Inverse Reinforcement Learning for Inferring Human-Centered Costmaps

Shao-Hung Chan
shaohung@usc.edu

Aravind Kumaraguru
akumarag@usc.edu

***Abstract*—Modern advances to robotics have allowed opportunities for robots to collaborate with humans in close proximity. However, motion planning around humans can be challenging, as it can be difficult to find the tradeoff between prioritizing task efficiency and human safety [1]. Oftentimes, the level of caution necessary is a function of the current task being performed by the robot. We propose a scalable approach to infer the minimum safe distance to a human through existing motion capture data. Using Inverse Reinforcement Learning [2], we learn a reward function that penalizes close proximity to the human. We then use the learned reward to fit a parametrized, human-centered costmap that is a function of the current task being performed. This costmap can be readily utilized in motion planning algorithms [3], [4]. We show the efficacy of our approach in a simple grid-world environment before scaling our approach to a motion capture dataset of two humans performing various tasks.**

***Index Terms*—HRI, costmap, inverse reinforcement learning**

I. Introduction

Developments to planning and control algorithms, combined with more sophisticated robots in smaller form-factors have opened the field to collaborative robots (cobots) [5] that are designed to work in close proximity with humans in a safe manner. However, while humans naturally have a notion of the "personal space" of other entities around them, cobots still have difficulty inferring the minimum safe distance to humans when motion planning [1]. While the workspace of a robot [6] can be rigorously defined, the minimum safe distance, or "safe space" a human requires to feel comfortable can vary from person to person, and even from task to task.

For example, consider the scenario where a robot and human work together in a kitchen. If the robot is holding a potentially dangerous object like a knife, it must afford more caution around the human than if it were holding a bowl. In order to successfully function as a general-purpose kitchen worker, the robot must have an understanding of humans' safe space that is a function of the current task it is executing.

In the context of motion planning, one convenient method to formalize a human's safe space is through costmaps. A costmap is a defined region in a robot's task space that represents an area the robot should avoid. By attaching a costmap to the human, we can provide a tunable, soft constraint to a motion planner representing the human's safe space.

To learn a task-based costmap that represents a human's safe space, we propose an approach that can leverage existing or cheaply available data. A natural approach to this problem would be to perform controlled experiments with a human and robot performing a collaborative task. However, such an approach will not scale to a potentially large corpus of tasks the robot would be capable of performing. Instead, we utilize a much more readily available source of information — motion capture data between two humans performing labelled tasks. In addition to not requiring a robot, we argue the increased proliferation of off-the-shelf motion capture systems allow these demonstrations to be performed much more cheaply at scale. Due to equipment and workplace limitations from the ongoing COVID-19 lockdown, we were unable to collect motion capture demonstrations. Instead, we opted to repurpose the NTU RGB+D motion-capture dataset of human-human interactions for our experiments [7], [8].

We formulate the problem as a Markov Decision Process (MDP) with an unknown reward function. Using techniques in Maximum Entropy Inverse Reinforcement Learning (ME-IRL), we first perform a feature transformation to the demonstration data before inferring a reward in feature space. Using this inferred reward, we then extract parameters for a human-centered costmap in task space.

Our contributions are as follows:

1) A scalable method to collect information on task-based personal space.
2) An algorithm for inferring a task-based costmap from a dataset of motion capture trajectories of two humans.

II. Background

The proliferation of cobots has led to increased interest in safe motion planning techniques in close proximity to humans, often referred to as human-aware motion planning. Kulic and Croft [9] proposed a strategy for planning safe paths for hand-off tasks by minimizing danger criterion — quantitative measures of the risk and severity of potential impacts between a robot and human. This approach requires the planner to take the configuration space and kinematics of the robot into consideration.

Mainprice, Sisbot, Jaillet, Cortés, Alami, and Siméon [1] developed a planner that generates collision-free paths in a cluttered environment with close proximity to a human. The planner utilized multiple costmaps that model a human's proximity and visual field as constraints in a randomized, cost-based path generation algorithm. However, while this approach provides a good end-to-end solution for motion planning near humans, the costmaps were manually specified by a domain expert.

Mainprice, Gharbi, Siméon, and Alami [10] build a more sophisticated set of constraints by modelling human mobility for handover tasks in a cluttered environment. Their planner optimizes for paths that minimize a set of HRI constraints, such as proximity, visibility, execution time, and human pose comfort. However, this approach requires the human to play an active role in the task execution. Our problem statement assumes the human to be uncontrolled and performing independent tasks.

In order for us to infer costmaps from human behaviors, we draw on techniques from Reinforcement Learning. Abeel and Ng [2] formulated Inverse Reinforcement Learning (IRL) as a method to infer an unknown reward function of a Markov Decision Process (MDP) from a dataset of trajectories performed by an expert. By constraining the reward function to be a linear combination of known features (a process called feature-expectation matching) they prove convergence to a reward function that is at least as optimal as the expert's.

However, Ziebart, Maas, Bagnell, and Dey [11] note that feature-expectation matching is an ill-posed problem that does not guarantee a globally optimal reward. They utilize the principle of maximum entropy as an additional constraint, to learn a reward function that is optimal with respect to the expert while minimizing additional modelling bias.

III. Problem Formulation

Given a human h and a robot r, our objective is to learn a human-centered costmap $C_t^h : \mathbb{R}^3 \mapsto \{0, 1\}$ for some task label $t \in T$ that maps a position $\mathbf{x} \in \mathbb{R}^3$ in task space to an indicator for whether the given position falls within the costmap region.

For our approach, we collect motion capture demonstrations of a pair of humans, with one acting as the human h and the other as the robot r. Demonstrations must be labelled such that r is performing task t. Differences in the second human's configuration space compared to the true robot can be ignored, as we only analyze trajectories in task space. The motion capture system must extract an estimate of key joint positions from both humans, often referred to as their skeletons [12]. From both skeletons, we select a set of u and v joints to represent their respective states, such that $s^h \in \mathbb{R}^{u\times 3}$ and $s^r \in \mathbb{R}^{v\times 3}$.

With this motion capture system, we collect n demonstrations for each task t, each of which we represent as a sequence of m states $\zeta_t^i = \{s_1^r, s_1^h, s_2^r, s_2^h, ..., s_m^r, s_m^h\}$. Our dataset is thus represented as $D_t = \{\zeta_t^1, \zeta_t^2, ..., \zeta_t^n\}$.

Before performing any reward inference on the trajectories, we must first transform our state space. This is primarily necessitated by ME-IRL, which operates on discrete state spaces. However, we also take this transformation step as an opportunity to perform feature extraction from the task space data — the details of which will be discussed in Section IV. This state transformation ϕ maps our joint human-robot state space $\mathbb{R}^{u\times 3} \times \mathbb{R}^{v\times 3}$ to a single, discrete feature space $\mathbb{Z}^k$. Functionally, this is notated as $\phi(s^r, s^h) = \bar{s}$. With slight abuse of notation, we describe a trajectory in feature space as $\phi(\zeta_t^i) = \{\phi(s_1^r, s_1^h), ..., \phi(s_m^r, s_m^h)\}$ and a dataset in feature space as $\phi(D_t) = \{\phi(\zeta_t^1), ..., \phi(\zeta_t^n)\}$.

Finally, we formalize our MDP with the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, where:

- $\mathcal{S}$ is the set of discrete states in feature space $\mathbb{Z}^k$ we map to using ϕ.
- $\mathcal{A}$ is the set of actions in feature space. An action $a \in \mathcal{A}$ is one of a finite set of actions that moves the agent from state $\bar{s}$ to a neighboring state $\bar{s}'$, which shares at least one edge or vertex with the original state. If the feature space was 2-dimensional, $|\mathcal{A}| = 9$. This definition allows us to calculate the actions taken in a trajectory $\phi(\zeta_t^i)$ from the delta between the current and subsequent state.
- The transition function $\mathcal{T}(\bar{s}, a, \bar{s}') = p(\bar{s}'|\bar{s}, a)$ represents the probability that performing action a in state $\bar{s}$ will lead to state $\bar{s}'$. Due to our formulation of the action space, we assume deterministic transitions.
- The reward function $\mathcal{R}(\bar{s})$ is the reward assigned to arriving at state $\bar{s}$. Note that the true reward is unknown, and must be estimated.

Given this MDP, we can infer an estimated reward function $\hat{\mathcal{R}}$ using ME-IRL. From this reward estimate, we can extract information about the costmap C_t^h through an inverse feature mapping function ψ. Details of this process will be provided in the following section.

IV. Algorithm

Our motion capture system provides a skeleton with 25 joints. To simplify our approach, we consider only the end effector joints of both r and h for our tasks, setting $u = v = 1$. Specifically, we choose the right hand as our end effector. We consider this a reasonable approximation, as their end effectors are observed to be the joints in closest contact with each other.

$$s^r = \mathbf{x}^r_{\text{end\_eff}} \in \mathbb{R}^3 \tag{1}$$

$$s^h = \mathbf{x}^h_{\text{end\_eff}} \in \mathbb{R}^3 \tag{2}$$

In order to fit a costmap from the estimated reward $\hat{\mathcal{R}}$, we must create a parameterization our costmap $C_t^h(s|\theta)$ such that it is fully described by the vector θ. For our experiments, we selected a spherical costmap centered at the the end effector of h, allowing us to parameterize our costmap simply as the sphere's radius.

For our feature mapping $\phi(s^r, s^h) = \bar{s}$, we selected $k = 2$ features — the L2-norm to the terminal state g, and the L2-norm to the end effector of h. These features are then rounded to the nearest multiple of a discretization constant z to give us our final feature mapping function. Training the IRL model on the feature space instead of task space provides the advantage of state reduction.

$$\phi(s^r, s^h) = \begin{bmatrix} \phi_0(s^r, s^h) \\ \phi_1(s^r, s^h) \end{bmatrix} = \begin{bmatrix} \text{ROUND}_z(\|s^r - g\|_2) \\ \text{ROUND}_z(\|s^r - s^h\|_2) \end{bmatrix} \tag{3}$$

Since the spatial extents of D_t are bounded, we may also bound the feature space. We notate the bounds for each feature as $[\phi_0^{min}, \phi_0^{max}]$ and $[\phi_1^{min}, \phi_1^{max}]$.

With our feature-mapped demonstrations $\phi(D_t)$ fully defined, we can run ME-IRL to learn a reward function $\hat{\mathcal{R}}$. Note this algorithm requires its own feature-mapping $\mathbf{f}$ to shape the learned reward function [11]. Since we have already transformed our trajectories to feature space, we specify an identity feature map for $\mathbf{f}$.

Finally, we extract the costmap parameters with $\psi(\hat{\mathcal{R}}) = \theta$. Thanks to our feature selection, we can infer the radius of the costmap as a sharp gradient in $\hat{\mathcal{R}}$ along the ϕ_1 axis. This gradient represents the boundary between a low-reward region (where the states are too close to the human, and intrude into C_t^h), and high-reward region (where the states are outside C_t^h). By examining this gradient for all possible values of ϕ_0 above a minimum threshold Th, we can determine the costmap boundary as the smallest ϕ_1 value. Thus, we specify our feature extraction function ψ as:

$$\psi(\hat{\mathcal{R}}) = \underset{\phi_1}{\operatorname{argmin}} \left(\frac{\partial \hat{\mathcal{R}}(\phi_0 = p)}{\partial \phi_1} - Th \right) \quad \forall p \in [\phi_0^{min}, \phi_0^{max}] \tag{4}$$

V. Experimental Results

A. Grid World Scenario

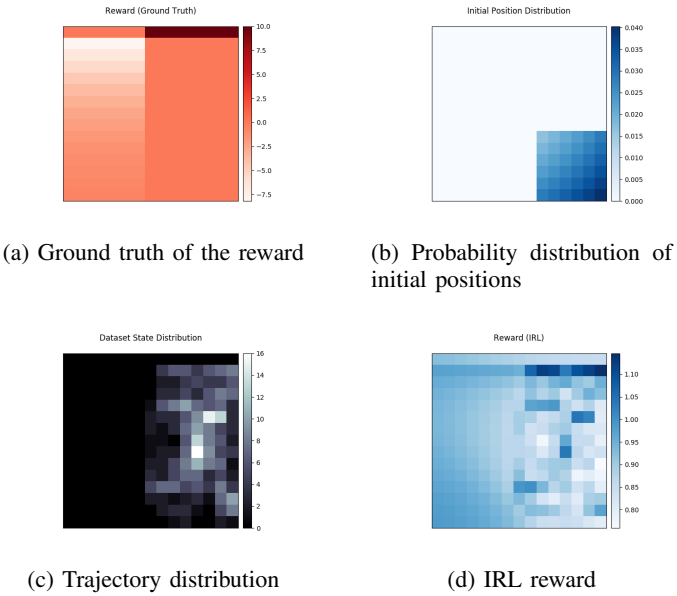
First of all, we test our algorithm in a simple grid world. The major difference between a traditional grid world scenario and our work lies in the definition of the action space, as the former usually moves to its neighboring grid every time step, while ours may move further in one step depending on the range of the features.

To verify our algorithm, we define the ground truth of our reward function as Figure 1a. Based on the hypothesis that the robot tries to avoid human while approaching the goal position, optimal trajectories should stay above a minimum ϕ_1 value and approach $\phi_0 = 0$. Next, we sample trajectories by utilizing value iteration on top of Figure 1a with the initial position sampling from distribution shown in 1b, as Figure 1c shows. Finally, after training ME-IRL model among these trajectories, we guarantee that our formulation of feature-space as well as action-space leads to a convergent reward function as shown in Figure 1d.

B. Motion Capture Trajectories

In this section, we apply our algorithm on the trajectories from human demonstrations captured by a RGBD camera, namely the NTU RGB+D Dataset. We target the actions that involves two people to complete, such as *Wielding a knife* and *Cheers and drink*, where one of them being viewed as the demonstrator and the other being the co-worker under the shared environment. For instance, in the *Wielding a knife* scenario, we concern the person holding the knife as the demonstrator that robot r should learn, and the goal is to avoid hurting the other person h while performing actions.

The data pipeline is shown in Figure 2. After obtaining the joint positions, we extract features mentioned in the previous section and train our IRL model to obtain the reward map of discretized features. The costmap is therefore generated as



(a) Ground truth of the reward

(b) Probability distribution of initial positions

(c) Trajectory distribution

(d) IRL reward

Fig. 1: Experimental results on the grid world scenario.

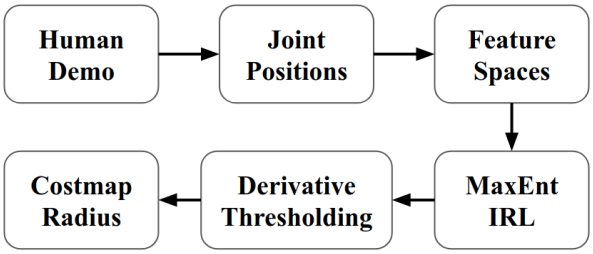


Fig. 2: Pipeline of applying algorithm to motion capture demonstrations.

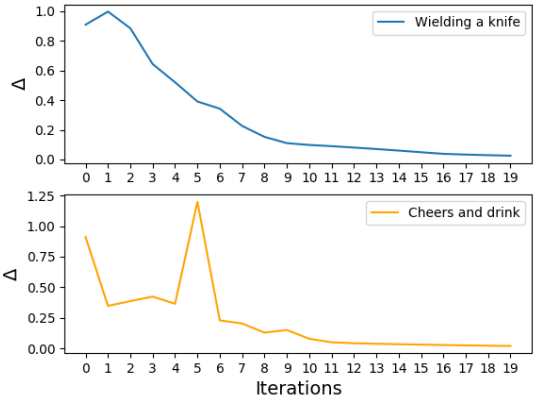
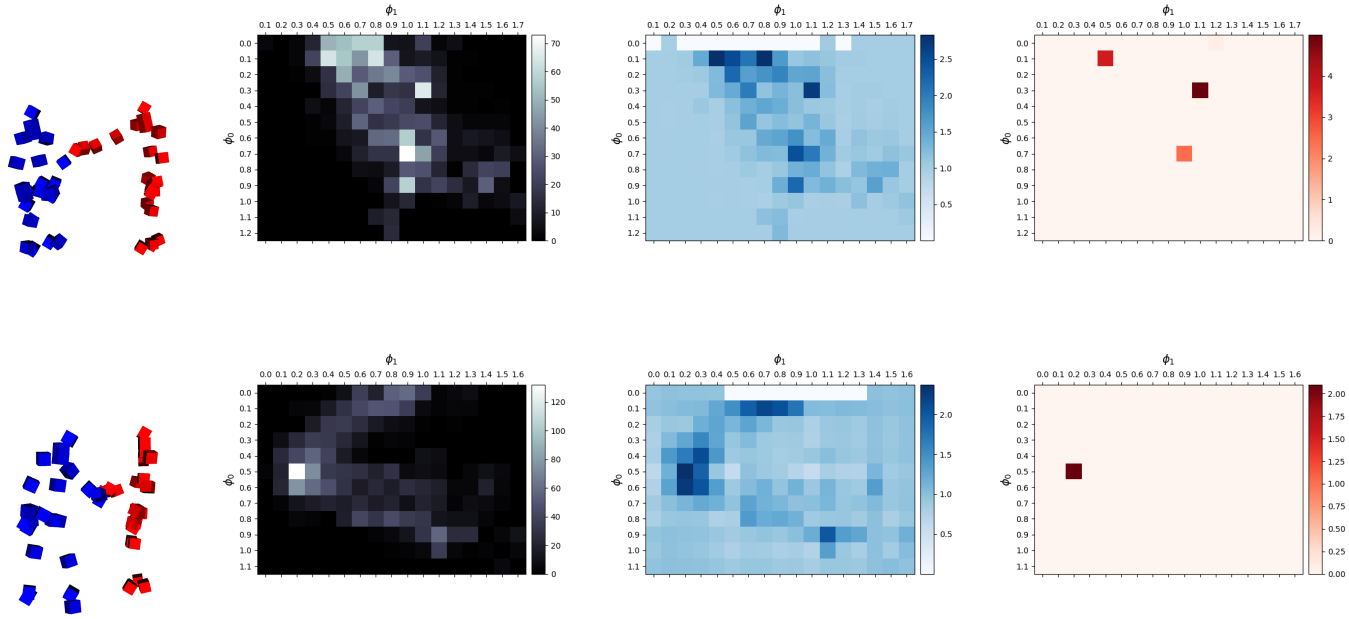


Fig. 3: Training curve of tasks *Wielding a knife* and *Cheers and drink*



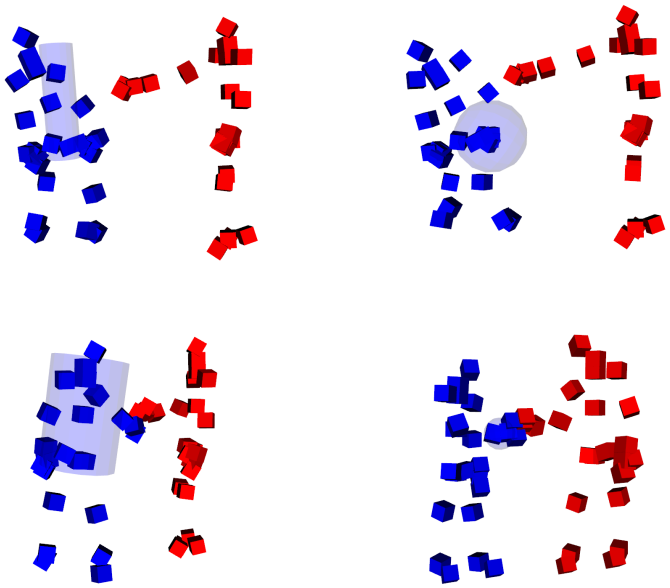
(a) Demonstrations (b) Trajectory distribution (c) IRL rewards $\hat{\mathcal{R}}$ (d) $\partial\hat{\mathcal{R}}/\partial\phi_1$ with threshold = 10

Fig. 4: Experimental results of different tasks. ME-IRL model is trained for 20 iterations under 50 trajectories for each task.

a sphere around the end effector of h with radius coming from the thresholding technique along partial derivative of the reward map along the distance between r and h, namely ϕ_1 (see Equation 4).

The training curves under 20 iterations using 50 trajectories for each task is shown in Figure 3, which shows the convergence of our model. Figure 4 shows the experimental results for two different tasks. As the trajectory distribution shown in Figure 4b, the distance between r and h remains large since human tends to avoid the knife. On the other hand, in *Cheers and drink* task, this distance becomes smaller as two people try to make a toast, and then increases again as the humans take a drink. The reward grid map shown in Figure 4c reveals similar distribution as our demonstrated trajectories, implying the convergence of our training process. Figure 4d is the results of partial derivative, where the radius being the smallest ϕ_1.

Accordingly, we can set 0.5 meter as the radius of the sphere surrounding the end effector of human for *Wielding a knife* task and 0.2 meter for *Cheers and drink* task respectively. Figure 5 shows the final costmap formulation. The baseline we choose is a cylinder-like costmap based on human's head, spin, and end effector. One of the drawbacks of this intuitive formulation is the heavy calculation on every frame, leading to some latency. On the contrary, our approach simply fixes the radius of the sphere on the end effector, which speed up during demonstrations. On top of that, our approach learns the radius among different tasks, which provides flexibility during close proximity human robot collaboration.



(a) Baseline (b) Ours

Fig. 5: Final costmap formulation comparing cylinder-like baseline and our approach. The top figures are the *Wielding a knife* task and the bottom ones are *Cheers and drink* task.

VI. Conclusion

In this paper, we propose a scalable method to collect trajectories of human on task-based personal space and come up with an IRL-based algorithm to infer the range of the costmap for human-robot collaborations under share environment. In addition, the feature-based state-action formulation in our work helps to reduce the size of the space during training procedure. The experiments show not only the convergence of our approach but also the efficiency of our final costmap

formulation. As for the future works, we aim to determine the cost value inside our costmap. Moreover, more costmaps can be created among different joints and links in order to cover the human body. Future work in the line of research will involve experimenting with motion planning techniques that take advantage of the learned costmaps.

The source code of our project can be viewed at https://github.com/r06921017/hri_costmap.git.

ACKNOWLEDGMENT

Thank you to Prof. Stefanos Nikolaidis for hosting the Computational Human-Robot Interaction course, for a great selection of papers, and for his advice and encouragement throughout the course.

REFERENCES

[1] J. Mainprice, E. Akin Sisbot, L. Jaillet, J. Cortés, R. Alami, and T. Siméon, "Planning human-aware motions using a sampling-based costmap planner," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 5012–5017, iSSN: 1050-4729.

[2] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, Jul. 2004, p. 1. [Online]. Available: https://doi.org/10.1145/1015330.1015430

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *1985 IEEE International Conference on Robotics and Automation Proceedings*, vol. 2, Mar. 1985, pp. 500–505.

[4] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 489–494, iSSN: 1050-4729.

[5] M. Peshkin, J. Colgate, W. Wannasuphoprasit, C. Moore, R. Gillespie, and P. Akella, "Cobot architecture," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, Aug. 2001, conference Name: IEEE Transactions on Robotics and Automation.

[6] Y. C. Tsai and A. H. Soni, "An Algorithm for the Workspace of a General n-R Robot," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 105, no. 1, pp. 52–57, Mar. 1983, publisher: American Society of Mechanical Engineers Digital Collection.

[7] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 1010–1019, 2016.

[8] J. Liu, A. Shahroudy, M. L. Perez, G. Wang, L.-Y. Duan, and A. Kot Chichung, "NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[9] D. Kulić and E. A. Croft, "Safe planning for human-robot interaction," *Journal of Robotic Systems*, vol. 22, no. 7, pp. 383–396, 2005, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20073. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20073

[10] J. Mainprice, M. Gharbi, T. Siméon, and R. Alami, "Sharing effort in planning human-robot handover tasks," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, Sep. 2012, pp. 764–770, iSSN: 1944-9437.

[11] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum Entropy Inverse Reinforcement Learning," *AAAI*, vol. 8, pp. 1433–1438, Jul. 2008.

[12] M.-C. Silaghi, R. Plänkers, R. Boulic, P. Fua, and D. Thalmann, "Local and Global Skeleton Fitting Techniques for Optical Motion Capture," in *Modelling and Motion Capture Techniques for Virtual Environments*, ser. Lecture Notes in Computer Science, N. Magnenat-Thalmann and D. Thalmann, Eds. Berlin, Heidelberg: Springer, 1998, pp. 26–40.