Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.079954

ARTICLE

Resolving Ambiguity in Pointing Gestures Using Contextual Reasoning from Large Language Models

Sumin Yeon, Minjae Lee, Jiho Bae and Suwon Lee*

Department of Computer Science and Engineering, Gyeongsang National University, Jinju-Si, Republic of Korea

*Corresponding Author: Suwon Lee. Email: leesuwon@gnu.ac.kr

Received: 31 January 2026; Accepted: 17 March 2026; Published: 27 April 2026

ABSTRACT: In everyday life, people effectively convey their intentions through pointing gestures without explicitly naming objects. In particular, pointing gestures used in conjunction with linguistic expressions such as "this" and "that" play a crucial role in intuitively indicating objects or locations in space. Although research on the recognition of such nonverbal gestures has been actively pursued within the field of human-computer interaction (HCI), accurately interpreting a user's intent remains challenging in situations where the pointing gesture is ambiguous. This paper proposes an integrated system that combines a large language model (LLM), capable of understanding complex human language expressions, with pointing gestures designed to designate targets in space, thereby effectively processing multimodal user commands. The system is designed to accurately recognize user intentions even in complex and uncertain environments (e.g., indoor spaces with multiple objects) by synergistically leveraging spatial information obtained from pointing gestures and contextual reasoning provided by the LLM. To validate the proposed approach, we constructed a dataset comprising complex real-world environments and diverse utterances, and conducted experiments to meticulously analyze the system's performance and limitations. This study demonstrates the potential for natural expansion of language-based spatial understanding within HCI, and suggests avenues for future research in related fields.

KEYWORDS: Multimodal interaction; pointing gesture; large language model; contextual reasoning; object referencing; human-computer interaction

1 Introduction

As computers become ever more deeply integrated into daily life, technologies that emulate or support natural and intuitive human communication methods have drawn increasing attention in the field of human-computer interaction(HCI). Although people already interact with various computing environments, such as smartphones, voice assistants, and chatbots, using natural language, relying solely on voice recognition or text input often falls short of capturing the nuances of human intention, especially within increasingly complex spatial contexts. In particular, with the emergence of social robots in homes and public spaces, these robots must move beyond merely providing information or performing mechanical operations; rather, they need to perceive and interpret linguistic and nonlinguistic information in a manner similar to humans [1]. Meeting this requirement necessitates a broad understanding of the context in which people express their intentions, and makes it essential to consider a variety of cues, including bodily gestures, facial expressions, and gaze.

In everyday life, people frequently use pointing gestures to indicate nearby objects or areas of interest. Even without explicitly describing the exact name or precise location of an object, individuals often combine deictic pronouns—such as "this," "that," or "over there"—with pointing to express their intentions in a simple and natural manner. Because this pointing behavior has a visually clear directive effect, it enables the listener or observer to quickly identify the relevant object or area without requiring a detailed verbal [2]. In other words, pointing at an object with a finger is considered a key nonverbal tool for inducing efficient and intuitive communication among humans, as it allows them to specify their object of interest in a very short period. In the HCI field, numerous studies have been conducted to recognize and apply such pointing gestures. In the early stages, the primary technical challenge involved methods for sensing user pose, accurately tracking that position, and interpreting the resultant data efforts that yielded substantial advancements in improving recognition rates for pointing gestures and minimizing misrecognition.

However, as illustrated in Fig. 1. while using spoken commands in conjunction with pointing gestures promotes more natural comprehension of spatial information and the communicative context from a human perspective, it poses a challenge for computers to process these signals in an integrated manner. Moreover, in real-world settings, the point at which a user is indicating may be ambiguous, and in some cases, it may not be clear which of several similar objects the user intends to reference.

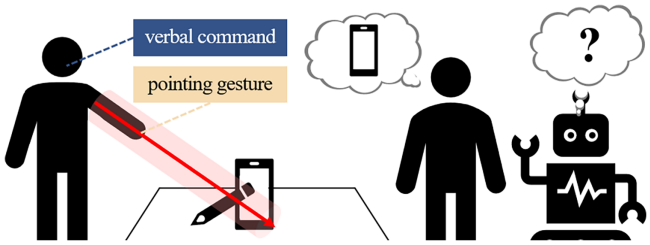

Figure 1: Problem definition. When a voice command and a pointing gesture are presented concurrently, a human observer naturally integrates the two information streams, whereas computational systems are not yet capable of processing these cues in a comprehensive manner.

Recent findings indicate that large language model (LLM) trained on vast amounts of textual data from the internet exhibit excellent performance in tasks involving commonsense reasoning and contextual understanding in a variety of domains [3,4]. This suggests that they are capable of exploiting extensive contextual information to understand and generate natural language. Thus, integrating the linguistic reasoning capabilities of LLMs with the task of interpreting pointing gestures offers a robust mechanism to mitigate the inherent ambiguities of isolated pointing gestures by leveraging linguistic context. Nonetheless, an LLM by itself cannot discern visual information such as the exact location a person is indicating. Therefore, spatial information, such as that provided by pointing gestures, remains indispensable.

While early studies focused on the technical accuracy of gesture recognition, recent advancements have shifted toward 'Embodied Intelligence,' where Large Language Models are used to bridge the gap between human linguistic intent and physical action execution [5]. However, most existing frameworks still struggle with fine-grained spatial ambiguity in cluttered environments, which this paper aims to address. In this paper, we propose an integrated system that combines an LLM capable of comprehending complex human language expressions with pointing gestures that designate targets in physical space, thereby enabling the processing of multimodal user commands.

Our proposed system is designed so that the language processing module and the gesture recognition module fulfill complementary roles, thus facilitating a more precise understanding of user intentions

even in complex and uncertain real-world situations. To validate our system, we collected a dataset in a cluttered domestic environment one in which various objects are scattered throughout and conducted both quantitative and qualitative experiments. The results of these experiments are presented in Section 5.

2 Related Work

2.1 Pointing-Based and Multimodal Interaction

Pointing gestures represent one of the most fundamental and intuitive ways of indicating objects in human communication and have been studied continuously in the HCI field. Methods for obtaining the human skeleton can be broadly categorized into those that estimate 2D coordinates using an RGB camera [6–8], and those that estimate 3D coordinates using stereo cameras or RGB-D sensors [9–12]. Using this skeleton information (e.g., wrist, elbow, shoulder, head), one can calculate the pointing direction. Tolgyessy et al. showed that, when tracking skeleton data at distances of up to 23 meters using a Kinect-based RGB-D sensor, the wrist-elbow vector provided the most accurate pointing direction estimate compared to other joint combinations [13].

When information is provided through a single modality (e.g., gesture only, or voice only), the user's intent may not be expressed with sufficient richness, making misinterpretation more likely in complex scenarios [14–16]. In particular, if the environment is cluttered or if there are multiple objects, gestures alone may be insufficient for precise referencing, whereas voice alone may lack adequate spatial or locational information.

To overcome these limitations, researchers have designed multimodal interfaces that combine a variety of input channels [17,18]. Fang et al., for instance, employed virtual reality (VR) technologies to integrate dynamic gesture recognition and voice recognition, thereby realizing intuitive interaction. Their experimental results indicated that this system significantly improved task accuracy, user satisfaction, and training efficiency compared to previous approaches [19]. Similarly, Salinas-Martínez et al. demonstrated in an industrial HRI scenario that controlling a collaborative robot via the integration of voice commands and freehand gestures is indeed feasible in factory settings, illustrating the practical applicability of multimodal interaction [20].

Research that combines pointing gestures and language has also been active. Paul et al. proposed a framework that integrates pointing-gesture information and object data from RGB images to extract task actions, objects of interest and their attributes, and location information from linguistic commands [21]. Lin et al. showed, through a framework combining an LLM and a MediaPipe-based gesture recognition module, that it is possible to successfully generate context-appropriate robot action plans, even in complex situations [22].

However, rule-based approaches that merely link pointing to existing multimodal systems face well-known constraints in handling diverse ambiguities that arise in open-ended scenarios (e.g., multiple overlapping objects, complex linguistic expressions) [23]. The system proposed in this paper aims to move beyond simple referencing toward higher-order intent inference by harnessing deeper linguistic context via a large language model and merging it with spatial information obtained from pointing.

2.2 Large Language Models and Interaction

Large language models (LLMs) have recently demonstrated remarkable performance in complex contextual reasoning and natural language processing tasks, leading to their widespread adoption in the HCI domain [4,24,25]. Zhang et al. compiled various HRI use cases that leverage LLMs for tasks such as executing

robotic commands, generating explanations, and planning interactions, concluding that large-scale language models render communication with robots more natural [3].

Nonetheless, several limitations remain when directly applying LLMs to interactive systems. Most LLMs fundamentally assume text-only input, so they lack perception or comprehension of the actual physical environment. Furthermore, due to the issue of "hallucination," there is a risk that an LLM might produce utterly baseless instructions or incorrect information in a plausible manner. Lai et al. have noted that, without strict constraints on the output when applying LLMs in HRI for control purposes, safety issues may arise [26]. To address this, alignment strategies and the provision of domain knowledge or contextual information are needed to reduce the potential for hallucination. For example, Gramopadhye and Szafir showed that supplying additional information on the robot's environment (e.g., the location and attributes of objects) led to the generation of more feasible commands [27].

In the present research, we supplement the LLM's lack of spatial and environmental awareness through pointing gestures, and we constrain the LLM's responses using a JSON schema to prevent control commands from becoming unrealistic.

Successful integration of physical environments and language-based AI requires a robust reference resolution process, which associates expressions such as "this" or "the red button" with actual objects. Traditional multimodal systems primarily utilized manually crafted fusion rules, but recent trends point toward data-driven methods that jointly learn from visual, linguistic, and gestural information. Zhang et al. highlighted the need for robots to meticulously interpret the real variations in human pointing strategies observed in open-ended task scenarios [3]. Moreover, studies such as Mon-Williams et al.'s ELLMER framework have begun merging GPT-4 with visual/tactile sensor data and retrieval-based memory, enabling systems to adapt to environmental changes during long-term task execution [5].

At present, most LLM-based interactions make only limited use of visual or spatial information, whereas gesture-based systems often fall short in advanced linguistic processing. In response, this paper proposes a multimodal system that integrates the language comprehension capabilities of an LLM with the intuitive spatial referencing afforded by pointing gestures. Through this approach, we aim to interpret user intent more accurately and comprehensively, while ensuring a more stable execution of instructions and commands in real-world environments.

Furthermore, recent work by Lai et al. [26] proposed a multimodal fusion system that integrates voice commands and deictic postures with Large Language Models (LLMs) to enable natural human-robot interaction. Their study is significant in that it enhances interaction flexibility by combining the robust contextual reasoning capabilities of LLMs with multimodal data. However, our work distinguishes itself technically by addressing subtle spatial ambiguities that arise in real-world environments—particularly where multiple similar objects are densely clustered or overlapping. We achieve this through cylindrical pointing ray modeling and a sophisticated weight-based fusion strategy designed to resolve these specific challenges.

3 Methodology

The primary objective of the proposed system is to accurately identify objects and respond to corresponding queries within a crowded indoor environment by integrating the user's skeleton information with linguistic commands. Rather than relying on a single sensor or a single modality, this approach combines pointing gestures with a large language model (LLM) to build a more sophisticated and reliable multimodal interface.

A system diagram is summarized in Fig. 2. The user's command is given via voice input, while the user simultaneously points to a particular object in the physical environment. This pointing information is linked

to a three-dimensional point cloud map, providing the fundamental spatial information needed to determine which object is being indicated. Next, the LLM interprets the linguistic context to respond to the user's query. The system is designed such that combining these responses with the spatial data obtained from the pointing gesture enables the correct identification of the target object—even when the pointing is slightly imprecise or the language use is ambiguous—ultimately generating an appropriate final answer.

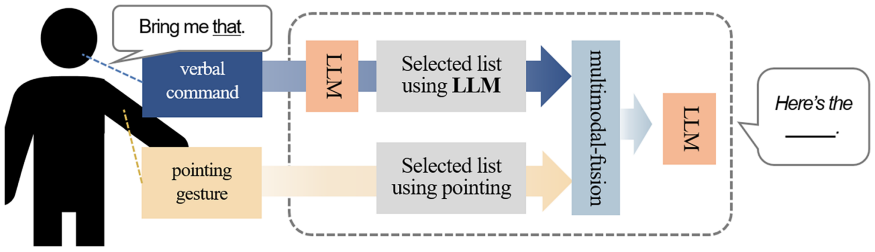

Figure 2: System overview. The LLM interprets the user's voice command (e.g., "Bring that over") and pointing gesture to determine the correct selection and generate a response.

3.1 Pointing Gesture and Object Collision

A pointing gesture is an action in which a user extends the arm and finger in a specific direction to indicate an object or location. In this study, pointing gestures are recognized by performing body tracking with an RGB-D camera to obtain the user's three-dimensional skeleton data. Because an RGB-D camera captures both color and depth information in real time, it can relatively accurately detect the positions of the user's elbow and wrist. According to previous research, defining the segment between the wrist and elbow as the pointing ray is especially effective for extracting pointing directions with high accuracy [13]. Hence, we adopt this method in the present study as well.

However, in a real environment, sensor noise, camera position, and user posture may cause the pointing ray to fail to precisely indicate a single point. To address this issue, our system allows for the possibility of multiple candidate objects rather than restricting the user to pointing at only a single object. Specifically, the pointing ray is extended into the shape of a cylinder so that any object within a certain range is registered as a collision candidate, even if the ray does not pass exactly through its center. The radius R of this cylinder is determined by an optimization process described in Section 'Pointing Gesture and Object Collision Evaluation'.

To recognize objects in three-dimensional space, we employ a point cloud map, which is composed of segmented objects (i.e., instances) derived from previously collected data, where each object is represented by a set of three-dimensional points. The user's body-tracking coordinates and the pointing ray are transformed from the camera coordinate system into the world coordinate system via a vision-based mapping algorithm, after which they are compared against the point cloud map.

Collision detection between each object and the cylindrical pointing ray uses an axis-aligned bounding box (AABB) technique. Prior to this step, each object instance undergoes preprocessing to remove outlier points. Specifically, the 25th and 75th percentile values of each point set are computed, and any points that lie beyond the interquartile range (IQR) are eliminated, thus minimizing outliers that might be introduced by noise. From the filtered points, the minimum and maximum are used to generate the final AABB for each object.

To reduce computational complexity in checking collision between the cylinder and the AABB, we adopt a slab-based method in which the cylinder's central axis is considered a line segment that is tested for intersection with the box expanded by the cylinder radius. Precise collision detection involving the

entire cylinder is very costly; hence, we approximate it by determining whether the line segment of the cylinder's central axis intersects the AABB once the box is expanded by the cylinder's radius. By avoiding complex intersection calculations for the entire cylindrical volume, this method ensures high computational efficiency, allowing for real-time interaction even in densely populated environments.

Objects that collide with the pointing ray (i.e., the extended cylinder) are then prioritized according to their distance from the ray. To achieve this, we construct five candidate cylinders with radius ranging from 0.1 up to a maximum value, incremented in quartile intervals. An object colliding with the cylinder of the smallest radius is considered the highest-priority candidate. If no object collides at the smallest radius, the system proceeds to check collisions with cylinders of gradually increasing radius. Fig. 3 provides a visual representation of this process, in which multiple radius enable correct object selection despite minor inaccuracies in the user's pointing direction.

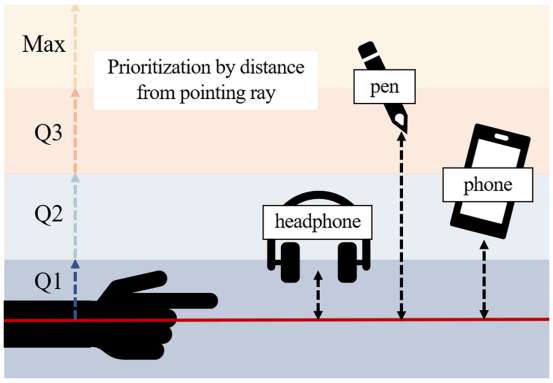

Figure 3: An example of object prioritization via a pointing gesture. Objects are prioritized based on their proximity to the pointing ray: headphone, pen, and phone.

Additionally, an algorithm is applied to detect the onset and termination of pointing gestures in the video stream as the user moves. If the same object is nominated as a collision candidate in k consecutive frames (with $k = 3$ in this study), the system interprets that moment as the start of a pointing gesture. Once the user ceases pointing or exhibits significantly different movements in subsequent frames, the system considers the pointing gesture to have ended.

3.2 Contextual Understanding via LLM

User voice commands typically take the form of sentences containing deictic pronouns such as "this" or "that," which are anchored to the object indicated by the pointing gesture. Examples include "Could you hand this over?" or "Can you give me that so I can listen to the song?" Such commands may pertain to a single object, multiple objects, or, in extreme cases, an open-ended query that applies to every object in the environment.

A LLM can analyze contextual and semantic information to infer which real-world target is referenced by a deictic expression or to narrow the scope of the user's intended question. For instance, in the request "Could you use this to turn off the lights?", it may be unclear whether "this" indicates a light switch or a smart speaker, but the LLM can leverage general knowledge—recognizing that lights are typically turned off via a switch or voice control through a speaker to assign priority to each candidate object.

In this study, we capture the user's voice commands with a microphone integrated into the RGB-D camera (e.g., Kinect). We then convert the audio to text using a speech-to-text API, and send the resulting

text as a query to the LLM. For prompt engineering, we provide the LLM with a guide text beforehand (see Fig. 4) to facilitate context analysis related to deictic terms (e.g., “this” or “that”). The prompt is designed so that, given a predefined list of objects, the system identifies those most relevant to the user command and calculates a normalized weight for each candidate. These weights are later utilized in the multimodal fusion process (see Section 3.3). If the command is ambiguous, multiple objects may receive plausible weights. If one particular object is very likely the intended reference, it is assigned a high weight, while all others are assigned lower values.

System: I will provide a command containing “this” or “that.” You need to infer what object “this” or “that” refers to.
The possible objects are as follows:
Objects = {‘mouse2’, ‘glasses’, ‘laptop1’, ⋯ ,‘headphones1’, ’tumbler2’, ‘pen’}
Respond with consideration of the priority order of the selected objects.
If the context of the sentence is unclear, return all possible matching objects instead of an answer like “unknown”.
Provide a weight for each object in your response. The sum of all weights in the list should equal 1.
The thought process should be as follows:
Command = “How much battery does this have left?”
1. This object must be an electronic device that uses a battery.
2. Among the items, the ones likely to contain a battery are laptop1, laptop2, mouse1, mouse2, phone1, phone2, headphones1, headphones2.
3. Let’s set a priority order, based on how often people tend to check the battery. And based on the priority, let’s assign a weight to each object.
4. Answer = [phone1, phone2, laptop1, laptop2, headphones1, headphones2, mouse1, mouse2], [0.2, 0.2, 0.15, 0.15, 0.1, 0.1, 0.05, 0.05]

Figure 4: An example of the task prompt defined for contextual reasoning. The system utilizes predefined objects, and the LLM learns the provided reasoning process to generate its response.

To mitigate the risk of linguistic hallucination, where the LLM might suggest non-existent objects, we employed a structural grounding strategy using a strict JSON schema (see Fig. 5). By constraining the model’s output space to a predefined list of physical object instances, the system ensures that the inferred intent is always anchored to the actual 3D environment.

```
json_schema = {
"title": "Response Schema",
"description": "This schema returns, for each input sentence, a list of object names (object_names) and a corresponding list of weights (weights).
Each answer object includes the fields: sentence, object_names, and weights.
The sum of all weights must equal 1, and the lengths of the object_names and weights arrays must be the same.",
"type": "object",
"properties": {
                "sentence": {  "type": "string",
                                "description": "The original sentence text."},
                "object_names": {   "type": "array",
                                "items": { "type": "string"},
                                "minItems": 1,
                                "description":"A list of inferred object names."},
                "weights": {"type": "array",
                                "items": { "type": "number"},
                                "minItems": 1,
                                "description": "A list of weights corresponding to each object (the sum of weights must equal 1)."}},
  "required": ["sentence", "object_names", "weights"],
}
```

Figure 5: The JSON schema format used to explicitly constrain the return values.

In this way, the LLM returns both a list of candidate objects and a corresponding set of weights for each object in the command. Responses such as “Unknown” are avoided so as to include all feasible candidates, enabling the subsequent multimodal fusion process to combine pointing data to arrive at a final object selection.

3.3 Multimodal Fusion

To establish the user's intent with high certainty, the proposed system integrates the candidate object list from the pointing gesture with the list inferred by the LLM. Both lists carry inherent priority information (weights), and the fusion of these priorities is a key factor in achieving accurate intent recognition.

The candidate objects identified by the pointing gesture are ranked based on the cylindrical radius and a sequential expansion technique. As illustrated in Fig. 6, objects are prioritized by their proximity to the pointing ray across multiple radius. An object colliding with a smaller radius cylinder is considered a higher-priority candidate.

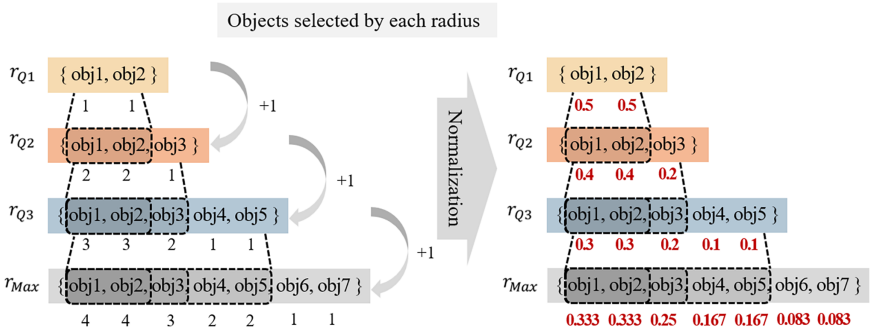

Figure 6: An example of calculating weights for a list selected by a pointing gesture. Scores are assigned based on radius ranges and then normalized to derive the final weights.

We formalize this process by defining a pointing score $S_P(o)$ for an object o. Let $R = \{r_{Q1}, r_{Q2}, r_{Q3}, r_{Max}\}$ be the set of radius corresponding to the quartiles of the radius list. The score $S_P(o)$ is calculated as the sum of occurrences of object o across the candidate sets C_r for each radius $r \in R$:

$$S_P(o) = \sum_{r \in R} \mathbb{1}(o \in C_r)$$

where $\mathbb{1}(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise. The final normalized pointing weight $W_P(o)$ is then derived as follows:

$$W_P(o) = \frac{S_P(o)}{\sum_{j \in P} S_P(j)}$$

where P is the set of all candidate objects detected across all radius.

Simultaneously, the LLM provides a list L of candidate objects derived from the linguistic context of the user's command. Each object $o \in L$ is assigned a weight $W_L(o)$, which is normalized based on its contextual importance to ensure the sum of weights in the list equals 1.

The final multimodal fusion identifies the target object O^* by taking the intersection of the two candidate lists ($L \cap P$). If the intersection is not empty, the system calculates the combined weight for each object; otherwise, it defaults to the LLM's highest-weighted candidate. The selection logic is defined as:

$$O^* = \begin{cases} \arg\max_{o \in (L \cap P)} (W_L(o) + W_P(o)) & \text{if } L \cap P \neq \varnothing \\ \arg\max_{o \in L} W_L(o) & \text{if } L \cap P = \varnothing \end{cases}$$

This dual-pathway approach allows the system to remain robust even if the pointing gesture is imprecise or the verbal command is ambiguous. For instance, if a user points vaguely toward a group of objects while

asking "What is this?", the spatial cues from the pointing gesture (via W_P) and the linguistic reasoning from the LLM (via W_L) complement each other to pinpoint the correct target.

4 Implementaion

To acquire user data, we employed an Azure Kinect RGB-D sensor and utilized Microsoft's Azure Kinect Body Tracking SDK for body tracking. For the coordinate transformation into the world coordinate system, we used ArUco markers [28] and applied the RANSAC algorithm [29] to multiple marker sets, thereby estimating a robust camera-world coordinate transformation matrix. An external API was used to call the LLM, which inferred contextual information related to the user's voice commands. For speech recognition of user commands, we adopted the Google Cloud Speech API.

5 Evaluation

In this section, we first conduct a systematic evaluation of the accuracy of pointing gestures, deriving an appropriate cylinder radius for the proposed system. Next, we assess the performance of contextual understanding via an LLM, ultimately selecting the model to be adopted in this study. Finally, we analyze the overall accuracy of the multimodal fusion approach, which integrates the outputs of pointing gestures and the LLM.

A total of 10 participants (3 women and 7 men, ages 23 to 26) took part in the experiment. Although none were native English speakers, all were sufficiently fluent for everyday conversation. The experiment was conducted in a laboratory space of approximately 26 m^3. A Kinect was placed near the center of the lab at a distance of about 1.5 m from the participant, such that the participant's entire body remained within the camera's field of view.

Fig. 7 shows the laboratory layout. 19 objects were selected as pointing targets. Among these, there were duplicates of several items (i.e., laptops, mouse, phones, headphones, tumblers, and diaries), belonging to the same class but differing slightly in shape. This setup models a realistic scenario in which multiple people might use similar items or several products of the same type might be placed together. Some objects were also stacked to reflect everyday clutter.

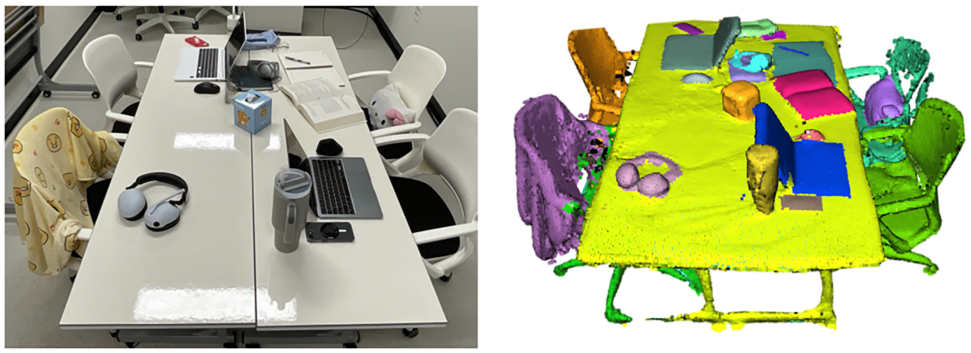

Figure 7: The laboratory setup and its corresponding 3D point cloud map. A total of 19 objects are positioned on the desk and chair. Objects are as follows: laptops, mouses, headphones, phones, tumblers, tissue, wet wipes, diaries, pen, cushion, blanket, book, glasses.

The utterance commands used in the experiment were designed to match each of the 19 objects, resulting in a total set of 350 utterances. Each object was associated with 17–20 commands. Identical commands were shared among objects of the same class (e.g., the two laptops). Additionally, certain statements consisted

solely of simple directives, such as "Bring that over," and were assigned to all objects; this allowed us to include situations where, in the absence of a pointing gesture, the intended reference would be entirely unclear. Excluding these duplicate statements, there were 80 unique queries in total.

The experiment proceeded as follows:

1. The participant stood approximately 1.5 m from the Kinect sensor and 1.5 m from a desk.
2. Following a predetermined sequence, the participant pointed at each of the 19 objects in turn. When the system detected the initiation of a pointing gesture, the participant uttered the corresponding spoken command.
3. The same process was repeated for all 19 objects, resulting in 350 spoken commands per participant.
4. Across the 10 participants, a total of 3500 data pairs were collected.

5.1 Pointing Gesture and Object Collision Evaluation

We first evaluated the accuracy of recognizing pointing gestures, as described in Section 'Pointing Gesture and Object Collision', focusing on how reliably the system generates candidate object lists. In particular, we compared five radius options to determine the ideal cylinder radius range for practical use.

The options were defined as follows:

- R1: [0.01, 5, 10, 15, 20]
- R2: [0.01, 12.5, 25, 37.5, 50]
- R3: [0.01, 25, 50, 75, 100]
- R4: [0.01, 50, 100, 150, 200]
- R5: [0.01, 125, 250, 375, 500]

The returned object list contains prioritized items. We therefore adopted a top-k evaluation method: we took the first k entries from each list and checked whether the user's actual target object (i.e., the "ground truth") was among them. We used $k = 1, 2, 3$ and 5.

As indicated in Table 1, R4 achieved the highest accuracy for all $k = 1, 2, 3$ and 5. By contrast, smaller or intermediate radius settings such as $R1$, $R2$, and $R3$ showed progressively improving accuracy as k increased, suggesting that if the pointing direction is somewhat imprecise, a modestly larger radius can still include the correct candidate object. On the other hand, $R5$'s upper radius limit was so large that many unintended objects were frequently included, making it difficult to establish precise priorities and ultimately resulting in lower accuracy.

Table 1: Accuracy (%) of the pointing gesture for five radius options (R1-R5) using various top-k thresholds. R4 achieved the highest accuracy in all cases.

Radius	R1	R2	R3	R4	R5
k = 1	9.77	11.31	11.97	12.94	11.30
k = 2	18.31	22.03	25.94	26.40	23.54
k = 3	19.94	25.17	33.34	34.51	30.26
k = 5	22.14	29.94	44.91	52.69	49.38

In summary, the intermediate radius range used by $R4$ most effectively compensated for pointing errors while avoiding an excessive number of spurious objects. Accordingly, we adopted $R4$ as the default setting for subsequent experiments (see Section 'Multimodal Fusion Evaluation').

5.2 Contextual Understanding via LLM Evaluation

Next, we analyzed the accuracy of LLM-based contextual inference introduced in Section 5.2. We provided the same set of queries (transcribed from spoken commands) to various LLM models—Gemini 1.5 flash, Gemini 2.0 flash, GPT 4o, and GPT 4o mini—and verified whether each model's top-k candidate objects included the correct target. We used $k = 1, 2, 3$ and 5, matching the pointing evaluation.

As shown in Table 2, Gemini 2.0 flash consistently achieved the highest accuracy across all values of k. For instance, at $k = 1$, Gemini 2.0 flash outperformed the other models by approximately 5–10 percentage points; at $k = 5$, it maintained its advantage. Although GPT 4o and GPT 4o mini initially exhibited lower accuracy, their performance improved as k increased. Gemini 1.5 flash also achieved moderate success, but the 2.0 version was measurably superior overall.

Table 2: Accuracy (%) of different language models under various top-k thresholds. Gemini 2.0 flash achieved the best performance in all cases.

Model	Gemini 1.5 Flash	Gemini 2.0 Flash	GPT 4o	GPT 4o Mini
k = 1	17.71	20.57	14.57	17.14
k = 2	30.00	36.00	32.57	31.43
k = 3	36.00	42.57	39.71	39.71
k = 5	44.00	52.00	48.00	47.71

Consequently, Gemini 2.0 flash was chosen as the final model in the question-context processing stage, given its strong capabilities in understanding the general conversational context and performing basic commonsense reasoning needed for this research.

5.3 Multimodal Fusion Evaluation

Finally, we assessed the overall accuracy of the multimodal fusion approach proposed in Section 5.3, which combines pointing-gesture data and LLM-based contextual inference. We configured the system using the optimal settings determined above: R4 for pointing and Gemini 2.0 flash for the LLM.

As shown in Table 3, using only pointing gestures yielded an accuracy of approximately 12.94%, and using only the LLM yielded an accuracy of approximately 20.57%. However, fusion of the two information sources raised accuracy to about 32.83%, yielding an improvement of at least 12%p over the single-modality cases.

Table 3: Comparison of top-1 accuracy (%) for different methods. Our multimodal fusion approach significantly outperforms both unimodal baselines and random chance (5.3%, i.e., 1 out of 19).

Pointing Gesture	Contextual Understanding	Multimodal Fusion
12.94	20.57	32.83

These results indicate that spatial cues (pointing gesture) and contextual reasoning (utterance commands) complement each other effectively, offering significant benefits for understanding user intent in complex real-world settings.

Additionally, analyses of individual participant logs revealed that in cases where pointing gestures were highly inaccurate or the LLM struggled to interpret the intended target, the remaining modality often

provided partial compensation. This finding suggests that the proposed system operates with relative stability in the presence of realistic noise or ambiguity.

5.4 System Latency Analysis

To evaluate the system's practical usability, we analyzed the average processing latency. The total time required for a single interaction cycle—starting from the initiation of a voice command to the final object identification—is approximately 2.0 s on average. Specifically, the audio transcription via Google Cloud Speech API takes about 1.0 s, and the contextual reasoning by Gemini 2.0 Flash takes approximately 0.7 s. The spatial collision detection and coordinate transformation processes are highly efficient, consuming less than 0.1 s. While these latency figures are sufficient for non-urgent assistive interactions, future work will aim to further reduce processing time to support more dynamic and real-time interaction scenarios.

6 Discussion

The primary objective of this study was to evaluate the technical feasibility of integrating Large Language Models (LLMs) with pointing gestures to resolve spatial ambiguity in densely populated indoor environments. Our findings indicate that the proposed multimodal fusion framework achieved a Top-1 accuracy of 32.83%, representing a significant improvement of over 12%p compared to unimodal baselines. While the absolute accuracy may appear modest, it reflects the intentional experimental complexity of our setup, which included identical object replicas and overlapping items. This configuration was specifically designed to simulate real-world challenges where single-modality sensors often fail to provide sufficient clarity for disambiguation.

The current additive weighting scheme was selected for its interpretability, as it explicitly demonstrates how spatial cues and linguistic context complement each other. However, we recognize that more adaptive or learning-based fusion strategies could further enhance the system's robustness against environmental noise and sensor uncertainties. Regarding generalizability, although the evaluation was limited to ten participants in a single laboratory setting, the collection of 3500 data pairs provided a robust statistical foundation for our analysis. Future research will encompass a broader range of spatial layouts and diverse user demographics to further validate the framework's applicability.

Furthermore, the reliance on strict JSON schemas and predefined object lists was a deliberate strategy to ensure structural integrity and mitigate the risk of LLM hallucinations. To enhance real-world scalability, transitioning toward open-vocabulary detection and dynamic mapping represents an essential next step. Finally, from a practical usability perspective, the system demonstrated an average processing latency of approximately 2.0 s. While this is acceptable for the auxiliary tasks examined in this study, incorporating qualitative metrics, such as user satisfaction and interaction fluidity, will be crucial for future system optimization.

7 Conclusion

We presented a multimodal approach that combines pointing gestures with large language models (LLMs) for robust object referencing in complex indoor environments. By integrating spatial cues derived from pointing gestures with contextual inferences from LLMs, the proposed system overcomes the limitations of relying on either modality in isolation.

Comparative experiments identified an intermediate cylindrical radius setting (R4) and the Gemini 2.0 flash model as the optimal configuration. The multimodal fusion of these components yielded an accuracy increase of more than 12 percentage points over single-modality baselines, underscoring the

complementary nature of spatial and linguistic information. Even if a pointing gesture is slightly misaligned, linguistic reasoning can narrow down possible object candidates, while spatial cues compensate for vague or ambiguous commands by pinpointing the user's intended target with greater precision. Our experiments confirmed that the system performs effectively in real-world scenarios involving cluttered environments or incomplete verbal descriptions.

As discussed, several promising directions remain for future investigation, including expanding to broader spatial layouts and user demographics, transitioning toward open-vocabulary detection and dynamic object mapping, and incorporating additional interaction modalities such as gaze tracking. Overall, this study demonstrates that combining computer vision with language models can substantially enhance more natural and intuitive human–computer interactions.

Acknowledgement: None.

Funding Statement: This work was supported by the Learning & Academic Research Institution for Master's. Ph.D. Students, and Postdocs (LAMP) Program of the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education (No. RS-2023-00301974).

Author Contributions: Study conception and design: Sumin Yeon, Suwon Lee; data collection: Minjae Lee, Jiho Bae; analysis and interpretation of results: Sumin Yeon, Suwon Lee; draft manuscript preparation: Sumin Yeon, Minjae Lee, Jiho Bae; revision of the manuscript: Sumin Yeon, Suwon Lee. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data and materials used in this study are currently part of an ongoing project. While the full dataset cannot be publicly released at this time due to project constraints and privacy policies, the authors are preparing a subset of the data and a benchmark version for public release upon the official conclusion of the project. Until then, access to the data for academic purposes may be considered upon reasonable request to the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hildt E. What sort of robots do we want to interact with reflecting on the human side of human-artificial intelligence interaction. Front Comput Sci. 2021;3:671012. doi:10.3389/fcomp.2021.671012.
2. Cooney S, Brady N, McKinney A. Pointing perception is precise. Cognition. 2018;177:226–33. doi:10.1016/j.cognition.2018.04.021.
3. Zhang C, Chen J, Li J, Peng Y, Mao Z. Large language models for human-robot interaction: a review. Biomim Intell Robot. 2023;3(4):100131. doi:10.1016/j.birob.2023.100131.
4. Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, et al. Chain-of-thought prompting elicits reasoning in large language models. Adv Neural Inf Process Syst. 2022;35:24824–37. doi:10.52202/068431-1800.
5. Mon-Williams R, Li G, Long R, Du W, Lucas CG. Embodied large language models enable robots to complete complex tasks in unpredictable environments. Nat Mach Intell. 2025;7:592–601. doi:10.1038/s42256-025-01005-x.
6. Constantin S, Eyiokur FI, Yaman D, Bärmann L, Waibel A. Multimodal error correction with natural language and pointing gestures. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023 Oct 2–6; Paris, France. p. 1976–86.
7. Pateraki M, Baltzakis H, Trahanias P. Visual estimation of pointed targets for robot guidance via fusion of face pose and hand orientation. Comput Vis Image Underst. 2014;120:1–13. doi:10.1016/j.cviu.2013.12.006.

8. Schauerte B, Fink GA. Focusing computational visual attention in multi-modal human-robot interaction. In: Proceedings of the International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction; 2010 Oct 9–13; Paris, France. p. 1–8.
9. Nickel K, Scemann E, Stiefelhagen R. 3D-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario. In: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition; 2004 May 17–19; Seoul, Republic of Korea. p. 565–70.
10. Nickel K, Stiefelhagen R. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In: Proceedings of the 5th International Conference on Multimodal Interfaces; 2003 Nov 5–7; Vancouver, BC, Canada. p. 140–6.
11. Jojic N, Brumitt B, Meyers B, Harris S, Huang T. Detection and estimation of pointing gestures in dense disparity maps. In: Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580); 2000 Mar 28–30; Grenoble, France. p. 468–75. doi:10.1109/afgr.2000.840676.
12. Lee M, Bae J, Choi SM, Lee S. Finger-pointing interface for human gesture recognition based on real-time geometric comprehension. In: SIGGRAPH Asia 2024 Posters; 2024 Dec 3–6; Tokyo, Japan. p. 1–2. doi:10.1145/3681756.3697892.
13. Tolgyessy M, Dekan M, Duchon F, Rodina J, Hubinsky P, Chovanec L. Foundations of visual linear human-robot interaction via pointing gesture navigation. Int J Soc Robot. 2017;9:509–23. doi:10.1007/s12369-017-0408-9.
14. Clough S, Duff MC. The role of gesture in communication and cognition: implications for understanding and treating neurogenic communication disorders. Front Hum Neurosci. 2020;14:323. doi:10.3389/fnhum.2020.00323.
15. Driess D, Xia F, Sajjadi MS, Lynch C, Chowdhery A, Ichter B, et al. Palm-e: an embodied multimodal language model. arXiv:2303.03378. 2023.
16. Dritsas E, Trigka M, Troussas C, Mylonas P. Multimodal interaction, interfaces, and communication: a survey. Multimodal Technol Interact. 2025;9(1):6. doi:10.3390/mti9010006.
17. De Angeli A, Gerbino W, Cassano G, Petrelli D. Visual display, pointing, and natural language: the power of multimodal interaction. In: Proceedings of the Working Conference on Advanced Visual Interfaces; 1998 May 24–27; L'Aquila, Italy. p. 164–73.
18. Alalyani N, Krishnaswamy N. Multimodal referring expression generation for human-computer interaction. In: International Conference on Human-Computer Interaction. Berlin/Heidelberg, Germany: Springer; 2024. p. 3–22.
19. Fang D, Chen J, Jiang Y, Zhang G. A multimodal virtual reality system for switchgear operation training: integration of dynamic gesture and speech recognition. In: Proceedings of the 2024 4th International Conference on Artificial Intelligence, Virtual Reality and Visualization; 2024 Nov 1–3; Nanjing, China. p. 125–33.
20. Salinas-Martínez ÁG, Cunillé-Rodríguez J, Aquino-López E, García-Moreno AI. Multimodal human-robot interaction using gestures and speech: a case study for printed circuit board manufacturing. J Manuf Mater Process. 2024;8(6):274.
21. Paul SK, Hoseini P, Gopinath AV, Nicolescu M, Nicolescu M. Simultaneous integration of multimodal interfaces for generating structured and reliable robotic task configurations. In: Proceedings of the 2022 5th International Conference on Machine Vision and Applications; 2022 Feb 18–20; Singapore. p. 61–6.
22. Lin LH, Cui Y, Hao Y, Xia F, Sadigh D. Gesture-informed robot assistance via foundation models. In: Proceedings of the 7th Annual Conference on Robot Learning; 2023 Nov 6–9; Atlanta, GA, USA.
23. Liang PP, Zadeh A, Morency LP. Foundations & trends in multimodal machine learning: principles, challenges, and open questions. ACM Comput Surv. 2024;56(10):1–42. doi:10.1145/3656580.
24. Schramowski P, Turan C, Andersen N, Rothkopf CA, Kersting K. Large pre-trained language models contain human-like biases of what is right and wrong to do. Nat Mach Intell. 2022;4(3):258–68. doi:10.1038/s42256-022-00458-8.
25. Kwon M, Xie SM, Bullard K, Sadigh D. Reward design with language models. arXiv:2303.00001. 2023.
26. Lai Y, Yuan S, Nassar Y, Fan M, Gopal A, Yorita A, et al. Natural multimodal fusion-based human–robot interaction: application with voice and deictic posture via large language model. arXiv:2501.00785. 2025.

27. Gramopadhye M, Szafir D. Generating executable action plans with environmentally-aware language models. In: Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2023 Oct 1–3; Detroit, MI, USA. p. 3568–75.
28. Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas FJ, Marín-Jiménez MJ. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognit. 2014;47(6):2280–92. doi:10.1016/j.patcog.2014.01.005.
29. Fischler MA. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM. 1981;24(6):381–95. doi:10.1145/358669.358692.