SynchronizAR: Instant Synchronization for Spontaneous and Spatial Collaborations in Augmented Reality

Ke Huo, Tianyi Wang, Luis Paredes, Ana M Villanueva, Yuanzhi Cao, Karthik Ramani *
School of Mechanical Engineering, Purdue University
West Lafayette, IN 47907, USA
{khuo, wang3259, lparede, amvillanuevap, cao158, ramani}@purdue.edu

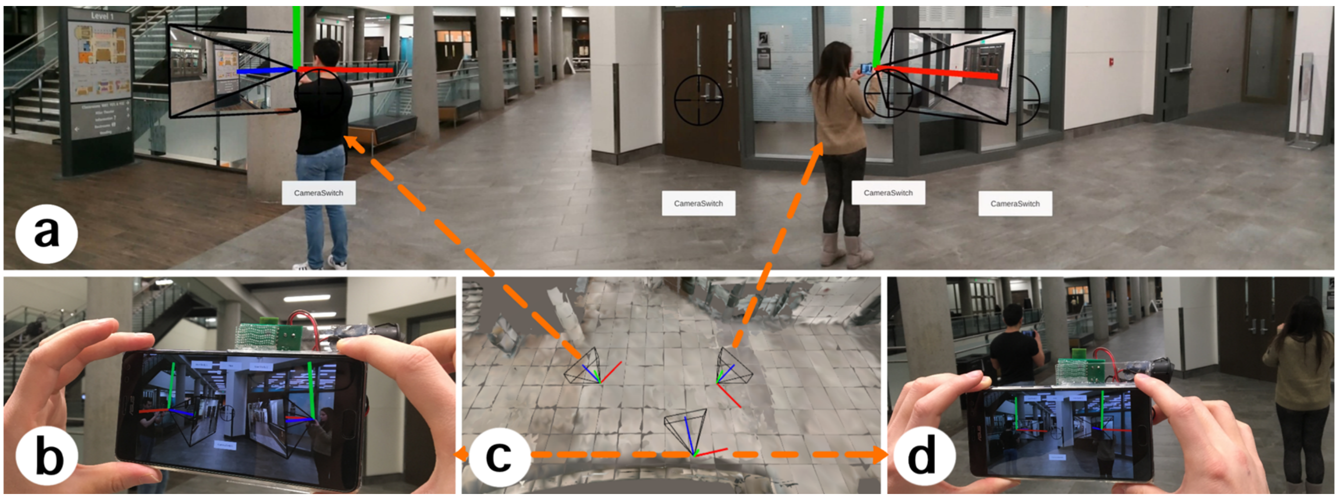

Figure 1. SynchronizAR allows for instant spatial registration among multiple users' mobile AR devices. Three SLAM based AR devices are registered with respect to each other (a, b, d). We enable AR collaboration activities such as spatial aware screen sharing (a) and miniature world navigation (c).

ABSTRACT
We present *SynchronizAR*, an approach to spatially register multiple SLAM devices together without sharing maps or involving external tracking infrastructures. *SynchronizAR* employs a distance based indirect registration which resolves the transformations between the separate SLAM coordinate systems. We attach an Ultra-Wide Bandwidth (UWB) based distance measurements module on each of the mobile AR devices which is capable of self-localization with respect to the environment. As users move on independent paths, we collect the positions of the AR devices in their local frames and the corresponding distance measurements. Based on the registration, we support to create a spontaneous collaborative AR environment to spatially coordinate users' interactions. We run both technical evaluation and user studies to investigate the registration accuracy and the usability towards spatial collaborations. Finally, we demonstrate various collaborative AR experience using *SynchronizAR*.

*School of Electrical and Computer Engineering (by courtesy)

UIST'18, October 14–17, 2018, Berlin, Germany

DOI: https://doi.org/10.1145/3242587.3242595

Author Keywords
Augmented Reality; Collaboration; Spatial Interactions; Registration

INTRODUCTION
Emerging mobile technologies allow augmented reality (AR) applications to become pervasive [23]. Especially, the advancing simultaneous localizing and mapping (SLAM) technique extends the interaction volume into a highly spatial space by providing highly accurate tracking. With SLAM, a mobile AR device is capable of instant self-localizing with respect to the surrounding environment without external tracking setups and prior maps [20, 34].

Involving multiple users in a collaborative co-located environment requires synchronizing spatial frames across different users [6, 47]. This aspect is different from a single-user AR application. To overcome this challenge, researchers often introduce an external tracking system [46, 7] to establish a global shared frame. However, the cumbersome infrastructure counteracts the imperative mobility and immediacy of AR collaboration activities.

A contemporary approach leverages SLAM to create a map of the environment in-situ and share it across users either offline or through a cloud service [2, 10, 36, 52]. Although this approach alleviates the restriction on mobility, it suffers from a laborious global map building process notably in a large space. Recently, researchers have proposed collaborative SLAM methods which automatically share the map in real-time as it expands [16, 24, 35, 45]. Yet, these methods require the users to start roughly at the same position with common views to synchronize the maps initially. This assumption markedly prevents a spontaneous collaboration as it requires specific positions and orientations to start the registration.

The state-of-the-art cloud based AR synchronization solutions essentially rely on a centralized data structure, i.e., a SLAM map contains one or multiple anchors or the full scan of the environment. Instead, we focus on instantly registering multiple SLAM based mobile AR devices without sharing maps or using external tracking setups to support spontaneous AR collaborations in this work.

A direct approach to resolve the peer-to-peer 6 degree-of-freedom (DOF) transformation requires tracking the collaborator's device and estimating its full pose from the local device. One straightforward method is applying vision-based tracking using the embedded camera SLAM device. Unlike the traditional fiducial marker based tracking [3], recent learning based methods have achieved remarkable successes on human/object pose estimations [9, 41, 51]. Yet the vision-based approaches still rely on keeping the collaborator within the local camera view to estimate the pose and derive the transformation. Furthermore, the wearable or hand-held form factors of AR devices demands segmenting them out from images which involve human-device interactions [29, 50]. An electromagnetic based alternative suffers from bulky size and sensitivity to the magnetic distortion in the environment [5, 42, 49].

We present *SynchronizAR*, an indirect synchronization approach which leverages local SLAM results and radio-frequency (RF) based distance measures among the SLAM devices. While the multiple SLAM devices move on independent paths, the distance measuring instances corresponds to the time varying positions in their local SLAM coordinate systems. Then we formulate a distance based registration to resolve the transformation across different local SLAM frames. In specific, we adopt the UWB based time-of-flight distance measuring, as it outperforms existing received signal strength indication (RSSI) based technique using Wifi or Bluetooth in terms of accuracy [32].

In summary, our registration follows a non-central approach by leveraging a self-contained hardware module (i.e., UWB). Comparing with the cloud-based synchronization, we better supports in-situ spontaneous AR collaborations: (i) more flexibility against a dynamic environment (e.g., lighting conditions, objects being moved) and zero cost when shifting to a new environment, (ii) less constraints on users' working zone as no "re-localization" is required, (iii) less dependences on cloud and network especially when Internet accessing is limited, (iv) more compatibility across devices which normally don't share the same perception hardware, SLAM algorithms and map files, and (v) better supports on privacy control when the map contains sensitive information. Here we list the main contributions of this paper as follows.

- An approach to resolving the relative translation and rotation between SLAM based mobile AR devices utilizing UWB distance measurement units.
- Implementation of a spontaneous collaborative AR system enabled by the instant registration and evaluation of the system performance.
- Exploration and demonstration of enabled co-located collaborative AR activities with our prototypes.

BACKGROUND

Collaborative AR Systems

The paradigm of AR has been introduced for both co-located and remote collaborations. Early explorations on co-located scenario such as Shared Space [8] and Studierstube [46] augmented face-to-face collaborative experience with AR. Vita [7] presented a 3D model visualization and manipulation system supporting multiple users. The interaction volume of the pioneer works were restricted by the external tracking setups, e.g., fiducial marker, electromagnetic, inertial, and multi-camera systems. As for remote collaborations, Gauglitz et. al. leveraged SLAM technique to reconstruct a surface model of the local scene supporting virtual navigation in a video conference [18]. Oda et al. proposed to use virtual replicas to assist remote collaboration in AR [38]. Further, researchers investigated telepresence systems to enable life-size dynamic interactions between remote users. Room2Room [40] showed a projected augmented reality system, and Holoportation [39] utilized a head-mounted display (HMD). In a remote collaboration scenario, the interactions either stay loosely connected with the physical scene [39, 40] or constrained within a controlled small volume [38] since the local environment differs from the remote one. With *SyncrhonizAR*, we focus on constructing an shared augmented physical space instantly by synchronizing multiple users' local SLAM coordinates.

Synchronization of Spatial Frames

In a collaborative environment, common spatial references are crucial for communication and coordination [17, 47]. Registering multiple users together within a global frame using external vision based tracking systems have been used in previous works [21, 30, 31, 55]. Other works set up the global frames with different sensor based alternatives including GPS for outdoor environment [44], electromagnetic [42, 49], inertial [7], ultrasonic[19], RF based tracking for indoor scenarios [11]. Besides, registering users to a common anchor scene spatially also derives transformation between users for coordinations. Researchers have used fiducial markers [8] or pre-captured scene images [28] as anchors. Further, with the emerging SLAM techniques, a SLAM map of the shared scene which is offloaded to multiple users allows for flexible and mobile coordinations [2, 10, 36]. Moreover, collaborative

SLAM supports multiple agents to share and build the map in real-time [16, 24, 35, 45]. In our work, instead of sharing the SLAM maps, we emphasize on promoting spontaneous AR collaborations.

Peer-to-Peer Tracking and Localization

The advantages of utilizing the embedded camera on the SLAM based AR device to directly track the pose of the collaborator are obvious. Despite the convenience of avoiding introducing extra components, it has been challenging to accurately estimate the full pose of a wearable or hand-held AR device accurately from images where it is being operated by a user. Other direct tracking alternatives such as electromagnetic sensing [25, 42, 49] are not applicable to mobile AR devices because of the high power consumption and bulky size of the base.

In contrast, the indirect approaches measure distances, angles-of-arrival or RSSI with RF based technologies and then derive the relative transformation between RF units. The indirect approaches have been widely used for wireless sensor network (WSN) localization [32]. Hazas et. al. applied ultrasonic based ranging for distance and angles-of-arrival measurement to derive the 2D localizations of statically placed devices [22]. Gellerson et.al. explored spatial aware mobile user interfaces with similar method [19]. A more recent work demonstrated an approach combining SLAM based mobile AR with UWB units to localize multiple Internet-of-things (IoT) devices distributed in 3D space [26]. Comparing with ultrasonic based sensing, UWB provides much larger sensing ranges with high accuracy [13]. However, these works primarily focused on either multi-user collaboration in a static setup or a single-user interacting with static surrounding devices. *SynchronizAR* contributes towards supporting spontaneous collaboration in general but highlights enabling spatial collaboration activities among freely moving users in AR. Besides, comparing with [26], our work derives not only translational but also rotational transformation between users.

SYNCHRONIZAR

We introduce *SyncrhonizAR*, an approach to instantly register co-located multiple SLAM devices spatially with respect to a shared environment. It is an enabling registration technique which can be used to coordinate the collaborative AR interactions. We attach an UWB unit on each mobile AR device which is capable of self-localizing with respect to the environment using SLAM. During the registration, the AR devices move on different paths correspondingly, and the UWB units measure the distances among the devices as shown in Figure 2. We then derive the relative transformations by solving a distance based optimization problem. In this section, we first describe the general formulation to solve 6 DOF registration between two device. Then we adapt the method according to our realistic requirements.

General Formulation

In Figure 2, each user holds a SLAM based mobile AR device which is equipped with a UWB unit. As we are not sharing the SLAM map, the devices (A and B) yields two independent coordinate systems, i.e., $\mathscr{F}_1$, $\mathscr{F}_2$ respectively.

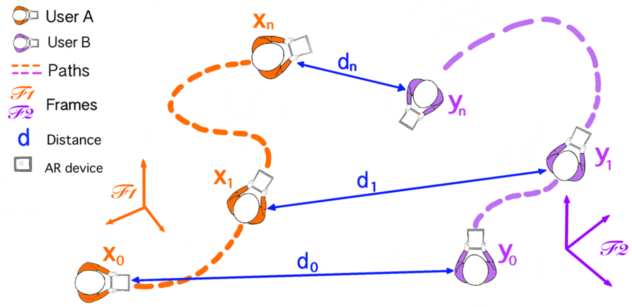

Figure 2. Registration between two users with SynchronizAR.

Without loss of generality, the registration essentially resolves the translational ($\mathbf{T}_2^1$) and rotational ($\mathbf{R}_2^1$) transformation from $\mathscr{F}_2$ back to $\mathscr{F}_1$, e.g., ${}^1\mathbf{x}_i = \mathbf{R}_2^1\,{}^2\mathbf{x}_i + \mathbf{T}_2^1$. As the users are moving, ${}^1\mathbf{x}_i, {}^2\mathbf{y}_i \in \mathbb{R}^3$ denotes positions of A and B at time $t = i$ in their corresponding frames, i.e., $\mathscr{F}_1$ and $\mathscr{F}_2$. The distance between A and B at each time instance while they move on their paths is derived as follows.

$$\begin{aligned} d_i &= \|{}^1\mathbf{x}_i - {}^1\mathbf{y}_i\| = \|{}^2\mathbf{x}_i - {}^2\mathbf{y}_i\| \\ &= \|{}^1\mathbf{x}_i - \mathbf{R}_2^1\,{}^2\mathbf{y}_i - \mathbf{T}_2^1\| \end{aligned}$$

Within the time period $t \in \{1,\ldots,N\}$, we collect the local positions, ${}^1\mathbf{X} = [{}^1\mathbf{x}_1,\ldots,{}^1\mathbf{x}_N]^T \in \mathbb{R}^{N\times 3}$ and ${}^2\mathbf{Y} = [{}^2\mathbf{y}_1,\ldots,{}^2\mathbf{y}_N]^T \in \mathbb{R}^{N\times 3}$ for A and B respectively. At the same time, the UWB units measure the distances $\hat{d}_i$. Because of the distance errors introduced by the measurements, we formulate an optimization to estimate the transformations as follows.

$$\min_{\mathbf{R}_2^1,\mathbf{T}_2^1} \mathrm{S}({}^1\mathbf{X}, {}^1\mathbf{Y}, \mathbf{R}_2^1, \mathbf{T}_2^1) = \min_{\mathbf{R}_2^1,\mathbf{T}_2^1} \sum_{i\leq N} \omega_i(\hat{d}_i - d_i({}^1\mathbf{X}, {}^1\mathbf{Y}, \mathbf{R}_2^1, \mathbf{T}_2^1))^2 \tag{1}$$

where the weight ω_i is defined based on the quality of the measurements. Note, in our current implementation, we simply set the weights equally to be 1.

Optimization with Reduced Dimensions

The general formulation of the problem requires to search solutions in a 6 dimensional space, as our unknown transformation has 6 DOF, i.e., 3 translational and 3 rotational DOF. However, with a close look at the SLAM system, we reduce the rotational DOF down to 1. Modern SLAM implementations on the of-the-shelf devices such as Google Tango and Hololens often leverage the built in inertial measurement unit (IMU). Such a visual-initial approach achieves a robust and accurate motion tracking. As shown in Figure 3, when the device initializes SLAM, a world coordinate system will be created with an origin at the instant position. Also, the orientation of the coordinate system will be compensated by the IMU measurements at the moment so that the $x-z$ plane remains horizontal. That said, we only need to consider the rotation angle θ about y axis. Then we reduce the search dimension from 6 to 4.

Furthermore, we employ a heuristic to constrain the search space with boundaries on the translational y axis. First we observed a simple fact that when a user interacts with an AR

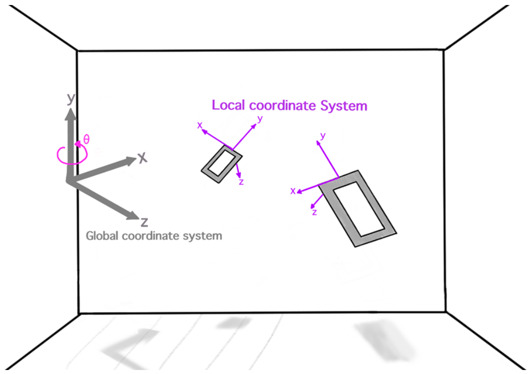

Figure 3. Coordinate system of a SLAM device.

device, the translational movements along y axis are limited considering ergonomic factors such as arm lengths and fatigues. Further, comparing with the movements on x and z axes which can easily reach to dozens of meters, the range on y axis appears a relative small level ($\sim$ 1 m). Besides, for a HMD, moving along y axis is obtrusive and unnatural.

However, for the distance based optimization problems, the flip ambiguity arises easily when the sample positions roughly appear on a plane which implies a irregular distribution, i.e. not a uniform distribution in 3D space [4, 26]. Our heuristic tackles these problems by taking the following steps: (i) initializing the SLAM at a fixed hight ($\sim$ 1.5m above the floor), (ii) constraining the the movements on y axis during the registration, (iii) set the y components of ${}^1\mathbf{X}$ and ${}^1\mathbf{Y}$ to their average values respectively, and (iv) adding boundaries on $\mathbf{T}_2^1$ to the optimization solver. To this end, we adjust Eq. 1 a constrained optimization problem with reduced dimensions as follows.

$$\min_{\theta,\mathbf{T}_2^1} S({}^1\mathbf{X}, {}^1\mathbf{Y}, \theta, \mathbf{T}_2^1) = \min_{\theta,\mathbf{T}_2^1} \sum_{i \le N} \omega_i (\hat{d}_i - d_i({}^1\mathbf{X}, {}^1\mathbf{Y}, \theta, \mathbf{T}_2^1))^2$$
$$s.t. t_{y_{min}} \le t_y \le t_{y_{max}} \tag{2}$$

where t_y denotes the y component of $\mathbf{T}_2^1$, and $t_{y_{min}}$ and $t_{y_{max}}$ are boundaries of t_y.

Scalability

To this extent, we offer an instant registration for spontaneous collaborations between two users. For more than two users, we consider different situations: (i) multiple users form a new collaboration and (ii) one or more users join an existing collaboration. For the first situation, a total number of k users results $k(k-1)/2$ transformations, among which only $k-1$ transformations are independent. For example, with independent transformations $\mathbf{R}_2^1, \mathbf{T}_2^1$ and $\mathbf{R}_3^1, \mathbf{T}_3^1$, we can derive the homogeneous transformation as follows.

$$\begin{bmatrix} \mathbf{R}_3^2 & \mathbf{T}_3^2 \\ \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_2^1 & \mathbf{T}_2^1 \\ \mathbf{0} & \mathbf{1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}_3^1 & \mathbf{T}_3^1 \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{3}$$

We select $k-1$ independent transformations in a manner of one-to-many. Namely, we measure the distances from a single device to the rest of devices within the UWB network. Together with the corresponding local positions, we run $k-1$ times one-to-one registrations. Then we calculate all $k(k-1)/2$ transformations similar to Eq. 3. For the second situation, we select one node from the existing collaboration and perform a registration between the new users and this node only. Again, we propagate the rest of transformations similarly.

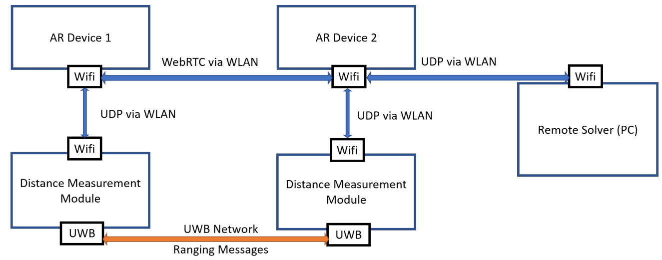

Figure 4. System overview of a prototype example with two AR devices and the distance measurement modules.

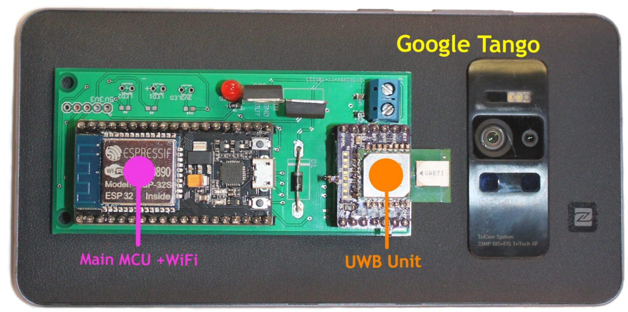

Figure 5. Hardware overview of the prototype. UWB based distance measurement module attached on a mobile AR device.

IMPLEMENTATION

SyncrhonizAR utilizes an indirect distance-based registration and requires no map sharing. As illustrated in Figure 4, our prototype system consists of AR devices, distance measurement modules, and a remote solver (e.g., PC) which were connected to a wireless local area network (WLAN). We developed the self-contained UWB based distance measurement module with off-the-shelf components. Through the UWB network, the distances were measured and packaged to an arbitrary MCU. Then the distance measurements were sent to the AR devices via UDP. The local coordinates of the AR devices have been shared through the WebRTC. Then, a remote solver fetched the sample packages which include the distances and the local coordinates by communicating with one of the AR devices through UDP. Our system supports heterogeneous SLAM based AR devices and corresponding SDKs [20, 34, 54] as long as we attach our UWB measurement modules onto them as shown in Figure 5.

Hardware & Firmware

Our distance measurement module consists of a microcontroller unit (MCU), a UWB unit and peripheral circuits. The overall size of the board with all components assembled is $90mm \times 40mm \times 20mm$. We select a ESP32 (NodeMCU 32S) module as our MCU since it provides built-in WiFi communication function [15]. The UWB unit (DWM1000) connects with the MCU through SPI bus. We utilize a rechargeable Li-ion battery (9V, 600mAh) and a dual regulator set to power the MCU (5V) and UWB unit (3.3V) separately. As for the AR devices, we prototype our system with the ZenFone (ZS571KL) which runs a Google Tango system.

UWB units measures distances through a double-sided two-way ranging scheme operating on the MCU. This scheme corrects the time drift for the time-of-flight measurement by exchanging two round-trip messages [27]. When performing one to n ranging, we estimate the update rate is around $1000/(80+21n)$Hz with our current parameters, e.g., one-to-one ranging results in ~ 9.9Hz and one-to-two ranging results in ~ 8.1Hz. Correspondingly, in a two-user or three-user registration, users are free to move with a normal speed ($\sim$ 1m). On the other hand, the SLAM from the AR device runs at a rate of $\sim$ 30Hz. Thus we keep synchronizing the newly received distance measurements with the most updated local positions as one complete sample, which yields an update rate $\sim$ 8.1Hz.

With continuous transceiving of UWB and WiFi, the whole board peak current reaches 300mA calculated based on the datasheet. A 600mAh battery lasts for $\sim$ 2hrs which means we can perform registration ($\sim$ 10s) about 720 times. After registration, we keep DWM1000 in sleep mode (550nA) so that the battery can last substantially.

Instant Registration

Recall Eq. 2, a sequential quadratic programming (SQP) algorithm is commonly used to effectively solve constrained optimization problems [37]. A number of software packages offer implementations for SQP. As in our prototype, we offload the solver onto a remote PC (CPU 2.5GHz, i7-6500U) which runs MATLAB Optimization Toolbox ([33]). We set the boundaries of t_y as $t_{y_{min}} = -0.1$m and $t_{y_{max}} = 0.1$m with the assumption that users initialize the SLAM within a height range of $[1.4-1.6]$m above the floor. For an one-to-one registration, we observe the algorithm converges in a short time (< 0.15s) with 100 samples. As a side note, we clarify that we do not focus on transplanting the SQP implementation onto mobile platforms here.

Collaborative AR Applications

Our applications need to manage three types of wireless communications: (i) the distance measurement modules and the AR devices, (ii) the AR devices and the remote solver and (iii) among different AR devices. We adopt the user datagram protocol (UDP) to transmit the measurements from the MCU to AR devices. As for synchronizing multiple users' positions, orientations, and collaborative activities, we set up a local server and utilize WebRTC [12] for real-time communications. Meanwhile, during the registration phase, we collect the local positions and distance measurements and feed them to the remote solver through UDP as well. The AR collaboration applications have been implemented within Unity3D [53] using Google Tango API.

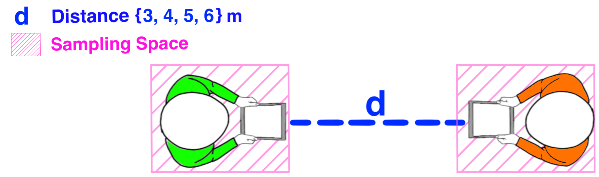

Figure 6. Technical evaluation setups.

TECHNICAL EVALUATION

To study the performance of our registration method, we set up a technical evaluation. Primarily, we considered a 2-user registration case. We studied the sampling parameters such as the sampling spaces and the distances between users. Since our approach requires users to roughly hold the device at a constant height during the sampling, we define the sampling space as an axis aligned bounding box ($l \times w$) on the horizontal plane $x-z$ plus a height level (h) along the y axis. And $r \in \{3,4,5,6\}$m denotes the distances between the sampling space centers of each user. We selected a sufficiently large 3D volume as the sampling space in order to capture the data systematically, i.e., $l \times w = 2 \times 2$m and $0.8 \leq h \leq 2.1$m. We collected 3000 samples for each r and repeated the same data capturing.

Our approach emphasizes on enabling spontaneous collaborations without sharing SLAM map. Thus we mainly compared with a registration given the shared map. For this purpose, the local positions of each AR device yielded the same coordinate system of the shared map. Then we synthetically created different frames by transforming the shared coordinate system with randomly generated $\theta_g \in [-\pi, \pi]$ and $\mathbf{T}_g = [t_x, t_y, t_y]^T$, $-10 \leq t_x, t_z \leq 10$m and $-0.2 \leq t_y \leq 0.2$m. We intentionally varied t_y in a small range to simulate the real situation where different users would not be able to initialize the SLAM at the exact same height. We sub-sampled the datasets based on different test conditions and computed the synthetic local positions with the given ground truth transformations. Then we fed the optimization solver with the synthetic local positions and the true distance measurements. In the results, the accuracy of the registration was indicated by root mean square error (RMSE) of the translational (t_x, t_y, t_z) and rotational (θ) transformation separately.

Sampling Space

We evaluated the sampling space given the furthest distance between two users, i.e., 6m. Then we varied the planar bounding box of the sampling space as $l = w = 1.4, 1.6, 1.8, 2$m and dissected the heights into three levels $h \in [0.9-1.5], [1.2-1.8], [1.5-2.1]$. With these test conditions, we repeated the sub-sampling and optimization for 10 times and took the averages. Prior to the evaluation, our preliminary tests indicate a sampling number of 100 is a good balance between sampling time and the accuracy. Further 100 different ground truth transformations were drawn for each test.

A two-way univariate ANOVA result showed the bounding box size and the height level were significant to the accuracies of T and $theta$. Then we performed a post hoc pairwise

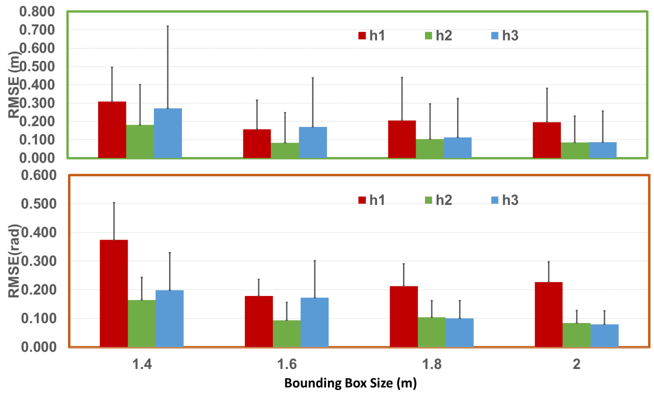

Figure 7. Results of evaluations of both translational (up) and rotational (down) accuracy on the sampling space with $l = w = 1.4, 1.6, 1.8, 2$ and three hight levels at $h_1 \in [0.9 - 1.5], h_2 \in [1.2 - 1.8]$, and $h_3 \in [1.5 - 2.1]$m.

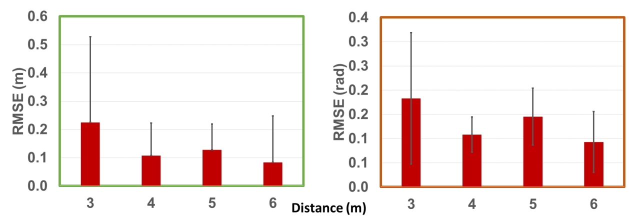

Figure 8. Results of evaluations of both translational (left) and rotational (right) accuracy on the distances ($r \in \{3,4,5,6\}$m) with $l = w = 1.6$m and $h \in [1.2 - 1.8]$m.

comparisons with Bonferroni correction to examine the conditions separately. For both translational and rotational accuracy, we observed that, for $l = w \in \{1.6, 1.8, 2\}$, there were no significant differences ($p > 0.05$), yet $l = w = 1.4$ yielded a significant difference from others ($p < 0.05$). Further, pairwise tests with h still indicated significant differences from each other. As shown in the Figure 7, we confirmed that the average translational error stayed below 0.2m, and rotational one less than 0.21 ($\sim 12^\circ$) as the bounding box size became larger than 1.6m. The optimization result was sensitive to the distribution of the samples, e.g., a larger zone makes the optimization more robust. But when the region is sufficiently large, we suspected the optimization reaches to a limit because of the UWB accuracy.

Although h appeared to be significantly affecting the accuracy, the overall accuracy still remained low as long as $l = w \geq 1.6$m. Further from an ergonomic point of view, we selected a height level within $[1.2 - 1.8]$m. Note, our test adopted a strict condition on height variations (0.6m) to guarantee the effectiveness of our practical guidance.

Distances

Based on the results from the sampling space evaluation, we selected $l = w = 1.6$m and $h \in [1.2 - 1.8]$m for studying the effect of distance $r \in \{3, 4, 5, 6\}$m on the registration accuracy. With a one-way ANOVA test, we found that r significantly affects both both translational and rotational accuracies ($p < 0.05$). Pairwise comparisons with Bonferroni correction showed that within group of $r \in \{4, 5, 6\}$, there were no significant differences. We suspected that within a close range, the measurement accuracy of UWB unit may degrade. From Figure 8, we observed that, the average errors for $\mathbf{T}$ yielded below 0.25m for all r, and θ less than 0.23 (13.2^o).

Results

The investigations from the technical evaluation indicated we support one-to-one registration at various distances. With limited resources, we conservatively suggest the following sampling parameters for the registration: (i) initialize the SLAM device at a height of ~ 1.5m from the floor, (ii) capture 100 synchronized local positions and distance measurements, (iii) during sampling, cover a space with $l = w \geq 1.6$m, (iv) hold the device at a constant height roughly ($h \in [1.2 - 1.8]$) for better accuracy. With these parameters, we observed an average translational accuracies of ~ 0.15m from Figure 7 and 8 and rotational one of ~ 0.13 (7.4°) when $r \geq 4$m.

TASK EVALUATION

To further verify the registration performance and examine the usability toward supporting spatial AR coordination activities, we conducted a task evaluation with users. We recruited 11 university students (10 male) with an average age of 25 to participate our study. The majority (9) of the participants were familiar with the concept of AR. We asked users to finish a two-session study which focused on view pointing and trace following with rendered AR cues respectively. Through these tasks, we emphasized comparing our distance based approach against the sharing map registration.

To setup a collaborative environment, one of the authors acted as *User A* and the participant played a role of *User B*. *User A* was provided with a pre-built SLAM map of the environment whereas *User B* always started the SLAM with arbitrary positions and orientations in the given environment. The visual cues were always created within the *User A*'s coordinate system at first . Then *User A* and *User B* held the device and kept moving on independent paths until enough samples were collected for the registration. With the runtime registration result, the visual cues were duplicated in User B's frame. Subsequently, with the AR cues, users were asked to finish the tasks. To remove possible learning effects, we offered a training and practice trial before the test.

We constrained the tasks to focus on evaluating the registration performance with the real users. Thus in this paper, we did not include any collaborative tasks and collect the subjective experiences. For the studies, we compared the performance against the central-map approach. Yet we did not let the user to explicitly experience the map sharing action (we set it up for users). For the *View Pointing* task, it took us about 15 minutes to scan the environment ($\sim 5 \times 7$m) and ~ 3 minutes to exchange the scanned map ($\sim$ 30MB) through a WLAN. As for the *Trace Following* task, we used a map ($\sim$ 50MB) for an environment of $\sim 10 \times 30$m. Further, we noticed the maps were sensitive to the ambient lighting condition.

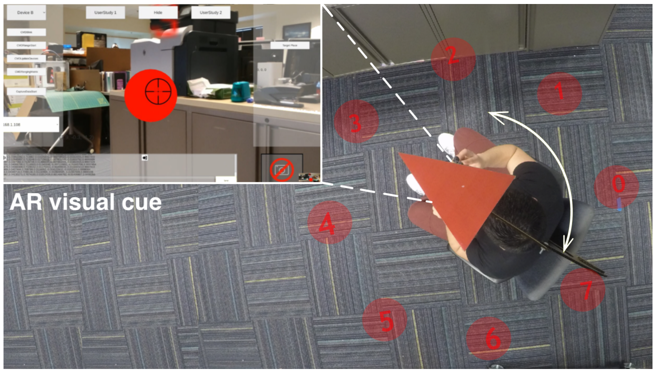

Figure 9. Setup for view pointing task evaluation. User sits on a rolling chair points to different directions with visual cues.

View Pointing

In a collaborative AR environment, it is essential to synchronize the orientations between users for spatial reference. As shown in Figure 9, we set up a top view camera in the physical environment so that the pointing results from *User A* and *User B* can be compared with a common reference. To be specific, *User A* positioned the virtual indicators while sitting in the rolling chair. After a registration, *User B* was asked to move towards the chair and sit in it. In each trial, we generated a randomized sequence containing 4 indices of the 8 evenly distributed virtual spheres. *User B* rotated the chair and pointed at a direction.

We asked the users to perform the registration followed by a trial 3 times in this task. In total, we obtained 132 images showing 11 users pointing at different directions. After processing the images with MATLAB, we recognized the triangle which is fixated on the chair and the corresponding direction in the image frame. Similarly, we captured the ground truth by averaging the pointing directions from 24 images of $UserA$ pointing with the prebuilt SLAM map. Then we averaged the trials and compared with our ground truth. The overall mean error of 3.7° with a standard deviation of 9.0° is comparable with a suggested viewfinder frustum field of view (8°) [1]. This result implies that *SynchronizAR* is applicable for orientation sensitive AR collaborations.

Trace Following

We selected a trace following task to evaluate the effects of both translational and rotational results on the AR guidance scenarios. Unlike the fixated rolling chair in task 1, users dynamically moved in a larger space ($\sim 5 \times 3$m). We generated a metric to evaluate the similarities between different paths from the recorded top-view videos. To eliminate the subjective motion from different users, we created baselines for each user. To be specific, instead of creating a ground truth from *User A* in prior, we requested users to follow the traces with the registration provided by a shared map twice. Then the ones with runtime registrations will be compared with this baseline.

As shown in Figure 10, we constructed 3 traces with different shapes (L-, S- shape, and a spline) with the same starting and ending points to represent curves with different curvatures.

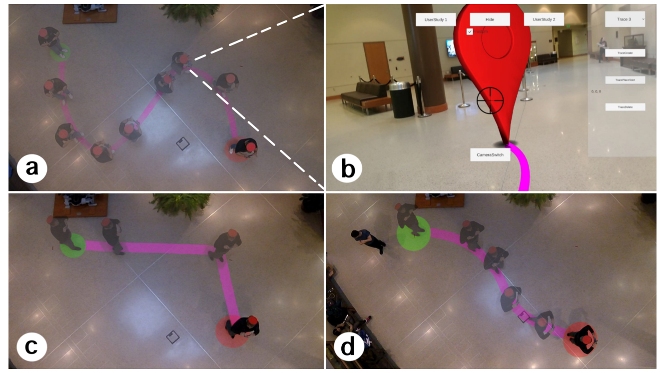

Figure 10. Illustration of path following task evaluation. Users follow 3 different virtual traces (a, c, d) in the AR scene (b).

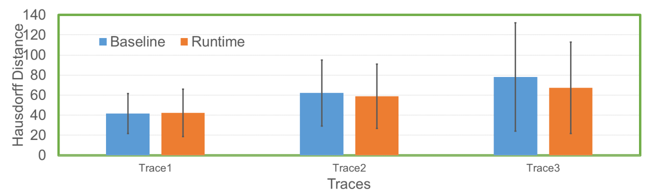

Figure 11. Results from trace following task.

Each user was asked to follow all three traces 4 times in total, i.e., twice with ground truth and twice with runtime registrations. The camera captured the trace following movements where users wore a hat which was covered by a red dot. After processing the video, we obtained the paths of users in the image frames. A modified Hausdorff distance (pixels) increases monotonically as the amount of differences between two sets of points increases [14]. It is often used to compare the similarities of two curves. Thus we employed the Hausdorff distance as it is sensitive to both translational and rotational errors between the curves. For each user, we denoted the two sets of paths with ground truth as G_1 and G_2, and the ones with runtime registration as H_1 and H_2. Further, for each user, we calculated the Hausdorff distances between paths in G_1 and G_2 ($D_{G_1G_2}$) with respect to different traces. We composed $D_{G_1H_1}$, $D_{G_2H_1}$, $D_{G_1H_2}$, and $D_{G_2H_2}$ together and performed a T-test against $D_{G_1G_2}$ from all of the users.

For all three traces, we observed no significant difference between the baselines and the runtime registration results ($p = 0.92, 0.77, 0.55$ respectively). The mean errors and standard deviations are plotted in Figure 11. Through this task evaluation, we confirmed that our registration accuracy supports creating visual guidance in AR collaborations.

EXAMPLE USE CASES

By applying the registration result, *SynchronizAR* enables every AR device to be spatially registered with each other instantly and conveniently. Taking advantages of the spatial awareness across the users in an AR environment, we showcased four use cases with *SynchronizAR*.

Figure 12. ***SynchronizAR*** **supports spontaneous collaboration, i.e., a new user (b) join an existing AR collaboration (a) instantly (c).**

Spontaneous Collaboration

Here we built a multiple-player ball catching game with support from *SynchronizAR*. We leveraged the spatial interactions such as pointing enabled by the registrations in AR collaborative games. Further we demonstrated our instant registration technique which enables a player to join any time during the game. At first two players started a game (Figure 12 a). Then a third player was able to join the game after a quick registration process with one of the original players (Figure 12 b). After that, the coordinate system of the new player was shared between the original collaboration environment and the game continued with three players (Figure 12 c).

Interactive AR Game Construction

With *SynchronizAR*, we created an interactive AR game construction and playing experience to multiple users. Here we allow users to construct AR games in the physical world as a game map and instantly share it with other users once registered. For example in this coin-collection game, a builder (Figure 13 a, b) first placed golden coins and rusted coins in the café and turned it into a game scene. Then a catcher (Figure 13 c, d) registered with the builder and synchronized with the game world. With proximity based spatial movement, the catcher collected coins in the AR scene. We also support asynchronized collaboration as we need no infrastructure prior. After registering once, any user can revisit the scene and view collaborator's activities which happened while he/she was gone.

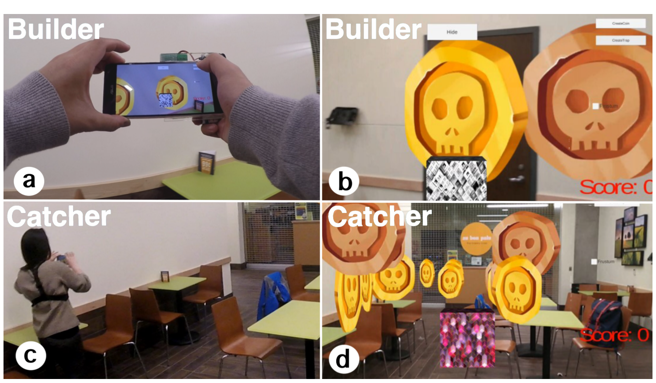

Figure 13. Interactive AR game creation. Two users act as a game world builder (a, b) and a player (c, d).

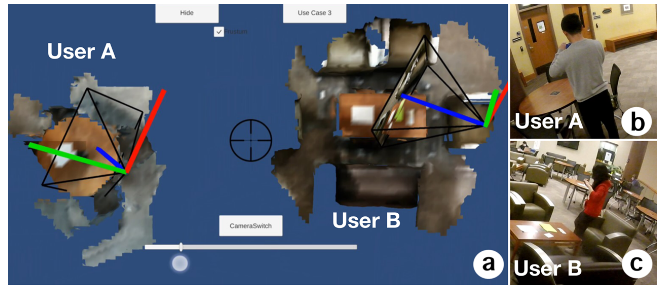

Figure 14. A spatially coherent virtual model (a) is created after user A and B scan their own surrounding environment (c, d). Two distant users can refer to each other's view with spatial references (a, b, e).

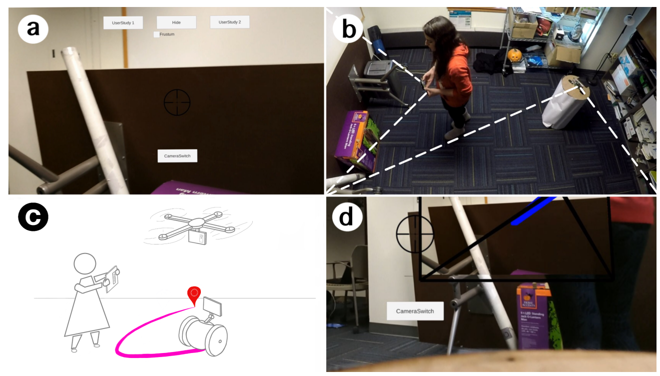

Figure 15. ***SynchronizAR*** **being used for human-robot interactions(c). The robot mimics the user's movement (b). And they can access each other's views (a, d).**

Spatial Aware Screen Sharing

In a co-located collaborative context, two users stays distant from each other may also want communicate through view sharing instantly. Different from a traditional video conferencing, *SynchronizAR* offered spatial awareness to the shared view. Also during the collaboration, we allow users to freely refer to each other's surrounding environment. Here, the users scanned the environment around each of them separately (Figure 14 c, d). Then the scanned geometry models can be registered using the spatial transformation from *SynchronzAR*. As the user walked around, the distant collaborator can access the first-person view through the frustum, also create an independent virtual navigation with the registered 3D model.

Human Robot Interactions

In the future, we envision that human beings and autonomous robots interact with each other naturally [48]. In this context, the spatial awareness will be critical. By attaching an AR device to an autonomous robot and registering it with a user, we coordinate the robot with respect to the user's position and orientation. Thus, the user can interact with the robot naturally through his/her spatial movement. For example, in this use case, we enable the robot to mimic the user's movement in the same direction and adjust the facing direction accordingly.

DISCUSSION AND LIMITATION

Sampling Parameters. With limited resources, we were not able to fully investigate the sampling parameters. In our current setup, we primarily rely on a shared SLAM map as ground truth for testing. Despite the stable performance on Google Tango devices, we observed drift from time to time in a feature less environment. In the future, we also plan to introduce an external tracking system e.g., a VICON like system to study the effects of possible drifts from the SLAM itself. Additionally, our distance based registration required users to move on independent paths. Although during the user study, we haven't observed identical walking patterns, it will be helpful to give AR walking cues to users during registration.

Temporal Synchronization. We run a 1-to-n pooling where the n distances were packaged on an arbitrary MCU and sent to the AR devices via UDP. The newly received distances package, together with last updated coordinates which were smoothed by a running average, were sent to the solver. Although we did not explicitly model the temporal differences between the measurements and the coordinates, the running average practically reduced the potential correspondence error. We acknowledge that the accuracy may improve with a dedicated synchronization scheme. Still we found the average positional RMSE (~ 0.25m) remains at the same level of the UWB accuracy (~ 0.1m).

Scalability. We believe the modern mobile device can solve our optimization problem given the fact it runs SLAM in real-time which usually involves heavy optimization. In a non-central deployment, the distance measurements and the local coordinates can be first synchronized and packaged on the local AR devices. Then the packed messages will be shared through a peer-to-peer communication. Finally, the optimization runs in the AR device instead of a remote server.

Potential Applications. Although the cloud based solution is capable of supporting the collaborative AR given a reliable map, our method is more suitable for cases where a reliable map is not available or hard to access (a dynamic environment), or not necessary (e.g., casual social AR activities). Also, for large spaces (e.g., urban planning), the users can start the collaboration at different locations instantly without scanning the map as shown in the *Spatial Aware Screen Sharing* case. Further our method can be used to augment other approaches. For examples, enhancing LBS with accurate registrations (e.g., Pokemon battles), and with cloudAR, enabling asynchronous and persistent experience.

Form Factors. For the AR devices, we selected Google Tango phones to prototype our AR applications. However, our registration is applicable to heterogeneous devices (HMD and handheld) running various SLAM algorithms since our indirect approach does not require sharing SLAM map. Further, our registration can be utilized for establishing a collocated collaboration for virtual reality (VR) devices which rely on SLAM tracking. On the distance measurement side, we would like to work on minimizing the package of the module. Besides, it will interesting to generalize distance based registration approach with matured RF technologies (Bluetooth and WiFi) with different types of distance estimation (time-of-flight, time-difference-of-arrival, and angle-of-arrival) [43].

Accuracy. Although we observed a good translational and rotational accuracy within a large area, we found the UWB measurements can be distorted under heavy non-line-of-sight (NLOS) conditions such as solid walls. In the future, we need to identify the NLOS measurements and compensate or remove them. Besides, the SLAM algorithm itself may drift in a featureless environment causing inaccurate registration or shifting the AR rendering after the registration. Also we observed the standard deviation of the error remains high as shown in Figure 7 and 8. We suspect this is caused by the SLAM drift primarily. Future, we plan to determine the error resources by comparing with a VICON system.

Number of Users. Our current supports for more than 3 users rely on pairwise peer-to-peer registration. To further support more users being registered simultaneously, we need to overcome two issues: (i) sampling rate of distance measuring, and (ii) introducing distance constraints into the optimization. We plan to resolve the sampling rate limitation by introducing time-different-of-arrival. As for the highly nonlinear constrained optimization, we still need to investigate and select a method which is applicable for mobile devices [56].

CONCLUSION

In this work, we proposed *SyncrhonizAR*, enabling a co-located collaborative AR experience by spatially registering multiple users in a spontaneous manner. Through our technical evaluation, we conservatively suggested guidelines for using *SynchronizAR*. We observed an average translational accuracy of 0.15m and rotational accuracy of 7.4° when two users are at a distance $r > 4$m. Within the user study, we validated that with our registration, users can successfully perform AR spatial interactions accurately including view pointing and trace following. Therefore, we believe our work is applicable to a wide range of use cases leveraging the spatial registration of multiple SLAM devices.

ACKNOWLEDGEMENT

We thank the reviewers for their valuable feedback and comments. This work was supported partially by the NSF Awards No. 1632154 (IIP). We also acknowledge the Feddersen chair funding for early work. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

1. Ferran Argelaguet and Carlos Andujar. 2009. Visual feedback techniques for virtual pointing on stereoscopic displays. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. ACM, 163–170.

2. Clemens Arth, Christian Pirchheim, Jonathan Ventura, Dieter Schmalstieg, and Vincent Lepetit. 2015. Instant outdoor localization and slam initialization from 2.5 d maps. *IEEE transactions on visualization and computer graphics* 21, 11 (2015), 1309–1318.

3. ARToolKit. 2017. ARToolKit. (2017). Retrieved Dec 1, 2017 from = https://www.hitl.washington.edu/artoolkit/.

4. Shuanghua Bai and Houduo Qi. 2016. Tackling the flip ambiguity in wireless sensor network localization and beyond. *Digital Signal Processing* 55 (2016), 85–97.

5. Yohan Baillot, L Davis, and J Rolland. 2001. A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augumented reality* (2001), 67.

6. Steve Benford, Chris Greenhalgh, Gail Reynard, Chris Brown, and Boriana Koleva. 1998. Understanding and constructing shared spaces with mixed-reality boundaries. *ACM Transactions on computer-human interaction (TOCHI)* 5, 3 (1998), 185–223.

7. Hrvoje Benko, Edward W Ishak, and Steven Feiner. 2004. Collaborative mixed reality visualization of an archaeological excavation. In *Proceedings of the 3rd IEEE/ACM international Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 132–140.

8. Mark Billinghurst, Ivan Poupyrev, Hirokazu Kato, and Richard May. 2000. Mixing realities in shared space: An augmented reality interface for collaborative computing. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, Vol. 3. IEEE, 1641–1644.

9. Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. 2014. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*. Springer, 536–551.

10. Robert Castle, Georg Klein, and David W Murray. 2008. Video-rate localization in multiple maps for wearable augmented reality. In *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*. IEEE, 15–22.

11. Hung-Lin Chi, Shih-Chung Kang, and Xiangyu Wang. 2013. Research trends and opportunities of augmented reality applications in architecture, engineering, and construction. *Automation in construction* 33 (2013), 116–122.

12. codelabs. 2017. WebRTC. (2017). Retrieved Dec 1, 2017 from = https://webrtc.org/.

13. decaWave. 2017. Production Documentation. (2017). Retrieved Dec 1, 2017 from = https://www.decawave.com/product-documentation .

14. M-P Dubuisson and Anil K Jain. 1994. A modified Hausdorff distance for object matching. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, Vol. 1. IEEE, 566–568.

15. Esspressif. 2017. ESP32. (2017). Retrieved Dec 1, 2017 from = https://www.espressif.com/en/products/hardware/esp32/overview .

16. Philipp Fleck, Clemens Arth, Christian Pirchheim, and Dieter Schmalstieg. 2015. [POSTER] Tracking and Mapping with a Swarm of Heterogeneous Clients. In *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*. IEEE, 136–139.

17. Susan R Fussell, Robert E Kraut, and Jane Siegel. 2000. Coordination of communication: Effects of shared visual context on collaborative work. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 21–30.

18. Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014. World-stabilized annotations and virtual scene navigation for remote collaboration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 449–459.

19. Hans Gellersen, Carl Fischer, Dominique Guinard, Roswitha Gostner, Gerd Kortuem, Christian Kray, Enrico Rukzio, and Sara Streng. 2009. Supporting device discovery and spontaneous interaction with spatial references. *Personal and Ubiquitous Computing* 13, 4 (2009), 255–264.

20. Google. 2017. ARCore. (2017). Retrieved September 1, 2017 from = https://developers.google.com/ar/.

21. Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. 2017. Sharevr: Enabling co-located experiences for virtual reality between hmd and non-hmd users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4021–4033.

22. Mike Hazas, Christian Kray, Hans Gellersen, Henoc Agbota, Gerd Kortuem, and Albert Krohn. 2005. A relative positioning system for co-located mobile devices. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 177–190.

23. Tobias Höllerer, Steven Feiner, Tachio Terauchi, Gus Rashid, and Drexel Hallaway. 1999. Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics* 23, 6 (1999), 779–785.

24. Anselm Hook, Pierre Fite-Georgel, Matt Meisnieks, Anthony Maes, Marc Gardeya, and Leonid Naimark. 2014. Generation and Sharing Coordinate System Between Users on Mobile. (Sept. 18 2014). US Patent App. 13/835,822.

25. Jiawei Huang, Tsuyoshi Mori, Kazuki Takashima, Shuichiro Hashi, and Yoshifumi Kitamura. 2015. IM6D: magnetic tracking system with 6-DOF passive markers for dexterous 3D interaction and motion. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 217.

26. Ke Huo, Yuanzhi Cao, Sangho Yoon, Zhuangying Xu, , Guiming Chen, and Karthik Ramani. 2018. Scenariot: Spatially Mapping Smart Things Within Augmented Reality Scenes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, –.

27. Yi Jiang and Victor CM Leung. 2007. An asymmetric double sided two-way ranging for crystal offset. In *Signals, Systems and Electronics, 2007. ISSSE'07. International Symposium on*. IEEE, 525–528.

28. Shunichi Kasahara, Valentin Heun, Austin S Lee, and Hiroshi Ishii. 2012. Second surface: multi-user spatial collaboration system based on augmented reality. In *SIGGRAPH Asia 2012 Emerging Technologies*. ACM, 20.

29. Nikolaos Kyriazis and Antonis Argyros. 2014. Scalable 3d tracking of multiple interacting objects. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 3430–3437.

30. David Lakatos, Matthew Blackshaw, Alex Olwal, Zachary Barryte, Ken Perlin, and Hiroshi Ishii. 2014. T (ether): spatially-aware handhelds, gestures and proprioception for multi-user 3D modeling and animation. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*. ACM, 90–93.

31. David Ledo, Saul Greenberg, Nicolai Marquardt, and Sebastian Boring. 2015. Proxemic-aware controls: Designing remote controls for ubiquitous computing ecologies. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 187–198.

32. Guoqiang Mao, Barış Fidan, and Brian DO Anderson. 2007. Wireless sensor network localization techniques. *Computer networks* 51, 10 (2007), 2529–2553.

33. MathWorks. 2017. MATLAB Optimization Toolbox. (2017). Retrieved Dec 1, 2017 from = https://www.mathworks.com/products/optimization.html .

34. Microsoft. 2017. Hololens. (2017). Retrieved September 1, 2017 from = https://www.microsoft.com/en-us/hololens.

35. Gajamohan Mohanarajah, Vladyslav Usenko, Mayank Singh, Raffaello D'Andrea, and Markus Waibel. 2015. Cloud-based collaborative 3D mapping in real-time with low-cost robots. *IEEE Transactions on Automation Science and Engineering* 12, 2 (2015), 423–431.

36. Jens Müller, Roman Rädle, and Harald Reiterer. 2016. Virtual Objects as Spatial Cues in Collaborative Mixed Reality Environments: How They Shape Communication Behavior and User Task Load. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1245–1249.

37. Jorge Nocedal and Stephen J Wright. 2006. *Sequential quadratic programming*. Springer.

38. Ohan Oda, Carmine Elvezio, Mengu Sukan, Steven Feiner, and Barbara Tversky. 2015. Virtual replicas for remote assistance in virtual and augmented reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 405–415.

39. Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, and others. 2016. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 741–754.

40. Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. 2016. Room2room: Enabling life-size telepresence in a projected augmented reality environment. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*. ACM, 1716–1725.

41. Sudeep Pillai and John Leonard. 2015. Monocular slam supported object recognition. *arXiv preprint arXiv:1506.01732* (2015).

42. Polhemus. 2017. Polhemus. (2017). Retrieved Dec 1, 2017 from = https://polhemus.com/applications/electromagnetics/.

43. quuppa. 2017. QUUPPA technology. (2017). Retrieved Dec 1, 2017 from = http://quuppa.com/.

44. Gerhard Reitmayr and Dieter Schmalstieg. 2004. *Collaborative augmented reality for outdoor navigation and information browsing*. na.

45. Luis Riazuelo, Javier Civera, and JMM Montiel. 2014. C2tam: A cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems* 62, 4 (2014), 401–413.

46. Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L Miguel Encarnaçao, Michael Gervautz, and Werner Purgathofer. 2002. The studierstube augmented reality project. *Presence: Teleoperators & Virtual Environments* 11, 1 (2002), 33–54.

47. Kjeld Schmidt and Carla Simonee. 1996. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work (CSCW)* 5, 2-3 (1996), 155–200.

48. Stela H Seo, James E Young, and Pourang Irani. 2017. Where are the robots? In-feed embedded techniques for visualizing robot team member locations. In *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on*. IEEE, 522–527.

49. Sixense. 2017. Sixense. (2017). Retrieved Dec 1, 2017 from = https://www.sixense.com/platform/hardware/.

50. Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. 2016. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision*. Springer, 294–310.

51. Alexander Toshev and Christian Szegedy. 2014. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1653–1660.

52. ubiquity6. 2018. ubiquity6. (2018). Retrieved July 1, 2018 from = http://ubiquity6.com/.

53. Unity3D. 2017. Unity3D. (2017). Retrieved September 1, 2017 from = https://unity3d.com/.

54. wikitude. 2017. wikitude. (2017). Retrieved September 1, 2017 from = https://www.wikitude.com/.

55. Chi-Jui Wu, Steven Houben, and Nicolai Marquardt. 2017. Eaglesense: Tracking people and devices in interactive spaces using real-time top-view depth-sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3929–3942.

56. Ozgur Yeniay. 2005. A comparative study on optimization methods for the constrained nonlinear programming problems. *Mathematical Problems in Engineering* 2005, 2 (2005), 165–173.