Augment Hum Res (2017)2:6
DOI 10.1007/s41133-017-0009-z

ORIGINAL PAPER

Collocated Multi-user Gestural Interactions with Unmodified Wearable Devices

Augmenting Multi-user and Multi-device Interactions with Proximity and Natural Gestures

Mihai Bâce[1] · **Sander Staal**[2] · **Gábor Sörös**[3] · **Giorgio Corbellini**[2]

Received: 31 May 2017 / Accepted: 14 July 2017

Abstract Many real-life scenarios can benefit from both physical proximity and natural gesture interaction. In this paper, we explore shared collocated interactions on unmodified wearable devices. We introduce an interaction technique which enables a small group of people to interact using natural gestures. The proximity of users and devices is detected through acoustic ranging using inaudible signals, while in-air hand gestures are recognized from three-axis accelerometers. The underlying wireless communication between the devices is handled over Bluetooth for scalability and extensibility. We present (1) an overview of the interaction technique and (2) an extensive evaluation using unmodified, off-the-shelf, mobile, and wearable devices which show the feasibility of the method. Finally, we demonstrate the resulting design space with three examples of multi-user application scenarios.

Keywords Gesture recognition · Collocated interactions · Group experience · Multi-user interaction · Smart objects

Introduction

Wearable devices such as smartwatches are becoming wide-spread personal companions for multiple activities ranging from activity tracking, communication, gaming, storytelling, playing sports, to controlling appliances, and many others. Social interaction and group activities are an important part of our lives, and technology allows us to interact with people from all over the world. Many of the previously mentioned activities also have a strong social component. Nevertheless, the social interaction experience is mostly *virtual*. A virtual interaction happens through the screen and actuators of our devices and does not take into account the physical proximity between the users. For example, even simple actions like sharing a photograph among two friends do not change when the friends are close to each other or spread over the globe.

We explore multi-user interactions between a group of people who are physically close to one another. Our work is inspired by the theory of proxemics established by Edward Hall. It studies the way people mediate their interactions with other people around them [12]. While this theory covers different dimensions, one which certainly influences and leads to higher interaction engagement is interpersonal distance. Imagine a scenario where three children are in a theme park and they want to interact with a treasure chest as illustrated in Fig. 1. For the chest to open, the three children have to be in front of the chest and

Electronic supplementary material The online version of this article (doi:10.1007/s41133-017-0009-z) contains supplementary material, which is available to authorized users.

✉ Mihai Bâce
mihai.bace@inf.ethz.ch

Sander Staal
sander.staal@inf.ethz.ch

Gábor Sörös
gabor.soros@inf.ethz.ch

Giorgio Corbellini
gcorbellini@gmail.com

[1] Department of Computer Science, ETH Zurich, CNB H 108, Universitätstrasse 6, 8092 Zurich, Switzerland

[2] Department of Computer Science, ETH Zurich, Universitätstrasse 6, 8092 Zurich, Switzerland

[3] Department of Computer Science, ETH Zurich, CNB H 103.2, Universitätstrasse 6, 8092 Zurich, Switzerland

Published online: 08 August 2017

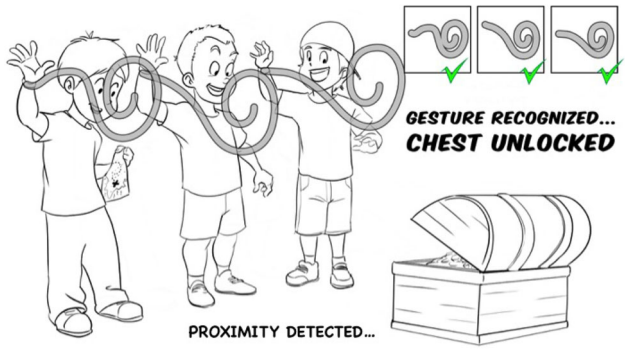

Fig. 1 Concept art: Three children wearing smartwatches unlock a treasure chest by performing a gesture together. The application is a combination of collective gesture recognition, physical proximity detection, and data transmission

all of them have to perform the same interaction (e.g. a hand gesture), roughly at the same time. The key to unlock the chest is a combination of physical proximity, collaboration between the users, and interaction through hand gestures. The treasure chest in our example is just an abstraction for any smart/digital object. The concepts we present here can be extended to public displays, toys, and many other applications.

Our proposed interaction technique builds on existing methods for gesture recognition and proximity detection to enable collocated multi-user gestural interactions. We leverage only standard features of off-the-shelf mobile and wearable devices to recognize gestures and to detect physical proximity: built-in motion sensors and inaudible acoustic signals are used for an unobtrusive and seamless interaction. The underlying communication is handled over Bluetooth for scalability and extensibility.

To our knowledge, we present the first collocated multi-user gestural interaction technique which uses acoustic ranging and runs on mobile devices. We assess the building block of this technique in different environments on unmodified, off-the-shelf hardware and give practical recommendations which lead to robust results. We explore the design space of this interaction technique and demonstrate it with three multi-user application scenarios.

Related Work

Before coming to the description of the method, we review related work that concerns the different aspects of this interaction technique. First, we discuss systems that enable multiple users to collaborate. Then, we look at systems that use physical proximity in ubicomp applications. Further on, we review techniques for recognizing gestures and finally discuss methods to detect proximity.

Multi-user and Multi-device Interactions

The proliferation of affordable smart devices resulted in consumers owning and carrying multiple connected devices with them. Multi-user collaboration scenarios require devices to communicate with each other. Embedded systems or smartphones can be paired through simultaneous shaking patterns [15, 23], simultaneous pressing of a button [38], bumping gestures [13], touch gestures spanning multiple displays [14], or pinching gestures [27]. SurfaceLink [10] is a system that associates devices given a specific gesture. This is an audio-based grouping, but it is limited to devices that must share the same surface. An overview of different ways to connect devices is given by Jokela et al. [18]. These systems are mostly focusing on the initial pairing of devices, while, in our work, we want to support continuous interaction between the devices and its users.

The problem of design and development of such cross-device systems is still subject to research [5, 16]. HuddleLamp [37] is a desk lamp with an integrated depth camera which enables multi-user and multi-device interaction around a tabletop. Pass-them-around [26] is a collaborative photograph sharing application. A system which aims to enable multi-user gestural interaction is PathSync [4]. To interact with digital objects, users must replicate a screen-presented pattern (e.g. a moving target around the edges of a rectangle) with their hand. This approach is similar to following a moving target with your eyes [7]. However, it suffers from the same drawback that objects have to be augmented with moving patterns which is not always feasible.

Tracko [17] is a system which proposes a similar interaction technique. Devices are located in 3D space with audio signals, and cross-device interactions are supported with touch gestures on the devices' display. Our proposed interaction technique supports in-air hand gestures and is not limited to the device's display.

HandshakAR [3] was our initial effort towards collocated gestural interactions. Two users can effortlessly exchange information when they perform the same greeting gesture and are close to each other. In this paper, we extend our initial proposal to multiple users, we present a more detailed evaluation, and we explore the design space with three application scenarios.

Collocated Interactions

Proxemic interactions have a great potential in a world of natural user interfaces [11]. Systems can learn to take advantage of people and devices as they move towards one another. Marquardt et al. [30] introduced a system that explored cross-device interaction using two different

constructs, micro-mobility and F-formations. While these methods require equipment to be installed within a room (a combination of motion sensors, radio modules, and overhead depth cameras), the system enabled users to collaboratively discuss digital artefacts without focusing on the underlying technology. Gradual engagement is the concept that connectivity and information exchange capabilities are exposed as a function of inter-device proximity [28]. This also requires an instrumented environment; proximity is detected by an infrared-based motion capture system. The Proximity Toolkit [29] supplies fine-grained proxemic information between people and digital devices. The toolkit gathers data from various hardware sensors and transforms it into rich high-level proxemic information but is not targeting off-the-shelf devices.

Gestural Interaction with Wearables

Gesture recognition from motion sensors has been comprehensively investigated in the literature. The recognition methods usually rely on data from inertial measurement units because such sensors are ubiquitous and available on all devices. Different classification approaches like hidden Markov models [35], artificial neural networks [24], support vector machines [44], or dynamic time warping (DTW) [25] can be used to discriminate between different gestures.

Our interaction technique leverages wearable devices that can capture gestural interactions. Shen et al. [39] have shown recently that it is possible to track the 3D posture of the entire arm, both wrist and elbow, by only using the inertial sensors on smartwatches. MoLe [42] is a system that analyses motion data from typing movements using smartwatches. A similar system has been proposed by Arduser et al. [1], where text written on a whiteboard is inferred from simple acceleration data. Fine finger gestures like pinching, tapping, or rubbing can also be recognized from the built-in motion sensors, as shown by [43]. Zhang et al. further extend the gesture interaction space around commodity smartwatches by enabling tap and swipe gestures around the bezel or the band of a watch [45].

Recognizing gestures is not tied to motion sensors only. Fu et al. [9] have proposed a system where nearby movements can be recognized from sound. BodyScan [8] relies on radio waves to sense human activities and vital signs. GestureWatch [21] and HoverFlow [22] are both systems that recognize in-air hand gestures performed over mobile or wearable devices relying on infrared proximity sensors. Hand gestures can also be recognized from a single RGB camera found on most mobile devices [40]. WatchMe [41] is a camera-based system that can track a pen or a laser pointer on a drawing canvas and use this as an input modality. Electromyography (EMG) with force-sensitive sensors is an alternative to detect hand gestures. EMPress [31] shows how the best arm position of EMG-based systems is the forearm, meaning that such system cannot be efficiently used with off-the-shelf smartwatches.

The above list is only a selection of existing systems that hint at how non-touch screen gestures in general [2] can extend the input space of wearables.

Proximity Detection

Many technologies for indoor device localization are available. Most of these estimate a distance or distances from known landmark(s) with the help of audio or radio waves. Unfortunately, measuring the distance with the radio received signal strength indicator (RSSI) is not accurate in short ranges. This is because the signal can vary greatly due to environmental factors. Another option is the usage of motion sensors (i.e. pedestrian dead reckoning [19]), which has the drawback that the initial orientation of the device has to be known and that the accuracy depends on the sensor precision. Given our requirements of wide availability, low cost, and high precision at short distances, we focus on ranging methods that use acoustic signals on commodity devices.

Tracko [17] is an acoustic tracking system which leverages both Bluetooth Low Energy (BLE) and acoustic signals to locate devices in 3D. The system is accurate for distance up to 1.5 m, but it requires devices to support BLE peripheral mode, a feature many devices do not support at the moment. Another approach is Microsoft's BeepBeep system [33, 34] which uses two-way acoustic ranging for calculating the positions of several devices relative to each other. This ranging method has been explored and optimized in different ways [6, 32, 36], and also lays the basis for our interaction technique.

Collocated Multi-user Gestural Interactions

Overview

Our interaction technique is enabled by three main components as shown in Fig. 2. The generic *communication*

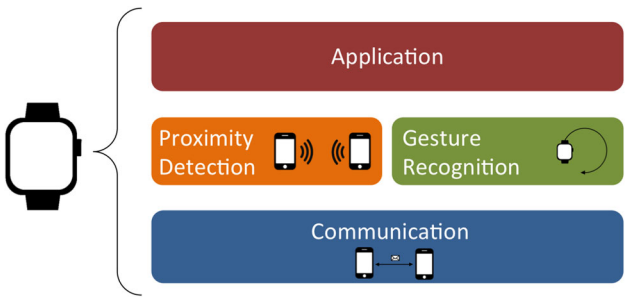

Fig. 2 Overview of the components which enable collocated multi-user gestural interactions

layer interconnects and enables interoperability between many different devices using Bluetooth as the communication protocol. The *proximity* layer estimates the pairwise distances between people or devices via non-intrusive acoustic ranging. The *gesture recognition* layer detects hand movements and identifies in-air gestures using the motion sensors of smartphones or wearables. On top of the stack, the *application* layer takes advantage of all the other layers to enable interaction among people who are in physical proximity to one another.

Communication via Bluetooth

Enabling multi-user interaction through wearables requires data exchange between the devices. Most modern wearables (smartwatches or smartglasses) rely on a companion smartphone and are used mainly to display calls, show calendar entries and other types of notifications. The current trend of manufacturers, however, is to switch towards a watch-centric approach in which smartwatches become fully independent computing platforms. There are many wireless protocols and technologies available for connecting multiple smart devices (e.g. NFC, Bluetooth, Wi-fi).

Considering the power consumption, the targeted distance ranges, and the support for interoperability between a wide variety of different devices, Bluetooth is the best fit. It is a client–server architecture and the network always forms a star topology with the server in the middle. Having a single device acting as a server can be a bottleneck, but performance issues and failure handling were not the main concerns of this work. In the future, multiple piconets could be connected together, forming mesh networks and thus solving this issue.

Proximity Detection via Acoustic Ranging

The proximity layer is based on two-way acoustic ranging, a cooperative method to estimate the distance between two smart devices. The main advantage of two-way ranging is that the devices do not need to be synchronized [34], which greatly simplifies the protocols. The participating devices sequentially emit an acoustic signal, and they simultaneously record their own signal and the signal(s) produced by the remote device(s). The distance estimate is based on measuring the time elapsed between the two received audio signals.

Distance Estimation

Figure 3 illustrates the ranging procedure for two devices. (1) Device A emits a signal at time t_{A} and records it with its own microphone at time $t_{\mathrm{A,A}}$. (2) Device B records this signal at time $t_{\mathrm{A,B}}$. (3) Device B emits another signal at time t_{B} and it records its own sound at time $t_{\mathrm{B,B}}$. (4) Device A records B's signal at time $t_{\mathrm{B,A}}$. If we denote $d_{\mathrm{X,Y}}$ as the distance between the speaker of device X and the microphone of device Y, the following distance can be determined, where $d_{\mathrm{A,A}}$ and $d_{\mathrm{B,B}}$ are device-dependent constants.

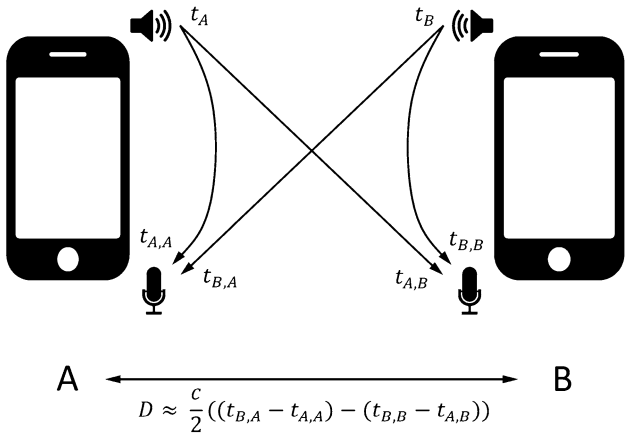

Fig. 3 Acoustic distance estimation between two devices. Both devices are simultaneously recording, where device A first emits a signal and then B emits another signal. The approximate distance can be derived by calculating the time difference of arrival

$$D = \frac{c}{2} * \left((t_{\mathrm{B,A}} - t_{\mathrm{A,A}}) - (t_{\mathrm{B,B}} - t_{\mathrm{A,B}})\right) + \frac{1}{2} * (d_{\mathrm{A,A}} + d_{\mathrm{B,B}}) \quad (1)$$

The round-trip time of flight between the two devices only depends on the two time intervals between the reception of the signals, i.e. $(t_{\mathrm{B,A}} - t_{\mathrm{A,A}})$ on device A and $(t_{\mathrm{B,B}} - t_{\mathrm{A,B}})$ on device B.

This method requires the devices to have a microphone, a speaker, and an underlying communication framework for exchanging messages and coordinating the transmission of the audio signals (in our case using Bluetooth). The microphones must record sounds through the entire process, and speakers have to emit signals sequentially (first device A, then device B).

Reference Audio Signal

The reference audio signal must have a strong autocorrelation property. This ensures robustness against ambient noise. Microsoft's BeepBeep uses audible linear chirps for ranging [34]. Curtis et al. improve the accuracy of BeepBeep by replacing the linear chirp with a maximum length sequence [6]. Jin et al. use inaudible linear chirps, where the signal includes a payload of additional information using bi-orthogonal chirps [17].

We chose a linear chirp signal with a Gaussian envelope due to its strong autocorrelation [6]. We have used a

48 kHz sampling rate for both the microphone and the speaker, which is the current maximum on Android devices [17]. With such a sampling rate, we can reconstruct signals which contain frequencies of up to 24 kHz (Nyquist frequency). The aliasing of higher frequencies can be avoided with a low-pass filter.

Two important aspects that affect the method's robustness are the signal length and the frequency bandwidth of the chirp. The described ranging method works with both audible and inaudible signals. Audible signals work better in practice, but they can be disturbing to people. Alternatively, we chose a frequency between 20 kHz and 24 kHz, since most humans can only hear up to 20 kHz. Empirically, we have selected a signal length of 75 ms.

Signal Detection

We detect the reference chirp in the recorded signal via cross-correlation. The computational complexity can be reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n * \log(n))$ using the fast Fourier transform (FFT). The Fourier transforms in the prototype applications were calculated using the JTransforms Java library.

The distance estimation method requires peak detection in the cross-correlation function with sub-sample accuracy. Considering the practical sampling rate of 48 kHz and the speed of sound, missing the cross-correlation sample number by one can already result in an error of 0.7-cm distance estimate. We have adopted the method proposed by Peng et al. to find the correlation peaks [34].

Gesture Recognition via Motion Sensors

There are many ways to recognize hand gestures using unmodified mobile and wearable devices. We adopted a method based on DTW and measurements from a single three-axis accelerometer. The gesture recognition component consists of two parts: the quantization of the acceleration measurements and DTW.

Quantization reduces the size of the time series and reduces the computational complexity of the DTW algorithm. This is desirable when using wearable devices because of their limited processing power. Furthermore, quantization improves the recognition accuracy by removing deviations that are not essential for the gesture, such as accelerometer noise [25]. However, some parameter settings may impair the recognition accuracy because they eliminate key acceleration features intrinsic for a particular gesture. Thus, choosing the correct parameters is important in this step.

Our method employs a sliding window mechanism. The acceleration data of the time series is dynamically compressed by an averaging window of size w and step size v. In the prototype applications, the parameters have been set to $w = 250$ and $v = 200$ ms.

Gesture recognition using DTW works by building a predefined template dictionary, where at least one template per gesture has to be stored. Gestures are recognized by comparing a newly collected time series with all the samples from the existing template dictionary and then the best matching pair is selected (Fig. 4). It is not just a binary decision, DTW returns a score for each gesture, so that candidates can be ranked.

Evaluation

This section presents a thorough evaluation of the individual components which enable the proposed interaction technique. Our goal is to support unmodified wearable and mobile devices. In our tests, we use two smartphones (LG G3 and LG Nexus 5×) and two smartwatches (Sony Smartwatch 3 and Motorola 360 Sport 2nd Generation). Our evaluation discards the communication layer because Bluetooth is an already established wireless standard and focuses on the remaining two components: gesture recognition and acoustic ranging.

Gesture Recognition

The gesture recognition component is based on DTW. To evaluate the recognition accuracy, we tested this method on two different gesture sets. One gesture set is based on geometric shapes (Fig. 5, introduced by Kela et al. [20]),

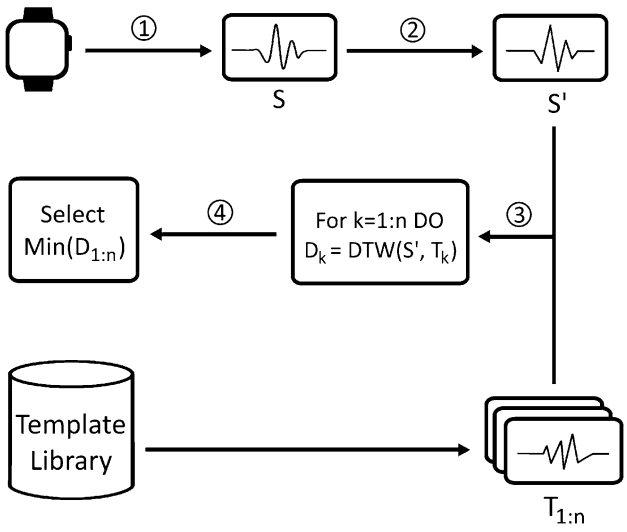

Fig. 4 *1* The accelerometer records a new sample *S*. *2* The sample gets quantized into *S'*. *3* *S'* is matched against every stored template *T* of the template library. DTW is used to calculate the matching cost. *4* The gesture is recognized by selecting the pair with the lowest cost

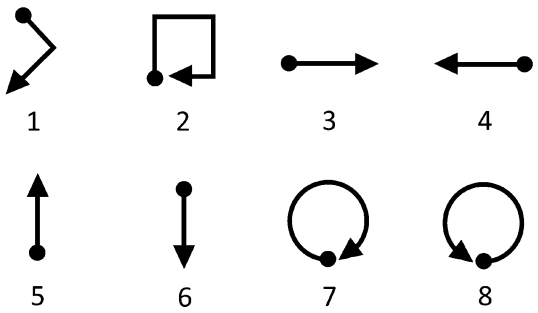

Fig. 5 Gesture set 1, geometric gestures, proposed by Kela et al. [20]

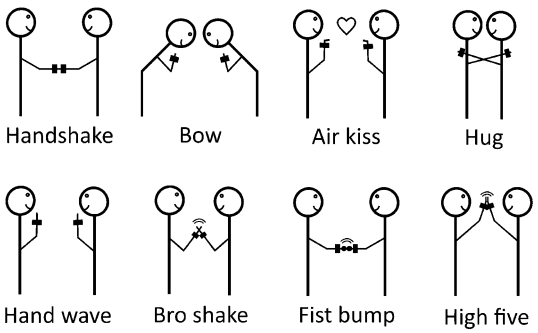

Fig. 6 Gesture set 2, greeting gestures

while the other represents a set of human to human gestures (Fig. 6). This is a small, well-known collection of greeting gestures from different cultures, but it is challenging enough to make simple heuristic approaches fail. To our knowledge, this collection of greeting gestures is an original proposal.

Since the geometric gestures have already been evaluated [25], we focus on the newly introduced greeting gestures. For every gesture, we collected 20 samples from 5 participants using a Sony Smartwatch 3, which makes a total of 800 samples. We tested our dataset under two different conditions: **user-dependent** and **user-independent**. In the user-dependent case, a test sample is matched against all the other samples, including those belonging to the same user and the same gesture class. In the user-independent case, a test sample cannot be matched to another sample that belongs to the same user and same gesture class.

The best recognition accuracy is achieved with a multi-dimensional DTW algorithm. The distance measure represents the cumulative distances of the three dimensions (x, y, and z) measured independently under DTW. Figure 7 shows the confusion matrix for the user-dependent case. The recognition accuracy was around 97%, with Precision $= 0.9739$, Recall $= 0.9736$, and $F_1 = 0.9736$. For the user-independent case (Fig. 8), the recognition accuracy drops significantly. The accuracy was around 53%, with Precision $= 0.5372$, Recall $= 0.5389$, and

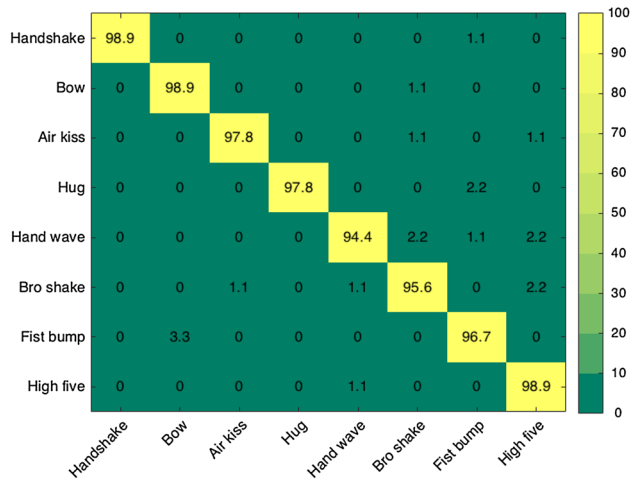

Fig. 7 Confusion matrix for **user-dependent** gesture recognition evaluation. Average accuracy 97%

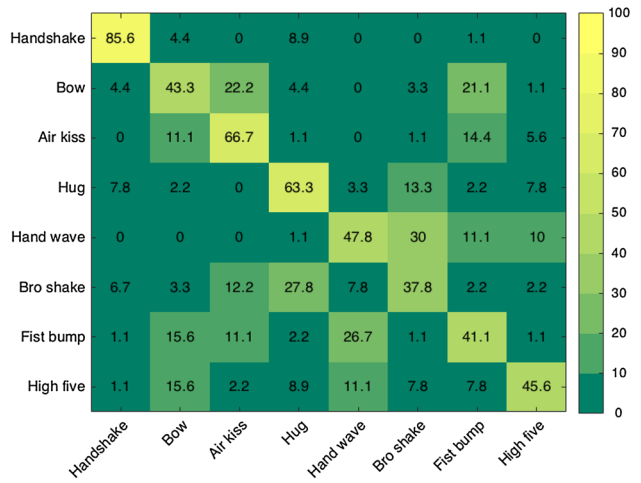

Fig. 8 Confusion matrix for **user-independent** gesture recognition evaluation. Average accuracy 53%

$F_1 = 0.5354$. These results were expected since a sample belonging to one user and gesture class can only be matched to samples belonging to different users. Moreover, the participants were not instructed to perform those eight greeting gestures in a particular way. Having the freedom to perform those gestures leads to more variance in the data.

The above experiments prove that DTW works well for user-dependent gesture recognition. If we further limit the scope of DTW to only the samples belonging to the same user and average the per-user results, the recognition accuracy is close to 99%.

Wearable devices are dependent on data quantization to speed up DTW. To find the optimal parameters for the quantization method, we compared both the window size and the step size, which are the most relevant parameters and measure the effect on both the recognition accuracy

and the running time of the algorithms. For each window size, we evaluate several step sizes. As expected, the recognition accuracy decreases when the window parameter is chosen too large (e.g. window size 2000 ms) or when the window size parameter is chosen too small (e.g. window size 10 ms).

Table 1 summarizes the running times of the optimal parameters (combination of window size and step size). We measured the time needed to recognize one gesture, namely to find the closest match of a new sample from the template dictionary. The running time only shows the time for the DTW algorithm, excluding the quantization of the data. This evaluation was performed on a PC with a Core 2 Quad CPU, 2.4 GHz. The table shows that the optimal window size is 250 ms, whereas the optimal step size is 200 ms. The optimal run time is about 7 ms. Performing the same experiment on the smartwatch resulted in a run time of about 700 ms. While the difference is significant, wearables are becoming more and more computationally powerful, so this should become less of an issue in the future.

Acoustic Ranging

We use inaudible chirp signals since they are not intrusive for the human ear. Given such a signal, the maximum distance that was measured using acoustic ranging was 23 m. Nevertheless, since our work focuses on close proximity multi-user activities, we narrow down the evaluation to distances up to 5 m.

Hardware Sensitivity

Speakers and microphones found on consumer mobile devices are supposed to be used to mainly reproduce and record human voice; thus, when using inaudible signals (signals in the ultrasound spectrum), they can show different sensitivity.

Figure 9 shows 4 frequency response plots relative to two smartphones and two smartwatches while recording a linear chirp, going from 200 Hz to 24 kHz, played from the same external speaker. The devices were placed at the same position while recording. As expected, the sensitivity of the microphones decreases if the frequency goes above 10 kHz. For one of the devices (Motorola 360 smartwatch), the sensitivity drastically decreases.

The sensitivity of the speakers is evaluated in a similar way as the sensitivity of the microphone. Each device plays a linear chirp, going from 200 Hz to 24 kHz, which is recorded using a high-quality microphone (Neumann TLM 102). Since the two smartwatches are not equipped with loudspeakers, we could perform this test only with the two smartphones. Compared to the microphones, the speakers do not vary much among the different devices (Fig. 10). Similarly, the sensitivity of the speakers decreases above a certain frequency, since most speakers are not designed for high-pitch sounds.

Indoor Ranging Precision

Most smartwatches nowadays are only equipped with a microphone for voice commands (i.e. ≤ 4 kHz). Due to the lack of loudspeakers on our smartwatch models, we were unable to test the acoustic ranging on these wearables.

Table 1 Comparison of accuracy and running time for different window sizes, evaluated on the first gesture set (Fig. 5)

Window (ms)	Accuracy (%)	Run time (ms)	Opt. step (ms)
2000	67.50	3.00	150
1000	88.75	4.94	200
500	92.50	20.80	100
250	92.50	7.74	200
100	91.25	248.41	30
80	92.50	96.71	50
50	91.25	257.88	30
25	90.00	514.32	20
10	90.00	1500.88	10

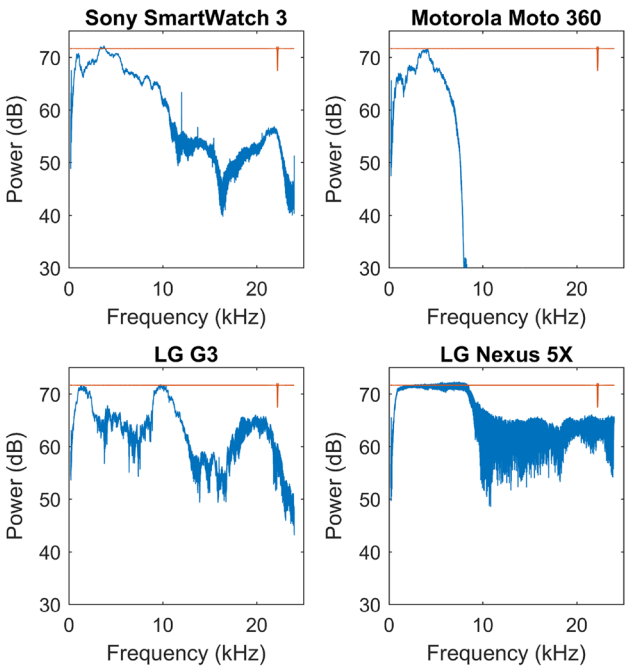

Fig. 9 Frequency response *plots* of a linear chirp recorded by the microphones of four different devices

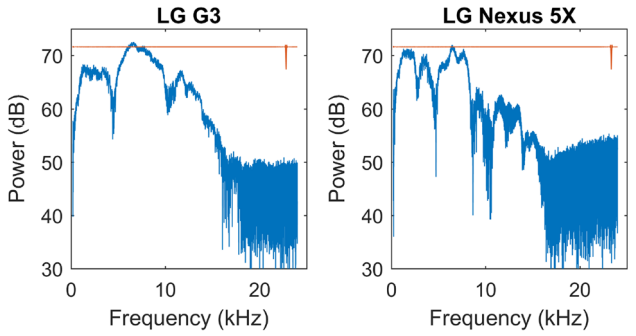

Fig. 10 Frequency response plots of a linear chirp played by the speakers of different mobile devices. The two smartwatches from Fig. 9 do not have a built-in speaker

Instead, we evaluated the acoustic ranging component on an LG G3 and an LG Nexus 5× smartphones.

The subplots of Fig. 11 show the evaluation of the precision of the acoustic ranging component at different distances [0.1, 1, 2, 3, 4, 5] m. The main focus of this evaluation are inaudible signals. Furthermore, to evaluate the method's reliability, multiple frequency bandwidths are tested in the inaudible spectrum (2000, 1000, 750, 500 and 250 Hz). Each configuration is tested for a duration of 5 min. This resulted in approximately 115 distance measurements for each configuration. The plots in the figure show the precision of the different configurations and the percentage of unsuccessful measurements (e.g. one or both devices could not detect the signal).

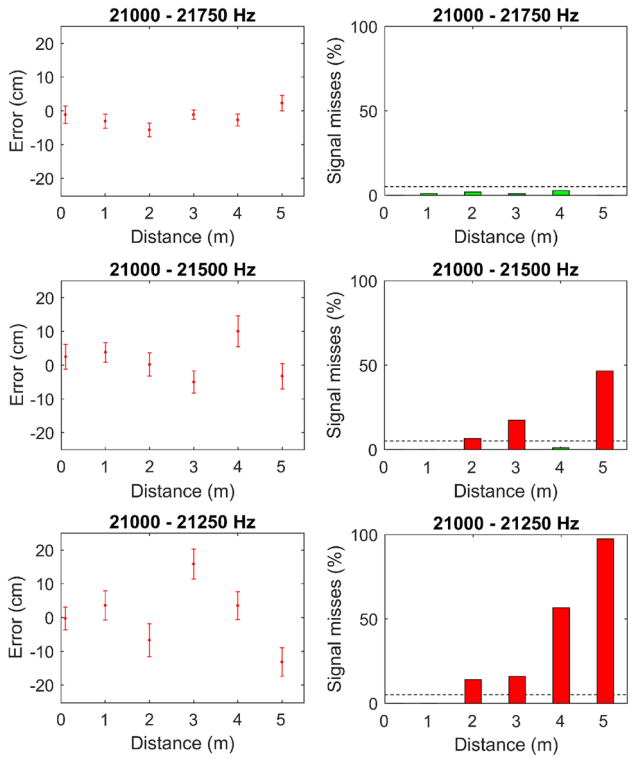

Fig. 11 Comparison of different frequency bandwidths. The term "signal miss" indicates the percentage of unsuccessful measurements. The *dotted line* indicates a probability of 5%. The *left side* shows the ranging errors are not significantly different, but the *right side* shows larger bandwidths are more reliable

Figure 12 shows the error deviation for all three timeframes at the above-mentioned distances. It is clear that the smallest timeframe (350 ms) is too short for the acoustic ranging, since the error deviation is for all distances higher than 150 cm. The other two timeframes, namely 550 and 750 ms, are significantly more robust, with a maximum error deviation of 99.9 cm for the 550 ms timeframe and 69.8 cm for the 750 ms timeframe at 5 metres.

Comparing the three timeframes, it seems that the 550 ms timeframe is the best trade-off between time and robustness.

Ranging Precision in Noisy Environments

The acoustic ranging component was also evaluated against environmental conditions, in two noisy and highly dynamic environments. Two LG Nexus 5× phones repeatedly measured the distance to each other until 32 successful measurements were collected. The duration of the chirp for recording was 550 ms, and the frequency bandwidth was 750 Hz (from 21,000 to 21,750 Hz). During these measurements, people passed through the line-of-sight of the two devices, meaning that the acoustic ranging was performed with obstacles in its way. For both scenarios, the experiment was repeated three times, each with 32 measurements.

The first scenario is indoors, at the entrance of a cafeteria at lunch time. The devices were placed on the ground, on both sides of a stairway (see Fig. 13), and the distance

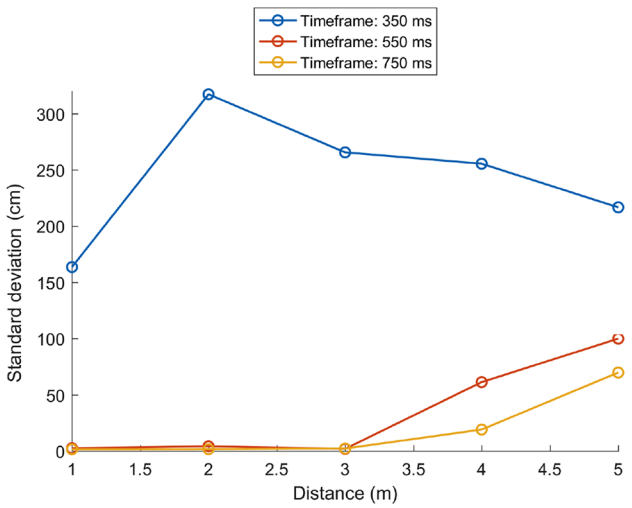

Fig. 12 Error deviation for different distances. The longer the recording timeframe, the smaller the deviation becomes, which makes the method more reliable

Fig. 13 Experiments in noisy environments

between the two devices was 3.5 m. The average measured distance was 350.86 cm, with a deviation of 49.79 cm and, 19.91% of unsuccessful measurements (e.g. one device did not hear the signal). Compared to the results of the quiet evaluation, for the same frequency spectrum, it can be seen that the number of unsuccessful measurements significantly increases in a noisy environment.

The second experiment was also performed in a cafeteria, but outdoors at a distance of 3.8 m. The average measured distance was 392.9 cm, with an average error of 12.9 cm, a standard deviation of 34.65 cm, and the ratio of unsuccessful measurement was 14.99%.

Acoustic Ranging with a Moving Device

This acoustic ranging experiment examines the distance measurement between a stationary and a moving device. The experiment is conducted using an LG Nexus 5× and an LG G3. The recording timeframe is 550 ms. This results in an approximate update rate of 1.56 s for each new estimation when using the two phones. The experiment is conducted in the following way (see Fig. 14): a mobile phone *s* is placed on a table in a quiet indoor location. A remote mobile device *r* is placed 5 m away. Afterwards, both devices continuously measure the distance to each other while *r* is moved on a direct path and with a constant speed towards *s*. After *r* reaches *s*, it is moved backwards on the same path and with the same speed towards its initial location.

The experiment is repeated for three different movement speeds: *slow* (≈0.3 m/s), *normal* (≈ 0.5 m/s), and *fast* (≈1 m/s). Figure 15 shows the observed distances as a function of time. The figure shows a clear path going from 5 m down to 0 m and then up to 5 m again for all three

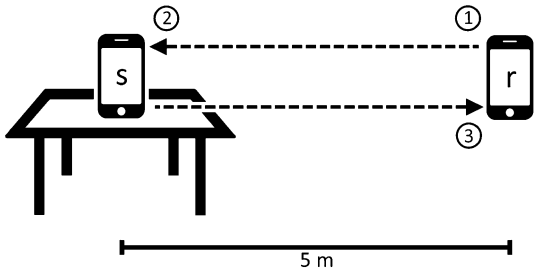

Fig. 14 Sequence of events during the experiment. *1* The phone *r* is placed 5 metres away from the phone *s*. *2* *r* moves towards *s* with a certain speed until it reaches the device. *3* *r* is moved backwards towards its initial position

movements. This indicates that the moment of interaction could be predicted only by looking at the distance between people.

Combined Evaluation Acoustic Ranging While Performing Hand Gestures

In the previous sections, we evaluated the two main components independently. Since the two components use a completely different set of sensors, there is no apparent dependency between the two. However, when performing in-air hand gestures, the mobile device is moved. The motion of the device can influence the reliability of the acoustic ranging component because it changes the way sound propagates through the medium. This section quantifies this influence.

We conducted an experiment similar to the one where we evaluate acoustic ranging with a moving device (Fig. 15). Additionally, the person carrying the remote device *r* continuously performs in-air hand gestures. The

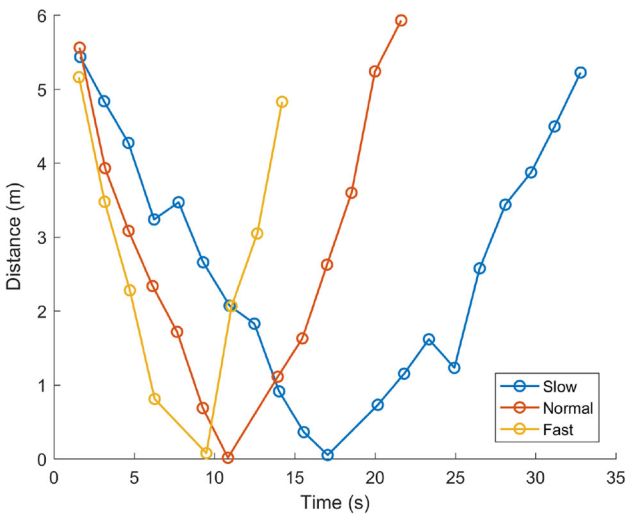

Fig. 15 Observed paths for different movement speeds. In each path, exactly one unsuccessful measurement occurred near the minimum when the devices were close to each other

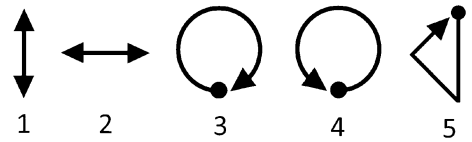

Fig. 16 Gesture set 3. Geometric gestures used for the combined evaluation of the main components, acoustic ranging while continuously performing hand gestures

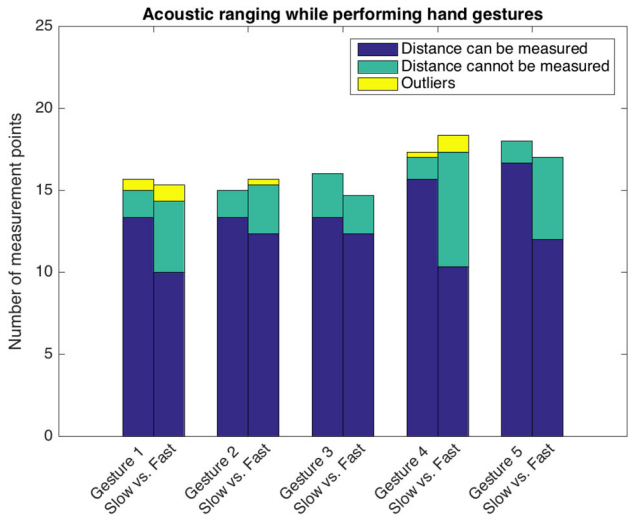

Fig. 17 An overall comparison of the number of distance measurements when performing different gestures with variable hand movement speed (*left* is slow; *right* is fast). The faster the movement, the fewer the number of times when the distance can be measured

geometric gestures (Fig. 5) are more suitable for this evaluation. They need, however, to be adapted to a repetitive movement, which can be performed continuously while one device moves towards the other and then backs away. Due to this constraint, we introduce a new set of five gestures (Fig. 16), which are the combination of the geometric gestures. Both devices used in this experiment were the Nexus 5×.

The experiment was performed 30 times: five different gestures, two different speeds for hand movement (slow and fast), each three times. The walking speed was kept constant, with each trial lasting between 25 and 30 s. A slow hand movement translates into about one gesture per second, while a fast hand movement generates about two to three gestures per second.

Figure 17 shows the average number of distance measurement points when performing each of the five gestures. For each setting, we counted the points when the distance can be measured, the number of points when the distance cannot be measured (due to one device not hearing the signal played by the other device), and the number of outliers (distance measurement larger than 15 m).

A more detailed analysis of the observed paths can be seen in Fig. 18. The two upper figures show two of the gestures from set 3, performed slowly while the distance between the devices decreases from 5 m to 0 and increases back to 5 m. The two lower graphs show two such gestures executed at higher speed. When the hand movement speed is increased, there are more situations when the distance cannot be measured, as well as an increase in the number of outliers.

The combined evaluation shows that the acoustic ranging component is influenced by the movement of the mobile device and depends on how fast the gesture is performed. Nevertheless, enough distance measurement points can be collected reliably, which can be used in the proposed interaction technique.

Applications

To demonstrate the feasibility and flexibility of our interaction technique, we implemented several application scenarios. Our first prototype was HandshakAR [3], an application where users can effortlessly share contact information when they perform the same greeting gesture and are close to each other. However, this application was limited to two participants. In this paper, we demonstrate three additional applications which are suitable for a small group of people. These examples also demonstrate the potential design space for future collocated multi-user gestural applications. For a more detailed demonstration, please refer to the supplementary video material.

Treasure Chest

This scenario shows a treasure chest that is part of a treasure hunt. The chest can only be opened if the whole team, a group of three or more children, are physically close to it (within 1 m) and perform the same hand gesture (Fig. 19). If one of the children is further away or if they do not perform the hand gesture together with their teammates, the chest stays locked.

Collaborative Fitness

Fitness activities like running have a strong social component. In this application scenario, teams of friends can compete against one another and compare different statistics like running distance, number of steps. The application tracks the statistics of each individual member. Additionally, if the team members run together (are in proximity), their counts will be accumulated towards a group total (Fig. 20). If one of the members is slower or much faster and separates from the group, the points will not be

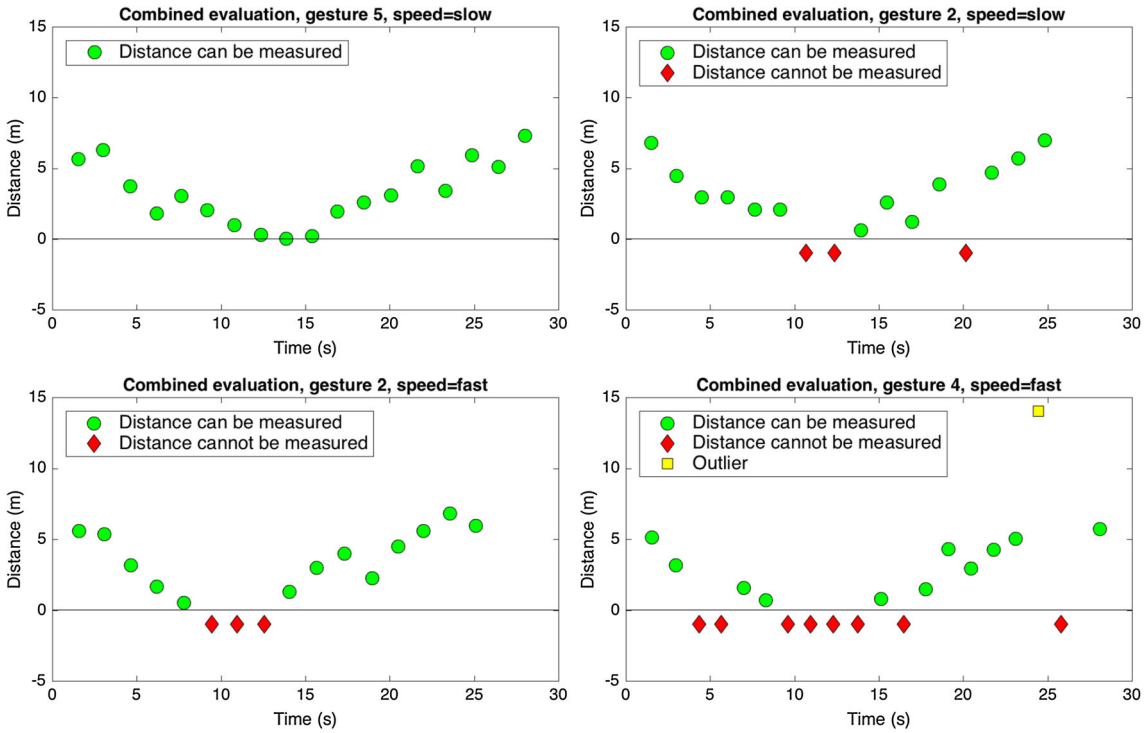

Fig. 18 A selection of 4 out of the total 30 observed paths which highlight different reliability cases. For example, the *upper left* figure shows no points where the distance cannot be measured, while in the *lower right* figure there are many such points

Fig. 19 First application scenario: Interacting with a magical treasure chest. Children can open the treasure chest when they are physically close to one another and perform the same hand gesture (spiral like motion with their device). The treasure chest is just a metaphor for any smart or digital object

Fig. 20 Second application scenario: Friends running together. Each device counts the individual number of steps and running distance. When teammates are in proximity to one another, their individual counts are accumulated towards the total group count

counted towards the group score. This creates an incentive for the members to work as a team.

Collaborative Music Band

The third application scenario is built around music. When friends get together, they can create an ad hoc band and simulate playing an instrument (e.g. air guitar) with only their mobile or wearable device. Each participant can play one instrument at a time. The instruments do not have to be allocated in advance. The first user who performs a certain gesture (e.g. play the drums) will control that specific instrument (Fig. 21). Our prototype supports two different songs and multiple instruments (e.g. drums, piano, or guitar).

Discussion

In this section, we discuss the advantages and the limitations of the building blocks that support collocated multi-user gestural interactions.

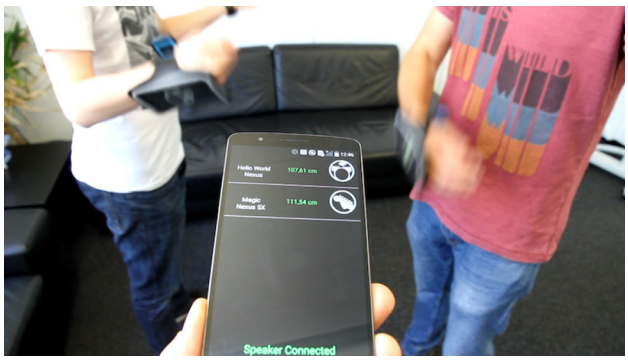

Fig. 21 Third application scenario: An ad hoc music band. A group of friends can create a band and play different instruments using their own devices. To play a specific instrument, a user has to simulate the motion of that instrument with their hands

Gesture Recognition via Motion Sensors

We presented a method based on DTW to recognize hand gestures as user input modality. This approach does not involve any learning, but it has been shown to provide good results for user-dependent gesture recognition. This entails that each user has to record a set of template gestures before using the system. In such a scenario, the method achieved a recognition rate close to 99%. With the proposed applications, we have shown the flexibility of the approach to reliably recognize different gestures from different domains.

Proximity Detection via Acoustic Ranging

Our proximity detection is based on two-way ranging using inaudible signals. This method can estimate the distance between two devices at once. The maximum distance we could measure was 23 m. In a quiet indoor environment, the method achieves an error below 10 cm for distances of up to 5 m, measured with two smartphones. For most group applications, these "guarantees" are sufficient. In particular, this method estimates the distance better than methods based on RSSI values (using off-the-shelf devices). A limitation of the provided method is the lack of simultaneous ranging of multiple devices since one cannot distinguish the source of multiple audio signals. To support multiple users, ranging is done in a round-robin fashion.

Further Aspects and Limitations

One important requirement of our method is the need for a microphone and a speaker that work in the inaudible range. This also influences the hardware and, as we have seen in the evaluation section, devices like the Motorola 360 smartwatch are not sensitive enough to capture such signals. Moreover, both smartwatches do not have any loudspeakers, which is why our prototypes have been implemented using smartphones. However, future devices are expected to have additional and better sensors. One limitation of Bluetooth is the power consumption which is addressed by the newer version of the standard, BLE. Extended use of the microphone, speaker, and motion sensors can have a significant impact on the battery. In the future, we plan to investigate the impact of our methods on mobile and wearable devices in terms of power consumption.

Conclusion

We presented a collocated multi-user gestural interaction technique with unmodified mobile and wearable devices. We support the development of new interaction possibilities that bring people physically close to one another. Proximity is detected with inaudible signals, hand gestures are recognized from motion sensors, and communication between the devices is handled over Bluetooth. All components are unobtrusive and do not break the interaction experience.

Our in-depth evaluation of the underlying components shows that the proposed interaction technique is feasible on unmodified devices. There are certain hardware limitations, namely the lack of loudspeakers or low-pass filters on microphones on some wearables, which hindered the end-to-end evaluation on smartwatches. However, new wearables might overcome these limitations and our work can be transferred to fully support these devices. Finally, we showcased the practical applicability of having collocated multi-user gestural interactions with three real-world applications.

Compliance with ethical standards

Conflicts of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Arduser L, Bissig P, Brandes P, Wattenhofer R (2016) Recognizing text using motion data from a smartwatch. In: Proceedings of the international conference on pervasive computing and communication workshops, PerCom '16 workshops, pp 1–6
2. Arefin Shimon SS, Lutton C, Xu Z, Morrison-Smith S, Boucher C, Ruiz J (2016) Exploring non-touchscreen gestures for smartwatches. In: Proceedings of the ACM conference on human factors in computing systems, CHI '16
3. Bâce M, Sörös G, Staal S, Corbellini G (2017) HandshakAR: wearable augmented reality system for effortless information sharing. In: Proceedings of the 8th augmented human

international conference, AH '17. ACM, New York, NY, USA, pp 34:1–34:5
4. Carter M, Velloso E, Downs J, Sellen A, O'Hara K, Vetere F (2016) PathSync: multi-user gestural interaction with touchless rhythmic path mimicry. In: Proceedings of the ACM conference on human factors in computing systems, CHI '16
5. Chi PYP, Li Y, Hartmann B (2015) Enhancing cross-device interaction scripting with interactive illustrations. IEEE Trans Hum–Mach Syst 45:263–271
6. Curtis P, Banavar MK, Zhang S, Spanias A, Weber V (2014) Android acoustic ranging. In: Proceedings of the international conference on information, intelligence, systems and applications, IISA '14
7. Esteves A, Velloso E, Bulling A, Gellersen H (2015) Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In: Proceedings of the ACM symposium on user interface software technology, UIST '15
8. Fang B, Lane ND, Zhang M, Boran A, Kawsar F (2016) BodyScan: enabling radio-based sensing on wearable devices for contactless activity and vital sign monitoring. In: Proceedings of the international conference on mobile systems, applications, and services, MobiSys '16, pp 97–110
9. Fu B, Karolus J, Grosse-Puppendahl T, Hermann J, Kuijper A (2015) Opportunities for activity recognition using ultrasound doppler sensing on unmodified mobile phones. In: Proceedings of the international workshop on sensor-based activity recognition and interaction, WOAR '15, pp 8:1–8:10
10. Goel M, Lee B, Islam Aumi MT, Patel S, Borriello G, Hibino S, Begole B (2014) Surfacelink: Using inertial and acoustic sensing to enable multi-device interaction on a surface. In: Proceedings of the ACM conference on human factors in computing systems, CHI '14, pp 1387–1396
11. Greenberg S, Marquardt N, Ballendat T, Diaz-Marino R, Wang M (2011) Proxemic interactions: The new ubicomp? Interactions 18:42–50
12. Hall E (1990) The hidden dimension. A doubleday anchor book. Anchor Books, New York
13. Hinckley K (2003) Synchronous gestures for multiple persons and computers. In: Proceedings of the ACM symposium on user interface software and technology, UIST '03
14. Hinckley K, Ramos G, Guimbretiere F, Baudisch P, Smith M (2004) Stitching: Pen gestures that span multiple displays. In: Proceedings of the working conference on advanced visual interfaces, AVI '04. ACM, New York, NY, USA, pp 23–31
15. Holmquist LE, Mattern F, Schiele B, Alahuhta P, Beigl M, Gellersen HW (2001) Smart-its friends: a technique for users to easily establish connections between smart artefacts. In: Proceedings of the international conference on ubiquitous computing, UbiComp '01
16. Houben S, Marquardt N (2015) Watchconnect: a toolkit for prototyping smartwatch-centric cross-device applications. In: Proceedings of the ACM conference on human factors in computing systems, CHI '15
17. Jin H, Holz C, Hornbæk K (2015) Tracko: Ad-hoc mobile 3d tracking using bluetooth low energy and inaudible signals for cross-device interaction. In: Proceedings of the ACM symposium on user interface software technology, UIST '15
18. Jokela T, Chong MK, Lucero A, Gellersen H (2015) Connecting devices for collaborative interactions. Interactions 22(4):39–43
19. Kang W, Han Y (2015) SmartPDR: smartphone-based pedestrian dead reckoning for indoor localization. IEEE Sens. J. 15(5):2906–2916
20. Kela J, Korpipää P, Mäntyjärvi J, Kallio S, Savino G, Jozzo L, Marca SD (2006) Accelerometer-based gesture control for a design environment. Personal Ubiquitous Comput 10(5):285–299
21. Kim J, He J, Lyons K, Starner T (2007) The gesture watch: a wireless contact-free gesture based wrist interface. In: Proceedings of the IEEE international symposium on wearable computers, ISWC '07
22. Kratz S, Rohs M (2009) HoverFlow: expanding the design space of around-device interaction. In: Proceedings of the international conference on human-computer interaction with mobile devices and services, MobileHCI '09, pp 4:1–4:8
23. Kriara L, Alsup M, Corbellini G, Trotter M, Griffin JD, Mangold S (2013) RFID shakables: pairing radio-frequency identification tags with the help of gesture recognition. In: Proceedings of the ACM conference on emerging networking experiments and technologies, CoNEXT '13
24. Lee-Cosio BM, Delgado-Mata C, Ibanez J (2012) Ann for gesture recognition using accelerometer data. Procedia Technol 3:109–120
25. Liu J, Zhong L, Wickramasuriya J, Vasudevan V (2009) uWave: accelerometer-based personalized gesture recognition and its applications. Pervasive Mob Comput 5(6):657–675
26. Lucero A, Holopainen J, Jokela T (2011) Pass-them-around: Collaborative use of mobile phones for photo sharing. Proceedings of the SIGCHI conference on human factors in computing systems, CHI '11. ACM, New York, NY, USA, pp 1787–1796
27. Lucero A, Keränen J, Korhonen H (2010) Collaborative use of mobile phones for brainstorming. In: Proceedings of the 12th international conference on human computer interaction with mobile devices and services. MobileHCI '10. ACM, New York, NY, USA, pp 337–340
28. Marquardt N, Ballendat T, Boring S, Greenberg S, Hinckley K (2012) Gradual engagement: Facilitating information exchange between digital devices as a function of proximity. In: Proceedings of the ACM international conference on interactive tabletops and surfaces, ITS '12, pp 31–40
29. Marquardt N, Diaz-Marino R, Boring S, Greenberg S (2011) The proximity toolkit: Prototyping proxemic interactions in ubiquitous computing ecologies. In: Proceedings of the ACM symposium on user interface software and technology, UIST '11, pp 315–326
30. Marquardt N, Hinckley K, Greenberg S (2012) Cross-device interaction via micro-mobility and f-formations. In: Proceedings of the ACM symposium on user interface software and technology, UIST '12
31. McIntosh J, McNeill C, Fraser M, Kerber F, Löchtefeld M, Krüger A (2016) EMPress: practical hand gesture classification with wrist-mounted EMG and pressure sensing. In: Proceedings of the ACM conference on human factors in computing systems, CHI '16
32. Moosavi-Dezfooli SM, Pignolet YA, Dzung D (2016) Simultaneous acoustic localization of multiple smartphones with euclidean distance matrices. In: Proceedings of the international conference on embedded wireless systems and networks, EWSN '16, pp 41–46
33. Peng C, Shen G, Zhang Y (2012) BeepBeep: a high-accuracy acoustic-based system for ranging and localization using COTS device. ACM Trans. Embed. Comput. Syst. 11(1):4:1–4:29
34. Peng C, Shen G, Zhang Y, Li Y, Tan K (2007) Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In: Proceedings of the ACM international conference on embedded networked sensor systems, SenSys '07
35. Pylv T, Pylvänäinen T (2005) Accelerometer based gesture recognition using continuous HMMs. Pattern Recognit Image Anal 3522:639–646
36. Qiu J, Chu D, Meng X, Moscibroda T (2011) On the feasibility of real-time phone-to-phone 3D localization. In: Proceedings of the ACM conference on embedded networked sensor systems, SenSys'11, p 190

37. Rädle R, Jetter HC, Marquardt N, Reiterer H, Rogers Y (2014) Huddlelamp: spatially-aware mobile displays for ad-hoc around-the-table collaboration. In: Proceedings of the 9th ACM international conference on interactive tabletops and surfaces. ITS '14. ACM, New York, NY, USA, pp 45–54
38. Rekimoto J (2004) SyncTap: synchronous user operation for spontaneous network connection. Personal Ubiquitous Comput 8(2):126–134
39. Shen S, Wang H, Roy Choudhury R (2016) I am a Smartwatch and I can Track my User's Arm. In: Proceedings of the ACM international conference on mobile systems, applications, and services, MobiSys '16
40. Song J, Sörös G, Pece F, Fanello SR, Izadi S, Keskin C, Hilliges O (2014) In-air gestures around unmodified mobile devices. In: Proceedings of the ACM symposium on user interface software and technology, UIST '14, pp 319–329
41. Van Vlaenderen W, Brulmans J, Vermeulen J, Schöning J (2015) Watchme: a novel input method combining a smartwatch and bimanual interaction. In: Proceedings of the ACM conference extended abstracts on human factors in computing systems, CHI EA '15, pp 2091–2095
42. Wang H, Lai TTT, Roy Choudhury R (2015) MoLe: motion leaks through Smartwatch sensors. In: Proceedings of the international conference on mobile computing and networking, MobiCom '15, pp 155–166
43. Wen H, Ramos Rojas J, Dey AK (2016) Serendipity: finger gesture recognition using an off-the-shelf smartwatch. In: Proceedings of the ACM conference on human factors in computing systems, CHI '16
44. Wu J, Pan G, Zhang D, Qi G (2009) Gesture recognition with a 3-d accelerometer. In: Proceedings IEEE International Conference on Ubiquitous Intelligence and Computing, UIC '09, pp 25–38
45. Zhang C, Yang J, Southern C, Starner TE, Abowd GD (2016) WatchOut: extending interactions on a smartwatch with inertial sensing. In: Proceedings of the ACM International Symposium on Wearable Computers, ISWC '16