Task-Oriented Robot-to-Human Handovers in Collaborative Tool-Use Tasks

Meiying Qin[1] and Jake Brawer[1] and Brian Scassellati[1]

***Abstract*— Robot-to-Human handovers are common exercises in many robotics application domains. The requirements of handovers may vary across these different domains. In this paper, we first devised a taxonomy to organize the diverse and sometimes contradictory requirements. Among these, task-oriented handovers were not well-studied but important because the purpose of the handovers in human–robot collaboration (HRC) is not merely to pass an object from a robot to a human receiver, but to enable the human receiver to use it in a subsequent tool-use task. A successful task-oriented handover should incorporate task-related information – orienting the tool such that the human can grasp it in a way that is suitable for the task. We identified multiple difficulty levels of task-oriented handovers, and implemented a system to generate task-oriented handovers with novel tools on a physical robot. Unlike previous studies on task-oriented handovers, we trained the robot with tool-use demonstrations rather than handover demonstrations, since task-oriented handovers are dependent on the tool usages in the subsequent task. We demonstrated that our method can adapt to all difficulty levels of task-oriented handovers, including tasks that matched the typical usage of the tool (level I), tasks that required an improvised and unusual usage of the tool (level II), and tasks where the handover was adapted to the pose of a manipulandum (level III). We evaluated the generated handovers with online surveys. Participants rated our handovers to appear more comfortable for the human receiver and more appropriate for subsequent tasks when compared with typical handovers from prior work.**

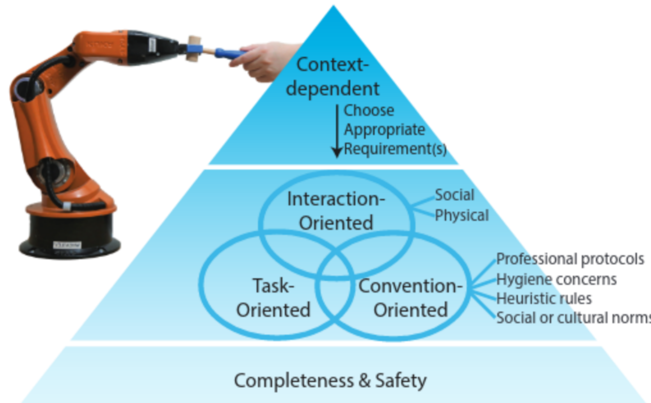

Fig. 1: Our taxonomy of robot-to-human handover requirements. Bottom to top: the basic, intermediate and advanced requirements.

I. INTRODUCTION AND RELATED WORKS

A robot-to-human handover is a joint action wherein a robot grasps, presents, and transfers an object held in its end-effector to a human receiver. It is a common exercise in numerous applications, including service robots handing flyers to pedestrians [1], personal assistive robots handing phones to people with disabilities [2], and factory robots handing hammers to collaborators [3]. To summarize the different requirements for handovers, we compiled a robot-to-humman handover taxonomy (for details, see Section I-A). The taxonomy serves the following purposes: 1) it helps to situate our study in the larger picture of robot-to-human handovers; 2) it helps to organize related work on handovers; 3) it may serve as a guide for future systems designed for handovers in terms of what requirements may need to be considered.

This study focused on one specific handover, the *task-oriented handover* that is commonly seen in the context of human-robot collaboration (HRC). However, as mentioned in recent publications [4], [5], task-oriented handovers have not yet gained enough attention in robot manipulations. In HRC, the purpose of a task-oriented handover typically is not merely to pass an object to a human, but also to enable the human to use the object to complete tasks. In order to maximize efficiency, the task-oriented handover should allow the human receiver to initiate a subsequent task with minimum in-hand object adjustment. Consequently, handovers of this type are dependent on how the tools should be used. Previous studies on task-oriented handovers generally demonstrated handovers of certain tools, without providing information regarding how the tools are used in the subsequent tasks. As a result, robots' lack of understanding of tool-use impedes their ability to generate handovers with novel tools. Therefore, our study aimed at designing a system that can generate appropriate task-oriented handovers with demonstrations of tool-use rather than handovers by integrating existing techniques. Furthermore, we also identified multiple levels of difficulties in task-oriented handovers and organized related work accordingly (for details, see Section I-B).

We built a system that generates task-oriented handovers. The system learned tool-affordances to allow the robot to understand the nature of the subsequent task. In our system, we chose and integrated a tool-affordance learning technique appropriate for handover tasks. We implemented the system on a physical robot and the results showed that the system can handle all difficulty levels of task-oriented handovers. We also conducted an online survey to evaluate the handovers executed by the robot. In summary, our contributions are:

[1]The authors are with Department of Computer Science, Yale University, New Haven, CT, United States meiying.qin@yale.edu, or meiyingqin@gmail.com

1) We defined a taxonomy of handover requirements.
2) Our system generated handovers based on learned tool affordances, rather than handover demonstrations, since task-oriented handovers are dependent on the subsequent tool-use task.
3) With the understanding of how tools should be used, our system was able to handle task-oriented handovers for all three difficulty levels that we identified.
4) Survey participants preferred our handovers and rated them as appearing to be comfortable for a potential human receiver and appropriate for the subsequent tasks.

A. *Taxonomy: Handover Requirements*

A handover is a complex manipulation with various requirements to satisfy. Therefore, we compiled a taxonomy of handover requirements and summarized it in Fig. 1. The requirements at the lower levels should be satisfied first before a higher-level requirement can be satisfied. In the taxonomy, the basic requirement is to be complete and safe. A complete handover refers to the successful delivery of an object to the receiver [6], [7], [8], [9], [10], [11], [12], [13], and a safe handover requires that no collision occurs at any time during the course of delivery [14], [15], [16], [17]. This is the focus of most handover studies.

Beyond the basic requirement of completeness and safety, satisfying one or more intermediate requirements will produce appropriate handovers. Compared with the studies focused on basic requirements, fewer handover studies focus on intermediate requirements.

The first intermediate requirement is that handovers should adapt to social or physical interactions between a human receiver and a robot (i.e., interaction-oriented). The social interactions include sending or perceiving various types of social signals such as eye contact [18], [19], [20], [21], while the physical interactions involve adjusting where [22], [23] or when [24] to conduct handovers based on the location or the physical state (e.g., availability) of a human receiver, or generating handovers that comply with human ergonomic needs [25], [26]. Satisfying these interaction–oriented requirements can help with generating customized handovers that are more comfortable for the receiver.

The second intermediate requirement is that a handover should abide by various conventions (i.e., convention-oriented), including professional protocols (e.g., handing over a surgical tool to a surgeon during a procedure in the operating room), hygiene concerns (e.g., one should not grasp the tines of a fork which will touch food), heuristic rules (e.g., one tend to orient an object horizontally for the receiver), and social or cultural norms (e.g., handing over a gift with a single hand is considered disrespectful). Satisfying convention-oriented requirements can help with generating handovers that match expectations.

The third intermediate requirement is that handovers should incorporate information about subsequent tasks (i.e., task-oriented) [27], [28], [5], [29], which allows the human receiver to perform the subsequent tasks more efficiently.

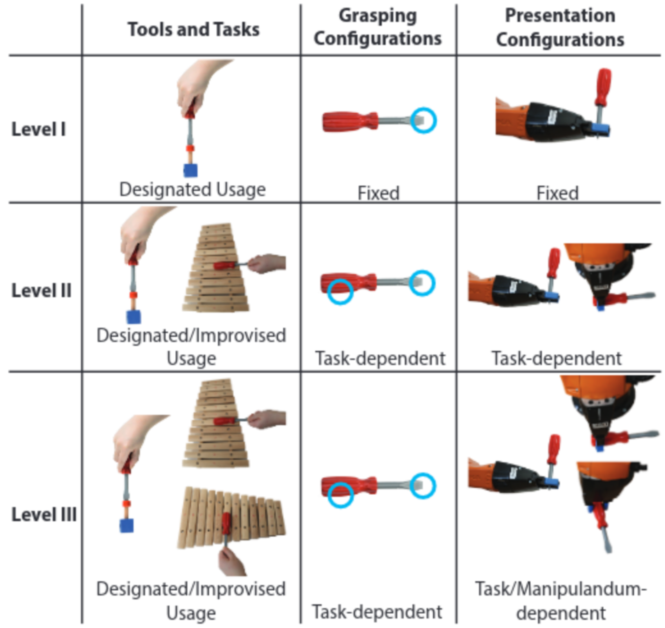

Fig. 2: Difficulty levels of task-oriented handover. Blue circles indicate function parts for different tasks.

Our study focus on this third intermediate requirement, task-oriented handovers, and other requirements are beyond the scope of this study.

The advanced requirement in our taxonomy is that a handover should be context-dependent. In other words, one should choose one or a combination of intermediate requirements to meet based on the specific context. The intermediate requirements may contradict each other, and not all requirements can be satisfied simultaneously. For example, during a convocation, an assistant hands the diploma to a dean in a way that prioritizes the interaction-oriented requirements so that the dean can receive the diploma more comfortably. However, when the dean hands the diploma to a graduate, the dean will not prioritize the interaction-oriented requirements as the assistant does, but will prioritize the convention-oriented requirements and use both hands to show respect. Therefore, a robot needs to recognize which intermediate requirements are important in the given context and choose one or a combination of intermediate requirements to meet the given context.

B. *Task-oriented Handovers*

As the objects to be handed over in task-oriented handovers are usually *tools*, we consider task-oriented handovers in the context of tool-use, and the object manipulated by a tool is referred to as *manipulandum* in this paper.

We identified three levels of difficulties in task-oriented handovers and organized related work on task-oriented handovers accordingly. Fig. 2 summarizes the difficulty levels and shows examples of each level. Level I is to properly hand over a tool to a human to perform a task typically matched with the tool (e.g., using a screwdriver to drive screws). Since a tool usually has a default usage, level I handovers could be achieved by building or learning a dataset to store handovers [27], [28], [5], and the dataset was learned with handover demonstrations rather than tool-use demonstrations.

In level II task-oriented handovers, a human receiver may use tools with their default usages, but may also improvise tool-use for tasks not generally associated with the tools (e.g., using a screwdriver to play a xylophone rather than to drive a screw). It is more challenging than level I because a pre-built dataset that can handle level I handovers may not be able to handle level II handovers due to the nearly limitless ways any particular tool can be used in different tasks. More importantly, the dataset may not be able to generalize to level II handovers due to a lack of understanding of how the tools should be used. To realize handovers at this level, a robot should recognize the functional segment of the tool and understand the usage to determine the handovers. In other words, learning *tool affordance* is the key to achieving level II task-oriented handover. To our knowledge, only one previous study considered learning tool affordances before performing handovers [29]. Although only level I handovers were demonstrated, their system may be capable of level II handovers. However, the design of this previous study makes their system impractical to be applied in many HRC scenarios. In this previous study, a human needed to demonstrate the usage of the novel tool to the robot in order to determine relevant handover configurations. However, a novel tool to be handed over is generally out of reach of the human receiver, so that a demonstration may be impossible without handing over the tool in the first place.

In addition to level II handover constraints, a robot should adjust the handover configurations based on the pose of the manipulandum (i.e., level III handovers). While some tasks impose consistent orientations irrespective of the tool used (e.g., stirring a pot of broth always requires a vertical tool orientation), the usages of tools in other tasks depend on the pose of the manipulandum (e.g., using a screwdriver to drive a screw placed either vertically into a tabletop or horizontally into a wall). This imposes challenges for previous systems [29] because each task was bound with specific handover configurations. Therefore, tool affordance may need to be learned in a different way to allow level III task-oriented handovers.

Previous studies on tool affordance have learned tool-use in various ways. However, they may not be appropriate for task-oriented handovers. Tool affordances were learned as a distribution of the outcomes [30], [31] instead of the relationship between a movement of a tool and the corresponding status change of a manipulandum. With tool affordances learned in this manner, a robot cannot achieve level III handovers because the relation between specific usages and specific contexts is unknown. When the abovementioned relationship was learned in a previous study [32], it learned in a way that was specific to the learned tools, and it was unknown whether a robot could generalize the learned tools to novel tools. It would be tedious to learn to use every tool prior to handing it over. While parallel Self-Organizing Maps (SOMs) can help with handling novel tools, novel tools needed to share similar shapes with the training tools [33], imposing restrictions on what kinds of novel tools a robot could hand over. This problem was overcome by using a large training set [34], [35], [36], which may be impractical in time-sensitive scenarios to hand over tools.

II. DESIGN AND IMPLEMENTATION

In our system, a robot first learned tool affordances or how to use a tool. Then in a robot-to-human handover task, the handovers were calculated based on how a tool should be used in subsequent tasks, and were then passed on to standard inverse kinematics and motion planning libraries to execute the motion. The tools may even be novel such that the robot never observed their usages in the required task. In this case, the robot first inferred its usage based on how the tools were used in the same task, and then generated corresponding handovers.

A. Object Model Generation

Preliminary 3D models of the objects were scanned by the robot if possible. A script that utilized MeshLab[1] was used to automatically process the 3D models to smooth, upsample, recenter, and resurface the point clouds into triangular meshes. The 3D models of the tools were then segmented using the shape diameter function (CDF). The objects that could not be scanned by the robot were obtained manually with Autodesk Recap Pro[2]. Detailed procedures for obtaining 3D models can be found in our previous work [37].

B. Vision Module

To obtain the pose of an object in the scene, a partial point cloud of the object needs to be extracted from the environment. To isolate the partial point cloud, a background point cloud without the object and a foreground point cloud with the object was captured from a depth sensor. Both point clouds were processed to leave only the workspace, and the desktop was removed with random sample consensus (RANSAC). The partial point cloud of the object was obtained by subtracting the processed background point cloud from the processed foreground point cloud.

After obtaining the partial point cloud of the object, the pose of the object was retrieved by registering the partial point cloud with the triangular mesh of the object using a modified Iterative Closest Point registration (ICP) algorithm. In this study, the pose of an object was represented with a 4×4 homogeneous transformation matrix $T \in SE(3)$ (superscript: reference frame, subscript: object), and $SE(3)$ represents the special Euclidean group:

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$

where R is a 3×3 rotation matrix representing the orientation, and p is a vector representing the position. The pose of the tool $T^{world}_{tool\_on\_desk}$ and the manipulandum $T^{world}_{manipulandum}$ in the world frame were perceived when they were placed on the desktop.

[1]MeshLab: https://www.meshlab.net/

[2]Autodesk software: https://www.autodesk.com/