Toward Mobile Mixed-Reality Interaction With Multi-Robot Systems

Jared Alan Frank, Sai Prasanth Krishnamoorthy, and Vikram Kapila

***Abstract*—Although human-multi-robot systems have received increased attention in recent years, current implementations rely on structured environments and utilize specialized, research-grade hardware to operate. This letter presents approaches that leverage the visual and inertial sensing of mobile devices to address the estimation and control challenges of multi-robot systems that function in shared spaces with human operators such that both the mobile device camera and robots can move freely in the environment. It is shown that a subset of robots in the system can be used to maintain a reference frame that facilitates tracking and control of the remaining robots to perform tasks, such as object retrieval, using an operator's mobile device as the only sensing and computational platform in the system. To evaluate the performance of the proposed approaches, experiments are conducted in which a system of mobile robots is commanded to retrieve objects in an environment. Results show that, compared to using the visual data alone, integrating both the visual and inertial data from mobile devices yields improvements in performance, flexibility, and computational efficiency in implementing human-multi-robot systems.**

***Index Terms*—Distributed robot systems, telerobotics and teleoperation, and virtual reality and interfaces.**

I. Introduction

THE use of multi-robot systems has received increased attention due to its potential in applications that call for distributed sensing or coverage, e.g., exploration [1] and collection [2]. To perform tasks, multi-robot systems are often controlled using specialized, research-grade hardware such as sophisticated vision and computing systems. This approach allows precise global sensing of the robots and their environment, and the application of centralized techniques to the tracking and control of multi-robot systems. However, such implementations are usually expensive, complex, and unportable, and limited to structured indoor environments. Thus, although this approach is useful for conducting research, it often cannot be employed in the field. To operate multi-robot systems in real-world environments, robots are equipped with the necessary hardware and software for perceiving their environment and other robots, and for planning and executing actions. However, this approach increases the size, cost, and complexity of an individual robot, and it does not scale well to large teams of robots. Alternatively, ongoing efforts aim at improving the scalability and robustness of multi-robot systems by using individual robots with increasingly simple designs. Such implementations face a number of challenges, including lack of centralized control, limited computational resources, and strictly local sensing and communication between robots [3]. To address these limitations, swarm robotics research investigates ways to manage coordination and interaction between large numbers of simple robots to achieve complex collective behaviors [4].

While a goal of multi-robot applications is often the autonomous performance of a task, many applications rely on the interaction between operators and robots in shared spaces, such as homes [5], classrooms [6], and warehouses [7]. In such applications, operators benefit by having access to information about the robots, environment, and task, as well as a means to effectively interact with the robots. Solutions include the use of body gestures [8], [9], but these gestures are often unintuitive for operators who must be trained to interact with unfamiliar equipment. Alternatively, recent efforts have begun to explore the use of mobile devices, such as smartphones and tablets, for interacting with multi-robot systems. However, current implementations use structured environments [10] and sophisticated motion capture systems and computational platforms [11], [12] to track and control robots in the system and are thus not amenable to real-world environments.

Mobile mixed-reality interfaces have been proposed for interacting with laboratory test-beds [13] and robotic manipulators [14]. In these applications, the sensing, storage, computation, and communication capabilities of mobile platforms are leveraged to deliver economic, portable, and engaging operation of physical systems through visual feedback and interaction with augmented reality. However, to provide accurate feedback and effective interactions even as mobile devices are held and moved arbitrarily by operators, the applications of [13], [14] employ fiducial markers that do not move relative to the world. In this paper, a novel mobile mixed-reality approach is proposed that allows operators to interact with and control systems of mobile robots, endowed with fiducial markers, without the need for a structured environment or specialized, research-grade equipment (see Fig. 1). An interface runs on a mobile device held by an operator such that the device's rear-facing camera points towards the multi-robot system. To facilitate accurate tracking and control, rendering of interactive augmented graphics, and

Manuscript received February 15, 2017; accepted June 4, 2017. Date of publication June 9, 2017; date of current version June 26, 2017. This paper was recommended for publication by Associate Editor K. Kyriakopoulos and Editor N. Y. Chong upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under DRK-12 Grant DRL-1417769, RET Site Grant EEC-1542286, and ITEST Grant DRL-1614085 and in part by the NY Space Grant Consortium under Grant 48240-7887. *(Corresponding author: Vikram Kapila.)*

The authors are with the Department of Mechanical and Aerospace Engineering, NYU Tandon School of Engineering, Brooklyn, NY 11201 USA (e-mail: jared.alan@nyu.edu; saiprasanth@nyu.edu; vkapila@nyu.edu).

Digital Object Identifier 10.1109/LRA.2017.2714128

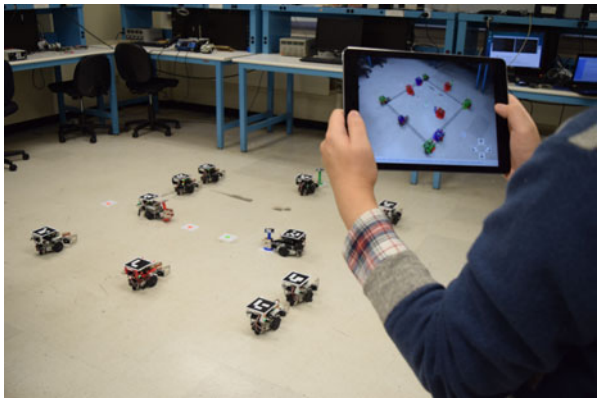

Fig. 1. This study explores human-multi-robot systems without structured environments or research-grade equipment.

mapping of user interactions to robot commands as the device is moved arbitrarily by the operator, a transformation between a coordinate frame attached to the device and a fixed reference frame is estimated using the sensing and computational capabilities of the device. With a visual marker attached to each robot in the system, we consider two distinct approaches to compute this transformation. In the first approach, a large number of robots are exploited by assigning a subset of the robots to start at known, fixed locations as the transformation is estimated using a standard vision technique based on point correspondences. In the second approach, the transformation is computed in two steps, one that uses estimates of device attitude from inertial data and the other that uses visual data of only one robot in the system, resulting in a reference frame that coincides with the local frame of that robot. For each approach, an algorithm is designed that allows the role of the robots to be switched between operating on objects and maintaining the reference frame as the operator's objectives or the camera's visual perspective change with time. By examining the performance, flexibility, and computational complexity of the proposed approaches, insights are obtained regarding the benefits and limitations associated with implementing human-multi-robot systems that leverage the technologies of mobile devices.

II. Methodology

The development of human-multi-robot systems must consider the estimation of robot states, mapping of user interactions to goal states, and control of robots to goal states. Typically, high resolution data from specialized, research-grade equipment is employed. Although conventional graphical applications have been developed for mobile platforms, a superior approach is to design mobile interfaces for human-robot systems by leveraging the sensing capabilities of the mobile device. Thus, two approaches are proposed that utilize the on-board sensors of mobile platforms to provide efficient and enhanced user interactions with multi-robot systems. In each approach, an operator holds a mobile device such that its rear-facing camera is pointed at a multi-robot system in a shared space from an arbitrary perspective. Robots are affixed with visual markers whose distinct patterns are inspired by Hamming codes [15] and are detected using an adaptive thresholding approach [16].

If a 3D point $P^{\mathrm{W}} = [X \;\; Y \;\; Z \;\; 1]^{\mathrm{T}}$ in a fixed world frame W is projected onto the image plane of a calibrated camera with frame C, the projected point $p^{\mathrm{C}} = [u \;\; v \;\; 1]^{\mathrm{T}}$ is obtained as

$$p^{\mathrm{C}} = \mathscr{T}_{\mathrm{W}}^{\mathrm{C}} \; P^{\mathrm{W}}, \quad \text{where} \quad \mathscr{T}_{\mathrm{W}}^{\mathrm{C}} \triangleq K \left[R_{\mathrm{W}}^{\mathrm{C}} \;\; \middle| \;\; t_{\mathrm{W}}^{\mathrm{C}} \right], \quad (1)$$

is a projective transformation consisting of a rotation matrix $R_{\mathrm{W}}^{\mathrm{C}}$, a translation vector $t_{\mathrm{W}}^{\mathrm{C}}$, and a camera matrix $K \in \mathbb{R}^{3\times3}$. Let points P^{W} be the center points of robot markers and assume that the robots are identical and drive in a plane (applications where robots operate on more complex surfaces are beyond the scope of this study). Then, W can be chosen such that $Z = 0$ for all P^{W}. Now $\mathscr{T}_{\mathrm{W}}^{\mathrm{C}}$, which represents the pose of the plane occupied by the robots relative to the camera pose, reduces to a 3×3 homography transformation denoted as $H_{\mathrm{W}}^{\mathrm{C}}$ [17]. To represent camera points p^{C} in frame W (with $Z = 0$) (i.e, as $p^{\mathrm{W}} = [X \;\; Y \;\; 1]^{\mathrm{T}}$), $H_{\mathrm{C}}^{\mathrm{W}} = (H_{\mathrm{W}}^{\mathrm{C}})^{-1}$ is of interest. For the case in which the camera is fixed, $H_{\mathrm{C}}^{\mathrm{W}}$ is computed using standard vision techniques that use measurements of the visual features p^{C} corresponding to known points p^{W}. Although standard vision techniques are used to estimate $H_{\mathrm{C}}^{\mathrm{W}}$ for the case in which the camera moves arbitrarily, they normally require that the points p^{W} be fixed in the scene. This section describes two approaches to estimate $H_{\mathrm{C}}^{\mathrm{W}}$ for the case in which the camera is free to be moved arbitrarily by an operator and the features are attached to robots that also move in the scene. The first approach, which is limited to the use of standard vision techniques, is presented for comparison with the second approach, which uses both the inertial and visual sensors of the mobile device.

A. Approach A: Vision Only

Since $H_{\mathrm{C}}^{\mathrm{W}}$ represents a homography transformation, it can be estimated provided that the locations of any four or more robots are known relative to the world frame W and the image locations of the corresponding robots' markers have been detected in the camera frame C [18]. This is because a homography matrix contains eight degrees of freedom and each point correspondence provides two equations from (1). Since there is uncertainty in the placement of the robots and uncertainty in the image locations (due to noise from the camera), a least squares estimate of $H_{\mathrm{C}}^{\mathrm{W}}$ is computed. We refer to the four or more robots responsible for estimating $H_{\mathrm{C}}^{\mathrm{W}}$ as *reference robots*. Since the mobile device's single camera cannot measure depth and is free to be moved to arbitrary perspectives, $H_{\mathrm{C}}^{\mathrm{W}}$ can be accurately recalculated provided that the reference robots are not moved. For further simplification, the origin of W is established at one of the reference robots (see Fig. 2). In addition to the reference robots, there are also *acting robots* in the environment that are identical to the reference robots but are permitted to move and manipulate objects of interest, which are also affixed with visual markers. Knowing the size of the markers affixed to the acting robots and manipulable objects, their relative poses can be estimated using four point correspondences [18] coming from the four corners of the markers. Once the relative poses $T_{\mathrm{R}_i}^{\mathrm{C}}$ and $T_{\mathrm{O}_j}^{\mathrm{C}}$ of all acting robots R_i and manipulable objects

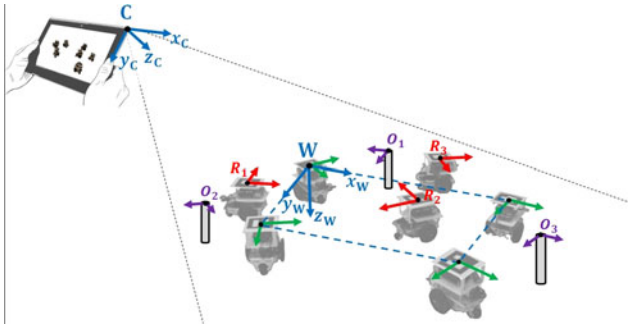

Fig. 2. Coordinate frames used by approach **A** for tracking, control, and interaction with robots.

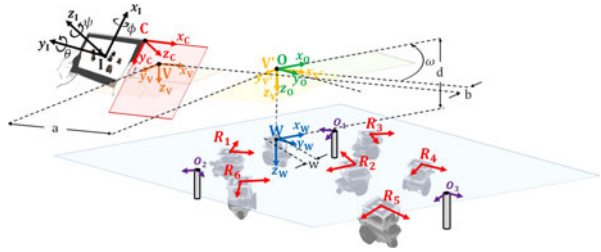

Fig. 3. Coordinate frames used by approach **B** for tracking, control, and interaction with robots.

O_j, respectively, have been estimated in the camera frame C, they are represented in W as $T^{\mathrm{W}}_{\mathrm{R}_i} = H^{\mathrm{W}}_{\mathrm{C}} T^{\mathrm{C}}_{\mathrm{R}_i}$ and $T^{\mathrm{W}}_{\mathrm{O}_j} = H^{\mathrm{W}}_{\mathrm{C}} T^{\mathrm{C}}_{\mathrm{O}_j}$ (see Fig. 2). Such a representation of relative poses allows the mobile device to be moved to arbitrary perspectives provided that the markers of robots R_i and objects O_j remain in view of the camera.

Since an estimate of the homography $H^{\mathrm{W}}_{\mathrm{C}}$ is computed using a least squares approach, its accuracy is expected to improve as the number of robots increases and as the robots are better distributed around the plane. For example, just as a line cannot be properly estimated by two coincident points, the plane of the robots cannot be properly estimated by robots positioned along a line. Moreover, the accuracy of robot and object poses transformed using $H^{\mathrm{W}}_{\mathrm{C}}$ is guaranteed only in the area covered by the robots and may diverge gradually for locations outside of the covered area. A key question is whether reference robots should consist of only robots with known poses or also robots whose poses are estimated. Since the poses of unknown robots are already estimated using the previously computed estimate of $H^{\mathrm{W}}_{\mathrm{C}}$, using their estimated poses in the computation of the next estimate of $H^{\mathrm{W}}_{\mathrm{C}}$ is not expected to improve the accuracy of the estimate. However, using the poses of estimated robots as references can help to maintain the covered area, allowing the roles of robots to be freely switched between maintaining the estimate of $H^{\mathrm{W}}_{\mathrm{C}}$ and acting on objects. Thus, an algorithm is developed for managing the assignment of reference robots such that there is a reversible exchange between reference robots r and acting robots a. According to the algorithm, all N robots in the workspace are used to maintain the world frame (i.e., as r reference robots), whether their poses are known or estimated, yielding $r = N$. When a certain number (a) of robots is tasked to move, the number of reference robots is temporarily reduced by that number, yielding $r = N - a$. If we denote the number of robots that have known initial poses by n and the number of robots whose initial poses are unknown by m, we have $N = n + m$ and $r = N - a = n + m - a$. Once acting robots have completed their assigned operation and have come to a stop, they are again added as reference robots and their estimated poses are again used in maintaining the world frame. In other words, a reduces back to zero and the number of reference robots is restored to $r = N$. Note that if a known robot is commanded, it leaves its known location and its pose is estimated for the duration of the task. Thus, choosing to use these estimated poses to maintain the world frame, the system's stability is largely dependent on the accuracy of the estimate of $H^{\mathrm{W}}_{\mathrm{C}}$.

B. Approach B: Inertial and Vision Data

Despite its potential, approach **A** is limited by the requirements that at least four robots start out as reference robots at known locations and the currently assigned reference robots remain fixed during an operation and are appropriately distributed around the workspace of the robots to estimate a sufficiently accurate $H^{\mathrm{W}}_{\mathrm{C}}$. These limitations arise because as both the camera and markers move in the scene, it is not known to what extent changes in the markers' image locations are due to the motion of the camera or due to the motion of the robots to which the markers are attached. To address this challenge, one may estimate the motions of robots using either odometric measurements or velocity commands issued by the mobile device. Unfortunately, maintaining a dynamic model of each robot in the system adds unnecessary cost and computational complexity that may not scale well with the number of robots and may not be feasible in real-time on a mobile processor. Alternatively, by estimating the motion of the camera using the mobile device's inertial sensors, the limitations of approach **A** can be resolved, assuming that the plane occupied by the robots is horizontal and thus orthogonal to the direction of gravity. The device's inertial measurement unit (IMU) can be used to obtain estimates of the device's attitude relative to a reference frame V, whose z-axis is aligned with the vertical and whose x and y axes are chosen arbitrarily according to device attitude at the start of the application (see Fig. 3). The estimates of device attitude can be computed from raw IMU measurements using various algorithms whose details are beyond the scope of this paper [19], [20]. Using these estimates from the mobile application development framework, $H^{\mathrm{W}}_{\mathrm{C}}$ is computed as

$$H^{\mathrm{W}}_{\mathrm{C}} = T^{\mathrm{W}}_{\mathrm{V}} H^{\mathrm{V}}_{\mathrm{C}}, \tag{2}$$

where $H^{\mathrm{V}}_{\mathrm{C}}$ is a homography transformation that is computed using the attitude estimates obtained from the device's inertial measurements and $T^{\mathrm{W}}_{\mathrm{V}}$ is a similarity transformation computed with image data extracted from only one robot marker using the device camera.

Imagine that a 3D point P^{W} in the fixed world frame W is captured by the device camera with matrix K and image plane C as well as by a virtual camera with the same matrix K but image plane V. Since V is parallel to W and differs from C by a pure

rotation, we evaluate (1) for each camera to yield an expression in terms of the rotation matrix $R_{\mathrm{C}}^{\mathrm{V}}$ for the homography matrix $H_{\mathrm{C}}^{\mathrm{V}}$ that transforms p^{C} to p^{V} [17]

$$p^{\mathrm{V}} = H_{\mathrm{C}}^{\mathrm{V}} p^{\mathrm{C}}, \quad H_{\mathrm{C}}^{\mathrm{V}} = K R_{\mathrm{C}}^{\mathrm{V}} K^{-1}. \tag{3}$$

Note that, in general, the IMU frame I is not aligned with the camera frame C (see Fig. 3). Thus, the orientation of C relative to I can be represented by $R_{\mathrm{C}}^{\mathrm{I}}$, which yields

$$H_{\mathrm{C}}^{\mathrm{V}} = K \left(R_{\mathrm{I}}^{\mathrm{V}} R_{\mathrm{C}}^{\mathrm{I}} \right) K^{-1}, \tag{4}$$

where $R_{\mathrm{I}}^{\mathrm{V}}$ is the orientation of I relative to V. Our algorithm obtains estimates of the orientation of V relative to I using the elementary rotations θ, ϕ, and ψ about the y_{V}, x'_{V}, and z''_{V} axes, successively (see Fig. 3). To compute $R_{\mathrm{I}}^{\mathrm{V}}$, these elementary rotations must be reversed. Since ψ is relative to an arbitrary direction, it can be disregarded without loss of generality. Thus, $R_{\mathrm{I}}^{\mathrm{V}}$ is computed as

$$R_{\mathrm{I}}^{\mathrm{V}} = \mathrm{R}_x(-\phi)\mathrm{R}_y(-\theta) = \begin{bmatrix} \mathrm{c}_\theta & 0 & -\mathrm{s}_\theta \\ \mathrm{s}_\phi \mathrm{s}_\theta & \mathrm{c}_\phi & \mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{c}_\phi \mathrm{s}_\theta & -\mathrm{s}_\phi & \mathrm{c}_\phi \mathrm{c}_\theta \end{bmatrix}. \tag{5}$$

Having computed $H_{\mathrm{C}}^{\mathrm{V}}$, estimates of device attitude obtained with the IMU are leveraged to express points p^{C} captured by the device camera into points p^{V} as if they were captured by a virtual camera pointed along the vertical. However, since V has 4 degrees of freedom corresponding to translation along x_{V}, y_{V}, z_{V}, and rotation about z_{V}, points p^{V} must undergo additional transformations to be represented in the fixed world frame W. We choose W to be attached to one of the robots in the system, which we refer to as the reference robot. If $p^{\mathrm{V}} = [a \;\; b \;\; 1]^{\mathrm{T}}$ represents the 2D location of the marker attached to reference robot as projected in V, then the translation operation $\mathrm{Trans}(-a, -b)$ transforms points in V to a frame V′ positioned directly above the reference robot (see Fig. 3). Since points can be represented relative to frame V′ pointing down the vertical, the angle ω and width w_{pix} in pixels of the square marker can be accurately computed. First, the angle ω is used to perform the rotation operation $\mathrm{Rot}(-\omega)$, which transforms points in V′ to a frame O that is positioned above and aligned with the reference robot. Next, using the known marker width w in real-world units and the width w_{pix} in pixels, the scaling operation $\mathrm{Scale}(\lambda_w)$ can be performed, where $\lambda_w = \frac{w}{w_{\mathrm{pix}}}$, to transform points in O to the frame W that is positioned above the reference robot, aligned with the reference robot, and whose points are expressed in real-world units. Composing the three transformations, we compute

$$\begin{aligned} T_{\mathrm{V}}^{\mathrm{W}} &= \mathrm{Scale}(\lambda_w)\mathrm{Rot}(-\omega)\mathrm{Trans}(-a, -b) \\ &= \begin{bmatrix} \lambda_w & 0 & 0 \\ 0 & \lambda_w & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{c}_\omega & \mathrm{s}_\omega & 0 \\ -\mathrm{s}_\omega & \mathrm{c}_\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \lambda_w \mathrm{c}_\omega & \lambda_w \mathrm{s}_\omega & -\lambda_w (a\mathrm{c}_\omega + b\mathrm{s}_\omega) \\ -\lambda_w \mathrm{s}_\theta & \lambda_w \mathrm{c}_\omega & \lambda_w (a\mathrm{s}_\omega - b\mathrm{c}_\omega) \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{6}$$

Once $H_{\mathrm{C}}^{\mathrm{W}}$ is computed from (2) using the estimates of device attitude and the image data of the marker on the reference robot, the relative poses of acting robots and manipulable objects are estimated in a more computationally efficient manner than in approach **A**. That is, rather than computing $T_{\mathrm{R}_i}^{\mathrm{C}}$ and $T_{\mathrm{O}_j}^{\mathrm{C}}$ from point correspondences for each robot and object in the scene, the coordinates of the four corners of each marker are detected in C and then transformed by $H_{\mathrm{C}}^{\mathrm{W}}$ to permit the direct computation of poses relative to the reference robot (i.e., relative to W).

Since the feedback control is vision-based, both approaches **A** and **B** require the reference robots, acting robots, and objects of interest to be in view of the mobile device camera. However, approach **B** offers more flexibility. For example, although both approaches fix the world frame to one of the robots in the workspace, approach **B** requires only one reference robot while other robots are free to be acting robots. Moreover, in approach **B**, by tracking and controlling robots relative to the frame attached to the reference robot, the reference robot does not need to start at a known pose and may be arbitrarily moved or switched as long as at least one other robot is in view and not being controlled. Thus, using approach **B**, less robots are required to be in view of the camera at one time and operators can readily interact with and control, one at a time, separate subgroups of robots. Specifically, if a reference robot $\mathrm{R_r}$ either leaves the view of the camera or is controlled to move, the role of the reference robot is assigned to an arbitrary robot R_i that is not being controlled. Since W was aligned with $\mathrm{R_r}$ before re-assignment and R_i after re-assignment, $H_{\mathrm{C}}^{\mathrm{W}} = T_{\mathrm{C}}^{\mathrm{R_r}}$ becomes $H_{\mathrm{C}}^{\mathrm{W}} = T_{\mathrm{C}}^{\mathrm{R}_i}$ after re-assignment. Thus, a robot R_k is estimated in the original frame W=$\mathrm{R_r}$ after re-assignment by

$$T_{\mathrm{R}_k}^{\mathrm{R_r}} = T_{\mathrm{R}_i}^{\mathrm{R_r}} T_{\mathrm{C}}^{\mathrm{R}_i} T_{\mathrm{R}_k}^{\mathrm{C}} = T_{\mathrm{C}}^{\mathrm{R_r}} T_{\mathrm{R}_k}^{\mathrm{C}} = H_{\mathrm{C}}^{\mathrm{W}} T_{\mathrm{R}_k}^{\mathrm{C}}. \tag{7}$$

Thus, the reference robot is switched by simply updating $H_{\mathrm{C}}^{\mathrm{W}}$ by $H_{\mathrm{C}}^{\mathrm{W}=\mathrm{R}_i} := H_{\mathrm{W}=\mathrm{R_r}}^{\mathrm{R}_i} H_{\mathrm{C}}^{\mathrm{W}=R_{\mathrm{r}}}$. Note that $H_{\mathrm{R}_i}^{\mathrm{R_r}} = (H_{\mathrm{R_r}}^{\mathrm{R}_i})^{-1}$ is the most recently computed relative pose of R_i before the re-assignment. If no acting robots are being controlled when the reference robot leaves the view of the camera, then there is no need to update $H_{\mathrm{C}}^{\mathrm{W}}$; instead a new reference robot is arbitrarily assigned and $H_{\mathrm{C}}^{\mathrm{W}}$ is reset. Thus, a limitation is when the reference robot is to be assigned for control and there are no other robots in view available to become the reference robot. In future work, a velocity motion model of the reference robot will be used to maintain W at the reference robot's initial pose so that a reference robot may arbitrarily move without the need for re-assignment to another robot. Finally, unlike approach **A**, the plane occupied by the robots must be assumed horizontal since device attitude is estimated relative to the direction of gravity. If the inclination of the plane is known relative to frame W, one can modify the approach to apply an additional 3D rotation that compensates for the inclination. However, since W is attached to the local frame of the reference robot, the reference robot is not permitted to move or be switched in this case. Applications in which the plane is not horizontal are beyond the scope of this study.

C. User Interaction and Robot Control

Once the world frame has been established, estimates of robot and object poses enable the display of interactive augmented

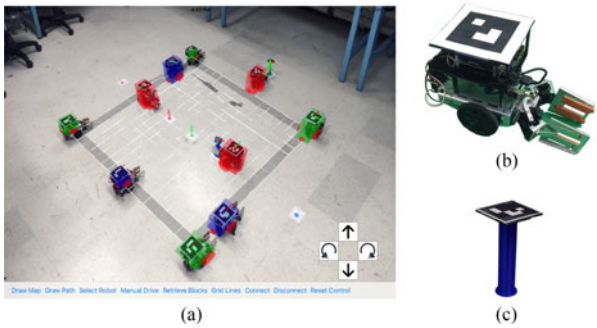

Fig. 4. (a) Mobile mixed-reality interface, (b) a robot, and (c) an object used in the human-robot interaction study.

graphics that enhance the operator's situational awareness and can be manipulated on the touchscreen to intuitively command robots. When the screen is contacted at a point p_{S} in pixels, a transformation is performed to map p_{S} to a corresponding real-world location p_{W} in the workspace

$$p_{\mathrm{W}} = H_{\mathrm{C}}^{\mathrm{W}} \mathrm{Scale}(\lambda_x, \lambda_y) p_{\mathrm{S}} = H_{\mathrm{C}}^{\mathrm{W}} \mathrm{diag}\,(\lambda_x, \lambda_y, 1)\, p_{\mathrm{S}}, \quad (8)$$

where $\lambda_x \triangleq \frac{w_{\mathrm{C}}}{w_{\mathrm{S}}}$ and $\lambda_y \triangleq \frac{h_{\mathrm{C}}}{h_{\mathrm{S}}}$ are the scaling factors that transform the screen coordinates to image coordinates, given the width and height of the image $(w_{\mathrm{C}}, h_{\mathrm{C}})$ and the width and height of the screen $(w_{\mathrm{S}}, h_{\mathrm{S}})$ in pixels. In this way, users can intuitively command robots by interacting directly with the augmented graphics from their own perspective of the shared space. Among the augmented graphics displayed by the interface is a virtual grid floor overlaid on the actual floor in the image. This augmented floor consists of evenly-spaced lines to provide a visual aid that assists operators in performing precise manipulations of the virtual objects in the mixed-reality environment (see Fig. 4(a)). Furthermore, the interface renders virtual robots that correspond to each robot detected in the scene and are highlighted in various colors to improve operators' situational awareness. With approach **A**, virtual robots highlighted in green, blue, or red indicate that the corresponding robots are known, *unknown* (i.e., its pose is estimated), or currently acting, respectively. With approach **B**, virtual robots highlighted in green, blue, or red indicate that the corresponding robots are the reference robot, acting robots not being controlled, or currently acting robots, respectively. By rendering a virtual manipulable object corresponding to each object detected in the scene, the interface allows operators to indirectly command robots to pick up and drop off objects of interest by tapping and dragging the virtual objects to desired locations in the mixed-reality environment, thus allowing operators to directly interact with the world. Once the operator's manipulations of a virtual object have been captured, the interface executes an algorithm to detect which robot is located closest to the object and is available to retrieve it. In this study, objects of interest are vertically standing cylinders that can be approached from an arbitrary direction. Since the space is open and without obstacles, a path planning algorithm uses the estimated positions of the object and nearest robot to plan straight-line paths directly between robots and objects, replanning at regular intervals to correct paths that may cause collisions between acting robots [21]. Once a path has been planned for a robot to retrieve an object, a visual feedback control algorithm is used that, inspired by sensor-based line-following algorithms, computes wheel velocity commands for the robot by making use of small regions on the left and right sides of a corresponding virtual robot that can "sense" the virtual path to be followed [22]. By monitoring the estimated poses of the robot and object, the interface is aware of the robot's progress and issues commands for the robot to stop, close its gripper to pick up the object, follow a planned path to the drop off location, and open its gripper to release the object. As a robot is controlled to act on an object, the operator is free to manipulate another object, in which case the process of planning and controlling the closest available robot is repeated and the interface controls multiple robots simultaneously. Note that approach **A** and **B** each provide an algorithm to handle the case when the closest robot to the object is a reference robot.

III. Experimental Implementation

To evaluate the performance associated with the proposed approaches, experiments are conducted using a tablet and ten wheeled mobile robots based on a small (12.7 cm $\times$ 8.26 cm $\times$ 3.18 cm) differential-drive platform (see Fig. 4(b)). Grippers mounted to the front of each robot can lift objects up to 5 cm wide and weighing up to 397 g. Objects used in this study are 1.9-cm diameter, 12-cm tall cylinders (see Fig. 4(c)). Mounted electronics include a motor controller that drives the wheels and gripper, as well as a single board computer that provides Wi-Fi connectivity and allows robots to subscribe to commands published by the tablet over a network maintained by one of the robots. Since the robots do not require any sensors to perform the object retrieval task in this study, they cost a fraction of the amount for most conventional platforms. Although the interface is implemented on an Apple iPad Pro in this study, any tablet computer with comparable sensing and computational capabilities can support the proposed mobile mixed-reality interfaces using open-source third-party libraries for computer vision, graphics rendering, and communication over the robots' network.

IV. Experimental Results

A. Application Performance

The utilization of mobile technologies to track, interact with, and control multi-robot systems remains largely unexplored. From Section II we expect that the accuracy with which robots are tracked, interacted with, and controlled to operate on objects using mobile mixed-reality applications depends on the accuracy with which the pose of the camera frame relative to the world frame, $H_{\mathrm{C}}^{\mathrm{W}}$, is estimated. The accuracy of $H_{\mathrm{C}}^{\mathrm{W}}$ relies on the accuracy of the visual detection of markers affixed to the robots in approach **A**, as well as the accuracy of device attitude estimates from inertial measurements in approach **B**. Since the real-time control of the robots is implemented on a mobile device, there exists a narrow time window to process sensor data and update the commands to the robots before their stability is degraded. Thus, the effects of visual and inertial processing on

TABLE I
RELATIONSHIP BETWEEN MAXIMUM ALLOWABLE DISTANCE, CAMERA RESOLUTION, AND MEAN VISUAL PROCESSING TIME

Resolution	Max Distance, m	Processing Time, ms
352×288	1.194	5.354(1.866)
640×480	1.702	11.524(2.097)
960×540	2.743	15.432(1.740)
1280×720	3.378	24.768(3.376)
1920×1080	4.267	30.930(1.601)
3840×2160	5.436	62.673(2.233)

Standard deviations are in parentheses.

the performance of both the multi-robot system and the mobile application are of interest. The selection of camera resolution has a direct effect on the operator's maximum allowable distance from a robot's marker and the application's computational load. Table I shows the maximum allowable distance of a robot on the ground from a tablet held 1.524 m above the ground as camera frames are captured at various resolutions, as well as the mean processing time required to detect the robot. Results confirm that robots farther away from the device require higher camera resolutions to be detected, which in turn necessitates longer processing. In this study, markers are printed to have the same width as the robots so that, without protruding from the robots, they may be detected from the farthest range. However, markers can be made smaller at the expense of reducing the maximum distance from which they can be detected. As evidenced from Table I, although the highest resolution offers the longest range, its mean processing time becomes too slow to support real-time control of the system. Since the largest burden on processing comes from searching through detected contours for potential markers, a minimum contour size threshold is used to filter contours too small to be markers. This allows frames to be processed at the maximum resolution in nearly half the time ($M = 31.767$ ms, $SD = 2.159$ ms) at the expense of 34.6% reduction in the maximum range since markers farther from the camera have smaller contours.

To compare the processing demands of each approach, the mean processing time is recorded for 20 trials using each approach with $N = 10$ robots. The mean processing time of approach **A** ($M = 45.47$ ms, $SD = 2.66$ ms) is 25.4% longer than that of approach **B** ($M = 33.91$ ms, $SD = 0.70$ ms), and the difference is statistically significant according to Welch's unequal variance t-test ($t(22) = 18.76, p < 0.05$). To track, interact with, and control a multi-robot system, the effect on computational load as the number of robots is increased is of interest. According to unpaired equal variance t-tests, a statistically significant increase is exhibited in the total processing time of approach **A** ($t(38) = 7.90, p < 0.05$) as N is increased from 4 (the minimum number for approach **A** to operate) to 10, as well as for approach **B** ($t(38) = 3.69, p < 0.05$) as N is increased from 1 (the minimum number for approach **B** to operate) to 10. This is expected since the visual processing required to track each robot consistently accounts for most of the computational load (more than 79.6% in approach **A** and 71.4% in approach **B**). However, whereas the processing time of approach **A** increases by 13.0%, the processing time of approach **B** increases by only 6.3%, indicating that the standard vision techniques used by approach **A**, which rely on least squares estimation using image data from many robots, scale less efficiently with increases in N than the proposed technique of approach **B**, whose dependence on IMU data and image data from only one robot is less sensitive to increases in N. Specifically, the first step of approach **B** computes device attitude estimates at 100 Hz and consistently requires less than 1% of the processing time to compute estimates of H_C^V ($M = 27.92\,\mu$s, $SD = 5.79\,\mu$s). Thus, approach **B** is computationally more efficient than approach **A**.

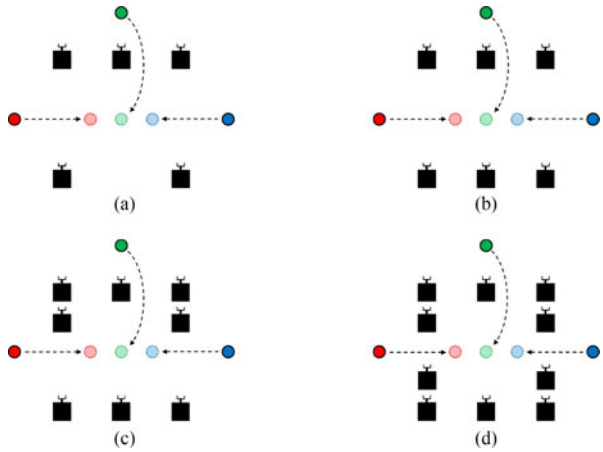

Fig. 5. The configurations of robots used to evaluate the approaches in the study. In activities, the robots are commanded to retrieve objects outside of the shown formations and drop them off at locations near the center of the formations.

B. Task Performance

The task performance of the multi-robot system is evaluated by conducting experiments in which teams of robots are arranged in the formations shown in Fig. 5 and commanded to retrieve three objects in their vicinity. To investigate the task performance achieved with each approach, trials are repeated 36 times with each approach for a total of 72 trials. Specifically, for approach **A**, each of the following 12 combinations of N and n are tested 3 times: with $N = 5$: $n = 4, 5$; with $N = 6$: $n = 4, 5, 6$; with $N = 8$: $n = 4, 6, 8$; and with $N = 10$: $n = 4, 6, 8, 10$. However, since n is undefined in approach **B**, each of the 4 values of N are tested 9 times to yield 36 trials. For mobile mixed-reality approaches to be effective for multi-robot systems, acceptable performance must be exhibited in terms of task success, time efficiency, as well as accuracy and precision of operations. A trial is completed successfully if each of the three objects is picked up and dropped off in a center area above a spot on the ground of the same color as the object. Although the multi-robot system successfully completed the task for all 36 trials using approach **B**, it was unsuccessful using approach **A** for the 15 trials with $N = 5, 6$. This is because $N = 5, 6$ permits only a small number of available robots with estimated poses, failing to provide a sufficient area coverage to accurately perform the task, an issue that is not encountered using approach **B**.

The time efficiency of the system is assessed by recording the time taken to complete the object retrieval task and the number

of robots controlled simultaneously to complete the task during each trial. Since the collected samples of completion time are independent, yield normally distributed residuals, and have variances that are relatively homoscedastic [23], a three-way analysis of variance (ANOVA) is conducted on the data in which the approach used, number of simultaneously controlled robots, and N are the factors. Results indicate significant differences in completion time achieved with one robot ($M = 130.17$ s, $SD = 9.67$ s), two robots ($M = 76.52$ s, $SD = 5.69$ s), and three robots ($M = 50.75$ s, $SD = 5.09$ s) being controlled at a time ($F(6,33) = 565.41, p < 0.05$). Using approach **A**, at least four robots must not move to maintain the world frame, allowing only one object to be retrieved at a time when $N = 5$, two objects when $N = 6$, and all three objects when $N = 8, 10$. Since only one robot cannot move in approach **B**, all three objects can be retrieved at a time when $N = 5$, with one robot free. Thus, approach **B** frees more robots to retrieve objects, achieving similar completion times with only four robots versus at least seven robots using approach **A**. However, since an equal number of trials is conducted with two approaches (**A** using $N = 8, 10$ and **B** using $N = 5, 6, 8, 10$) in which one, two, and three objects are retrieved at a time, this advantage is not reflected in the results of the ANOVA comparing the mean completion time with each approach ($F(6,33) = 0.25, p = 0.6182$). Finally, since N has a direct effect on the maximum number of simultaneously acting robots ($a_{\max} = N - 4$ in approach **A**, $a_{\max} = N - 1$ in approach **B**), N indirectly affects completion time.

To assess the accuracy and precision associated with task performance, trials are conducted in which the pick-up location estimated by the interface, the drop-off location generated by interactions with the touchscreen, and the drop-off location achieved by robots are measured for each object. Accuracy is assessed by performing a series of two-tailed t-tests over successful trials that compare the means of these locations with their corresponding goal locations. The results of the t-tests indicate that there is no statistically significant difference between the mean estimated pick-up location of each object and its start location (controlled for experiments). Thus, the approaches yield accurate vision-based estimates of real-world locations of objects (and robots) in the environment. Moreover, results of additional t-tests indicate that there is no statistically significant difference between the mean drop-off location of each object as commanded by touchscreen interactions and its goal location (controlled for experiments). Thus, the approaches allow operators to effectively use the interface to accurately communicate intended spatial commands to multi-robot systems. Finally, results of t-tests indicate that there is no statistically significant difference between the measured drop-off location of each object and its goal location. Thus, the approaches effectively use the estimated poses of objects and robots to perform vision-based control of the robots such that the objects are accurately placed in the environment.

To identify factors that influence precision in estimated pick-up, commanded drop-off, and achieved drop-off locations using each approach, a series of ANOVAs are conducted after confirming that recorded locations adhere to the conditions for applying ANOVAs [23]. For approach **A**, N and n are considered as factors. The results of the ANOVAs indicate that

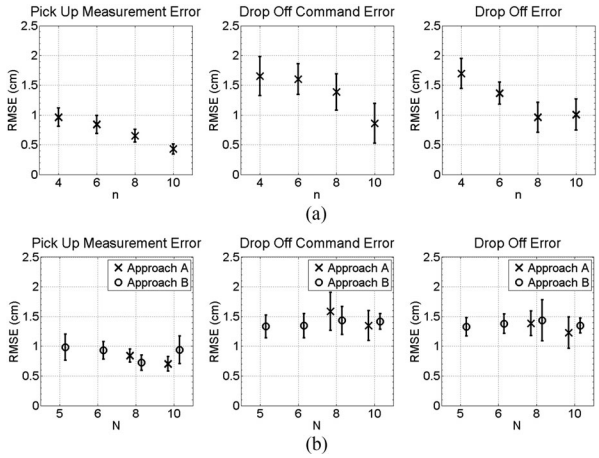

Fig. 6. RMSE in measured pick-up, commanded drop-off, and achieved drop-off locations while using each approach.

changes in n have a statistically significant effect on measured pick-up locations ($F(3, 13) = 19.84, p < 0.05$), commanded drop-off locations ($F(3, 13) = 5.17, p < 0.05$), and achieved drop-off locations ($F(3, 13) = 3.42, p < 0.05$). For approach **B**, only N is considered as a factor, however no significant effect is observed on the measured pick-up locations ($F(3, 32) = 1.47, p = 0.2400$), commanded drop-off locations ($F(3, 32) = 0.25, p = 0.8603$), or achieved drop-off locations ($F(3, 32) = 0.76, p = 0.5256$) as N is changed. These results are expected since the standard vision techniques used by approach **A** to estimate H_C^W improve in accuracy as the number of corresponding points increases [24], whereas the technique proposed in approach **B**, which relies on inertial data as well as image data of only one robot, is not affected by changes in N. To further investigate the precision obtained with each approach, the root mean square errors (RMSE) of the location data are computed. Fig. 6(a) shows the RMSE incurred by approach **A** for the different values of n, while Fig. 6(b) shows the RMSE incurred by approaches **A** and **B** for different values of N. Note that larger RMSE is incurred for commanded and achieved drop-off locations than for measured pick-up locations. While the error in vision-based measurements of pick-up location is due to error in estimating H_C^W, commanded drop-off location has additional error due to variability in fine motor skills of operators. Moreover, achieved drop-off location has the error of commanded drop-off location, plus additional error from robot control. Fig. 6(a) confirms that the precision of computer vision, user interaction, and robot control improve with approach **A** as n is increased. In Fig. 6(b), data is not available for approach **A** when $N = 5, 6$ since approach **A** is found to require more robots to cover the area for trials to be successful. When $N = 8, 10$, a mean RMSE is obtained with approach **A** that is not statistically significant from the mean RMSE obtained with approach **B**. Moreover, Fig. 6(b) confirms that the performance of approach B does not significantly change with changes in N.

As the total number of robots in the system increases, more robots are available to either manipulate objects, improving time efficiency of tasks, or to maintain the world frame, improving the accuracy and precision of approach **A**. As the size of the

workspace to be covered and the complexity of the task increase, so do the required number of robots to improve performance. However, these conditions cause increases in vision processing, the time spent planning and controlling robots, and the required camera resolution. Thus, for approach **A** the number of robots in the system is a critical design parameter that impacts both task performance and the computational demands of the mobile device. Because the estimation of $H_{\mathrm{C}}^{\mathrm{W}}$ in approach **A** relies on point correspondences, the accuracy of the computation is sensitive not only to the noise in the image data of each of the robots but also in the placement of each of the robots. Unlike approach **A**, approach **B** uses inertial data and the image data of only one robot to estimate $H_{\mathrm{C}}^{\mathrm{W}}$. Thus, neither the application performance nor task performance varies significantly with robot placement or with changes in the number of robots, making it more scalable, a desirable quality for implementation of human-multi-robot systems. However, as observed by the error bars in Fig. 6(b), the precision exhibited by approach **B** varies largely within each of the values of N. We believe this is caused by noise during trials in the inertial data, which approach **A** is not affected by, and raw image data, which approach **A** is robust to using least squares estimation. Thus, approach **B** is less accurate while an operator moves the device. This means that the improved flexibility and computational time of approach **B** over approach **A** come at the expense of the increased sensitivity to sensor noise. Future work will reduce the sensitivity of approach **B** by filtering the inertial data to allow moving of the device at operator-induced frequencies.

V. Conclusion

This paper proposed a mobile mixed-reality approach to interact with and control a multi-robot system. It is shown that the proposed approach allows mobile devices to perform pose estimation and vision-based control of multiple robots and simultaneously render 3D augmented graphics and monitor operators' touchscreen interactions while being held and moved arbitrarily by operators. Experimental results show that, compared to using standard vision techniques, employing both the inertial and visual sensors of the mobile device yields improvements in the flexibility, performance, and computational efficiency at the expense of increased sensitivity to sensor noise. The proposed approach has been experimentally validated when robots drive in a plane, are fitted with visual markers, and are located within the field of view of the mobile device camera. These conditions, wherein operators and visually detectable robots share a space with level ground, exist in some of the most common environments where robots are beginning to appear, e.g., our homes, offices, classrooms, and warehouses. Nevertheless, future work will explore solutions that allow the removal of these limitations and will involve a full scale user study to evaluate user experiences associated with the proposed interface approach.

Acknowledgment

B. Nguyen and A. Zhang helped build the robots used in this study.

References

[1] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 376–386, Jun. 2005.

[2] D. Goldberg and M. J. Matarić, "Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks," in *Robot Teams: From Diversity to Polymorphism*. New York, NY, USA: Taylor & Francis, 2001, pp. 1–24.

[3] B. B. Werger, "Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams," *Artif. Intell.*, vol. 110, no. 2, pp. 293–320, 1999.

[4] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics*. Berlin, Germany: Springer, 2005, Ch. 2, pp. 10–20.

[5] D. Sakamoto, Y. Sugiura, M. Inami, and T. Igarashi, "Graphical instruction for home robots," *Computer*, vol. 49, no. 7, pp. 20–25, 2016.

[6] J. McLurkin, J. Rykowski, M. John, Q. Kaseman, and A. J. Lynch, "Using multi-robot systems for engineering education: Teaching and outreach with large numbers of an advanced, low-cost robot," *IEEE Trans. Edu.*, vol. 56, no. 1, pp. 24–33, Feb. 2013.

[7] A. Rosenfeld, A. Noa, O. Maksimov, and S. Kraus, "Human-multi-robot team collaboration for efficient warehouse operation," in *Proc. Workshop Auton. Robots Multirobot Syst.*, Singapore, 2016.

[8] J. Nagi, A. Giusti, L. M. Gambardella, and G. A. Di Caro, "Human-swarm interaction using spatial gestures," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3834–3841.

[9] J. Alonso-Mora, S. H. Lohaus, P. Leemann, R. Siegwart, and P. Beardsley, "Gesture based human-multi-robot swarm interaction and its application to an interactive display," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5948–5953.

[10] R. Grieder, J. Alonso-Mora, C. Bloechlinger, R. Siegwart, and P. Beardsley, "Multi-robot control and interaction with a hand-held tablet," in *Proc. ICRA Workshop Multiple Robot Syst.*, 2014.

[11] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 489–493, Apr. 2015.

[12] M. Fiala, "A robot control and augmented reality interface for multiple robots," in *Proc. Can. Conf. Comput. Robot Vis.*, 2009, pp. 31–36.

[13] J. A. Frank and V. Kapila, "Using mobile devices for mixed-reality interactions with educational laboratory test-beds," *Mech. Eng.*, vol. 138, no. 6, pp. 2–6, 2016.

[14] J. A. Frank, M. Moorhead, and V. Kapila, "Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks," in *Proc. IEEE Int. Symp. Robot Human Interact. Commun.*, New York, NY, USA, 2016, pp. 302–307.

[15] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[16] D. L. Baggio, *Mastering OpenCV With Practical Computer Vision Projects*. Birmingham, U.K.: Packt Publishing Ltd., 2012.

[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[18] D. Oberkampf, D. F. DeMenthon, and L.S. Davis, "Iterative pose estimation using coplanar feature points," *Comput. Vis. Image Underst.*, vol. 63, no. 37, pp. 495–511, 1996.

[19] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, Atlanta, GA, USA, 2011, pp. 1–7.

[20] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, Jun. 2008.

[21] J. A. Frank and V. Kapila, "Path bending: Interactive human-robot interfaces with collision-free correction of user-drawn paths," in *Proc. Int. Conf. Intell. User Interfaces*, Atlanta, GA, USA, 2015, pp. 186–190.

[22] J. A. Frank, Y. Sahasrabudhe, and V. Kapila, "An augmented reality approach for reliable autonomous path navigation of mobile robots," in *Proc. Indian Control Conf.*, Chennai, India, 2015, pp. 437–443.

[23] B. G. Tabachnick and L. S. Fidell, *Using Multivariate Statistics*, 6th ed. Boston, MA, USA: Pearson, 2013.

[24] R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim, "Pose estimation from corresponding point data," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 6, pp. 1426–1446, Nov./Dec. 1989.