

Getting the Shot: Socially-Aware Viewpoints for Autonomously Observing Tasks

Adam Allevato*, Andrew Sharp†, and Mitch Pryor‡

Department of Mechanical Engineering,
The University of Texas at Austin
Austin, TX

Email: *allevato@utexas.edu, †asharp@utexas.edu, ‡mpryor@utexas.edu

Abstract—In this work, we present an algorithm for autonomously determining the appropriate location from which to observe a human or robot agent (actor) while it completes a task in dynamic environments. We develop theory for selecting such a location using forward physical simulation of randomly-selected candidate viewpoints. The simulated points provide obstacle avoidance, and by incorporating a modified version of the Social Force Model, candidate viewpoints adjust themselves so that they do not encroach on the actor’s personal space and/or safety region. The best observer position is chosen from these candidates to provide the most complete view of the task volume, taking into account the occlusion caused by the actor itself. We show that our algorithm works under a variety of task volume configurations, actor types (human and robot), and environmental constraints. Finally, the paper shows the results of hardware deployment on a two-robot system—one observer, and one actor. The paper concludes by examining the social impacts of deploying autonomous observation algorithms on real-world systems.

I. INTRODUCTION

As autonomous physical systems become more widespread in society, the need arises for robots that are both functional and socially aware. Many functional robots, such as unmanned aerial vehicles (UAVs) used for sport filming, have limited “follow” capabilities, but may not include robust obstacle avoidance, or may not respect humans’ personal space. This can cause issues when operating in confined or crowded spaces, or may simply make the filming subject uncomfortable because of the UAV’s proximity. On the other end of the spectrum lie socially aware robots, which have arisen from research in human-robot interaction. Robots such as Jibo [1] and Leonardo [2] may be “human-compatible” and instill human confidence, and can perform many software-based actions, such as face recognition, but do not perform many useful physical tasks.

This paper attempts to strike a balance between the two extremes, developing an algorithm for a socially-aware robot that observes a target agent’s action (a useful physical navigation-based task), while respecting the personal space of the agent (human or otherwise). We refer to the observing robot as the “observer,” and the target agent performing the action, the “actor.” While many research efforts have addressed the theoretical problem of following another agent and keeping it in view in the presence of obstacles, this paper

seeks to address the problem of task observation with social awareness, which adds additional constraints to the problem. With regards to applications, some research has addressed robot arm-based camera placement for local task observation [3] [4], but our approach explicitly and generally models both obstacles and visibility, which other works fail to take into account. Our research combines ideas from human-robot interaction, social robotics, and path planning. It also holds implications for many existing consumer robotics platforms, most notably UAVs.

When working with a human actor, the observer needs to find the optimal position from which to view the task, while satisfying three objectives (listed in order of decreasing importance):

- 1) The observer should avoid obstacles in the environment.
- 2) The observer should respect the actor’s personal space to avoid interfering with the task.
- 3) The task being performed should be completely in view of the observer, with minimal occlusion from both the environment and the actor.

The same objectives can be applied when the observer is a robot, substituting the idea of a “safety zone” for personal space, which could vary from robot to robot. Based on these requirements, our work uses an explicit visibility calculation, geometric definitions of the world, actor, and task region, and the social force model (popular in the human-robot interaction literature) to determine social dynamics and personal space. The algorithm is continuous and sampling-based, making it able to operate in larger areas more accurately, rather than calculating an explicit cost function over a small discretized region.

The next section explores related work from various fields of robotics, and discusses how this research builds upon it. Section 3 defines the world and sampling algorithm. In Section 4, we present implementation and hardware experiment details. Section 5 contains the results of the experiments. The last two sections contain a discussion of results, conclusions, and future work.

II. RELATED WORK

One of the earliest works on autonomously maintaining visibility of a target agent was performed by LaValle et al. [5]. This paper introduced the idea of solving for a path through a cluttered environment. The path was found to always maintain line of sight without foreknowledge of the target agent’s path through the environment. The theory of maintaining visibility has been extended in various ways [6], including solving the reverse problem of hiding from a target agent [7], pursuing with limited sensors [8], and observing multiple target agents with multiple observers [9]. Cancemi et al. [10] extend the visibility problem to include a probabilistic representation of the target agent, and maximize a spatial visibility function via gradient ascent. Our approach is similar. We optimize for visibility of a region rather than a single point; however, we sample points in space and then apply social cues to determine appropriate viewing positions.

Autonomous tracking systems have been extended to various domains in both simulation and the real world, inspiring the research in this paper. An exploration of tracking strategies on rover-like vehicles [11] showed that the best results were obtained using laser-based tracking (a technology that has since been improved upon with 3D depth sensors). In simulation, researchers have studied finding smooth and cinematographically pleasing paths in 3D space [12]. Another study pre-calculates a cost map to quickly develop leader-follower paths in a virtual tour guide setting [13].

This paper also draws from the rich research on human-aware robot navigation. Specifically, the social force model of human behavior [14] has been extended into the realm of robot navigation. In this work, we utilize an adapted version of Papadakis et al.’s distorted personal space model [15], which posits that the zone of personal space around a human is elliptical; that is, humans are more sensitive to having other agents directly in front of and behind them (as opposed to agents being to the human’s right or left). Other examples of human-aware navigation include [16] and [17]. Researchers have studied human social cues specifically in the context of interacting with robots. These works include learning motion patterns in indoor settings to predict behavior [18], as well as robots following subjects in wheelchairs and dense indoor environments [19]. Yu et al. recently explored the possibility of commanding an UAV to move and perform simple actions using only human gaze and gaze trajectories. [20]. A similar study shows that humans often interact with UAVs as they would with a pet, using interpersonal gestures such as beckoning or waving [21]. These works are good examples of how social cues can be applied to deployed functional robots, which is what we try to achieve in this work.

Our work is most similar to that of Schroeter et al. [22], which seeks to find the optimal location to observe a human subject in good lighting (reducing glare, overexposure, etc.). Our work also considers locations for viewpoint suitability, but optimizes for a clear view of the actor’s task (while also respecting personal space or a safety zone).

III. THEORY

We begin the formal definition of the *viewpoint selection algorithm* by stating our assumptions. The world is 2D Cartesian and every point can be classified as either free space or impassable obstacles. (Note that this work could easily be generalized to 3D, but 2D is sufficient for our use cases.) We make the following assumptions to constrain our problem. The obstacles are static and the observer is holonomic, as is (roughly) the case with quadrotor UAVs, or holonomically-steered ground vehicles. We also assume that the actor’s identity (human, robot or otherwise) is known, and that the task volume \mathcal{S} (see below) is known for the actor. In a real-world setting, new robots could communicate their occlusion footprint \mathcal{O}_T and task volume \mathcal{S} to the observer.

While other works discretize the state space to build a visibility map, our system performs visibility calculations online to allow more flexibility as the actor completes tasks in different locations. With this in mind, we keep a continuous representation of the world and sample from it randomly, generating a set of candidate observation locations (compare to [23]). Objective 1 is partially satisfied by immediately discarding any candidates that lie within obstacles. Objective 2 is achieved by simulating the social dynamics for a fixed length of time to allow the candidates to reach points where the social force reaches an equilibrium. The simulated candidates are each evaluated based on how well they can see the task being performed, and the candidate providing maximum visibility is chosen as the optimal viewpoint.

The physical environment is represented as follows. The world consists of a set of obstacles, \mathcal{O}_{world} , each element of which is a polygon in 2-space. In addition to the obstacles, the world contains an actor, which has a pose $\mathcal{T} = (x_T, y_T, \theta_T)$, as well as a camera-holding observer with pose $\mathcal{C} = (x_C, y_C, \theta_C)$. The actor has a physical occlusion volume, \mathcal{O}_T , represented by a polygon. The total set of occluding objects \mathcal{O} contains both the actor as well as the world: $\mathcal{O} = \mathcal{O}_{world} \cup \mathcal{O}_T$. In this paper, we also use lowercase vectors to represent only the translational elements of pose: $\mathbf{t} = (x_T, y_T)$, $\mathbf{c} = (x_C, y_C)$. Figure 1 shows various components of the algorithm visually.

First, a candidate set of n observation positions $\Gamma_0 = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n)$ is drawn randomly from the environment. This is immediately filtered to exclude candidates that lie within obstacles: $\Gamma_i = \Gamma_0 \setminus \mathcal{O}$.

The next set of variables defines the social context of the environment. Following a basic version of the social force model [14], we assume that a human exerts “social forces” on nearby entities as those entities approach the actor’s personal space (see Figure 1). Following the lead of Papadakis et al. [15], we distort the personal space region to be larger directly in front of and behind the human. We achieve this with an anisotropy factor α , which distorts the circular personal space into an ellipse according to the following formula:

$$\alpha = (1 - \gamma) + (\gamma |\cos(\theta)|) \quad (1)$$

Where γ determines the amount of circular distortion ($\gamma = 0$ gives a circle), and θ is the angle between the actor’s heading and the observer’s position: $\theta = \theta_T - \text{angle}(\mathbf{t} - \mathbf{c})$.

We calculate the social force, F_s according to the equation

$$F_s = mk_s\alpha \max(0, d - r_s)(-\mathbf{t} - \mathbf{c}) \quad (2)$$

Where m is the mass of the robot, k_s and r_s are the strength and range of the social force, respectively, α is given by Equation 1, and d is the straight-line distance between the observer and actor, $d = \|\mathbf{t} - \mathbf{c}\|$. Note that in the case of a robot actor, the concept of personal space does not make sense, but instead can be replaced by a “minimum safe distance” to avoid a robot collision.

The physical force F_p repels the robot from any obstacles:

$$F_p = \sum_{\mathcal{O}}^{\sigma} mk_p \max(0, d - r_p)(-\mathbf{v}_{closest}(\sigma)) \quad (3)$$

Where, again, k_p and r_p are the strength and range of the physical force. In this equation, $\mathbf{v}_{closest}$ is a vector pointing from the observer position \mathbf{c} to the nearest point lying on the obstacle σ ’s boundary.

The total force F on the robot is the sum of the individual forces, $F = F_s + F_p$.

While simulating social navigation trajectories, our robot obeys the Newtonian motion equation $F = ma$ (making the holonomic assumption is safe here, since the robot need not actually follow these trajectories). The simulation lasts for time t_f with a timestep of Δt . We call the set of candidates after completing social simulation Γ_f .

Finally, we must calculate the visibility score for each candidate, taking all occlusions into account. The observer has a 2D field of view (FOV), φ . We define a *task volume* \mathcal{S} , which is attached to the actor’s reference frame \mathcal{T} . \mathcal{S} is a set of polygons representing the area in which manipulation tasks will take place. For example, a human will most likely work in the area directly in front of them, with extents limited by their arm length. This task space can be further refined based on other factors (see [23]). When the actor is a robot, the task space can be defined based on simulation and empirical evaluation of the best manipulation zones around the robot. These zones will be based primarily on known reach and workspace of the manipulator, or, alternatively, on the concept of “action-related places” (for more information, see [24]).

To find the visibility score, we cast rays from \mathbf{c}_i in all directions within the observer’s cone of visibility, $[\theta_C - \varphi/2, \theta_C + \varphi/2]$, until they strike an obstacle. By connecting the endpoints of the ray, we build a *visibility volume* polygon, \mathcal{V}_i . The intersection, \mathcal{W}_i , between the visibility volume and the task volume is calculated as $\mathcal{W}_i = \mathcal{V}_i \cap \mathcal{S}$ (refer to Figure 1). From this definition we calculate the visibility score, including a penalty for distance from the actor to discourage extremely wide-angle shots:

$$S_{vis,i} = \text{area}(\mathcal{W}_i) / \text{area}(\mathcal{S}) - p\|\mathbf{T} - \mathbf{c}_i\|_2^2 \quad (4)$$

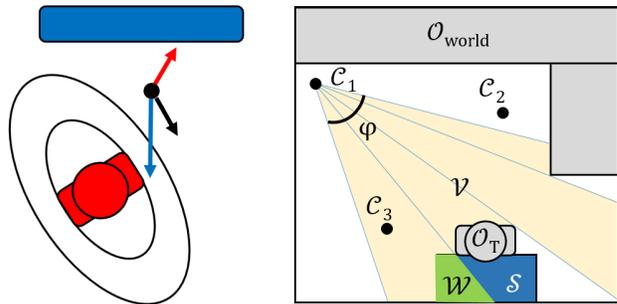


Fig. 1: Left: the primary forces exerted by the social force model: obstacle avoidance (blue) and personal space avoidance (red). The resultant force vector is shown in black. Right: A visual representation of the most important variables in the viewpoint selection algorithm. The green viewable area, \mathcal{W} , is the union of the blue task volume, \mathcal{S} , and the yellow view volume, \mathcal{V} (shown here for camera point \mathcal{C}_1). The set of occluders, \mathcal{O} is the union of the world \mathcal{O}_{world} and the actor volume \mathcal{O}_T . Best viewed in color.

TABLE I: Parameter values for viewpoint selection

Parameter	Symbol	Value
Camera field of view, rad	ϕ	1.5
Range of physical force, m	r_p	0.35
Range of social force, m	r_s	0.7
Number of viewpoint candidates	n	100
Social force simulation time, s	t_f	10
Simulation timestep, s	Δt	0.2
Simulated robot mass, kg	m	5
Anisotropy factor	γ	0.4
Distance penalty factor	p	0.25
Human occluder width, m	–	0.4572 [25]
Human occluder depth, m	–	0.24384 [25]

Where p is a tuning parameter for the distance penalty. The final selected viewpoint is chosen as that having the highest visibility score:

$$\mathcal{C}_{best} = \max_{\Gamma_f}(S_{vis}) \quad (5)$$

The robot can then use its navigation software to go to this point and observe the task.

IV. IMPLEMENTATION

The viewpoint generation algorithm was implemented in Python, using the parameter values shown in Table I. A version of the source code has been made publicly available under the open-source BSD license [26]. The code also contains a binding to the Robot Operating System (ROS) framework for hardware control. ROS is open-source software used for motion planning, trajectory execution, navigation, and sensor integration. The software is modular in nature and emphasizes a high level of abstraction [27]. The main software packages used for manipulator control and visualization are *MoveIt!* and *RViz* respectively.

The hardware systems used in this effort were the Clearpath TurtleBot 2 and UT Austin’s highly customized VaultBot dual-arm mobile manipulator. The TurtleBot is a

low-cost differentially steered platform. The VaultBot (Figure 2) uses a Clearpath Husky mobile platform and two Universal Robots UR5 6DOF industrial manipulators, which include controller-level collision detection for safe operation. These systems are controlled via ROS drivers provided by Clearpath (TurtleBot, Husky) and ROS-Industrial [28] (UR5s). The VaultBot is equipped with various sensors, including a SICK LIDAR, a Robotiq two-finger gripper, and an Intel RealSense R200 depth camera (Figure 2, right and left UR5 respectively).

For this work the TurtleBot was the observer, and was equipped with an action camera to watch the VaultBot actor. The TurtleBot selected viewpoints based on the VaultBot’s navigation map and world position. We use contour detection to vectorize the world map, since our algorithm is not designed for the discrete maps generated by ROS. We generate candidates in a 3 m x 3 m box centered around the actor’s position (rather than the whole world) to increase efficiency.

When the VaultBot is the actor, the volume of interest is the manipulator workspace (each UR5 has a 850 mm working radius). The UR5s are mounted to the bulkhead about 0.30 m apart, resulting in workspace overlap and some volume occupied by the VaultBot base (Figure 2). Tasks above the VaultBot were not taken into account for this work, leaving a volume on either side for single manipulator tasks, as well as a small volume in front that is accessible to both UR5s. The task volume for each manipulator was approximated by a semicircle, ignoring volume of interest regions above the VaultBot’s footprint. While the VaultBot can use both UR5s simultaneously, most tasks require only one, limiting the volume of interest to a single side.



Fig. 2: The VaultBot dual-arm mobile platform (left). Schematic showing the VaultBot’s workspace for each manipulator (right UR5: blue, left UR5: red), viewed from above (right). Figure best viewed in color.

V. RESULTS

The results from the VaultBot/TurtleBot task are shown in Figure 3. As described in the Theory section, The initial sampling of viewpoints is random. After forward simulation of the social dynamics (shown by the black lines in the figure, the final viewpoint candidates tend to cluster together, and avoid obstacles. Note that while we use a simplified map to generate the desired viewpoint, the robots operate using the full-resolution map, allowing robust localization and navigation. The TurtleBot’s navigation software also accounts for dynamic obstacles while seeking a. The result is a viewpoint

that is closer to the task and more visually informative than what could be achieved using an observer located at a “safe following distance” or stationary observation position.

In addition to real-world implementation, simulations also give results and validate the method for task configurations. Figure 4 shows the results of viewpoint selections performed in simulation. Worlds with various amounts of free space are used, including a tightly constrained “end of hallway” environment. The selected viewpoints satisfy the original goals developed in the introduction, providing a view of the task volume while respecting distance from obstacles and the actor.

A few interesting failure cases should be noted in Figure 4, including sampled camera locations which lie within obstacles (due to numerical error), as well as “optimal” locations which lie within the task volume (still giving the highest score according to Equation 4). Both failure cases can be easily controlled by disallowing camera viewpoints to lie within the task volume and/or obstacles, and enforcing this rule after the social dynamics have been simulated.

VI. DISCUSSION

The results of the experiments (both hardware and simulation) are encouraging, as they suggest that socially-aware following and observation is achievable in unstructured environments and using low-cost platforms such as the TurtleBot. The described method comes with some limitations. One of the most important is that the world is assumed to be stationary during viewpoint selection. Depending on the setting and task, this assumption may prove unrealistic. However, these were not concerns for our motivating application: that of filming a single task in a generally static hazardous environment. Another important simplification is that our algorithm does not regard the reachability of a point in the world - the observer may not be able to navigate to the best viewpoint because it is physically blocked by the actor. In this case, the correct behavior would be to request that the actor move, or simply to choose the next best navigable viewpoint. Despite these limitations, however, our algorithm has potential in a variety of situations

For human actors, following, observing, and/or filming raises ethical and privacy-related questions. These concerns reinforce the need to navigate in socially-aware modes. Additional task-specific constraints could allow filming only in public areas, or in areas where the robot perceives that it is acceptable to film. Other specific objectives, such as avoiding crowded areas, could also be encoded into our force-based model.

Once ethical and privacy concerns have been addressed, the ability to autonomously observe other actors with a robot can impact society in profound and often unexpected ways. For example, conservationists have used camera-equipped UAVs to observe the population and distribution of Sumatran orangutans [29]. Another possibility is a “robot documentary”, in which a robot could automatically follow an actor around an environment, observing and filming from locations that are safe, yet provide a good view of the actor’s actions.

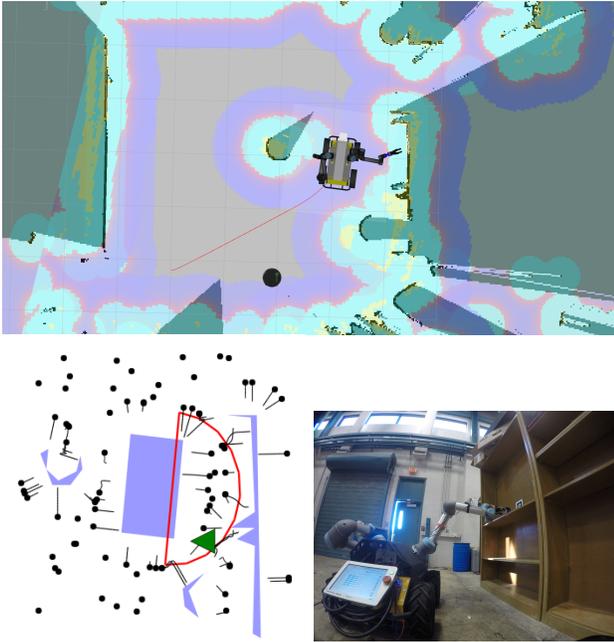


Fig. 3: Top: overhead view of the two robots and world map in the RViz visualizer. The TurtleBot observer is the black dot at bottom center. Lower left: viewpoint selection results for this case, centered on the VaultBot footprint. Blue regions are obstacles, and the task volume is outlined in red. The black lines show the simulated socially-guided motion of the viewpoint candidates, ending at the black circles. The green triangle represents the camera location giving the highest score (the candidates are always oriented so as to be pointing at the center of the task volume). Lower right: the TurtleBot's view after navigating to the selected position.

Using this method, the robot could create an informative or instructional video. Yet another application is surveillance and situational awareness in supervised autonomy or teleoperation environments. In this case, an auxiliary robot could be used to observe an actor operating in a hazardous situation, giving better data and context to human supervisors or operators.

VII. CONCLUSIONS AND FUTURE WORK

This work developed a theory of socially-aware viewpoint selection, where an autonomous *observer* agent equipped with vision determines the best position from which to view a target *actor* agent as it performs a task. Simulation results were generated for various configurations of obstacles, actors, and task volumes, showing the technique's flexibility, and the algorithm was deployed onto robot hardware with good results.

We envision several possible extensions to this work. The case of time-varying trajectories could be introduced, perhaps with a stationary observer. The algorithm would then be modified to determine the best position from which to track an actor's task volume as that actor follows a given trajectory. Alternatively, given the actor's trajectory, a particle filter

approach could be used to predict the best observer trajectory through time. Another possible extension is the generalization of the influences of the social force model to a "task-specific" cost function, including other cues, such as motion legibility [24]. We also would like to explore extensions of this work into co-robotics, including natural language processing to help guide the robot's observation style.

ACKNOWLEDGMENTS

This material is based upon work supported under an Integrated University Program Graduate Fellowship and Los Alamos National Laboratory. We thank Benjamin Ebersole for providing hardware support.

REFERENCES

- [1] "Jibo, The World's First Social Robot." [Online]. Available: <https://www.jibo.com>
- [2] W. D. Stiehl, L. Lalla, and C. Breazeal, "A "somatic alphabet" approach to "sensitive skin"," in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, vol. 3, Apr. 2004, pp. 2865–2870 Vol.3.
- [3] R. Hurteau, S. DeSantis, E. Begin, and M. Gagner, "Laparoscopic surgery assisted by a robotic cameraman: concept and experimental results," in *1994 IEEE International Conference on Robotics and Automation, 1994. Proceedings*, May 1994, pp. 2286–2289 vol.3.
- [4] V. F. Munoz, I. Garcia-Morales, J. Morales, J. M. Gomez-DeGabriel, J. Fernandez-Lozano, and A. Garcia-Cerezo, "Adaptive cartesian motion control approach for a surgical robotic cameraman," in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, vol. 3, Apr. 2004, pp. 3069–3074 Vol.3.
- [5] S. M. LaValle, H. H. Gonzalez-Banos, C. Becker, and J. C. Latombe, "Motion strategies for maintaining visibility of a moving target," in *1997 IEEE International Conference on Robotics and Automation, 1997. Proceedings*, vol. 1, Apr. 1997, pp. 731–736 vol.1.
- [6] N. Karnad, "Robot Motion Planning for Tracking and Capturing Adversarial, Cooperative and Independent Targets," Ph.D. dissertation, University of Minnesota, Sep. 2015. [Online]. Available: <http://conservancy.umn.edu/handle/11299/175532>
- [7] M. A. Marzouqi and R. A. Jarvis, "Robotic covert path planning: A survey," in *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*, Sep. 2011, pp. 77–82.
- [8] N. M. Stiffler and J. M. O'Kane, "Pursuit-evasion with fixed beams," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4251–4258.
- [9] L. E. Parker, "Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets," *Auton. Robots*, vol. 12, no. 3, pp. 231–255, May 2002. [Online]. Available: <http://dx.doi.org/10.1023/A:1015256330750>
- [10] L. Cancemi, M. Innocenti, and L. Pollini, "Guidance Augmentation for Improved Target Visibility," in *AIAA Guidance, Navigation and Control Conference 2014*. American Institute of Aeronautics and Astronautics, Jan. 2014. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/6.2014-1477>
- [11] J. Graham and K. Shillcutt, "Robot Tracking of Human Subjects in Field Environments," *ResearchGate*, 2003.
- [12] D. Nieuwenhuisen and M. H. Overmars, "Motion planning for camera movements," in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, vol. 4, Apr. 2004, pp. 3870–3876 Vol.4.
- [13] T.-Y. Li, T.-h. Yu, and Y.-c. Shie, *On-Line Planning For An Intelligent Observer In A Virtual Factory*, 2000.
- [14] D. Helbing and P. Molnr, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, May 1995. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevE.51.4282>
- [15] P. Papadakis, P. Rives, and A. Spalanzani, "Adaptive spacing in human-robot interactions," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 2627–2632.
- [16] G. Diego and T. K. O. Arras, "Please do not disturb! Minimum interference coverage for social robots," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 1968–1973.

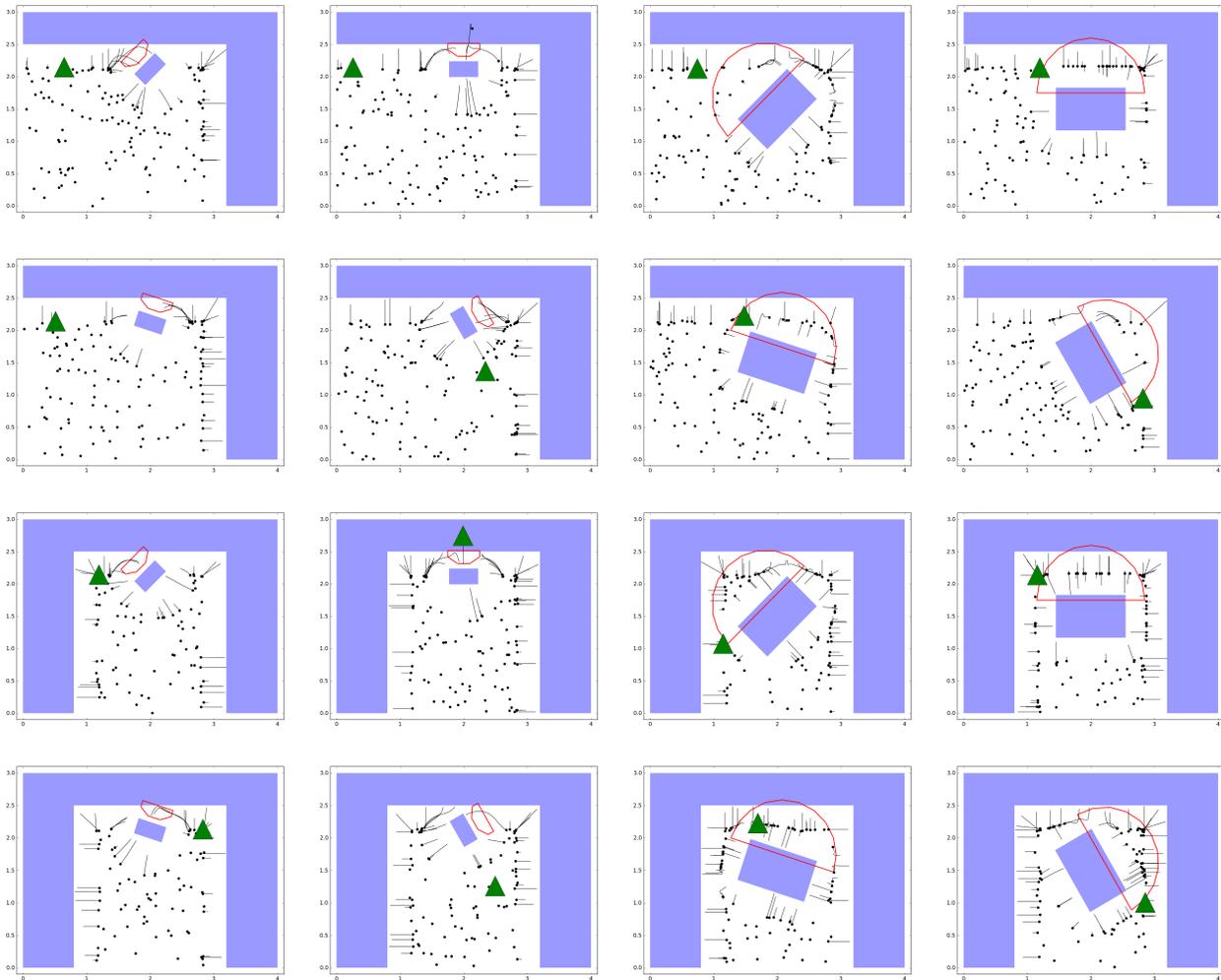


Fig. 4: Viewpoint selections for a variety of actors, poses, and obstacle configurations. (for an explanation of the diagram’s components, please see 3. The first two columns use a human as the actor, performing a small-volume task in front of them. The last two columns use geometry and task volumes from the VaultBot robot.

[17] R. A. Knepper and D. Rus, “Pedestrian-inspired sampling-based multi-robot collision avoidance,” in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, Sep. 2012, pp. 94–100.

[18] M. Bennewitz, W. Burgard, and S. Thrun, “Learning motion patterns of persons for mobile service robots,” in *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA ’02*, vol. 4, 2002, pp. 3601–3606 vol.4.

[19] T. Dewi, N. Uchiyama, and S. Sano, “Service mobile robot control for tracking a moving object with collision avoidance,” in *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, Jun. 2015, pp. 1–6.

[20] M. Yu, Y. Lin, D. Schmidt, X. Wang, and Y. Wang, “Human-Robot Interaction Based on Gaze Gestures for the Drone Teleoperation,” *ResearchGate*, vol. 7, no. 4, pp. 1–14, Sep. 2014.

[21] J. R. Cauchard, J. L. E. K. Y. Zhai, and J. A. Landay, “Drone & Me: An Exploration into Natural Human-drone Interaction,” in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp ’15. New York, NY, USA: ACM, 2015, pp. 361–365. [Online]. Available: <http://doi.acm.org/10.1145/2750858.2805823>

[22] C. Schroeter, M. Hoehemer, S. Mueller, and H. M. Gross, “Autonomous robot cameraman - Observation pose optimization for a mobile service robot in indoor living space,” in *IEEE International Conference on Robotics and Automation, 2009. ICRA ’09*, May 2009, pp. 424–429.

[23] E. A. Sisbot and R. Alami, “A Human-Aware Manipulation Planner,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1045–1057, Oct. 2012.

[24] M. Beetz, F. Stulp, P. Esden-Tempski, A. Fedrizzi, U. Klank, I. Kresse, A. Maldonado, and F. Ruiz, “Generality and legibility in mobile manipulation,” *Autonomous Robots*, vol. 28, no. 1, p. 21, Sep. 2009. [Online]. Available: <http://link.springer.com/article/10.1007/s10514-009-9152-9>

[25] “Human Figure average measurements.” [Online]. Available: <https://www.fas.harvard.edu/~loebinfo/loebinfo/Proportions/humanfigure.html>

[26] “Kukanani/camerabot.” [Online]. Available: <https://github.com/Kukanani/camerabot>

[27] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O’Reilly Media, Inc., 2015.

[28] “ROS-Industrial: Home.” [Online]. Available: <http://rosindustrial.org/>

[29] S. Wich, D. Dellatore, M. Houghton, R. Ardi, and L. P. Koh, “A preliminary assessment of using conservation drones for Sumatran orangutan (*Pongo abelii*) distribution and density,” *Journal of Unmanned Vehicle Systems*, vol. 4, no. 1, pp. 45–52, Nov. 2015. [Online]. Available: <http://www.nrcresearchpress.com/doi/abs/10.1139/jvus-2015-0015>