b.	Umm. Does this orientation looks right to you?
	7d. 7h. 8b.	What if we rotated the block, Wouldn't it look better?
Level	8a.	The level is wrong. Move the block to upper/lower level.
	3. 8b.	What if we moved the block to upper/lower layer?
	7c. 8b.	Hmm, would you check if the block is at the right height?
Position	8a.	The position is wrong, Move the block __.
	1.	I think the block is not at the right position. Would you try fixing it?
	7d. 7h.	Hmm, Let's try moving this block around a little bit to see if we get it right.
	3. 4. 7c.	If this is the right position for this block, think what would be on top of it?
For all	7d.	Let's look at the given picture to see if this looks alright.
No error	2	One of following <i>continuers</i> with a nod: Go on. Continue. Hmm. Good. Right.

Table 3-1. Feedback statements based on selected error and inspired from literature as summarized in Table 2-1.

Chapter 4

Exploratory Study

We conducted a study to explore how various feedback strategies (implicit and explicit) can impact user's ability to self-regulate and scaffold learning process. We want to understand if and how a robot tutor can be of assistance during a spatial visualization task and how specific robot behaviour will impact user's perception of the robot. Various aspects of this exploratory study are explained below:

Participant

Our participant was an elementary-aged female student. She is currently in fourth grade. Her past experience with technology in the classroom is limited to using tablet computers. She has interacted with Maki (the robot) previously for a different pilot study.

Task

For the sake of this exploratory study simple structures as shown in 3-3 are selected as 3D block building tasks. The complex structures(Figure 3-4) require at least 2 different views to be completed and after initial interaction with the participant, we found that these structures are too complex for her to follow even with feedback from Maki. Thus, we use the simple structures for this study.

Procedure

As seen in Figure 1-1, the setting of tasks consisted of Maki the Robot, playground with blocks, play area and control boxes, a camera mounted on a tripod stand, an iPad for displaying the target 3D block structure and the participant seated on a chair.

First of all, the participant is seated. She is presented with a 3D structure to build with blocks as a pre-test for the study. Pre-test is administered to develop an understanding of her existing knowledge of block building. The task given as pre-test is shown in Figure 3-3a. The experimenter explains the functionality of box controls and play area, and asks the participant to build the given structure. Participant was aware of the functionality partially because of similar interaction in early stages of this project. This prior interaction was carried out to understand if the proposed playground with control boxes is understandable by school-going kids and to analyze the complexity of 3D block building tasks that can be targeted towards elementary-aged students.

After pre-test, Maki introduces itself and explains what the participant will be doing today and prompts her to start her first task, as shown in Figure 3-3b. During the task, Maki provides feedback whenever a mistake is made. On correct actions, Maki narrates statements encouraging to continue or affirming correctness of action e.g., *Okay. Good. Go on. Continue.* etc. so that the participant does not neglect the robot in the process in case she is not making mistakes for long time. The run ends when she successfully completes her task. Maki congratulates her on success and she is asked to take a break before next task with a different target structure. Maki welcomes her back to each next task and prompts her to start the task following same instructions. Four more structures (Figure 3-3c, 3-3d, 3-3e and 3-3f) are done in next tasks. After all the tasks, a post-test similar to pre-test but with different target structure as shown in Figure 3-3g is taken. After last task, a post-study interview is

conducted in which the experimenter asks the participant about her experience with the system and Maki, whether she found Maki to be helpful or bothersome, whether she would trust Maki to guide her in learning about the given task and if she felt encouraged to continue the task.

Results and Discussion

There are three main sources of data: *pre and post test*, *post- study interview* and *videos* of the interaction. The videos are analyzed by the author. Rudimentary qualitative analysis is done on the video data. We analyzed the pre and post test, videos from 5 tasks and post-study interview qualitatively. Results are discussed here:

Pre and Post-test

The participant successfully completed both pre-test and post-test task. Both tasks were of similar complexity. Overall, outcome for both the tasks was correct but for the pre-test the participant used remove control once self-correcting her mistake. However, in the post-test, she did not make any mistake and only used add control. Also during the pre-test task, experimenter had to explain the functionality of adjust and remove blocks once again. Since there was only one session and tests ended up being rather simple, nothing can be concluded about learning gain from one session qualitatively.

Video Analysis

Many observations were made by the viewer of videos of interaction with Maki. The main observations over the 5 tasks are summarized here:

1. The participant looks at Maki when it communicates with her for introduction of task, feedback statements and robot actions (e.g., nodding and referential

gaze). But as she goes further into the session, time of her eye contact with Maki reduces while it is repeating the same statements at start and end of each task. But, in the last task, the ending statement was different since Maki tells her about end of her session and bids her goodbye. At the start of this ending statement, she was looking down but as soon as the statement was different from the previous ending statements, she turned her gaze towards Maki to process new information. It is interesting to note that she stopped looking at Maki and started listening passively when it said predictable and repetitive statements but as soon as it narrated new information, she paid full attention to Maki.

2. She follows what Maki is asking or suggesting her to do. For instance, during introduction, when Maki says to look at the screen for picture of the target structure, she looks at the screen. When Maki narrates continuers like *Go on*, *continue*, and so on. she moves on trusting Maki's suggestion that she is correct. She considered *nodding* of Maki as confirmation of her being correct as well.
3. She looks at robot after every action for confirmation if she did it right or not. Sometimes, she would skip looking at robot but as soon as the robot nods and says a continuer, she would look at robot and move on. Sometimes if robot would not respond to her action, she would wait a couple seconds looking at robot for a response to her action. She would move on if she did not get a confirmation assuming that she did it right. This suggests that if you provide frequent feedback, the user can get accustomed to it and wait for the feedback. Also, it is interesting to note that the lack of feedback made the participant assume she was doing alright. She trusted Maki to point out her mistake.
4. During task 4, even though Maki was congratulating her on her successful completion of task, she was looking at the picture of the target structure. She mentioned it to the experimenter that she thought there was still one more block

left to be added in the structure. This might be because of few glitches in the same task. The system made a wrong detection because she took too much time to fix the block and it got detected midway and robot hinted to look at the picture again to see if it is alright. While adjusting again, the robot mentioned that the position is wrong and asked to move the block upwards. This confused the participant since she was placing the block at right place. Experimenter interfered and suggested her to adjust the block again. After adjust action, Maki nodded and suggested to move on since system was able to correct its wrong detection during adjust action. This shows that the participant did not blindly trust Maki and since it made a mistake earlier in the same task, she was skeptical of Maki congratulating her on completion of her task and wanted to make sure for herself.

5. Apart from task 3, Figure 3-3d, she did not make mistake in any other task suggesting that the simple tasks (Figure 3-3) are easy enough for kids of her age group.
6. Task 3 mistakes: She made 2 mistakes in task 3. The target structure is shown in Figure 3-3d. The first mistake was that she used a yellow rectangular block instead of a square one at the fourth step. As it is clear from the picture of the target structure, the structure has a possible confusion i.e., *a rectangle of yellow*, instead of *a square of yellow and a square of blue* at level 2. Maki responded saying "*Are you sure you need a rectangle here?*". She inferred that the hidden block is blue square not continued yellow rectangular block and she fixed her mistake by removing the yellow rectangular block and replacing it with two square blocks of yellow and blue color. The second mistake was that she placed the blue rectangular block at level 3 before the green square block at the level 2. So, Maki pointed out that the shape should be a square instead of a rectangle.

This feedback statement was not very clear i.e., the blue rectangular block at level 3 is the correct block but green square needs to be added at level 2 before that, she corrected her mistake anyways. This implies that merely pointing out that a mistake has occurred nudged the participant to correct herself. Thus, it is safe to say that the participant received feedback from the robot positively and self-corrected her mistakes.

Post-study Interview

To support observations from the video analysis, we conducted a post-study interview. We asked the participant questions about her overall experience, her perception of Maki, her comfortability with the setup, her feelings about Maki's appearance and behaviour. This interview revealed lot of interesting information. Some of the remarks from the participant and our interpretation of these remarks are detailed below:

1. **Fun and engaged:** She thought that the overall experience was fun despite some of the bugs and wrong detections. She also said that she would enjoy more sessions with Maki because it was fun. She expressed that if she had to do the tasks by herself, it would be boring.
2. **Fascination:** She was fascinated with Maki since she does not work with robots usually so she thought it was fun. She would prefer working on similar tasks with a robot over a human because she finds that humans can be boring sometimes and *"humans just do not work with other humans"*.
3. **Helpfulness:** She perceived Maki to be cool and helpful because she noticed that whenever she was correct, the robot nodded and said *Go on, continue* etc. and whenever she was wrong, it helped her with a hint. She expressed that she would like more sessions with Maki and she learnt a bit about block building task and she will be better at it now. The session boosted her confidence in her

ability to work on similar tasks. She expressed that without Maki it would not be easy to complete the tasks.

4. **Teacher vs Peer:** She found Maki to be acting like a friend when it gave hints to help her through the task but she thought of Maki as a teacher when it told her if she was wrong or right. This suggests that direct feedback, e.g., using the word *wrong*, creates impression of Maki as a teacher while helpful hints created more of a friend-like personality.
5. **Trust:** She trusted the robot but not blindly. She trusted Maki when it provided feedback on her mistakes. At some point, she thought she was right and the robot was wrong. Even if the robot made mistakes, she was okay with it because Maki was not a human and she could trust it because it was right sometimes. Maki's mistakes in providing feedback would not stop her from working on further sessions with Maki.
6. **Scaffolding and self-regulation:** When Maki used the word *wrong*, her emotions were hurt. She said it made her sad but when Maki acted in a friend-like manner and gave suggestions without using the word *wrong*, she felt better and she knew that she can correct her mistake. She found Maki to not judge her for her mistakes unlike humans. This is an interesting indication that using softer words to suggest mistakes might have resulted in the participant scaffolding and self-regulating her learning process.
7. **Clarity of feedback:** She found Maki's remarks to be loud and clear. She understood her mistakes from feedback statements and was able to fix those.
8. **Frequency of feedback:** The participant was asked if she found Maki to be excessively talkative, annoying and interrupting her. She said that this was not the case. In fact, whenever Maki would not respond she was concerned that it

should have said something.

9. **Appearance and robot behaviour:** The loud noise made by robot while moving its head was scary but not in the sense that it would harm her since Maki has no limbs. Whenever Maki moved, she was looking forward to hear something from it. It is interesting to note that robot behaviour such as nod and referential gaze are interpreted as robot's intent to communicate with the participant. The participant was also able to distinguish the meaning of nodding (correct action) and referential gaze (mistake made). Maki's voice was perceived neutral and non-expressive. In her words, it was lacking highs and lows. It did not sound excited or if anything mattered to it. She was of the opinion that Maki should at least be expressive. She did not expect Maki to have emotions as such.

Design Implications and Suggested Improvements

This exploratory study has helped understand the effectiveness and drawbacks of our perception system, feedback strategy and robot behaviour. It has shed light on the potential of robot assisted 3D block building to augment spatial visualization skills in elementary-aged students using implicit feedback strategies to suggest mistakes made in place of explicit corrective feedback. 3D block building task is found to be more engaging and interesting in the presence of the robot. The robot assisted the participant to correct her mistakes through frequent feedback. The participant anticipated feedback from the robot after every action and it concerned her if the robot did not respond. Implicit feedback, such as suggestion, hints and giving her second chance to correct herself, created positive emotions such as confidence in the participant and helped her self-regulate her learning process. Implicit feedback had clarity regardless of not putting the exact mistake in words. Using strong words such

as *wrong* had a negative impact on the participant thus, suggesting explicit corrective feedback can harm self-confidence. The robot was able to build some level of trust despite a couple of mistakes on its part. The robot was perceived as a teacher when it communicated whether the action of the participant is right or wrong whereas it was considered a friend when it gave suggestions and hints to assist the participant. Robot's behaviour i.e., nodding and referential gaze were easily distinguishable. New statements or actions by robot would get the participant's full attention whereas repetitive statements caused participant to go into passive listening.

This exploratory study will enable us create a framework for a user study to evaluate effectiveness of our improved system and study design. Following are the suggested improvements that will enable us in resolving certain issues of our system and get it ready to launch a full-fledged study to analyze learning gain, self-regulation and scaffolding ability of students and their perception of a robot tutor in one-on-one session while conducting a 3D block building task to augment spatial visualization:

1. Two factors need to be part of elaborated study: Type of feedback (implicit vs explicit) and frequency of feedback (when to give feedback). Although given the results from this study, implicit indication of mistake implied better ability to self-regulate and friendliness and likability of the robot, data is too less to make a conclusion. Similarly, the participant preferred frequent feedback in this study but nothing can be said about its effects on learning gain. It is possible that interrupting frequently to provide feedback might make the user too dependent on the robot to figure out when and what mistake is made. Nothing conclusive can be said about these two factors with limited interaction data. These two factors need to be explored further in detail.
2. The perception system needs to be improved to increase credibility of feedback provided by the robot.

3. To be able to evaluate learning gains, ability to scaffold and self-regulate, it would be better to have multiple sessions over a longer period of time since spatial visualization skills are hard to quantify in one session. Also, these skills improve over time and we are interested in longer-term effects of feedback strategies and robot assistance on spatial visualization.
4. Designing the 3D target structures needs to be guided by experts in the field e.g., mathematics teachers, psychologists, and so on. Complexity of pre and post-test needs to be adjusted as well. Standard tests may be conducted to evaluate learning gain in spatial ability.
5. Introductory and ending statements provided by the robot at the beginning and end of each task respectively should be made less repetitive to better engage users.
6. Loud noises of robot's joints while moving should be minimized to avoid sense of fear in users. Robot voice is neutral and non-expressive in the current system. It could be made more human-like. The robot should sound excited, concerned, happy or disappointed etc.

Suggested improvements in the proposed system will enable us to progress in this area of research. More studies about the tradeoff between implicit and explicit feedback, effectiveness of various types of implicit feedback and its personalization and right timing for intervention and impact of other various types of feedback strategies etc. will follow.

Conclusions and Future Work

We have demonstrated a system that tracks 3D block building task in real-time. In the *Learning* mode, the system records and stores the assembly process through block additions, removals and adjustments done by the expert (experimenter or teacher etc.) In the *Teaching* mode, a user is prompted to construct a 3D structure provided as a 2D picture. It keeps track of assembly process and detects mistakes made by user and based on those mistakes it provides a selected feedback statement to the user. The system is able to detect shape, color, orientation, level from base and position errors of current block from all possible correct actions. The feedback statements that we have employed mostly fall under the category of implicit and explicit feedback. The feedback is narrated by a robot, Maki. Maki uses nodding with continuers or referential gaze with feedback statements based on type of error that has occurred to inform the user if the user is doing alright or has made a mistake respectively.

A brief exploratory study depicts the potential of robot assisted 3D block building task. System was easy to used despite few glitches. The participant perceived Maki to be helpful and essential for task completion. She trusted Maki to guider her despite a couple of poor feedback statements. She was critical of Maki's mistakes indicating absence of blind trust. Maki was perceived as teacher when it used direct feedback whereas it was considered friendly when it presented useful hints and suggestions. Using softer words to implicitly suggest that the participant has made a mistake helped the participant gain confidence in her ability to correct her mistakes by trying again. The feedback statements were loud and clear for the participant to follow easily.

The participant became accustomed to frequent feedback. Maki's appearance and behaviour has room for improvement, although, its intentions were easily interpreted by the participant. Predictable statements shifted the participant from paying full attention to listening to Maki passively.

There are some limitations of the perception system that can be improved on. We would like to extend this system to work with more shapes, sizes and colors of blocks which is not hard since our system is flexible to include different kinds of blocks. It would be interesting to move towards algorithms that can handle addition and removal of multiple blocks at a time and allow for merging sub-assemblies. To achieve this, changes might be needed in the model representation. Eventually, it would be interesting to see if similar framework can be extended to more complex building tasks that are not limited to voxelized space.

The exploratory study has opened up avenues for multiple studies. One of many possibilities is a study designed to evaluate the impact of implicit vs explicit feedback on self-efficacy and learning gain of school-going students, and their perception of such robot tutor while working on a spatial visualization task. Another multi-session study over a long period of time to evaluate the effect of personalization of feedback strategy on scaffolding and self-regulations for school-going kids for spatial visualization tasks such as 3D block building would prove to be beneficial. It would also be interesting to study the effect of timing, frequency and type of feedback on user's learning gains, self-regulation and perception of robot. Analyzing effect of various non-verbal robot actions such as nodding, referential gaze, hesitation, and so on, as a source of implicitly communicating errors instead of narrated feedback statements could lead to interesting findings. We hope this initial work inspires future research in the field of social robotics and education.

References

1. Nicol, D. J. & Macfarlane-Dick, D. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education* **31**, 199–218 (2006).
2. Narciss, S. Feedback strategies for interactive learning tasks. *Handbook of research on educational communications and technology* **3**, 125–144 (2008).
3. Lepper, M. R. *Self-perception and social-perception processes in tutoring: Subtle social control strategies of expert tutors in Self-inference processes: The ontario Symposium* (1990), 217–237.
4. Fox, B. A. Cognitive and interactional aspects of correction in tutoring. *Teaching knowledge and intelligent tutoring* **1** (1991).
5. Lewis, M. W. & Anderson, J. R. Discrimination of operator schemata in problem solving: Learning from examples. *Cognitive psychology* **17**, 26–65 (1985).
6. Koedinger, K. & Alevan, V. Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. *Educational Psychology Review* **19**, 239–264 (Sept. 2007).
7. Leite, I., Martinho, C. & Paiva, A. Social Robots for Long-Term Interaction: A Survey. *International Journal of Social Robotics* **5** (Apr. 2013).
8. Oh, K. & Kim, M.-S. Social attributes of robotic products: observations of child-robot interactions in a school environment. *International Journal of Design* **4**, 45–55 (2010).
9. Gordon, G., Spaulding, S., Westlund, J. K., Lee, J. J., Plummer, L., Martinez, M., Das, M. & Breazeal, C. *Affective personalization of a social robot tutor for children's second language skills in Thirtieth AAAI Conference on Artificial Intelligence* (2016).
10. Ramachandran, A., Sebo, S. S. & Scassellati, B. *Personalized Robot Tutoring Using the Assistive Tutor POMDP (AT-POMDP) in Proceedings of the AAAI Conference on Artificial Intelligence* **33** (2019), 8050–8057.
11. Yılmaz, H. B. On the development and measurement of spatial ability. *International Electronic Journal of Elementary Education* **1**, 83–96 (2017).
12. Keren, G., Ben-David, A. & Fridin, M. *Kindergarten assistive robotics (KAR) as a tool for spatial cognition development in pre-school education in 2012 IEEE/RSJ international conference on intelligent robots and systems* (2012), 1084–1089.
13. Linn, M. & Petersen, A. Emergence and Characterization of Sex Differences in Spatial Ability: A Meta-Analysis. *Child development* **56**, 1479–98 (Jan. 1986).
14. Piaget, J. & Inhelder, B. *The growth of logical thinking from childhood to adolescence: An essay on the construction of formal operational structures* (Routledge, 2013).

15. Cooperau, L. A. & Shepard, R. N. The time required to prepare for a rotated stimulus. *Memory & Cognition* **1**, 246–250 (1973).
16. Shepard, R. N. & Metzler, J. Mental rotation of three-dimensional objects. *Science* **171**, 701–703 (1971).
17. McGee, M. G. *Human spatial abilities: Sources of sex differences* (Praeger, 1979).
18. Davis, B., Group, S. R. S., et al. *Spatial reasoning in the early years: Principles, assertions, and speculations* (Routledge, 2015).
19. Uttal, D. H., Meadow, N. G., Tipton, E., Hand, L. L., Alden, A. R., Warren, C. & Newcombe, N. S. The malleability of spatial skills: A meta-analysis of training studies. *Psychological bulletin* **139**, 352 (2013).
20. Piaget, J. & Inhelder, B. The child's conception of space. *FJ Langdon & JL Lunzer, trans.* London: Routledge & Kegan Paul (1956).
21. Huttenlocher, J., Newcombe, N. & Vasilyeva, M. Spatial scaling in young children. *Psychological Science* **10**, 393–398 (1999).
22. Frick, A. & Wang, S. *Round and round she goes: Effects of hands-on training on mental rotation in 13-to 16-month-olds in Poster presented at the XVIIth Biennial International Conference on Infant Studies, Baltimore* (2010).
23. Harris, J., Newcombe, N. S. & Hirsh-Pasek, K. A new twist on studying the development of dynamic spatial transformations: Mental paper folding in young children. *Mind, Brain, and Education* **7**, 49–55 (2013).
24. Crescentini, C., Fabbro, F. & Urgesi, C. Mental spatial transformations of objects and bodies: Different developmental trajectories in children from 7 to 11 years of age. *Developmental Psychology* **50**, 370 (2014).
25. Super, D. & Bachrack, P. *Scientific careers and vocational development theory*. New York: Teachers College, Columbia University 1957.
26. Wai, J., Lubinski, D. & Benbow, C. P. Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of educational Psychology* **101**, 817 (2009).
27. Feigenson, L., Dehaene, S. & Spelke, E. Core systems of number. *Trends in cognitive sciences* **8**, 307–314 (2004).
28. Piazza, M., Pica, P., Izard, V., Spelke, E. S. & Dehaene, S. Education enhances the acuity of the nonverbal approximate number system. *Psychological science* **24**, 1037–1043 (2013).
29. Clements, D. H., Battista, M. T., Sarama, J. & Swaminathan, S. Development of students' spatial thinking in a unit on geometric motions and area. *The Elementary School Journal* **98**, 171–186 (1997).
30. Manger, T. & Eikeland, O.-J. The effects of spatial visualization and students' sex on mathematical achievement. *British Journal of Psychology* **89**, 17–25 (1998).
31. Dearing, E., Casey, B. M., Ganley, C. M., Tillinger, M., Laski, E. & Montecillo, C. Young girls' arithmetic and spatial skills: The distal and proximal roles of family socioeconomics and home learning experiences. *Early Childhood Research Quarterly* **27**, 458–470 (2012).

32. Casey, B. M., Lombardi, C. M., Pollock, A., Fineman, B. & Pezaris, E. Girls' spatial skills and arithmetic strategies in first grade as predictors of fifth-grade analytical math reasoning. *Journal of Cognition and Development* **18**, 530–555 (2017).
33. Zhang, X., Koponen, T., Räsänen, P., Aunola, K., Lerkkanen, M.-K. & Nurmi, J.-E. Linguistic and spatial skills predict early arithmetic development via counting sequence knowledge. *Child development* **85**, 1091–1107 (2014).
34. Hawes, Z., Moss, J., Caswell, B., Naqvi, S. & MacKinnon, S. Enhancing Children's Spatial and Numerical Skills through a Dynamic Spatial Approach to Early Geometry Instruction: Effects of a 32-Week Intervention. *Cognition and Instruction* **35**, 236–264. eprint: <https://doi.org/10.1080/07370008.2017.1323902> (2017).
35. Carr, M., Alexeev, N., Wang, L., Barsed, N., Horan, E. & Reed, A. The development of spatial skills in elementary school students. *Child development* **89**, 446–460 (2018).
36. Casasola, M. The development of infants' spatial categories. *Current Directions in Psychological Science* **17**, 21–25 (2008).
37. Singer, M. A. & Goldin-Meadow, S. Children learn when their teacher's gestures and speech differ. *Psychological Science* **16**, 85–89 (2005).
38. Joh, A. S., Jaswal, V. K. & Keen, R. Imagining a way out of the gravity bias: Preschoolers can visualize the solution to a spatial problem. *Child Development* **82**, 744–750 (2011).
39. Siew, N. M., Chong, C. L. & Abdullah, M. R. Facilitating students' geometric thinking through Van Hiele's phase-based learning using tangram. *Journal of Social Sciences* **9**, 101 (2013).
40. Wolfgang, C., Stannard, L. & Jones, I. Advanced constructional play with LEGOs among preschoolers as a predictor of later school achievement in mathematics. *Early Child Development and Care* **173**, 467–475 (2003).
41. Pruden, S. M., Levine, S. C. & Huttenlocher, J. Children's spatial thinking: Does talk about the spatial world matter? *Developmental science* **14**, 1417–1430 (2011).
42. Kersh, J., Casey, B. M. & Young, J. M. Research on spatial skills and block building in girls and boys. *Contemporary perspectives on mathematics in early childhood education*, 233–251 (2008).
43. Tepylo, D. H., Moss, J. & Stephenson, C. A developmental look at a rigorous block play program. *YC Young Children* **70**, 18 (2015).
44. Levine, S. C., Ratliff, K. R., Huttenlocher, J. & Cannon, J. Early puzzle play: a predictor of preschoolers' spatial transformation skill. *Developmental psychology* **48**, 530 (2012).
45. Belpaeme, T., Kennedy, J., Ramachandran, A., Scassellati, B. & Tanaka, F. Social robots for education: A review. *Science robotics* **3**, eaat5954 (2018).
46. Köse, H., Uluer, P., Akalın, N., Yorgancı, R., Özkul, A. & Ince, G. The effect of embodiment in sign language tutoring with assistive humanoid robots. *International Journal of Social Robotics* **7**, 537–548 (2015).
47. Kennedy, J., Baxter, P. & Belpaeme, T. Comparing robot embodiments in a guided discovery learning interaction with children. *International Journal of Social Robotics* **7**, 293–308 (2015).

48. Spaulding, S., Gordon, G. & Breazeal, C. *Affect-aware student models for robot tutors in Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems* (2016), 864–872.
49. Hood, D., Lemaignan, S. & Dillenbourg, P. *When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting in Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (2015), 83–90.
50. Ramachandran, A., Litoiu, A. & Scassellati, B. *Shaping productive help-seeking behavior during robot-child tutoring interactions in 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2016), 247–254.
51. Edwards, L. D. Children’s learning in a computer microworld for transformation geometry. *Journal for Research in Mathematics Education*, 122–137 (1991).
52. Chang, K.-E., Sung, Y.-T. & Lin, S.-Y. Developing geometry thinking through multimedia learning activities. *Computers in Human Behavior* **23**, 2212–2229 (2007).
53. Rafi, A., Anuar, K., Samad, A., Hayati, M. & Mahadzir, M. Improving spatial ability using a Web-based Virtual Environment (WbVE). *Automation in construction* **14**, 707–715 (2005).
54. Fridin, M. & Belokopytov, M. Robotics agent coacher for CP motor function (RAC CP Fun). *Robotica* **32**, 1265–1279 (2014).
55. Narciss, S. The impact of informative tutoring feedback and self-efficacy on motivation and achievement in concept learning. *Experimental Psychology* **51**, 214–228 (2004).
56. Narciss, S. *Informatives tutorielles Feedback: Ableitung und empirische Überprüfung von Entwicklungs- und Evaluationsprinzipien auf der Basis instruktionspsychologischer Erkenntnisse* (Waxmann Verlag, 2005).
57. Narciss, S. Designing and evaluating tutoring feedback strategies for digital learning. *Digital Education Review*, 7–26 (2013).
58. Narciss, S. Feedback in instructional contexts. *Encyclopedia of the Learning Sciences, Volume F (6)*, 1285–1289 (2012).
59. Van der Kleij, F. M., Feskens, R. C. & Eggen, T. J. Effects of feedback in a computer-based learning environment on students’ learning outcomes: A meta-analysis. *Review of educational research* **85**, 475–511 (2015).
60. Narciss, S., Sosnovsky, S., Schnaubert, L., Andrès, E., Eichelmann, A., Goguadze, G. & Melis, E. Exploring feedback and student characteristics relevant for personalizing feedback strategies. *Computers & Education* **71**, 56–76 (2014).
61. Parvez, S. M. & Blank, G. D. *Individualizing tutoring with learning style based feedback in International Conference on Intelligent Tutoring Systems* (2008), 291–301.
62. Ferreira, A., Moore, J. D. & Mellish, C. A study of feedback strategies in foreign language classrooms and tutorials with implications for intelligent computer-assisted language learning systems. *International Journal of Artificial Intelligence in Education* **17**, 389–422 (2007).
63. Gupta, A., Fox, D., Curless, B. & Cohen, M. *DuploTrack: a real-time system for authoring and guiding duplo block assembly in Proceedings of the 25th annual ACM symposium on User interface software and technology* (2012), 389–402.

64. Jones, J., Hager, G. D. & Khudanpur, S. *Toward computer vision systems that understand real-world assembly processes* in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2019), 426–434.

Johns Hopkins University
 Baltimore, Maryland 21218, USA
 (+1) 410.508.5727

amahmo11@jhu.edu

<https://amamamahmood.wixsite.com/website>

EDUCATION AND DEGREES

2018–Present Graduate student, Fulbright Scholar Laboratory for Computational Sensing + Robotics (LCSR)

Johns Hopkins University

2013–2017 Undergraduate student, Department of Electrical Engineering, National University of Sciences and Technology

RESEARCH EXPERIENCE

Johns Hopkins University, Intuitive Computing Lab

LCSR and Department of Computer Science

(August 2019–Present)

Graduate Student in the laboratory of Dr. Chien-Ming Huang

- Developed a perception system to track user's progress in 3D block building spatial task.
- Developed feedback system to assist user in completion of 3D block building task.
- Designed robot behaviour.

Johns Hopkins University, Teleoperation group

LCSR and Department of Computer Science

(March 2019–December, 2019)

Graduate Student in the laboratory of Dr. Peter Kazanzides

- Worked on satellite servicing mission project.
- Used computer vision techniques on video stream of blade cutting through multilayer insulation hat on the satellite body to get an estimate of forces acting on blade.

Signal, Image and Video Processing lab at Lahore University of Management Sciences, Lahore, Pakistan
(October 2017–July 2018)

Post-baccalaureate Researcher in the laboratory of Dr. Nadeem Khan

- Worked on applications of brain computer interfacing.
- Presented feasibility analysis of existing multi-class motor imagery systems for real-time applications.

Neuro-informatics lab, SEECS-NUST, Pakistan

(June 2016–September 2017)

A previous position in the laboratory of Dr. Awais Mehmood Kamboh

- Worked on brain computer interface to drive a telepresence robot with motor imagery EEG commands. This assembly can be used as a stroke rehabilitation system.

Sigma lab, SEECS-NUST, Pakistan

(Feb 2016–July 2017)

A previous position in the laboratory of Dr. Hassan Aqeel

- Worked on Machine Learning and Automatic Speech Recognition (ASR) using Kaldi.

SMART lab, SEECS-NUST, Pakistan

(November 2015–June 2017)

A previous position in the laboratory of Dr. Osman Hasan

- Worked on SmartSIM, a virtual reality simulator for training in laparoscopic surgery.

- Worked on Phantom Omni, a forced feedback haptic device.

KEY SKILLS

Programming

Python, C, C++, MATLAB, Verilog HDL, G, Assembly and Embedded C for Microcontrollers

Simulation Gazebo, Rviz, Cadence, Simulink, Orcad Pspice, AutoCAD, Proteus, Keil, Xilinx, MPLAB, Arduino, ADS, OpenVibe

Miscellaneous ROS, Microsoft Windows, Microsoft Office, Ubuntu, LaTeX, Inkscape

TEACHING EXPERIENCE

Johns Hopkins University, LCSR and Department of Computer Science (January 2020-May 2020)

Teaching Assistant

- Graduate Student Instructor for Human-Robot Interaction

PUBLICATIONS

[Classification of Multi-Class Motor Imagery EEG using Four Band Common Spatial Pattern](#) **A. Mahmood**, R. Zainab, R.B. Ahmad, M. Saeed and A.M. Kamboh. *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2017).

[SmartSIM - A Virtual Reality Simulator for Laparoscopy Training using a Generic Physics Engine](#) Z.A. Khan, N. Kamal, A. Hameed, **A. Mahmood**, R. Zainab, B. Sadia, S.B. Mansoor and O. Hasan. *The International Journal of Robotics and Computer Aided Surgery* **13**. e1771 (2017).