Implicit Projection: Improving Team Situation Awareness for Tacit Human-Robot Interaction via Virtual Shadows

Andrew Boateng, Wenlong Zhang and Yu Zhang

Abstract—Fluent teaming is characterized by tacit interaction without explicit communication. Such interaction requires team situation awareness (TSA) to facilitate. However, existing approaches often rely on explicit communication (such as visual projection) to support TSA, resulting in a paradox. In this paper, we consider implicit projection (IP) to improve TSA for tacit human-robot interaction. IP minimizes interruption and can thus reduce the cognitive demand to maintain TSA in teaming. We introduce a novel process for achieving IP via virtual shadows (referred to as IPS). We compare our method with two baselines that use explicit projection to maintain TSA. Results via human factors studies demonstrate that IPS supports better TSA and significantly improves unsolicited human responsiveness to robots, a key feature of fluent teaming. Participants acknowledged robots implementing IPS more favorable as a teammate. Simultaneously, our results also demonstrate that IPS is comparable to, and sometimes better than, the best-performing baselines on information accuracy.

I. INTRODUCTION

Over the past decade, there have been accelerated growths and advancements in robotic research, making it no longer far-fetched to envision robots as part of our lives. One of the most appealing applications involves teaming domains where humans and robots complement each other to achieve complex tasks [1]. Teaming often requires coordination that may be facilitated by either explicit or implicit communication. While explicit communication (such as using natural languages) is highly effective at conveying information, it requires substantial attention from the receiver, leading to interruptions and thus less fluent teaming. Consequently, effective teaming is often characterized by tacit interaction with little or no explicit communication [2]. To facilitate tacit interaction, it is critical for the team members to maintain team situation awareness (TSA) so that each member can separately maintain and predict the team status for fluent teaming [3]. Paradoxically, the existing interface for human-robot interaction (HRI) often relies on explicit communication to maintain TSA [4], [5], [6], which introduces interruptions to teaming in the first place!

The challenge of maintaining TSA with implicit (non-explicit) communication has been left mostly unattended. In our work, we take a generic stance and refer to explicit communication, regardless of its modality, as communication with an established channel [7], which implies that the intentions of the sender to convey information and of the receiver to expect information should both be present. In implicit communication, in contrast, the receiver plays a passive role such that less attention would be drawn.

Consider a scenario in a semi-automated car assembly shop where a human worker, Mark, works along with a partner robot. Each agent has its own tasks in hand but must also collaborate occasionally to make progress. In one scenario, Mark sends the robot to fetch a hot soldering rod. Mark needs to attend to the rod soon after the robot arrives so must keep track of where, when, and how the robot returns to timely and safely handle the rod. For productivity, Mark would context switch to other tasks (e.g., reading the assembly manual) before the robot returns instead of idly waiting for the rod. However, focusing on other tasks can potentially cause him to lose track of the moving robot, resulting in the loss of TSA (i.e., from which direction the robot is approaching and the distance between the robot and Mark in real-time) and safety risks (i.e., thermal burns). Using explicit communication to address the situation may be the first thought. However, having the robot display its position via explicit visual communication, such as using a virtual map on a computer or portable device, would be distracting as it requires Mark to frequently check the map, while having the robot announce its arrival via prompts or sounds would be insufficient for the continuous TSA maintenance requirement. In such cases, we would benefit from an approach that uses implicit communication to enable Mark to maintain continuous TSA and timely responsiveness to the robot without diverting much attention.

However, coming up with an implicit communication interface that can support and positively impact the maintenance of TSA to address such a gap is challenging. Since information needs to be continuously conveyed to the receiver (albeit not necessarily perceived as communication), the interface must be minimally intrusive to the receiver. In this work, we introduce implicit projection (IP) via virtual shadows (IPS). First, IPS is an implicit visual communication method since shadows are not normally perceived by their viewers as a form of communication: they are simply part of the natural environment. This also implies that they are minimally intrusive. In addition, shadows can provide rich real-time information about objects, which makes them ideal for implicit communication. Last but not least, given our familiarity with shadows, they can be instantaneously interpreted [9]. In the scenario above, we can apply IPS by projecting a virtual shadow of the robot into the view of Mark to help him monitor the robot in real-time and maintain...
Our contribution in this work is four-fold. First, we identify an important gap of using implicit communication in visual communication. In particular, we introduce implicit projection (IP), propose a way to implement IP using virtual shadows (IPS), and apply IPS to an HRI scenario for maintaining TSA to achieve tacit interaction. This is in contrast to the traditional visual communication methods that predominantly rely on explicit communication [10], resulting in less effective teaming. Second, we develop a novel engineering process for generating realistic shadows with Microsoft Hololens, which includes environment modeling and virtual shadow rendering. It generates shadows by superimposing a cutout model onto the real-world from a 3D scan of the environment. The result is high-fidelity shadows generated according to the environment layout as with real shadows. See Fig. 1 for an illustration of virtual shadows generated by such a process. Third, we propose a method for achieving realistic shadow dynamics under changing light source configurations by integrating a mapping mechanism with a control method. The virtual shadows resulting from such a method are approximately true to the real object's dynamics while retaining smoothness. More intuitively, the shadow mapping mechanism makes sure that the robot's state (such as its velocity and rotational speed) is effectively captured by the virtual shadow and the control method ensures smoothness to minimize the negative effects on perception during interaction. Fourth, we integrate IPS with a physical robotic system and evaluate it by comparing with baselines that use explicit projection. Results validate our hypotheses and demonstrate the effectiveness of IPS in facilitating tacit interaction in proximal HRI scenarios.

II. RELATED WORK

In linguistics, explicit communication [12] is defined as information conveyed via spoken or written words. Where multiple modalities are concerned, explicit communication can be extended to refer to information transferred through an established channel [7]. Such a characterization implies that both the sender and receiver must be aware of the communication being made as information transfers through the channel: the intent of communication is mutual. We follow this extended definition in our work to distinguish between explicit and implicit communication, regardless of its modalities. Note that such a definition is also consistent with that commonly adopted in HRI research [13], [14] with a special focus on the receiver. For example, legible and explicable motions [15], [16], [17] are considered implicit ways of communication (via communicative behaviors) since the receiver may not realize the communicative intent of the sender. In this regard, IPS addresses the challenge of realizing implicit visual communication using virtual shadows, which is a first of its kind. With a properly designed interface, implicit communication draws less attention from the receiver who engages in a passive role during communication, making it less distracting [18] and more desirable for maintaining TSA in teaming.

As an emerging human-computer interaction (HCI) interface, Augmented Reality (AR) empowers us to visually perceive and interact with objects that are not present in the physical world [19]. Due to the intuitive appeal of such visual augmentations, AR has been used successfully and gaining popularity in various domains [20], such as military [21], marketing [22], education [23], manufacturing [24], medical [25], entertainment [26], and robotics [27]. Most prior
work (e.g., [28], [10], [29]) uses AR as an explicit visual communication method. Our work on implicit projection thus bridges an important gap in visual communication. Prior work identified that environment complexity, such as occlusions from physical objects, affects the fidelity of the virtual depiction of AR objects [30]. IPS presents a process for realistic and naturalistic virtual shadows of high-fidelity by generating them directly on the true environment layout.

AR has been used in numerous applications. Our interest aligns with those that relate to content creation [31], such as making VR objects more realistic and interactable. For example, Wang et al. [32] make use of lighting and shading of real scenes to modify AR objects to make them more lifelike. Such AR applications in robotics are also gaining popularity. For example, AR objects have been used as part of the interface to facilitate human-robot interaction [33]. AR has also been used to project virtual shapes onto a physical object to highlight the desired places to insert parts for assembly tasks [10]. Makris et al. [29] develop a method to project the trajectory of the effector and other information of a robot onto wearable devices before the robot starts moving. In [30], [34], the authors introduce an AR application that assists users having little to no knowledge in robotic systems with programming robot motions and recovering from failures. To the best of our knowledge, no prior work studied the application of AR for maintaining continuous TSA in challenging proximal HRI scenarios where the robot may frequently move out of sight. Since our work is focused on visual communication, other modalities, such as sounds and haptics [35], [36], are out of the scope.

III. APPROACH

To concretize the technical discussion, we consider a proximal HRI scenario as illustrated in Fig. 2 where a robot works behind a human teammate and must occasionally interact with the human, similar to the motivating scenario. For timely responsiveness, it is critical for the human to continuously monitor the robot to maintain real-time TSA to facilitate the interaction (i.e., responding to the robot’s arrival). To simplify the technical development, we assume that the human would not need to change his viewing direction during the task. Also, we assume that the robot would always operate behind the human (i.e., outside the human’s field of view). Such a situation may occur when the human must context switch to reading from a computer screen while not responding to the robot. The relaxation of these assumptions is discussed in Section VI. To help the human maintain real-time TSA in such scenarios, IPS projects a virtual shadow of the robot that is always observable to the human.

A. Shadow Mapping

One of the challenges to rendering the virtual shadow always observable is that the Hololens has a small field of view (FOV) of 34° with a maximum distance of 5m from the human to the holograms (i.e., AR objects). To achieve this, we use Shadow Mapping to project the robot’s position from outside the human’s FOV in the real-world to its desired shadow position in the virtual world within the FOV of Hololens. Furthermore, to ensure that the shadow is informative about the robot’s status, we would like the dynamics of the shadow to effectively capture the dynamics of the robot. Intuitively, when the robot moves faster (slower), the shadow should also move faster (slower); for sufficiently small enough position updates, such as when the robot moves left, right, up, or down, the shadow should also move likewise. To satisfy these requirements while ensuring the shadow is always visible, we choose to implement a linear mapping between the robot’s position and the shadow’s position in their respective polar coordinate systems. First, we consider the real-world outside the human’s FOV to be a semi-circular area (i.e., $\theta_w = 180^\circ$) with a pre-defined radius $l_w$ (i.e., the maximum distance from the human to the robot where the robot’s status is of concern to the human), and the virtual world as a sector with apex angle $\theta_v = 34^\circ$ and $l_v = 5m$. The mapping is specified as follows:

$$r_v = l_v - r_w \frac{l_v}{l_w}, \quad \beta_v = \beta_w \frac{\theta_v}{\theta_w},$$

where $r_w$ and $\beta_w$ above refer to the polar coordinates of a point in the real-world, and $r_v$ and $\beta_v$ refer to the polar coordinates of the corresponding point in the virtual world. Note that $r_w$ is a decreasing function of $r_w$ since the proportion of visible shadow should grow as the robot moves closer to the human. Such a mapping is illustrated in Fig. 3 where the human is at the intersection of the two worlds illustrated as a green dot. We also introduce the global coordinate system as a Cartesian system (i.e., $\varphi_x$ and $\varphi_y$).

B. Shadow Projection

The 3D development platform (Unity) for Hololens uses a depth buffer system to keep track of all surfaces close to the light source. If any surface comes in direct line with the light source, the surface will be illuminated (similar to ray tracing). The unilluminated surface therefore creates the shadow effect [37]. The benefit of using such a process is so that the shadow generated will be realistic as it naturally caters to the virtual surface onto which the shadow is projected (see Fig. 6). This means that we will only need to focus on projecting the shadow to the desired shadow position ($P_{d}$) without having to worry about the geometry of the virtual environment model, as long as the model is an
We choose to apply a PID control method that is often used in robotics to generate smoother state transition processes [38]. It is a combination of Proportional (P), Integral (I), and Derivative (D) control actions. P is proportional to the error between a set point and the observed process variable. I considers the past errors and integrates them over time to correct the accumulated error. D acts on temporal error difference. The control function of PID is given by:

$$u(k) = K_p e(k) + K_i \sum_{\tau=0}^{k} e(\tau) + K_d (e(k) - e(k-1))$$

(3)

where $K_p$, $K_i$ and $K_d$ are the coefficient $2 \times 2$ matrices of P, I and D, respectively. For shadow smoothing, with changes to the light source angles (i.e., $u$) as our control inputs and given the robot’s position in the real-world (i.e., $P_r$), we must drive the shadow towards the desired output $P_d$. Such a model can be modeled with the plant as follows:

$$x(k+1) = f(x(k), u(k), \Delta P_r(k))$$

(4)

where $x$ is the virtual shadow position, and $u = [\Delta \alpha, \Delta \gamma]^T$ encodes changes to the tilt and pan angles of the light source that we are actively controlling. $\Delta P_r$ is the change in the robot’s position in the real-world, which is treated as an exogenous input. In this paper, we consider $f$ as a first-order discrete-time dynamic model:

$$x(k+1) = x(k) + \begin{bmatrix} -a & 0 \\ 0 & b \end{bmatrix} u(k) + \begin{bmatrix} -h & 0 \\ 0 & g \end{bmatrix} \Delta P_r(k)$$

(5)

where $a$, $b$, $h$, and $g$ are positive constants. These values are chosen to capture $\Delta P_r(k)$ and $u(k)$’s expected relationship with the change of the shadow position from step $k$ to $k+1$. We assume that $\Delta P_r(k)$ is expressed in the polar coordinate system of the real-world.

Now, we can derive a simple PID controller using Eq. (4) with the setpoint at step $k$ being $P_d(k)$. This is the position we would like the shadow to be rendered. $x(k)$ represents the shadow position actually rendered at step $k$. We assume that both $x(k)$ and $P_d(k)$ are expressed in the polar coordinate system of the virtual world. The difference between $P_d(k)$ and $x(k)$ then leads to the error $e(k) = P_d(k) - x(k)$.

D. Shadow Rendering

To generate realistic shadows in IPS, shadow rendering is composed of environment modeling, shadow generation, and shadow superimposition. Our environment modeling technique uses the semi-autonomous nature of SLAM-like modeling (provided by HoloLens). To be able to find anchoring surfaces in the real world to place virtual objects (Holograms), the HoloLens constantly maps its environment. This also ensures that when there is a change in the environment (e.g., when an object is moved in the environment), it will be updated to the new arrangement.

In order to make use of the 3D map created by the HoloLens, we use vertex-lighting technique to create a custom shadow-receiving shader. Although pixel lighting provides more details by calculating the illumination for each pixel, it is computationally expensive. In contrast, by using
We apply this shader to the exported HoloLens-generated map (see an example in Fig. 5) and enable its shadow receiving properties. This creates our transparent shadow-receiving model of the environment (see an example in Fig. 6 for the environment model in Fig. 5). Finally, this model is superimposed onto the real-world to render the shadow.

Fig. 5: 3D scan by Hololens for environment modeling.

**IV. EXPERIMENTAL DESIGN**

We compare IPS against two baselines that use explicit visual communication with AR. We carefully selected the baselines based on common practices used for displaying dynamic objects [39], [40], [41]. All baselines and IPS display a sufficient amount of information about the robot for the tasks considered. The information is projected continuously to the human teammate. IPS and the baselines are described in more details below:

1) **Map** displays a top-down and real-time view of the robot on the map (Fig. 7 (left)). The pink sphere indicates the human and the robot is shown in black, with the arms in the front.
2) **Arrow** uses an arrow that always points to the robot in real-time (Fig. 7 (right)). The arrow pans in a plane.
3) **IPS** uses a virtual shadow of the robot to communicate real-time information (Fig. 7 (middle)). A video that illustrates how IPS works is included.

Both Map and Arrow appear frequently in real-world applications. Due to our familiarity with these baselines, they are chosen to best represent explicit projection methods. Our experimental design is used to verify these hypotheses:

- **H1.** IPS improves responsiveness to the robot compared to the baselines while remaining comparable to the baselines in terms of cognitive workload.
- **H2.** IPS maintains accurate TSA that is comparable to the baselines.
- **H3.** Robot with IPS is viewed more favorably as a work partner than the baselines.

We deployed all three methods onto Hololens and placed the robot outside the participant’s field of view (FOV). Each participant was informed about the delivery tasks and the robotic partner. The participant and the robot were supposed to complete the delivery tasks together in the least amount of time so that the participant knew that timely responses to the robot after its arrival were important. A response was recorded when the participant acknowledged the robot’s arrival (for delivery service or to deliver). The time elapsed since the robot’s arrival was used to quantify the responsiveness to the robot (i.e., the less the better). Note that responsiveness heavily depends on the quality of TSA maintenance. At the same time, the participant was given some document to read and was advised to not turn to observe the robot during the study while wearing Hololens. After the delivery tasks, the participant was then presented with several estimation tasks to evaluate the accuracy of TSA. To prepare for the tasks, each participant was given a printed map of the space in a discretized form for position identification (see Fig. 8). The map has numbers on it indicating different parts of the space. During the estimation tasks, questions such as the robot’s current position and predicted destination were asked at certain time points as the robot moved around in the space in preprogrammed paths. The participant was tasked to choose the number on the map to match the estimation or prediction. The questions were divided into static perception (estimating the robot’s current position), post-movement perception (estimating the
robot’s position after movement), and prediction (predicting the destination of the robot). We computed accuracy as the Manhattan distance between the participant’s estimation and the ground truth. The smaller the distance was, the more accurate the estimation was.

24 CS students in their senior year participated in a within-subjects study. They were made up of 14 female and 10 male students. Each participant participated in three sessions, one for each method. We recorded a video for each part of a session (with 2 parts in each session). The first video for the delivery tasks was used to measure the waiting time between the robot’s arrival and the participant’s response. The second video was recorded for the estimation tasks. This resulted in 6 videos per participant and a total of 144 videos recorded. However, due to objects and robot blocking the camera, or participants turning to observe the robot during the study, we discarded the data from 7 participants, which left us with the remaining 17 participants for result analyses.

After all the three sessions, each participant was given a final survey that included an AttrakDiff survey and the participant’s preferences towards the robot as a work partner and towards the different projection methods with respect to their naturalness and user friendliness. In the AttrakDiff survey, the participants were asked to rate the methods based on different qualitative metrics on a scale of 1 to 7.

V. RESULTS AND ANALYSES

An alpha level of 0.005 is used for all statistical tests.

1) Responsiveness to Robot: Fig. 9 presents the results for the waiting time of the robot between when it arrived at the delivery location and when the participant responded. We observe that participants were much faster to react to the robot when IPS was used. Paired Student’s t-tests gave $t(16)=2.12$, $p=.003$ between IPS ($M=10.3$, $SD=9.18$) and Map ($M=13.5$, $SD=7.9$), and $t(16)=2.12$, $p=.002$ between IPS ($M=10.3$, $SD=9.18$) and Arrow ($M=15.7$, $SD=8.8$). Fig. 10 presents the results for the waiting time of the robot between when it arrived at the pickup location on the participant’s request for a delivery and when the participant responded. Similar results were observed. T-tests resulted in $t(16) = 2.13$, $p = .003$ between IPS ($M=5.50$, $SD=2.11$) and Map ($M=8.80$, $SD=3.62$), and $t(16)=1.75$, $p < .001$ between IPS ($M=5.50$, $SD=2.11$) and Arrow ($M=13.04$, $SD=4.66$). These results verified that the responsiveness to robot with IPS was significantly better than the baselines, which supported part of $H1$ regarding responsiveness.

2) TSA Accuracy: Fig. 11 presents the results with respect to how accurately each method maintained TSA in the estimation tasks. Generally, IPS did better than Arrow and was on a par with Map. Map did well, which was likely due to the fact that participants were generally familiar with maps in one form or another in real life. Arrow performed the worst as expected since the depth information must be inferred from how fast the arrow moved, which made it difficult for the participants to accurately estimate the robot’s position and changes in position. It can be seen from Fig. 11 that IPS performed comparably to Map in perception tasks, with Map having a slight edge in post-movement perception. T-tests revealed no significant differences between IPS and Map ($H2$). It is however interesting to note that IPS proved to provide more context information for TSA in prediction.
and did much better there than the baselines in prediction as shown in Fig. 11. We attributed such performance to better TSA since prediction required the participants to maintain the context of movements (i.e., which direction the robot was heading for), instead of solely the position. Student’s paired t-tests on prediction resulted in \( t(16)=2.12, p < .001 \) between IPS (\( M =0.24, SD =0.42 \)) and Arrow (\( M =0.57, SD =1.08 \)), and \( t(16)=2.12, p < .001 \) between IPS and Map (\( M =0.32, SD =0.55 \)).

![Fig. 11: Accuracy (via a distance metric) in estimation tasks.](image)

![Fig. 12: AttrakDiff survey results.](image)

3) attractiveness: AttrakDiff evaluated the attractiveness of the robot with different methods. The result is presented in Fig. 12. Results indicate that the robot with IPS was viewed as more attractive than the robot with the baselines (H3). In particular, participants were much more motivated to work with the robot when using IPS than the baselines (i.e., Captivating and Novel). We interpreted it as the participants felt IPS provided a more immersive teaming experience than the baselines (see 4) below), thus encouraging them to be more responsive to the robot. Albeit being more novel, as a method of implicit communication, IPS drew no more attention than the baselines, which was reflected by the result showing that the participants considered IPS and Map comparable in manageability (part of H1). Given our frequent exposure to various forms of maps in real life, this result is encouraging. Overall, it is observed that IPS obtained the best ratings among almost all features. We averaged the values and ran a Student’s paired t-test. The results were \( t(16)=2.12, p < .001 \) between IPS (\( M =2.59, SD =0.84 \)) and Map (\( M =3.76, SD =0.60 \)), and \( t(16)=2.13, p < .001 \) between IPS (\( M =2.59, SD =0.84 \)) and Arrow (\( M =4.31, SD =0.79 \)).

4) Partnership and Naturalness: We explicitly asked the participants to indicate which of the three methods gave them a feeling of partnership. 75% of the participants indicated that IPS gave them the feeling that the robot was a work partner. 18.8% felt towards Map and only 6.2% towards Arrow. Fig. 13(a) shows the results. This result verifies that the robot with IPS was viewed more favorably as a work partner (H3). Finally, we asked the participants to indicate which methods they felt the most natural. 50% answered Map while the remaining 50% felt towards IPS. This was somewhat surprising since maps had been an integrated part of our lives while IPS was a novel human-robot interface to the participants. The choice of virtual shadows clearly helped IPS achieve a comparable performance here.

![Fig. 13: Results of the participants’ (a) feeling of the robot as a work partner and (b) feeling of naturalness of interaction.](image)

VI. CONCLUSIONS AND DISCUSSIONS

In this paper, we introduced implicit projection via virtual shadows (IPS) for tacit HRI. We showed that IPS improved TSA and responsiveness to robots in a proximal HRI scenario. We addressed the challenges in realizing IPS in four steps: shadow mapping, shadow projection, shadow smoothing, and shadow rendering. IPS represents the class of implicit visual communication methods, which contrast with prior visual communication methods that are predominantly explicit. Results also showed IPS incurring comparable cognitive and attention demands with the best explicit projection baselines studied. To the best of our knowledge, IPS is the first work that considers implicit visual communication and bridges an important gap in visual communication. IPS will have a variety of applications involving proximal HRI scenarios where tacit teaming is desired. An interesting extension of IPS is to project the future state of the robot, which can be used to proactively facilitate teaming activities [42].

In future work, we plan to gradually relax the assumptions we made. The first of these assumptions is that the human teammate would not change his viewing direction during the task. Although our approach is expected to work with slight changes in the viewing direction, for sudden and abrupt changes, the shadow could be thrown out of the field of view. Such a problem can be addressed by allowing the shadows to temporarily stay out of the view to avoid flickering and gradually reenter the view. We will explore different control methods to enable smooth transitions. Another assumption that the robot always stays behind can be relaxed in a similar way by enabling smooth transitions for the robot to move from behind the human to the front. Another limitation of IPS is that it requires the environment to be scanned and mapped. This could pose a problem with a frequently changing environment. We will address this in future work.