Context Sensing with the Twiddler Keyboard

Daniel Ashbrook
College of Computing
Georgia Institute of Technology
gt5257d@prism.gatech.edu

Abstract

Context sensitivity is an important application of wearable computers. This paper describes research on using the Twiddler one-handed keyboard for sensing motion-associated context. Two possible methods for detecting walking are described, and other types of movement context that it might be possible to sense are discussed. Results of early experiments with a preliminary Twiddler driver modification are discussed.

1. Introduction

The ability to sense context is an important component of wearable computing applications. It allows the application to automatically respond to changing situations in a way that makes the wearable even more useful. The Twiddler one-handed keyboard, manufactured by HandyKey, is very popular among users of wearable computers and is therefore the perfect device to augment with context-sensitivity.

The Twiddler's usefulness lies in the fact that it has two tilt sensors incorporated into it for use as a mouse. One sensor detects the degree of tilt in the horizontal direction, and the other in the vertical. According to HandyKey's technical information, "the tilt output is nearly linear over a ±30º range with a corresponding count range of ±50. At ±90° the count value is approximately ±200."[1]

The idea behind my research is to allow the computer to sense when the user is in various stages of motion. The first, and most basic type of motion to be detected is walking. Since walking involves a range of relatively predictable movements, it is an ideal context to attempt to sense.

Aside from the ease of sensing, context awareness of walking has several obviously useful applications. One use, suggested by Thad Starner[2] (who initially inspired this research) is to automatically pull up a to-do list, which he regularly accesses while walking from one appointment to the next. Another use, if individual steps were detected, could be an odometer, to calculate the distance walked in a day. This could be combined with data on the usage of the computer to create yet another compelling argument for using a wearable!

Since the Twiddler's sensors only provide two degrees of freedom, positioning using only the inertial sensors is an impossibility. However, in conjunction with some fairly inexpensive (and therefore inaccurate) radio positioning tools, it might be possible to get fairly precise positional measurement.

2. Sensing walking

Looking at a graph of a simulated walk, such as the example in Figure 1 (created by bouncing the Twiddler up and down manually), two methods for detecting a walk immediately suggest themselves. One derives from the fact that the maximum hovers around 500 for about half a second for each step. This would be trivial to detect. The other possibility is to identify each individual step by some process.

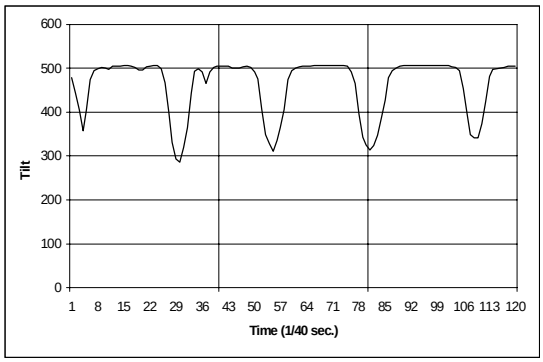

Figure 1. Manually created data set

2.1. Maximum detection

Looking at the graph in Figure 1, it is obvious that the maximum for each step floats close to 500 periodically, for about half a second at a time. During the second step, however, there is a small downspike; this is a condition that would have to be ignored in any calculations. Aside from this, watching the maximum values over a period of

[1] http://www.handykey.com/twiddler-data-format.gif
[2] testarne@cc.gatech.edu

several seconds, and seeing if they stay within a specific range (such as 500 ±10) would provide a very satisfactory result.

Figure 2 shows a graph of actual data collected from one of the Twiddler's tilt sensors. Although there is quite a bit more variation in the values than is present in the manually created graph, the majority of nearly-straight segments exist around the 500 area. Checking for a majority of time spent at 500 ±20 would probably yield correct results.

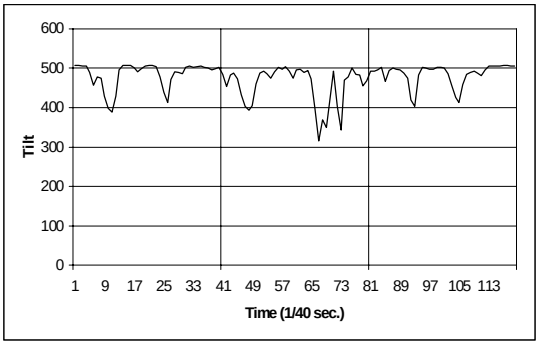

Figure 2. Real-life usage data

2.2. Individual step detection

A natural progression of detecting periods of values around a certain point is to combine this measurement with measurements of minimum values and attempt to detect individual steps.

Looking at Figure 1, very clear hills and valleys are recognizable. In Figure 2, they are not as well defined, but are still recognizable. A basic algorithm would watch for a minimum value (defined as the smallest measurement encountered in some interval that is lower than some value, both interval and value probably user defined), and then watch for several values in a row with close to the same values. When another minimum is hit, a step would be registered.

One of the possible problems that could be encountered with this method is represented on the graph in Figure 2. At just under 2 seconds, there is a large spike in the graph, caused perhaps by a misstep or twitch of some kind. By watching for several values in a row near each other, rather than just one maximum, spikes are ignored. Another potential problem is erroneous detection; for example, if the user stood still for several seconds, that could be registered as a step. By specifying a minimum value for the steps to be detected, however, this could be avoided.

2.3. Other methods

One possible method for greater accuracy in detecting steps is to apply a Fourier transform to the collected data; when a strong low-frequency oscillation is detected (around 2 Hz?), it could be interpreted as walking. This could also lead to an easier method of detecting other locomotive activities, such as walking down stairs or running.

Another possibility is to analyze data from both the horizontal and vertical sensors of the Twiddler; however, since there appear to be few side-to-side movements in most peoples' walks, there is some doubt as to whether this would prove practical.

Further improvements to these ideas would definitely be helped by research into walking patterns of different people, as well as data collected from various users of wearable computers.