Multiwave: Doppler Effect Based Gesture Recognition in Multiple Dimensions

Corey Pittman
University of Central Florida
Orlando, FL 32816, USA
cpittman@knights.ucf.edu

Conner Brooks
University of Central Florida
Orlando, FL 32816, USA
cbrooks@cs.ucf.edu

Pamela Wisniewski
University of Central Florida
Orlando, FL 32816, USA
pamwis@ucf.edu

Joseph J. LaViola Jr.
University of Central Florida
Orlando, FL 32816, USA
jjl@cs.ucf.edu

CHI'16 Extended Abstracts, May 07–May 12, 2016, San Jose, CA, USA.
ACM 978-1-4503-4082-3/16/05.
http://dx.doi.org/10.1145/2851581.2892286

Abstract
We constructed an acoustic, gesture-based recognition system called Multiwave, which leverages the Doppler Effect to translate multidimensional movements into user interface commands. Our system only requires the use of two speakers and a microphone to be operational. Since these components are already built in to most end user systems, our design makes gesture-based input more accessible to a wider range of end users. By generating a known high frequency tone from multiple speakers and detecting movement using changes in the sound waves, we are able to calculate a Euclidean representation of hand velocity that is then used for more natural gesture recognition and thus, more meaningful interaction mappings.

We present the results of a user study of Multiwave to evaluate recognition rates for different gestures and report accuracy rates comparable to or better than the current state of the art. We also report subjective user feedback and some lessons learned from our system that provide additional insight for future applications of multidimensional gesture recognition.

Author Keywords
Doppler Effect; 3D Interaction; Gesture Recognition; User Studies;

ACM Classification Keywords
H.5.m. [Information Interfaces and Presentation (e.g. HCI)]: Miscellaneous

General Terms
Design, Experimentation

Introduction
Gesture-based interfaces are beginning to see widespread adoption in consumer electronics, for example, smartphones like Amazon's Fire Phone [1] and wearable devices, such as the Moto 360 [2]. A common technique for gesture capture is to collect data using camera-based tracking technologies. Common limitations with such systems include hardware requirements (i.e., the camera), visual occlusion, high processing power requirements, and security and privacy issues inherent in image capture technologies [15]. Some inroads have been made by utilizing different input mediums, such as depth sensing, electromagnetic, and inertial sensing mechanisms [6]. Yet, capturing user gestures using these techniques often requires additional sensing devices.

Some recent developments have been made in recognizing gestures using other ubiquitous devices like smartphone motion sensors, speakers, microphones and even Wi-Fi signals; all of which are already present in a majority of people's homes [4, 8, 9, 12, 13, 16, 18]. These techniques aim to to free the user from the instrumented tracking that current commercial devices require and reduce the need for dedicated sensors. Our work builds upon one such system called Soundwave, which allows for interaction with a laptop or desktop using only a microphone and speaker [10].

Because Soundwave was implemented with only one speaker, it was limited in the type of gestures users could perform. For instance, gestures were constrained to movements toward or away from the single speaker; Soundwave could not detect motions from side-to-side. In this way, the system was innovative but was limited in scope due to simple gestures that often had no contextual meaning in their proposed applications. To illustrate, Soundwave used slow and fast tapping movements to control left and right movement respectively in a game of Tetris. A more optimal mapping in this case would have been to map side-to-side movements of the hand to the analogous movement in Tetris.

In this paper, we present Multiwave, a system for detecting gestures using uninstrumented acoustic sensing based on the approach used in the Soundwave system. To deal with the shortcomings of Soundwave, we generalize the proposed detection algorithms and apply them to multiple speakers to increase the possible complexity of gestures and allow for richer interaction. We discuss the necessary calculations to generate a Euclidean representation of the motion information given the positions of the speakers. We carried out a user study to determine gesture recognition rates, gain insight on user opinions of acoustic gestures, and discuss appropriate applications and shortcomings with acoustic gestures.

Related Work
The design of Multiwave draws from several areas of related literature, including capturing object velocity from sound, converting velocity data into positional data, and using velocity information in place of positional data for gesture recognition using machine learning algorithms. Here, we will describe the related work that was drawn upon for designing Multiwave's theory of operation. First and foremost, our work builds upon Soundwave, which uses the Doppler Effect to detect a set of simple gestures using ubiquitous devices like integrated microphones and speakers [10]. Soundwave illustrates how it is possible to detect the

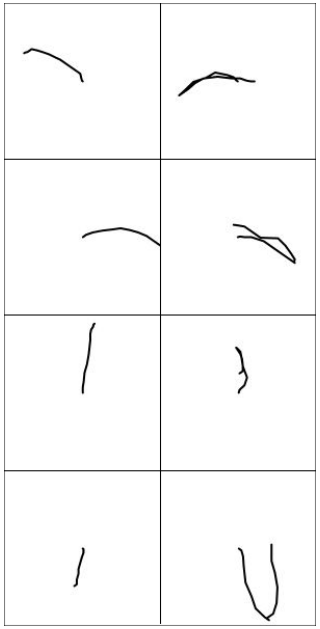

Figure 1: Sample stroke representations of the eight gestures. Left column shows swipes, right shows taps. From top to bottom: left, right, forward, back.

shift in the frequency of a known pilot tone emitted from a speaker using a Fourier transform. Motion is detected using the sign and magnitude of the change in bandwidth over a given period of time. The direction of motion can be determined relative to the speaker source by looking at the direction of the bandwidth change from the expected frequency center. The magnitude of the shift gives information about the inertia of the object moving within the environment: slow movements of large objects look similar to smaller, rapid objects but sustain a shift for a longer period of time. Utilizing these assumptions, Soundwave is able to correctly classify a set of five one-dimensional gestures at about 92 percent accuracy. Soundwave has been shown to be robust to large amounts of ambient noise and different speaker orientations. Multiwave extends Soundwave by using multiple speakers instead of just a single speaker allowing for the detection of more natural gestures which can be applied to a more diverse set of applications.

Converting accelerometer data into strokes has been done previously to assist in gesture recognition. The PhonePoint Pen generated strokes by taking the double integral of the accelerometer information collected from a smartphone integrated sensors at every time step to find displacements [3]. The generated stroke is passed to a classifier to determine what English character was drawn in space. The data can be passed directly to a recognizer without being converted into positional information [7]. Our system uses some of the machine learning features presented in these papers to assist in gesture recognition, such as utilizing velocity information as position data for recognition.

Theory of Operation

Multiwave uses multiple speakers in a known configuration for gesture recognition. Therefore, the speakers positions must be determined during a one time calibration using triangulation. To generate usable data from an arbitrary configuration, each calibrated speaker emits a unique frequency from which changes in bandwidth of the tones can be extracted, as in Soundwave [10]. This data is transformed into Euclidean space and is used to generate a path representation over time. The path represents inferred motion which is then passed through a classifier to determine what gesture was executed. Multiwave can augment interaction on devices without existing gesture support.

We tested a number of frequencies and found that the minimum spacing should be around 500 Hz. Lower values ($\approx$ 400 Hz) performed adequately but occasionally experienced interference from other tones. We selected 18 kHz as our minimum frequency, with each additional tone coming in 500 Hz increments, to avoid possible discomfort due to sensitivity to high frequency tones.

Speaker Calibration

The change in bandwidth detailed in Soundwave represents a velocity change that can be utilized directly for simple gesture recognition; however, further information about the speaker layout must be known when we want to combine data from multiple speakers. In order to create a meaningful representation of the velocities, we determine the position of each of the speakers relative to the microphone to create a known geometry. With two speakers, we can assume that the speakers will be on the same plane and are somewhere between 50 and 60 degrees away from the baseline formed by the user and the center of the monitor, as this is the standard layout on end user laptops and most desktop setups. Because we use machine learning algorithms, Multiwave is robust to changes in configuration.

In cases where we want to use more than two speakers, we can utilize time difference of arrival (TDOA) to find the angles in spherical coordinates [14]. To perform calibration

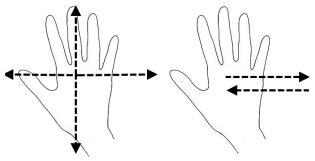

Figure 2: Illustration of expected motion for gestures. From left to right: swipe (forward, back, left, right), tap (forward, back, left, right

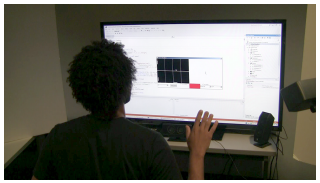

Figure 3: The experimental setup.

with more than two speakers, we can use two microphones to determine the angle between the baseline formed by the microphones and the direction of the speaker emitting a known sound wave by looking at the phase shift of the sound between the two microphones using crosscorrelation. To find the two angles to represent the speaker, the calculation is carried out with the microphones side-by-side (θ) and stacked one over the other (ρ). This process can be repeated for an arbitrary number of speakers, as the calibration process for each speaker is independent.

Generating Euclidean Data

Dealing with the raw data is difficult because each speaker provides data along an arbitrary axis. It is difficult to generate heuristics from this data. Given a known speaker configuration, we can then generate a meaningful representation of motion within the environment from a number of speakers. Each speaker generates a scalar value given by the change in bandwidth of its corresponding frequency. This scalar is the magnitude of a vector in spherical coordinates. Because we know the angles of the speakers, we convert to Euclidean space using the following equations:

$$x = \sum_{i=1}^{n} v_i \sin\theta_i \cos\rho_i \tag{1a}$$

$$y = \sum_{i=1}^{n} v_i \cos\theta_i \cos\rho_i \tag{1b}$$

$$z = \sum_{i=1}^{n} v_i \sin\rho_i \tag{1c}$$

$$\mathbf{V} = [x, y, z] \tag{1d}$$

where i represents the speaker index, n is the number of speakers, v_i represents the inferred velocity from bandwidth shift for speaker i, and θ_i and ρ_i are the speaker angles relative to the baseline from the previous section.

Essentially, we have created a generalized abstraction of an n speaker array that allows for any number of speakers to be used in Multiwave. Using $\mathbf{V}$ over time, we can generate a 3D path for gesture recognition. Because we have no positional information about the beginning of the movement, we center all movements at the origin and append the current $\mathbf{V}$ to the end of the path at each step. We leverage the generated path information for feature extraction.

In this paper, we focus on the two speaker example, for which we can ignore the Z axis. This simplifies the computation by reducing the number of features and leaves us with a two dimensional representation of motion. The resultant path can be visualized as a stroke on a canvas. Figure 1 illustrates the appearance of strokes after being extracted from detected motion.

Gesture Detection

Multiwave allows for two types of single hand gestures, swipes and taps, in four directions: left, right, towards the monitor, and away from the monitor, as seen in Figure 2. A swipe moves in the specified direction and stays there, similar to swiping on a touch screen. A tap quickly moves in a direction and then in the opposite direction as it returns to its resting position. Given the number of gestures used for input, we had to find a way to accurately classify which gesture users performed so that we could map it to a particular action. We chose to use machine learning algorithms for gesture recognition as we have a fairly large number of prospective gestures which we will be classifying.

Features to be used with the recognizer were selected from previous work in sketch recognition [5] and gesture recognition [7]. We calculated the features on both the set of individual vectors and the generated stroke data giving 39 features. The features extracted from the detected gestures passed through a Random Forests classifier generated us-

Post Study Questions	
Q1	The sound-based gesture system was fun to use.
Q2	The system accurately recognized the gestures I was making.
Q3	I liked using this system.
Q4	I felt tired using this system.
Q5	The sound the speakers was making irritated me.
Q6	I would recommend this system to a friend to use.

Table 1: Survey questions asked after the study.

Gesture	Swipe	Tap
Left	91.0%	92.0%
Right	100.0%	96.0%
Toward	98.0%	95.0%
Away	99.0%	80.0%
Combined	97.0%	90.8%
Overall	93.8%	

Table 2: Average % of swipe and tap gestures correctly recognized.

ing WEKA [11]. To segment gestures, we looked at periods of motion that stopped for 200 ms. To prevent any meaningless input, we set a minimum gesture duration at 150 ms. We assumed that the user's hand started near their body in a resting position when beginning a gesture.

User Evaluation

We developed a functional implementation of Multiwave for evaluation purposes.[1] The experimental setup, as seen in Figure 3 consisted of a 55 inch HDTV, a stereo speaker system, and a low cost USB microphone with no processing enabled, each of which was connected to a PC with a Intel Xeon dual core processor and 12 GB of RAM. Multiwave was implemented in C# using the NAudio .NET audio library. The microphone was placed on a tripod in front of the user. Participants were asked to sit behind the microphone. We carried out a user study to evaluate our proof-of-concept from the perspective of our participants. Our goal was to determine the accuracy of Multiwave and get user feedback.

Procedure

Participants first provided training data by performing ten samples of each Multiwave input gesture. This training data was collected and a Random Forest classifier was generated using WEKA. After training the system, users were then asked to perform each gesture ten more times. This data was used to calculate the accuracy rate of our gesture recognizer. A post study survey was administered to our participants to gather information about their opinions about the system. We asked users to rate their responses to the questions in Table 1 on a Likert scale of 1 = Strongly Disagree to 7 = Strongly Agree. Users were also encouraged to leave comments about their experience.

[1] Source code available at https://github.com/ISUE/MultiWave/tree/Simplified

Results

Ten students (9 male, 1 female) were recruited from a local university to participate in the study. Ages ranged from 19 to 27 with a median age of 21. Of all the participants, five had previous experience with body tracking of some sort. The duration of the study ranged from 45 to 60 minutes. The overall accuracy for swipe and tap gestures in each configuration is shown in Table 2. The swipe gestures were slightly more reliably recognized by the gesture recognizer. The tap gesture accuracy was on par with the quick tap and slow tap accuracy showed in Soundwave.

Users found the system to be fun to use (M = 6.2, SD = 1.14) and liked the experience of using it (M = 5.9, SD = 1.25). We also analyzed the open ended survey questions to gain more insight into opinions about the system in general. Five of the ten participants liked the idea of leveraging existing devices to support gesture recognition. Two participants mentioned enjoyed the gestures that were supported. Another two found the interactions fluid. Yet, three found the experiment to be tiresome, likely due to the repetitive nature of the gestures we asked them to perform. One stated that the high pitched sound was irritating.

Discussion

Our user study of Multiwave showed that the accuracy of our system was comparable to Soundwave [10], the nearest predecessor of Multiwave. The implication of this is that our abstraction of acoustic gesture recognition maintains the responsiveness of Soundwave while allowing for more expressive user interactions. The main limitation of Soundwave was that gestures were one-dimensional due to the use of a single speaker. Because of this, gestures did not map well to most applications in a meaningful way. By adding this second dimension for interaction, Multiwave can easily map motions from an environment to a number

of end user applications. Further, the ability to do this with only a microphone and two speakers minimizes the barriers of use that are often a limitation of other sensor based gesture recognition systems.

In our implementation of Multiwave, the volume of the speakers controlled how sensitive the system was to movement in the environment. Smaller motions were more difficult to sense if the system volume was set too low. For future implementations of Multiwave, we plan to have a brief user calibration period where the proper volume is dynamically detected. Yet, there is a clear trade-off in overall accuracy of the system when the volume is increased by too much, as spurious inputs become a problem. Even seating position adjustments can be picked up as significant movements if the volume is too high. Further, some users found the high pitched sound to be annoying.

Also, we occasionally found that segmentation errors, mistakes in determining what is and is not an intentional gesture, caused problems with the overall experience when participants executed a tap gesture, which included a sharp turn. Our clutching solution was designed to ignore all motion following a detected gesture for a set period of time (250 ms), which functioned well to prevent false positives. However, some users wanted the ability to continue swiping without returning to their resting position. For future implementations of Multiwave, we plan to add an enable gesture, like a finger snap, which will activate detection for a short period of time.

Future Work

Now that we have proven that acoustic gesture recognition can be used to detect multidimensional input through the use of additional speakers, we are interested in further extending Multiwave in other ways. For instance, detecting depth of motion using a surround sound system may be a promising way to translate gesture input into 3D virtual environments. In fact, we have done some initial testing with such a system. Our preliminary findings have showed good performance with simple swiping gestures, but tapping accuracy has degraded. We are in the process of improving gesture recognition accuracy rates prior to conducting a full fledged user study of the new system.

Another extension that we have done some preliminary work on is exploring complex gestures based on geometric shapes like circles, squares, and X's. We tried two methods for recognition: one being to simply add them into the machine learning algorithm as additional classifications and the other being to use taps and swipes as primitives to build the shapes. Our findings in both cases were that recognition rates of complex gestures were not sufficient for actual use, with pilot studies showing accuracies of less than 60% in two-dimensional configurations. We plan to explore non-parametric gesture recognizers like template matching as a possible avenue to alleviate some of the errors that were occurring with complex gesture recognition [17].

Conclusion

We presented Multiwave, a system which extends Soundwave to multiple dimensions to allow for better mapping of hand gestures to applications. We showed a method of transforming extracted motion information into Euclidean space to generalize gestures for any given speaker geometry. We documented the selection process of the recognition algorithms used in Multiwave. We ran a user study to determine the accuracy of the system. Our results show that Multiwave is as accurate in two dimensions as Soundwave was in a single dimension, giving Multiwave the advantage of allowing for intuitive mappings into a growing number of applications that accept gesture-based input.

References

[1] *Amazon Fire Phone*. http://www.amazon.com/Fire-Phone/

[2] *Motorola 360 Smartwatch*. https://moto360.motorola.com/

[3] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. 2011. Using mobile phones to write in air. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 15–28.

[4] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. 2013. Doplink: Using the doppler effect for multi-device interaction. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 583–586.

[5] Rachel Blagojevic, Samuel Hsiao-Heng Chang, and Beryl Plimmer. 2010. The power of automatic feature selection: Rubine on steroids. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*. Eurographics Association, 79–86.

[6] Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. 2004. *3D user interfaces: theory and practice*. Addison-Wesley.

[7] Salman Cheema, Michael Hoffman, and Joseph J LaViola Jr. 2013. 3D gesture classification with linear acceleration and angular velocity sensing devices for video games. *Entertainment Computing* 4, 1 (2013), 11–24.

[8] Gabe Cohn, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Humantenna: using the body as an antenna for real-time whole-body interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1901–1910.

[9] Gabe Cohn, Daniel Morris, Shwetak N Patel, and Desney S Tan. 2011. Your noise is my command: sensing gestures using the body as an antenna. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 791–800.

[10] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Soundwave: using the doppler effect to sense gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1911–1914.

[11] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.

[12] Kaustubh Kalgaonkar and Bhiksha Raj. 2009. One-handed gesture recognition using ultrasonic Doppler sonar. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 1889–1892.

[13] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing gesture recognition to all devices. In *Usenix NSDI*, Vol. 14.

[14] Charles Knapp and G Clifford Carter. 1976. The generalized correlation method for estimation of time delay. *Acoustics, Speech and Signal Processing, IEEE Transactions on* 24, 4 (1976), 320–327.

[15] Joseph J LaViola Jr. 2014. An introduction to 3D gestural interfaces. In *ACM SIGGRAPH 2014 Courses*. ACM, 25.

[16] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 27–38.

[17] Eugene M. Taranta, II and Joseph J. LaViola, Jr. 2015. Penny Pincher: A Blazing Fast, Highly Accurate $-family Recognizer. In *Proceedings of the 41st Graphics Interface Conference (GI '15)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 195–202. http://dl.acm.org/citation.cfm?id=2788890.2788925

[18] Chen Zhao, Ke-Yu Chen, Md Tanvir Islam Aumi, Shwetak Patel, and Matthew S Reynolds. 2014. SideSwipe: detecting in-air gestures around mobile devices using actual GSM signal. (2014).