Enhancing Semi-Autonomous Navigation via Visualization of Human Path Prediction

ISAACS: Gang Yao, Hsin-Pei Lee, Kewei Sui, Xi Chen

Abstract: The industrial trend for autonomous robots is to develop ever more sophisticated self-navigation algorithms and systems as an endeavor to remove human control ultimately. However, most of the self-navigation systems today are not entirely reliable, meaning the possibility of collision still stands despite the effort of object avoidance algorithms. In particular, situations, where human and autonomous robots are present in the same physical space, have become a common scene. We assume that under such circumstances, introducing human control for autonomous systems could increase the level of safety and that semi-autonomous robot navigation would better fit into the current social context. While human input can be as simple as a binary switch, for example, stop and go, we extend the scope by allowing the user to control a variable representing the level of autonomy directly. We then develop an AR system to visualize the human path prediction result as a visual cue for the robot operator to adjust robot navigation strategies accordingly. Due to the outbreak of Covid-19, we shift our program online. We will conduct the user study we designed and verify the assumption that semi-autonomous robots supervised by humans are equally efficient and even safer than fully autonomous ones in the future.

Augmented Reality | Visualization | Human Path Prediction | Semi-Autonomous | Human-computer Interaction

1. INTRODUCTION

In the present world, autonomous robots are becoming widely spread, accomplishing various intelligent and complex tasks. Their applications are very broad, ranging from casual tasks, including entertainment, filming, and delivery, to more serious scenarios such as in military and medical settings(1). As robotic technology is emerging, and the limits of autonomous robots' abilities are pushed constantly, many challenges such as accuracy and efficiency also appeared. Among those challenges, safety concern is the most critical one since failures have fatal consequences. The problem of robot's constant collision with humans in daily life is largely due to unpredictable human behavior and the lack of human involvement in the remote control. According to Steinfeld(2), strategies with the human in the loop provides adaptability to the unexpected environment. Thus we decided to follow and aim to further improve safety by introducing human factors into robot control through visualizations of future human paths. We refer the robot with human control as the semi-autonomous robot.

The unique point of the semi-autonomous robot is that it incorporates real-world conditions when the environment is unpredictable and dynamic. Assuming adding human control would bring down the potential risk of collision in case of autonomous algorithm goes wrong, we aim to provide visualizations to help the user (robot operator) manipulate robots' reactions when a potential collision is detected. In particular, we apply a path prediction algorithm for predicting walking humans' future path and a path planning algorithm for controlling the robot's behavior. At last, we visualize human's future path and robots' planned trajectories using AR technologies in Unity3D.

This paper first reviews related works in visualizing human path prediction and then illustrates our overall system description and implementation. Then, we explain how we set up an online simulation of our system in Unity3D due to the outbreak of Covid-19. Next, we evaluate and discuss the metrics used for user study and expectations. The last section concludes our result with a discussion of some limitations and how they might be addressed in our future works.

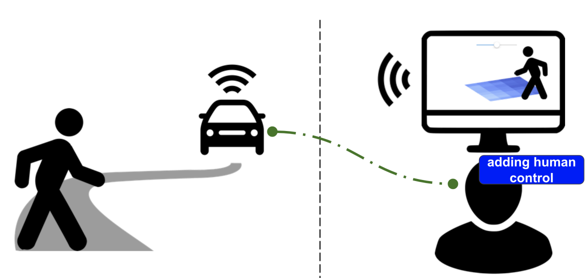

Fig. 1. General system structure

2. RELATED WORKS

To familiarize ourselves with ongoing methods of visualizing human path prediction results and robots' path planning algorithms, we first looked into current works in related fields.

Nishino et al.(3) design a human model for predicting future human path. They conclude that their robot navigation method based on the characteristic of pedestrian flow could effectively improve the efficiency of the robot. However, from their experiment data, we find out that the minimal distance between the robot and human sometimes goes down to 20 cm, even though they trying hard to keep the average minimal distance to be 50 cm. This situation is inevitable because the modeled human behavior cannot always match the actual human motion. Fernando et al.(4) use adapted Long Short-Term Memory (LSTM) neural networks to predict the human trajectory and detect human anomaly behavior. In their research, there are about 15% abnormal events that their algorithm cannot detect. Fisac et al.(5) indicate that even a well-informed human model cannot always make a correct prediction of the human's unpredictable behavior. Thus, they design a human motion prediction algorithm with model confidence. This model confidence variable depicts how confident the robot thinks the real human motion will match the predicted motion. Algorithms cannot handle all complicated situations in real life. Thus, we aim to design a user interface to include human control for changing this model confidence variable and enhancing robot safety. We utilize the human path prediction algorithm of Fisac et al. and the Fast and Safe Tracking (FaSTrack) framework designed by Fridovich-Keil et al. (6) for the robot's path planning.

People in the industry are trying to visualize virtual information into the real world to enhance safety. For example, the new F015 model by Mercedes-Benz(7) uses the laser projection system to project important information onto the road like a virtual zebra crossing or a stop sign to alert pedestrians. Tesla(8) visualizes the possible areas of blind spots on the screen to alert the driver while lane changing or steering. Visualization of virtual elements has also been implemented in AR. Yuta et al.(9) utilize a grid system to construct the world coordinate in virtual space. They combine object tracking, trajectory prediction, spatial calibration, delay compensation, and rendering techniques to visualize motion prediction in AR. They stated that by tracking real-world objects and estimating their trajectories, their system could help the user to predict the path of an oncoming object, for instance, an oncoming car. These examples give us an idea that we can incorporate the visualization of human path prediction into our system, and let robot operator change robot action to avoid possible collision based on this key visual effect.

Then, we look into related 3D trajectory visualization works. An example would be to visualize human motion through motion vectors around the human body parts(10). This is achieved by visualizing motion vectors in the frame with color gradients and arrows, which represent the moving direction of a specific part. By incorporating probability intensity information and mapping the 2D data into a 3D plane, we can achieve something similar to the prediction strokes as Alahi et al. did (11), where the predicted distribution strokes of their future trajectories are shown in the heat-map.

For designing an effective user interface, we get insights from Steinfeld's paper(2). He introduces some important thoughts on interface lessons that we need to ensure at minimum. He classifies those points into seven categories: safety, remote awareness, control, command inputs, status, and state. To assess user experience, Hietanen et al. (12) provide a metric for the interactive AR user interface with respect to safety, intuitive, autonomy, and competence. We utilize this metric to design our user study.

3. METHODS & ALGORITHMS

In this section, we address the potential problems with autonomous navigation and illustrate our proposed approach.

An alarming fact about autonomous navigation is that when the autonomous robot co-exists with the human, the collision detection algorithm is sometimes flawed, indicating a problem related to safety. However, we cannot blame it all on the algorithm itself because human behavior is so unpredictable that sometimes it can surpass the algorithm's predictive power, which presents quite an issue.

Another problem with autonomous navigation is that we have to decide how involved we want to be while driving. Some people maybe want the robot to operate entirely by itself; some might want to retain some control of the robot. However, if we drive an autonomous vehicle like Tesla, we can only switch between autonomous driving (Autopilot) and self-driving, one option or the other, all machine or all human.(13) There is nothing between these two. Thus, creating an autonomous robot that behaves differently, depending on the user's preferences, is essential. In other words, if we prefer to drive more aggressively, we should be able to tell the robot to drive more aggressively, too. Thus, the level of "aggressiveness" could be quantitatively specified to guide robot navigation.

Our vision is to blur the boundary between autonomous and manual control at the same time, increase navigation safety. We are taking one step towards this vision by changing the way the human interacts with the robot. Since sometimes we won't be able to prevent a collision by hitting an emergency stop button when we think the car is making a mistake, We hope to be able to let the robot operators adjust a confidence parameter, which allows them to operate on a continuum of human involvement instead of intervening by pushing the emergency stop button. In addition, to be able to effectively and intuitively monitor the navigation, we need to let humans share the same physical space with robots. Therefore, we project visual elements into the scene shot by robots as a reference for the user to adjust robots' navigation. Specifically, we visualize the human path prediction and robot trajectory in AR.

3.1 Confidence-Based Human Prediction

This section is a summary of the work from David et al (6), which we use as a basic algorithm to implement our human path prediction logic. We describe the state of the human x_H as the 2D coordinate of its physical location. Assuming that the world space is partitioned into N by N grid spaces, then the human can only have N^2 possible states, $x_H \in \mathbb{R}^{N*N}$. At each state, the human has K different possible actions $u_H \in^K$. For the 2D grid world, K equals 9, represents an action set that consists of 8 possible transitions and 1 to stay still. Given a human state-action pair, the human model dynamic can be defined as:

$$\dot{x}_H = f_H(x_H, u_H) \tag{1}$$

where f_H represents the transition function for state-action pairs.

The algorithm uses a reward function to describe how much reward a human gain for any given state-action pairs. The objective of a perfectly rational human always tries to maximize his or her accumulative reward. Assuming we know the human goal point, the reward function is defined as the negative distance traveled plus the Euclidean distance between the goal position and the future position after an action is taken given a known state:

$$Q_H(x_H, u_H; g) = -\|u_H\|_2 - \|x_H + u_H - g\|_2 \tag{2}$$

where g stands for the goal point.

Then given the current state, the human is more likely to choose actions with the highest expected utility. The reference paper models the possibility of each action from a given state as:

$$P(u_H^t | x_H^t; \beta, g) = \frac{e^{\beta Q_H(x_h^t, u_H^t; g)}}{\sum_u e^{\beta Q_H(x_H^t, u; g)}} \tag{3}$$

The coefficient β is the model confidence coefficient, and it determines the degree to which the operation trusts the prediction result. In other words, β represents the accuracy with which the robot's model of the human can explain human motion. When the model is irrational, $\beta = 0$, the prediction of human motion completely ignores the real human action. In this situation, the prediction becomes too conservative in predicting human motion effectively. While $\beta \to \infty$, the human model corresponds to a "perfectly rational" human, which means the prediction completely depends on previous real human behavior. In this way, the human model becomes too radical to include any random human behavior into the prediction of human motion.

It is never reasonable to set β as a constant number. Thus, Fisac et al. (5) produce a model that the confidence level, β, will change based on real human motion in time. In this model, at every time step t, the robot obtains a new measurement[1] of the human's action, u_H^t. This measurement can be used as evidence to update the robot's belief $b^t(\cdot)$ about β overtime via a Bayesian update:

$$b^{t+1}(\beta) = \frac{P(u_H^t \mid x_H^t; \beta, \theta) b^t(\beta)}{\sum_{\hat{\beta}} P(u_H^t \mid x_H^t; \hat{\beta}, \theta) b^t(\hat{\beta})} \quad , \tag{4}$$

θ denotes the set of parameters in the human's utility model. By using this algorithm, we can produce an adaptive model that has better performance.

However, this algorithm is reasonable but not practical enough. To shorten computational time and update the robot's belief in time, Fisac et al. narrow the original continuous hypothesis space $\beta \in [0, \infty)$ to $\beta \in [0.05, 10]$ based on experiments (5). Thus, the adaptive model may perform poorly in some circumstances because of the narrowed hypothesis space of β.

Therefore, we introduce human control into our system. The robot operator can choose different models of prediction manually for handling various circumstances and further increasing the correctness of prediction. Or we can fully leverage the duty to the robot operators and let them manipulate the β directly. More details about human control are intro-

[1] In practice, the robot measures the evolution of the human state and computes the associated action by inverting the motion model.

duced later in the User Interface section and simulation section.

3.2 Autonomous Robot Navigation

FaSTrack is a tool developed at Berkeley Robotics Lab that is used in conjunction with any model-based motion planner to provide safety guarantees while planning and executing trajectories in real-time. FaSTrack allows the planner to find trajectories using simple and easily computed planning models; this allows for the planner to operate in real-time. (14) This is well suited in our use case. The robot is continuously running the FaSTrack algorithm, which generates a path that takes into account the human position and any obstacles in the scene. In the grid world, if the prediction possibility at one grid is above the threshold value, the robot will treat this grid as an obstacle and plan to avoid it. Instead of controlling the robot directly, we let the user to indirectly supervise its navigation behavior by tweaking parameters, which is less cognitive intensive while preserving the level of safety.

4. SYSTEM DESCRIPTION

Our current version of the physical system consists of four modules, as shown in Figure2. We use the Optitrack motion capture (mocap) system to track the position of the human and the robot (mounted with a first-person camera). Their position data are streamed to a workstation running ROS via a local Ethernet connection. Then the human path prediction algorithm running on ROS generates prediction data continuously. These data combined with raw motion capture location data are then broadcasted to the monitor's computer running Unity via ROS bridge. The Unity computer integrates this data and creates the virtual elements at their correct position. We render the virtual elements as an overlay to the webcam viewport, generating an AR image. Finally, the robot operator can adjust robot navigation strategies through the user interface.

4.1 Motion Capture

We use OptiTrack(15) system to track the pose of the human and the robot. To ensure the best-tracking quality, we stitch markers on the cap (Figure3 left) and the robot (Figure3 right) and make sure these two objects stays rigid while tracking. The webcam is mounted onto the robot as the first person see-through view.[2]

[2]In figure 3, the station under the camera represents the robot itself

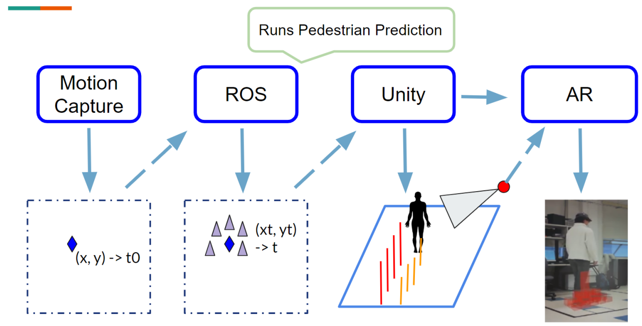

Fig. 2. System structure

Fig. 3. Adding markers on the cap and the base to track human and robot.

4.2 Human Prediction on ROS

The python scripts for human path prediction is built as a ROS node that processes the real-time human position as input and generates the possibility distribution of prediction as output. This ROS node then packs and broadcasts the prediction result along with the real-time motion capture data of human pose, camera pose to the client PC to generate AR visualizations there.

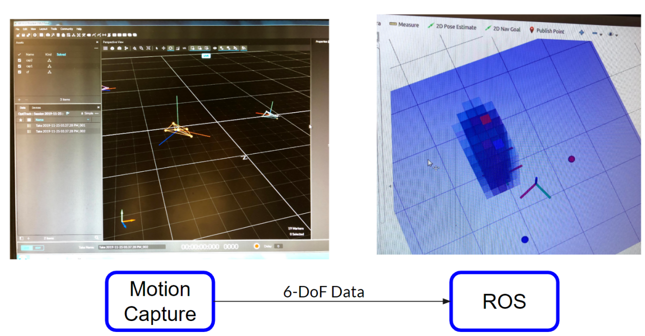

Fig. 4. OptiTrack and 2.5D visualization on RVIS

4.3 AR Visualization on Unity

The OptiTrack system tracks objects at a one-to-one scale as the real world. When using that data to create virtual objects in Unity, we make sure that the world coordinates in Unity scale the same as the coordinates in the motion capture system. Firstly, to seamlessly merge the real world into Unity, we create a giant virtual plane to display the real-time webcam image and scale the plane to take up the full camera view. (Figure 5) Then we assign this plane as a child object of the main camera object and the main cam-

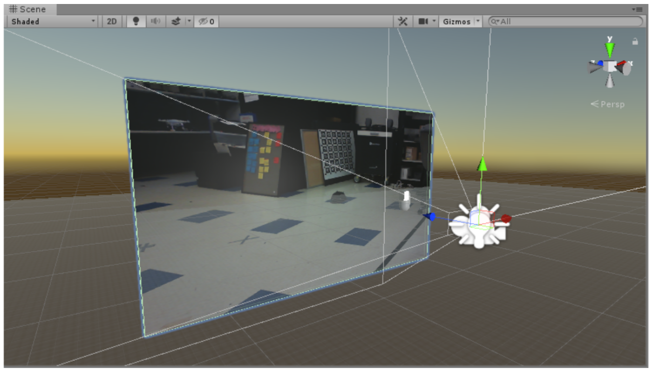

Fig. 5. Unity camera view

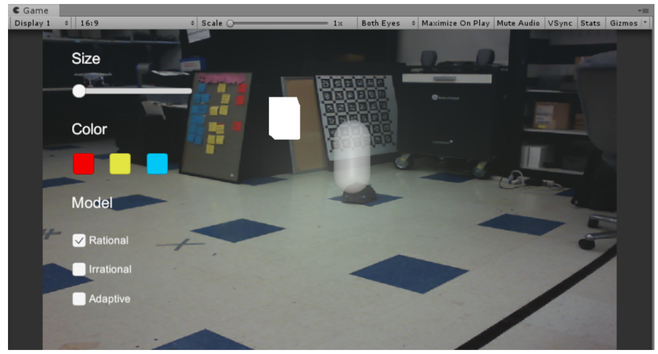

Fig. 6. AR Visualization of Path Prediction showing an example of how the final interface would appear.

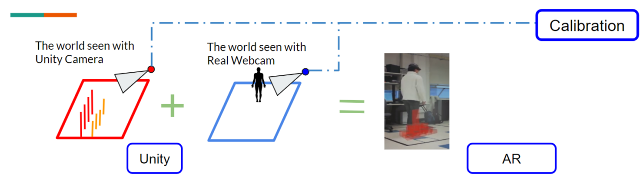

Fig. 7. Camera Calibration

era as a child of drone object in Unity hierarchy. Secondly, to align the virtual camera in Unity and the real-world webcam at the same pose, we assign the 6 domain of freedom motion capture data to the Unity object so that any motion could be synchronized to the same coordinate system. Other virtual objects such as the ground plane and visualized probability pillars can also be mapped to the correct position in the Unity world. Lastly, to correctly merge the picture of the virtual Unity camera and the picture of the webcam, we employ the standard camera calibration technique. We compute both the camera matrix (also called the intrinsic matrix) and the distortion coefficients. The camera matrix represents the translation and scaling of the image introduced by the camera. It is defined as:

$$camera\ matrix = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(5)}$$

where (f_x, f_y) are the focal length parameters, and (c_x, c_y) is the position of center point on image. (s) is the axis skew which is 0 in our case. Assuming that the webcam image is not severely distorted, we assign the intrinsic parameters of the webcam to Unity camera and make the field of view of the two cameras the same. Once the two cameras are calibrated (Figure7) and two coordinate frames align, the virtual objects are promised to be rendered to the correct position in the real camera image.

4.4 User Interface

The core insight of semi-autonomous robots is to introduce human control as an assisting element for autonomous navigation. Thus we design an interface for the user to change the visualization parameters providing ways to reflect such human control. We show the human path prediction distribution as extruded colored pillars around the human as in Figure8. The height of each pillar represents the level of possibility that the human will be at that position. The model picker changes the intrinsic β of the prediction model. When set to irrational, β equals infinity, meaning that the prediction is completely irrelevant from the human positions. (Figure8) When set to rational, β equals zero, meaning that the prediction is completely dependent on the position of the human. (Figure9) When set to adaptive, β is set to vary in a range [0.5, 10] depending on the real human action. (Figure10) These three settings greatly change the prediction result. The user can use the most optimal setups in their opinion in response to any specific situation. Then, we change the model picker to be a slider, which allows the operator to adjust β in a range freely. We will discuss the β changing slider more in the simulation section.

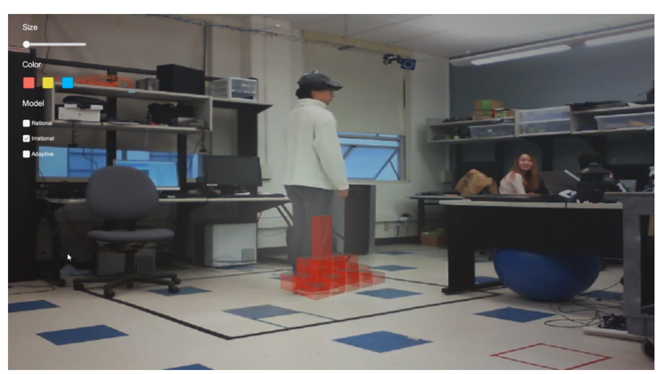

Fig. 8. Irrational model with $\beta \rightarrow \infty$

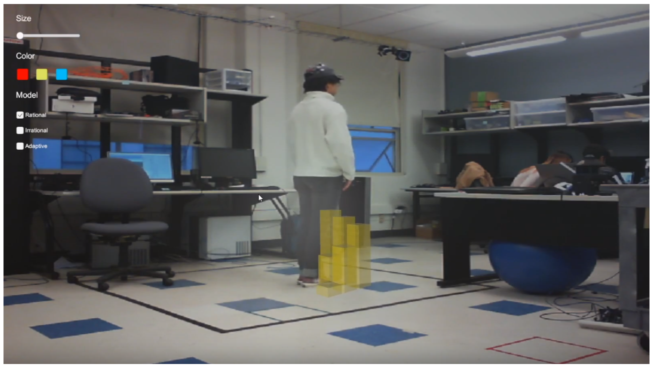

Fig. 9. Rational model with $\beta = 0$

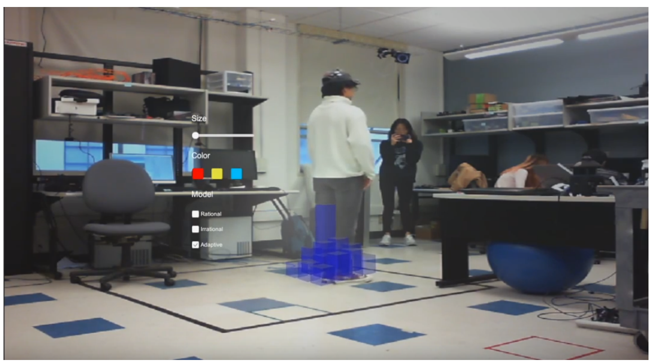

Fig. 10. Adaptive model with $\beta \in [0.05, 10]$

4.5 EXPERIMENT SETUPS

The human path prediction algorithm output depends on the specified start and endpoint. To properly test our system, we set up two experiment routes. One directly walks from the start point to the endpoint and the other takes a right angle. Two routes are shown in Figure11.

Under circumstances where the prediction looks like very reliable, as we designed for route 1, we expect an efficient robot operator to set the confidence level to a large number to prevent the robot from taking detours. Conversely, when the operator judges that the human is view is like to deviate from path prediction, we expect the operator to switch to a small number to eliminate any possible collision.

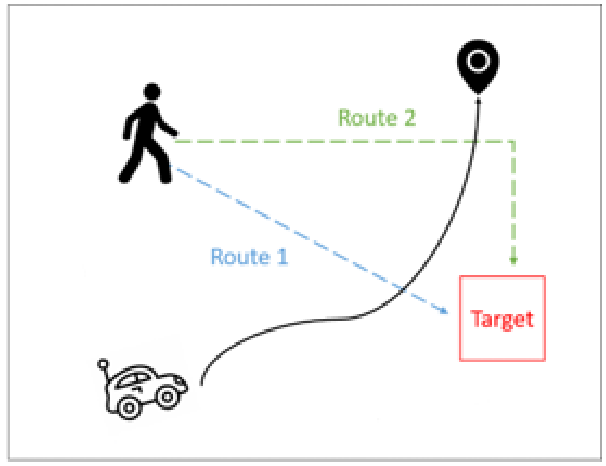

Fig. 11. Experiment setup showing two human routes that intersect with car's path.

5. SIMULATION

Due to the world pandemic Covid-19, our access to our experimental setup was largely limited. Thus we switched to online simulation. For achieving consistency with our previous setup, we use Unity and apply A* algorithm on robots' path planning and David et al.'s algorithm on human path prediction in the simulation system.

The colored heat map around the walking human represents human path predictions. When a traverse possibility at a given grid point is less than a threshold, this block is ignored. When this possibility is higher than that threshold, this grid point is inserted into the obstacle list, then visualized as a square tile whose color is interpolated from its possibility.

At a general view, when the distributed colors are light and more blocks are present, the directions where humans might go are spread out. When colors become more saturated and fewer blocks are present, the human is very dedicated about which direction to go next.

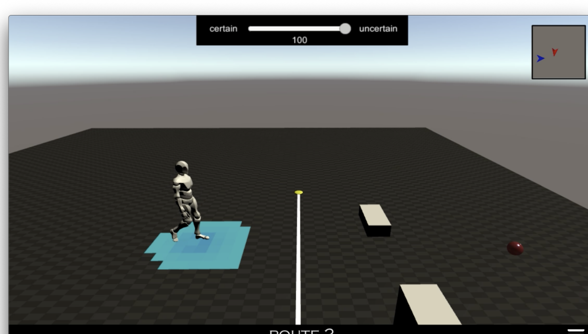

Fig. 12. Heap map with low confidence

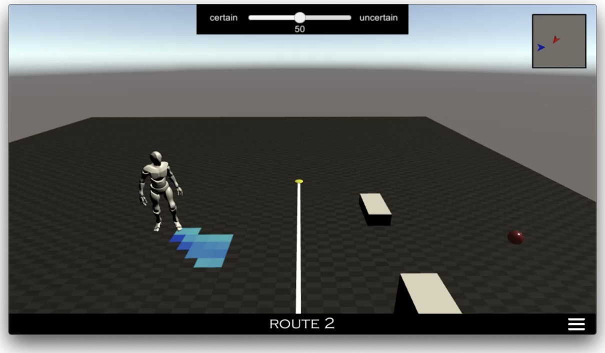

Fig. 13. Heap map with average confidence

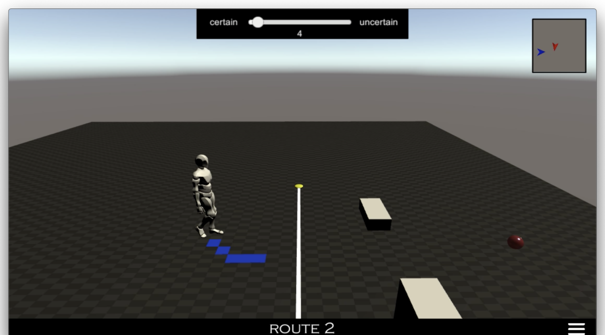

Fig. 14. Heat map with high confidence

Our simulation adopts two modes: the stop mode and the confidence mode. Stop modes simulates robots such as kiwi bots, which monitors can only issue "stop" and "go" orders on a button in response to emergency situations. While confidence mode blurs the boundary, by using a slider bar to change a simple one-dimensional variable called the "confidence level". We tweaked around the mapping from the confidence level, which is directly set by the user, to β in the prediction algorithm. For the current setup, $\beta = 0.01 * 1.096^{1-confidence}, confidence \in [0,100]$. When a low confidence level is given, the prediction of the human path is more conservative, which means the monitor does not trust the human path prediction result very much and vice versa. It leads to a more conservative routing strategy for the robot and makes it safer for the human to pass without collisions.

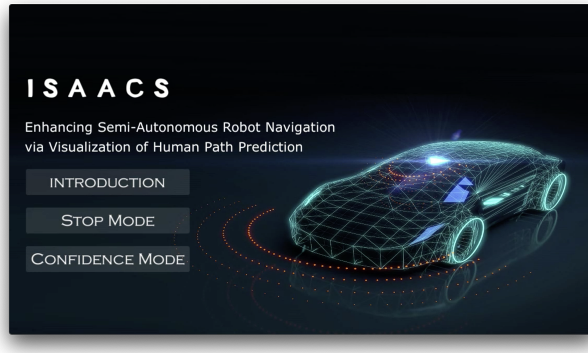

Fig. 15. Intro page of the simulation system

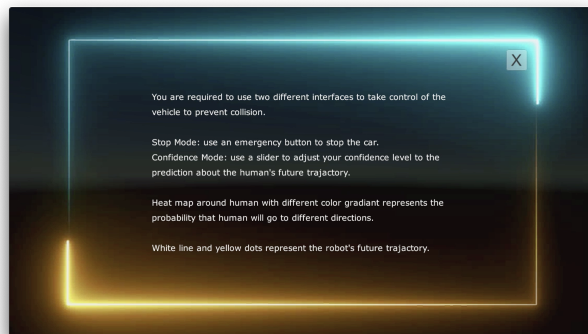

Fig. 16. Introduction the simulation system

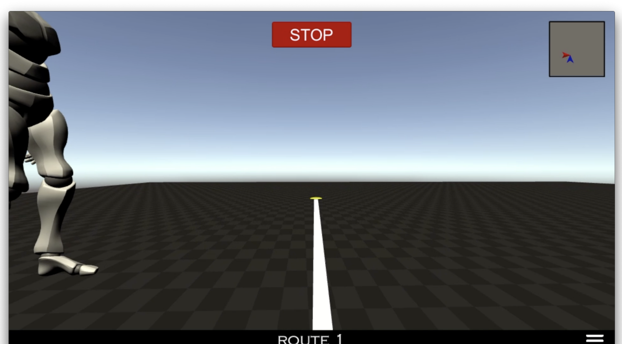

Fig. 17. Taking route 1 by using stop mode

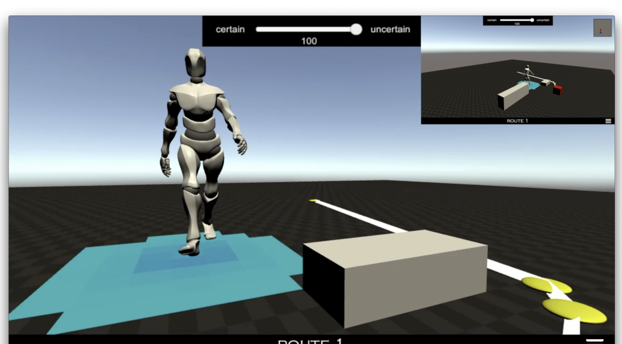

Fig. 18. Taking route 1 by using confidence mode with low confidence

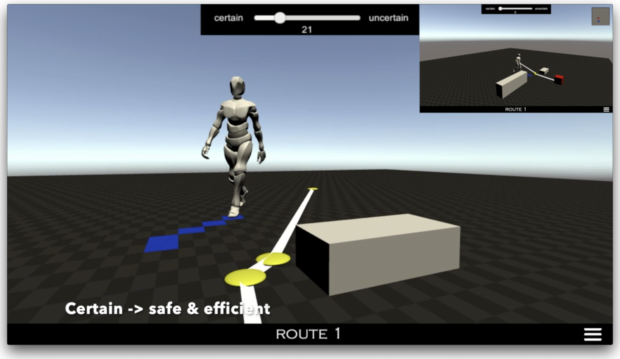

Fig. 19. Taking route 1 by using confidence mode with high confidence

6. RESULTS & EVALUATION

6.1 Pending User Study

Due to the outbreak of Covid-19, we were unable to invoke people in the user study at this time. However, we have completed the design of tasks and evaluation metrics for future user studies. When introducing our simulation system to users, we will ask them to drive the robot to the destination point and make efforts to avoid collision with the walking human in the scene by adjusting the confidence level slider bar. After they complete this task, we would refer our result as semi-autonomous control, and compare it to autonomous control. Specifically, we will apply two relatively quantitative evaluation metrics, one is the minimal distance between the robot and human for safety measure, and the other one includes robot's movement distance and total execution time for efficiency measure. Also, to measure user satisfaction level, we developed a questionnaire suggested by table 1. The goal of this questionnaire is to evaluate the physical and mental stress of users during completing the task we assigned. The topics covered safety, intuition, autonomy, competence, and other open-ended questions. Users will be asked to score each question using the scale from 1 (totally disagree) to 5 (totally agree). Following this entire user study plan, we aim to reflect our system's efficiency and user-friendliness and conclude whether a semi-autonomous system would make robots safer compared to autonomous systems.

7. DISCUSSION

Our visualization design has several limitations. First, the FaSTrack algorithm can only predict routes when the robot knows the destination of the human. Therefore, in the real world scenario, when the target of the human remains uncertain, a new predicting algorithm must be applied. Second, the current prototype uses the motion capture system to track the position and orientation of the robot and human. In order to meet this criterion, our system will be limit to indoor use only. However, there exist some object tracking models that use deep learning-based computer vision techniques to detect humans. By applying these techniques, we might be able to use the web-cam of the robots to track humans easily. Third, although our system is currently built in AR on the computer, it can be expanded to AR on HoloLens or virtual reality environments. By supporting different devices, more extensive needs can be satisfied. Forth, due to the delay between the mocap system and, ROS, and Unity, humans can only walk at a slow pace to wait for data transmitting and processing. By integrating our Unity interface and ROS together on a single computer might solve this problem.

8. CONCLUSION & FUTURE WORK

Predicting the next events to gain insights about the future is an emerging event analytic task. Studies suggest that people are more confident in making decisions when alternative predictions are displayed(16). We have proposed a novel AR system for visualizing the confidence level of path prediction from the human path prediction algorithm and the FaSTrack algorithm. This real-time system aims to handle unpredictable situations to bring down potential risks and enhance safety. However, the assumption that semi-autonomous robots supervised by humans are equally efficient and even safer than fully autonomous ones still needs case-studies to be verified. In future work, we will conduct experiments to prove the effectiveness of our design and conduct user studies as planned. After that, we will incorporate new functions and redefine the user interface based on the feedback.

ACKNOWLEDGEMENTS

We thank all VR lab team for their valuable assistance and feedback, in particular, professor Allen Yang, PHD students Joseph Menke, David McPherson, and David Fridovich-Keil.

Table 1. The questionnaire used in the user study.

Categories	Questions
Safety	Q1: The job has a low risk of accident.
	Q2: The system decreases the chance of collision.
Intuitive	Q3: A lot of time was required to learn the user interface.
	Q4: The job requires me to analyze a lot of information.
Autonomy	Q5: During task, I felt a sense of choice and freedom in the things I undertake.
	Q6: Robot considers how I would like to do things.
Competence	Q7: I feel disappointed with my performance in my task.
	Q8: I feel confident in my ability to complete my task safely.
Other	Q9: How does this experience compare to your expectations? Was there anything surprising or unexpected?
	Q10: How do you think this is going to help you?
	Q11: How could we present the information in a more meaningful way?
	Q12: Is there anything you would change/add/remove to make this better for you?

9. REFERENCES

1. George A. Bekey. Autonomous robots from biological inspiration to implementation and control. 2017.
2. Aaron Steinfeld. Interface lessons for fully and semi-autonomous mobile robots. In *Proceeding of the 2004 IEEE International Conference on Robotic and Automation*, 2004.
3. N. Nishino, R. Tsugita, D. Chugo, S. Yokota, S. Muramatsu, and H. Hashimoto. Robot navigation according to the characteristics of pedestrian flow. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 5947–5952, 2016.
4. Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural networks*, 108:466–478, 2018.
5. Jaime F. Fisac, Andrea Bajcsy, Sylvia L. Herbert, David Fridovich-Keil, Steven Wang, Claire J. Tomlin, and Anca D. Dragan. Probabilistically safe robot planning with confidence-based human predictions, 2018.
6. David Fridovich-Keil, Sylvia L. Herbert, Jaime F. Fisac, Sampada Deglurkar, and Claire J. Tomlin. Planning, fast and slow: A framework for adaptive real-time safe trajectory planning, 2017.
7. The mercedes-benz f-015 luxury in motion. https://www.mercedes-benz.com/en/innovation/autonomous/research-vehicle-f-015-luxury-in-motion/. Accessed: 2020-04-01.
8. Madhuri Phute Ingle Shantanu. Tesla autopilot: semi autonomous driving, an uptick for future autonomy. *International Research Journal of Engineering and Technology*, 3, 2016.
9. Yuta Itoh, Jason Orlosky, Kiyoshi Kiyokawa, and Gudrun Klinker. Laplacian vision: Augmenting motion prediction via optical see-through head-mounted displays. In *Proceedings of the 7th Augmented Human International Conference 2016*, page 16. ACM, 2016.
10. keijiro. Kinovision. In *https://github.com/keijiro/KinoVision*, September 2017.
11. Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
12. Minna Lanz Jyrki Latokartano Joni-Kristian Kämäräinen Antti Hietanen, Roel Pieters. Ar-based interaction for human-robot collaborative manufacturing. In *Robotics and Computer-Integrated Manufacturing 63*, 2020.
13. Tesla: Autopilot and full self-driving capability. https://www.tesla.com/support/autopilot. Accessed: 2020-05-03.
14. David Fridovich-Keil, Andrea Bajcsy, Jaime F Fisac, Sylvia L Herbert, Steven Wang, Anca D Dragan, and Claire J Tomlin. Confidence-aware motion prediction for real-time collision avoidance1. *The International Journal of Robotics Research*, 39(2-3): 250–265, 2020.
15. OptiTrack: Motion Capture System.
16. Renate Häuslschmid, Max von Buelow, Bastian Pfleging, and Andreas Butz. Supportingtrust in autonomous driving. In *Proceedings of the 22nd international conference on intelligent user interfaces*, pages 319–329, 2017.