Busy Beeway: A Game for Testing Human-Automation Collaboration for Navigation

Torin Adamson
Department of Computer Science
University of New Mexico
toriadam@cs.unm.edu

Hao-Tien (Lewis) Chiang
Department of Computer Science
University of New Mexico
lewispro@unm.edu

Meeko Oishi
Department of Electrical and Computer Engineering
University of New Mexico
oishi@unm.edu

Lydia Tapia
Department of Computer Science
University of New Mexico
tapia@cs.unm.edu

ABSTRACT

This study presents *Busy Beeway*, a mobile game platform to investigate human-automation collaboration in dynamic environments. In *Busy Beeway*, users collaborate with automation to evade stochastically moving obstacles and reach a series of goals, in game levels of increasing difficulty. We are motivated by the need for reliable navigation aids in stochastic, dynamic environments, which are highly relevant for self-driving vehicles, UAVs, underwater and surface vehicles, and other applications. The proposed mobile game platform is agnostic to the particular algorithm underlying the autonomous system, can be used to evaluate both fully autonomous as well as human-in-the-loop systems, and is easily deployable, for large, remote user studies. This last element is key for rigorous study of human factors in navigation aids. Through a small 32-user study, we evaluate preliminary findings regarding the relative efficacy of collaborative and fully autonomous navigation, the relationship between success rate and users' learned trust in the automation (gathered via pre- and post-experiment surveys), and tolerance to error (for decisions made by the automation and by the user). This study validates the feasibility of *Busy Beeway* as a platform for human subject studies on human-automation collaboration, and suggests directions for future research in human-aided planning in difficult environments.

CCS CONCEPTS

•**Human-centered computing** → *Empirical studies in collaborative and social computing;*

KEYWORDS

Motion Planning, human-automation interaction, collaborative navigation

MiG '17, Barcelona, Spain

DOI: 10.1145/3136457.3136471

ACM Reference format:
Torin Adamson, Hao-Tien (Lewis) Chiang, Meeko Oishi, and Lydia Tapia. 2017. Busy Beeway: A Game for Testing Human-Automation Collaboration for Navigation. In *Proceedings of MiG '17, Barcelona, Spain, November 8–10, 2017,* 6 pages.
DOI: 10.1145/3136457.3136471

1 INTRODUCTION

Navigation in environments with stochastic dynamics is a hard problem that plagues self-driving cars and other transportation applications [Glover and Lygeros 2004; LaValle and Sharma 1995]. Indeed, fully autonomous solutions are known to be NP-Hard and in PSPACE [Canny 1988; Canny and Reif 1987]. Navigation aids that enable collaboration between the human and the automated system have the potential to improve safety and reduce workload [Flemisch et al. 2008], and are a key element of many nascent solutions, such as the Google self-driving car. However, important questions remain in the development of reliable autonomy and the effectiveness of human interfaces in such systems [de Winter and Dodou 2011]. Interfaces to autonomous systems should ideally promote situational awareness and trust through transparency [Hoff and Bashir 2015], yet not overwhelm the human with unnecessary or untimely information. Effective ways to convey appropriate actions under uncertainty are also an open area of human factors research. For example, interfaces that promote engagement at appropriate times may prevent over- or under-reliance on the automation [Parasuraman and Riley 1997], and therefore improve overall performance of the human-automation system. The need for reliable tools and methods to facilitate effective human-automation collaboration, particularly in challenging environments, such as those that are stochastic and dynamic, is paramount.

We propose a mobile game platform, *Busy Beeway*, to enable large-scale studies of collaborative navigation systems in stochastic, dynamic environments. In *Busy Beeway*, users collaborate with the automation over control of a bee character (named Belinda) to evade stochastically moving obstacles and reach a series of goals, in game levels of increasing difficulty. *Busy Beeway* is agnostic to the particular algorithm underlying the autonomous system and can be used to evaluate both fully autonomous as well as human-in-the-loop systems. We envision our platform as a precursor to human subject testing in expensive, immersive, scenario-specific environments, such as those in [Johns et al. 2016; Xu and Dudek 2016]. Our

platform enables exploratory inquiries in a cheap, fast, and easily deployable manner, and hence is ideal for research in the rapidly developing area of autonomy for human-on-the-loop systems. Our platform exploits a simple touch screen interface, however, we believe that this is sufficiently generalizable; Google self-driving cars and the DJI Phantom 4 also use touch screen interfaces.

We present a small, 32-user study, which consists of game play as well as pre- and post-surveys that focus on demographics, attitude towards automation in general, and familiarity with video games. Preliminary results from this study indicate that *Busy Beeway* is a feasible platform for study of collaborative navigation aids; users actively collaborate with automation in *Busy Beeway*, the game's scoring system keeps users engaged, and *Busy Beeway* has sufficient complexity that interesting human-automation interaction phenomena can be observed. Our pilot study revealed trends regarding relative efficacy of collaborative and fully autonomous navigation, the relationship between success rate and users' learned trust in the automation, and users' tolerance to error (for decisions made by the automation and by the user). *Busy Beeway* reveals that users are more likely to take control in the presence of large numbers of obstacles, but this intervention often does not necessarily improve navigation success. Based on the promising results of this preliminary study, we believe that *Busy Beeway* is well suited to serve as a platform for future studies in collaborative navigation.

2 RELATED WORK

Very little work has been done on video games as collaborative navigation aids. Most platforms for collaborative navigation occur in automotive applications and exploit highly customized, problem specific simulator platforms [de Waard et al. 1999; Johns et al. 2016; Xu and Dudek 2016]. Work in human-robot teleoperation has shown that a simple touch screen interface outperforms an interface that uses eye-tracking and a computer mouse [Dunser et al. 2015]; other work shows that shared control can be used to compensate for input delay using a game controller [Storms et al. 2017]. In contrast, we propose a touchscreen-based platform that is amenable to repeatable, large-scale user studies, and is capable of generating scenarios rich in relevant human-automation interaction issues. Video games have been successfully used to convey computationally challenging tasks such as protein folding [Cooper et al. 2010], image recognition [von Ahn and Dabbish 2004], and weather prediction [Eveleigh et al. 2013]. Such applications seek to harness human intuition, outperforming state of the art algorithms in many cases. However, for these games to be effective, they must motivate the user, e.g., through competitive features [Eveleigh et al. 2014], to ensure high quality data, and encourage repeated play.

While *Busy Beeway* can be used with any planning or control technique, we implement solutions based on algorithms for motion planning in stochastic, dynamic environments. This difficult problem has received considerable attention recently, as it is relevant to many real-world applications, yet identifying real-time collision-free solutions is practically impossible due to factors such as large numbers of obstacles, obstacle motion uncertainty, and dynamics constraints of the robot. Real-time planning algorithms typically rely on heuristics and often cannot provide complete collision-free solutions. Common heuristics include Artificial Potential Fields (APF) [Khatib 1986; Malone et al. 2017], Social Forces [Helbing and Molnár 1995] and Velocity Obstacles (VO) [Fiorini and Shiller 1998]. These methods react to nearby obstacles and compute motion control actions at every time step. Other approaches use state- [Bruce and Veloso 2002; Ferguson et al. 2006] or state-time sampling [Benenson et al. 2006; Chiang et al. 2017, 2015], and typically consider obstacles in a wider region to plan a path over known horizon.

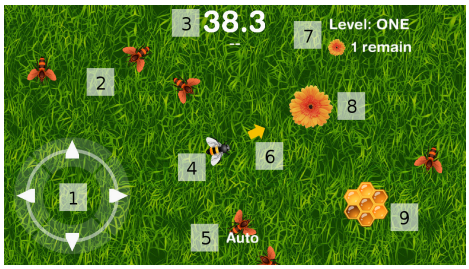


Figure 1: *Busy Beeway* user interface. Elements of the display include: 1) on screen joystick for manual control (users can also touch the bee directly), 2) stochastically moving obstacles (red wasps), 3) current score as time elapsed (high score below, if available), 4) Belinda, 5) indicator for whether Belinda is under manual or automation control, 6) guidance arrow pointing to nearest subgoal or final goal, 7) number of remaining goals, 8) subgoals (flowers), 9) final goal (hive).

3 BUSY BEEWAY

Design objectives for *Busy Beeway* include the ability to motivate users toward future participation, sufficient difficulty that requires collaboration with automation, and compatibility with benchmarks that allow comparison to existing state-of-the-art algorithms.

3.1 Game Mechanics

Figure 1 shows a screenshot of *Busy Beeway*. The user's objective is to guide Belinda to all of the subgoals, i.e., collect honey from flowers in any order (8 in Figure 1), before heading towards the final goal, i.e., returning to the hive (9 in Figure 1). To help users locate the subgoals, a guidance arrow (6 in Figure 1) points toward the nearest subgoal or the the final goal, the hive, if all subgoals have been reached. A training level is presented that allows the user to become familiar with the controls using a series of example subgoals. The training level contains no dynamic obstacles. After training, users must navigate without colliding with moving obstacles, wasps (2 in Figure 1). If Belinda reaches the final goal, it enters the next level, which has more and/or faster stochastically moving obstacles. Users are given up to five attempts at each level before *Busy Beeway* automatically proceeds to the next level. The score is defined as the lowest amount of time to complete the user's objective (3 in Figure 1), and the users are encouraged to obtain the lowest score possible.

3.2 Automated System

In principle, any efficient motion planning algorithm can be used that is capable of returning a motion control action within each game play frame. We used the lower-end video game industry standard, 30 Hz, that is acceptable for action game play [Claypool and Claypool 2007].

In *Busy Beeway* the user shares control of Belinda with the automation in a variety of ways. We presume a collaboration scheme in which in the absence of user input, e.g., when the user is not touching the screen, the automation is engaged and the motion planner dictates the bee's action. Otherwise, when the user is touching the screen, the user's control action is used. We chose this collaboration scheme for its similarity to SAE level 2, state-of-the-art for self-driving cars (e.g., the driver supervises and takes control when needed) [de Winter and Dodou 2011]. In the Human-Automation Collaboration Taxonomy [Bruni et al. 2007], our scheme has a level 1 moderator, a level 1 or 5 generator, a level 1 decider, a black functional transparency level, raw information transparency level, and command level interactivity.

3.3 Data Collection and Study Protocol

Busy Beeway records, at 30Hz, raw screen input, control input to Belinda, the location of Belinda and all dynamic obstacles, and all other inputs from the user (e.g., screen touches that do not affect Belinda because they are outside of the joystick area). This data enables full reproducibility of the user's experience and choices, capturing both user actions and the stochastic nature of the game.

The human subject study protocol, recruitment, and compensation was approved by the Institutional Review Board of University of New Mexico (IRB ref. 01517). Users were recruited via e-mails, invitation cards and flyers posted on the university campus. Interested users completed the online pre-survey to sign up for the study. Before each session, the users completed a consent form. Each play session lasted 10 to 15 minutes. A post-survey was administered at the end of each participation session. Users were compensated $10 upon completion of the study.

3.4 Experimental Setup

We implemented an artificial potential field (APF) motion planner [Khatib 1986] as the automation system. Within *Busy Beeway*, the APF planner takes approximately 11 to 32 ms to compute a motion control action (the duration is dependent on the number of dynamic obstacles), which is 33% to 96% of the frame time. A fully automated solution using APF results in collision with moving obstacles about 44% of the time in the most difficult *Busy Beeway* level. Therefore, it is likely a user will observe a failure of the automation during the game play. This makes the task difficult enough that the user is incentivized to collaborate with the automation, yet still takes over when warranted.

The action generated by the APF planner follows the gradient of the weighted sum of the attractive potential and repulsive potentials. The attractive potential is proportional to the squared distance between Belinda and the goal. For each dynamic obstacle within range $d_{max} = 6$, a repulsive potential inversely proportional to the clearance between Belinda and the obstacle is constructed. The repulsive potentials are weighted 15 times stronger than the attractive potential.

Four levels were used in the study. First, users were trained on the control of Belinda and the use of automation. In this level, the user is encouraged to reach a series of subgoals without the threat of moving obstacles. The average amount of time users spent in the training level was 48.3±27.1s; a minimum of 10s is required in the training level. In the subsequent three levels, Belinda must reach five subgoals and one final goal under increasingly higher numbers of obstacles. Level 1 (Figure 2 (b)) had 100 obstacles, Level 2 (Figure 2 (c)) had 200 obstacles, and Level 3 (Figure 2 (d)) had 300 obstacles. Both Belinda and the obstacles had a collision radius of 0.3 units and are confined to a world of radius 50. When an obstacle reaches the edge of the world circle, it is teleported to the antipodal position, with unchanged velocity. At the beginning of game play, obstacles are placed randomly in the world with random headings and move linearly. During game play, obstacles stochastically re-sample their speed at 10Hz from the set $\{2, 4, 6, 8\}$unit/s with probability $\{.25, .35, .25, .15\}$. Belinda can move in any direction at a maximum speed of 4 unit/s. Note that the relative speed between the obstacles and robot favors the obstacles. Also, the number of obstacles is very high, especially in the higher levels. Both of these facts make the game challenging for an ordinary user to complete without the help of automation. Belinda starts at $(-20, 0)$ and must visit five subgoals, located at $(5, 7)$, $(11, 0)$, $(10, 5)$, $(15, 10)$ and $(13, -9)$, before heading to the final goal, located at $(20, 0)$.

Thirty two users completed the pre- and post-surveys and *Busy Beeway* game sessions. Users were between the ages of 18 and 49 with an average of 25 years. Half were male, and half were female.

4 RESULTS AND DISCUSSION

4.1 User Engagement

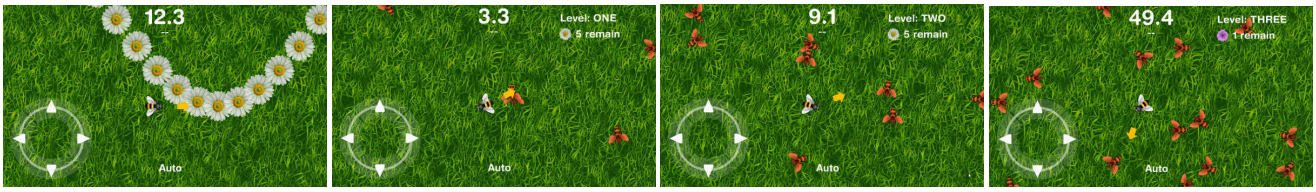
As a mobile game, *Busy Beeway* has a much lower user perceived risk than that associated with realistic driving vehicle simulators [de Winter and Dodou 2011; Flemisch et al. 2008; Zeeb et al. 2015]. This could potentially reduce incentive to collaborate with automation, however, we found that users demonstrated a general willingness to collaborate with the automation. Over half (59%) of the users utilized the automation at least 50% of the time during game play. In addition, 64% of users self-reported positive attitudes towards automation in the post-survey question, "Describe your feelings towards using the autopilot." User 5 indicated, "I enjoy using it. Probably one of the first game[s] where I experience using autopilot and manual control. I think it's pretty useful…"

Encouraging continued contribution from users is important to crowdsourcing platforms, and the post-survey results indicate that *Busy Beeway* motivates users to further contribute to research. Users responded to a 5-point Likert scale question, "I would like to play more to contribute to human-automation collaboration research," with 81% agreeing (4 or 5), 16% neutral (3), and 3% disagreeing (1 or 2). In addition, the survey question, "I would like to play more to increase my scores," reveals that the scoring system in *Busy Beeway* keeps users engaged, with 72% of users agreeing, 16% neutral, and 12% disagreeing.

4.2 Game Interaction

We explored user behaviors and beliefs using both the survey questions and data collected by *Busy Beeway*, including use of and willingness to collaborate with automation, difficulty of the game controls, and learned trust in automation.

We investigated the collaborative navigation success and efficiency as a function of automation usage. Figure 3(a) shows that



(a) Training (b) Level 1 (100 obstacles) (c) Level 2 (200 obstacles) (d) Level 3 (300 obstacles)

Figure 2: The training level (a) and three trial levels (b-d) used in Busy Beeway. The flower path in (a) is designed to train users on collaborative navigation of Belinda.

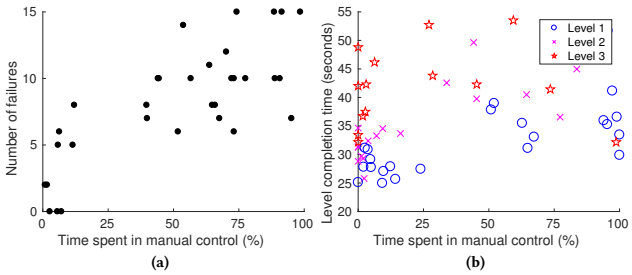


Figure 3: Percentage of time users spent in manual control vs. (a) number of failures across all three levels and (b) total time to complete all levels.

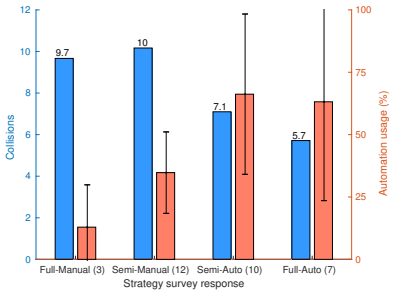


Figure 4: Survey responses about a user's preferred strategy for automation usage with number of users that self-reported each response shown in parenthesis. The average number of collisions (blue bars) is shown with the average automation usage (red bars) for each group.

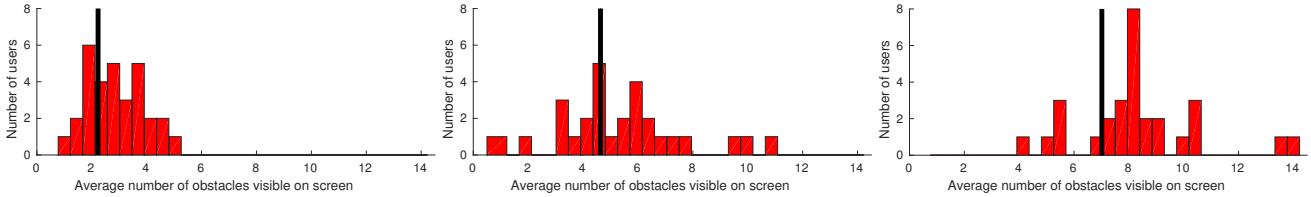
users who spend more time in manual navigation typically encounter more collision events. On the other hand, Figure 3(b) indicates that the completion time is not heavily impacted by users' automation usage. These results indicate that the collaboration is less effective when excessive control is exercised, as it is more likely to result in collisions. These results are likely due to the fact that collision avoidance with large number of high speed stochastically moving obstacles is very difficult for users without automation.

In response to the post-survey question, "Was it easy to control Belinda?", 44% of users answered "Yes," while 56% answered "No". We believe that since levels were designed to be difficult, it was hard for the users to complete levels without using of automation. In response to the survey question, "Were any parts of the user interface useful?", 50% of users mentioned the guidance arrow (6 in Figure 1).

We also investigated the users' situational and learned trust in the automation. In response to, "The autopilot was competent (capable and control) guiding Belinda on its own," 78% of users agreed, 6% disagreed, and 16% were neutral. In addition, in response to, "Without the help of any guidance assistants, I can do better than the autopilot," 9% agreed, 69% disagreed, and 22% were neutral. These responses indicate that users trust the capabilities of automation and believe that it can aid performance. It did not mean that users felt incapable of controlling Belinda, since in response to "I felt competent (capable and control) guiding Belinda on my own," 53% of users agreed, 38% disagreed, and 9% were neutral. However, we believe that our user group was biased towards automation, since in response to two questions about learned trust in automation, "I would enjoy having a self-driving car take me home from work," and "I would trust an automated bus to drive my kids to school," 91% agreed with the former, and 63% agreed with the latter.

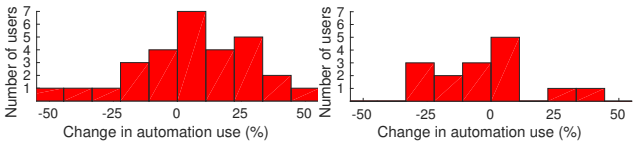
We investigated the preferred collaboration strategy with automation via a multiple choice question on users' self-reported preferred strategy and data from game play. The numbers in parenthesis on the horizontal axis in Figure 4 show that most users self-reported that they prefer to "Guide Belinda yourself and let the autopilot jump in occasionally" (Semi-Manual) and "Let the autopilot run and jump in occasionally" (Semi-Auto). The red bars show these users' actual automation usage, and the blue bars show the average number of collisions. Users' self-reported usage demonstrates general consistency with their actual automation usage. The higher reliance on automation can also result in lower collisions, also seen in Figure 3(a).

We also analyzed the change in automation usage after a user witnessed a collision that occurred during user (Figure 6 (a)) or automation (Figure 6 (b)) control. Figure 6 indicates that users use automation more after witnessing a collision under manual control. However, automation usage is unchanged after users witness a failure caused by the automation. While the assessment that the former is likely due to the user quickly realizing that the level is difficult for them, and automation may be a way for them to improve performance, is straightforward, users were unexpectedly impervious to the lack of transparency in the cause of automation failure [Dragan et al. 2013]. User 24 responded, "It is difficult to trust where it may go," and User 3 responded, "It ran against my sense of control." This non-intuitive result has important implications for transparency of automation in stochastic environments, as it indicates that there is a certain tolerance for error in the automation, perhaps because of the underlying uncertainty in the environment. Adapting the interface to explore legibility and transparency further in future experiments would be straightforward within *Busy Beeway*.



(a) Level 1 (b) Level 2 (c) Level 3

Figure 5: Average number of obstacles visible on the screen when users took manual control. Average number of obstacles visible on screen throughout the level marked (black line) is 2.2 in Level 1, 4.6 in Level 2, and 7.0 in Level 3.



(a) (b)

Figure 6: Differences in percentage of time spent under automation control after first collision in (a) manual and (b) automation control. Negative values indicate less automation usage, while positive indicates more.

In response to a post-survey question, "Were there particular scenarios in which you felt it was necessary for you to take over the autopilot (automation)?", 36% of users indicated that they took control when many obstacles were nearby. Figure 5 shows the average number of obstacles visible on screen at the instant that users take manual control, with the average number of obstacles for each level shown in the black vertical bar for reference. Figure 5 supports this claim, as more users applied manual control when the number of on screen obstacles exceeded the average for that level. Additionally, Figure 3(b) shows that automation usage increases with difficulty.

4.3 Comparison to Full Automation

We apply available benchmarks to compare performance of collaborative navigation with state-of-the-art planning algorithms. The success rate under collaborative navigation is determined by the total number of successfully completed levels divided by the total number of attempts made by all the users; the success rate of the fully automated methods were calculated out of 100 runs.

We compare performance of collaborative navigation with APF, A* [Bacha et al. 2008], VO [Fiorini and Shiller 1998], and Dynamic Risk Tolerance planning (DRT) [Chiang et al. 2017]. A* discretizes the environment into 2D grids (at 0.13 units resolution) in order to find the shortest resolution-complete collision-free path (considering all obstacles). To accommodate moving obstacles, the graph search procedure is repeated at 3Hz. VO computes the control action in the robot's velocity space, considering obstacles only within 4 units, and presuming that those obstacles move at a constant velocity. Unlike APF and A*, which require only the knowledge of obstacle position information, VO also requires obstacle velocities, which is typically more difficult to obtain. DRT predicts the stochastic obstacle motion via forward reachability analysis offline,

Table 1: Performance of all users and automation in terms of success rate and average completion time

	Level 1		Level 2		Level 3	
	Success (%)	Time (s)	Success (%)	Time (s)	Success (%)	Time (s)
All Users	54	33±6.9	34	35±6.4	32	42±7.0
APF	95	21±2.1	81	26±3.8	56	30±4.7
VO	94	24±0.4	81	24±0.7	57	25±1.2
A*	46	25±0.4	33	24±0.5	5	25±0.6
DRT	100	27±5.3	98	36±7.9	92	40±9.5

and thus requires not only obstacle velocity, but also knowledge of the stochastic dynamics of obstacles. DRT incrementally constructs a tree in the robot's state-time space via random sampling; nodes in the tree are robot positions and times with "acceptable" collision probabilities, e.g., those that are less than a time-varying heuristic function. The parameters used for DRT are identical to those in [Chiang et al. 2017] except for the tree time step, which is set to 1/3s, and the obstacle detection range, which is set to 25 units. Note that aside from VO, all planning algorithms can react to obstacles outside of the users' screen area. The screen centers around Belinda and is 16 units wide and 9 units high.

Table 1 shows that collaborative navigation has a success rate higher or on par with A*, while APF and VO outperforms collaborative navigation significantly in both success rate and completion time. APF and VO compute actions every time step, thus they can respond to potential collisions faster than a user can. However, the success rate of these planning algorithms reduces dramatically from Level 2 to Level 3, while the success rate of collaborative navigation remains similar in Levels 2 and 3. This is possibly due to the fact that APF and VO do not consider obstacle density, while users tend to take manual control when the obstacle density increases (in Section 4.2). Lastly, DRT has a much higher success rate than all methods. This is due to the fact that it explicitly samples collision probability in the robot's state-time space and thus can identify and avoid regions of high collision probability.

Table 2 lists times for the five users that completed each level in the shortest amount of time. For each run, a comparison to the fully automated case is also provided. Recall that due to obstacle stochasticity, each run is unique. Due to differences in Android system performance, standard deviation of the fully automated case is also provided. Top users can achieve shorter completion time more often in Level 2 and 3, which have higher numbesr of obstacles. Interestingly, little manual control is used in these cases. For example, Users 10 and 11 used the automation 97.7% and 99.4%

Table 2: Shortest completion times by users, as compared to full automation. (Times listed are in real-time and may vary slightly with Android system performance; standard deviation is computed from 20 trials.)

Level	User Number	Manual Control [%]	User Time [s]	Automation Time [s]
1	14	9.3%	25.0	**22.7±0.1**
	18	3.3%	**25.1**	25.2±0.1
	10	14.3%	25.7	**24.2±0.1**
	31	9.7%	27.1	**25.3±0.1**
	13	23.9%	**27.5**	30.5±0.1
2	10	2.3%	**25.8**	(fail)
	31	0.0%	**28.9**	29.0±0.1
	1	1.9%	29.3	**29.1±0.1**
	11	0.6%	**29.7**	(fail)
	17	0.0%	**31.2**	31.4±0.1
3	24	97.6%	**32.2**	36.0±0.3
	18	1.4%	**32.2**	32.4±0.2
	13	0.0%	**33.4**	33.7±0.4
	10	1.8%	**36.7**	39.3±0.3
	12	22.2%	37.5	**34.9±0.3**

of the time, while the fully automated solution failed. We believe this shows top users can guide Belinda out of local minima formed by moving obstacles (a common issue for APF-based algorithms), and thereby achieve shorter completion times.

5 CONCLUSION

Busy Beeway provides a platform for investigation of collaborative navigation in large human subject studies. A preliminary study suggests that, in combination with pre- and post-surveys focused on the human factors of interest, *Busy Beeway* can reveal interesting phenomena for further investigation in more realistic experimental platforms. Given the dearth of platforms available to rigorously quantify performance in dynamic, stochastic environments, and that it is deployable on widely ubiquitous mobile devices, we believe *Busy Beeway* is an enabling technology for crowdsourced studies in human-centered autonomy.

ACKNOWLEDGEMENTS

We thank Valerie Sheffey, Dr. Patrick Kelley, Marcos Lemus, Laura Patrizi, and Danny Gomez for their contributions in the development of *Busy Beeway*, and Neera Jain and Tahira Reid for discussions on trust. This material is based upon work supported by the National Science Foundation under Grant Numbers IIS-1528047, IIS-1553266 (Tapia, CAREER), and CMMI-1254990 (Oishi, CAREER). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Andrew Bacha, Cheryl Bauman, Ruel Faruque, Michael Fleming, Chris Terwelp, Charles Reinholtz, Dennis Hong, Al Wicks, Thomas Alberi, David Anderson, et al. 2008. Odin: Team VictorTango's entry in the DARPA urban challenge. *J. Field Rob.* 25, 8 (2008), 467–492.

Rodrigo Benenson, Stéphane Petti, Thierry Fraichard, and Michel Parent. 2006. Integrating perception and planning for autonomous navigation of urban vehicles. In *Intel. Rob. Sys. IEEE/RSJ Intl. Conf.* 98–104.

James Bruce and Manuela Veloso. 2002. Real-time randomized path planning for robot navigation. In *Intel. Rob. Sys. IEEE/RSJ Intl. Conf.* 2383–2388.

Sylvain Bruni, Jessica J Marquez, Amy Brzezinski, Carl Nehme, and Yves Boussemart. 2007. Introducing a human-automation collaboration taxonomy (HACT) in command and control decision-support systems. In *Intl. Command and Control Res. and Technol. Symp.* 1–13.

John Canny. 1988. Some algebraic and geometric computations in PSPACE. In *ACM Symp. Theory Proc.* 460–467.

John Canny and John Reif. 1987. New lower bound techniques for robot motion planning problems. In *Found. Comp. Sci. Symp.* 49–60.

Hao-Tien Chiang, Baisravan HomChaudhuri, Abraham P. Vinod, Meeko Oishi, and Lydia Tapia. To appear, 2017. Dynamic Risk Tolerance: Motion Planning by Balancing Short-Term and Long-Term Stochastic Dynamic Predictions. In *Proc. IEEE Intl. Conf. Rob. Auto.* 3762–3769.

Hao-Tien Chiang, Nathanael Rackley, and Lydia Tapia. 2015. Stochastic ensemble simulation motion planning in stochastic dynamic environments. In *Intel. Rob. Sys. IEEE/RSJ Intl. Conf.* 3836–3843.

Kajal T. Claypool and Mark Claypool. 2007. On frame rate and player performance in first person shooter games. *Multimedia Sys.* 13, 1 (2007), 3–17.

Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popovic, and Foldit Players. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466 (2010), 756–760.

Dick de Waard, Monique van der Hulst, Marika Hoedemaeker, and Karel A Brookhuis. 1999. Driver behavior in an emergency situation in the Automated Highway System. *Trans. Hum. Fact.* 1, 1 (1999), 67–82.

Joost CF de Winter and Dimitra Dodou. 2011. Preparing drivers for dangerous situations: A critical reflection on continuous shared control. In *Sys., Man, and Cybern. IEEE Intl. Conf.* 1050–1056.

Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. 2013. Legibility and predictability of robot motion. In *Hum. Rob. Inter. ACM/IEEE Intl. Conf.* 301–308.

Andreas Dunser, Martin Lochner, Ulrich Engelke, and David Rozado Fernandez. 2015. Visual and manual control for human-robot teleoperation. *Comp. Graph. App.* 35, 3 (2015), 22–32.

Alexandra Eveleigh, Charlene Jennett, Ann Blandford, Philip Brohan, and Anna L. Cox. 2014. Designing for Dabblers and Deterring Drop-outs in Citizen Science. In *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.* 2985–2994.

Alexandra Eveleigh, Charlene Jennett, Stuart Lynn, and Anna L. Cox. 2013. 'I Want to Be a Captain! I Want to Be a Captain!': Gamification in the Old Weather Citizen Science Project. In *Proc. Intl. Conf. Gameful Design, Research, and Applications.* 79–82.

Dave Ferguson, Nidhi Kalra, and Anthony Stentz. 2006. Replanning with RRTs. In *Proc. IEEE Intl. Conf. Rob. Auto.* 1243–1248.

Paolo Fiorini and Zvi Shiller. 1998. Motion Planning in Dynamic Environments using Velocity Obstacles. *Intl. J. Rob. Res.* 17, 7 (1998), 760–772.

Frank Flemisch, Johann Kelsch, Christan Löper, Anna Schieben, Julian Schindler, and Matthias Heesen. 2008. Cooperative Control and Active Interfaces for Vehicle Assistance and Automation. *FISITA World Automotive Congress* (2008).

William Glover and John Lygeros. 2004. A Stochastic Hybrid Model for Air Traffic Control Simulation. In *Proc. ACM Intl. Conf. Hybrid Sys.: Comp. Control.* 372–386.

Dirk Helbing and Péter Molnár. 1995. Social force model for pedestrian dynamics. *Phys. Rev. E.* 51, 5 (1995), 4282–4286.

Kevin Anthony Hoff and Masooda Bashir. 2015. Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust. *Trans. Hum. Fact.* 57, 3 (2015), 407–434.

Mishel Johns, Brian Mok, David Michael Sirkin, Nikhil Manjunath Gowda, Catherine Allison Smith, Walter J. Talamonti Jr, and Wendy Ju. 2016. Exploring Shared Control in Automated Driving. In *Hum. Rob. Inter. ACM/IEEE Intl. Conf.* 91–98.

Oussama Khatib. 1986. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Intl. J. Rob. Res.* 5, 1 (1986), 90–98.

Steven LaValle and Rajeev Sharma. 1995. A framework for motion planning in stochastic environments: modeling and analysis. In *Proc. IEEE Intl. Conf. Rob. Auto.*, Vol. 3. 3057–3062.

Nick Malone, Hao-Tien Chiang, Kendra Lesser, Meeko Oishi, and Lydia Tapia. 2017. Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field. *IEEE Trans. Rob.* PP (2017), 1–15. Issue 99.

Raja Parasuraman and Victor Riley. 1997. Humans and Automation: Use, Misuse, Disuse, Abuse. *Trans. Hum. Fact.* 39, 2 (1997), 230–253.

Justin Storms, Kevin Chen, and Dawn Tilbury. 2017. A shared control method for obstacle avoidance with mobile robots and its interaction with communication delay. *Intl. J. Rob. Res.* 36, 5-7 (2017), 820–839.

Luis von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.* 319–326.

Anqi Xu and Gregory Dudek. 2016. Maintaining Efficient Collaboration with Trust-Seeking Robots. In *Intel. Rob. Sys. IEEE/RSJ Intl. Conf.* 3312–3319.

Kathrin Zeeb, Axel Buchner, and Michael Schrauf. 2015. What determines the take-over time? An integrated model approach of driver take-over after automated driving. *Accident Analysis & Prevention* 78 (2015), 212–221.