

Accurate Caloric Expenditure of Bicyclists using Cellphones

Andong Zhan, Marcus Chang, Yin Chen, Andreas Terzis

Computer Science Department

Johns Hopkins University

Baltimore, MD 21218

{andong, mchang, yinchen, terzis}@cs.jhu.edu

Abstract

Biking is one of the most efficient and environmentally friendly ways to control weight and commute. To precisely estimate caloric expenditure, bikers have to install a bike computer or use a smartphone connected to additional sensors such as heart rate monitors worn on their chest, or cadence sensors mounted on their bikes. However, these peripherals are still expensive and inconvenient for daily use. This work poses the following question: *is it possible to use just a smartphone to reliably estimate cycling activity?* We answer this question positively through a pocket sensing approach that can reliably measure cadence using the phone's on-board accelerometer with less than 2% error. Our method estimates caloric expenditure through a model that takes as inputs GPS traces, the USGS elevation service, and the detailed road database from OpenStreetMap. The overall caloric estimation error is 60% smaller than other smartphone-based approaches. Finally, the smartphone can aggressively duty-cycle its GPS receiver, reducing energy consumption by 57%, without any degradation in the accuracy of caloric expenditure estimates. This is possible because we can recover the bike's route, even with fewer GPS location samples, using map information from the USGS and OpenStreetMap databases.

1 Introduction

A biking renaissance has been underway over the past two decades in North America [34]. During this time, the annual number of bike trips increased from 1.7 billion to 4 billion while the number of daily bike commuters in the United States nearly doubled [16].

This higher rate of biking has diverse benefits including improved health and well-being for bikers, reduced transportation costs and road congestion, and lower energy costs and environmental impact [16]. Especially from a health per-

spective, biking is one of the best exercises for improving one's health and fitness, giving the heart and circulatory system a vigorous workout [23].

This ever-increasing class of biking enthusiasts has been prompted by the availability of low-cost fitness devices, GPS receivers, and cellphones to track their outdoor biking trips and calculate their performance, including calories consumed, during those trips. Of particular interest are smartphone applications that use GPS tracks to allow users to analyze their trips and compare their efforts with their friends. For example, MapMyRIDE [9], a popular biking application (or *app*), has been downloaded more than 500,000 times from the Android Market. In addition to tracking the routes and distance traveled during bike trips some of the cellphone apps estimate the calories consumed during those trips [6, 9]. At a high level, these estimates are calculated by looking up the biker's weight and average travel speed in a standardized caloric expenditure table.

Nevertheless, precise measurement of cycling activity is still a challenging task. Specifically, existing GPS-only biking apps do not directly measure the cyclist's activities such as the number of revolutions of the bike's crank per minute (known as *cadence*). Moreover, the inherent altitude errors in GPS measurements complicate the slope change estimations during a bike trip significantly. Put together, these shortcomings lead to imprecise estimates of caloric expenditure. Our measurements, detailed in Section 4, suggest that the errors can be significant.

A straightforward approach to increase accuracy would be to include sensors that measure the biker's effort directly. Examples of such sensors include power meters, cadence sensors, and heart rate monitors. Together with an on-bike computer or smartphone, these sensors not only report accurate caloric expenditure, but can also show bikers their real-time biking power and help them remain in their target heart rate zone for best training results. Furthermore, those sensors can also measure cadence and thereby let the bikers stay in an RPM range that is safe for their knees, which can be especially valuable during long rides [17].

Unfortunately, such an on-bike sensing system is still expensive (can cost above \$1,000) and can be cumbersome for daily use. The question we want to answer in this work is: *is it possible to use just a smartphone to reliably provide all this biking information?*

We describe our efforts to answer this question through

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SenSys'12, November 6–9, 2012, Toronto, ON, Canada.
Copyright © 2012 ACM 978-1-4503-1169-4 ...\$10.00

the development of a precise and smartphone-only biking tracking solution which only requires bikers to wear a smartphone in their front pant pockets during their bike trips. The smartphone uses its on-board accelerometer to sense the periodic movement of the bikers’ legs as they pedal, thereby eliminating the need for a separate cadence sensor. In addition, the smartphone provides a precise three-dimensional trace of the biker’s route. This trace is reconstructed by processing raw GPS samples with information about the road network and detailed elevation maps provided by OpenStreetMap [11] and the US Geological Service (USGS) [18]. The smartphone inputs all this information along with real-time weather reports to a model that estimates biking power and caloric expenditure.

Results from a total of 30 km bike trips using our bike sensing system (shown in Fig.2) show that the smartphone can reliably infer the cadence from the accelerometer data with an overall error of less than 2%. Furthermore, results from 20 bikers using 7 bikes over 70 bike trips show that the model-based approach can accurately estimate caloric expenditure and reduce error by 60% compared to other smartphone based approaches. Finally, the smartphone can aggressively duty-cycle its GPS receiver without any degradations in the accuracy of caloric expenditure estimations. Doing so reduces its energy consumption by up to 57%. This is possible because we can recover the bike’s route, even with fewer GPS samples, using a robust route recovery mechanism that takes as input detailed map information from OpenStreetMap and USGS.

We make three contributions: (1) we propose a pocket sensing approach to accurately sense biking activity that replaces expensive on-bike hardware; (2) we compare and analyze major elevation services. We find the “bridge elevation” error on both USGS and Google Maps services and minimize this error to provide highly precise elevation for biking caloric expenditure calculation; (3) we show that leveraging detailed map information from USGS and OpenStreetMap enables very aggressive duty-cycling of the GPS receiver thereby saving a significant amount of energy.

This paper has six more sections. Section 2 outlines existing caloric expenditure techniques and Section 3 describes the mechanisms and techniques we explored in our work. In Section 4 we evaluate the accuracy of the proposed system’s different components and compare them with existing techniques for caloric expenditure. Section 5 discusses the feasibility of implementing the proposed method as a real-time smartphone app and presents some of the lessons learned. We introduce the related work in Section 6 and conclude in Section 7 with a summary.

2 Background and Motivation

There is currently a strong trend to use biking as a way of controlling or reducing one’s weight. Cyclists are therefore interested in knowing their caloric expenditures while riding outside. A straightforward approach is to outfit one’s bike with a power meter that converts the power required to move the bike into caloric expenditure. Professional bikers use this method but power meter cranksets are expensive (the costs are usually higher than \$1,000 [13]) and therefore

Speed	130 lbs	155 lbs	190 lbs
<10mph, leisure	236	281	345
10-11.9mph, light effort	354	422	518
12-13.9mph, mod. effort	472	563	690
14-15.9mph, vig. effort	590	704	863
16-19mph, very fast	708	844	1035
>20mph, racing	944	1126	1380

Table 1. Calories burned per hour while biking (originally from [14]).

unlikely to be used by amateurs. Instead, several caloric expenditure models have been developed that approximate the actual calories burned. In what follows we introduce four caloric expenditure estimators.

2.1 Search Table

Several healthcare and sports institutes provide caloric expenditure tables that bikers can use to estimate their caloric consumption [20, 14]. Energy expenditure is tabulated according to average speed, trip duration, and the biker’s weight. For example, according to Table 1, originally found in [14], biking at 13 mph burns 563 calories per hour if you weigh 155 lbs and 690 calories per hour if you weigh 190 lbs.

This estimation method is used by multiple mobile sport and dietary tracking applications [5, 6, 7, 9]. These applications track users through the phone’s GPS receiver during cycling or let them manually input the distance and their speed after the trips, and then estimate calories burned. While easy to use this method has low accuracy: lacking elevation information, it overestimates calories burned on downhill trips during which effort is low but speed is high and underestimates calories burned on uphill ones.

2.2 Cadence and Speed Sensing

Cadence and speed sensing is another way to estimate caloric expenditure while biking. In addition to recording the bike’s cruising speed, this method uses a cadence sensor to measure pedaling speed in revolutions per minute (RPM). Such sensors are commercially available including ones with Bluetooth or ANT+ radios. According to (1), found in [22], given bike velocity V and pedaling speed S the oxygen consumption VO_2 in liters per min is given by:

$$VO_2 = 0.00494(0.261V^3 + 0.671mV)^{0.589}S^{0.168} \quad (1)$$

where m is the total mass of the rider and the bike in kilos.

Then, we can estimate the calories burned by multiplying VO_2 in l/min by a factor of 5 [22]. For example, suppose that $V = 25$ km/h, $S = 60$ rpm, $m = 70$ kg and the cyclist is adopting the touring posture. Then (1) results in $VO_2 = 1.53$ l/min which is equivalent to 7.66 Kcal/min.

Compared to the search table method, estimating caloric expenditure using cadence and speed sensing avoids counting calories when the rider is not pedaling, thereby significantly decreasing the overestimation of calories burned during downhill trips. On the other hand, cadence and speed sensing tends to underestimate caloric expenditure during uphill trips during which pedaling and bike speed are low.

2.3 Heart Rate Monitoring

Heart rate monitors, most commonly found in the form of chest straps, are a popular and accurate method of estimating calories burned while exercising. Keytel et al. proposed a model which takes heart rate as input and estimates calories burned while biking after adjusting for age, gender, body mass, and fitness level [28].

Their experiments showed that the correlation coefficient between the actual and estimated energy expenditure was 0.913 [28]. According to their model, caloric expenditure depends on a variety of factors, but primarily the biker's heart rate and VO_2 max. VO_2 max is the maximum oxygen consumption rate which is a good measure of aerobic conditioning. To estimate VO_2 max, bikers need to perform a Cooper VO_2 max test which requires them to run as far as possible in 12 minutes. The VO_2 max estimate then is:

$$VO_2 \text{ max} = (D - 504.9)/44.73 \quad (2)$$

where D is the distance covered in meters. Taking a male biker as an example, we can calculate caloric expenditure as:

$$\begin{aligned} \text{Calories} = & [(0.380 \cdot VO_2 \text{ max}) + (0.450 \cdot BPM) \\ & + (0.274 \cdot \text{age}) + (0.0468 \cdot \text{weight}) - 59.3954] \\ & \cdot \text{time}/4.184 \end{aligned} \quad (3)$$

where BPM is the heart rate in beats/min.

The heart rate monitoring approach enables bikers to estimate calories burned with high accuracy at an affordable price. However, wearing a chest strap may be too cumbersome for daily use. An informal survey among the 20 volunteers that participated in our evaluation (see Sec.4) confirmed this intuition. Furthermore, the chest strap has to be worn properly (e.g., tight against the bare chest) for heart rate measurements to be accurate. Ensuring proper use can be non-trivial in practice. In fact, some recent studies have shown that the adoption of heart rate monitors is severely limited by the deficient ergonomics of the chest straps [21].

Unfortunately, heart rate monitors using chest straps are still by far the most convenient and accurate solution for outdoor exercise. Other popular types of heart rate monitors either require both hands to touch the device [12] or need a finger clip [10], both impractical and uncomfortable for outdoor biking.

2.4 Power Measurement

The heart rate monitor approach estimates caloric expenditure by directly measuring changes in the biker's physiology. The power measurement approach on the other hand leverages the observation that the number of calories burned is linearly related to the total amount of work necessary to move the combined mass of the biker and the bike from start to finish. James et al. showed that cycling power can be accurately predicted by a mathematical model with very high accuracy ($R^2 = 0.97$) [32]. In what follows we describe this model step by step.

According to the laws of Mechanics the power P that a rider needs to generate to maintain a constant ground speed V_g is equal to:

$$P = FV_g \quad (4)$$

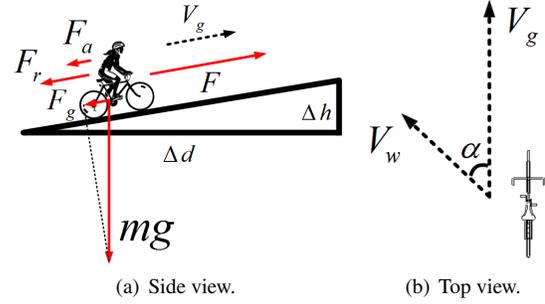


Figure 1. Forces related to determining the power necessary to move a bike at constant speed V_g given wind vector V_w and road slope s .

where F is the force generated by the rider along the direction of movement.

Based on Newton's First and Third Laws of Motion, F is equal to the total resistance which has the following three components:

$$F = F_r + F_g + F_a \quad (5)$$

where F_r is the rolling resistance from the bike, F_g is the component of gravity along the direction of movement, and F_a is the force of aerodynamic drag. Analyzing these forces further, we have:

$$F_r = mgC_r \quad (6)$$

$$F_g = mgs \quad (7)$$

$$F_a = \rho(T)C_aV_a^2 \quad (8)$$

According to (6) the force F_r due to rolling resistance is the product of the combined mass of the rider and the bike m , the acceleration g due to Earth's gravity, and the lumped coefficient of rolling resistance C_r which accounts for tires, bearings, chain, and so on. F_g , that represents the component of the gravitational force along the direction of movement, is given by (7). The slope s is equal to $\Delta h/\Delta d$ (cf. Fig.1). Finally, (8) is the total aerodynamic drag force which is related to the frontal area and shape of the bike and rider and to the air density and air velocity. As illustrated in Figure 1(b), we have $V_a = V_g + V_w \cos \alpha$, where V_w is wind velocity. $\rho(T)$ is the temperature dependent air density and C_a is a lumped constant for aerodynamic drag.

Substituting F in (4) with the results from (5), (6), (7), and (8) we have:

$$P = (mg \cdot (C_r + s) + \rho(T) \cdot C_a V_a^2) \cdot V_g \quad (9)$$

Given P , one can estimate the number of calories burned per second by multiplying it with the mechanical efficiency of the human body while cycling. This efficiency is found to be $\approx 25\%$ [33].

In practice, estimating P requires measuring the road's slope s accurately and accounting for the variations across bikers and bikes, reflected in the constants C_a and C_r .

The rest of the paper describes our efforts in implementing the four caloric expenditure mechanisms and comparing their accuracy across different routes, bikes, and bikers.

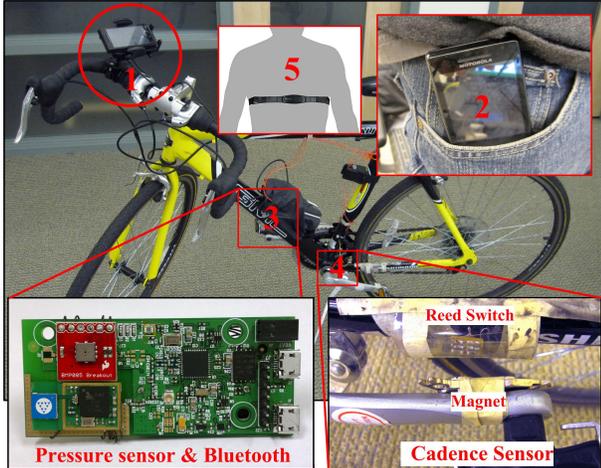


Figure 2. System overview: (1) on-bike smartphone collecting data from cadence, pressure, and heart rate sensors via Bluetooth; (2) on-body smartphone collecting data from on-board accelerometer and GPS; (3) Barometric pressure sensor (red board) connected to mote with Bluetooth radio inside enclosure; (4) Cadence sensor made by magnet and reed switch; (5) heart rate monitor worn around the biker’s chest.

3 System Design

We built a modular sensing testbed that continuously monitors key physical processes related to biking. We use this system to compare the different caloric expenditure estimation methods from the previous section. As Figure 2 illustrates, the system consists of five major components: a cadence sensor, a barometric pressure sensor, a heart rate sensor, and two smartphones.

The central hub of the sensing system is a smartphone mounted on the bike’s handlebar, shown as Item 1 in Figure 2. This smartphone wirelessly collects data from the cadence sensor, the barometric pressure sensor, and the heart rate sensor via Bluetooth. The second smartphone, shown as Item 2 in Figure 2, is placed inside the biker’s front pant pocket and captures the movements of the biker’s leg using its accelerometer sensor. This smartphone also continuously logs data from the GPS receiver providing detailed route and speed information for every bike trip. We note that these five components are needed to evaluate the different caloric expenditure estimation techniques. As we will show in Section 4, just one smartphone in the biker’s pocket is enough to accurately estimate caloric expenditure.

While the off-the-shelf heart rate sensor already has a built-in Bluetooth module, neither the cadence nor the barometric pressure sensor has such an interface. Therefore, we connect these two sensors to a custom mote equipped with a Bluetooth module. Specifically, the pressure sensor is mounted on the bike’s frame (Item 3 in Figure 2) inside an opaque plastic container. The container protects the sensor from the effects of wind and light, allowing more accurate barometric pressure measurements. We use an off-the-shelf Bosch BMP085 digital pressure sensor which provides measurements with ± 1.0 hPa absolute accuracy and ± 0.2 hPa

Route	Dist.(km)	Road Conditions
R1	1.5	neighborhood, uphill
R2	2.1	neighborhood, uphill
R3	0.8	neighborhood, downhill
R4	0.8	neighborhood, uphill
R5	2.1	neighborhood, downhill
R6	1.1	neighborhood, downhill
SMDN & SMDS	1.5	woods, river valley, ups and downs, winding path
SMDC	2.4	woods, river valley, ups and downs, winding path
DL	2.5	lakeside, flat, open field
WW	1.7	bridges, ups and downs
WE	1.7	bridges, ups and downs
HJ	2.9	neighborhood, bridge, downhill
JH	2.9	neighborhood, bridge, uphill
C	3.9	flat, circle, open field

Table 2. Bike routes used in caloric estimation experiments.

relative accuracy at 25°C from 700 to 1100 hPa [3]. The sensor is connected to the mote via the I²C bus.

The cadence sensor has two components, a magnet attached to one of the bike’s crank arms and a small circuit board on the chain that hosts two parallel reed switches and a capacitor (see Item 4 in Figure 2). The reed switches are connected to ground and a GPIO pin with an internal pull-up resistor on the mote. Therefore, when the magnet moves close to the board, the reed switches close the circuit and the mote reads a logic ‘0’; when the magnet moves away from the board, the reed switches open and the mote will read a logic ‘1’ due to the internal pull-up resistor.

We use a Zephyr BioHarness [19] as the heart rate sensor, shown as Item 5 in Figure 2. This sensor has to be worn tightly around the biker’s bare chest and reports the measured beats-per-minute (BPM) data to the smartphone via its Bluetooth module once per second.

Although the two smartphones run separately, we implemented a single Android application to configure and sample these sensors as a background service. We have used the Motorola Droid, Samsung Nexus S, and Samsung Galaxy 2 phones running Android 2.2.2 to 2.3.6.

3.1 Data collection

We used this system to collect data from 15 bike routes located around Johns Hopkins University’s Homewood campus in Baltimore, MD from January to April 2012. All the routes can be completed within 20 minutes and therefore we assume that weather conditions remain stable during each trip. Table 2 lists the routes’ characteristics. While the routes consist mostly of paved road segments, they include flat, uphill, and downhill sections. The routes’ surroundings include urban areas, wooded areas, and open fields.

The system samples the GPS, pressure sensor, and heart rate monitor once per second and the accelerometer at 50 Hz.

3.2 Cadence Sensing in the Pocket

Cadence, defined as the number of revolutions of the bike’s crank per minute, can be a key component in determining exercise intensity. For example, (1) uses cadence

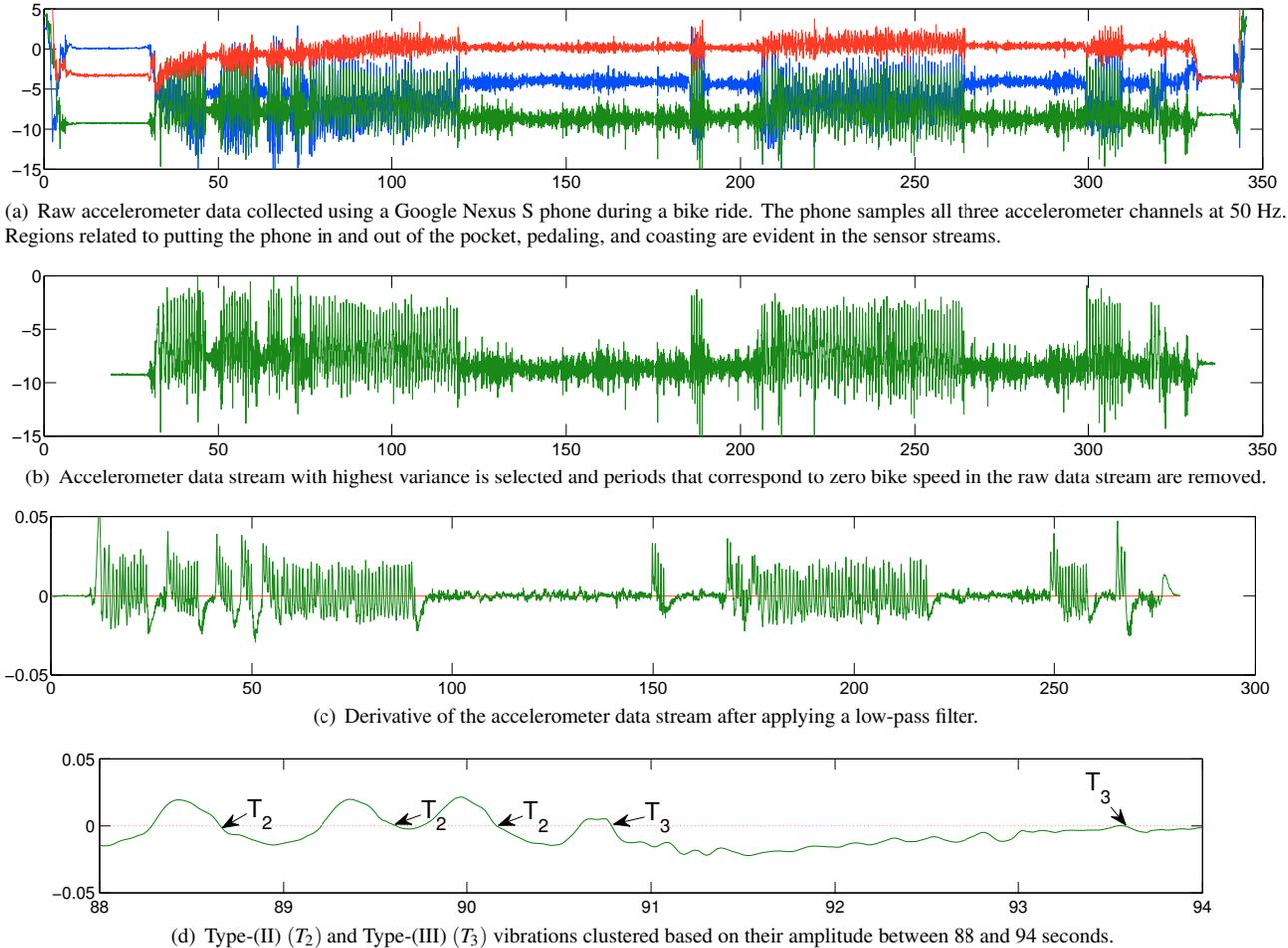


Figure 3. Data processing steps taken to estimate cadence from raw accelerometer samples.

and speed to estimate the number of calories burned during bike trips. In addition, knowing the cadence in real-time helps bikers burn calories efficiently by maintaining the ideal rhythm (60-80 RPM for most people [17]).

Existing cadence sensing schemes use a cadence sensor mounted on the bike’s frame which communicates wirelessly with a cycling computer or a smartphone. While effective, this approach requires purchasing and mounting the sensor on the bike and periodically replacing the sensor’s batteries. Instead, we introduce a mechanism that does not require a separate hardware sensor but estimates cadence indirectly by sensing the periodic movements of the biker’s leg.

The proposed method requires bikers putting their cell-phones in one of their front pant pockets. We then use the phone’s accelerometer to track the leg’s periodic movement as the biker pushes the pedals and estimate the number of crank revolutions per minute. In addition to saving the cost of an external cadence sensor, the software-only approach reduces the phone’s energy consumption since no communication with an external sensor is necessary and the energy consumption of the accelerometer is low (e.g., 0.2mA on the Google Nexus S according to the Android Sensor API [1]).

Figure 3(a) presents an example of raw accelerometer

data collected during a 350 second trip sampled at 50Hz. Three distinct vibration types are present in the figure: (I) sharp fluctuations during the beginning and the end of the trip, reflecting the person moving the phone in and out of his pocket; (II) severe vibrations (e.g., interval between 210 and 260 seconds) which reflect the biker moving his legs as he is pedaling; (III) minor vibrations (e.g., period between 130 and 160 seconds) corresponding to intervals during which the biker is not pedaling. Vibrations during these periods are due to the bike’s interaction with the road and the biker’s general movement unrelated to pedaling.

This example suggests that in order to detect movement associated with pedaling we need to filter out patterns (I) and (III) as well as random noise. Type-(I) patterns are easy to remove via a time threshold filter that removes the first and last x seconds of data during a bike trip. One can also use the speed information from the GPS receiver to remove accelerometer data collected when the biker’s speed is zero.

Separating Type-(II) from Type-(III) vibrations is more challenging. The first step is to select among the three accelerometer axes the one which simplifies the distinction between Type-(II) and (III) vibrations the most. As Figure 3(b) shows, we choose the axis with the largest variance after

removing Type-(I) vibrations. Intuitively, this acceleration axis captures the up-down movement of the biker’s leg as he pushes the crank. Next, we use a low pass filter to remove noise and Type-(III) vibrations.

Given that the selected accelerometer axis is aligned with the direction that the biker’s leg moves the most, every revolution of the crank will result in a local maximum and minimum. We calculate the first derivative of the acceleration signal and detect these extrema by recording the zero crossings (we only consider the zero crossing from positive to negative here). We then estimate the cadence by counting the number of zero crossings per minute.

However, as one can see in Figure 3(c), Type-(III) vibrations persist in the signal even after applying the low pass filter and calculating the first derivative. Therefore we cannot include all zero crossings in the cadence estimation. Fortunately, Type-(III) vibrations have much smaller accelerations than Type-(II) vibrations. Based on this observation we utilize the k-means algorithm to cluster the local maxima to one of the two types based on the amplitude of the immediately previous peak in time. Figure 3(d) shows an example of this process. The first Type-(II) vibration is detected because it follows a local peak (maximum) whose high value classifies it in the Type-(II) cluster.

In Section 4.1 we compare the proposed cadence sensing algorithm with the hardware cadence sensor from Section 3.

3.3 Elevation measurement

According to (9), elevation is a crucial factor in estimating caloric expenditure. The digital barometric pressure sensor described earlier can be used to calculate absolute altitude using the international barometric formula:

$$\text{altitude} = 44330 \cdot \left(1 - \left(\frac{P}{p_0}\right)^{\frac{1}{5.255}}\right)$$

where p_0 is the pressure at sea level, i.e., 1013.25hPa at 15°C. Thus, a pressure change of $\Delta p = 1\text{hPa}$ corresponds to 8.43m at sea level. Since the BMP085 measures pressure with an relative accuracy of $\pm 0.2\text{hPa}$, this translates into an elevation error within 2 meters.

We note that the absolute altitude calculated from this formula is affected by weather conditions such as temperature and humidity. Since all our trips last less than 20 minutes, we treat the weather conditions as constant. Furthermore, since we are interested in the relative altitudes of neighboring locations along the bike route rather than the absolute altitude, biases in the absolute altitude can be ignored.

Similar to estimating cadence, we consider methods to infer the value of interest (i.e., altitude) indirectly. In this case we explore the data collected from the phone’s GPS receiver. Although the GPS receiver already provides altitude estimations directly, as our evaluation below shows, these estimates are less accurate than the estimated longitude and latitude.

Instead, we use the latitude and longitude coordinates to query a Geographic Information System (GIS) that provides highly accurate elevation data. In this study we use the services provided by the U.S. Geological Survey (USGS) [18] and Google Maps [15]. The USGS elevation service provides an HTTP interface to the National Elevation Dataset

(NED), which covers all of the United States with 10-meter resolution and most of the densely populated areas within the United States with 3-meter resolution. The Baltimore metropolitan area in our experiments is part of the 3-meter resolution dataset. In comparison, Google only provides a 19-meter resolution dataset for the same metropolitan area.

Figure 4(a) illustrates the performance of the four different elevation estimation techniques over a sample path. The estimate from the pressure sensor in this example is shown after it was processed with a moving average to remove noise, while the GPS curve corresponds to raw readings from the GPS receiver on the phone. It is evident that the elevation estimates from the pressure sensor, USGS, and Google elevation services are close to each other, while the directly measured GPS estimates are both an order of magnitude less accurate and precise.

The most interesting part of this path is around the middle of the trip when the biker crossed a 10-meter long bridge. Figure 4(b) zooms in to this section where it is obvious that elevation data from the pressure sensor, USGS and Google do not match in this particular region. The pressure sensor in this case reflects the real elevation changes on the bridge, while querying the USGS elevation service returns the elevation of the terrain (i.e. the riverbed under the bridge). Estimates from the Google elevation data are particularly inaccurate in this location due to the averaging effect generated by the database’s low resolution.

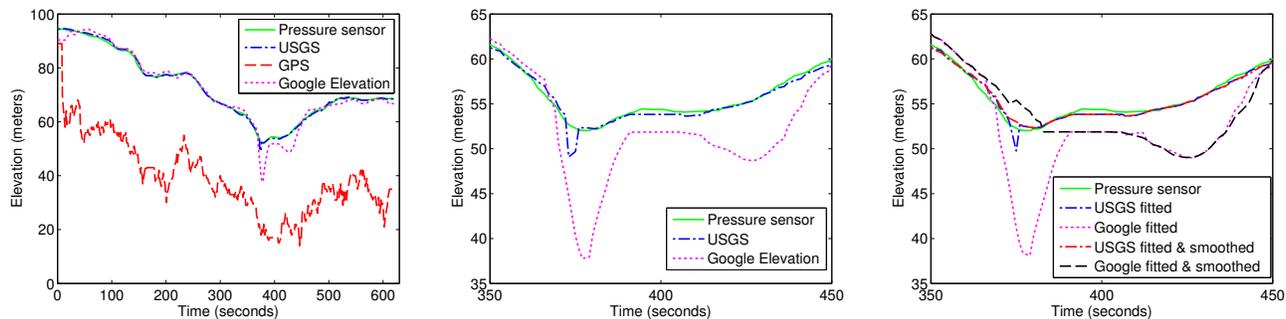
The first step towards refining the elevation data derived from the USGS and Google services is to minimize the error propagated from the GPS-provided latitude and longitude, which themselves are error prone. To do so, we assume that all bike trips take place on either marked paths or roads. We then use the open source GIS, OpenStreetMap (<http://www.openstreetmap.org/>), to match the collected GPS trace to the nearest roads forming a route (map matching) and subsequently project each GPS coordinate to the nearest point on this route (map snapping).

Specifically, we represent a particular GPS trace as a sequence of points $P = (p_0, \dots, p_{n-1})$ and project each point p_i from this trace to a route $Q = (q_0, \dots, q_{n-1})$ matching road segments via the following steps:

1. Download all street segments in the vicinity (100 meters) of each $p_i \in P$ from OpenStreetMap.
2. For each $p_i \in P$ find the nearest road segment, L_i .
3. If the distance from p_i to the previous point’s (p_{i-1}) road segment, L_{i-1} , is smaller than the GPS reported accuracy, set L_{i-1} as the new L_i , otherwise keep the old one.
4. Project p_i onto the line segment L_i and store the point as q_i in Q (in case the projection exceeds the line segment, set q_i as the nearest endpoint).

Through this process, every coordinate from the GPS trace is fitted to a road segment.

While generally useful, the road fitting mechanism cannot remove the “bridge error” shown in Figure 4(c) completely. Instead, we treat data from the USGS and Google elevation



(a) Estimated elevation across sample bike path from different services. (b) Elevation services that use external databases (USGS, Google) fail when meeting a bridge. (c) Smoothing allows elevation services to approach the accuracy of the pressure sensor.

Figure 4. Comparison of elevation estimates calculated by four different elevation services over the same path. Altitude estimates provided by the GPS receiver are the least accurate, while smoothed versions of data provided by the elevation services can be as accurate as the altitude derived from the pressure sensor.

services over the bridge section of the path as outliers and smooth it using a robust local regression method [8] which is not influenced by these outliers. Specifically, we use a quadratic polynomial model to fit the elevation data and set the span to be nine data points. The robust weights for each data point in the span are given by the bisquare function,

$$w_i = \begin{cases} (1 - (r_i/6MAD)^2)^2, & |r_i| < 6MAD, \\ 0, & |r_i| \geq 6MAD, \end{cases}$$

where r_i is the residual of the i -th data point produced by the quadratic polynomial smoothing and MAD is the median absolute deviation of the residuals, i.e., $MAD = \text{median}(|r|)$. So if r_i is larger than $6MAD$, we treat it as an outlier.

The curve named *USGS fitted & smoothed* in Figure 4(c) is the smoothed version of the *USGS fitted* curve and reduces the “bridge error” completely without introducing new errors. On the other hand, even though smoothing with the robust regression method can reduce the “bridge error” in the Google elevation data, the inherent error from the elevation service is more difficult to eliminate.

Section 4.2 compares the accuracy of the caloric expenditure estimations based on these two GPS-derived elevation estimates with the estimates achieved by using the directly measured elevations (GPS and pressure sensor).

3.4 Calibration of Power Measurement

With the help of the location and elevation services described in the previous sections we can determine the speed of the bike at regular sample points and the slope of the road between successive sample points. Moreover, the total mass of the biker and the bike can easily be measured.

The factors missing from (9) are then the lumped coefficients of rolling resistance C_r and aerodynamic drag C_a and the weather conditions, including temperature, wind speed, and wind direction. This weather information can be accessed from online weather services or from local weather stations. In our experiments we use the Weather Underground (<http://www.wunderground.com/>) station that is closest to our campus.

We designed a simple procedure to estimate the lumped coefficient of rolling resistance C_r : the biker needs to find a

flat straight path that is 50 meters or longer, (e.g., an outdoor parking lot), and activate GPS tracking. The biker then needs to follow three simple steps: (1) accelerate the bike before the start of the 50-meter path by pedaling, (2) stop pedaling and keep the bike straight on the path without braking, and (3) stop the bike at the end of the path.

Assuming no wind, the only force that applies on the bike along the direction of movement, when the biker is not pedaling, is the rolling resistance. From Newton’s Second Law of Motion we have $F_r = ma$ where a is the acceleration. Replacing F_r with (6) and solving for C_r we have:

$$C_r = \frac{a}{g} = \frac{\Delta v}{\Delta t g} \quad (10)$$

where Δv is the change in speed over the time interval Δt . Given the conditions of the experiment, the change in speed is a direct result of the rolling resistance and by using (10) we can find C_r for any specific bike and rider by collecting this calibration trace. In our calibration tests, using seven bikes of different types including road bikes and mountain bikes, we found C_r to be between 0.07 and 1.15.

The lumped coefficient of aerodynamic drag C_a is harder to estimate, requiring a wind tunnel to be measured accurately. Instead, we use an empirical reference value for C_a . Depending on the biker and bike type, C_a varies from 0.185 to 0.299 (whereas the drag coefficient varies from 0.7 to 1.15) [2]. According to the UK’s National Cyclists’ Organization [24], the drag coefficient for an everyday average biker is 1.0 and the aerodynamic drag is $C_a = 0.26$. We adopt this value for our caloric expenditure estimator.

4 Evaluation

In the first half of this section we evaluate the quality of the software-based sensors and compare them against their hardware counterparts. The second half compares the different caloric expenditure models against the heart rate based one and evaluates the impact that each of the different sensors have on the estimation error.

4.1 Cadence Sensing

We use the hardware cadence sensor mounted on the bike as ground truth and compare its results to the in-the-pocket

Relative error per trip (%)	0.19 ± 1.59
Error per kilometer	-0.09 ± 3.40

Table 3. Errors in estimated crank rotations between the hardware cadence sensor and the accelerometer-based software approach.

sensing method from Section 3.2. The traces are collected on roads around the JHU campus that include features such as uphill and downhill segments and sharp turns. Since these roads are public, several real-life situations may happen, such as changing traffic lights and yielding to traffic.

The cumulative length of the traces used in this comparison is 30.3 km, totaling 5,377 crank revolutions. The traces were collected by two volunteers over 29 different bike runs. In all cases we sample the accelerometer at 50 Hz¹. Table 3 shows the difference between the software-based sensor and the hardware cadence sensor. The first row represents the relative error with respect to the total revolutions of the hardware sensor over a single trip, while the second row corresponds to the error per kilometer. We note that both the mean error and standard deviation are extremely low which means the error we introduce by replacing the physical cadence sensor with the software method will be negligible.

4.2 Elevation Services

Elevation Service	R	RMS (m)
USGS	0.9993	0.9
USGS fitted	0.9995	0.7
USGS fitted & smoothed	0.9997	0.6
Google	0.9957	2.4
Google fitted	0.9958	2.4
Google fitted & smoothed	0.9960	2.3
GPS	0.9540	39

Table 4. Correlation coefficient (R) and Root Mean Square Error (RMS) of different elevation services when compared against elevation from barometric pressure sensor.

In order to compare the different elevation methods we collected pressure measurements and GPS coordinates from 15 trips on 12 different routes throughout March and April of 2012. The barometric pressure and GPS receiver were sampled at 1 Hz and we collected a total of 4,780 GPS and pressure sample pairs.

Table 4 shows the correlation coefficient, R, and the Root Mean Square (RMS) error between the barometric elevation and different elevation services. As expected, both map fitting and outlier smoothing increase the correlation coefficient and decrease the error. The reason why fitting works so well lies in the distance between the raw and fitted GPS coordinates, which on average is 4 ± 3 meters. On a flat terrain, this difference has no impact on the elevation lookups, but in an area with varying terrain this uncertainty can result in a large elevation error even when the value reported by

¹In Android, we used the rate suitable for games, i.e., `SENSOR_DELAY_GAME`, which is approx. 50 Hz.

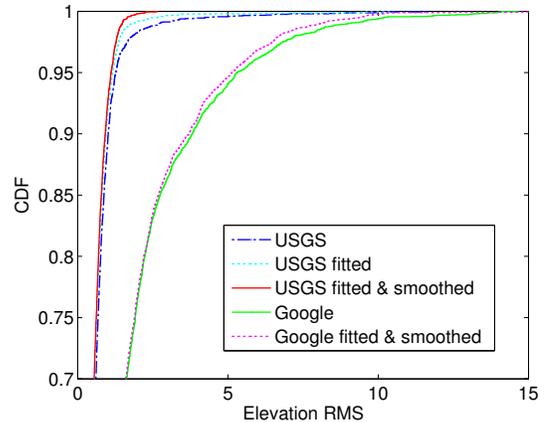


Figure 5. CDF of RMS from the USGS and Google elevation services. While smoothing and fitting benefit the accuracy of the data derived from both services, the USGS service consistently outperforms the Google service.

the elevation service is accurate. For instance, the RMS for the elevation over the SMD route (cf. Table 2) decreases by 40% by fitting the GPS coordinates to the road, while the corresponding decrease on the R1 route is only 10% since the former route lies along a creek while the latter goes through a flat residential area.

Figure 5 presents the CDF of the RMS errors for the USGS and Google Maps services. Regardless of the post-processing applied, the USGS elevation service is clearly superior to Google Maps. After fitting and smoothing, 95% of USGS’s RMS are less than 1.2 m, while 95% of Google Maps’ RMS are less than 5.4 m.

In order to verify that the USGS elevation service is indeed the best to use we perform Welch’s *t*-test [39]. The *p*-values for the fitted and smoothed USGS, Google Maps, and GPS elevation data sets were 0.9606, 0.2301, and 0, respectively thereby confirming our hypothesis.

4.3 Caloric Expenditure Estimation for a Single Biker

We start our caloric expenditure evaluation with a single bike and biker to better illustrate the different methods and introduce the terminology. The next section expands the evaluation to multiple bikers on multiple bikes.

The biker traveled 14 trips with his bike, totaling 24.8 km on all the different types of routes in the evaluation. The data collected from these trips are used to estimate the caloric expenditure using the following approaches:

- *Search Table (TAB)*: we add up all the distances between neighboring fitted GPS samples and use this sum as the trip distance. Together with the trip duration and the biker’s weight, we can easily get a calorie estimation by looking up these parameters at a search table similar to Table 1.
- *Cadence and Speed Sensing (CAD)*: we calculate the RPM during each trip from the software cadence sensor in the pocket and average the speed over every minute using the GPS samples. We then apply (1) to find the

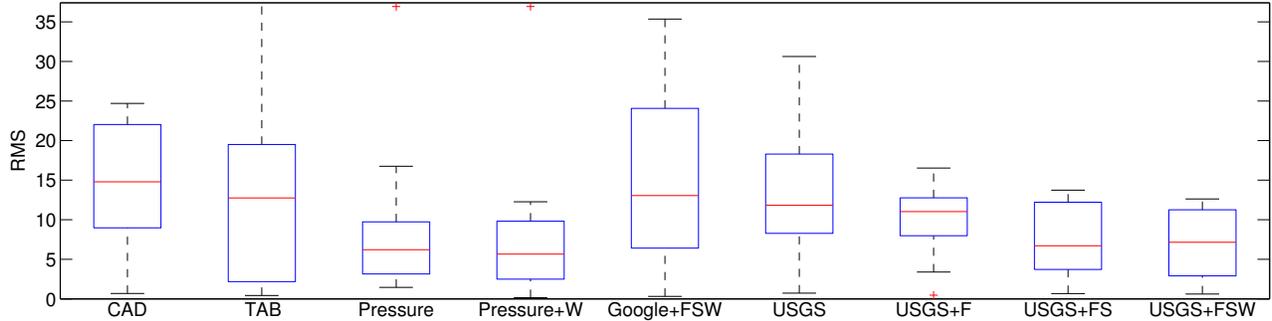


Figure 6. Overall RMS error of different caloric expenditure estimation methods using a heart rate monitoring method as ground truth.

estimated caloric expenditure.

- *Power measurement*: we calculate the power required each second by using the speed from the GPS, finding the road slope using the elevation from neighboring GPS samples, and the current weather conditions (temperature, wind speed, and direction) from the Weather Underground weather station. We note that all bikes used throughout the evaluation have been calibrated before traveling.
- *Heart rate monitoring as ground truth*: we follow the method mentioned in [28] using BPM to calculate calories, which only accounts for calories when the BPM is greater than 90 using (3). The single biker’s VO_2 max is estimated by a 12-minute Cooper test.

Throughout the evaluation, we use TAB as shorthand for the search table method and CAD for cadence and speed sensing. For the power measurement method, the Pressure tag means we used the barometric pressure for elevation data, USGS for USGS’s elevation service, and Google for Google Maps’ elevation service. Furthermore, +F and +S mean fitted location and smoothed elevation, respectively. +W stands for including weather information, i.e., temperature, wind speed, and direction.

Figure 6 compares the nine different caloric expenditure methods and shows the overall RMS errors for each method. The figure shows that with precise elevation and weather information, the power measurement approach can achieve much higher accuracy than both the search table and cadence methods. Likewise, the fitted and smoothed USGS elevation data performs on par with the barometric pressure elevation, although the variance of the former is slightly higher than the latter. On the other hand, using the fitted and smoothed Google Maps elevation data does not perform well due to the inherent error and low resolution of the underlying service. Interestingly, we note that the fitting and smoothing steps both decrease the error when applied sequentially showing that both processing steps are important.

As we will show later, the reason why TAB has much higher variance than USGS+FSW is because it overestimates calories during downhill stretches and underestimates calories during uphill segments. CAD, on the other hand, performs worse than TAB during uphill stretches but works well in most downhill trips.

It appears from this figure that including wind information does not improve the accuracy of the power measurement method at all. The reasons are two fold: first, the wind speed during these trips is low (the Beaufort number is ≤ 3 , i.e., gentle breeze); second, if we look at the trips passing through SMD (including SMDN, SMDS, and SMDC), we see that the wind correction actually increases the error slightly instead of decreasing it. This is caused by SMD being in a low-lying river valley with both sides of the road covered by tall trees, effectively reducing wind significantly. For trips that do not pass through SMD the RMS error decreases by 13% when we apply the wind correction. Last, we do not include the power measurement achieved by using the GPS elevation directly since the RMS errors of this method are an order of magnitude larger.

4.4 Caloric Expenditure Estimation for Multiple Bikers

To evaluate the scalability of our method on different bikes and bikers, we recruited 17 male and 3 female healthy student volunteers from our university. Their ages are between 24 and 32 and the weights from 110 to 175 lbs. In total, 7 bikes are calibrated and used in our evaluation including 3 road bikes, 4 cruiser bikes, and 1 mountain bike. The coefficient of rolling resistance C_r on these bikes ranges from 0.07 to 0.21. We set C_a to 0.26 for all bikes and bikers, as explained in Section 3.4, due to the average bike quality and our bikers’ casual clothing. During our multi-biker evaluation, all volunteers completed at least three trips, and in total 70 trips were collected during one week. Volunteers wear a heart rate strap on their chest and a smartphone in their pocket. To enable heart rate based caloric expenditure estimation as the ground truth, we also need the 12-minute Cooper test for every biker. However, since only a few of them were able to provide a precise number, we used approximate values based on each individual’s physical shape using the standard values from the Cooper Test [4] for those particular volunteers.

We first demonstrate the evaluation results from some typical routes. Starting with Route DL — a circular bike lane around Druid Lake. The total length of this route is 2.4 km. We collected 10 trips from 7 bikers on this route on different days. As Figure 7 shows, even on a flat route like this one the errors from CAD and TAB are more than three times

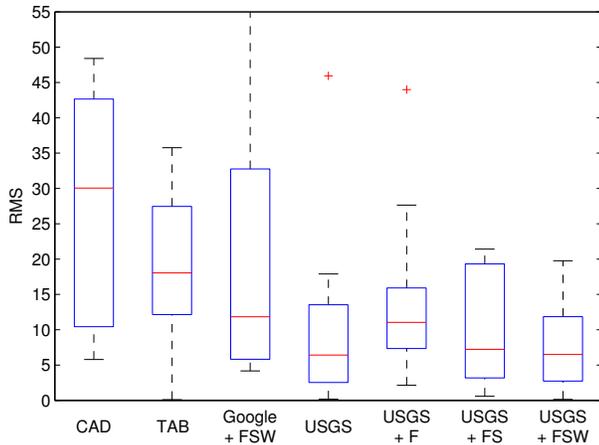


Figure 7. RMS error of different caloric expenditure estimation methods on Route DL — a flat and open field. Wind correction limits the error within a short range.

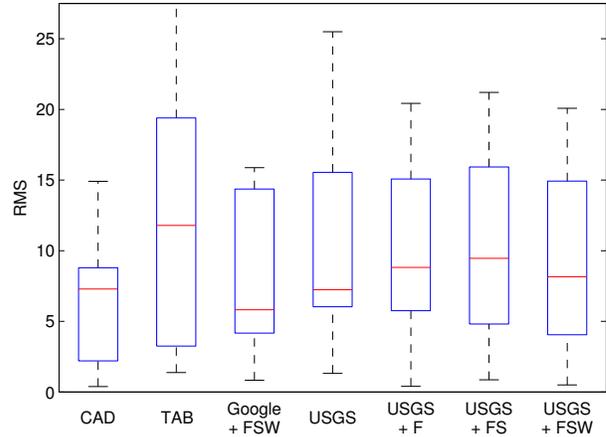


Figure 9. RMS error of different caloric expenditure estimation methods on Route R6 — a downhill path. CAD works well on downhill routes. Figure 8 and 9 shows our method is adaptive to both downhill and uphill routes.

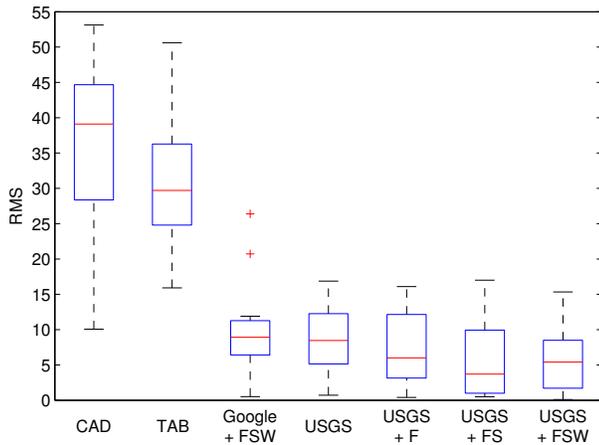


Figure 8. RMS error of different caloric expenditure estimation methods on Route R1 — an uphill path. Power measurement can estimate calories burned on uphill routes accurately while both CAD and TAB underestimate the caloric expenditure uphill.

higher than USGS+FSW. Google does not perform well because of the low elevation resolution, causing the elevation to be averaged with the lake.

The figure also shows that the smoothing process recovers the outliers by using USGS only. In addition, even if the wind is not strong during all the trips, the wind correction decreases the variance compared to USGS+FS. So, even on flat terrain, the power-based caloric expenditure method achieves higher accuracy than both CAD and TAB.

Next, Figure 8 shows the same comparison but on an uphill route — Route R1, a 1.8 km one-direction route with a bike lane through a residential area. The elevation difference between the start and the end is 36 meters. We performed 12 trips with 10 bikers on this particular route. The figure shows that both CAD and TAB fail to provide an accurate caloric expenditure estimation for uphill trips, especially CAD. Dur-

ing uphill trips, our bikers can only maintain a low RPM and speed, which CAD erroneously treats as a low caloric expenditure situation. In this particular case, Google+FSW has a similar accuracy as USGS+FSW, because the slope of the surrounding area is similar to the road, which means the elevation average from Google Maps is close to the USGS one.

To test the opposite case, Route R6 is the downhill route parallel to R1. Eight bikers exercised this route over a total of 11 trips. From the results shown in Figure 9, we see that although CAD performs best, the power measurement methods are only slightly worse, illustrating their ability to adapt to both uphill and downhill stretches.

Figure 10 shows the results from all the trips passing the SMD which is a winding road along a river valley. The elevation difference between the two sides of the road can be as much as 10 meters due to a steep slope at the edge of the road. We have 11 trips crossing SMD across 8 bikers. Since the surrounding terrain of the route is complicated and always changing, we see that direct use of the USGS data results in large errors. However, our fitting method eliminates most of the errors in this situation. On the other hand, Google elevation still does not perform well on this route. TAB performs surprisingly well because bikers ride both uphill and downhill on this route resulting in any over- and underestimation errors canceling each other out.

Figure 11 shows the results from all the trips passing the route between our campus and Druid Lake. This 1.7 km route passes two bridges, one is 67 meters long and the other is 140 meters long, with a height difference of 15 meters between the road and the bottom of the bridge. For this route we have 10 trips driven by 8 bikers. Obviously, without smoothing, the USGS does not perform well as the figure shows. Fortunately, our smoothing method corrects most of the elevation errors without adding additional ones.

Last, Figure 12 shows the overall RMS error for all 70 trips, for each caloric expenditure estimation method. We find that averaged across all the different bikers and all the

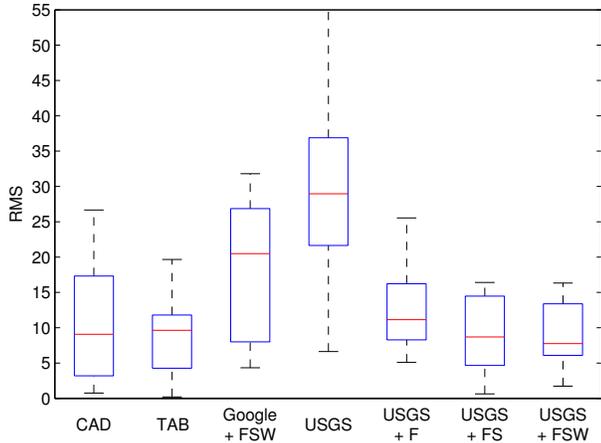


Figure 10. RMS error of different caloric expenditure estimation methods on Route SMD — winding route. Results show our fitting method on USGS works well. However, Google Maps’ elevation API does not perform well on complicated and changing terrain because of its inherent error and low resolution.

different routes, USGS+FSW achieves the lowest error. Especially when compared to Figure 6, the single biker case, there is no significant increase in the RMS for USGS+FSW while all the other methods have higher medians and variance. This illustrates the strength of our calibration since our method is able to adapt to a wide range of bikes and bikers, which none of the other methods are capable of.

4.5 Reducing GPS Power Consumption

Sampling Interval (s)	Interpolation (RMS)	EnAcq (RMS)	Power (%)
0	28.6	28.6	100
5	31.6	28.5	100
10	34.7	27.8	100
15	36.7	30.5	83
20	42.6	29.8	65
25	55.0	31.1	54
30	72.1	34.4	43

Table 5. Error and power consumption for different GPS sampling intervals and route reconstruction methods.

In order to increase the usefulness of our application and decrease the power it consumes on the user’s smartphone, we investigate the impact of duty-cycling the GPS receiver on caloric expenditure accuracy.

Considering that GPS duty-cycling has been studied extensively and is not directly related to our main topic, we only consider two extreme cases: (1) the most simplistic approach in which we reconstruct the missing bike route points by interpolating between the known points, and (2) we simulate the effect of applying a state-of-the-art map matching, route reconstruction algorithm [27]. In both cases we compare the calorie estimation errors with the non-duty-cycled case. We use the original *RI-6* datasets as ground truth and

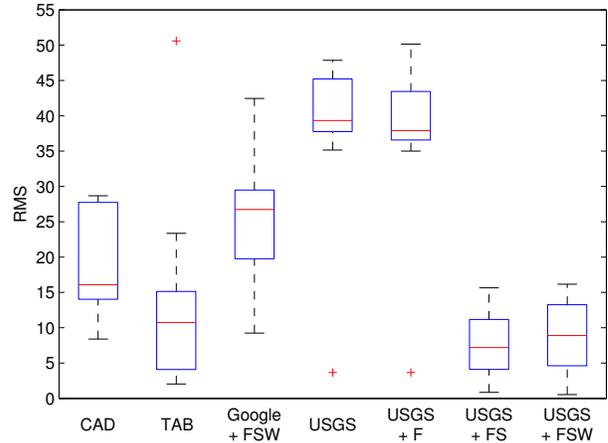


Figure 11. RMS error of different caloric expenditure estimation methods on Route W — a route crossing two bridges. Our smoothing algorithm successfully corrects the “bridge error” in this extreme case and ensures that the power measurement based method works properly.

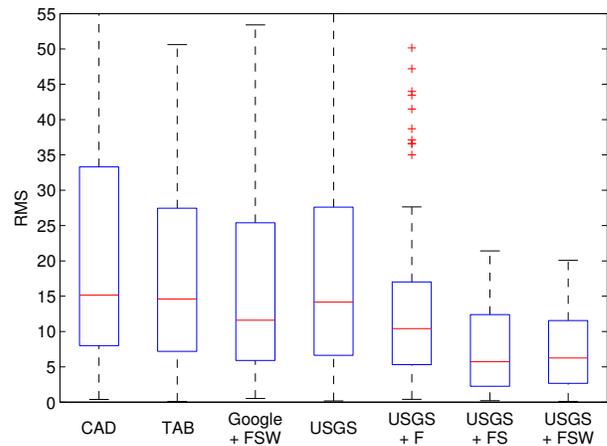


Figure 12. RMS error of different caloric expenditure estimation methods using heart rate monitoring as ground truth, across all 70 bike trips.

down-sample test sets for both algorithms. Figures 13(a)-13(c) show three examples of such down-sampled datasets and the interpolated speed. As the figures show, the longer the sampling interval the more route features are lost.

We use the EnAcq algorithm by Fang et al. [27] as the state-of-the-art route reconstruction mechanism. This algorithm reconstructs the route between two consecutive GPS samples by matching the most likely route taken given the available roads on a map and snaps the position to the closest point on this road. To detect turns, acceleration and compass readings are fed into a Hidden Markov Model and then used to select the most probable direction to take. When the selected road later leads to an improbable route the system automatically backtracks and tries the next most probable alternative. Given the route reconstructed by EnAcq we proceed as described in Section 2 with the online weather and

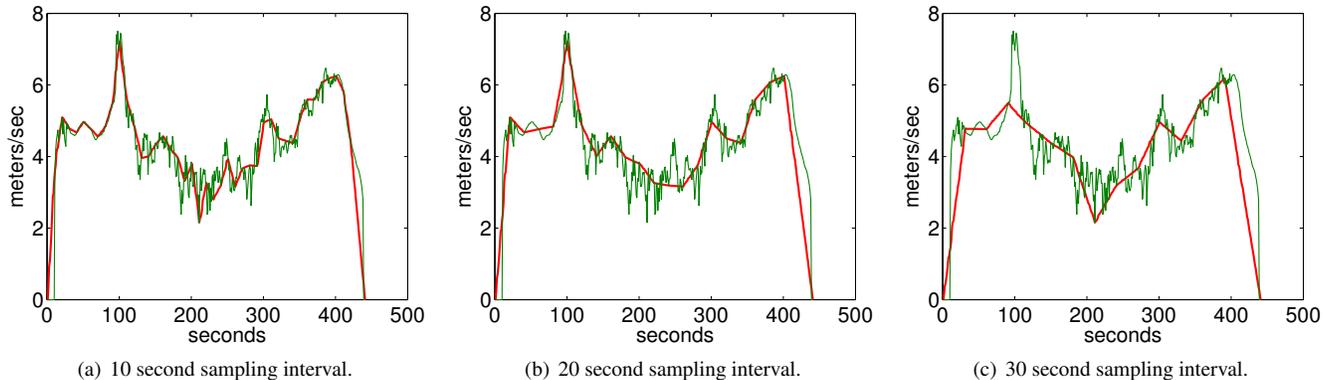


Figure 13. Increasing loss of speed accuracy as the interval between GPS samples increases.

elevation lookup. For the missing speeds, we use the average speed between the two GPS samples that we have and apply them to the reconstructed route between these two points.

Table 5 shows the error and power consumption for the different sampling intervals s as a percentage of the non-duty-cycled case ($s = 0$). We use the PowerTutor smartphone application [40] to measure the relative power consumption across the different sampling intervals. Since the Android API does not allow direct GPS instrumentation, but handles duty cycling itself, it is not possible to turn on the GPS for less than 10 seconds at a time. This is why the power column shows no reduction in power consumption until the sampling interval increases to 15 seconds. The largest GPS sampling interval that does not significantly increase the error of the naive route interpolation method is $s = 15$ s, leading to a potential reduction in power consumption by 17%. There is no appreciable difference in error when we use the EnAcq reconstructed route for calorie estimation. Since Fang et al. report that EnAcq can successfully reconstruct routes with less than 4% mismatch from 30 second sample intervals we could potentially achieve a 57% power reduction by adopting the EnAcq route reconstruction scheme.

5 Discussion

In the first half of this section we discuss the feasibility of implementing our method as a real-time smartphone app. The second half presents the lessons that we learned from this work.

5.1 Towards a smartphone application

Although our evaluation is based on offline data analysis, it is feasible to implement the cadence sensing and power measurement approaches in a real-time bicycling app. Such an app will offer higher accuracy than existing smartphone-only caloric expenditure calculators at zero cost for most smartphone users. When a biker wants to track a trip, he can open the tracking app and click start. If he just wants to track his caloric expenditure, he can mount his phone on his bike’s handlebar, or anywhere he wants, e.g. in the backpack. Since the caloric expenditure calculation is based on accumulating the caloric expenditure during every time unit, the application can show current raw calorie consumption on the screen or by voice. In addition, if he also needs to track his cadence in RPM, he only needs to click a button to enable the cadence

sensing function and put the phone in his front pocket. When reaching his destination, the biker can stop tracking his route and upload the raw trace to the server. In power saving mode, the application can choose a low frequent sampling rate on GPS, and the elevation fitting and smoothing can be done on the server. Finally, the server can provide an accurate caloric expenditure estimation for the biker.

Ideally, calibration only needs to be done once when a biker first uses this application or changes his bike. Even better, with more statistical results on the coefficient of rolling resistance C_r and the lumped constant for aerodynamic drag C_a , auto-calibration can be setup for easier use. For example, a biker inputs his weight, height, and the bike model, the application can setup these parameters for him.

Last, with such an easy way to accurately track bikers’ caloric expenditure, more interesting functions can be enabled. For instance, with the collection of bikers traces on a server, the application can provide caloric expenditure prediction for a particular route in the database; in turn, with a calorie input, e.g., 500 Kcal, the application can calculate the best route for the biker.

5.2 Lessons Learned

As Figure 2 illustrates, we started this work with modular design that included multiple on-bike and on-body sensors to estimate caloric expenditure during biking. Nevertheless, the evaluation process proved that just using a smartphone provides comparable accuracy to the best method that uses external sensors. This has been possible by combining the phone’s sensors (accelerometer and GPS), its high speed Internet access, and Web accessible databases. We believe this shift from physical to *virtual* or *software* sensors will find other applications in quantifying people’s daily lives and activities.

6 Related Work

As we pointed out in the introduction, more and more people have started to ride their bikes which has lead to research in sensor networks on and for bikes. Similar to this work, the primary focus of this line of research has been on attaching sensors to the bike and collecting measurements from bike rides.

One of these data collection networks, BikeNet by Eisenman et al. [25, 26] used TMote Invent and Nokia N80 smart-

phones to collect samples from a broad range of sensors, including accelerometers for tilt measurements and reed relays for cadence and wheel rotation counts. Although BikeNet also collected GPS samples, they were not used to estimate elevation but rather for route tracking. Moreover, BikeNet was primarily designed for data collection from a broad range of sensors with minimal analysis of each modality. Our goal on the other hand is to sample as few sensors as possible and yet still estimate caloric expenditure accurately.

Lu et al. proposed the Jigsaw sensing engine for mobile phone applications [31]. Jigsaw continuously monitors and classifies user activity and infers context. Activities such as walking, cycling, running, etc. can be inferred using the accelerometer. Unlike Jigsaw, we do not seek to infer activities such as biking, but rather quantify the physical aspects of the biking activity, i.e., cadence and calories burned. To the best of our knowledge, our approach is the first to infer cadence from accelerometer data.

Thepvilojanapong et al. mounted an Android phone on the bike's handlebar and collected data from the phone's accelerometer and magnetometer sensors in addition to the GPS receiver [37]. Based on these data, they developed a Hidden Markov Model to recognize the biking states such as going straight or turning left. While we use the GPS receiver to estimate the route bikers took, their work could be used as an alternative method for tracking the bike's route with the added benefit of conserving energy.

Our route based approach is similar to Biketastic by Reddy et al. [35] and BISCAY by Sugo et al. [36]. The former uses accelerometers and microphones mounted on bikes to gauge the roughness of a route, while the latter uses gyroscopes to estimate the comfort of a ride based on the smoothness of the turns. Both papers use the collected data to infer the relative and qualitative characteristics of a bike route; our goal is to provide an absolute and quantitative caloric estimate that rivals heart rate monitors and surpasses simpler empirical formulas.

Several methods have been proposed to estimate caloric expenditure based on sensor measurements. Vyas et al. use a regression model to find a correlation between the measured skin temperature and conductivity, heat flux, ambient temperature, and acceleration to energy expenditure [38]. Liu et al. use Support Vector Machines to classify acceleration and respiration measurements [30]. However, both of these methods require bikers to wear hardware sensors that are as, if not more, intrusive than the chest-worn heart rate monitors described in Section 2.

The work closest to ours, to the best of our knowledge, is by Lester et al. [29]. The authors use a smart phone to collect acceleration measurements and GPS samples while walking and running. They also evaluate the effect of different elevation methods based on the route taken, but do not correct for the elevation discrepancies reported by elevation services such the one provided by the USGS. Moreover, they rely on an empirical formula that does not take wind into account. On the other hand, we use a physics-based model with individually calibrated constants that also takes aerodynamic drag into account. The results presented in Section 4 show that discounting the wind's contribution can lead to signifi-

cant errors in certain cases.

7 Conclusion

Biking is one of the most effective and environmental friendly exercises for improving one's health and fitness. On-bike sensors and computers, although expensive, can significantly improve the overall biking experience. For example, knowing the exact number of calories burned during each bike trip provides a positive feedback loop for the biker who immediately sees the rewards from exercising and thereby schedules future bike trips. Likewise, knowing the biking power in real time helps bikers remain in their target heart rate zones for best training results. Finally, knowing their cadence allows bikers to stay in a safe and efficient RPM range to protect their knees, which is especially important for long rides.

In this work, we systematically evaluated whether bikers can get all this information by using only a smartphone, carried in their pant pockets. Extensive experimental results from 20 bikers over 70 bike trips confirmed that the smartphone can reliably measure the cadence with the on-board accelerometer, and accurately calculate the caloric expenditure by combining the GPS data, the USGS elevation service, and the detailed road database from OpenStreetMap.

With the rapidly increasing popularity of smartphones, this work immediately gives millions of bikers a zero-cost solution towards significantly improved biking experience, and hopefully a higher quality of life in the long run.

Acknowledgments

We are grateful to all the volunteers that participated in the biking tests; this paper would not have been possible without them. We would like to thank our shepherd Vijay Raghunathan and the anonymous reviewers for helping us improve this paper. This work is partially supported by the National Science Foundation under project #0855191 and by Google through a generous equipment gift.

8 References

- [1] Android Sensor API. <http://developer.android.com/reference/android/hardware/Sensor.html>.
- [2] Bicycle performance. http://en.wikipedia.org/wiki/Bicycle_performance.
- [3] BMP085 digital pressure sensor. <http://www.bosch-sensortec.com/>.
- [4] Cooper Test. http://en.wikipedia.org/wiki/Cooper_test.
- [5] DailyBurn. <http://www.dailyburn.com/>.
- [6] Endomondo. <http://www.endomondo.com/>.
- [7] FatSecret. <http://www.fatsecret.com/>.
- [8] Filtering and Smoothing Data, Matlab. <http://www.mathworks.com/help/toolbox/curvefit/>.
- [9] MapMyFITNESS. <http://www.mapmyfitness.com/>.
- [10] Nonin GO₂ Achieve Fingertip Pulse Oximeter. <http://www.buynonin.com/go2nonin/productdetail.htm>.
- [11] OpenStreetMap, the free wiki world map. <http://www.openstreetmap.org>.
- [12] Sportline Solo 915 Heart Rate Monitor. <http://www.sportline.com/>.
- [13] SRAM Powermeter crankset. <http://www.sram.com>.
- [14] State of Wisconsin Department of Health and Family Services: Calories Burned Per Hour.
- [15] The Google Elevation API. <https://developers.google.com/maps/documentation/elevation/>.

- [16] The National Cycling and Walking Study: 15-Year Status Report. Pedestrian and Bicycle Information Center (PBIC), Federal Highway Administration (FHWA).
- [17] Twenty-Three Tips for the Best Cycling. <http://www.wellnessletter.com/html/fw/fwFit06Cycling.html>.
- [18] U.S. Geological Survey Elevation Service. <http://www.usgs.gov/>.
- [19] Zephyr BioHarness BT. <http://www.zephyr-technology.com/>.
- [20] Harvard Heart Letter, Calories Burned in 30 Minutes for People of Three Weights, July 2004.
- [21] A. Ahtinen, J. Mantjarvi, and J. Hakkila. Using heart rate monitors for personal wellness - The user experience perspective. In *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2008 (EMBS '08)*, pages 1591–1597, Aug. 2008.
- [22] M. H. Al-Haboubi. Modelling energy expenditure during cycling. *Ergonomics*, 42:3:416–427, 1999.
- [23] K. H. Cooper, T. C. Cooper, and W. Proctor. *Start Strong, Finish Strong: Prescriptions for a Lifetime of Great Health*. Penguin, Bloomington, IN, USA, 2008.
- [24] CTC - the UK's National Cyclists's Organization. <http://www.ctc.org.uk/DesktopDefault.aspx?TabID=3523>.
- [25] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. The BikeNet mobile sensing system for cyclist experience mapping. In *Proceedings of the 5th ACM Conference on Embedded networked sensor systems (SenSys '07)*, pages 87–101, New York, NY, USA, 2007.
- [26] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Trans. Sen. Netw.*, 6(1), Jan. 2010.
- [27] S. Fang and R. Zimmermann. EnAcq: energy-efficient GPS trajectory data acquisition based on improved map matching. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11)*, pages 221–230, New York, NY, USA, 2011.
- [28] L. Keytel, J. Goedecke, T. Noakes, H. Hiiloskorpi, R. Laukkanen, L. van der Merwe, and E. Lambert. Prediction of energy expenditure from heart rate monitoring during submaximal exercise. *Journal of Sports Sciences*, 23:289–297, 2005.
- [29] J. Lester, C. Hartung, L. Pina, R. Libby, G. Borriello, and G. Duncan. Validated caloric expenditure estimation using a single body-worn sensor. In *Proceedings of the 11th ACM International Conference on Ubiquitous computing, (Ubicomp '09)*, pages 225–234, New York, NY, USA, 2009.
- [30] S. Liu, R. Gao, D. John, J. Staudenmayer, and P. Freedson. Multisensor Data Fusion for Physical Activity Assessment. *IEEE Transactions on Biomedical Engineering*, 59(3):687–696, March 2012.
- [31] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, pages 71–84, New York, NY, USA, 2010.
- [32] J. C. Martin, D. L. Milliken, J. E. Cobb, K. L. McFadden, and A. R. Coggan. Validation of a Mathematical Model for Road Cycling Power. *Journal of Applied Biomechanics*, 14:276–291, 1998.
- [33] P. Pietro. Cycling on Earth, in Space, on the Moon. *European Journal of Applied Physiology*, 82:345, 2000.
- [34] J. Pucher, R. Buehler, and M. Seinen. Bicycling renaissance in North America? An update and re-appraisal of cycling trends and policies. *Transportation Research Part A*, 45:451–475, 2011.
- [35] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, and M. Srivastava. Biketastic: sensing and mapping for better biking. In *Proceedings of the ACM 28th international conference on Human factors in computing systems (CHI '10)*, pages 1817–1820, New York, NY, USA, 2010.
- [36] K. Sugo, M. Miyazaki, S. Konomi, M. Iwai, and Y. Tobe. BIS-CAY: Extracting Riding Context from Bike Ride Data. *Database Systems for Advanced Applications, Lecture Notes in Computer Science*, 5982:408–411.
- [37] N. Thepvilajanapong, K. Sugo, Y. Namiki, and Y. Tobe. Recognizing bicycling states with hmm based on accelerometer and magnetometer data. In *Proceedings of SICE Annual Conference (SICE '11)*, pages 831–832, Sept. 2011.
- [38] N. Vyas, J. Farrington, D. Andre, and J. I. Stivorc. Machine Learning and Sensor Fusion for Estimating Continuous Energy Expenditure. 2011.
- [39] R. E. Walpole, R. H. Myers, and S. L. Myers. *Probability and Statistics for Engineers and Scientists*. Pearson Education, 2002.
- [40] L. Zhang, B. Tiwana, R. Dick, Z. Qian, Z. Mao, Z. Wang, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *2010 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS '10)*, pages 105–114, Oct. 2010.