Adding Directional Context to Gestures Using Doppler Effect

Adeola Bannis
Carnegie Mellon University
5000 Forbes Ave
Dept. of Electrical and Computer Engineering
Pittsburgh, PA 15213-3890
abannis@andrew.cmu.edu

Shijia Pan
Carnegie Mellon Silicon Valley
Building 23
Dept. of Electrical and Computer Engineering
Moffett Field, CA 94035
shijiapan@cmu.edu

Pei Zhang
Carnegie Mellon Silicon Valley
Building 23
Dept. of Electrical and Computer Engineering
Moffett Field, CA 94035
peizhang@cmu.edu

Ubicomp '14 Adjunct, September 13-17 2014, Seattle, WA, USA
ACM 978-1-4503-3047-3/14/09.
http://dx.doi.org/10.1145/2638728.2638774

Abstract
Human beings often give non-verbal instructions through motions of the hand and arm, such as pointing or waving. These motions convey not just actions, but the direction or target of those actions. In this paper, we integrate direction into gesture definitions by detecting frequency shifts created by relative motion between a receiver and transmitter and combining this with inertial motion data captured by a smartphone. With the combined data we are able separate similar gestures with 71.7% accuracy in a typical home use environment.

Author Keywords
gestures; ultrasound; sensor fusion

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces: input devices and strategies; interaction styles.

General Terms
Algorithms

Introduction
Arm-based gestures have gained recent attention as an intuitive method of communication with devices and interfaces, as they are already used between people. Gestures such as pointing and waving are common, well-understood

and distinctive. Most approaches to recognising arm gestures use images or video of the user, and rely on recognising specific poses or a sequence of poses [**?**] [**?**]. If a gesture is used to identify a target, both target and user must be in view of the camera, and the system must be able to determine what objects in the scene are background and which are potential targets.

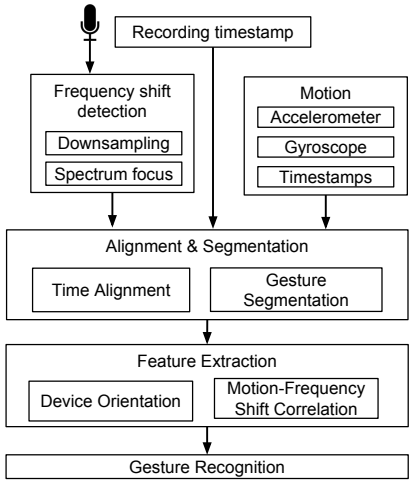

Figure 1: System Overview

There have been a growing number of systems using ultrasound to determine relative motion or positioning of devices, rather than vision-based methods. The Spartacus [**?**] and DopLink [**?**] systems both use the Doppler effect on audio frequency due to user motion to select one device from a group, or to determine the relative position of multiple devices.

In this paper, we present a method of combining the relative motion information gained from observing the Doppler effect with the inertial sensor data of a smartphone in motion to specify gestures with a directional component. This method can be extended to define targeted gestures, such as directing a command to a selection of targets from a group. An outline of this system is given in Fig. 1.

Frequency Shift Detection

When a sound source and receiver are in motion relative to each other, there is a frequency shift dependent on their relative velocities. The sign of this relative velocity indicates the direction of motion between the receiver and transmitter, i.e. whether they are moving together or apart. The relationship is approximately:

$$\Delta f = \frac{\Delta v}{c} f_0 \tag{1}$$

where Δf is the frequency shift, Δv is the relative velocity, c is the speed of sound, and f_0 is the true frequency emitted by the transmitter. Thus the observed frequency shift is directly proportional to the speed of the user's device.

Since the audio tones used are 20 kHz and above, a sampling rate of 44.1 kHz is used to satisfy the Nyquist requirement. In order to extract the timing of the frequency shift, we use a short-time Fourier transform. The desired frequency resolution at the original sampling rate would require an FFT length of about 8192 samples, which is too computationally intensive, and includes many unwanted frequency bins. To reduce the necessary FFT size, the audio is downsampled by a factor of 5, giving a new sampling rate of 8820 Hz, and a Chirp-Z transform is used to focus on a region of interest. For the gestures currently recognised, the maximum frequency shift does not exceed 200 Hz. We have chosen a band of $\pm$400 Hz from the transmitter frequency as a safe margin.

Time Alignment

In a real-time environment, it would be possible to obtain the motion sensor readings and audio samples simultaneously. However, on a typical smartphone, the motion data is sampled at intervals and delivered on a callback thread, independent of the audio capture. Furthermore, the timestamps that are provided with motion events on popular smartphone platforms are guaranteed to be consistent with other motion timestamps, but not with timestamps from other sources, e.g. the system clock used for audio capture timing. Therefore, the first step in processing is to convert the motion sensor timestamps to the same reference as the audio capture time. This is done by taking a system timestamp with the first sensor readings and subtracting the sensor timestamp to provide a rough estimate of the offset between clocks.

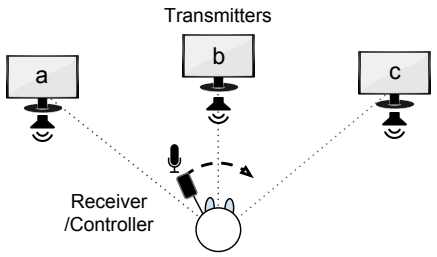

Figure 2: Sweeping motion away from (a), across (b) and towards (c)

Gesture Segmentation

Once the data is roughly aligned in time, the different timestamps and sampling intervals mean that the readings still cannot be directly compared. Instead,the frequency and motion readings are segmented into regions, separated by periods of rest. First, positive and negative thresholds are applied so that only a certain magnitude of frequency shift, acceleration or angular velocity can trigger a gesture (dashed lines in Fig. 3). Secondly, a threshold is set on the number of samples below the trigger magnitude needed before the system is considered at rest again. This is to allow crossings to register as part of the same gesture instead of being split when the thresholds are crossed. The start and end times of the segments are compared, and if they are the same to within 1 s, they are taken as components of the same gesture.

Feature Extraction

Fig.3 shows sensor readings from a series of gestures, either towards, away from, or across an ultrasound transmitter (Fig 2).

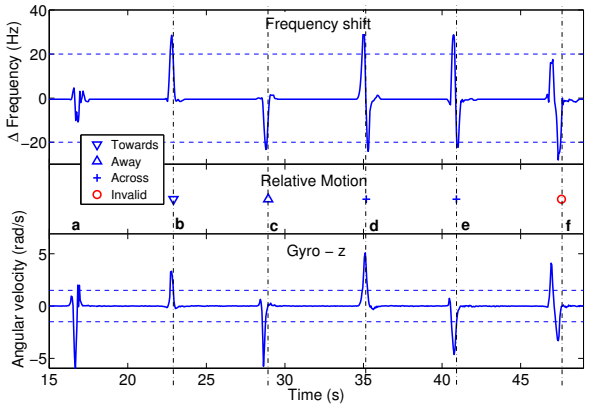

Figure 3: Relative motion to transmitter derived from motion and frequency shift

The first event (Fig. 3a) is ignored because it falls below the threshold. Frequency shift alone could be used to determine a movement towards (Fig. 3b) or away from (Fig. 3c) the transmitter, but motion data is needed to distinguish a motion across (Fig. 3d+e) the transmitter from a back-and-forth motion (Fig. 3f).

	Relative motion	Orientation
1	Towards	flat
2	Across	flat
3	Across	upright
4	Away	upright

Table 1: Example gestures

		Predicted				
		1	2	3	4	invalid
Actual	1	14	0	0	0	1
	2	2	5	0	0	8
	3	0	0	12	2	1
	4	0	0	0	12	3

Table 2: Gesture detection confusion matrix

		Predicted			
		across	towards	away	invalid
Actual	across	17	2	8	3
	towards	0	14	0	1
	away	0	0	12	3

Table 3: Relative motion confusion matrix

Results

The system was implemented with a LG Nexus 5 as the receiver, and an iPhone placed on a table as a transmitter. Table 1 shows a sample of 4 different gestures out of a possible 9 (3 orientations and 3 types of relative motion) that were chosen for recognition. Each shares a feature with one other motion in order to demonstrate the ability to distinguish similar gestures. The motion used is shown in Fig 2, with the phone held in portrait mode either perpendicular ('upright') or parallel ('flat') to the floor. The gestures were performed in random sequence at 3 m from the transmitter in a typical room containing furniture and other surfaces that could cause distortion and echoes. The resulting confusion matrix for classification is shown in Table 2. Combinations outside the chosen 4, such as 'away/flat', are marked as invalid for classification.

The system had an overall accuracy of 71.7% on these gestures. Note that gestures with disjoint features (e.g. 'towards/flat' and 'across/upright') are not confused.

Conclusion

We have shown how relative motion obtained from Doppler shift measurements can be combined with inertial motion to create a gesture scheme that allows for a wider range of gestures than either system alone. Building on these combined gesture types, we plan to develop more complex gestures, such as drawing a line between two devices to initiate a link or indicate some relationship between them. We also plan to conduct more detailed user experience evaluations for these gestures.

References

[1] Aumi, M. T. I., Gupta, S., Goel, M., Larson, E., and Patel, S. Doplink: Using the doppler effect for multi-device interaction. In *Proc. Ubicomp*, ACM Press (2013), 583–586.

[2] Sigalas, M., Baltzakis, H., and Trahanias, P. Gesture recognition based on arm tracking for human-robot interaction. In *Proc. IROS* (2010), 5424–5429.

[3] Sun, Z., Purohit, A., Bose, R., and Zhang, P. Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing. In *Proc. MobiSys* (2013), 263–276.

[4] Zigelbaum, J., Browning, A., Leithinger, D., Bau, O., and Ishii, H. G-stalt: A chirocentric, spatiotemporal, and telekinetic gestural interface. In *Proc. TEI*, ACM Press (2010), 261–264.