Multiwave: Complex Hand Gesture Recognition Using the Doppler Effect

Corey R. Pittman[*]
University of Central Florida

Joseph J. LaViola Jr.[†]
University of Central Florida

Abstract

We built 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 a speaker and microphone to be operational, but can be augmented with more speakers. Since these components are already included in most end user systems, our design makes gesture-based input more accessible to a wider range of end users. We are able to detect complex gestures by generating a known high frequency tone from multiple speakers and detecting movement using changes in the sound waves.

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

Index Terms: H.5.2. [User Interfaces]: Input devices and strategies (e.g., mouse, touchscreen)—Gesture;

1 Introduction

Until recently, gesture based interfaces have been limited to niche sectors, specifically motion capture, entertainment, research, and gaming. Gesture based interfaces are beginning to see widespread adoption in consumer electronics, including smartphones like Amazon's Fire Phone [1] and wearable devices like the Moto 360 smartwatch [3] and the Apple Watch [2]. The relatively low number of people adopting gesture interfaces can be attributed to shortcomings with the most common input device, the RGB camera. Common problems with camera based tracking devices include occlusion, significant processing power requirements, and security issues in corporate or industrial environments. Consumers have also complained about issues with electronics that contain cameras that could compromise their privacy. Another significant barrier is the hardware requirement for gesture interfaces. Capturing user gestures with a large portion of today's interfaces requires the purchase of additional sensing devices. These devices utilize different input mediums; optical, depth sensing, electromagnetic, and inertial being among the most common [7].

Uninstrumented tracking using ubiquitous devices can serve as a low-cost approach to enabling gesture interaction with no additional hardware requirements in most devices. Using smartphones, speakers, microphones, and even Wi-Fi signals, it is possible to enable gestures on commodity devices [6, 9, 10, 14, 17, 20, 28]. Two systems for detecting gestures from only acoustic signals and the Doppler effect, Soundwave [11] and Audiogest [21], have been presented as possible alternatives to traditional camera-based recognition. However, both systems leverage rule-based heuristics for gesture recognition, and it is known that such recognizers do not scale beyond a small handful of gestures [16]. An alternative method of recognition must be used to detect larger sets of gestures in more complex applications. Additionally, neither system addresses how to bring the range of detectable gestures closer to those which are possible with camera-based approaches.

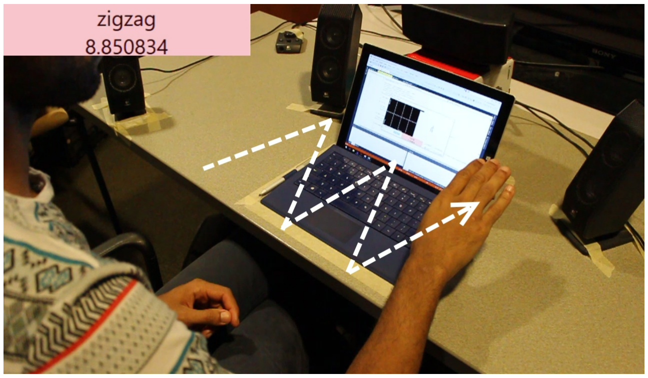


Figure 1: Multiwave is a Doppler Effect based sensing system that can detect multidimensional gestures.

To enable robust gesture recognition using acoustic systems, we develop a method of sound data extraction and representation that is amenable to time-series pattern recognition and can be extended to an arbitrary number of speakers. We implement these techniques in a system called Multiwave and demonstrate that our approach is able to support a large, 14 class, gesture vocabulary with high accuracy, using only a few training samples. We carry out a user study to determine system recognition rates and discuss appropriate applications and shortcomings with acoustic gestures. Based on the results of our user study and Multiwave's support of expressive and complex gestures, we can conclude that acoustic-based gesture recognition is a viable alternative to camera-based gesture recognition for human-computer interaction.

2 Related Work

The design of Multiwave draws from several areas of related literature, including capturing object velocity from sound, manipulating acoustic data via filtering, and gesture recognition using machine learning algorithms. An important precursor to our work is Soundwave, which uses the Doppler effect to detect a set of simple gestures using ubiquitous devices like integrated microphones and speakers [11]. Soundwave illustrates how it is possible to detect the shift in the frequency of a known pilot tone emitted from a speaker using a Fourier transform. Soundwave is able to correctly classify a set of five one-dimensional gestures at about 92% accuracy. Our work differs from Soundwave in that it uses the all of the FFT data surrounding the expected frequency center to recognize movements instead of just the magnitude and direction of motion. Our system is designed with extensibility in mind and can leverage multiple speakers and a generalized gesture recognizer instead of heuristics.

AudioGest is another system for extracting hand motion from the environment using the Doppler Effect [21]. AudioGest takes a unique approach to gesture recognition by using the spectrogram of the microphone feed to detect gestures in lieu of looking at the history of bandwidth changes. This allows AudioGest to obtain

[*] e-mail:cpittman@knights.ucf.edu
[†] e-mail:jjl@cs.ucf.edu

details about the approximate distance and speed from the sound signal. One of the primary requirements of their system was a clean signal, so obtaining a noise profile of the microphone over 3600 seconds to build an appropriate filter was necessary. The spectrogram approach also allows AudioGest to use a purely heuristic technique for gesture recognition by interpreting geometric properties of the filtered spectrogram. The system detects six hand gestures at different relative speeds and ranges at up to 95% accuracy using a single speaker/microphone pair. Multiwave requires a small number of training samples for gesture recognition, but requires no a priori noise profile of the environment. It also can leverage additional speakers beyond the first to reduce error rates and supports a larger gesture vocabulary.

Airlink uses a similar approach to Soundwave but applies it to a multi-device paradigm [9]. It is able to determine relative positions of gestures between devices and determine if a gesture is moving towards, away from, or back and forth relative to each device. Taking these three detected states and applying them to a line of three smartphones, the authors were able to detect a set of six relative single direction hand movements at an accuracy of about 97%. While it is not aimed specifically at building a large corpus of detectable gestures like Multiwave, Airlink makes assumptions about the duration and speed of the gestures they detect which limits potential gesture complexity.

Kalgaonkar and Raj presented a three dimensional gesture recognition setup using Doppler sonar [14]. The system was constructed using a low cost ultrasonic transmitter and three receivers that were placed in a known environment which allowed for the detection of single handed gestures. The system allowed for the detection of eight hand moving gestures that include six directional swiping movements in three dimensions and two rotational gestures: clockwise and anticlockwise circles. Using a Gaussian Mixture Model in conjunction with a Bayesian classifier, the system was capable of detecting these gestures with accuracy above 88%. Our system differs from this work by repurposing existing components of a laptop, desktop, or home theater system instead of utilizing a purpose built rigging, as well as working with raw frequency domain data to preserve as much incoming data as possible.

Instrumented acoustic gesture recognition has also been a popular research field in recent years. An ultrasound based activity and context recognition algorithm utilizing wearable devices was developed [25]. The receiver for the system was a voice recorder while speakers were placed on the user's wrists and neck. The velocity of each speaker was determined using the Doppler effect and the distance between the speakers and the voice recorder was found based on the amplitude of the emitted frequency for each speaker. The system was aimed at offline recognition, thus allowing the use of a portable voice recorder in lieu of a smart phone or other processing device. While wearable computing is currently an evolving field, adoption remains low and utilizing a large number of distributed emitters could be cumbersome. Similar work that emphasized smartphones for presence detection also exists [12, 27].

Ultrasonic tracking devices are not a new development with many previous applications in virtual reality [26]. Attaching high frequency microphones to smaller scale devices allows for accurate tracking of tools within a given workspace [13]. Almost any set of small speakers and ultrasonic microphones can be utilized for tracking when using a dynamic signal processor. Other VR-centric methods of input have been studied which emphasize commodity devices in lieu of purpose built hardware [4].

In recent literature, gesture recognition has typically favored depth sensors and other time of flight cameras which then utilize proprietary techniques to model a human body or hand skeleton [16]. The joint positions and other visual information about the state of the body are then used to develop gesture recognition algorithms that possess varying levels of intelligence, from simple heuristics to full-fledged large scale machine learning algorithms. We elected to focus on a time-series based template matching algorithm for recognition which was domain agnostic.

3 Theory of Operation

Multiwave can use an arbitrary number of speakers with a single microphone to extract motion information from the environment for gesture recognition. To generate usable data, each speaker emits a unique, inaudible tone from which the changes in the bandwidth of the tone can be extracted. Collecting the data from the frequencies around the expected tones over time allows us to generate a time series of vectors which we then pass into a template-based recognizer to classify gestures. Multiwave can therefore work on any system that already has a microphone and speakers without any additional hardware.

3.1 Frequency Selection

We did not want to interfere with normal usage habits of users with a low frequency tones, so we selected frequencies above 17 kHz, which is above the range of frequencies that most adults can hear [18]. Hardware selection limited the upper bound of the range to 22 kHz, as a speaker operating at 44.1 kHz cannot produce output frequencies above that point. However, most consumer speakers do not have high fidelity above 20kHz. With this in mind, we selected pure tones that covered the range from 17 kHz to 20 kHz spaced 500 Hz apart with a bias towards the lower end of the spectrum to allow the speakers to operate closer to their effective range. When using fewer speakers, the minimum frequency can be moved to 18 kHz to prevent potential user irritation. To generate the inaudible tones, each speaker i was passed a sine waveform:

$$y_i(t) = A_i * sin(2\pi f_i t) \tag{1}$$

where A_i is the amplitude of the wave, f_i is the frequency chosen for the speaker, and t is the current time in the wave. Each speaker emits a unique frequency at a constant amplitude. To visualize the magnitude of the frequencies in the spectrum, a Fourier transform is calculated for the entire sound spectrum.

3.2 Detecting Motion Using Doppler Effect

We leverage the Doppler Effect to detect motion in the environment around the microphone. The Doppler Effect is defined as a shift in the frequency of sound waves due to the movement of an object [15]. The effect is typically visualized using the example of sirens on emergency vehicles. When the vehicle is moving towards a stationary person, the sound steadily becomes higher pitched. As the vehicle passes by the person and begins moving away, the sound is audibly lower pitched. Applied to the stationary speakers and microphones in our system, we can assume that all shifts of the frequency are the product of user movements and reflections from the user's body.

The equation for the Doppler shift is given by:

$$f = \left(\frac{(c + v_r)}{(c + v_s)}\right) * f_0 \tag{2}$$

where f is the observed frequency, f_0 is the emitted frequency, c is the speed of the wave in the medium (air in this case), v_s is the velocity of the source, and v_r is the velocity of the receiver. For this particular application, the source is a stationary speaker and the receiver is a stationary microphone. Therefore, we aim to detect the change in velocity by looking at the generated reflections of some object within the microphone-speaker range, which changes the relative velocities of each thereby generating a different perceived frequency (f). The frequency change is directly proportional to the velocity detected in the environment. An example of this shift is illustrated in Figure 2. The discrete velocity change can be inferred

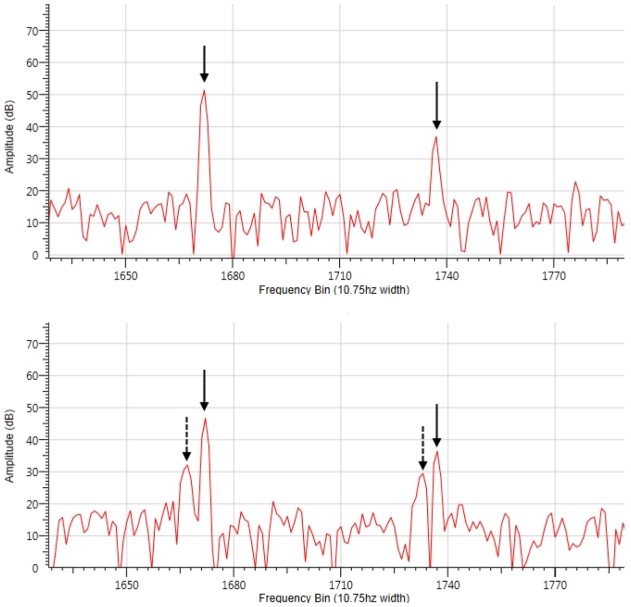


Figure 2: The Fourier transform of the sound wave detected by the microphone in a two speaker environment. The top graph shows the Fourier transform with no movement in the environment. The bottom graph shows movement away from the speakers. Notice the shift in the peak tones denoted by the dotted arrow.

by analyzing the number of frequency bins from the peak that a second peak is found.

3.3 Gesture Detection

In this section, we will detail the process of converting the frequency domain data we have previously extracted into a gesture. A previous prototype of Multiwave attempted to detect gestures using multiple calibrated speakers and a single microphone by transforming the inferred peak bandwidth shifts into a three-dimensional Euclidean representation using triangulation and relative position estimation [19]. The bandwidth shifts were components of velocity vectors which were concatenated over time to form paths which were then passed into a random forest recognizer. The prototype performed well for the set of simple gestures (directional swipes/taps) it was tested on, but when expanded out to a larger number of more complex geometric gestures (circles, squares, characters), the accuracy fell and gesture confusion increased. This was due to the method in which the data was converted to a Euclidean representation and the filtering that was necessary to cull spurious input from the user. In making the data easier to handle for a developer, some of the characteristic signals of the data were eliminated. Based on those findings, a different approach to representing the input data was implemented where instead of using a gesture path in Euclidean space, we would encode the frequency domain data surrounding each peak tone into a high component vector which limits data loss.

3.3.1 Constructing a Sample

We do not want to rely solely on the maximum bandwidth change, as in previous work [11, 19, 21], as it oversimplifies the motion we are detecting. Detecting movement from just the bandwidth change does not take into account the size or speed of the object moving in the environment, and occasionally filters out intentional gestures. Instead, we want to look at a range of frequencies around a given emitted frequency to maximize the amount of information we have to classify. After some empirical analysis, a Hamming window of 4096 was determined to provide us with sufficient fidelity to observe small details in our gestures without too much additional noise. Our region of interest is 250 Hz in either direction around the expected frequency. This range prevents any overlap between our evenly spaced frequencies. The motion relative to each speaker can be described by a 33 bin subregion of the Fourier transform of raw audio input centered at our expected frequency with a radius of 16 10.75 Hz-wide bins. The system is fairly robust to background audio because it only pulls from the FFT data around the emitted frequencies (17 kHz), which is far above the frequencies that most human noise is generated.

Because we are working with raw audio through a microphone, we have a large amount of noise to filter out. We clean up our data by utilizing a filtering algorithm to remove (set to 0) all noise below a decibel threshold that is 30% greater than the mean of the entire spectrum. Then, for each speaker we are using, we scale the data in the corresponding subregion such that each bin is a decimal value between 0 and 1, inclusive. We then concatenate all of the subregions together and treat this data as a $N * 33$-vector, where N is the number of speakers we are using in our configuration.

By repeating this operation over a period of time, we can observe changes in the bandwidth of the expected frequency as well as determine if there are reflections being detected at disjoint frequencies in the spectrum. We maintain a history of the previous vectors and then use this data to represent motion over time. This gives us a time series on with which we can leverage existing gesture recognition methods. We poll the microphone at a frequency of 21 hz to allow for fine gesture detection.

3.3.2 Recognizing a Gesture

With a reliable time-series representation of acoustic gesture data now in hand, we are able to leverage time-series pattern recognition techniques. Specifically, we use the recently introduced DTW-based recognizer, Jackknife [23], which is fed our normalized frequency spectrum time series. In Multiwave, we define a gesture as a time sequence of m dimensional points, where m is the normalized frequency spectrum bin count ($N * 33$ in Multiwave). The time series is then treated as a gesture path through m dimensional space. Given a time series of n points, we resample the time-series to $n = 16$ equidistant points along the path and extract the fifteen m-dimensional direction vectors between the resampled points. We then use the normalized direction vectors to measure the dissimilarity between a given sample and each template in the training dataset using DTW with the dot product as its local cost function. This is similar to a number of nearest neighbor classifiers where a candidate is compared against every template in the training set and the template with the best score is selected as the matching class. Similar techniques are used for handwritten text recognition [5, 22].

In practice, we look at the past 5, 10, 15, 20, and 25 frames and pass each to the recognizer to be classified. Each of the five frame history segments is first resampled to length 16 before carrying out the dynamic time warping function. Higher lengths were tested, but performance did not improve. Templates in the dataset are all the same length to make use of the Sakoe-Chiba band to prevent pathological warping and, in conjunction with the resampled length, improve temporal performance of the DTW operation. When we have a persistent best gesture over at least three frames, we consider that to be the gesture class that best represents the detected motion.

One of the major benefits of using Jackknife is that it requires a low number of templates (less than 3) to match recognition rates of some other template-based methods. We can cut down on the amount of time it takes to train a user dependent recognizer by leveraging this property. False positives were minimized via rejection criteria which uses artificial negative samples to create non-gesture samples to find a threshold which a match score must exceed to be considered a positive recognition result. This rejection component allows us to further speed up the recognition process by serving as an early exit

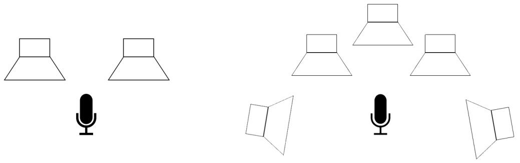


Figure 3: Speaker positions for the two tested configurations. When using only two speakers, integrated laptops speakers are used to ensure ecological validity. Speakers form a semicircle in the 5.1 PC speaker configuration.

case. When using the recognizer in practice, we segment gestures by clearing the time series when a long gesture like star or zigzag is detected. Shorter gestures will only fully register as detected if there is a period of idle movement after their completion. This is required for gestures that are short or define a component of another.

4 User Evaluation

We designed and carried out a user study to evaluate our proof-of-concept from the perspective of our participants. Our goal was to determine the recognition accuracy of Multiwave in two and five speaker scenarios and to obtain user feedback. We also wanted to determine how well gestures which were performed in directions other than perpendicular to the display could be detected using only two speakers.

4.1 Test Setup

For our test system, we focused primarily on two configurations: two speakers and one microphone, which most commodity laptops have integrated into their chassis; and five speakers and one microphone, which most home theater or 5.1 PC speaker setups come with. A diagram of these two setups can be seen in Figure 3.

The experimental setup consisted of a Surface Pro 3 with an external sound card and a 5.1 surround sound computer speaker system. We chose to use the internal microphone of the Surface Pro 3 after empirical testing determined that it presented very little noise. We used the internal speakers of the laptop in the two speaker condition to emulate realistic laptop use cases and the external speakers in the five speaker setup to emulate home theater or desktop audio systems. Multiwave was implemented in C# using the NAudio .NET audio library.[1] Participants were asked to sit in the center of the five speaker condition to simulate the seating position within a home theater setup, albeit scaled down significantly, as shown in Figure 4.

For evaluation purposes, a handful of the parameters of the system are tuned empirically. In addition to the frequency range selected for the system, the volume of the speakers was also important. Too much volume would cause every minute movement in the environment to register reflections in the spectrum around the expected frequency. Too little volume and those users who had small hands would have trouble producing reliable data. These problems tend to be more prominent when a smaller number of speakers are used. The volume of the speakers was set prior to use in our experiments but could be automated by asking the user to execute a test gesture, like a star, and adjusting the amplitude of the emitted tones until the reflected tones exceed a minimum threshold. This, combined with the rejection criteria from Jackknife, is sufficient for rejecting false positives.

4.2 Selected Gestures

We wanted to select a set of gestures to be recognized that were both complex but still the sort of movements people would be comfortable making when sitting in front of a laptop. Movements like directional swipes, taps, and clear two handed motion have been demonstrated to work well enough, but those gestures do not fully represent the complex gesticulation that is possible when seated [11, 21]. We do not want to include sports actions like tennis serves or baseball swings as in older Wiimote gesture papers [8]. Instead, we turned to two productivity metaphors for the sort of movements people make while seated.

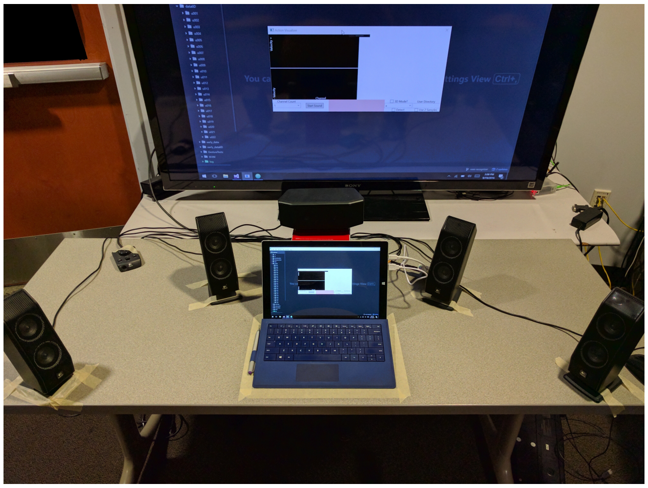


Figure 4: The experimental setup. The microphone is embedded in the top bezel of the laptop. Note that the external speakers are only used for evaluating five speaker setups, otherwise the two integrated speakers on the laptop are used.

	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 were making irritated me.
Q6	I would recommend this system to a friend to use.

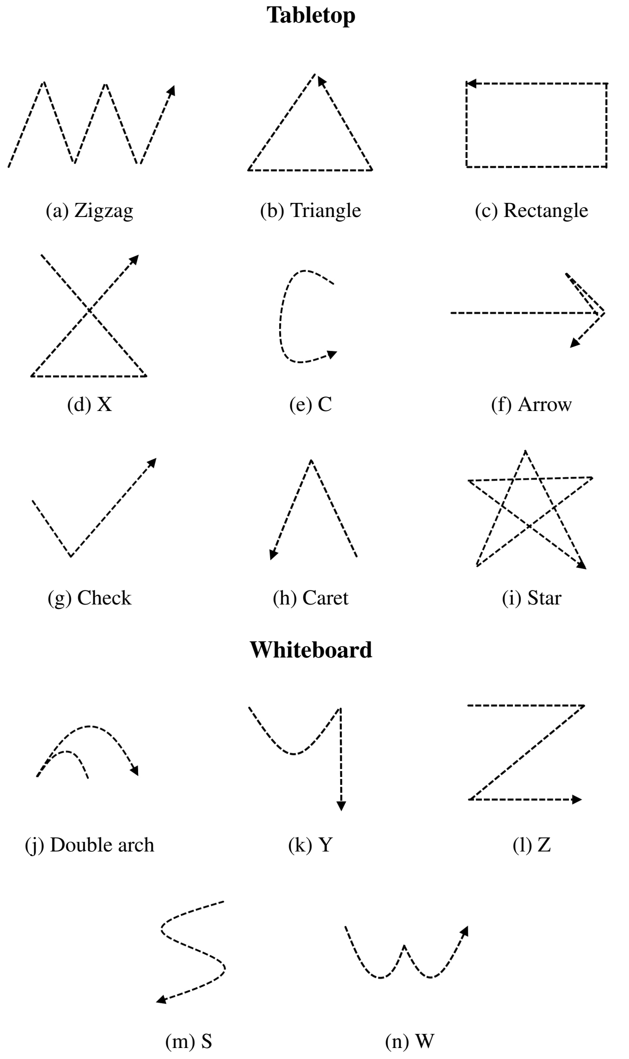
Table 1: Survey questions asked after the study.

The first metaphor is “tabletop.” Imagine moving a mouse on a planar tabletop which is above the keyboard. All gestures which were derived from this metaphor move along this imaginary plane. These movements were designed to move in the same plane as the two speakers and microphone in most modern laptops. The second metaphor is “whiteboard.” Imagine moving an eraser on an imaginary planar whiteboard which is in front of the user. These movements were designed to test out the performance of gestures which featured a vertical component, while still being accessible and not game oriented. Figure 5 details the movement defined by each gesture.

4.3 Subjects

Twenty-two students (17 male, 5 female) were recruited from a local university to participate in the study. Ages ranged from 18 to 30 with a median age of 23.5. Of all the participants, four owned home theater systems and nine had previous experience with body tracking of some sort. Fourteen of the twenty-two students were graduate students. All but three had used some form of commodity gesture device (Microsoft Kinect, Nintendo Wiimote, PS Move Controller). The duration of the study ranged from 20 to 30 minutes.

[1] Full source and notes for this implementation can be found at https://github.com/ISUE/Multiwave



Tabletop

(a) Zigzag (b) Triangle (c) Rectangle

(d) X (e) C (f) Arrow

(g) Check (h) Caret (i) Star

Whiteboard

(j) Double arch (k) Y (l) Z

(m) S (n) W

Figure 5: The gestures selected for evaluation. (a)-(i) follow the tabletop metaphor, meaning they are executed parallel to a table like plane. (j)-(n) follow the whiteboard metaphor, so they are done perpendicular to the tabletop as though on a board in front of the user.

4.4 Procedure

Participants first provided training data by performing five samples of each Multiwave gesture using the two internal laptop speakers. This training data was manually segmented on input by the study proctor. Participants were asked to maintain consistent speed across all gestures, and to perform the gestures deliberately and clearly. After training the system, users were then asked to perform each gesture to determine if the system was able to properly classify new gestures using the templates they had provided. These gestures were automatically segmented using the Multiwave's segmentation component. This process was then repeated for the five external speaker scenario. Participants were asked to evaluate their user dependent recognizer by repeating some of the gestures. A post study survey was administered to our participants to gather comments detailing their opinions about the system. For instance, 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 comment about their experience.

Speaker Configuration	Tabletop	Whiteboard	Overall
Two Internal Speaker	95.8%	94.2%	93.9%
Five External Speaker	93.9%	96.2%	92.6%

Table 2: Average accuracy of different gesture types when cross-validated between only tabletop gestures, only whiteboard gestures, and between all gestures.

Detection Approach	SoundWave [11]	AudioGest [21]	Multiwave
Reported Accuracy	94.5%	95.1%	93.9%
Speakers Supported	1	1	Any
Gesture Complexity	Low	Low	High

Table 3: Comparison of recent sound-based gesture recognition systems. We compare against Multiwave's two speaker results with the full gesture set, as they are the most likely to be utilized.

4.5 Results

After recording the data from the participants, we used user dependent leave-one-out cross-validation to determine the error rate of the system in each configuration. We removed one sample from the user provided training data and then attempted to recognize using the remaining 5x14-1 templates. Results of this test are displayed in Table 2. The observed user dependent accuracy hovers right around the area of the previous state of the art [11,21], but with more than twice as many gestures of significantly higher complexity. Note that prior work only reports user-independent accuracy. Confusion matrices of the two user dependent tests can be seen in Table 4.

Participants found the system to be fun to use (M = 6.2, SD = 1.07) and liked the experience of using it (M = 6.1, SD = 1.08). The system was perceived as mostly accurate (M = 5.05, SD = 1.22) and participants felt that they would recommend using it to as friend (M = 5.41, SD = 1.30). Participants did not feel strongly either way about how tiring the system was to use (M = 3.68, SD = 1.68). Overall, the sound generated was not considered irritating to participants (M = 1.32, SD = 0.92). We also analyzed the open ended survey questions to gain more insight into opinions about the system in general. Eight of the twenty-two participants liked the idea of leveraging existing devices to support gesture recognition and the simplicity that it provided. Six reiterated that the system was fun to use. Three participants liked that no contact was required. Yet, five found the experiment to be tiresome, likely due to the repetitive nature of the gestures we asked them to perform. No participants commented on the sound emitted from the speakers, mostly due to the selected frequencies.

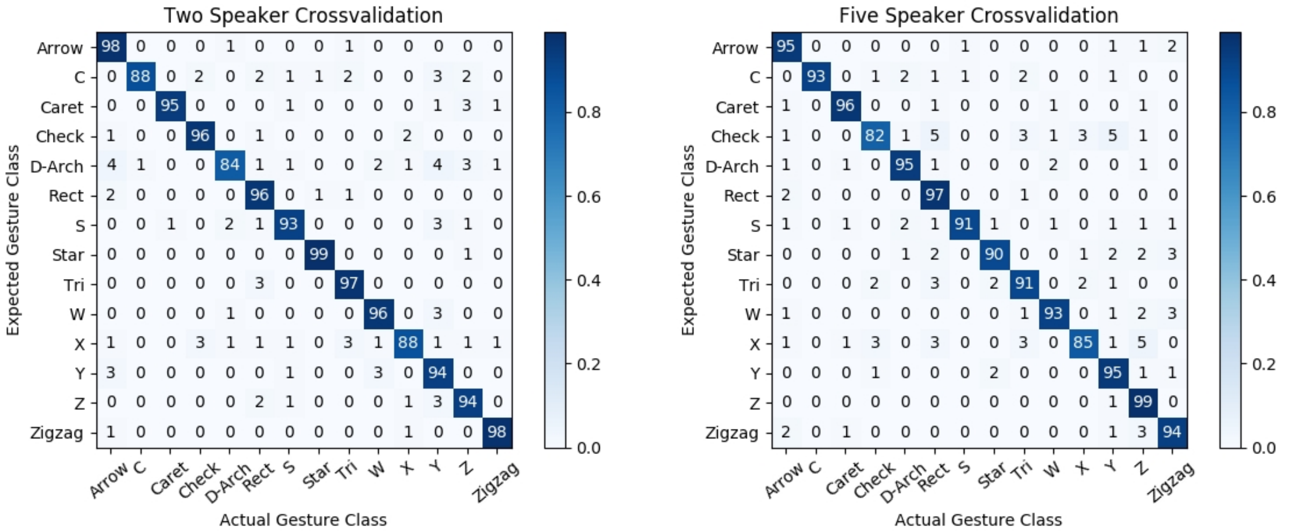
5 Discussion

5.1 Research Contributions

The main contribution of this work was presenting a technique to enable gesture recognition on existing devices with speakers and microphones that relies solely on the raw FFT data changes over time. We are able to classify natural, complex hand movements as user gestures. Additionally, we were able to successfully instantiate our theory of operation into a working proof-of-concept that used only one microphone and two speakers. Multiwave can be deployed on any existing hardware that has a microphone and speakers.

5.2 Comparison to Existing Approaches

Our evaluation of Multiwave showed that the user dependent accuracy of our system was comparable to the state of the art, as shown in Table 3, but featured significantly more complex gestures. This implies that our proposed method of gesture recognition maintains the precision of existing techniques while allowing for more meaningful



Two Speaker Crossvalidation

Expected Gesture Class \ Actual Gesture Class	Arrow	C	Caret	Check	D-Arch	Rect	S	Star	Tri	W	X	Y	Z	Zigzag
Arrow	98	0	0	0	1	0	0	0	1	0	0	0	0	0
C	0	88	0	2	0	2	1	1	2	0	0	3	2	0
Caret	0	0	95	0	0	0	1	0	0	0	0	1	3	1
Check	1	0	0	96	0	1	0	0	0	0	2	0	0	0
D-Arch	4	1	0	0	84	1	1	0	0	2	1	4	3	1
Rect	2	0	0	0	0	96	0	1	1	0	0	0	0	0
S	0	0	1	0	2	1	93	0	0	0	0	3	1	0
Star	0	0	0	0	0	0	0	99	0	0	0	0	1	0
Tri	0	0	0	0	0	3	0	0	97	0	0	0	0	0
W	0	0	0	0	1	0	0	0	0	96	0	3	0	0
X	1	0	0	3	1	1	1	0	3	1	88	1	1	1
Y	3	0	0	0	0	0	1	0	0	3	0	94	0	0
Z	0	0	0	0	0	2	1	0	0	0	1	3	94	0
Zigzag	1	0	0	0	0	0	0	0	0	0	1	0	0	98

Five Speaker Crossvalidation

Expected Gesture Class \ Actual Gesture Class	Arrow	C	Caret	Check	D-Arch	Rect	S	Star	Tri	W	X	Y	Z	Zigzag
Arrow	95	0	0	0	0	0	1	0	0	0	0	1	1	2
C	0	93	0	1	2	1	1	0	2	0	0	1	0	0
Caret	1	0	96	0	0	1	0	0	0	1	0	0	1	0
Check	1	0	0	82	1	5	0	0	3	1	3	5	1	0
D-Arch	1	0	1	0	95	1	0	0	0	2	0	0	1	0
Rect	2	0	0	0	0	97	0	0	1	0	0	0	0	0
S	1	0	1	0	2	1	91	1	0	1	0	1	1	1
Star	0	0	0	0	1	2	0	90	0	0	1	2	2	3
Tri	0	0	0	2	0	3	0	2	91	0	2	1	0	0
W	1	0	0	0	0	0	0	0	1	93	0	1	2	3
X	1	0	1	3	0	3	0	0	3	0	85	1	5	0
Y	0	0	0	1	0	0	0	2	0	0	0	95	1	1
Z	0	0	0	0	0	0	0	0	0	0	0	1	99	0
Zigzag	2	0	1	0	0	0	0	0	0	0	0	1	3	94

Table 4: Confusion matrices for the two speaker and five speaker data collected during the user study. Y-axis is expected gesture, x-axis is actual recognized gesture. Results are user dependent. There is very little structured confusion between gestures. C, Double Arch, and X have the most confusion in our two speaker setup. Check and X have the lowest accuracy in the five speaker configuration.

diverse and complex interaction. A limitation of the previous systems was that gestures were simple one-directional movements. Left and right movement were not detected due to over-simplification of the input data. Because of this, gestures did not map well to most applications in a meaningful way. By utilizing the entire spectrum, Multiwave can more directly map motions from an environment to a number of end user applications. Further, the ability to do this with only a microphone and speakers minimizes the barriers of use that are often a limitation of other sensor based gesture recognition systems.

5.3 Limitations and Lessons Learned

Our evaluation did not emphasize testing every possible number of speakers or speaker arrangement. We selected commonly used speaker counts in what would be considered standard positions relative to the user. Multiwave as implemented is capable of scaling to any number of speakers in any configuration; however, we did not test less common configurations, though we would infer that it would follow a trend formed by the results we do have. We have empirically tested that so long as there was not an excessive amount of change in the arrangement of the speakers, the system would function without any appreciable change in error rate with the same set of gesture templates. In real world use, speakers and microphone arrays rarely change position relative to the user (laptops have fixed speaker/microphone arrangements, desktops are static).

One notable oddity that occurred during the user study was that when testing their personalized recognizer, participants were unable to replicate some of their gestures because they changed the speed or size of the movements. This can occur due to gorilla arm syndrome after prolonged use, which is a known shortcoming of many gesture-based interfaces [24]. The effect of speed discrepancies in gestures goes beyond just the duration of the gestures. Much in the same way that the gait of a person changes between walking and running, so too do hand gestures appear differently at different speeds. Different pauses and hooks are present. Additionally, the velocity of the object affects the distance from the peak tone that the observed frequency is seen, meaning that the vectors differ in which bins the data falls in. Resampling helps somewhat, but if the difference between the speeds is too great, they are essentially two distinct gestures. Adding additional templates of different speeds to the dataset is one way of mitigating this problem. While Multiwave is sufficiently robust to not incorrectly classify the result, it sometimes rejected the correct classification because of the rejection threshold being too high. This problem is not as prominent with experienced users, but can be troublesome for a first time user. A similar issue presented itself in our attempts to create a user independent dataset. The arm and upper body are detected when they move which makes it difficult to create a dataset that works for people with different sized hands and arms.

Younger users occasionally commented about being able to hear the sound that the speakers were making although the sounds were well outside of the audible range for humans at their ages. The quality of the speakers and sound card determines the audio fidelity. In the case of the speakers used during the experiments, they were built for better low end response so the pure tone was not quite as pure as was intended thereby allowing it to be perceived by younger participants. This problem was not experienced on integrated speakers on laptops, like the Surface Pro 3 used during our experiment, or smaller desktop speakers using a quality external sound card, but was present with the external speakers used for the five speaker scenario. The lower audio fidelity in the five speaker condition could explain why there was no reduction in recognition error rate. By improving the sample rates of both the speakers and the microphone to 96 kHz, it is possible to emit sounds at frequencies up to 40 kHz, which would prevent the sounds from being heard by any human.

A shortcoming of Multiwave and similar systems is that as the distance between the speakers and the microphone increases, the sensitivity of the system decreases. The amplitude of the sounds fall as the distance traveled increases. Every surface in the environment causes some sound reflection which also causes the sound wave to diminish. These problems were avoided in the experimental setup, but would need to be calibrated for in any other environment by modifying the volume of each speaker to allow for a clear peak to be detected at each frequency. Effective range is also a problem in camera based gesture systems, which are limited by the resolution of the camera hardware [16].

Previous literature proposed additional emitters as a possible method for increasing recognition rate of their Doppler Effect based recognizer [14]. We saw a slight difference in the recognition rate when using additional speakers, but our result was inconclusive. In some ways, increasing the number of speakers was beneficial, such as dealing with the uncertainty of gestures which were not performed in the same plane as most of the other speakers. Conversely, increasing the number of speakers increases the number of sources of signal noise, which may lead to error propagation throughout the system. These problems are inherent to the medium, as previous work has shown [21].

6 Future Work

A possible extension of this work would be to implement it in a multimodal recognition system, with Doppler effect based acoustic tracking used as a way to augment other camera free sensing methods, like Wi-Fi or cellular signals [20, 28]. The individual modalities would benefit from sensor level fusion to improve accuracy. The velocity information could be used to improve the accuracy of vision-based trackers when there is user occlusion. The data could be used like optical flow information, albeit with fewer necessary calculations. Developing an application that allows for users to add personalized gestures in real time would be trivial and would also enable an easy way to add gesture based authentication to a system.

7 Conclusion

We presented Multiwave, a system which enables gesture recognition using multiple commodity speakers and a microphone to detect complex hand movements which can be mapped naturally into applications. We demonstrated a method of detecting gestures by extracting a multidimensional vector from the frequency domain representation of the raw audio feed and then using a generalized gesture recognizer on the time series of the vectors. We ran a user study to determine the accuracy of the system. Our results show that Multiwave is capable of detecting a set of 14 complex gestures at about 94% accuracy with two speakers. Multiwave is as accurate with two or more speakers as the previous state of the art was with a single speaker, giving Multiwave the advantage of allowing for more intuitive mappings into a growing number of applications that accept gesture input.

Acknowledgments

This work is supported in part by NSF CAREER award IIS-0845921, ARL, NRL, and Lockheed Martin. We would also like to thank the other ISUE lab members and expert reviewers for their helpful feedback.

References

[1] *Amazon Fire Phone*.

[2] *Apple Watch*.

[3] *Motorola 360 Smartwatch*.

[4] M. Al Zayer, S. Tregillus, and E. Folmer. Pawdio: Hand input for mobile vr using acoustic sensing.

[5] L. Anthony and J. O. Wobbrock. A lightweight multistroke recognizer for user interface prototypes. In *Proceedings of Graphics Interface 2010*, pp. 245–252. Canadian Information Processing Society, 2010.

[6] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel. Doplink: Using the doppler effect for multi-device interaction. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 583–586. ACM, 2013.

[7] D. A. Bowman, E. Kruijff, J. J. LaViola Jr, and I. Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley, 2004.

[8] S. Cheema, M. Hoffman, and J. J. LaViola Jr. 3d gesture classification with linear acceleration and angular velocity sensing devices for video games. *Entertainment Computing*, 4(1):11–24, 2013.

[9] K.-Y. Chen, D. Ashbrook, M. Goel, S.-H. Lee, and S. Patel. Airlink: sharing files between multiple devices using in-air gestures. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 565–569. ACM, 2014.

[10] G. Cohn, D. Morris, S. N. Patel, and D. S. Tan. 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*, pp. 791–800. ACM, 2011.

[11] S. Gupta, D. Morris, S. Patel, and D. Tan. Soundwave: using the doppler effect to sense gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1911–1914. ACM, 2012.

[12] M. Hessar, V. Iyer, and S. Gollakota. Enabling on-body transmissions with commodity devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1100–1111. ACM, 2016.

[13] R. Ionescu, R. Carotenuto, F. Urbani, et al. 3d localization and tracking of objects using miniature microphones. *Wireless Sensor Network*, 3(05):147, 2011.

[14] K. Kalgaonkar and B. Raj. One-handed gesture recognition using ultrasonic doppler sonar. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 1889–1892. IEEE, 2009.

[15] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders. Fundamentals of acoustics. *Fundamentals of Acoustics, 4th Edition, by Lawrence E. Kinsler, Austin R. Frey, Alan B. Coppens, James V. Sanders, pp. 560. ISBN 0-471-84789-5. Wiley-VCH, December 1999.*, 1, 1999.

[16] J. J. LaViola Jr. An introduction to 3d gestural interfaces. In *ACM SIGGRAPH 2014 Courses*, p. 25. ACM, 2014.

[17] H. Li, W. Yang, J. Wang, Y. Xu, and L. Huang. Wifinger: talk to your smart devices with finger-grained gesture. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 250–261. ACM, 2016.

[18] B. C. Moore. *An introduction to the psychology of hearing*. Brill, 2012.

[19] C. Pittman, P. Wisniewski, C. Brooks, and J. J. LaViola Jr. Multiwave: Doppler effect based gesture recognition in multiple dimensions. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 1729–1736. ACM, 2016.

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

[21] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan. Audiogest: enabling fine-grained hand gesture detection by decoding echo signal. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 474–485. ACM, 2016.

[22] E. M. Taranta, II and J. J. LaViola, Jr. Penny pincher: A blazing fast, highly accurate $-family recognizer. In *Proceedings of the 41st Graphics Interface Conference*, GI '15, pp. 195–202. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 2015.

[23] E. M. Taranta, II, A. Samiei, M. Maghoumi, P. Khaloo, C. Pittman, and J. J. LaViola, Jr. Jackknife: A reliable recognizer for few samples and many modalities. In *To Appear: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2017.

[24] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan. Vision-based hand-gesture applications. *Communications of the ACM*, 54(2):60–71, 2011.

[25] H. Watanabe, T. Terada, and M. Tsukamoto. Ultrasound-based movement sensing, gesture-, and context-recognition. In *Proceedings of the 17th annual international symposium on International symposium on wearable computers*, pp. 57–64. ACM, 2013.

[26] G. Welch and E. Foxlin. Motion tracking survey. *IEEE Computer graphics and Applications*, pp. 24–38, 2002.

[27] H. Zhang, W. Du, P. Zhou, M. Li, and P. Mohapatra. Dopenc: Acoustic-based encounter profiling using smartphones.

[28] C. Zhao, K.-Y. Chen, M. T. I. Aumi, S. Patel, and M. S. Reynolds. Sideswipe: detecting in-air gestures around mobile devices using actual gsm signal, 2014.