Drone-Augmented Human Vision:
Exocentric Control for Drones Exploring Hidden Areas

Okan Erat, Werner Alexander Isop, Denis Kalkofen and Dieter Schmalstieg, *Member, IEEE*

Fig. 1. **This image was captured live with a second HoloLens on a tripod.** The drone-augmented human is interacting with the drone in an exocentric view. The human is steering the camera drone via gaze direction and perceives an X-ray-like vision into occluded areas. The brick pattern is a physical wallpaper. Lower part of the mannequin is applied as a perspectively correct texture and extends the users perception of the visible upper part of the mannequin.

Abstract—Drones allow exploring dangerous or impassable areas safely from a distant point of view. However, flight control from an egocentric view in narrow or constrained environments can be challenging. Arguably, an exocentric view would afford a better overview and, thus, more intuitive flight control of the drone. Unfortunately, such an exocentric view is unavailable when exploring indoor environments. This paper investigates the potential of drone-augmented human vision, i.e., of exploring the environment and controlling the drone indirectly from an exocentric viewpoint. If used with a see-through display, this approach can simulate X-ray vision to provide a natural view into an otherwise occluded environment. The user's view is synthesized from a three-dimensional reconstruction of the indoor environment using image-based rendering. This user interface is designed to reduce the cognitive load of the drone's flight control. The user can concentrate on the exploration of the inaccessible space, while flight control is largely delegated to the drone's autopilot system. We assess our system with a first experiment showing how drone-augmented human vision supports spatial understanding and improves natural interaction with the drone.

Index Terms—X-ray,mixed reality,hololens,drone,pick-and-place.

---◆---

1 INTRODUCTION

Drones have recently soared in popularity, largely driven by the exciting applications of drones with on-board cameras. Camera drones can easily fly into locations which are impassable or too dangerous for humans to reach. Using camera drones allows exploring such areas from a safe distance, providing essential data for diverse applications, such as rescue missions, infrastructure inspection or just photographic

- *Okan Erat is with Institute of Computer Graphics and Vision at Graz University of Technology. E-mail: okan.erat@icg.tugraz.at.*
- *Alexander W. Isop is with Institute of Computer Graphics and Vision at Graz University of Technology. E-mail: isop@icg.tugraz.at.*
- *Denis Kalkofen is with Institute of Computer Graphics and Vision at Graz University of Technology. E-mail: kalkofen@icg.tugraz.at.*
- *Dieter Schmalstieg is with Institute of Computer Graphics and Vision at Graz University of Technology. E-mail: dieter@icg.tugraz.at.*

exploration.

Depending on the situation, a drone pilot may choose between two principal modes of flight control. In either mode, a conventional hand-held controller is used to steer the drone, but the viewing differs between modes: In *exocentric viewing* mode, the pilot observes the drone from the ground while steering. In *egocentric viewing* – or first-person – mode, video from the drone's on-board camera is streamed to the pilot to inform the steering. Recently, wearing a head-mounted display to watch the streaming video has become a popular enhancement of the egocentric mode.

Obviously, piloting a drone in exocentric mode is difficult in the presence of significant occlusions, and it becomes impossible when the drone is exploring the inside of a building. In this case, the pilot is forced to use an egocentric mode based on streaming the image from the on-board camera. But navigation in a narrow environment while relying exclusively on an on-board camera with a potentially limited

field of view can be difficult.

In such a difficult situation, the pilot may substantially benefit from an autopilot system, which enables indirect flight control. The pilot specifies a destination, and the autopilot steers the drone there autonomously. State-of-the-art autopilots stabilize the drone pose during motion and prevent crashing into obstacles, but cannot perform path-planning or way-finding. The pilot must still maintain the overview and guide the drone step by step.

Precisely this overview is lacking if only an egocentric view is available, making flight control more difficult than necessary. An exocentric view, showing the drone's surrounding from the pilot's rather than the drone's point of view, would clearly be preferable.

Virtual reality (VR) can provide a synthetic exocentric view by combining the live video with a 3D model of the occluded environment from any viewpoint, independent of user's physical viewpoint. Using image-based rendering of the drone's video stream delivers a realistic impression with live updates [25]. By coupling the drone autopilot to the user's gaze direction, the experience is redefined from remotely piloting a drone to perceiving the occluded world with *drone-augmented human vision*. As a special case of VR, augmented reality (AR) additionally adds the illusion of X-ray vision: A pilot wearing a see-through display can make the walls or other occluders partially transparent to reveal the area currently observed by the drone.

In addition to supporting a virtual viewpoint, AR also allows users to investigate the scene from their physical viewpoint and spatially relate occluded geometry with the visible world. In case of a disaster scenario, a rescue team can quickly locate an imminent danger, such as a fire or explosion behind an occluder, and proceed with caution.

Spatial relationships between visible and occluded geometry become especially important when infrastructure modifications are needed. For example, drilling a hole in a wall without damaging the cables or pipes located on the occluded side of the wall requires estimating their positions from the visible part of the wall.

The concept of drone-augmented human vision is entirely technically feasible today. Contemporary drone autopilots, such as the PX4 Pixhawk [21], rely on visio-inertial odometry to monitor their relative motion in space. Research prototypes using additional off-board computational resources [9] extend odometry into dense simultaneous localization and mapping (SLAM), delivering an instant 3D model of the area observed by the drone. This 3D model informs the autopilot, but also provides the necessary geometry component for high-quality image-based rendering.

In this paper, we demonstrate the first proof-of-concept implementation of drone-augmented human vision. We couple an indoor drone with a head-mounted display (HMD) to deliver an exocentric perspective on the drone, letting the pilot control the drone via gaze direction. The drone carries an autopilot, but relies on external tracking, since we wanted optimal flight stability for our prototype. We present a first experiment showing how virtual exocentric visualization supports spatial understanding and thus enables exploration and natural interaction with the drone. In a second experiment, we use VR (non-see through) for its virtual viewpoint nature and compare it with the physical viewpoint that is additionally provided by AR (see-through).

2 RELATED WORK

Existing work on occluded or remote space discovery with drones proposes a variety of interaction techniques to steer the drone and visualize the data coming from its sensors. Depending on the visualization of the sensor data, mostly from cameras, related work can be categorized into egocentric control and exocentric control. Moreover, our work is related to remote visualization techniques involving live video.

2.1 Egocentric drone control

Egocentric control techniques visualize camera images from the first-person view of the drone and immerse the user into the remote location currently occupied by the drone. Using an HMD to display video from a drone, Mirk and Hlavacs [23] created a virtual tourist application. However, the user was not given full control of the drone to prevent crashes; only the user's head movements were translated into the yaw rotation of the drone. Hansen et al. [10] capture eye gaze, while the drone pilot is looking at the camera stream. The 2D vector formed between screen center and the point gazed at on the screen is mapped to speed and rotation around a 3D axis in the drone's local frame. As humans tend to rapidly change their gaze direction, this technique may be problematic for flight control whenever the pilot looses concentration.

Higuchi et al. [13] synchronize head movements of the user with a drone, except for pitch and roll rotations. While this gives an intuitive control, the latency between the pilot's movements and response of the drone can quickly create motion sickness. In addition, the motion dynamics of the drone make it impossible for the drone to exactly replicate the path taken by pilot's head, negatively affecting the spatial understanding of the human. As summarized by Chen et al. [5], egocentric robot control presents the user with the several problems, the most severe ones being narrow field of view (FOV), orientation and altitude misjudgement and a general lack of scene understanding.

2.2 Exocentric drone control

In contrast to egocentric control, an exocentric control technique steers the drone while the user is observing it directly. As discussed by Cho et al. [6], exocentric drone control is prone to accidents due to left-right confusion between user's and drone's local coordinate frames. Kashara et al. [19] tackle this problem by allowing users to control the drone with a touch screen device in their own reference frame and mapping control commands into the drone's local coordinates automatically. However, the users had to observe the drone with the device's camera for pose estimation and move it on the 2D screen, which is not possible in the presence of occlusions. In addition, 2D gestures do not allow for an intuitive interface for generating a motion vector that is a combination of axes. Similar to Kashara et al., Hashimoto et al. [12] also provides a touch screen based control, but they place a camera to a fixed viewpoint to observe the robot. This is not feasible during an occluded scene investigation.

Saakes et al. [27] uses a drone camera to observe a ground robot from a third person view. In an unknown occluded environment, using another robot just increases the complexity. Sugimoto et al. and Hing et al. [14, 31] provides a visualization to observe the robot from an exocentric point of view. However, their systems limit the freedom of the viewpoint and makes it hard to relate surrounding colliders to the robot. Karanam et al. [18] use WiFi signals transmitted by drones to monitor them behind the occluding structures.

Zollmann et al. [34] focuses on the spatial understanding problems that arise when the drone is far away from the user. They use an exocentric AR display based on the backfacing camera of a handheld tablet. The drone's altitude over the terrain and distance to the user is visualized in 3D on top of the video. However, if the drone faces dense obstacles in close proximity, this technique does not provide a detailed enough visualization for accurate control. Bergé et.al. [3] create a synthetic point cloud resembling a 3D reconstruction obtained by a drone and visualize it in immersive VR. They also develop a method to evaluate the difficulty of finding a target.

All these techniques demonstrate the potential of using an exocentric viewpoint for drone control, but do not allow for easy and intuitive navigation. Introducing direct manipulation for this purpose is the main contribution of our work.

2.3 Visualization of remote or occluded information

Simulating X-ray vision for the purpose of revealing hidden infrastructure has been a goal of AR research for a long time [8]. Most of the X-ray vision techniques compose a video image with purely virtual information or simulate a cutaway of the occluder [7].

Our work also relates to research on visualizing and interacting in a distant environments and, by extension, to telepresence systems. For example, remote visualization in real-world scale was presented by Kasahara et al. [20]. The system provides omnidirectional remote visualization, enabling a user to participate in the remote user's application. While the system allows to decouple orientation, it does not provide control over the user's position.

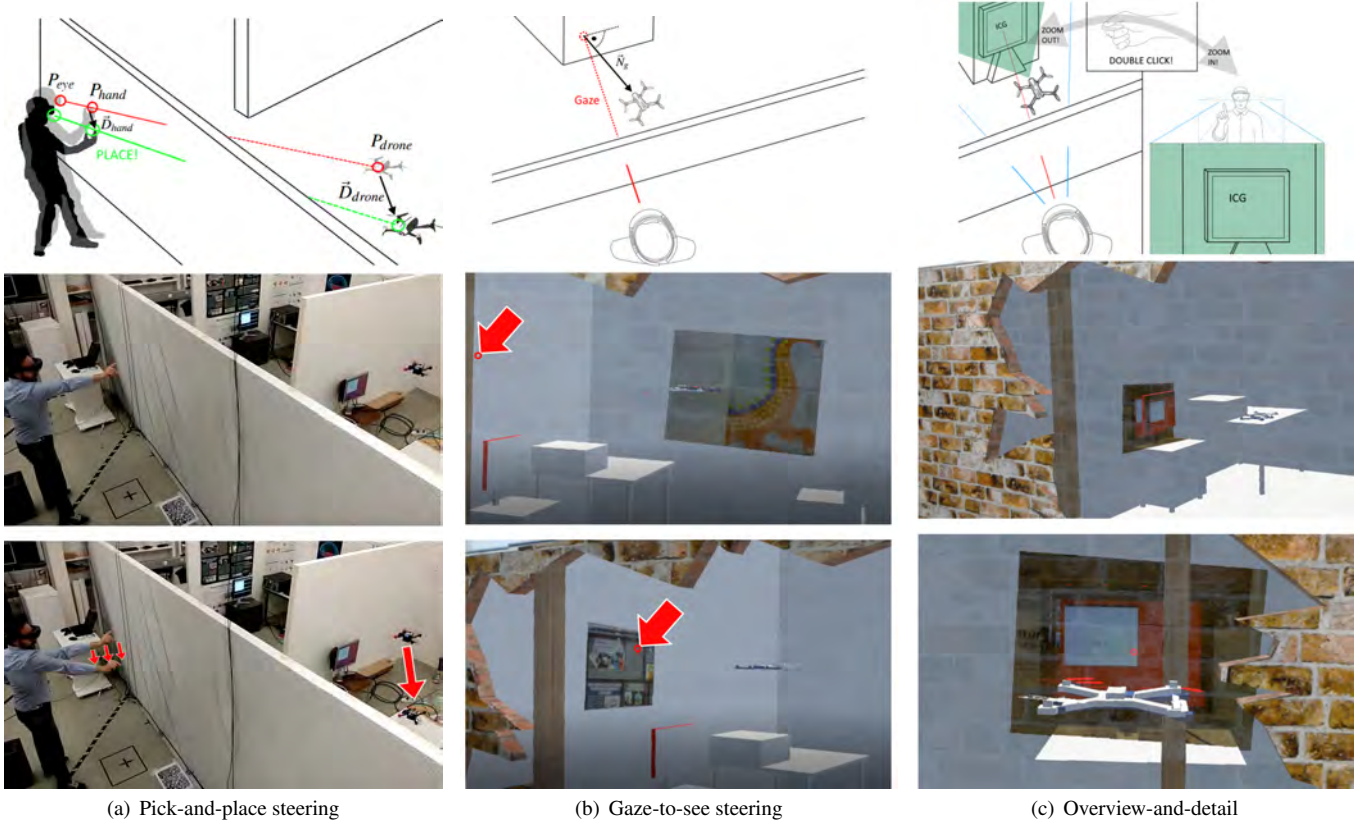(a) Pick-and-place steering   (b) Gaze-to-see steering   (c) Overview-and-detail

Fig. 2. Interaction techniques of our interface: (a) Pick-and-place – the user picks and places the drone, as if the drone is at the reach of the user's arm.(b) gaze-to-see – the user looks at a point as shown by the red arrows and steers the drone to a position where the drone observes the point of interest and ensures a close up view. (c) Overview-and-detail – when a far point in the scene is investigated, the user can virtually fly close to the point of interest and have a virtual viewpoint in the scene. However, all virtual scene elements remain behind the wall to avoid confusing depth perception.

Neumann et al. [25] proposed the idea of surveillance based on augmented video environments, which rely on projective texture mapping of live video to a reconstruction of an outdoor environment. For indoor surveillance, presenting the video streams in the context of a spatial model rather than via a more conventional multi-windowing display was explored by Wang et al. [33].

These systems assume an observer in a control room, but similar ideas have been explored for mobile users. Kameda et al. [17] report on a mobile AR system displaying registered video streams from remote cameras. Avery and Sandor [2] use ghosted-view X-ray vision to look through walls. Their system shows videos received from a remote robot controlled by the user via joystick [1]. Sandor et al. [29] later proposed a "melting" metaphor to disocclude buildings. Sandor et al. [28] show a method for X-ray rendering using salient features of occluders.

Another aspect of remote information display is camera navigation. For example, Mulloni et al. [24] describe how to transition between the video from multiple cameras placed in an outdoor environment without loosing spatial context. Hoang et al. [15] investigate remote viewpoint manipulation for close-up observation. We draw inspiration from all of these methods, but additionally control the flight path of a drone indirectly by introducing interaction techniques for the interactive definition of the desired viewpoint.

3 INTERFACE DESIGN

Our drone-augmented human vision system lets the pilot control a drone inside an occluded space indirectly, via an exocentric visualization provided in a see-through HMD (Microsoft HoloLens). While the drone travels in the remote environment, the video frames streamed from the on-board camera are projectively texture-mapped onto a geo-

metric model of the scene. The scene is rendered from user's current perspective, as measured by the built-in self-localization of the HMD.

In addition, a virtual representation of the drone is rendered at the position reported by the physical drone, to give the pilot an overview of the physical configuration of the occluded space. The interior scene with partial texture mapping is made to appear inside a "cutaway" magic lens that appears as a hole in the occluding wall structure.

For flight control and navigation in the occluded space without hitting obstacles, we introduce two interaction techniques, called *pick-and-place* and *gaze-to-see*. Moreover, we introduce *overview-and-detail*, a transitional interface [4] to reveal details on demand.

3.1 Pick-and-place

This interaction technique allows users to pick a drone by looking at it and applying a pinch gesture. After picking the drone, moving one's hand repositions the drone in 3D space, as illustrated in Figure 2a. The hand movement is scaled proportionally to the distance of the picked object, as in the scaled-world-grab technique proposed by Mine et al. [22]. More formally, the displacement vector $\vec{D}_{drone}$ of drone's position P_{drone} in $\mathbb{R}^3$ is calculated as

$$\vec{D}_{drone} = \frac{\left\| P_{drone} - P_{eye} \right\|}{\left\| P_{hand} - P_{eye} \right\|} \cdot \vec{D}_{hand}$$

where P_{eye} and P_{hand} represent the positions of the eye and the hand, respectively, while $\vec{D}_{hand}$ indicates hand's motion vector in $\mathbb{R}^3$ (Figure 2a). P_{eye} and P_{hand} are directly provided by HoloLens, whereas P_{drone} is received from the drone tracking system. Note that, depending

on factors like dominant eye, HMD position on the head or the distance of the currently focused object, P_{eye} measurement may be subject to brittle calibration. However, during our experiments, users did not indicate that they needed (re-)calibration.

3.2 Gaze-to-see

Using the view vector and eye position provided by HoloLens, one can calculate the point of interest P_{gaze} a user is gazing at by intersecting the viewing ray with the scene model. Knowing gaze position allows to predict which part of the occluded scene a user is interested in. Therefore, in this interaction technique, the drone focuses on the high level goal of the user and automatically repositions to observe the area around the user's point of interest with its on-board camera. Let $\vec{N}_g$ be the normal vector at P_{gaze}, and let $\vec{Z} = \{0, 0, 1\}$ denote the up-axis of the scene. The drone is positioned at

$$P_{drone} = P_{gaze} + \frac{\vec{N}_g - (\vec{N}_g \cdot \vec{Z}) \cdot \vec{Z}}{\left\| (\vec{N}_g - (\vec{N}_g \cdot \vec{Z}) \cdot \vec{Z}) \right\|} \cdot x$$

$$if \left\| \vec{N}_g \cdot \vec{Z} \right\| < \left\| \vec{N}_g \right\| \cdot 0.9$$

Unless we are looking at a horizontal surface, the drone will reposition x meters away from the point of interest along a displacement vector corresponding to the surface normal projected to a horizontal plane (Figure 2b).

In our experiments, we set x to 0.5 meters for ensuring a close-up view of the surface. The drone's yaw orientation is adjusted to align with the negative displacement vector. In case the user looks at a horizontal surface, the drone is positioned between the user and the point of interest, mimicking the user's view vector in the horizontal plane. If the calculated position is not inside the safe flight zone, the repositioning terminates at the nearest border of the permitted flight zone.

3.3 Overview-and-detail

By visualizing the occluded scene and the drone from user's perspective, our system allows a drone pilot to better understand the spatial relationships between scene geometry, drone and the pilot's body. However, this visualization lacks details, as the drone can be far away and both camera and display suffer from a rather limited field of view. Therefore, we introduced an overview-and-detail technique, which fills the gap between egocentric and exocentric drone control modes in the form of a transitional interface [4] using image-based warping [32]. After steering the drone to a point of interest, users are given option to either virtually move closer to the drone or to the currently gazed-at surface point in the occluded scene, by selecting the corresponding interface hotspot. During the detail visualization, we apply the occluding wall structure to clip the zoomed detail geometry in order to avoid confusion between real and occluded virtual geometry (Figure 2c). Zooming in is achieved by positioning the virtual hole in front of the gazed point while preserving the relative transformation between the virtual hole and the user's camera view. The position of the virtual hole is computed in the same way as the positioning of the drone in gaze-to-see interaction.

4 IMPLEMENTATION

A detailed overview of our experimental system architecture, including hardware and software components and data flow between them, is shown in Figure 4. The system builds on the drone design described by Isop et al. [16] and is based on six main components:
We use an (1) Optitrack motion tracking system consisting of a server system with 12 cameras to externally localize the drone. The Optitrack is connected to a (2) ground-station, which further communicates to the (3) drone's on-board computer, an Odroid XU3, and the (4) HoloLens via WiFi. For our user study, we complemented the system with a remote control user interface including a (5) joypad for steering and a (6) visualization station. All components communicate via Ethernet or WiFi.
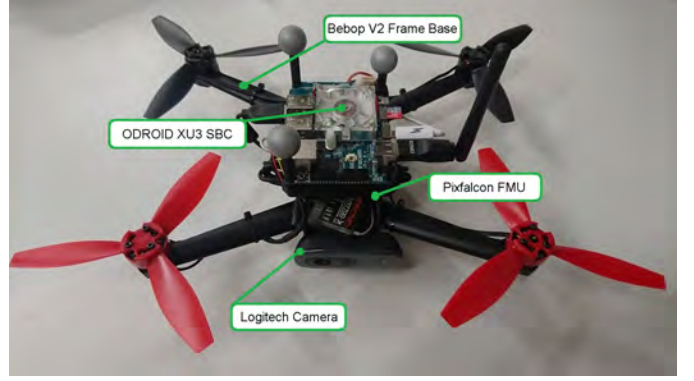

Fig. 3. Experimental drone setup including the main components. The camera is mounted facing forward. Inside the frame, the autopilot is located. The battery is mounted on the bottom to balance weight distribution. The onboard computer is located on top.

The software components are integrated via ROS [26] nodes. We use Unity 3D for visualization on the HoloLens and the ROS tool *RViz* for monitoring on the ground-station.

The motion capturing node on the ground-station relays UDP packages from the Optitrack system, which describe timestamped poses of the tracked objects, to the Odroid. The Odroid transforms the poses into local coordinates of the drone and forwards them to the MAVROS node, a ROS wrapper to communicate with the PX4 via publish/subscribe messages. It is responsible for acquiring IMU data, pose updates, target coordinates (setpoints), internal pose estimates, etc.

The drone controller node on the HoloLens maps gestures into target drone position and visualizes the drone's current position and target positions in the mixed reality view. Setpoints can be generated either by the HoloLens interface or by the joypad interface.

4.1 Drone setup

The drone (Figure 3), which has a frame with 25cm diameter and weighs 450g, uses a semi-customized design with rotors and frame taken from a Parrot Bebop 2 platform. The flight time is about 11-15 minutes, while running all relevant components and tasks. We added a PX4 Pixfalcon autopilot as a low-level flight control unit and an Odroid XU3 single-board processor computer.

The forward-looking camera captures image data at 30Hz with 640x480 resolution and delivers it to the Odroid via USB. The video is streamed to the HoloLens in MJPEG format, annotated with timestamp and camera poses to allow precise image-based rendering. All high-level tasks, including processing of image data, estimated poses from the motion tracker and control commands received from the pilot run on-board and are implemented in the ROS framework.

4.2 Flight management control

For localization of the drone, we use the external Optitrack system with 12 ceiling-mounted tracking cameras, covering an area of roughly $5 \times 4 \times 3m$. The Optitrack system provides pose estimations at 120Hz, which are delivered over WiFi to the Odroid at a latency of ˜25ms. The serial link from the Odroid to the PX4 adds another ˜10ms of delay. The system time between Optitrack server, ground-station and PX4 is synchronized based on NTP using the Chrony service.
Designing a drone for autopilot-controlled flight at low heights in small confined spaces is challenging, because of the imminent danger of hitting obstacles. We combined several measures to ensure safe operation. Since high flight speed was not a primary goal, we used low-thrust engines (taken from a Bebob 1 platform) and soft materials for the propellers. This produces less turbulences when flying close to walls and around objects. We further relied on the ability of the PX4 to use pure inertial navigation for short periods, when the measurements from the motion capture system are noisy or intermittent. The pose
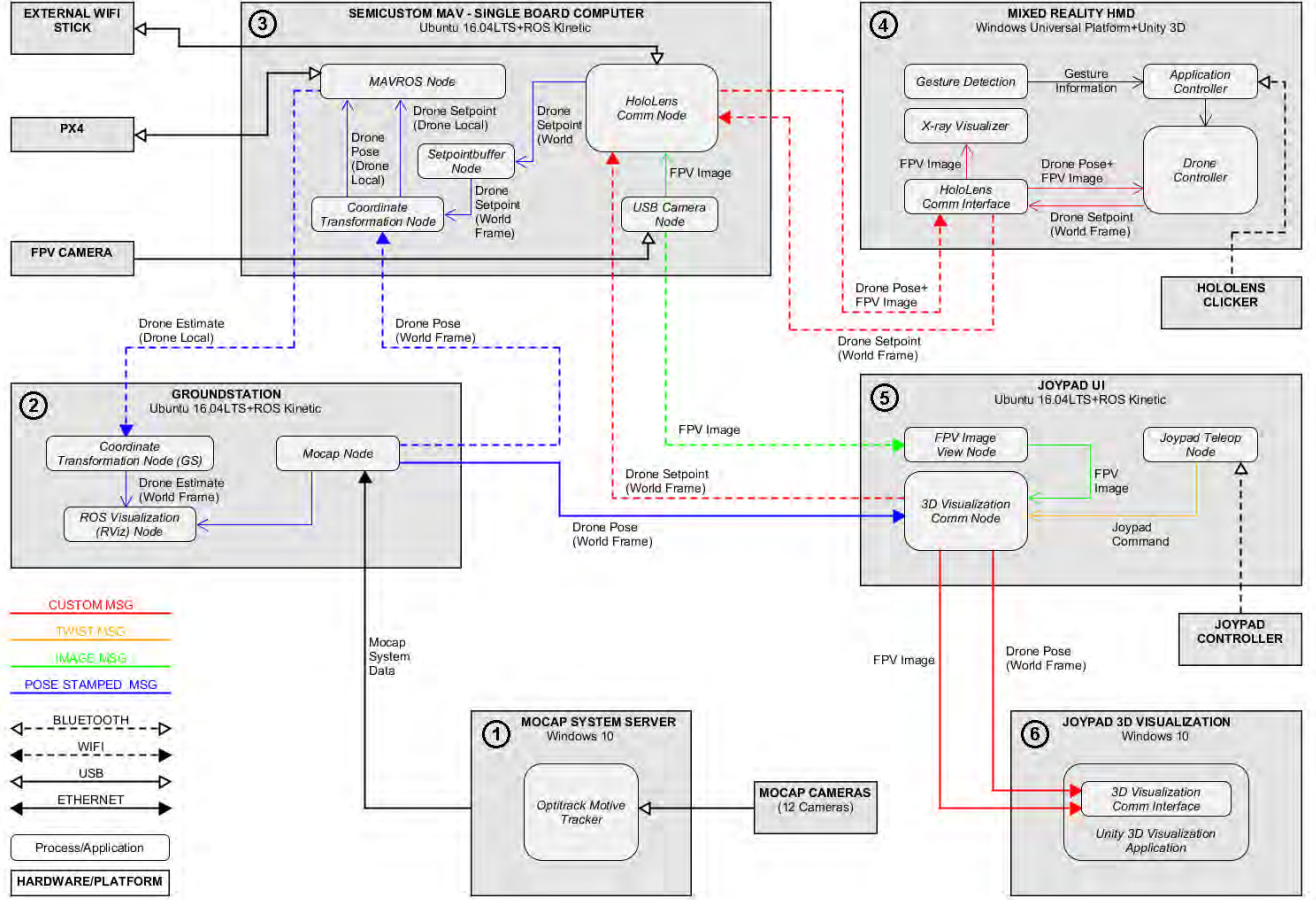
Fig. 4. Overview of main hardware and software components of our experimental setup including (1) the motion tracking system, (2) a ground-station, the (3) on-board computer of our semi-customized drone, (4) the HoloLens, (5) a joypad interface, and (6) a visualization station.

updates are buffered on the Odroid to minimize the occasions where the PX4 switches unintendedly from autonomous flight mode into manual mode if the Wifi link stalls or drops position updates from the Optitrack.

4.3 Control of drones movements

Control of the drone is based on measuring its 6DOF pose by the motion capture system in world coordinate representation. We make use of the PX4 inertial estimator to fuse the motion capture data with the inertial sensors of the PX4, deriving 3D position $[x, y, z]$ and the yaw θ required for the drones's position control. These measurements, obtained at discrete times $i = 0 \ldots n$, are denoted as Y_i.

For position control, we use the internal linear control approaches of the PX4. The methods consist of an inner attitude rate PID (proportional/integral/derivative) controller with pitch, roll and yaw angular velocities as inputs. This control loop is enclosed by an attitude P-controller with attitude setpoints for roll, pitch and yaw angles and throttle as reference input. The inner control loop is nested in a position control loop, which takes 3D position $[x, y, z]$ and yaw θ as reference inputs H_i, which can, for example, be derived from the HoloLens interaction. The yaw reference is directly fed into the inner attitude control loop.

$$
\begin{aligned}
Y_i &= \{x_i, y_i, z_i, \theta_i\} \quad (i = 0...n) & (1) \\
H_i &= \{x_i, y_i, z_i, \theta_i\} \quad (i = 0...n) & (2) \\
E_i &= H_i - Y_i & (3)
\end{aligned}
$$

The derived position error, given in Equation 3, is calculated in every iteration i and fed into the control structure of the PX4. We use aggressive controller gains, which are based on the default gains of the more heavyweight DJI F330 model, to establish fast response times

and accept slight overshooting of approximately 5%, when the drone's actual position converges towards the given setpoint.

4.4 Precomputed path planning

For our experiments, we wanted to relieve the pilot as much as possible from path planning, providing the illusion of augmented vision without concerns about flight safety. However, a fully featured path planning is computationally expensive and can be brittle. Since we track the drone externally, rather than by SLAM, we can pre-compute the necessary path planning information from the floor plan. In our test environment, we divided the space into three regions, two rooms connected by a corridor.

If the pilot issues a repositioning command that requires changing the region, the path planning first approaches a predefined waypoint at the boundary before progressing to the neighboring region. Overall, our path planning is simplistic, but works instantaneously and reliably prevents accidents due to hitting obstacles or walls of the scene. A more realistic path planning based on SLAM would run an A* algorithm on a map of the environment that has already been explored by the drone.

4.5 Joypad control

Alternatively to the path planning, the drone can be controlled via a joypad. In this case, a custom ROS node integrates the inputs from four axes of the joypad and converts them into a 3D position and yaw of the drone. We derive the position reference commands by integration of the joypad's linear axis commands J_i over the time intervals between discrete times i. The position error E_i in this case is given as $E_i = J_i - Y_i$.

To enable a fair comparison between the exocentric interaction techniques introduced in section 3 and the joypad interface, we added

advanced features to the joypad interface, which go beyond what is conventionally available in commercial drone control.

First, we provide drift-free stabilization of the MAV position during navigation in the scene. This kind of stabilization is not available when using off-the-shelf drone technology. Conventional tracking and stabilization, especially in the x-y plane, is usually based on optical-flow or inertial sensors, which suffer from drift over time. With the drift-free tracking, we also enable a basic level of disturbance rejection against turbulences which occur during flight in narrow parts of the scene.

Second, we chose Mode-2 axis mapping on the joypad, which is a well-known and widely accepted mapping for drone control. It is also the default configuration in a variety of off-the-shelf drone products, e.g., the Parrot AR Drone 2.0, the Parrot Bebop 1/2, and the DJI Marvic. Mode-2 mapping employs the left joystick for commanding vertical velocity and velocity around the rotational z-axis of the drone. No direct thrust control is required by the user, reducing cognitive load. The right joystick controls the translational velocity in X and Y direction.

Third, we created a safe-guard for the use by introducing artificial boundaries inside the scene, so the user is not able to crash the drone into walls or hit obstacles. Before each experiment, the user was informed that crashing the MAV is not possible. We presented visual feedback when the user hits the artificial boundaries via warning message, and we visualized the valid flight areas inside a 3D perspective view with green bounding boxes (Fig. 7). If the user hit the boundaries, the drone did not fully stop, but continued movement along the boundary with the resulting speed vector. Thus, the user was able to "slide along" the artificial boundaries. Another safety mechanism allowed the joypad user a simple and safe transition between the rooms. Once the user approached the narrow corridor between the rooms, the drone was automatically transported to the other room. We did not impose any limit in z-direction, so the user was able to safely transit between the rooms at any flight height.

4.6 Head-mounted display

The pilot interface runs on the HoloLens. Its tinted visor holds transparent combiner lenses, in which projected images are shown to the user. We rely on the built-in SLAM system of the HoloLens to provide continuous self-localization. In order to register the localization data reported by the HoloLens with the Optitrack coordinates (OC), we use a Vuforia tracking target. The tracking target is placed on the floor in front of the occluding wall, which corresponds to the plane $Z = 0$ in OC. The transformation between OC and tracking target was calibrated offline. Using the Vufario SDK for HoloLens, we obtained the transformation from the origin of the HoloLens SLAM tracking to the tracking target at startup time and concatenated to the OC transformation. Thus, a drone pose reported in OC can transformed into HoloLens coordinates.

4.7 X-ray vision

We apply AR X-ray vision while providing the user with an exocentric interface for nearby remote scenes. We use the Unity 3D game engine for rendering the scene geometry on the HoloLens. A stencil masking technique is applied to render X-ray visualization only where the virtual geometry is observed through the virtual hole in the wall.

Images for first-person view are streamed from the drone-mounted camera as MJPEG, annotated with the drone's pose when the frame was taken. The MJPEG is decoded and uploaded as a texture to the GPU of the HoloLens to generate the Mixed Reality view.
For each fragment displayed on the HoloLens, the texture is sampled during the shading process by projecting fragment positions in world space with the view projection matrix of the drone's camera. In order to eliminate virtual geometry from being rendered between the occluding wall and the user, fragments with world coordinates that are located behind the wall plane are discarded.

5 USER STUDY

We conducted two user studies to collect quantitative and qualitative data on the performance and scalability of our system.

5.1 Physical viewpoint study

First, we were interested in the users' spatial awareness using the exocentric viewing interface and X-ray vision, compared to a standard egocentric interface that lets the pilot control the drone with a joypad. Specifically, we studied the case in which the user is in-place investigating the occluded scene, which is close (e.g. behind a wall) but cannot be reached from the current viewpoint. To ensure a fair comparison, we supported the joypad user not only with the live egocentric video from the drone, but also with a screen-based 3D visualization of the hidden space, showing real-time updates of the drone's position. We formulated our hypotheses as follows:

H_1: *"Steering a drone for collecting information in distant spaces is faster with the exocentric interface than using a common joypad interface."*
H_2: *"Steering a drone for positioning in distant 3D spaces is faster with the exocentric interface than using a common joypad interface."*
H_3: *"Steering a drone for collecting information and positioning in distant 3D spaces is more intuitive with the exocentric interface than using a common joypad interface."*

Study design and tasks   In order to test our hypotheses, we chose interaction mode as an independent variable with two conditions: Exocentric interface (EXO) and egocentric interface (EGO). In addition, we selected completion time as a dependent variable. Workload was measured using NASA TLX [11], and overall preferences of the users were assessed via semi-structured interviews. Based on a within-subjects design, participants were given two instances of a search-and-explore task to be accomplished with either of the interaction methods, in randomized order.

Reading text on monitors   We asked subjects to steer the drone with both interfaces and report random texts displayed on two monitors positioned in different places of the occluded space. The monitors showed different background colors (red and green) to uniquely identify them from an arbitrary distance. During the training, users were informed with the positions of the red and green monitors in the 3D environment model. The environment model contains three physical monitors, but only two of them were active at any time in order to necessitate different flight paths (Figure 5). The time spent to read from each monitor was recorded as soon as the user reported the text correctly. We asked participants to use the gaze-to-see interaction technique and we suggested to additionally use the overview-and-detail technique in EXO.

Positioning of the drone   In this task, participants were expected to position the drone at three known target locations, which were visualized as boxes in the 3D models shown in both interfaces (Figure 7b and Figure 6). We logged the time spent to visit the target locations, whenever the system reported that the drone approached a target to
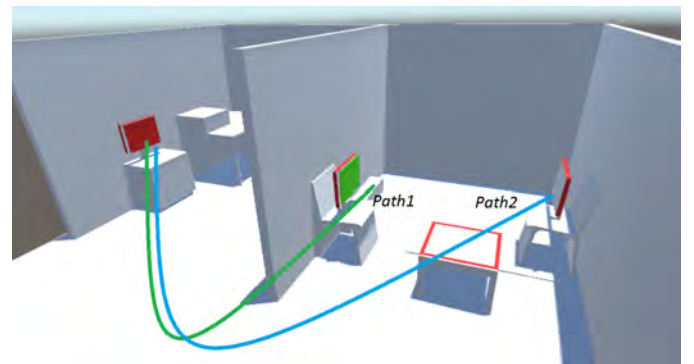

Fig. 5. By altering the position of the green monitor, two different flight paths are generated per user.

within 10 cm tolerance. As the task involved accurate and fast positioning of the drone for this tasks, we suggested to the EXO users to use the pick-and-place technique.

Participants   Ten participants (0 female, $\overline{X}$ =23.1 (sd=2.07) years old) volunteered in our experiment. All of them had extensive experiences with mobile devices, none was a regular drone pilot.

Experimental setup   Participants performed the tasks while standing in front of a wall completely occluding the flight zone. In the EXO condition, participants wore a HoloLens for seeing through the wall. In the EGO condition, a joypad was used to steer the drone, while a monitor (19 inch) was used to display the video stream delivered by the camera of the drone. EGO users were also provided with 3D views of the flight zone from different perspectives (top and top-side view), displayed on a second monitor (15 inch) (Figure 7). A laptop was used to record the participants' qualitative and quantitative input during the experiment.

Procedure   Participants were brought to the participant zone and informed about the setup of the experiment environment without giving detailed information about the flight zone. After the briefing, we assessed their demographics and explained how to use both interfaces. Participants were allowed to practice both interfaces, until they expressed confidence to use them.

Participants were asked to accomplish the tasks in randomized order, to eliminate training effects. For the text reading task, the position of the green monitor was changed to alter the flight path from the first to the second condition. After finishing each task with one interface, participants filled in the NASA TLX. Upon the completion of all tasks for both interfaces, participants filled out a preference questionnaire, and a semi-structured interview was conducted. Sessions lasted 50 min.

Results   The task completion time was evaluated using paired t-tests, and the TLX data was analyzed using pairwise Wilcoxon signed-rank tests. The t-tests revealed significant differences between HoloLens and Joystick interface for both, the reading task (p=0.001347) and the reaching positions task (p=0.002369), in terms of task completion time. On average, task completion times were less than half of EGO (Figure 8a). In the text reading task, EXO took 19.85 seconds average (standard error 2.2 seconds) to read the texts on both monitors, whereas EGO took 39.1 seconds on average (standard error 6.3 seconds). For reaching the given 3D positions, EXO users completed the task on average in 34.2 seconds (standard error 2.3 seconds). EGO took 73.4 seconds on average (standard error 9.57 seconds).

According to the overall scores of the NASA TLX forms, for both of the tasks, users found EXO to have a slightly better usability than EGO. For the first task, users gave an average score of 24 for EXO and 32 for EGO, whereas, for second task, EXO scored 25 and EGO scored 30 (Figure 8b). Probably due to the small number of participants, the

Fig. 6. As part of positioning the drone task, target positions are visualized as yellow boxes in the EXO interface.

TLX data did not show significant differences between the interfaces. However, we found a noticeable trend in the TLX data towards the HoloLens interface for the reading task (Z=1.68, p=0.105).

Relatively high deviations in task completion time of EGO suggest that EGO requires a good 3D interpretation or experience with joypad control. In contrast, EXO seems to efficiently leverage human abilities, resulting in consistent performance, specifically for pick-and-place.

In the informal feedback during the post-interview, users commented on their preferences. All the participants stated that they would prefer EXO for the given tasks or similar task for investigation of the occluded space. Verbal feedback from the interviews for both conditions included:

- I felt more confident of being precise when using EXO, specifically using pick-and-place.
- I was feeling inside the scene with EXO.
- Depth feeling was amazing with EXO.
- I confused my orientation with EGO.
- I couldn't decide which view to concentrate on with EGO.
- Pick-and-place was cool, natural and accurate.
- Observing the drone from a distance, but still being able to get close to it, was pleasant.

On EGO, several users commented that the joypad axis confusion between drone's local frame and global frame during steering was difficult. They also sometimes confused buttons, a problem that may be overcome with longer training. Nonetheless, the direct manipulation in EXO was more easily adopted. Users also criticized the limited field of view of EGO and reported a confusion of heights. Finally, they found that they could not easily decide which view (camera image or perspective views) to concentrate on.

On EXO, one user stated he preferred the precision of the joypad interface for collecting boxes, and several users found the HoloLens pick gesture inconvenient. However, both comments were likely caused by the unreliable gesture detection provided on the HoloLens. We hope that a future update of the HoloLens SDK will include a more stable gesture detection, which directly will make our pick-and-place interface appear more convenient and more precise. In summary, the results of our experiment allow to accept H_1, H_2, Furthermore, we partially accept H_3 based on the trend towards EXO provided by the user comments and the data retrieved from TLX questionnaires.

5.2   Virtual viewpoint study

The physical viewpoint experiment demonstrated the use of exocentric interaction techniques at close distances. If the drone is further away from the user, drone control by hand gestures obviously becomes increasingly sensitive to fine-motor control of the hand and to tracking errors. We empirically verified that, indeed, satisfactory drone control with gestures is not possible at distances of 20m or more.

However, since our exocentric (X-ray) interface uses the physical environment – the brick wall – only to provide relative motion cues to the user, a VR interface using the same setup is also possible. In VR, the head-mounted display is operated in a non-see-through mode, and the user is placed in a purely virtual environment, with the exception of the texture-mapped remote video stream. This setup can always place the user's virtual viewpoint in convenient proximity to the drone to allows direct manipulation. The VR interface is also necessary if physical proximity to the drone is not possible, for example, in dangerous environments.

We speculated that the virtual viewpoint (VV) interface would perform similar as the physical viewpoint (PV) interface (the latter is essentially the same as EXO in the previous experiment). We formulated our hypotheses as follows:

H_4: *"Users will perform similar in terms of execution time for a virtual viewpoint as for a physical viewpoint"*

H_5: *"A virtual viewpoint does not affect how a user completes the tasks, while being away from the scene"*

We tested these hypotheses by repeating the previous experiment with VV and PV conditions, as follows.
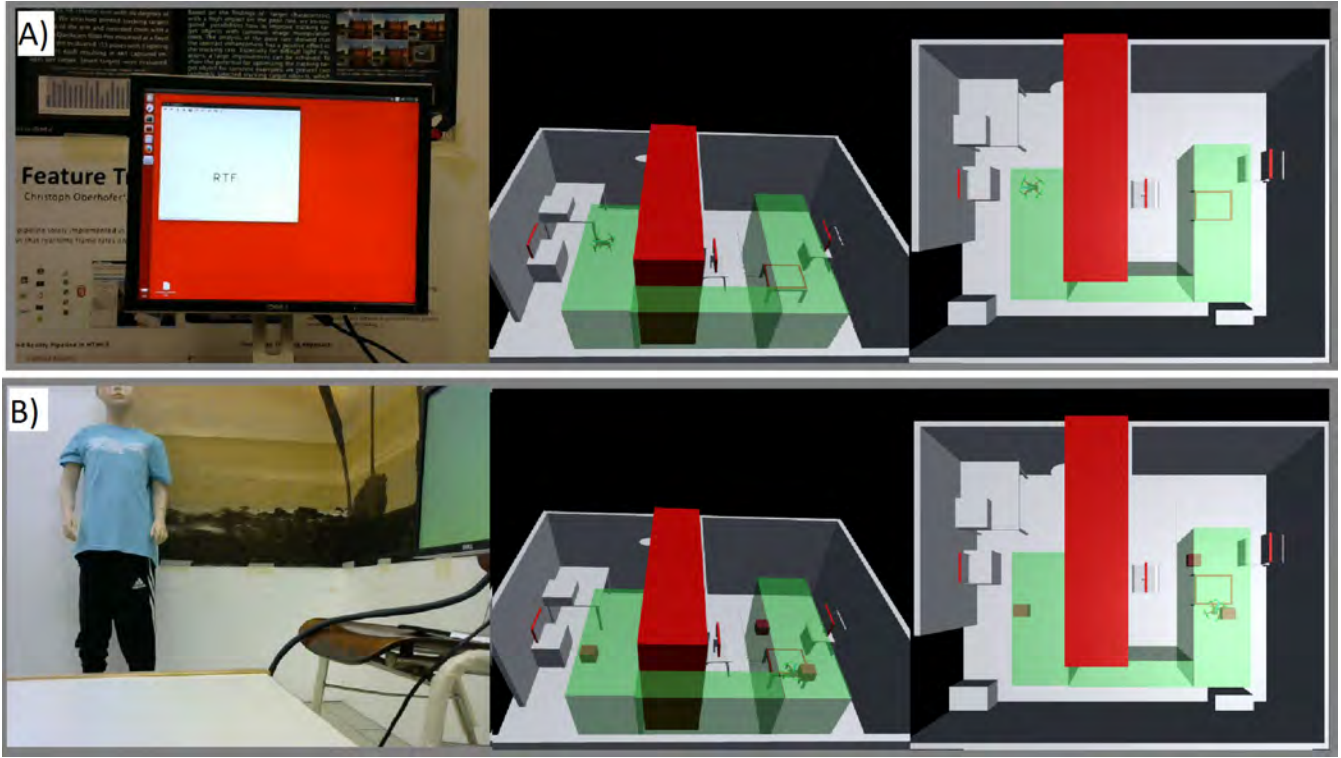
Fig. 7. The first-person view and two additional views of the flight space were available for EGO user during both tasks. The red zone in the middle of the 3D model indicates a restricted flight zone, where drone's position is confined to remain inside boundary, while the green zone delimits the allowed flight space. (a) The user's view while engaged in the screen-reading task. The model shows three screens with red border, but only two of them are active per user. (b) The user's views during the drone positioning task. Cubes in 3D model indicate the target positions to be reached by the drone with a tolerance of 10 cm.

Procedure In VV, participants performed the tasks while standing completely away from the occluded space. The visor of the HoloLens was entirely covered with a blinder to disable its see through display nature and turn it into a VR device. At the beginning of the experiment, VV users witnessed an animated camera transition from their current physical viewpoint to the virtual viewpoint at the remote location. The animation gave them the impression of flying to the target zone and landing where they had to perform the experiment.

In contrast, PV users were standing just behind the occluding brick wall like in the physical viewpoint study. Compared to the first study, we had a slightly larger flight space with the same floor plan characteristics. In the virtual viewpoint study, again ten participants (0 female, $\overline{X}$ =27.5 (sd=2.33) years old) volunteered in our experiment. All of them had extensive experiences with mobile devices, none was a regular drone pilot (different subjects from the physical viewpoint study).

Results In the text reading task, the PV condition took 48.62 seconds average (standard error 2.5 seconds) to read the texts on both monitors, whereas the VV condition took 43.37 seconds on average (standard error 2.9 seconds). For reaching the given 3D positions, PV users completed the task on average in 44.03 seconds (standard error 3.72 seconds). VV took 41.21 seconds on average (standard error 1.53 seconds). It should be noted that flight times are slightly increased compared to the first study due to the enlarged space and longer paths.

According to the overall scores of the NASA TLX forms, for both of the tasks, users found PV to have a slightly better usability than VV. For the first task, users gave an average score of 23 for PV and 26 for VV, whereas, for the second task, PV scored 25 and VV scored 27. While users commented to perceive both systems as almost identical for completing the tasks, they reported to prefer the PV condition more due to its see through-visualization capability.

The results let us accept H_4 and H_5.

6 DISCUSSION

We propose using real scale interactions for steering remote drones. This enables simple control of the drone with low cognitive effort. Based on the feedback of users and the quantitative results of our experiments, we believe that pick-and-place interaction is useful for quickly positioning the drone when fully automatic navigation is not enough. While wearing the HMD, users have stereo vision to perceive depth. In addition, users can quickly change their viewpoint by simply moving around in a natural way to understand where an object is located in 3D. In contrast, a traditional desktop interface requires several scene manipulations to understand the 3D position of an object in the scene, especially when the object is floating in the air. Simple and natural exploration of the position of the drone in 3D space enables quick understanding of spatial relations, which is a fundamental requirement for navigating the drone in 3D.

Our pick-and-place technique uses a single target point to position the drone. While we could continuously sample points along a path defined by the user, we restrict the number of waypoints to a single start point and end point to ensure a precise placement and to avoid unnecessary drone motion. Mapping any user motion directly to the position of the drone would not allow the user to search for the final position, while the drone follows the user's hand motion.

While pick-and-place can be used to precisely place a drone in 3D space, the gaze-to-see technique can be used to continuously explore and search the environment. Gaze-to-see is a high-level, goal-oriented interaction between drone and human with low cognitive requirements. It provides a tool for quickly observing a region of interest without dealing how to position to drone.

Both of our interaction techniques outperform the traditional egocentric interface for controlling a drone. Note that the significant time difference observed between our experimental conditions are not the result of different reaction times, such as the time spent on moving the head when wearing an HMD versus pressing a button on joypad. The differences can rather be largely attributed to the user's efforts towards

fine-tuning the position of the drone to solve the task. For example, finding the correct pose for the drone to read a small text clearly while experiencing motion blur during the movement phase takes more time with EGO. In contrast, EXO users can easily assume a convenient pose thanks to gaze-to-see technique.

Apart from the motion blur, no text rendering artifacts were disturbing the EGO users, as can be seen in (Figure 7a). In contrast, the EXO users experienced both motion blur and slight artifacts due to the limited resolution of HMD (Figure 2c). We expect that, with better HMD quality, the advantages of EXO may even be more pronounced.

Similarly, during the positioning drone task, EGO users had difficulties understanding if the drone was at the correct position from the given perspective views, whereas EXO users quickly identified the right position by virtue of the stereoscopic view.

Despite the good performance of EXO, we noticed a number of limitations during the experiments, which we describe in the following, along with recommendations for overcoming them based on our experience with the system.

Limited resolution. Our placement precision depends on the distance. As the drone moves away from the user, the increased distance affects the precision of pick-and-place. In addition, when the surface is far away from the user, it is hard to gaze at it. This provides a challenge for selecting the drone with pick-and-place interaction, and it makes it harder to position the drone in front of the right surface during gaze-to-see interaction. This limitation arises, as humans cannot keep their head stable at millimeter-level accuracy. These limitations are solved when the user is virtually teleported to a viewpoint close to the drone, as demonstrated by virtual viewpoint study. In fact, the virtual viewpoint technique can be seen as a generalization of the overview-and-detail technique. The user can always use the VV mode to virtually move closer to the drone and thus increase the precision. The blinder on the visor may not even be necessary, as implied by user's preferences.

Projection error. When the outside in tracking is not precise enough, misregistration causes the projected images do not line up properly with the 3D model. In addition, if the poses are not synchronized with the camera images, the error is further increased. However, these problem can be overcome by better tracking, ideally incorporating dense reconstructions obtained in real time from a drone equipped with suitable sensors, such as structure-from-light sensors or stereo cameras.

Tracking error. Depending on the tracking accuracy of the system, the drone may position itself slightly off the target destination, although the results would be still visualized as if the drone was at the correct location. During tasks requiring accurate spatial positioning, such as drilling a hole at the right spot, the user may be misled. A hybrid interface showing both the exocentric synthetic view and the egocentric video stream side by side may partially alleviate this problem.

Reconstruction error. Gaze-to-see can be strongly effected by a wrongly estimated surface normals, if the 3D model is automatically

reconstructed using structure from motion algorithms. However, many exploration tasks do not require photorealistic rendering and tolerate heavy low-pass filtering of normals to suppress unwanted outliers.

Eye calibration error. Like any ray-picking technique, pick-and-place performance is affected by eye calibration. Without a good estimation of the eye position, any deviations of physical eye and virtual camera will be magnified by the projected distance, letting the picked virtual drift from the hand after some displacement. During our experiments, we noticed that users coped with such situations by simply releasing their grip and quickly re-picking the drone, essentially improvising a form of clutching to minimize the aggregation of unwanted drift.

3D interaction. The mathematics of scaled world grab imply that when the user moves an object away from be body, movement precision will drop quickly. As a remedy, users can re-adjust their virtual viewpoint to move closer to the target location or look at the drone from a different perspective in order to control the drone more precisely. Likewise, if surfaces face away from the user, gaze-to-see requires first assuming a rotated virtual viewpoint to look at the target position.

Aerodynamic restrictions. Drone's aerodynamics restrict it from quickly adapting into a new given position. Therefore, gaze-to-see interaction technique had to be limited to discrete number of position commands instead of continues ones where a new position command is sent each time the user looks to a different surface point.

Selecting the remote scene. Assuming a virtual viewpoint is natural for immersive VR users, while AR user must switch from their physical viewpoint to a virtual one. This can lead to confusion between real and virtual objects. The overview-and-detail technique mostly avoids such confusion, but introduces the restriction that users can only move closer to a point they are already gazing at. While this is sufficient for a number of tasks, choosing a new viewpoint relative to gaze has clear limitations. In particular, gazing becomes less precise and more difficult at larger distances.

However, common techniques such as world-in-miniature [30] (WIM) can be used to easily overcome this limitation. Using a gesture, users can obtain a miniaturized copy of the scene in front of them, in the same orientation as their current viewpoint. It is straightforward to apply scene manipulation techniques from traditional desktop interfaces to a WIM. Users can rotate the WIM towards the desired view and apply clipping or transparency to expose interior structures. They can apply exocentric selection of movement targets in the WIM rather than in the egocentric perspective. In case of a rescue operation, the use of a WIM naturally extends towards a remote control center overview of multiple drones and rescuers from an exocentric perspective.

7 CONCLUSION AND FUTURE WORK

We have developed a prototypical system to discover a remote or occluded scene in an intuitive way by visualizing live imagery streamed from a camera drone in a three-dimensional, exocentric context. To control the exploration, we have implemented experimental high-level interaction techniques that control the drone indirectly, by relating to the enclosing space in which the drone is flying rather than the drone's own local coordinate system and flight parameters, such as speed or altitude. This gives the user the impression of being present next to the drone, or having X-ray vision when using a see-through display. Our experiments confirm that this style of interaction is efficient compared to conventional remote piloting and that it is at attractive for users.

This paper represents only the beginning of our work. While we have argued that drone-augmented human vision is technically feasible, we have relied on simplifying assumptions with respect to several challenging technical aspects: The drone does relies on external tracking for autopiloting and does not reconstruct the environment dynamically. Instead, we use a 3D model of the environment that was prepared offline. Moreover, path planning is simplistic and does not scale up to real unknown locations. Many technical parameters, such as image resolution, flight times, display field of view etc. are not yet satisfactory. Nonetheless, we are confident that ongoing technical developments in several fields will turn out in favor of the proposed system design. We will concentrate on testing and enhancing the interaction metaphors and on enhancing image-based rendering of the remote model.
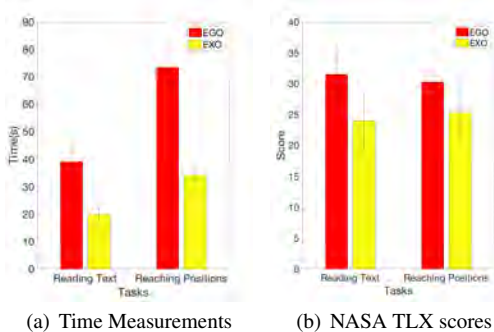

(a) Time Measurements   (b) NASA TLX scores

Fig. 8. Results. a) Average time spent on the tasks with our two interfaces. EXO users performed much faster in both of the tasks, with similar performance, as indicated by the standard error b) NASA TLX scores of the both interfaces for the given tasks.

REFERENCES

[1] B. Avery, W. Piekarski, and B. H. Thomas. Visualizing occluded physical objects in unfamiliar outdoor augmented reality environments. In *Proceedings of ISMAR*, pages 1–2, Washington, DC, USA, 2007.

[2] B. Avery, C. Sandor, and B. H. Thomas. Improving spatial perception for augmented reality x-ray vision. In *Proc. IEEE VR*, pages 79–82, 2009.

[3] L.-P. Bergé, N. Aouf, T. Duval, and G. Coppin. Generation and vr visualization of 3d point clouds for drone target validation assisted by an operator. In *Computer Science and Electronic Engineering (CEEC), 2016 8th*, pages 66–70. IEEE, 2016.

[4] M. Billinghurst, H. Kato, and I. Poupyrev. The MagicBook - Moving Seamlessly between Reality and Virtuality. *IEEE Computer Graphics and Applications*, 21(1):6–9, 2001.

[5] J. Y. Chen, E. C. Haas, and M. J. Barnes. Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(6):1231–1245, 2007.

[6] K. Cho, M. Cho, and J. Jeon. Fly a drone safely: Evaluation of an embodied egocentric drone controller interface. *Interacting with Computers*, 2016.

[7] C. Coffin and T. Hollerer. Interactive perspective cut-away views for general 3d scenes. In *Proc. 3DUI*, pages 25–28, 2006.

[8] S. K. Feiner, A. C. Webster, T. E. Krueger, B. MacIntyre, and E. J. Keller. Architectural anatomy. *Presence*, 4(3):318–325, Jan. 1995.

[9] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor mav. In *IROS*, pages 4557–4564. IEEE, 2012.

[10] J. Hansen, A. Alapetite, I. MacKenzie, and E. Mollenbach. The use of gaze to control drones. In *Proc. ETRA*, pages 27–34, 2014.

[11] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human mental workload*, 1(3):139–183, 1988.

[12] S. Hashimoto, A. Ishida, M. Inami, and T. Igarashi. Touchme: An augmented reality based remote robot manipulation. In *Proc. ICAT*, 2011.

[13] K. Higuchi and J. Rekimoto. Flying head: a head motion synchronization mechanism for unmanned aerial vehicle control. In *CHI'13 Extended Abstracts*, pages 2029–2038. ACM, 2013.

[14] J. T. Hing, K. W. Sevcik, and P. Y. Oh. Development and evaluation of a chase view for uav operations in cluttered environments. *Journal of Intelligent & Robotic Systems*, 57(1):485–503, 2010.

[15] T. N. Hoang and B. H. Thomas. Augmented viewport: An action at a distance technique for outdoor ar using distant and zoom lens cameras. In *Proc. ISWC*, pages 1–4, Oct 2010.

[16] W. A. Isop, J. Pestana, G. Ermacora, F. Fraundorfer, and D. Schmalstieg. Micro aerial projector - stabilizing projected images of an airborne robotics projection platform. In *Proc. IROS*, pages 5618–5625, Oct 2016.

[17] Y. Kameda, T. Takemasa, and Y. Ohta. Outdoor see-through vision utilizing surveillance cameras. In *Proc. ISMAR*, pages 151–160, 2004.

[18] C. R. Karanam and Y. Mostofi. 3d through-wall imaging with unmanned aerial vehicles using wifi. In *Proc. IPSN*, pages 131–142, 2017.

[19] S. Kasahara, R. Niiyama, V. Heun, and H. Ishii. extouch: spatially-aware embodied manipulation of actuated objects mediated by augmented reality. In *Proc. ACM TEI*, pages 223–228, 2013.

[20] S. Kasahara and J. Rekimoto. Jackin: Integrating first-person view with out-of-body vision generation for human-human augmentation. In *Proc. Augmented Human*, pages 46:1–46:8, 2014.

[21] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. In *Proc. ICRA*, pages 2992–2997, 2011.

[22] M. R. Mine, F. P. Brooks Jr, and C. H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings SIGGRAPH*, pages 19–26, 1997.

[23] D. Mirk and H. Hlavacs. *Using Drones for Virtual Tourism*, pages 144–147. Springer International Publishing, Cham, 2014.

[24] A. Mulloni, E. Veas, E. Kruijff, and D. Schmalstieg. Techniques for view transition in multi-camera outdoor environments. In *Proc. Graphics Interface 2010*, Ottawa, Canada, May 2010.

[25] U. Neumann, S. You, J. Hu, B. Jiang, and J. Lee. Augmented virtual environments (ave): Dynamic fusion of imagery and 3d models. In *Proc. IEEE VR, VR '03*, pages 61–, Washington, DC, USA, 2003.

[26] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[27] D. Saakes, V. Choudhary, D. Sakamoto, M. Inami, and T. Lgarashi. A teleoperating interface for ground vehicles using autonomous flying cameras. In *Proc. ICAT*, pages 13–19, 2013.

[28] C. Sandor, A. Cunningham, A. Dey, and V.-V. Mattila. An augmented reality x-ray system based on visual saliency. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 27–36. IEEE, 2010.

[29] C. Sandor, A. Cunningham, U. Eck, D. Urquhart, G. Jarvis, A. Dey, S. Barbier, M. R. Marner, and S. Rhee. Egocentric space-distorting visualizations for rapid environment exploration in mobile mixed reality. In *Proc. ISMAR*, pages 211–212, Oct 2009.

[30] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: Interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 265–272, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[31] M. Sugimoto, G. Kagotani, H. Nii, N. Shiroma, F. Matsuno, and M. Inami. Time follower's vision: a teleoperation interface with past images. *IEEE Computer Graphics and Applications*, 25(1):54–63, 2005.

[32] M. Tatzgern, R. Grasset, D. Kalkofen, and D. Schmalstieg. Transitional AR Navigation for Live Captured Scenes. In *Proc. IEEE VR*, 2014.

[33] Y. Wang, D. M. Krum, E. M. Coelho, and D. A. Bowman. Contextualized videos: Combining videos with environment models to support situational understanding. *IEEE TVCG*, 13(6):1568–1575, Nov 2007.

[34] S. Zollmann, C. Hoppe, T. Langlotz, and G. Reitmayr. Flyar: AR supported micro aerial vehicle navigation. *TVCG*, 20(4):560–568, 2014.