Collaboration with Virtual Agents using Natural Language Processing and Computer Vision in Virtual Reality

Andrew Miller* ECEN Department
Pranav Vaidik Dhulipala† ECEN Department
Abishalini Sivaraman‡ ECEN Department
Swarnabha Roy § ECEN Department
Stavros Kalafatis ¶ ECEN Department
Mohd Faisal Khan ‖ VIZA Department
Ann McNamara ** VIZA Department

Texas A&M University

Abstract

Modern virtual reality environments consist of a huge variety of virtual objects, but creating a knowledge base about each item's associated attributes can be very intensive and limit the flexibility of user interaction with artificial intelligence components. We introduce a method of natural language processing driven computer vision for improved human-like intelligence in virtual agents. A natural language processing interface allows a user to easily and flexibly specify an attribute that is uniquely related to virtual objects, while computer vision is used to identify the object to which that attribute applies to. This combination allows for potential significant increase in the perceived intelligence of the virtual artificial intelligence agent through the ability of the agent to learn from the user. Preliminary user studies showed support that natural language driven computer vision can effectively be used for human computer collaboration in virtual environments.

Index Terms: Human-centered computing—Interaction Paradigms—Virtual Reality; Human-centered computing—Interaction Paradigms—Natural Language Interfaces; Human-centered computing—Interaction Paradigms—Natural Language Interfaces; Human-centered computing—Visualization—Visualization design and evaluation methods

1 Introduction

Modern advancements of virtual reality (VR) systems have led to improvements in how we can physically interact with computers.

Non-playable characters (NPCs) or virtual agents are very common components in all gaming modalities. NPCs create a sense of interaction [14] and presence [26] in the environment, and are also one of the major modalities game environments use to interact with humans. Recent developments in AI have introduced intelligent NPCs in games to make the user experience more natural and interesting [22, 27].Lately, there have also been a rise of AI based NPCs in virtual reality environments [10].

Action controllers for NPCs in games usually use a Finite State Machine [21]. This technique can make the virtual agent's reactions repetitive and can make the interaction feel unnatural. Another method to incorporate natural reacting virtual agents which involves the use of Monte Carlo Tree Searching [9]. This method is very computationally intensive. The paper explores using a finite state machine but with dynamic voice interaction. Since the robot is made to look like human, Natural Language Processing (NLP) is incorporated to achieve human like interaction. NLP will allow for the NPC interaction feel more natural. Another technique that can allow for more natural interaction is when the virtual objects can sense objects using visual observations. This can allow the virtual objects to ask questions about its environment without using pre-existing labels, which may or may not exist. Environment observation can be done using Computer Vision (CV).

CV is a branch of AI that is being applied to almost every field that involves visual information such as security [1, 20, 25], self-driving cars [6, 12], medicine [19, 24], and robotics [3, 5] . It has also found many applications in augmented reality (AR) [2], but it is usually ignored in VR. This is due to the nature of the VR environment where every object has attributes assigned to it before it is placed in the environment. However, as the complexity of the environment or scene increases, the objects in VR gain many visual attributes with respect to the environment that hard coded attributes cannot totally describe. Also, it is infeasible to predict and add all the new attributes that describe the object with respect to the environment. In recent years, a tremendous progress in VR hardware technology and computation power of computers have made rendering complex environments possible. Computer vision can potentially be used in this scenario where these visual descriptive attributes can be more easily detected and added to the objects.

In this simulation, we have used Computer Vision to help the robot recognize and assign the button color attributes visually through a python API called ML agents [13]. In the scene, none of the buttons have information about the color stored. The CV currently uses Hough transform [17] to detect the buttons and adds the color attributes to the buttons, which the robot uses to identify the button colors.

NLP, another rapidly growing branch of AI, is continually offering new advanced ways of thinking about natural human computer interaction [11, 16, 18, 23], but has yet to widely be adopted within VR applications. However, with VR being focused on providing the user with an immersive experience, NLP is a natural fit as a candidate to further this immersion. Humans use language for communication everywhere, from having general conversations, to giving task descriptions, to asking for clarification when collaborating on a task, to simply asking questions to understand something. NLP not only opens the door for intuitive verbal interactions with a computer, but also for deeper contextual understanding and clarification [15].

The combination of these benefits from both NLP and CV offer great promise as there are many times a user might need a flexible and natural form of interaction with a virtual agent, but which may require the virtual agent to learn more about its environment. Specifically for VR, our research is targeted toward evaluating how using NLP and CV will compare to commonly used controller-based VR interaction techniques for human-computer collaborative tasks

*e-mail: andrew.miller@tamu.edu
†e-mail: pranav.d1993@tamu.edu
‡e-mail: siva558@tamu.edu
§e-mail: swarnabha7@tamu.edu
¶e-mail: skalafatis-tamu@tamu.edu
‖e-mail: faisal_khan1994@tamu.edu
**e-mail: ann@tamu.edu

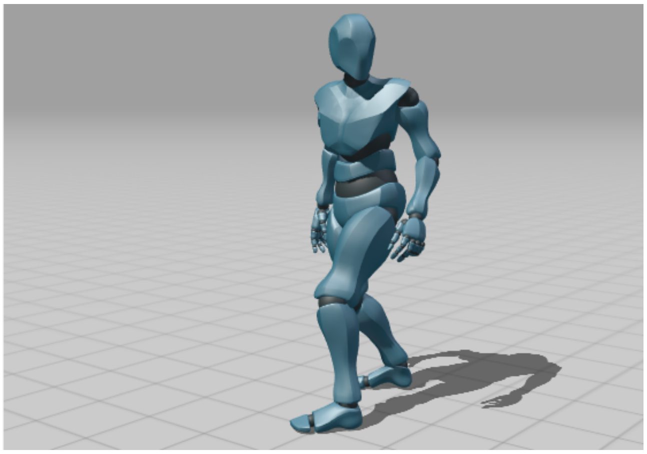

Figure 1: Virtual robot used in the environment

where the human and AI agent are not directly physically interacting with each other.

2 Related Work

The need for Computer Vision was highlighted in the paper on Collision avoidance in the presence of a virtual agent in small-scale virtual environments [4] (Bonsch, et al). The virtual agent in the experiment was set up to move away from the user when the user got close to the virtual agent. This action happens purely using location. Since, the virtual agent does not see the user getting closer, the act of moving away seemed unnatural. This can be fixed by adding computer vision to make the virtual agent see the user before moving away.

This rise in natural language interaction with robots was explored by Matuszek, et al in the article 'Learning to Parse Natural Language Commands to a Robot Control System'. The robot was trained to learn a parser based on example pairs of English commands and corresponding control language expressions. This ideas was extended to virtual robots in our project.

3 Experimental Setup

The experiment is setup in a VR environment containing a series of puzzles. The environment is built as a set of three rooms connected to each other via doorways. Each room is divided into two sections by a pit, where one section is for a virtual robot, and the other section is for the user. In each section there is a set of six colored buttons: purple, cyan, blue, red, green, and yellow. The objective of each room is for both the robot and user to push the button in their section in a specific sequence. The sequences for both the user and the robot are depicted on a clipboard, which is placed on the user's side. To complete each room, the user must correctly push the buttons according to the given sequence for their side, as well as tell the robot which order to press its buttons in according to the sequence for the robot's side. One of the puzzle rooms can be seen in Fig. 2.

The VR apparatus consisted of an HTC Vive Pro headset and controllers, along with the HTC Vive wireless adapter. The VR environment was built using Unity 2019.1.11f1. The Unity environment was run on a Windows 10 PC with 64 GB of DDR4 ram, an 8th gen i7-8700k CPU at 3.7-4.7 GHz, 2x NVIDIA GeForce GTX 1080 Ti Graphics Cards running in SLI, and 0.5 TB solid state hard drive. The playable VR space was 4.3 by 3.4 meters.

In this research, NLP is used to allow the user to specify the color attribute of the button they want the robot to press. The user can see what command has been interpreted from them, as well as has the flexibility to use related color names to refer to specific colors.

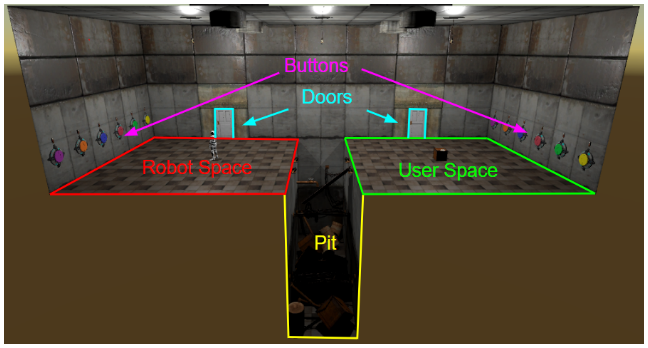

Figure 2: An example room from the VR environment.

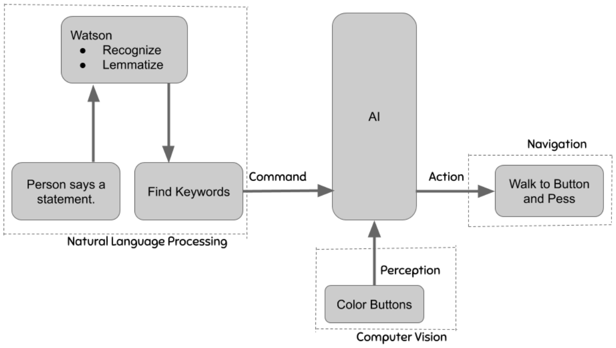

Figure 3: Block diagram of the environment building components.

The user may also choose to say more than a one color in a single statement in order to pass multiple interaction commands at once to the robot.

Participants were asked to complete tasks in a series of three connected rooms. In room one, the participant is given a tutorial of how the VR environment works. The participant shown how to navigate using teleportation or walking, how to use the VR controllers to interact with the buttons and pick up the clipboard, as well as how to interact with the robot using either voice commands or a point-and-click method. In room two, the participant was only able to interact with the robot using voice commands. In room three, the participant was only able to interact with the robot using the point-and-click method.

4 Methods

The implementation of this research was comprised of five primary components: computer vision, natural language processing, navigation, general artificial intelligence, and ray-casting selection.

4.1 Computer Vision

A basic computer vision (CV) method was used using OpenCV [8] to identify the buttons. The buttons were circular, hence hough transform [17] was used to detect circles and the colors were identified using the RGB values. The visual observations and identified colors are shared between the computer vision module and Unity through a python API provided by the ML-Agents toolkit [13]. In Unity environment, the buttons are ”tagged” with the respective colors identified, which NLP can take advantage of to identify the buttons.

4.2 Natural Language Processing

Natural Language Processing (NLP) was chosen to supplement the computer vision component as it provides the user with a natural way of specifying the item they wish the robot to find. Once the computer vision component has autonomously found the buttons and their respective colors in the VR environment, the NLP component allows the user to verbally specify which colored button they want the robot to interact with. The NLP model is built using IBM's Watson cloud services. The VR environment is connected to Watson using the IBM Watson Unity SDK. A C# script creates a service connection using Watson's Natural Language Understanding API for live audio processing.

While the user presses and holds down the left controller's trigger a holographic microphone is displayed and the script streams the live audio feed to Watson's speech recognizer. As the recognizer processes the audio, the interpreted text is displayed live to the user above the holographic microphone so that they can see what was interpreted from their statement. If specific keywords, such as a color or ”press” are stated, that text is highlighted in green . Is this research only button colors are used to drive the computer vision search and corresponding robot control, so only the stated colors of the buttons are extracted as commands and passed to the robot. To allow the user to have some flexibility in specifying colors, colors that are commonly called by several names can be referenced in several ways. The red button can be referenced as ”red” or ”orange”. The cyan button can be referenced as ”cyan”, ”teal”, ”turquoise”, or ”sky”. The purple button can be referenced as ”purple” or ”pink”.

Only simple color keywords were extracted and used to drive the robot interaction in this research as the addition of any descriptive keyword characteristics about the button, such as ”smaller” or ”to the left of”, would significantly complicate the computer vision model. Nevertheless, this combination of computer vision and NLP gives the user the flexibility of using their preferred naming convention for colors, while still letting the robot learn about the environment without prior knowledge.

4.3 Navigation

The user can move around the virtual environment by using a combination of teleportation and natural walking. Traditional walking might not be always feasible depending on the size of VR setup area.

The robot travels to destinations by calculating the optimal path using A* algorithm. The A* algorithm is the fastest graph search algorithm that finds the path that costs the least from source node to goal node. The nodes in our game references to dividing up the virtual robot floor space into grids. This method was tested to work inferior to the Unity navigation system, which was used in the final product for robot navigation.

The *NavMesh* function in Unity divides the terrain into polygons which serves as nodes for the A* algorithm [7]. The function approximates the walk-able areas and obstacles in the virtual environment for optimal path finding. The *NavMeshAgent* function is used to create the virtual robot which walks on the navigable terrain specified by the *NavMesh* function.

4.4 General Artificial Intelligence

This component involves interfacing between NLP, CV and Navigation components. This is like the brain of the robot that interacts with environment and the user by using CV, Navigation, and NLP. It processes the attributes received from the NLP algorithm to determine the instruction given by the user, takes a decision and then performs the tasks accordingly. The Artificial Intelligence (AI) extracts the color attributes from the user's speech. Once the attributes are processed and the decision is taken, it searches through the list of tagged buttons and finds the locations of those buttons. It then interacts with the navigation system to move to the buttons requested by the user. If the AI agent is unable to find the colors in the tagged attributes, it invokes CV to search through the environment which tags the untagged game objects in real time. The AI agent keeps on handshaking with the CV until the object is found and interacts with navigation to allow the CV component to change the Field of View.

4.5 Ray-Casting Selection

A commonly used ray-casting selection interaction was used as the baseline to compare our NLP driven CV interaction method against. It provides an alternative pointing technique for the user to interact with the robot. It creates a beam of light that comes out of the user's right hand and extends until it collides with an object. The user points the ray to the appropriate buttons on the robot's side one at a time, and pulls the trigger on their controller to make the robot press that particular button.

5 Hypothesis

In this project, we have introduced NLP and CV capabilities in virtual environment to make it more natural and immersive to the user. We hypothesize that CV is advantageous than traditional manual tagging of objects in the virtual environment since manual tagging is feasible only for small environments. CV allows real time auto labelling or assigning tags to objects without any manual intervention. Our second hypothesis is that NLP can improve the interaction of the user with the non-player character by making it more natural and immersive. Artificial Intelligence can help by learning more about the environment while also performing a two-way communication with the user. This project is to prove the concept that NLP and CV can be used to make these interactions without using a dialog tree for the human-robot conversation to achieve a dynamic voice in response to the player's questions.

6 Metrics

The aim of this research was to determine whether enabling a virtual robot to use CV and allowing a user to verbally describe an action to take will feel more natural than using standard ray-casting selection methods. All participants were asked questions to address the following questions:

1. Did it feel like the robot could understand what you were trying to communicate?

2. Did the robot struggle often with pressing the desired button?

3. Did it feel like the robot recognized the wrong button color?

4. Did it feel natural to interact with the robot using the NLP method?

5. Did it feel natural to interact with the robot using the ray-casting method?

6. Did you like interacting with the robot using the NLP method?

7. Did you like interacting with the robot using the ray-casting method?

8. How human/intelligent did the robot seem using the NLP method?

9. How human/intelligent did the robot seem using the ray-casting method?

10. How difficult was it to see/find the buttons you wanted to select?

11. How difficult was it to select the button you wanted to select?

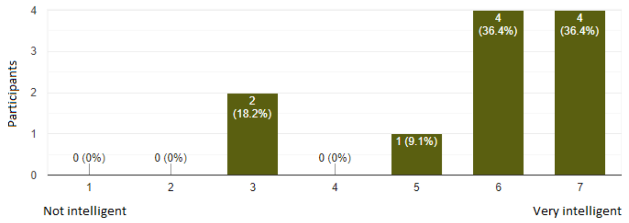

Figure 4: Responses to the question : How human/intelligent did the Point Click robot feel?

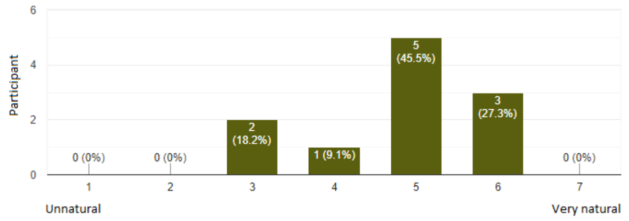

Figure 5: Responses to the question: Did it feel natural to interact with the NLP robot?

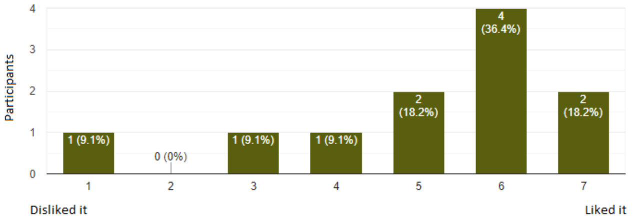

Figure 7: Responses to the question: How well did you like interacting with the NLP robot?

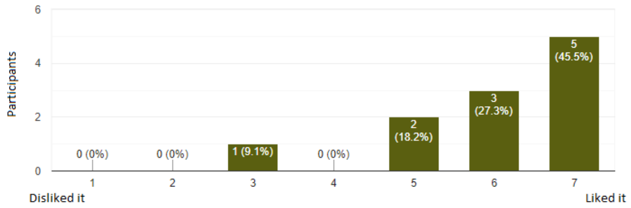

Figure 8: Responses to the question: How well did you like interacting with the ”Point-Click” robot?

7 Results

A preliminary study was performed with set of 11 participants to evaluate the existing VR environment implementation so that the necessary modification can be made for the full study to come. Of the 11 participants, 4 of them were female and 7 were male, the participants aged between 22 to 26, 5 participants had prior gaming experience, 5 participants had prior VR experience, and 10 participants were non-native English speakers. The results questionnaire each participant was asked after they completed the experiment are shown in Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11, Fig. 12, and Fig. 13. We observed that the participants, in general, preferred traditional pointing method over the NLP method for interaction with the robot. The overall user experience was more tedious for the Non-Gamers as compared to the Gamers. As per the results, the IBM Watson’s accent recognition was better for the females as compared to the males.

8 Discussion

While the collected test results are intended to be used a preliminary feedback to make necessary changes before the full study, several interesting observations were made. While some participants said they preferred the ray-casting selection method over the voice control due to the encountered translating problems, it was noted that most all of the participants had to point and click numerous times before they selected the button due to it being a small target if the user wasn’t at the edge of the pit. One user specifically noted this as making the ray-casting selection method feel unnatural, while the NLP method remained natural. Participants also noted that despite the complications with both methods as noted in the Limitations section, the task and environment seemed too simple to make a good distinction of performance with each interface. Lastly, while most of the results showed a close scoring of both methods across almost all the participants, the ray-casting selection method did seem to consistently perform better than the the NLP driven CV method. This was, however, most likely caused by the NLP model’s lack of training on non-native English speaker’s with slight accents on their pronunciations of certain colors such as ”purple”, ”teal”, ”cyan”, and ”turquoise”. Some users felt that the robot was slow to perform the task.

9 Limitations

While the participant responses did provide some insight into the efficacy of NLP driven CV to improve the perceived intelligence of a simple AI agent, a number of limitations to both the methodology and implementation were found. As some participants noted, the task of pushing the buttons and commanding the robot to push buttons was relatively easy to perform, and thus both methods of interacting with the robot scored relatively well regarding ease of

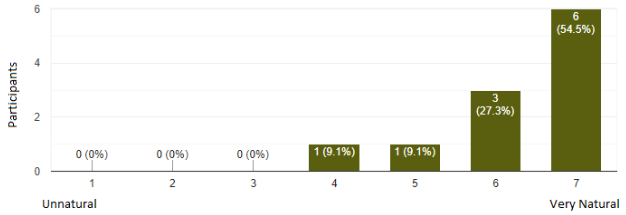

Figure 6: Responses to the question: Did it feel natural to interact with the ”Point-Click” robot?

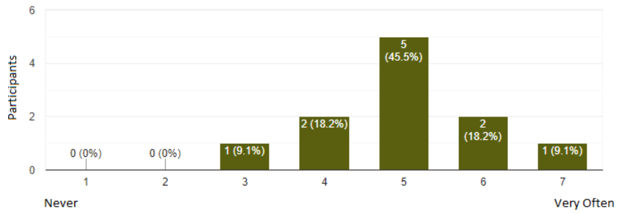

Figure 9: Responses to the argument: It felt like the NLP robot could understand what I was trying to communicate

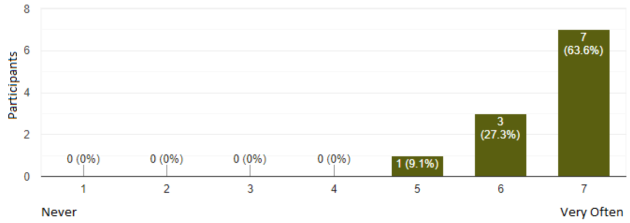

Figure 10: Responses to the argument: It felt like the ”Point-Click” robot could understand what I was trying to communicate

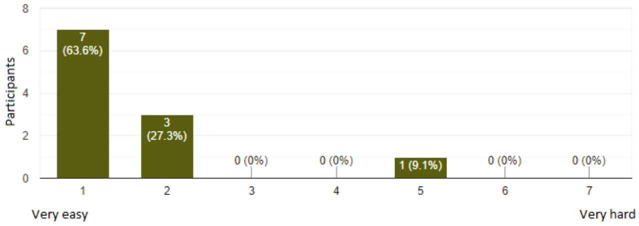

Figure 11: Responses to the question: How difficult was it to see/find the buttons you wanted to select?

use in selecting a button. Similarly, because the task was both easy to understand and to complete quickly, users found both methods enjoyable for interacting with the robot. Implementing both a more complex puzzle as well and one which takes longer to solve might offer greater delineation between comfort and ease-of-use of these two methods

Many of the participants had difficulty getting the NLP component to recognize when they said certain colors, causing them to have to repeat themselves numerous times until the NLP component recognized the color they were trying to say. While these participants could fluently speak English, they were not native English speakers, and it is suspected that the NLP model running on Watson’s cloud had difficulty understanding their accent.

Due to some complications using ML-Agents, participants experienced some temporary lag while moving between rooms while the buttons were identified and classified. This was largely caused by the current inability to simultaneously run the CV while the user was moving and an the robot animation was active.

Lastly, the NLP component was significantly trimmed down to only search for colors of the buttons. This is due the the inherent complexity of training a CV model to identify specific attributes. Every addition of a description such as ”find the larger button compared to...”, or ”find the button next to...” would require the CV model to understand relative sizes of objects, relative positions of objects, and so on.

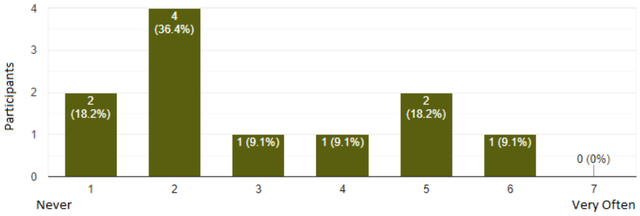

Figure 12: Responses to the argument: The NLP robot often struggled with pressing buttons

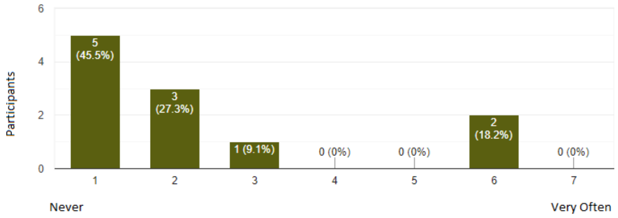

Figure 13: Responses to the argument: The Point Click robot often struggled with pressing buttons

10 Conclusions And Future Work

The research shows great promise for using NLP driven CV as an effective method of interaction with AI. The preliminary test results provided great feedback into needed improvements to conclusively compare NLP driven CV with other standard VR methods in future tests. Though the preliminary test results did not draw a clear distinction of which method of interaction with the robot consistently performed better and why, it did show support that NLP driven CV was still relatively intuitive and effective despite the encountered limitations. Future improvements to the CV and NLP components through training on larger domain data sets will likely remove much of the lack in distinctive performance of the two methods. An increase in participants and number of task trials, as well as improved task variety, task difficulty, task length, and user feedback while performing the task could further confirm the effectiveness of NLP driven CV in VR applications.

Acknowledgments

We would like to thank Professor Stavros Kalafatis, Dr. Ann McNamara, and Faisal Khan for advising this research project. We would also like to thank the Texas A&M Electrical and Computer Engineering Department and the Visualization Department for use of lab space to conduct testing and for the use of VR equipement and computers.

References

[1] A. Ali and E. Dagless. Computer vision for security surveillance and movement control. In *IEE Colloquium on Electronic Images and Image Processing in Security and Forensic Science*, pp. 6–1. IET, 1990.

[2] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.

[3] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008.

[4] A. Bonsch, B. Weyers, J. Wendt, S. Freitag, and T. W. Kuhlen. Collision avoidance in the presence of a virtual agent in small-scale virtual environments. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 145–148. IEEE, 2016.

[5] S. Chen, Y. Li, and N. M. Kwok. Active vision in robotic systems: A survey of recent developments. *International Journal of Robotics Research*, 30(11):1343–1377, 2011.

[6] Z. Chen and X. Huang. End-to-end learning for lane keeping of self-driving cars. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1856–1860. IEEE, 2017.

[7] X. Cui and H. Shi. An overview of pathfinding in navigation mesh. *IJCSNS*, 12(12):48–51, 2012.

[8] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek. A brief introduction to opencv. In *2012 proceedings of the 35th international convention MIPRO*, pp. 1725–1730. IEEE, 2012.

[9] S. Devlin, A. Anspoka, N. Sephton, P. I. Cowling, and J. Rollason. Combining gameplay data with monte carlo tree search to emulate human play. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[10] R. M. Geraci. *Apocalyptic AI: Visions of heaven in robotics, artificial intelligence, and virtual reality*. Oxford University Press, 2012.

[11] C. I. Guinn and R. J. Montoya. Natural language processing in virtual reality training environments. 1998.

[12] G. Hee Lee, F. Faundorfer, and M. Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2746–2753, 2013.

[13] A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, and D. Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

[14] L. Klastrup. Interaction forms, agents and tellable events in everquest. In *CGDC Conf.*, 2002.

[15] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. *ArXiv*, abs/1506.07285, 2015.

[16] S. Lauria, G. Bugmann, T. Kyriacou, and E. Klein. Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38:171–181, 2002.

[17] V. F. Leavers. *Shape detection in computer vision using the Hough transform*. Springer, 1992.

[18] C. Matuszek, E. Herbst, L. S. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *ISER*, 2012.

[19] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical image analysis*, 1(2):91–108, 1996.

[20] H. Meng, N. Pears, and C. Bailey. A human action recognition system for embedded computer vision application. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6. IEEE, 2007.

[21] J. D. Miles and R. Tashakkori. Improving the believability of non-player characters in simulations. In *Proceedings of the 2nd Conference on Artificiel General Intelligence (2009)*. Atlantis Press, 2009.

[22] A. Nareyek. Ai in computer games. *Queue*, 1(10):58, 2004.

[23] D. Perzanowski, A. C. Schultz, and W. Adams. Integrating natural language and gesture in a robotics domain. *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell*, pp. 247–252, 1998.

[24] T. Pun, G. Gerig, and O. Ratib. Image analysis and computer vision in medicine. *Computerized Medical Imaging and Graphics*, 18(2):85–96, 1994.

[25] K. Sage and S. Young. Security applications of computer vision. *IEEE aerospace and electronic systems magazine*, 14(4):19–29, 1999.

[26] D. G. Shapiro, J. McCoy, A. Grow, B. Samuel, A. Stern, R. Swanson, M. Treanor, and M. Mateas. Creating playable social experiences through whole-body interaction with virtual characters. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[27] G. N. Yannakakis. Game ai revisited. In *Proceedings of the 9th conference on Computing Frontiers*, pp. 285–292. ACM, 2012.