

In-air ultrasonic 3D-touchscreen with gesture recognition using existing hardware for smart devices

Bert Van Dam*, Yuri Murillo*, Min Li[†] and Sofie Pollin*

*KU Leuven, Department of Electrical Engineering - ESAT

Kasteelpark Arenberg 10, Bus 2441, 3001 Leuven, Belgium

[†] NXP Semiconductors, Interleuvenlaan 80, 3001 Leuven, Belgium

Abstract—A versatile and intuitive way of interacting with smart devices consists of a 3D-touchscreen in the air above the device, combined with gesture recognition. In this paper, the limitations of hand tracking and gesture recognition using existing hardware in smartphones will be investigated. To accomplish hand tracking and gesture recognition, the two loudspeakers of the device send ultrasonic tones in combination with a Maximum Length Sequence (MLS), after which the microphones pick up the potentially altered reflected signal again. The Doppler-response on the tones yields information about the speed of the hand above the device and the acoustic channel estimation obtained from the MLS estimates the distance at which the hand is present. The position estimation module estimates the 3D-position of the hand in a predefined grid of eight regions and hereby achieves an accuracy above 77 % in all cases. For the gesture recognition, features are extracted from both the Doppler-response as well as the acoustic channel estimation. A simple decision tree then classifies the performed gesture. Experimental results show that the accuracy of the gesture recognition module achieves an accuracy between 87 and 100 %.

Keywords—HCI, ultrasound, gesture recognition, hand tracking, Doppler, acoustic channel estimation, MLS, smart devices

I. INTRODUCTION

Gesture recognition is getting more and more attention as a key technique for interacting with smart devices, alongside the traditional touchscreen and voice recognition [1]. Because most touchscreens of these devices are only a few fingers wide, a virtual "3D-touchscreen" would be a very attractive alternative [2]. Such a gestural interface above the device would allow a user to control his/her smart device without touching it, just by performing gestures in mid-air. Many different techniques have already been explored to achieve gesture recognition, including image processing [3], RF-based [4] and ultrasound [5]. Hand tracking however has been attempted in 2D by [2], but to the author's knowledge, 3D hand tracking using ultrasound in smart devices has not been attempted before. In this work, the limits for gesture recognition and hand tracking using only the two loudspeakers and two microphones already present in most high-end smart devices like the iPhone 5 or Samsung Galaxy Note 4 will be investigated.

The paper is structured as follows: in section II, an overview of the related literature is given. Section III discusses the proposed framework and section IV describes the proposed approach on how to recognize gestures or estimate the hand position using this framework. Finally, section V introduces the hardware used for developing our testbed and provides the accuracy results.

II. RELATED WORK

A gesture recognition-implementation which only leverages one microphone and one loudspeaker of a smartphone and makes use of the Doppler effect, was introduced by [5] as "Dolphin". A single 21 kHz tone is emitted by the loudspeaker, after which the microphone receives the potentially altered signal again. Both speaker and microphone have a sampling frequency of 48 kHz. If a hand or other object is moving above the smartphone, other frequencies are present in the frequency domain of the received signal as a result of the Doppler effect. These frequencies are higher than 21 kHz if the hand moves towards the microphone and lower if the hand moves away, respectively according to formulas (1) and (2) [5], [6]. In these formulas, v_s is the speed of sound, v_h is the speed of the hand and f_{tr} and f_{rec} are transmitted and received tone's frequencies respectively.

A gesture consisting of a sequence of movements over time, can be represented by a vector of +1 and -1's, according to the direction of the movement in each time bin. This vector is then compressed into a motion indicating sequence, which consists only of discontinuous +1 and -1's. Using this approach, the performed gesture is first classified in a gesture group according to the direction vector, after which the original spectrum vector is used for more fine-grained machine classification. The classification accuracy is as high as 94 % over a set of 24 predefined gestures. This last result is obtained after a thorough investigation of which machine learning technique performed best on the provided features.

$$f_{rec} = \left(\frac{v_s + v_h}{v_s - v_h} \right) \cdot f_{tr}, \quad (1)$$

$$f_{rec} = \left(\frac{v_s - v_h}{v_s + v_h} \right) \cdot f_{tr}, \quad (2)$$

Another attempt to recognize gestures above a mobile device using only microphones and loudspeakers, is Microsoft's "SoundWave" [6]. SoundWave also plays tones between 18 and 22 kHz to extract the speed of a target using a Doppler-analysis, similar to [5]. Also, the proximity and size of the hand can be estimated from the amplitude of the observed reflected signal. All the above properties are measured over time, which allows to classify a scrolling gesture, a single- or double tap, a two-handed seesaw (the hands moving opposite directions) and a sustained motion gesture. SoundWave works on a number of different laptops and tablets and reaches accuracies between 87 and 100 % depending on the target.

The only paper the author is aware of that discusses hand tracking in smartphones using ultrasonic technology, is FingerIO [2]. The creators present a way to track the motion of a fingertip in 2D. They transform the device into an active sonar system which transmits 18 - 20 kHz sound waves with a speaker. A nearby finger causes additional reflections of these signals, which are picked up by the two microphones. The arrival time of these reflections at both microphones depends on the position of the finger. This difference in arrival time can be estimated using a key property of Orthogonal Frequency Division Multiplexing (OFDM), namely that a sample error in identifying the beginning of the symbol translates linearly into phase changes in the frequency domain. Using this scheme, the sample offset between transmitted and received signal can be extracted and thus the time elapsed between them is known. Also knowing the speed of sound allows calculating the distance travelled by the sound signal. Combining the distance measurements from both microphones and constraining the search-space beforehand to one side of the considered device leads to an unambiguous position estimation in 2D-space.

The advantages of using ultrasound methods for gesture recognition or hand tracking are in general the low computational power this solution requires [5], the availability of the needed hardware and the lack of requiring special wearable equipment. This approach is also not disturbed by ordinary noise or light [1] and it provides a wider operating range and angle than vision-based approaches. All these factors make it an attractive solution for implementation on portable devices. [5], [7] A disadvantage of the ultrasound method is that an accurate distance measurement for a fast-changing environment is difficult to calculate because of the relatively low sampling frequency in comparison with the speed of sound [2]. Also, the static hand position can roughly be derived, but it is not possible to derive the shape of the hand. Static gesture recognition will thus be impossible with this method, and as a result computer vision techniques are much more suited for this purpose.

III. FRAMEWORK

This section discusses the framework, as shown in Figure 1, that is used throughout this work. It consists of three main building blocks: a Doppler information gathering block, an acoustic channel estimation block and a gesture recognition/position estimation block. The first two gather relevant information about the environment of the smart device and the last one processes this information such that a predefined gesture can be recognized or the hand position can be estimated. The two information gathering blocks perform active sensing, which means they each send a specific signal and analyze its response. These signals are transmitted and received at the same time in buffers of 2048 samples at a sampling frequency (f_s) of 48 kHz. Section III-C further discusses these choices of parameters.

A. Doppler information

The first information gathering block aims to extract the speed of an object moving above the smart device and makes use of the Doppler-effect to do so. One speaker emits a 21 kHz-tone, while the other emits a 22.031 25 kHz-tone. These exact frequencies are chosen because they assure there are integer

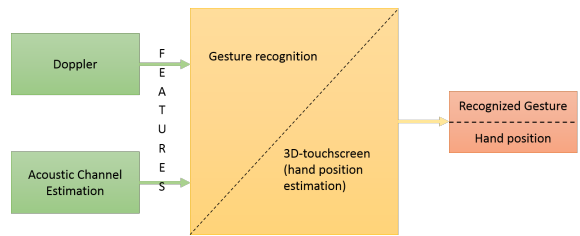


Fig. 1: The two blocks on the left collect information on the environment, of which features are extracted. These are passed to the central block, which recognizes the performed gesture or estimates the hand position.

multiples of cycles of the sine wave per buffer, 896 and 940 respectively. Thus, a clear peak in the frequency domain without sidelobes caused by discontinuities at the edges of each buffer is obtained. A moving hand above the smart device causes reflections with higher or lower frequencies than the emitted ones. The speed of this hand can then be derived according to equation (3), where f_{rec} and f_{tr} are the received and transmitted frequencies in Hz and v_s and v_h are the speed of sound and the speed of the hand in m s^{-1} .

$$v_h = \frac{\left(\frac{f_{rec}}{f_{tr}}\right) - 1}{\left(\frac{f_{rec}}{f_{tr}}\right) + 1} \cdot v_s, \quad (3)$$

Figure 2 shows a two seconds long spectrogram of the received signal around the emitted 21 and 22 kHz tones. The duration of one time bin is 2048 samples, which corresponds to 42.67 ms. The frequency resolution per bin is then $\frac{f_s}{N} = 23.4375$ Hz, which provides a speed resolution of 0.2 m s^{-1} according to equation (3). Experimental results show that the average hand speed during a gesture is about 1.2 m s^{-1} , so this accuracy suffices. The speed is computed for each time bin during this two seconds and plotted versus time, which can be seen in Figure 3.

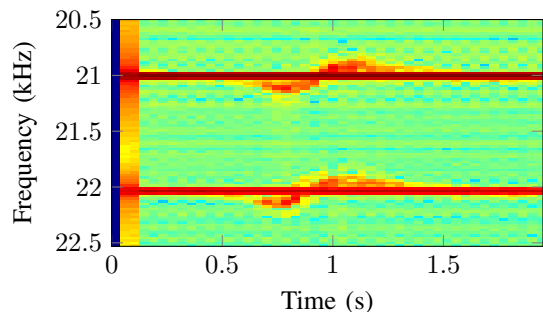


Fig. 2: A spectrogram of the received signal, where the transmitted tone and the Doppler-shifted tones can be distinguished. Here, a sweep from bottom to top has been performed.

1) *Speed features*: Many possible features can be extracted from the graph in Figure 3, of which some are listed next. The duration of the gesture is calculated as the difference between the first and the last index at which the speed is nonzero. The

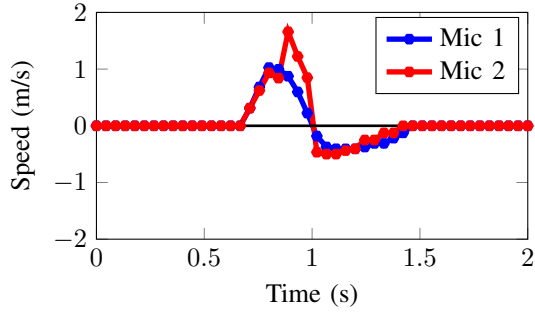


Fig. 3: The speed of a hand above the device, calculated using the dominant frequencies in the spectrogram above.

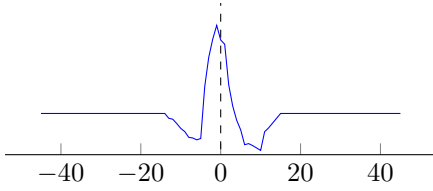
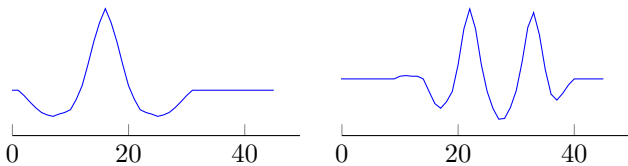


Fig. 4: The cross-correlation of the speed vectors depicted in Figure 3. The index of the peak is negative, which means the hand has crossed the bottom microphone first, before crossing the top microphone.

minimum and maximum speed(s) are the positive and negative peaks and the time difference of when these occur are the differences between the indices of these peaks multiplied by the sampling frequency. The power of these speed signals is calculated as $\sum \mathbf{v}_i^2$, with \mathbf{v}_i the speed vector of microphone i . Also, the speed vectors for both microphones are correlated to get information about the time difference of the hand crossing both microphones. Depending on the sign of the index of the maximum of this cross-correlation, the hand has crossed the top or the bottom microphone first. Figure 4 shows a plot of this cross-correlation. Last, the total signal is correlated with one cycle of moving towards and away from the microphone in this same signal. The number of peaks in this correlation indicates how many taps the user performed, as can be seen in Figure 5. This is a key feature for detecting whether a double- or triple tap has been performed.



(a) When no doubletap is performed, only one peak can be distinguished. (b) When a doubletap is performed, two peaks can be distinguished

Fig. 5: Cross-correlations of one tap with the total speed vector for one microphone.

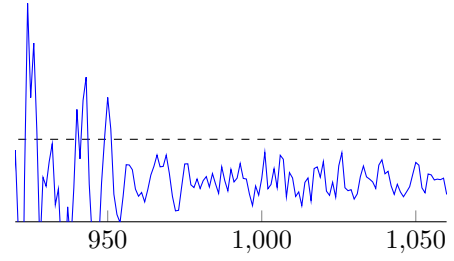


Fig. 6: Impulse response of the acoustic channel calculated as the cross correlation of the transmitted and received signals. The first peak in this IR is the direct path signal. Subsequent peaks are caused by reflections of a hand above the microphone.

B. Acoustic Channel Estimation

The second information gathering block tries to obtain some distance measurements between the hand and the speakers and microphones. When a Maximum Length Sequence (MLS) of 2048 samples is transmitted by the loudspeakers and received again by the microphones, the impulse response (IR) of the channel can easily be calculated as the cross-correlation of the two [8], [9]. The reason a MLS is chosen for this purpose is because of its good autocorrelation properties. If the impulse response of a system is $h[n]$ and the MLS is $s[n]$, then the output of a linear time-invariant system is given by $y[n] = h[n] * s[n]$. Correlating with the MLS on both sides of this equation gives $\phi_{sy} = h[n] * R_{ss}$ and because $R_{ss}(\tau) \approx \delta$, $\phi_{sy} \approx h[n]$ is valid [10]. Figure 6 shows a plot of this IR. Because the transmitter and receiver are not synchronized, the exact timing of when the transmission began is unknown to the receiver. However, the direct path peak is extracted from the IR and the relative positions of the sensors are known, so a correction term can be added to the distance estimation. This correction term consists of the time t_{corr} the signal has travelled from the speaker to the corresponding microphone. Reflections in the calculated IR show up as peaks, and the timing of these peaks leads to a distance estimation as in equation (4).

$$d = \frac{N + M}{f_s} \cdot v_s, \quad (4)$$

In the above equation, N is the number of samples between the direct path component and the reflected signal and M is the number of samples corresponding to the correction term t_{corr} . This way d_{ij} , which is the distance the sound wave traveled from speaker i to the hand and then to microphone j , can be calculated for all four possible combinations of d_{ij} . The maximum distance to be detected is set to be 17 cm, as for greater distances the peaks in the cross-correlation due to reflections of the hand can no longer be distinguished.

Another possibility for the gesture recognition is to only measure two distances, being d_{11} and d_{22} . The reason for this is that gestures are often performed too fast to get decent consecutive distance measurements.

1) *Distance features*: First, the distance vectors containing d_{11} and d_{22} for each buffer in the two seconds long interval are

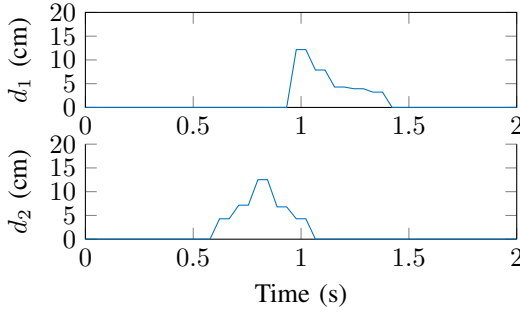


Fig. 7: The distance of the hand from the top and bottom microphone respectively. It can be seen that the hand has passed the bottom microphone first.

subtracted from the maximum to be detected distance in order to get a vector that starts and ends at zero amplitude ($\mathbf{d}_i = 17 - \mathbf{d}_i$). As such, the cross-correlation of the distance vectors from each microphone becomes more readable. The resulting vectors per microphone are plotted in Figure 7. An overview of the features that can be extracted from these distance vectors is given next. The differences between the first and last indices where the signal is nonzero gives an indication for the duration of the gesture. The peaks in these vectors correspond to the minimum distances between the hand and each microphone during a gesture and their indices indicate when this occurs. Similar to the speed vectors, the power of each distance vector is calculated as $\sum \mathbf{d}_i^2$. Last, correlating the distance vectors of both microphones in exactly the same way as the speed vectors also yields valuable timing information.

C. System parameters

A summary of the most important system parameters of the framework can be found in Table I. For this proof-of-concept, the sampling frequency is kept at 48 kHz and thus the MLS is situated in the hearable spectrum. For a real implementation however, it is advised to set the sampling frequency to at least 96 kHz, so the MLS can be placed in the ultrasound spectrum. This is not possible in Matlab because the incoming samples cannot be processed fast enough at this rate. The buffersize is chosen to be 2048 samples, which is the lowest possible for the algorithm to be able to run in Matlab. A smaller buffersize would provide faster subsequent distance measurements due to a shorter MLS without having an impact on the speed measurements, provided the time bins in the spectrogram are chosen equally long. However, the Playrec utility does not perfectly synchronize the transmitter and receiver so a considerable part of the transmitted signal would be lost. To minimize the relative size of this lost part, the buffersize should be large enough.

System parameters	
Sampling frequency f_s	48 kHz
Doppler frequencies	21 kHz / 22.031 25 kHz
Bandwidth MLS	0 → 18 kHz
Buffersize B	2048

TABLE I

IV. GESTURE RECOGNITION & POSITION ESTIMATION

This section investigates how to recognize a set of gestures using the most relevant of all possible features extracted from the gathered information and how to estimate the hand position using the distance measurements.

A. Gesture Recognition

Considering all the extracted features, the following gestures are proposed to be recognized:

Sweeps	Taps	Other
Left - Right	Double tap @ bottomside	Downwards - Upwards
Right-Left	Double tap @ topside	Upwards - Downwards
Bottom-Top		
Top-Bottom		

TABLE II: The different gestures

The recognition of these gestures can be performed using a heuristic method as well as a machine learning method. The main goal of this work is not to achieve the best possible gesture recognition as such, nor find out what the best machine learning algorithms are for this purpose, but rather gaining a good insight in the possible features which best characterise a set of gestures and paving the way for future implementations on gesture recognition using a similar framework. Thus, a simple heuristic decision tree is built as gesture recognizer. Important features in this tree are the power of the speed- and distance vectors, the cross correlation of one tap with the total speed vector and the index of the maximum in the cross correlation of the speed vectors. A few possibilities for machine learning methods would be a trained decision tree or random forest, a Hidden Markov Model, an Artificial Neural Network or a Support Vector Machine. Given the decent results using the decision tree, it is expected that a random forest will provide even better results. The implementation of any of this methods however is left for future work.

B. Position estimation

In section III-B, four distance measurements have been performed to be able to estimate the three position coordinates of a user's hand in a coordinate system fixed to the smart device. The measured distances can be written as follows, with $\mathbf{S}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ of speaker $i \in 1, 2$ the 3D position coordinates of the speakers. Similarly, \mathbf{M}_i are the coordinates of the microphones and \mathbf{H} the coordinates of the hand.

$$\begin{cases} d_{11} = \|\mathbf{S}_1 - \mathbf{H}\| + \|\mathbf{M}_1 - \mathbf{H}\| \\ d_{22} = \|\mathbf{S}_1 - \mathbf{H}\| + \|\mathbf{M}_2 - \mathbf{H}\| \\ d_{22} = \|\mathbf{S}_2 - \mathbf{H}\| + \|\mathbf{M}_2 - \mathbf{H}\| \\ d_{21} = \|\mathbf{S}_2 - \mathbf{H}\| + \|\mathbf{M}_1 - \mathbf{H}\| \end{cases}$$

As the speaker- and microphone positions are known, there are only 3 unknowns in this system: the x-, y- and z-coordinate of the user's hand. So actually only 3 distances would be required for it to have an exact solution. There are four

distances measured however, which provide some redundant information. This redundancy can be exploited to achieve a more accurate position estimation in the presence of noise. Instead of solving a system of three equations exactly, all four measured distances are used to compose an overdetermined system, which is then solved in the least-squares sense. This reduces the bias error and variance on the measurements and thus introduces some noise reduction on the estimation.

V. HARDWARE SETUP AND RESULTS

For this proof-of-concept, a computer runs the gesture recognition and position estimation algorithm in Matlab. The Playrec utility [11] is used for playing and recording the audio-signals in real-time. To simulate the smart device, two external speakers and microphones are attached to a smartphone model, the Samsung Galaxy Note 4. The speakers are the standard ones for a Samsung Galaxy S6 and the microphones are simple electret condenser microphones [12]. To interface these with the computer, a soundcard and an amplifier are needed. The soundcard is a Roland Octa-capture [13] and the amplifier is an RDL RU-PA518 [14]. This described setup, without the computer, is depicted in Figure 8.



Fig. 8: The used hardware setup.

The results of the gesture recognition using some of the features and the position estimation according to the least-squares solution are discussed next.

A. Gesture recognition

In order to illustrate the extracted features are sufficient to distinguish the different gestures with an acceptable accuracy, a simple decision tree gesture recognizer is built and its performance is assessed on a validation set. This tree only makes use of a selected set of features, but the recognition results are already decent. Table III represents the confusion matrix of which gestures are recognized if a certain gesture is performed. These tests have been carried out by the author himself and each gesture has been performed thirty times. As can be found in the table, the accuracy lies between 87 and 100 %, even when making use of a simple decision tree to recognize the gestures.

B. Position estimation

Unlike the gesture recognizer, the position estimation is less accurate in estimating the exact position of the hand. This is due to the inaccurate distance measurements and the

Performed	Recognized							
	DT @ bottom	DT @ top	BT sweep	TB sweep	LR sweep	RL sweep	UD	DU
Double Tap @ bottom	90 %	-	10 %	-	-	-	-	-
Double Tap @ top	-	100 %	-	-	-	-	-	-
Bottom - Top sweep	-	-	100 %	-	-	-	-	-
Top - Bottom sweep	-	-	-	90 %	-	7 %	-	3 %
Left - Right sweep	-	-	3 %	-	97 %	-	-	-
Right - Left sweep	-	-	-	3 %	7 %	90 %	-	-
Up - Down	-	-	13 %	-	-	-	87 %	-
Down - Up	-	-	10 %	3 %	-	-	-	87 %

TABLE III: Performed gestures - confusion matrix

Real	Estimated							
	TL HIGH	TL LOW	BL HIGH	BL LOW	TR HIGH	TR LOW	BR HIGH	BR LOW
Top Left HIGH	87 %	3 %	7 %	3 %	-	-	-	-
Top Left LOW	10 %	87 %	-	3 %	-	-	-	-
Bottom Left HIGH	3 %	-	90 %	7 %	-	-	-	-
Bottom Left LOW	-	-	10 %	90 %	-	-	-	-
Top Right HIGH	-	-	-	-	77 %	10 %	13 %	-
Top Right LOW	-	-	-	-	10 %	77 %	-	13 %
Bottom Right HIGH	-	-	-	-	-	-	90 %	10 %
Bottom Right LOW	-	-	-	-	-	3 %	3 %	93 %

TABLE IV: Position estimation - confusion matrix

noise sensitivity of the least-squares estimation. To overcome this limitation of the used minimal setup of two speakers and two microphones, the space above the device is divided into a number of regions. The algorithm then detects in which region the hand is present instead of estimating an exact position. This approach makes sense because the hand is not a point in space but occupies a certain region of a finite size itself. The number of regions is experimentally set to eight, after establishing the accuracy was acceptable for this amount. A grid with fewer regions would yield a higher accuracy, but would limit the applications for the system. In a grid with more regions, the hand would always be present in multiple ones. Thus, the space above the device is divided into four 2D regions and two height regions. The results of the position estimation in these eight regions can be found in Table IV, with accuracies for each region varying between 77 and 93 %.

C. Comparison with state-of-the-art

The results of the presented gesture recognition algorithm are comparable with the method discussed in "SoundWave" [6], while the method presented in this work is able to distinguish a much larger gesture variety thanks to the extra sensors and associated features. The classification results of "Dolphin" [5] are better than the ones obtained in this paper, because they reported their results after a thorough investigation which machine learning method gave the best results on their data, which is left for future work in this paper. A major benefit of

the approach discussed in this paper with respect to "Dolphin" however is that this approach allows to perform both hand tracking and gesture recognition simultaneously.

The tracking accuracy of FingerIO is much higher than the accuracy of the method in this paper, but FingerIO only estimates a position in 2D, while this paper offers a 3D-tracking solution. As such, it allows its use in a wider variety of applications, such as extensive touchless interaction smart devices or operating the multimedia system in cars.

VI. CONCLUSION

A framework for gesture recognition and position estimation in devices with two loudspeakers and two microphones has been described. The emphasis of the work on gesture recognition lies mostly on an investigation of the extracted features themselves, rather than on the method used to recognize the gestures. The novelty with respect to other implementations lies in the use of two speakers and two microphones that are currently present in most high-end smart devices and adding channel estimation-features in addition to the Doppler-features. In the end, a performed gesture could be recognized with an accuracy between 87 and 100 % and the hand position could be estimated with an accuracy above 77 % in all of the eight regions above the device.

REFERENCES

- [1] Richard J. Przybyla, David A. Horsley, Hao-Yen Tang, Bernhard E. Boser, and Stefon E. Shelton. 3D Ultrasonic Gesture Recognition. *IEEE International Solid-State Circuits Conference*, February 2014.
- [2] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. *Proceedings of ACM CHI*, May 2016.
- [3] Siddharth S. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, November 2012.
- [4] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. Bringing Gesture Recognition to All Devices. In *11th USENIX Symposium on Networked Systems Design and Implementation*, pages 303–316, Seattle, WA, April 2014. USENIX Association.
- [5] Yang Qifan, Tang Hao, Zhao Xuebing, Li Yin, and Zhang Sanfeng. Dolphin: Ultrasonic-based Gesture Recognition on Smartphone Platform. *IEEE 17th International Conference on Computational Science and Engineering*, December 2014.
- [6] Sidhant Gupta, Dan Morris, Shwetak N. Patel, and Desney Tan. Soundwave: Using the doppler effect to sense gestures. *Proceedings of ACM CHI*, May 2012.
- [7] Kaustubh Kalgaonkar and Bhiksha Raj. One-handed Gesture Recognition Using Ultrasonic Doppler Sonar. *IEEE Conference on Acoustics, Speech and Signal Processing*, April 2009.
- [8] Martin Holters, Tobias Corbach, and Udo Zölzer. Impulse Response Measurement Techniques And Their Applicability In The Real World. *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, September 2009.
- [9] Guy-Bart Stan, Jean-Jacques Embrechts, and Dominique Archambeau. Comparison of different impulse response measurement techniques. *Journal of the Audio Engineering Society*, 50(4):249–262, April 2002.
- [10] Ole Herman Bjor. Maximum length sequence. Technical report, Norsonic AS, 1995-2000.
- [11] Robert Humphrey. Playrec. URL: <http://www.playrec.co.uk/>, last checked on 2016-07-03.
- [12] Knowles. Electret Condenser Microphone MB3015ASC-1. URL: <http://datasheet.octopart.com/SPM0102NE3-2-Knowles-Acoustics-datasheet-21203029.pdf>, last checked on 2016-11-05.
- [13] Roland. Octa-capture. URL: <http://www.roland.com/products/octa-capture/>, last checked on 2016-11-05.
- [14] RDL. RU-PA518 Audio Amplifier. URL: <http://www.rdlnet.com/downloads/datasheets/ru-pa518.pdf>, last checked on 2016-11-05.