10-2020

Gesture enhanced comprehension of ambiguous human-to-robot instructions

WEERAKOON MUDIYANSELAGE DULANGA KAVEESHA WEERAKOON
Singapore Management University, mweerakoon.2019@phdcs.smu.edu.sg

Vigneshwaran SUBBARAJU

Nipuni KARUMPULLI

Minh Anh Tuan TRAN
Singapore Management University, tuantran@smu.edu.sg

Qianli XU

See next page for additional authors

Gesture Enhanced Comprehension of Ambiguous Human-*to*-Robot Instructions

Dulanga Weerakoon
Singapore Management University
mweerakoon.2019@phdcs.smu.edu.sg

Vigneshwaran Subbaraju
Institute of High Performance Computing
vigneshwaran_subbaraju@ihpc.a-star.edu.sg

Nipuni Karumpulli
Singapore University of Technology and Design
nipuni_karumpulli@sutd.edu.sg

Tuan Tran
Singapore Management University
tuantran@smu.edu.sg

Qianli Xu
Institute for Infocomm Research, Singapore
qxu@i2r.a-star.edu.sg

U-Xuan Tan
Singapore University of Technology and Design
uxuan_tan@sutd.edu.sg

Joo Hwee Lim
Institute for Infocomm Research, Singapore
joohwee@i2r.a-star.edu.sg

Archan Misra
Singapore Management University
archanm@smu.edu.sg

ABSTRACT

This work demonstrates the feasibility and benefits of using pointing gestures, a naturally-generated additional input modality, to improve the multi-modal comprehension accuracy of human instructions to robotic agents for collaborative tasks. We present *M2Gestic*, a system that combines neural-based text parsing with a novel knowledge-graph traversal mechanism, over a multi-modal input of vision, natural language text and pointing. Via multiple studies related to a benchmark table top manipulation task, we show that (a) *M2Gestic* can achieve close-to-human performance in reasoning over unambiguous verbal instructions, and (b) incorporating pointing input (even with its inherent location uncertainty) in *M2Gestic* results in a significant (~ 30%) accuracy improvement when verbal instructions are ambiguous.

CCS CONCEPTS

• **Human-centered computing** → **Pointing**; **Gestural input**; *Empirical studies in HCI*; • **Computer systems organization** → *Robotics.*

KEYWORDS

Multimodal Human Robot Interaction; Pointing Gesture; Ambiguity; Table-top manipulation

ACM Reference Format:
Dulanga Weerakoon, Vigneshwaran Subbaraju, Nipuni Karumpulli, Tuan Tran, Qianli Xu, U-Xuan Tan, Joo Hwee Lim, and Archan Misra. 2020. Gesture Enhanced Comprehension of Ambiguous Human-*to*-Robot Instructions. In *Proceedings of the 2020 International Conference on Multimodal Interaction (ICMI '20), October 25–29, 2020, Virtual event, Netherlands.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3382507.3418863

ICMI '20, October 25–29, 2020, Virtual event, Netherlands

ACM ISBN 978-1-4503-7581-8/20/10…$15.00
https://doi.org/10.1145/3382507.3418863

1 INTRODUCTION

The adoption of collaborative robots that interact and share a common working space along with human worker(s) has generated heightened interest in developing machine comprehension techniques for natural human-robot interaction. The ability of such a robot to comprehend natural human instructions is critical for seamless human-robot co-working. For example, a worker may instruct an assistive robot to '*pick up that red wrench*' from a clutter of several similar objects on a table. In such situations, human workers typically communicate intent via a mixture of multiple modalities, such as sight, speech *and gestures* (e.g., pointing). Therefore, supporting such natural human-robot interaction will require machine comprehension techniques that are *multi-modal*. Recent efforts on visual search and reasoning systems (e.g., [8, 22]) have explored the possibility of combining visual scene analysis with text understanding, albeit within fairly unambiguous task contexts. In contrast, we consider the possibility of multi-modal instruction comprehension for the *collaborative table top manipulation* task, where a robot attempts to interpret ambiguous "target acquisition" commands issued by a human. Using a benchmark dataset, Scalise et al. [19] assessed *human performance* in both generating and interpreting such visual perception-driven, natural language text instructions and demonstrated the challenge of *instructional ambiguity*. A sample table-top block-setup from this dataset can be seen in Fig 1 (on the left). Scalise et al. found that ambiguity resulting from the visual scene (e.g., many blocks with same attributes are closely packed) or imprecise perspective (e.g., does 'left' refer to your or my left?) affects accurate human comprehension of such language instructions.

In natural human communication, such verbal *instructional ambiguity* is often resolved via an accompanying gesture (e.g., *pointing*). Therefore, in this paper we explore the design of machine comprehension techniques that tackle ambiguous table-top manipulation instructions, by incorporating both *pointing gestures* and natural language text. Our premise is that a simple pointing gesture, overlaid on top of verbal and visual/scene analysis, can help reduce ambiguity significantly. While we confine our investigations to this table-top scenario, we strongly believe that, given past evidence on the importance of non-verbal cues, such as gestures, gaze & posture,

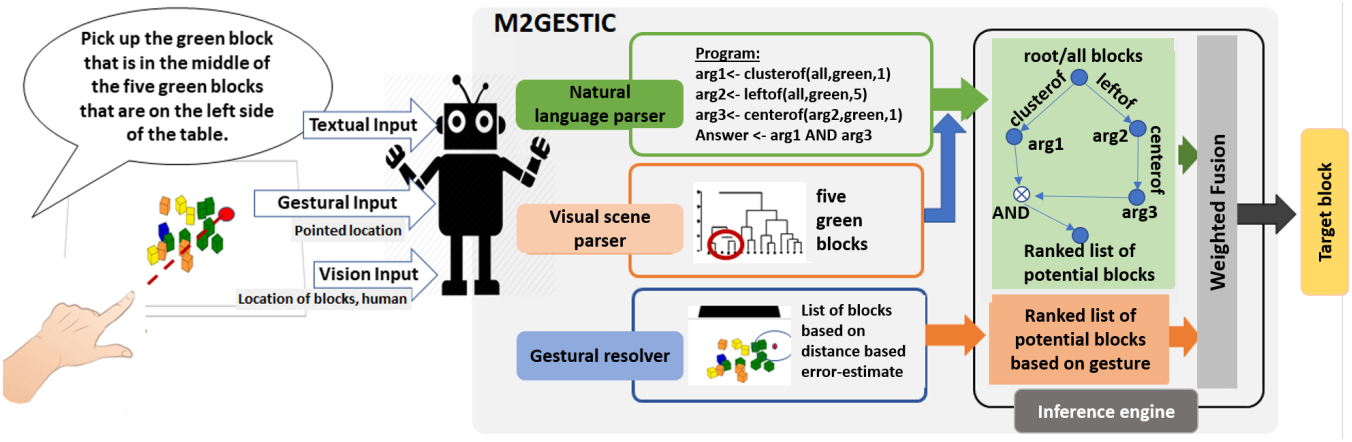

Figure 1: *M2Gestic*: System Components & Functionality

in human-robot collaboration (e.g., [3] which looked at non-verbal responses *by* a social robot), our work has broader significance by *demonstrating the feasibility and benefits of incorporating gestural inputs in comprehending such ambiguous human→robot instructions.* We must, however, address two challenges:

(i) *Lack of Gestural Precision:* The pointing gesture itself is unlikely to be exact—humans may make an error between the pointed location and that of the intended target [4]. Additionally, it is likely that such errors may increase with the human-to-target distance.

(ii) *Multi-Modal Fusion:* For automated machine comprehension, we will need a consistent mechanism to identify a target object given multiple sensory inputs. In particular, we must find a way to (a) parse and extract relevant spatial and/or descriptive attributes from the verbal command and use those attributes to *reason* over the object-level attributes (e.g., location, color, shape) provided by AI-based vision techniques, and (b) combine such reasoning with the potentially-erroneous, pointing-based spatial cues.

We address these challenges by (a) quantifying the nature of pointing-driven error in representative tasks and (b) developing a novel target selection mechanism that creates and parses a knowledge graph structure, based on multi-modal attributes generated by state-of-the-art deep learning techniques.

Key Contributions: Our work will demonstrate that, in spite of inevitable errors, the combination of pointing gestures and natural language text can lead to a significant improvement (~ 30% for robotic agents, and ~ 5% for human subjects) in the accuracy of comprehending ambiguous human-to-robot instructions in our benchmark table top dataset [19]). We make the following key contributions:

- *Develop a Multi-modal Target Selection Algorithm:* We describe a first-of-a-kind hybrid approach AI-cum-knowledge graph based technique for instruction comprehension, called *M2Gestic* (**M**ulti-**M**odal **Gest**ure-enhanced **I**nstruction **C**omprehension System, pronounced 'majestic'). *M2Gestic* combines (a) a neural (RNN-based) approach to automatically generate machine-understandable selection commands from natural language instructions, (b) a vision-based hierarchical clustering mechanism to represent salient spatial relationships under varying levels of clutter, and (c) a fusion mechanism that additionally ranks the 'fit' of objects based on their spatial alignment with the potentially-erroneous pointing location.
- *Quantify & Accommodate Pointing Gesture Error:* Through detailed empirical in-the-lab studies, we quantify the range of human error associated with natural pointing gestures. More specifically, we show that the distance error (at the table-top) increases non-linearly as a function of the human instructor's distance from the object (mean pointing error= 23.4 pixels at a distance of 88 cm, increases to 155.8 pixels at 264cm), which can imperil the usefulness of pointing input.
- *Establish the Efficacy of* M2Gestic*-based Comprehension, both with and without Pointing:* Using the benchmark table-top manipulation dataset [19], we first show that agent-based comprehension using text-only instructions (no pointing gestures) can achieve 61.12% accuracy in target-selection, compared to 73.64% accuracy previously reported for human respondents in [19]. Subsequently, using a series of in-the-lab and crowd-sourced studies, we demonstrate how the incorporation of *pointing input* (along with verbal and visual comprehension) helps improve this comprehension accuracy. From the realistic studies conducted using Amazon Mechanical Turk [2], with 622 respondents and 4200 unique task instances, we show that pointing input from a close distance enhances human comprehension accuracy from 73.64% to 77.5%, but exhibits a ~5% drop when the instructor-object distance increases. For the *M2Gestic*-based AI/robotic agent, the comprehension accuracy of automated multi-modal comprehension (under empirically-derived distributions of pointing error) on the entire dataset improves from 61.12% to 74.75% when the instructor is close to the objects. Moreover, this comprehension improvement is dramatic (30%) for the ambiguous verbal instructions. Finally, we show how a distance-weighted variant of *M2Gestic* provides *robustness*, ensuring that *M2Gestic's* performance, while suffering degradation, does not drop below the no-gesture baseline even when the instructor is at larger distances (implying larger spread of pointing error).

2 RELATED WORK

Comprehending natural language instructions: The task of 'grounding' natural language instructions in robotics involves parsing the instruction to extract phrases and assigning a direct meaning to them in the context of the real-world perceived by the robot. Paul et al. [17] have introduced probabilistic models to achieve grounding in a table-top manipulation setup involving objects that can handle spatial references and abstract concepts of cardinality (group of 2 blocks on the left) and ordinality (2^{nd} block from the left). Interactive dialog based approaches have been considered for disambiguation of natural language instruction by the INGRESS system in [20] and the interactive text2pickup network in [1]. Reasoning based systems, that recognize individual objects in the scene and perform high level reasoning to answer verbal questions (e.g., answering "how many blue blocks are behind the red block?") have been proposed in [8, 22]. However, the instructions studied in these works are primarily uni-modal and they do not consider the perspective and scene ambiguity challenges described earlier.

Ambiguity in natural language instructions: Using a carefully selected set of table-top block arrangements that induce varying levels of ambiguity, Scalise et al. [19] collected a large set of human generated instructions to pick a particular block. They then quantified the effect of instructional ambiguity by asking other human subjects to interpret these instructions. Using this publicly-released, 'collaborative manipulation corpus', [10] showed that instructions that do not involve perspective references suffer from poor human comprehension. Our work goes beyond such human-based comprehension by building and evaluating a machine comprehension technique.

Pointing Gestures in robotics/HCI: Mayer et al. [15] performed comprehensive studies on the accuracy of several ray-casting based pointing models, by asking several users to point at a target location on a large screen placed about 2-3 meters away. They found that a simple offset correction [14] can improve the accuracy in both virtual and real-worlds. Gromov et al. [5] recently demonstrated the ability to use pointing gestures to guide a drone towards a target location. However, Herbort and Kunde [6] have shown that a second human consistently fails to determine the exact pointing location intended by an individual, and that such determination is made by nonlinear extrapolation of the pointer's arm-finger line. Further, unlike our exemplar scenario, these works assume an unambiguous *clutter-free* setup. Our work also differs from the above by using cross-modal information (pointing, text and visual scene analysis) to achieve accurate target selection, instead of relying solely on accurate ray-casting models.

Multi-modal instruction grounding (natural language + pointing): A system that combines gestures and natural language for interpreting object references in a table-top setup was proposed by Matuszek et al. [13]. But in this study, the gestures were performed from just inches away from the target object. More recently, Whitney et al. [21] proposed a real-time system that can identify one of four objects in a table-top setting involving common kitchen items, by combining the language references such as *hand me the bowl* as well as a pointing gesture which were performed from a few feet away. Kennington and Schlangen [9] proposed a system incremental resolution of multi-modal instructions (verbal + pointing). However, since the speaker generates the instructions from the same point-of-view as the listener, they do not consider the ambiguity due to perspective, which is highlighted as an important factor by Scalise et al. [19]. Thus, the language as well as the table-top setup in Scalise et al. [19], is much more ambiguous or cluttered than the other previous works, and hence we choose to build upon the work by Scalise et al. [19] to address the challenges of accommodating ambiguity in natural multi-modal instructions.

3 PRELIMINARIES

We use the collaborative manipulation experimental setup [19] as our canonical use-case. This setup involves 28 different images of block arrangements (a typical block arrangement is illustrated in the top-left part of Figure 1), from which a *single* target block needs to be identified. The arrangements and the corresponding target-blocks have been designed to generate different forms of ambiguity when human subjects generate verbal instructions to pick up the target block. In the data published by [19], each setup is also accompanied by a set of 50 different human-generated natural language text 'pickup' instructions to pick-up the target block. However, no gesture-related data is included.

Our primary goal is to develop an automated system, *M2Gestic*, which combines verbal reasoning over visual content with accompanying pointing gestures for enhanced comprehension of multi-modal 'pickup' instructions. Note that *M2Gestic* does not aim to improve the technology for accurate tracking of the pointing gesture; nor does it focus on techniques for conversion of audio to textual input or performing object detection. Instead, it assumes the use of state-of-the-art systems to perform these perceptual tasks.

Figure 1 illustrates the components and overall workflow of *M2Gestic*. At a high-level, it consists of the following components (detailed descriptions are deferred to Section 4): (i) The *Visual Scene Parser* processes the table-top image (consisting of the blocks) to create a multi-attribute representation of the objects; (ii) the *Natural Language Parser* similarly processes the textual instruction, converting it into a set of machine-understandable primitives; (iii) the *Gesture Resolver* uses the pointing gesture to identify a subset of candidate blocks (based on the distance from the table) on the table-top surface, while the (iv) *Multi-Modal Inference Engine* fuses the inputs from these 3 previous components to perform target selection.

3.1 Empirical User Studies

To design and evaluate such a system, we shall utilize the following experimental studies (using a setup similar to [19]).

Study 1: Characterizing Pointing Gestures: We conduct this study (detailed in Section 3.2) to gauge the error characteristics of pointing gestures performed by human subjects, especially as a function of the distance between the human instructor and the target.

Study 2: Baselining Human Performance with Pointing Enhanced Instructions: - In this study (detailed in Section 5.2), a virtual 3D environment was developed to recreate the same ambiguous table-top setups used in [19] and a human avatar is shown performing a pointing gesture (with zero pointing error) towards

Figure 2: User performing a pointing gesture towards a target block (in the projected image) using HTC VIVE

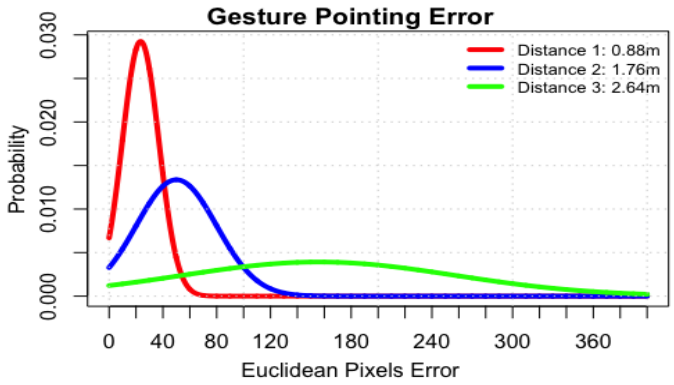

Figure 3: Pointing error distribution vs. screen-human distance

the desired block from a distance. We thus created a new dataset of images that show the robot's view of the setup where both the pointing gesture as well as the blocks arrangement are visible. These images were provided to human subjects (recruited via the Amazon Mechanical Turk crowdsourcing platform) to infer the correct target-block by combining inputs from the text instructions with additional gestural (pointing) input. This serves as a upperbound of human comprehension capability (under idealized zero pointing error) against which to compare *M2Gestic's* performance.

We additionally employ the following experimental evaluations to compare *M2Gestic's* performance against these human baselines: (1) *Automated Comprehension without Gestures*: we evaluated the ability of the *M2Gestic* system to combine its Visual Scene Parser and Natural Language Parser to choose the right target-block, for each of the 28 images and corresponding 50 text instructions in the original dataset [19]; (2) *Automated Gesture-Enhanced Multi-Modal Comprehension:* we evaluate the improvement in block selection accuracy achieved by *M2Gestic* in the presence of such noisy *synthetically-generated pointing input.*

3.2 Study 1: Characterizing Pointing Gestures

Given our high-level goal of incorporating pointing input for multimodal instruction comprehension, we first study & characterize the nature of human pointing input. In this study (illustrated in Figure 2), the images of each of the 28 block setups were projected on a screen and 15 human subjects were asked to perform a pointing gesture towards the specified target block using a calibrated HTC VIVE system [7]. A calibrated HTC VIVE is known to be a highly accurate (error $\leq$ 0.02cm) [16] in tracing the pointed location. This system uses two pre-calibrated cameras that help provide information regarding the pointed location on the screen. For the experiments, we set the two cameras 5 meters apart. The pointing gestures were performed by the subjects from three specifically marked positions (denoted p_1, p_2, p_3) in the room along a straight-line drawn from the center of the screen, that were $d = \{88cm, 176cm, 264cm\}$, respectively, away from the screen. We chose these 3 distances because of the technical limitations of the VIVE System: the two VIVE cameras need to be separated by a diagonal distance of less than 5m to ensure that the VIVE controller is track-able by the cameras. Given this limitation, we chose 3 equidistant points between the screen and the maximal distance (264cm) that allows the controller to remain detectable. To provide operational familiarity with, and perceptual calibration on, the VIVE system, each subject had a training period (of a few minutes) where the pointing cursor was 'on'–i.e., the subjects could receive real-time visual feedback about the pointed location on the screen–and were asked to specifically 'target' the 4 edges of the screen. To mimic the real-world environment (such as a smart factory floor) where the human instructor will not have any such visual feedback, the cursor was, however, disabled during the actual 'pointing' study. The image-setups were shown in randomized order and each image-setup was shown to each subject thrice. Thus a total of $15 \times 3 \times 28 = 1260$ pointing gesture data were collected for each position p_i.

Fig 3 shows the probability distribution of the gesture pointing error ($\delta_i, i = \{1, 2, 3\}$) across all users, for the 3 distances $\{p_1, p_2, p_3\}$. We observe that: (a) the average pointing error (in pixels) increases non-linearly with increasing d (average error= 23.43 pixels, 50.05 pixels, 155.76 pixels at $d = 88cm, 176cm, 264cm$ respectively), and (b) error variance increases with d as well. Additionally, we found that the average *error angle*, subtended at the human's location, was $< 3^\circ$, across all 3 distances. We can thus conclude: M2Gestic's *Inference Engine must be able to tolerate moderate errors in the instructor's pointing input, with the likelihood of such error being higher at greater human-table distances.*

4 SYSTEM OVERVIEW

In this section, we describe the detailed design of the 4 key functional components of *M2Gestic* (illustrated in Figure 1).

4.1 Visual Scene Parser

The *visual scene parser* is responsible for generating a representation of the relative positions, and selected attributes, of the various objects in the image-setup. In our experimental setup, the objects in the scene are the blocks, the robot, the table and the human instructor. The blocks are all cube-shaped and have one of the four colours (green, blue, yellow or orange). The objects in this setup are fairly simple to detect using standard computer vision methods (e.g., using YoLo [18] or SSD [11]) and is not the subject of this paper. We assume that we know the position of the center of all the objects. However, as mentioned in [19], the natural language instructions generated by human subjects often contain hybrid 'density-based'

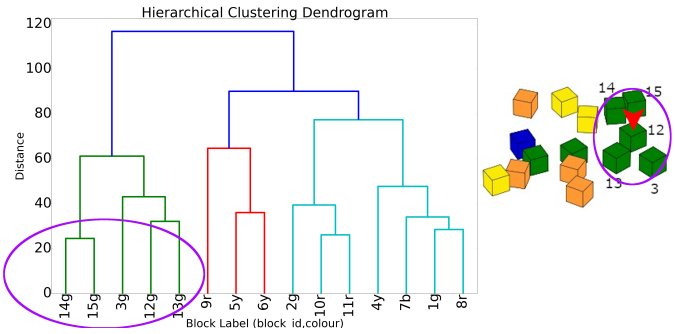

Figure 4: Hierarchical clustering dendogram to identify clusters of objects

references such as "the blue cluster in the middle", "the three blocks near you" etc. which require a hierarchical understanding of the objects in the scene. Therefore, we use a hierarchical agglomerative clustering approach to enable understanding of such phrases. To achieve this hierarchical representation, the distance between each object pair, calculated from the co-ordinates of the centers, is used to perform agglomerative clustering. (In addition to such clustering, the visual parser annotates each object with its 'color' & other relevant attributes such as shape or texture.) As an illustrative example, consider the dendrogram shown in Fig 4. Now consider the phrase "5 green blocks that are on the left side of the table". From the dendrogram, the 5 marked blocks that are potentially referred to by this phrase can be immediately identified. Besides the hierarchical clustering, the parser also uses standard spatial reasoning techniques to create additional knowledge representations that capture: (a) the *perspective* information (e.g., closest/furthest/leftof/rightof from me/you), and (b) the *relational* information (e.g., pair-wise object distance, objects in the center, etc).

4.2 Natural language text parser

The natural language parser is responsible for converting the human-generated text instructions, describing the specific object to be picked up, into a computer program consisting of predefined functions. As the instructions typically contain one or more spatial prepositions denoting the relative positions of objects, the pre-defined functions typically correspond to spatial relations such as *leftof, rightof*, etc. A list of all the pre-defined functions used in *M2Gestic* and their respective descriptions are given in table 1

As demonstrated in [22], neural network techniques can be used to convert the entire natural language instruction into a sequence of such pre-defined functions. This is illustrated by the following simple natural language instruction, *"Please grab the yellow block that is the second from your far left"*. The corresponding structured program, which has just a single function call in this case, would be **farleft(you,yellow,2)**. In this example, *farleft(. . .)* is a function in our robot's command vocabulary. The first input parameter 'you' refers to the perspective (i.e your left vs. my left). The second parameter 'yellow' says that only yellow blocks need to be targeted. The third parameter '2' specifies that we are looking for the 2^{nd} yellow block. Upon execution, this function will return a ranked vector of 15 elements (each element corresponding to one of the 15 blocks in the scene), with a lower rank implying a closer match. In our example, the rank will be the lowest for the 2^{nd} yellow block from the left of the robot. Other yellow blocks will get their ranks based on how close they are to the 'far-left' of the black-figure, with all non-yellow blocks assigned a rank=16 (the highest distance rank). We call this ranked list as a *sub-scene*, because this intermediate representation filters/prioritizes the blocks from the original/reference scene for subsequent operations. For recursive application, each of the input parameters may actually be specified as a previously-computed sub-scene. Thus, each function in our robot's vocabulary, is designed to take 3 input arguments, viz. (a) perspective/reference subscene, (b) target subscene and (c) rank/number. The output of the function is another subscene, that rank blocks based on this function's logic.

Table 1: A list of all the pre-defined functions and their descriptions

Pre-defined function	Function description
closest/furthest/nextto	find objects closest/furthest/next to
clusterof	find clusters of objects
leftof/rightof/topof/ bottomof/centerof/	find objects on left/right/top/ bottom/center
farleftof/farrightof	find objects from far left/far right

Now let us look at a more complex instruction - "*Grab the orange block that is furthest to the right and at the bottom beside a yellow block*", which results in multiple such structured robotic functions. The corresponding program is shown in (1).

$$\begin{aligned} farright(none, orange, 1) &= arg1 \\ bottomof(none, orange, 1) &= arg2 \\ and(arg1, arg2) &= arg3 \\ nextto(yellow, orange, 1) &= arg4 \\ and(arg3, arg4) &= ans \end{aligned} \tag{1}$$

The above example demonstrates the potentially recursive nature of such functions: output sub-scenes from a function may be used as the input subscene for another function (as exemplified by the two *and* functions).

Each such manipulation instruction can thus be converted into a sequence of functions, with additional {AND, OR, NOT} operators expressing the selection predicates. To establish a ground truth corpus, we first manually converted each of the 1400 instructions in the dataset into such programs. Subsequently, we trained a neural network model, as part of *M2Gestic's* 'natural language parser' component, to generate such structured program syntax automatically from the natural-language text instructions. Inspired by state-of-the-art DNN-based machine translation techniques, we use the Attentional Recurrent Neural Network proposed in [12] to perform such a *sequence-to-sequence* mapping. Of course, such training requires a large training dataset. As the original collaborative manipulation corpus has just 1400 instructions, we *augmented* this dataset with additional instructions that are synthetically generated by changing the colours, perspective, words, phrases and instruction type. We also combined simple instructions in the original dataset to add more complex instructional examples to this training dataset.

Note that the original dataset contains several examples of ambiguous instructions that are typical of human-human conversations. For example, the instruction "*Please grab the yellow block that*

is the furthest" suffers from perspective ambiguity: the target block could be the furthest yellow block **from the user** or **from the robot**. Similarly, the instruction "*Please pickup the topmost block*" shows ambiguity in both perspective as well as color attributes of the target block.

4.3 Gesture Resolver

We use state-of-the-art gesture/pose tracking systems to help track the arm movement/pose of the instructor's limb, and obtain an estimate of the table-top location of the pointing gesture. Based on this table-top location, we derive a gesture-based subscene, which is a ranked list for blocks based on the distance from the pointed position on the table-top. It is important to distinguish between two distinct sources of pointing error: (a) the *intrinsic instructional error*, which arises from the fact that a human is unable to direct his pointing gesture *precisely* at the object that he intends to target, and (b) the *pointing tracking error*, which arises from the limitation/inaccuracy of the tracking technology. Note that *M2Gestic's* logic is independent of (b), and is primarily concerned with accommodating the error arising out of intrinsic human limitations. For our current implementation of *M2Gestic*, we utilize a calibrated (HTC VIVE) [7] tracker to provide an estimate of the human instructor's pointed location. While there is clearly tracking error, the pose estimation error is usually very small ($\leq$ 1cm) in such well-calibrated systems.

4.4 Inference Engine

The outputs from the visual scene parser, the natural language text parser and the gesture resolver are provided to the Inference Engine, which makes a decision on the target block. The program generated by the text parser may be represented as a tree structure whose individual nodes represent a *sub-scene* and the edges represent one of AND/OR/NOT operations. The various knowledge representations provided by the visual parser (e.g., object clusters, perspective relationships, object attributes and object-pair relationships) are used to execute individual functions of the program, and thereby generate a *ranked sub-scene vector of objects* for each node on this AND-OR-NOT tree. By traversing this tree from the leaves to the root, we can compute the final composite ranking vector, denoting the relative *fit* of individual blocks to the original instruction. We combine two ranked sub-scenes using a linearly-weighted formula, illustrated below for the AND operator. In the Eq. (3), consider R^1 to be the ranking vector of sub-scene 1, R^2 be the ranking vector of sub-scene 2 and let R^{ret} be the subscene obtained by combining $R^1 \& R^2$ using the AND operator. Let $R^1 = \{R^1_1, R^1_2, \ldots R^1_k, \ldots R^1_m\}$ *and* $R^2 = \{R^2_1, R^2_2, \ldots R^2_k, \ldots R^2_m\}$ Then, the Rank-Sum s is given by

$$s = w_1 * R^1 + w_2 * R^2 \qquad \forall k \in [1, m] \tag{2}$$

For purpose of generality, we define w_1 and w_2 as weights given to each sub-scene. For the current implementation of the text parser, we consider all sub-scenes to be of equal importance (i.e, w_1=w_2=0.5). However, in future, it might be possible to assign importance to certain parts of the sentence (e.g., using attentional mechanisms). This ranking vector is used as an input to the subsequent subscenes. If this ranking vector represents the return subscene, final output can be inferred from the indices of the blocks with the lowest rank-sum, given by $k_{opt} = argmin(s)$. (If the instruction is ambiguous there could be multiple k_{opt} values, otherwise there exists only one k_{opt} value). Let $k_{opt} = [k^1_{opt}, k^2_{opt}, ..., k^l_{opt}]$. Therefore the final return vector R^{ret} as a result of the AND operation can be obtained as below,

$$\begin{gathered} Let\ R^{ret} = \{R^{ret}_1, R^{ret}_2, \ldots, R^{ret}_k, \ldots R^{ret}_m\}\ ; k \in [1, m] \\ if\ k \in k_{opt}\ then\ R^{ret}_k = 1\ else\ R^{ret}_k = 0 \end{gathered} \tag{3}$$

A similar approach is used for the OR operation as well.

Extension to Incorporate Pointing Information: We apply the same 'weighted' approach (introduced in Eq. 3) to fuse the knowledge from pointing gestures. Let R^l be the final ranking vector obtained from the text and vision parsers. Given a pointing location, we can similarly obtain another ranking vector, where the ranks are sorted by the distance of each block from the pointed location. Let R^g represent this gesture-driven ranking vector. We can then apply the same reasoning outlined in Equation (3)–i.e., first compute a linear weight $w * R^l + (1 - w) * R^g$ for each object, and then select the object with the lowest 'distance rank'.

M2Gestic's gesture-fusion technique, however, takes into account the increase in the pointing uncertainty/error with an increase in the instructor-object distance. Because the pointing uncertainty is lower when the user is closer (and vice versa), we use a larger value of w (reduced importance to the pointing input) when the instructor-object distance is larger, and vice versa. In Section 5.2), we shall see that this 'weighted technique' proves vital to ensuring that *M2Gestic's* comprehension accuracy proves robust (and outperforms human performance) even with increasing distance.

5 EVALUATION

We now present our evaluation results for instruction comprehension, comparing the automated *M2Gestic* system with the corresponding human perception performance, both with and without the added pointing input.

5.1 Text Instruction Understanding (No Gestures)

5.1.1 Accuracy of Text parser. We first evaluated the accuracy of the Attentional RNN-based technique for converting verbal instructions to *programs*. We trained the natural language text parser model on the *augmented* dataset with 80% − 20% train/test split and obtained an accuracy of 99.7% (the accuracy was slightly lower (95%) on the original data). Note that the augmented dataset did not include the original 1400 instructions. This confirms the ability of *M2Gestic's* RNN to convert the natural language input into accurate machine-readable programs.

5.1.2 Block Identification Accuracy. Then we evaluated the accuracy of the overall *M2Gestic* system, where its Inference Engine utilizes *only* the visual and text parsing pipelines. The original dataset also classified 1400 instructions as ambiguous (626) vs. unambiguous (774), based on whether more than one block in the scene potentially satisfies the instruction's combination of block or perspective predicates. Table 2 (specifically, the two columns categorized under 'No Gestures') provides the results for this scenario, both overall and under the presence/absence of ambiguity.

The results for human accuracy (73.62%, based on an Amazon Mechanical Turk study) are reproduced from [19]. We find that the automated *M2Gestic* approach achieves *human-comparable* performance (80.84%) for non-ambiguous instructions, but *exhibits dramatic performance degradation* (accuracy= 29.26%) *in the presence of instruction ambiguity.* Clearly, machine comprehension requires additional cues (specifically, pointing input) to tackle such real-world instructional ambiguity.

5.2 Multi-modal Understanding (With Gestures)

We next quantify the added benefits provided by the inclusion of pointing input about the *likely* location of the target block.

5.2.1 Study 2: Human Performance With Pointing Input. We first quantified the ability of humans (thus, both providing a competitive baseline for *M2Gestic*) to use the combination of pointing gesture information and text instructions to infer the target block.

Experimental setup: For this study, we used a virtual 3D environment (using Unity 3D) to simulate the same 28 table-top block arrangements in the original dataset. However, the original images represented the view-point of the human instructor. Since the multi-modal inference is performed by another agent on the opposite side of the table, we transformed the images to represent the perspective of the agent performing the comprehension task. Figure 5 provides an example of this transformed perspective, which includes the table-top objects, as well as the pointing gesture made by the human instructor (the avatar in the figure). We generated such views (corresponding to the 3 different distances used in Study 1), by fixing the instructor's height at (190 cm), the block size to (5 cm) and the agent's height at (200 cm). To generate accurate pointing input, we adjusted the pose of the pointing hand of the human instructor (the avatar in Figure 5) to first point *exactly* towards the intended block by using a Unity-provided ray tracing model that can track & visually illustrate the pointed location. Then we took a screen grab of the resulting scene, *as viewed by the agent performing the inference.* 622 human-subjects, recruited via the Amazon Mechanical Turk platform, were then asked to use these pointing-included images, along with the text instructions, to infer the target block (as illustrated in Figure 5). Participants were also asked an additional question *"Did you find the pointed location useful to identify the target block"*, with one of 3 possible answers, viz. {'Not at all', 'Useful, but I can do with just the text instruction', 'It was crucial'} to help understand how human subjects assign more/less importance to the pointing gesture. Note that the pointing input for these human studies had *no error*; accordingly, the human perceptual performance provides the baseline under the most-optimistic gestural context.

Demographics: Each of the 622 Amazon Mechanical Turk workers were asked to perform at least 25 HITs. From the data collected we rejected the 'low-quality' assignments (39 workers) that matched any one of the following criteria: (a) Reject if accuracy $< 30\%$; (b) Reject if number of HITs done by participant < 25; and (c) Reject if user selected multiple points/objects. After rejections, each image was annotated by an average of 7 workers. The workers were requested to provide three demographic details: 1) Age (73.51% in the $20-40$ age group), 2) Gender (46.5% male)and 3) Whether English is their first language (English=80.6%).

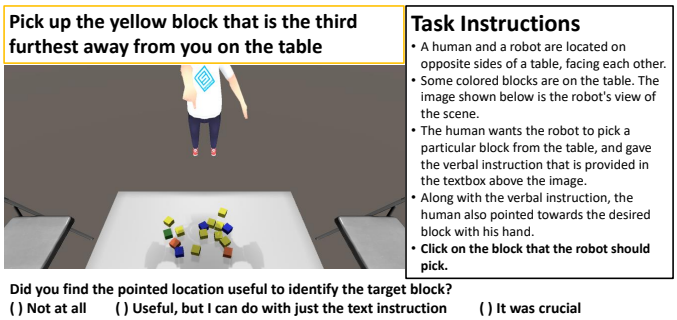

Figure 5: Study 2 - Setup used to study human performance in interpreting the instructions along with a gesture.

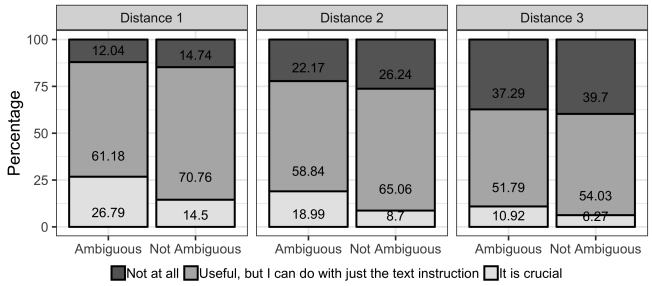

Figure 6: User perception of utility of pointing input

Human Performance: Table 2 (under the "With Gestures" heading) summarizes the average comprehension accuracy of both human subjects and *M2Gestic*. The last row of Table 2 provides the results when comprehension is performed *solely* using pointing input– i.e.,without parsing the text instruction. Even at a close distance of 88cm, the accuracy of human subjects in choosing the correct block based on pointing alone is very low (23.8%), when compared to using only the text instructions (73.64%); this accuracy drops by 10% when the instructor is 264cm away. Clearly, pointing gestures are insufficient for such cluttered table-top conditions, in contrast to earlier pointing based studies [5, 14], which use uncluttered setups.

More importantly, incorporating pointing input (in tandem with text parsing and visual scene analysis) improved human accuracy to 70.18% (5.3% higher than text parsing) for distance 1 (88cm away from the screen). However, human comprehension performance degrades with the instructor-object distance; in fact, at distance 3 (264 cm), the use of gestures actually causes selection accuracy to degrade below that achievable without gestural input! Clearly, while pointing can be beneficial, the 'blind' use of pointing input may be counter-productive if it is too noisy (performed from longer distances). The questionnaire responses, plotted in Figure 6, corroborate this insight: users perceived very low utility from the pointing information when the human instructor was farther away from the table ($d3 = 264$cm).

5.2.2 Performance of M2Gestic . We then evaluated the performance of *M2Gestic*, when its Inference Engine is provided the pointing data from study 1. *Note that* M2Gestic *does not, unlike the Amazon Mechanical Turk study, have an accurate pointing input, but assumes an error spread around the pointed table-top location.* For

Table 2: Potential improvement in accuracy of system using weighted inference scheme

	No Gestures		With Gestures					
	Accuracy (Text only)		Accuracy (d1=88cm)		Accuracy (d2=176cm)		Accuracy (d3=264cm)	
	Human	*M2Gestic*	Human	*M2Gestic*	Human	*M2Gestic*	Human	*M2Gestic*
Ambiguous Inst.	64.79%	29.26%	70.18%	60.73%	63.71%	42.37%	60.99%	29.26%
Unambiguous Inst.	80.79%	80.84%	83.29%	83.48%	83.89%	79.06%	78.88%	80.84%
All Inst.	73.64%	61.12%	77.50%	74.75%	74.88%	65.14%	70.93%	61.12%
Only pointing	–	–	23.78%	21.43%	11.04%	3.57%	10.0%	0%

M2Gestic, the target selection accuracy improves dramatically (to 60.73% for the ambiguous instructions, as opposed to a baseline of just 29.26%) when pointing (from distance $d1 = 88$cm) is used to augment the textual instructions. For gestures from distance $d2$, the accuracy improvement for ambiguous instructions is still significant (about 13% higher vs. text-only). These results were obtained by empirically setting the weight factor ($w2$) values to $\{0.4, 0.2\}$, for distances $d1$ and $d2$ respectively. However, for distance $d3 = 264$cm, we observed that *M2Gestic* performed best with $w2 = 0$–i.e., when the pointing input was completely ignored, causing the performance to revert to its baseline ($\sim$ 29.26% & 80.84% for ambiguous and unambiguous instructions respectively, in Table 2). Accordingly, similar to humans, the robotic agent should be capable of adjusting its fusion logic automatically, and discard pointing input (due to the likely large noise) if the instructor is too far from the objects. In addition, similar to the observation with human agents, the accuracy of *M2Gestic* is also very low (21.43% at $d1 = 88$ cm and 0% at $d3 = 264$ cm) when *solely* using the pointing input, further corroborating the limitation of pointing-only interactions in cluttered environments.

We also investigated the performance of *M2Gestic* with a weight of 0.5 for the gesture input. In this case we obtained lower accuracy values of 72.20%, 62.43% and 43.57 respectively, for distances $d1$, $d2$ and $d3$ over all the instructions in the dataset. Clearly, giving equal weightage to the pointing input is inadvisable and counterproductive. On further analysis, we found that pointing input helps resolve both perspective and block ambiguities from distance $d1$. At distance=$d2$, the pointing input still helped to solve certain perspective-related ambiguities, but not block-related ones.

6 DISCUSSION & FUTURE WORK

M2Gestic opens up several possible directions for using multi-modal sensing to enhance the AI-based instruction comprehension system.

Automatic selection of weights based on distance: The results from section 5, show that gestures help to improve the accuracy of target-selection, especially when verbal instructions are ambiguous. Human performance is also affected by erroneous gestures made from a distance, but humans implicitly compensate by assigning lesser importance to gestures from longer distances. In future work, the weight assigned to the gesture $(1 - w)$ may be automatically optimized, given the estimate of the instructor's distance (which is relatively easy to obtain via modern vision techniques) and the (mean, variance) of the corresponding pointing error distribution.

Probabilistic inference & interactive comprehension: In the current design, the inference engine performs a tree traversal based on the ranked list obtained at each node. This rank may also be accompanied with a confidence vector to give an idea of amount of ambiguity at each node–e.g., a higher entropy (greater uniformity) of the confidence values implies greater ambiguity. This mechanism can then be used to identify nodes and paths that contribute most to the accuracy of final inference, which in turn can help the AI agent to engage in an iterative 'conversation' to help resolve the most ambiguous parts of the instruction.

Temporal sequence of gestures: *M2Gestic* currently utilizes a single pointing gesture directed towards the target block. However, when verbalising a long instruction, human subjects often use multiple, complex 3-D gestures, such as performing a circular motion around a group of objects or outlining the shape of specific objects. We believe that incorporating such gestures, contextualized to the relevant parts of the verbal instruction, can significantly help improve comprehension accuracy.

7 CONCLUSIONS

We have presented *M2Gestic*, a system for multi-modal comprehension of target-acquisition instructions typically issued by humans to collaborative robots. *M2Gestic* combines neural network-based parsing of visual and verbal inputs (with a weighted knowledge graph traversal mechanism) with potentially-erroneous pointing input, to identify candidate objects even under high visual clutter. By evaluating *M2Gestic* using the benchmark table-top dataset consisting of 28 distinct block arrangements, we show that *M2Gestic* is able to achieve about 61% accuracy in object selection (compared to 73% achieved by humans) in the absence of pointing input. Incorporating gestural input helps to significantly improve (by over 30%) *M2Gestic's* accuracy (especially for ambiguous instructions) when the instructor is close to the table top, with such pointing-driven gains effectively disappearing (due to the noisiness of the pointing input) when the instructor is $\sim 2.6m$ away. More broadly, our work underscores the importance of using AI-driven techniques to incorporate gestural & pointing input as an integral part of real-time, natural interactions between humans and collaborative robots.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative, NRF Investigatorship (NRF-NRFI05-2019-0007) and Agency for Science, Technology and Research(A*STAR) under its AME Programmatic Funding Scheme (Project A18A2b0046). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore or A*STAR.

REFERENCES

[1] Hyemin Ahn, Sungjoon Choi, Nuri Kim, Geonho Cha, and Songhwai Oh. 2018. Interactive Text2Pickup Networks for Natural Language-Based Human–Robot Collaboration. *IEEE Robotics and Automation Letters* 3, 4 (2018), 3308–3315.

[2] Amazon. [n.d.]. Mechanical Turk. www.mturk.com. Accessed: 2019-09-30.

[3] Cynthia Breazeal, Cory D Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin. 2005. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 708–713.

[4] J. M. Foley and Richard Held. 1972. Visually directed pointing as a function of target distance, direction, and available cues. *Perception & Psychophysics* (May 1972), 263–268.

[5] Boris Gromov, Gabriele Abbate, Luca M Gambardella, and Alessandro Giusti. 2019. Proximity human-robot interaction using pointing gestures and a wrist-mounted IMU. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 8084–8091.

[6] Oliver Herbort and Wilfried Kunde. 2016. Spatial (mis-)interpretation of pointing gestures to distal referents. *Journal of experimental psychology. Human perception and performance* 42 (2016), 78–89.

[7] HTC. [n.d.]. VIVE. www.vive.com. Accessed: 2019-09-30.

[8] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Inferring and Executing Programs for Visual Reasoning. In *ICCV*.

[9] Casey Kennington and David Schlangen. 2017. A simple generative model of incremental reference resolution for situated dialogue. *Computer Speech & Language* 41 (2017), 43–67.

[10] Shen Li, Rosario Scalise, Henny Admoni, Stephanie Rosenthal, and Siddhartha S Srinivasa. 2016. Spatial references and perspective in natural language instructions for collaborative manipulation. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 44–51.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[12] Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.

[13] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. 2014. Learning from unscripted deictic gesture and language for human-robot interactions. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[14] Sven Mayer, Valentin Schwind, Robin Schweigert, and Niels Henze. 2018. The Effect of Offset Correction and Cursor on Mid-Air Pointing in Real and Virtual Environments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[15] Sven Mayer, Katrin Wolf, Stefan Schneegass, and Niels Henze. 2015. Modeling distant pointing for compensating systematic displacements. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 4165–4168.

[16] Diederick C Niehorster, Li Li, and Markus Lappe. 2017. The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research. *i-Perception* 8, 3 (2017), 2041669517708205.

[17] Rohan Paul, Jacob Arkin, Derya Aksaray, Nicholas Roy, and Thomas M. Howard. 2018. Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. *International Journal of Robotics Research* 37, 10 (1 9 2018), 1269–1299. https://doi.org/10.1177/0278364918777627

[18] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*. 779–788.

[19] Rosario Scalise, Shen Li, Henny Admoni, Stephanie Rosenthal, and Siddhartha S Srinivasa. 2018. Natural language instructions for human–robot collaborative manipulation. *The International Journal of Robotics Research* 37, 6 (2018), 558–565.

[20] Mohit Shridhar and David Hsu. 2018. Interactive Visual Grounding of Referring Expressions for Human-Robot Interaction. In *Proceedings of Robotics: Science and Systems*.

[21] David Whitney, Miles Eldon, John Oberlin, and Stefanie Tellex. 2016. Interpreting multimodal referring expressions in real time. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3331–3338.

[22] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*. 1031–1042.