Point Target Detection for Multimodal Communication

Hannah VanderHoeven[0000−0003−3234−6797], Nathaniel Blanchard[0000−0002−2653−0873], and Nikhil Krishnaswamy[0000−0001−7878−7227]

Colorado State University, Fort Collins CO, 80523, USA
{hannah.vanderhoeven,nathaniel.blanchard,nikhil.krishnaswamy}@colostate.edu

Abstract. The future of multimodal communication between humans and AIs will rest on AI's ability to recognize and interpret non-linguistic cues, such as gestures. In the context of shared collaborative tasks, a central gesture is deixis, or pointing, used to indicate objects and referents in context. In this paper, we extend our previously-developed methods for gesture recognition and apply them to a collaborative task dataset where objects are frequently indicated using deixis. We apply gesture detection to deictic gestures in the task context and use a "pointing frustum" to retrieve objects that are the likely targets of deixis. We perform a series of experiments to assess both the quality of gesture detection and optimal values for the radii of the conical frustum, and discuss the application of target detection using pointing to multimodal collaborative tasks between humans and computers.

Keywords: Deictic gesture · Gesture semantics · Multimodal dialogue.

1 Introduction

As artificial intelligence becomes more ubiquitous and sophisticated, users will increasingly expect computers to behave more like humans. This includes the capacity to understand not only common input modalities like language, but also non-linguistic modalities such as gestures. A critical component of multimodal human-human interaction is deictic gesture (pointing), and therefore accurate identification of pointing targets in real time is an important feature for multimodal language understanding and human-computer interaction. By using pointing vectors to aid in identifying targets in three-dimensional space, the semantic denotata intended by a user can be extracted from a video stream. In addition, when combined with other features, such as speech, the data extracted can further aid the overall understanding of how humans are communicating with one another or with an intelligent system. For the most accurate analysis and seamless use, correctly and consistently identifying people, gestures, and their intended semantic targets in real time is vital. We previously developed a pipeline to automatically detect preparatory, "stroke," and recovery phases of gestures [26], based on the gesture semantics previously developed by the

community [1, 9, 18]. Our method showed promising results in automatic identification of complex multi-frame gestures in real time, with lower computational overhead than competing approaches.

In this paper, we incorporate this model into a pipeline that detects the semantic target of pointing gestures, specifically. This serves as a direct operationalization of the pointing cone semantics of Kranstedt et al. [13], among others. We demonstrate this capability in the context of a small group task where people communicate with each other using both gesture and language [10]. To effectively extract instances of pointing and the associated targets from a small group scenario, a few things must be considered. First, accurate gesture detection on a per-participant basis is necessary to consistently match the pointing vector with who is communicating. Second, precision errors can occur when a single vector is used to select objects, since it is unlikely that the objects and vector line up perfectly. To account for this, a "pointing frustum" is formed around the pointing vector to create a "detection" region in three-dimensional space. Objects that intersect with this region, based on the center of the object are selected as targets on interest [12]. As pointing specificity degrades with distance from the pointer to the target but is still interpretable by other humans at a distance [25], selecting the most fitting near and far base radii for the pointing frustum is important to correctly identify intended targets in a small space, without selecting unintended targets. We compare an automatic pointing detection method with a human-annotated ground truth, and frustum radii to determine the feasibility of point and target detection of small objects. We establish a novel baseline for object selection in a joint situated task using deictic gesture only, and in the process expose how challenging automatic inference of indicated objects in a collaborative setting can be, due to variation across individuals and groups in communication and deictic strategies. We discuss how target objects detected through pointing can then provide important context to the automated understanding and interpretation of interactions in a small group task, and how additional features might help add more context to overcome inaccuracies.

2 Related Work

In various human computer interaction studies, pointing is a common gesture used to indicate the intended target of a user or study participant. Use of pointing for deixis spans many different languages and cultures [11], making it an ideal gesture to be integrated with HCI systems. Pointing may be used to execute hardware commands or interact with a user interface [7]. Pointing is also an important feature of small group communication especially when combined with speech, as it allows individuals to ground their utterances to the physical environment around them, which adds critical context. For example, any use of demonstratives ("this one," "those," etc.) to refer to physical entities must almost necessarily be coupled with a deictic gesture to be interpretable.

While pointing can add useful context to communication, relying only on non-verbal deictic gesture, such as pointing, does not always guarantee accurate

target selection. Various experiments have been run to determine the potential increase in accuracy of pointing when combined with other features, such as speech [5]. In the mentioned study researchers experimented the effectiveness of single plane pointing in an augmented reality, from various perspectives, with and without speech. Participants were required to either point at or "identify" (with pointing and speech) an intended target from various perspectives. They found that combining speech and gesture, accuracy was increased, however there were still errors selecting the intended target.

In [16], participants interacted with a virtual avatar using a combination of gesture and speech in a shared construction task. Subjects were placed in one of four conditions that varied the information presented to them and the presence of physical cues in the environment that served as distractors. It was found that users adapted the direction of their deixis toward the correct target region, except in cases when explicitly misleading information about the role of the surrounding physical environment was presented.

These studies and more follow from a history of gesture semantics that continues traditions of viewing gesture as either *simulated action* [6, 19] or a general mode of reference [4, 28]. Lascarides and Stone [18] interpret gesture on the basis of the co-perception of gesture and denotatum. This is critical for deictic gesture in particular as the use of deictic gesture G presupposes that its interpretation function $[\![G]\!]$ is also co-perceptible by the intended recipient of the gesture. Given the typical use of deixis as an indicator of physical items, $[\![G]\!]$ readily resolves to an item in the environment under this model. The gesture abstract meaning representation (GAMR) language that we leverage in this paper [3] also builds directly on Lascarides and Stone's division of deictic and iconic gestures [17].

van der Sluis and Krahmer [25] studied deixis in the context of multimodal referring expressions and found a main effect of distance. The decreased specificity of pointing over distance can be modeled as a "cone" *a la* Kranstedt et al. [12]—a volume narrower at the vertex (the pointing digit) and wider as distance from the digit increases. In this paper we experiment with a "pointing *frustum*" (viz. a cone with the tip truncated) to create a region of detection around the pointing vector. This combined with other features in a multimodal system may further improve the accuracy of pointing as a means to select the intended target.

3 Methodologies

In this section we introduce our dataset, and the tools and methods used in our experiments.

3.1 Weights Task Dataset

The Weights Task Dataset [10] (WTD) is a collection of audiovisual recordings of a collaborative problem solving (CPS) task. Groups of 3 work together to determine the weights of various small colored blocks using a tabletop balance scale. The data comprises 10 groups, each including videos from 3 Azure Kinect

RGBD cameras at different angles [2]. The participants do not know that weights of the blocks follow an instance of the Fibonacci sequence, where each block is the combined weight of the previous two smaller blocks. At the end of the task the group is asked to determine the weight of one mystery block which, according to the pattern, is the combined weight of the previous two blocks. The dataset totals approximately 4 hours of recordings. Figure 1 shows an example still from Group 1 and Group 2 of the Weight Task Dataset.

Fig. 1: Sample stills from Groups 1 and 2 of the Weights Task Dataset.

This dataset contains many different forms of real-world multimodal communication in the course of the task-oriented collaboration, including but not limited to speech, gesture, body language, and gaze. While these features exist in each of the ten groups, the exact language and gestures used to communicate can vary, often widely, between groups and even participants. In the domain of deixis alone, for example, specific deictic gestures might range from gesturing to a target with the entire hand to using one or more fingers, or even an object like a pen, giving us diverse, challenging and realistic data to experiment with. Additionally, the blocks themselves are quite small, at 1.5" (38.1mm) or 2" (50.8mm) on a side, in a working space (table and chairs) that is approximately 5' × 5' (1.52m × 1.52m). Naturally, pointing with the fingers only (rather than extending the entire arm) is the most common form of deictic gesture used to indicate targets in this dataset. Thus pointing is an important feature that can be used to select the intended target; deixis might be used to indicate objects that are the subject of a current question or subgoal, or used to draw attention of other group members to specific items, meaning that it is a potentially important predictor of how the collaborative task will unfold. Referring back to Figure 1, in both examples participants are seen gesturing to a specific block, using deictic pointing gestures.

Data Preprocessing A few data prepossessing steps were required in order to test our proposed target selection solution using the WTD. Human-annotation of frames in which pointing gestures occurred were gathered for a subset of groups. This process involved manually stepping through each frame and marking a

participant ID[1] along with the start and stop frame for each deictic gesture. For each manually annotated frame we also saved the block's color, quaternion describing its location, location in 3D Cartesian space, and 2D bounding box information. This gave us a maximally precise object location in each frame against which to assess the quality of object selection with automated deixis detection compared to a human-annotated ground truth. From there we ran a linear interpolation algorithm to fill in the object locations for the intervening video frames. Figure 2 shows an example of the target blocks on the scale, with and without the overlaid 2D bounding box drawn from the manual annotations. Because of the time required to annotate each video, only a representative subset of groups were selected for our experiments.

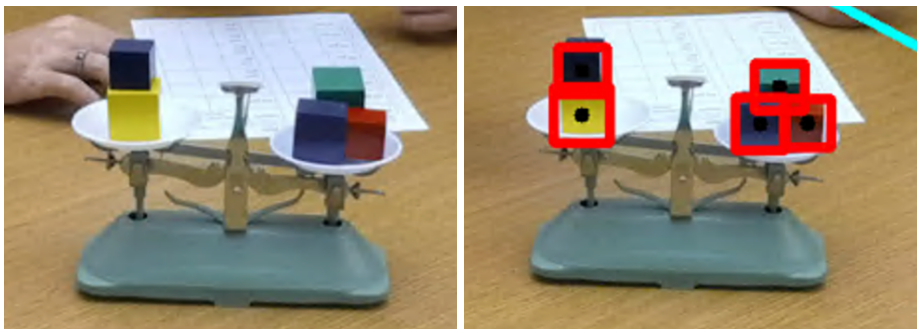


Fig. 2: Target blocks, with and without bounding box overlay.

3.2 Robust Gesture Recognition

In order to accurately determine both when a participant is pointing, and the intended target of that point within in a scene, certain specific information needs to be extracted from the video frame. Gesture recognition has previously been treated as data-hungry computer vision problem [21], and while sophisticated approaches like vision transformers remain state of the art, a high-throughput experimental scenario like ours demands a more lightweight solution. In [8] we demonstrated that our robust gesture recognition pipeline [26] displayed competitive performance with much larger models on the gestures of interest in the WTD using only automatically-extracted 6 degrees of freedom joint positions instead of full pixel or depth channels.

Our method consists of a pipeline to automatically detect *preparatory*, *stroke*, and *recovery* phases of gestures, using joint positions automatically extracted from the video signal before classification [26]. Design choices when developing that framework were based on the gesture semantics previously defined by the

[1] Participants are conventionally indexed 1–3 from left to right in the video frame.

community [1, 9, 18] (namely Kendon's pre-/post-stroke *hold* formulation). Here, we leverage that system to aid in determining when someone is pointing in order to determine targets of interest in 3D space. Our pipeline which consists of three stages—a static classification model, movement segmentation algorithm, and phase breakdown—distills videos down to "key frames," which we define as the union of the pre-stroke, stroke, and post-stroke phases, where the most of the semantically significant movement for a gesture takes place. In this use case, these key frame span the semantically significant movement of pointing gestures. Figure 3 shows the gesture detection pipeline, with the addition of our steps taken for point based target detection.

The static classification model recognizes the general static shape of complex gestures when in a *hold* phase. The movement segmentation routine aids in breaking down a video into segments of similar movements. The phase breakdown uses the classification model and video segments to identify and classify the segments and frames that are in a *hold* phase, and thus most semantically significant, or adjacent to the most semantically-significant frames.

We hypothesize that for deictic gestures such as pointing, the "key frames" dictate when a participant is not just pointing but also pointing toward the intended target, thus lining up the object, reference point (in this case, the body), and the frame of reference [20]. Using the output of our gesture detection pipeline, we can determine which frames are candidates for containing pointing gestures, and determine from there determine the intended target.

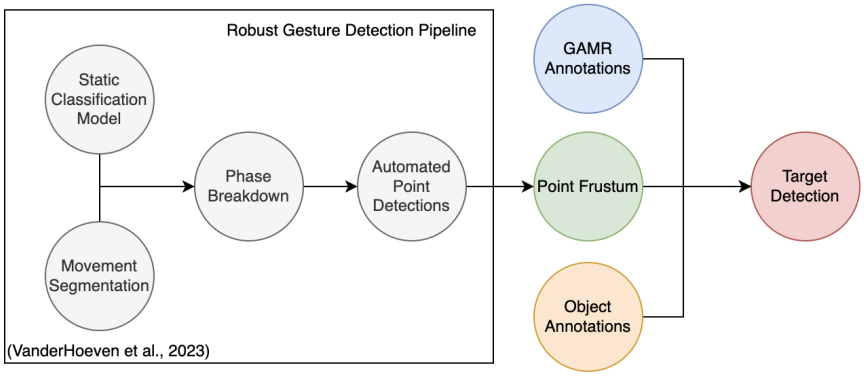


Fig. 3: Complex gesture and target detection pipeline. Items within the box denote components already established in VanderHoeven et al. (2023) [26].

3.3 MediaPipe

As mentioned, our recognition pipeline depends on automatically-extracted joint positions rather than raw pixels, and these must be extracted using some off-the-shelf software. Hand detection tools, such as MediaPipe, an open source library developed by Google [29], support such gesture recognition methods by performing this automatic extraction (see Figure 4). These joint positions, or *landmarks*, of detected hands [29] can then be used to train custom gesture recognition models such as ours with a wide range of applications.

MediaPipe has a few limitations that need to be overcome to handle more complex scenarios with multiple participants. While MediaPipe has the ability to return multiple hands from a single frame, the ordering of the hands is not consistent and can vary frame to frame. Because of this, participants' hands can be mixed up, leading to inconsistent hand tracking. In Section 3.5 we detail how we handled this issue.

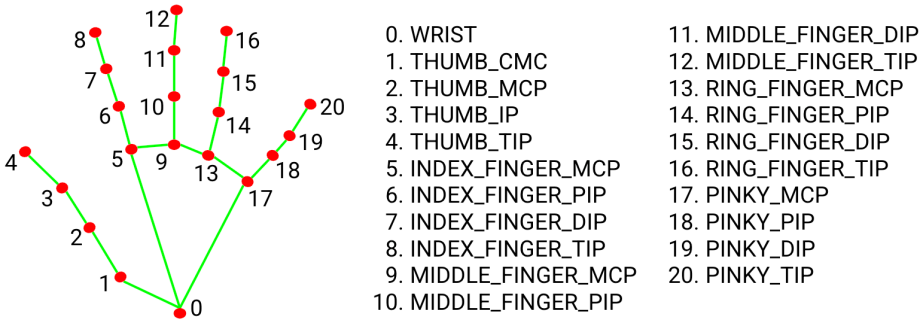


Fig. 4: MediaPipe Hand Landmarks (reproduced from [29]).

3.4 Pointing Frustum

Leveraging MediaPipe and our robust gesture recognition pipeline, we can identify frames of interest for deictic gestures, and from there use the hand landmarks to calculate a pointing vector to identify target objects in a scene. For the purposes of our experiments we calculate our pointing vector by extending a ray through the base and tip of the index finger, comprising the MediaPipe landmarks at index 5 and 8, respectively. We then extend the vector out into the environment 5 times the distance from finger base joint to fingertip, starting from the tip of the index finger (joint 8). Figure 5 shows the joints used to create the pointing vector relative to the MediaPipe landmarks.

When using a vector embedded within a single plane to detect targets of interest, it is very unlikely that the vector and object of interest will line up perfectly. Because of this, we use a "pointing frustum" to create a target detection region. A frustum is a geometric shape resembling a cone, where a radius

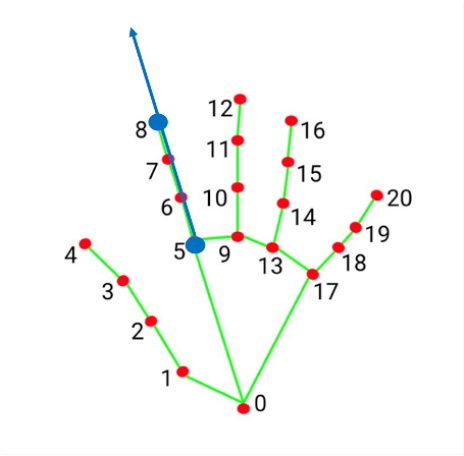


Fig. 5: Pointing vector relative to MediaPipe landmarks.

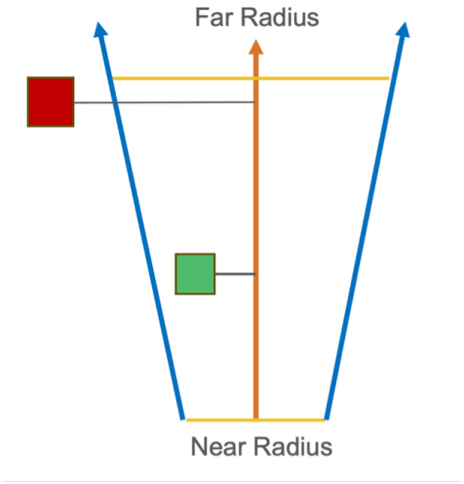


Fig. 6: Top down view of the pointing frustum. The red block is an example of an object that falls *outside* the detection region, the green block is an example of an object that falls *inside* the detection region and would be marked a target of interest.

value is set for the top and bottom of the cone. By using a pointing frustum we can specify a "near" (or top) radius at the tip of the index finger and "far" (or bottom) radius at the end of the pointing vector to allow increased granularity when experimenting with detection regions. This reproduces the pointing cone semantics of [12], and makes it extensible to allow for different levels of

$$\begin{aligned}\vec{P} &= p2 - p1 \\ \vec{B} &= p3 - p1 \\ \vec{A} &= \frac{\vec{B} \cdot \vec{P}}{\|\vec{P}\|^2}\vec{P} \\ x &= p1 + \vec{A}\end{aligned} \tag{1}$$

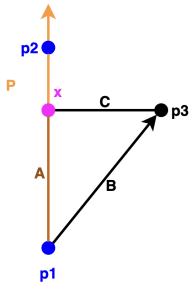


Fig. 7: Calculation of pointing vector target, where P represents the pointing vector, $p3$ represents the center of a target object, and x represents $p3$ projected onto P

imprecision at distance. Figure 6 shows a top down view of the frustum, noting that in a real scenario, the frustum is a three-dimensional volume with circular cross-sections. The red block in the figure denotes a target object outside the detection region, whereas the green box is an example of a target of interest. We determine if a target is in the detection region by first finding the location of the target perpendicular to the vector, outlined in Figure 7. From there we can find the radius of the frustum at that point and determine if the center of the object is within that radius. This process is depicted in Figure 7 and Equation 1. x denotes the center of the candidate target object projected onto the pointing vector (P). $r_d = r_n + \frac{r_f - r_n}{\|\vec{P}\|}\|\vec{A}\|$ gives the radius of the pointing frustum at distance $\|\vec{A}\|$ from the "near" plane (where r_n and r_f are the near and far radii, respectively). If the distance from the center of the candidate target to its projection x is less than or equal to r_d, the candidate lies within the pointing frustum and is considered "retrieved."

3.5 Azure Landmarks

In order to implement gesture recognition on multiple participants, additional assurances needed to be built on top of the MediaPipe hand recognition [26]. MediaPipe includes the ability to track multiple hands but does not guarantee the order of the returned hands. This means that participants' hands can get mixed up (for instance, if they overlap, or leave the frame and return), leading to incorrect assignment and therefore gesture classifications and attributions. For instance if participant 1 is pointing at the blue block, but the gesture is associated with participant 2, the result would be inaccurate representations of gestures within the scene. We therefore took the locations of joints on the bodies of the different participants, which were extracted from the depth video stream (see Figure 8). Using these, we calculated a bounding box on each of the participants' hands, to allow MediaPipe to retrieve the hand joints from a localized area. By tracking the bounding box to the wrist joint according to

Azure, we could more consistently associate hands (and thus complex gestures and movements) with a participant.

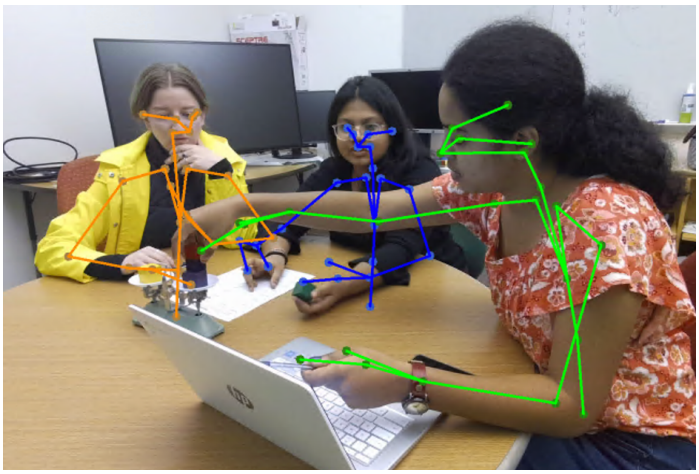


Fig. 8: Azure body landmarks overlaid on a frame.

3.6 Depth Information

In addition to body landmark locations, the Azure SDK facilitates retrieving framewise depth information. This depth information is saved as grayscale Z-coordinate values for each pixel in field of view, measured in millimeters. Figure 9 shows an example depth frame, with a semi-opaque overlay of the RGB data. The depth information allowed us to convert hand landmarks and object locations between two-dimensional and three-dimensional space, thus allowing us to create a pointing vector, and detect targets in three-dimensional space.

3.7 GAMR

Ground truth target object annotations are provided in the WTD in the form of Gesture Abstract Meaning Representation (GAMR) annotations [3]. GAMR annotations comprise up of four main parts, the gesture type, gesturer, semantic content, and addressee. For the purposes of our experiments we focus only on the *deictic* gesture type, or gestures that refer to a location by pointing. Figure 10 shows an example of a pointing gesture referencing a block. `ARG0` denotes the gesturer, `ARG1` the semantic content of the gesture and `ARG2` is the addressee or intended recipient. Figures 11 and 12 show examples of GAMR annotations from the WTD. Note that in, e.g., Figure 11, the gesturer (`ARG0`) is `participant_1` and semantic content of the gesture (`ARG1`) is the `blue_block`. We used this information in conjunction with the targets selected by intersection with the pointing frustum to verify if object selections were correct.

Fig. 9: Azure Depth Data with RGB overlay

```
(d / deixis-GA
  :ARG0 (g / gesturer)
  :ARG1 (b / block)
  :ARG2 (a / addressee))
```

Fig. 10: Deixis GAMR template according to [3].

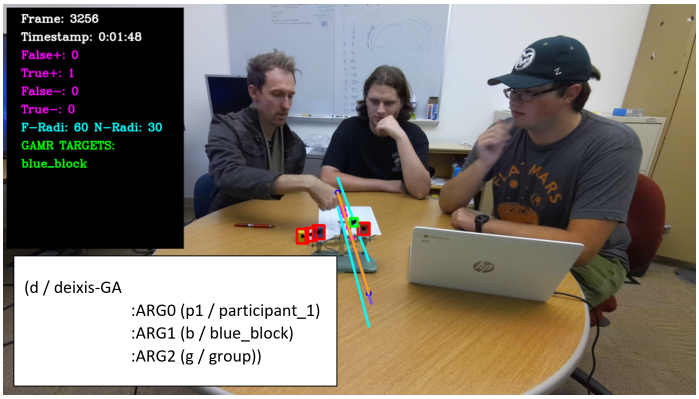


Fig. 11: Group 1 deixis GAMR example

4 Experiments

Our experimental protocol involved assessing the values of near and far frustum radii that provided the best possible and most consistent object selection across multiple groups. Relevant video frames that were tested against included those which were annotated with GAMR type `deixis-GA`, and had been annotated as containing a point gesture, or the gesture recognizer detected one. From there

Fig. 12: Group 2 deixis GAMR example.

we assessed which objects intersected the pointing frustum. Per-frame recall, precision, and F1 were calculated against the `ARG1` of the GAMR annotation. This repeated for each relevant frame and we kept a running average for each metric across the entire video. In order to determine the number of correct inferences and type I or type II errors based on selected blocks, we created the following guidelines using the GAMR annotations:

- If `ARG1` is a single block, if the selected block matches the annotation, it is considered a true positive. If selected blocks do not match the annotation, they are considered false positives.
- If `ARG1` is a combination of blocks, such as when the GAMR annotations did not specify a single block as the denoted target, but rather a set, each selected block that matches a block in `ARG1` is considered a true positive. Selected blocks not contained in `ARG1` are considered false positives.
- If a block included in ARG1 is not selected, it is considered a false negative.

The subset of groups we evaluated against included groups 1, 2, 4, and 5 of the WTD (see Section. 3.1).

5 Results and Discussion

Table 1 shows the average F1, recall and precision across all 4 videos for different combinations of radii, assessed against both the human-annotated pointing frames and those retrieved by the automated gesture detection. It is worth noting the variability in F1 scores; in many cases the standard deviation is almost the same as the average. This indicates the challenge in selecting a single set of frustum radii for the most effective target selection. Additionally, the F1 scores over the the human-annotated frames and the automatically selected frames are generally very similar, showing that our automated pipeline's frame selection achieves similar results when compared to human annotators.

Table 1: Average target detection F1 for human annotated and automatically detected frames from groups 1, 2, 4 and 5.

Near	Far	μ Human F1	σ Human F1	μ Auto F1	σ Auto F1
20	50	0.187	0.170	0.185	0.192
30	60	0.213	0.175	0.199	0.191
40	70	0.282	0.254	0.275	0.278
50	80	0.349	0.235	0.338	0.274
60	90	0.401	0.216	0.384	0.267
70	100	0.417	0.207	0.404	0.260
80	110	0.420	0.210	0.404	0.261
90	120	0.418	0.206	0.400	0.261
100	130	0.416	0.195	0.396	0.253

The relatively high standard deviations shown in Table 1 indicate the variation present across groups in the dataset. We also present group-wise results showing the average metrics across all frames vs. the radius sizes. This provides additional granularity in determining the most effective radius combination on a per group basis, as opposed to selecting and testing radius combinations one at a time.

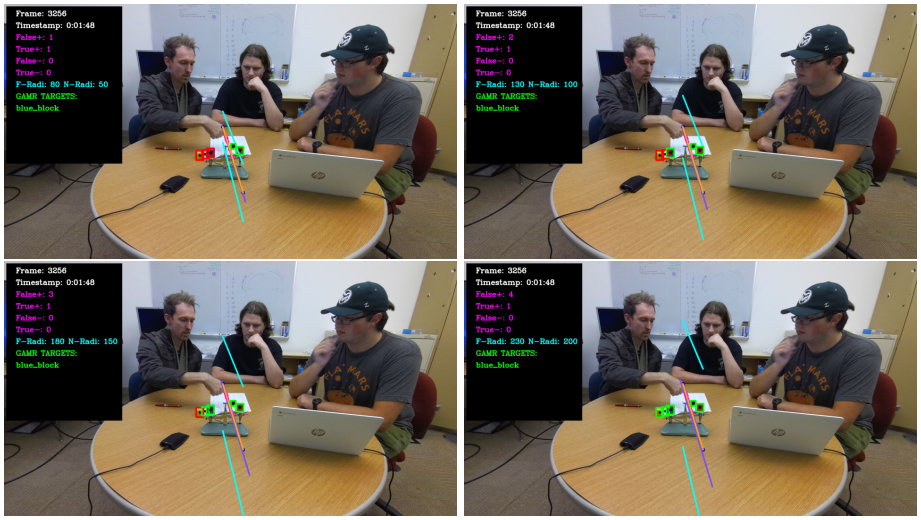


Fig. 13: Incremental radius step example, Group 1.

Figures 13 and 14 show examples of the incremental radius steps, and show how as the size of the pointing frustum grows, as the detected targets change. Blocks outlines in green indicate those selected as targets of pointing. Red indi-

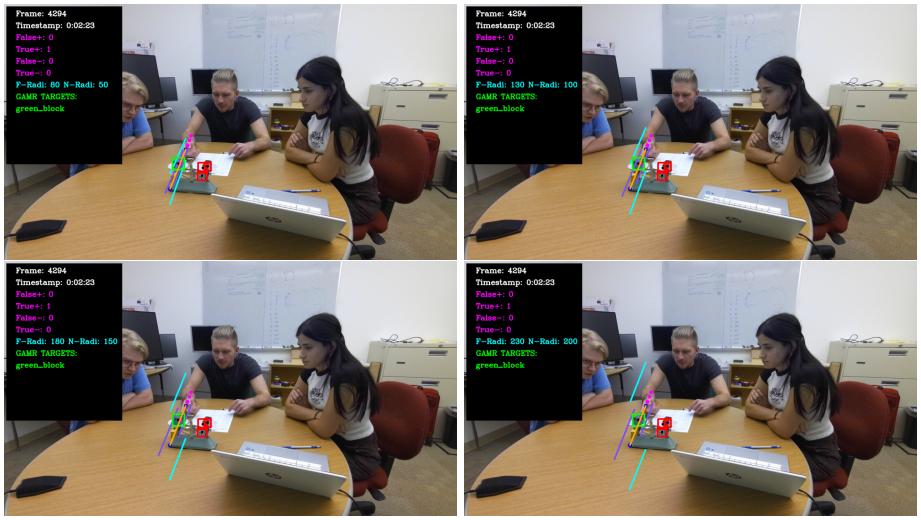


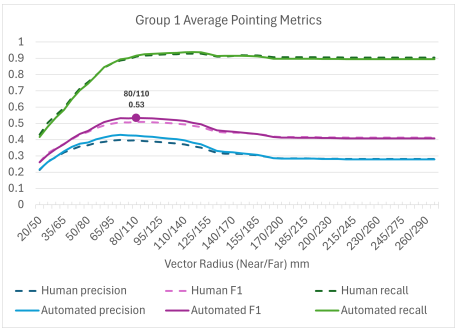
Fig. 14: Incremental radius step Example, Group 2.

cates those not selected. Note that in Figure 14, as the radius grows the green block is the only one ever selected. This is because the remaining blocks are resting on the paper behind the scale, and are therefore behind the origin of the pointing vector and the frustum's near plane.

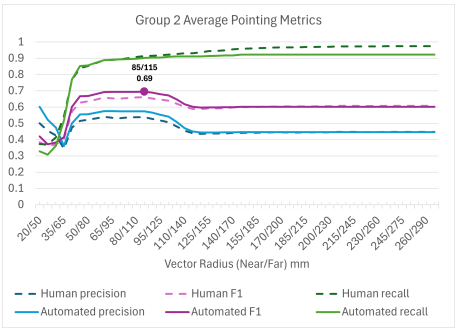
Figures 15a–15d show experimental results across a range of different near and far radius combination for each group, averaged across all relevant frames of the video. We compare metrics over those frames annotated by humans (ground truth) and those where pointing gestures were detected by the automated detection pipeline (therefore comprising an end-to-end system with gesture detection and target selection in a single step). Maximum F1 score is indicated with the purple dot. The maximum F1 scores vary anywhere between 0.33 (Group 4) and 0.69 (Group 2). This range is likely due to variability in accuracy and style of pointing as they are used by each participant/group. In addition to human inaccuracy, pointing in such a small space is likely to return more than one object as the radii grow. Because of this, as the frustum size increases we have the potential to return more false positive targets. This is reflected in the increase in recall as the radii grow, but the eventual decline in precision and F1.

In most cases, except Group 4, using the end-to-end system, where frames were selected by the automated gesture pipeline, outperformed detection over human annotations. In Group 5 particularly, the gesture pipeline frames eventually overtook the human annotated frames by about 0.1 F1 overall and thereafter remained consistent. We hypothesize this may be because the automatically selected frames are ones that the static classification model recognizes as a point with the index finger. In Group 5, sometimes participants would point with pens, or would gesture at the blocks using their entire hand. Overcoming this limitation is another potential area for future work. In other groups the accuracy statistic

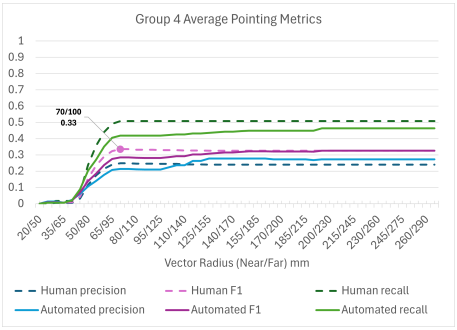
of the human annotated frames more closely matches the automated frames, indicating that the participants were more likely to point using the index finger (as expected).



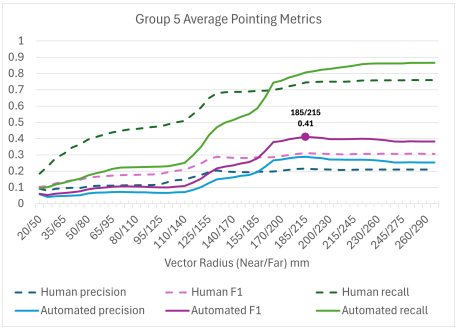
(a) Group 1 target selection statistics.



(b) Group 2 target selection statistics.



(c) Group 4 target selection statistics.



(d) Group 5 target selection statistics.

Fig. 15: Average precision, recall, and F1 for 4 test groups, averaged over all frames. Solid lines indicate where frame selection was performed using the robust gesture detection pipeline [26]. Dashed lines indicate where frame selection was performed by human annotation.

6 Conclusion and Future Work

Deictic gestures are common in small group communication as a means to indicate objects and referents in context. The ability to both identify deictic gestures, such as pointing, and to identify their denotata in context is a critical capability in interpreting multimodal communicative acts in situated physical shared tasks. In this paper we leveraged a previously-developed pipeline to help automatically detect semantically significant movement for a given gesture, and save off the "key frames" for future use [26]. Here we leveraged that pipeline as a means to

automatically detect pointing gestures, and created a pointing frustum based on that information for target detection. The combination of our automated pointing detection and the pointing frustum showed promise as a means to detected the intended target of a participant. Our detection pipeline preformed similarly too, and in some situations better than, the human annotated ground truth annotation as a reliable and robust way of performing object selection via deixis in challenging, real-world data.

There is room for improvement when it comes to the overall accuracy of target detection, with the maximum F1 achieved across individual groups and a range of radius combinations being about 0.69. Future work includes leveraging additional context from other features, like speech, in a larger multimodal system to further aid in narrowing down the exact intended target in a small space. Speech can help provide a signal to select which of a set of objects is the true intended target. It is important to emphasize here that in this paper target selection is conducted using only deixis, and in the actual group task, pointing inaccuracies likely did not hinder participants' ability communicate with each other, as they relied on additional communicative modalities, such as speech and gaze, to provide additional context and information to each other. These additional features could be used as tools to help further signal the true intended target from a group of selected targets. For instance, this could be done by aligning the speech signal with gesture for disambiguation, such as using language to select one among a set of objects indicated through deixis, as done in [14, 15, 22–24].

Better accuracy in object selection using deixis may also be aided with the addition of other features. For instance, objects that were the anchor of recent actions (e.g., recently moved blocks) may be more likely to be a deictic target, because partial information about them is more likely to be known after the action. Therefore they may be more likely to be the denotatum of a spoken demonstrative, and thus singled out with deixis.

Furthermore, we relied on human object annotations as the ground truth against which to assess our performance. A true end-to-end system for target selection via deixis would not only perform pointing detection and frustum construction automatically, as we do, but also automatically detect the positions of the blocks in the video. Object detection via methods such as 6 degrees of freedom object pose would significantly reduce the preprocessing time required to leverage our pipeline, allowing us to experiment on more real world scenarios.

Finally, when humans engage in collaborative problem solving (CPS) tasks such as the Weights Task, multiple simultaneous communicative modalities are implicated. The ability to detect gestures and make inferences about their meanings is a critical capability for automated agents that support human-human collaboration, as in real-time project teams or classrooms. Approaches need to be lightweight and extensible to create tractable methods for interactive AI in supporting a wider range of CPS tasks [27].

Acknowledgements

This work was partially supported by the National Science Foundation under award DRL 2019805 to Colorado State University. The views expressed are those of the authors and do not reflect the official policy or position of the U.S. Government. All errors and mistakes are, of course, the responsibilities of the authors. Special thanks to Nathan Kampbell for the linear interpolation tool used in Section 3.1, and to Jade Collins and Carlos Mabrey for extensive data annotation.

References

1. Arnheim, R.: Hand and Mind: What Gestures Reveal about Thought by David McNeill. Leonardo **27**(4), 358–358 (1994), publisher: The MIT Press
2. Bradford, M., Khebour, I., Blanchard, N., Krishnaswamy, N.: Automatic detection of collaborative states in small groups using multimodal features. In: Proceedings of the 24th International Conference on Artificial Intelligence in Education (2023)
3. Brutti, R., Donatelli, L., Lai, K., Pustejovsky, J.: Abstract Meaning Representation for Gesture. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. pp. 1576–1583. European Language Resources Association, Marseille, France (Jun 2022), https://aclanthology.org/2022.lrec-1.169
4. Clark, H.H., Schreuder, R., Buttrick, S.: Common ground at the understanding of demonstrative reference. Journal of verbal learning and verbal behavior **22**(2), 245–258 (1983)
5. Herbort, O., Krause, L.M.: The Efficiency of Augmented Pointing with and Without Speech in a Collaborative Virtual Environment. In: Duffy, V.G. (ed.) Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. pp. 510–524. Lecture Notes in Computer Science, Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-35741-1_37
6. Hostetter, A.B., Alibali, M.W.: Visible embodiment: Gestures as simulated action. Psychonomic bulletin & review **15**, 495–514 (2008)
7. Hu, Z., Xu, Y., Lin, W., Wang, Z., Sun, Z.: Augmented pointing gesture estimation for human-robot interaction. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 6416–6422 (2022). https://doi.org/10.1109/ICRA46639.2022.9811617
8. Kandoi, C., Jung, C., Mannan, S., VanderHoeven, H., Meisman, Q., Krishnaswamy, N., Blanchard, N.: Intentional microgesture recognition for extended human-computer interaction. In: International Conference on Human-Computer Interaction. pp. 499–518. Springer (2023)
9. Kendon, A.: Gesticulation and speech: Two aspects of the process of utterance. The relationship of verbal and nonverbal communication **25**(1980), 207–227 (1980)
10. Khebour, I., Brutti, R., Dey, I., Dickler, R., Sikes, K., Lai, K., Bradford, M., Cates, B., Hansen, P., Jung, C., et al.: When text and speech are not enough: A multimodal dataset of collaboration in a situated task (2024)
11. Kita, S.: Pointing: A foundational building block of human communication. Pointing: Where language, culture, and cognition meet pp. 1–8 (2003)
12. Kranstedt, A., Lücking, A., Pfeiffer, T., Rieser, H., Wachsmuth, I.: Deixis: How to determine demonstrated objects using a pointing cone. In: Gesture in Human-Computer Interaction and Simulation: 6th International Gesture Workshop, GW

2005, Berder Island, France, May 18-20, 2005, Revised Selected Papers 6. pp. 300–311. Springer (2006)
13. Kranstedt, A., Wachsmuth, I.: Incremental generation of multimodal deixis referring to objects. In: Proceedings of the Tenth European Workshop on Natural Language Generation (ENLG-05) (2005)
14. Krishnaswamy, N., Narayana, P., Bangar, R., Rim, K., Patil, D., McNeely-White, D., Ruiz, J., Draper, B., Beveridge, R., Pustejovsky, J.: Diana's world: A situated multimodal interactive agent. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13618–13619 (2020)
15. Krishnaswamy, N., Narayana, P., Wang, I., Rim, K., Bangar, R., Patil, D., Mulay, G., Beveridge, R., Ruiz, J., Draper, B., et al.: Communicating and acting: Understanding gesture in simulation semantics. In: IWCS 2017—12th International Conference on Computational Semantics—Short papers (2017)
16. Krishnaswamy, N., Pustejovsky, J.: Deictic adaptation in a virtual environment. In: Spatial Cognition XI: 11th International Conference, Spatial Cognition 2018, Tübingen, Germany, September 5-8, 2018, Proceedings 11. pp. 180–196. Springer (2018)
17. Lascarides, A., Stone, M.: Formal semantics for iconic gesture. Universität Potsdam (2006)
18. Lascarides, A., Stone, M.: A formal semantic analysis of gesture. Journal of Semantics **26**(4), 393–449 (2009), publisher: Oxford University Press
19. McNeill, D.: Language and gesture, vol. 2. Cambridge University Press (2000)
20. Moratz, R., Nebel, B., Freksa, C.: Qualitative spatial reasoning about relative position: The tradeoff between strong formal properties and successful reasoning about route graphs. In: Spatial Cognition III: Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning 8. pp. 385–400. Springer (2003)
21. Narayana, P., Beveridge, R., Draper, B.A.: Gesture recognition: Focus on the hands. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5235–5244 (2018)
22. Pustejovsky, J., Krishnaswamy, N.: Embodied human-computer interactions through situated grounding. In: Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents. pp. 1–3 (2020)
23. Pustejovsky, J., Krishnaswamy, N.: Situated meaning in multimodal dialogue: human-robot and human-computer interactions. Traitement Automatique des Langues **61**(3), 17–41 (2020)
24. Pustejovsky, J., Krishnaswamy, N.: Embodied human computer interaction. KI-Künstliche Intelligenz **35**(3-4), 307–327 (2021)
25. van der Sluis, I., Krahmer, E.: The influence of target size and distance on the production of speech and gesture in multimodal referring expressions. In: Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'04) (2004)
26. VanderHoeven, H., Blanchard, N., Krishnaswamy, N.: Robust motion recognition using gesture phase annotation. In: Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management: 14th International Conference, DHM 2023, Held as Part of the 25th HCI International Conference, HCII 2023, Copenhagen, Denmark, July 23–28, 2023, Proceedings, Part I,. pp. 592–608. Springer-Verlag, Berlin, Heidelberg (Jul 2023), https://doi.org/10.1007/978-3-031-35741-1_42

27. VanderHoeven, H., Bradford, M., Jung, C., Khebour, I., Lai, K., Pustejovsky, J., Krishnaswamy, N., Blanchard, N.: Multimodal design for interactive collaborative problem-solving support. In: Human-Computer Interaction. Theoretical Approaches and Design Methods: Thematic Area, HCI 2024, Held as Part of the 26th HCI International Conference, HCII 2024. Springer (2024)
28. Volterra, V., Caselli, M.C., Capirci, O., Pizzuto, E.: Gesture and the emergence and development of language. Beyond nature-nurture pp. 53–90 (2004)
29. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L., Grundmann, M.: Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214 (2020)