RecoFit: Using a Wearable Sensor to Find, Recognize, and Count Repetitive Exercises

Dan Morris, T. Scott Saponas, Andrew Guillory, Ilya Kelner
Microsoft Research
{dan,ssaponas,ikelner}@microsoft.com, guillory@cs.washington.edu

ABSTRACT
Although numerous devices exist to track and share exercise routines based on running and walking, these devices offer limited functionality for strength-training exercises. We introduce RecoFit, a system for automatically tracking repetitive exercises – such as weight training and calisthenics – via an arm-worn inertial sensor. Our goal is to provide real-time and post-workout feedback, with no user-specific training and no intervention during a workout. Toward this end, we address three challenges: (1) *segmenting* exercise from intermittent non-exercise periods, (2) *recognizing* which exercise is being performed, and (3) *counting* repetitions. We present cross-validation results on our training data and results from a study assessing the final system, totaling 114 participants over 146 sessions. We achieve precision and recall greater than 95% in identifying exercise periods, recognition of 99%, 98%, and 96% on circuits of 4, 7, and 13 exercises respectively, and counting that is accurate to ±1 repetition 93% of the time. These results suggest that our approach enables a new category of fitness tracking devices.

Author Keywords
Inertial sensors; machine learning; fitness

ACM Classification Keywords
H.5.2 User Interfaces: Input devices and strategies

INTRODUCTION
Regular exercise offers numerous health benefits, including improved cardiovascular fitness and a reduction in the risk of obesity [15,16,27]. Furthermore, regular exercise offers entertainment and social value, and has been correlated with improved cognitive and emotional well-being [11,13,34].

Despite growing awareness of these benefits, maintaining or expanding a regular exercise regimen is challenging [20,30]. Consequently, most individuals do not maintain recommended levels of activity [5]. Fortunately, research has demonstrated that automatically tracking exercise can motivate activity, particularly in the case of pedometry, which has been widely deployed and extensively studied [4,7,23].

The consumer electronics market has recognized this opportunity: devices already exist to track several fitness activities. Pedometers and GPS devices, for example, primarily target walking and running. Console accessories, including the Microsoft Xbox Kinect and the Nintendo Wii Fit, target indoor workouts that are closely tied to a stationary display. High-end cardio machines (e.g., stationary bikes, elliptical, etc.) can send workout summaries to social media or to a phone.

However, this landscape of devices misses two major categories of fitness activities: weight training and calisthenics. Here we use "calisthenics" to refer to strength-training exercises that do not necessarily involve weights: sit-ups, push-ups, jumping jacks, etc. For some individuals, these categories of exercise may be more sustainable than walking or running, for lifestyle or preference reasons. Even for those who regularly walk or run, weight training and calisthenics can be a critical part of a balanced exercise program: the Centers for Disease Control recommends muscle-strengthening activities at least twice a week for adults [6], backed by research showing the benefits of muscle-strengthening for weight loss and overall health [20], but compliance with these recommendations is even lower than for aerobic activities [5].

Wearable fitness sensors, such as the Nike FuelBand and FitBit Flex, track caloric expenditure and overall activity and can be used during exercise; however, no wearable device provides high-fidelity information specifically relevant to strength training (i.e., reps, sets, and time; data types that are typically tracked manually). Techniques based on sensors in the environment (e.g. cameras) cannot robustly handle the variety of postures or the complexity of surroundings associated with weight training and calisthenics.

In this work, we aim to bring the benefits of wearable, automatic tracking to strength-training exercises, adding to the growing landscape of technology support for exercise motivation and measurement. Just as a GPS watch offers a runner the ability to "set it and forget it" when starting a workout, we seek to provide real-time feedback and post-workout analysis with no intervention by the user during a workout.

In this paper, we describe and evaluate a novel pipeline for separating exercise from background activity, automatic exercise labeling, and repetition counting.

CHI 2014, April 26–May 1, 2014, Toronto, ON, Canada

ACM 978-1-4503-2473-1/14/04…$15.00.
http://dx.doi.org/10.1145/2556288.2557116

RELATED WORK

Exercise Analysis

The most relevant previous work is that of Muehlbauer et al. [24], who similarly divide the problem into segmentation, recognition, and counting. They achieve 85% segmentation accuracy and 94% recognition accuracy (10 classes) using subject-independent training. We build on this work to improve robustness and accuracy, and we validate RecoFit at a larger scale. Like the present work, their approach to segmentation uses features based on the autocorrelation. However, [24] uses heuristic thresholds for segmentation; RecoFit uses learned segmentation, which we believe is essential to robustness and scalability. [24] identifies sensor placement variation as a key challenge. However, they don't address this algorithmically; RecoFit introduces dimensionality reduction to allow orientation-invariant analysis. RecoFit adds a deeper approach to repetition counting, addressing orientation-invariance and false peak rejection, which results in higher counting accuracy. Finally, [24] doesn't address real-time scenarios; we discuss all stages in the context of an online application, identify latency tradeoffs, and include results for online and offline recognition.

myHealthAssistant [31] classifies gym exercises from three accelerometers (on the hand, arm, and leg), using a Bayesian classifier trained on the mean and variance on each accelerometer axis. They achieve 92% accuracy for 13 exercises, but they use subject-specific training. They address repetition counting with a combination of autocorrelation-based period estimation and peak counting on one of the accelerometer axes; we build on this approach to handle non-axis-aligned movements and more complex temporal patterns (e.g. secondary peaks within repetitions, preparatory movements) that are common in natural exercise behavior. This work does not address segmentation.

Chang et al. [8] use two accelerometers (on the hand and waist), and compare a Hidden Markov Model (HMM) and a Bayes Classifier for exercise recognition. They achieve 95% accuracy for nine exercises using subject-specific training, and 85% accuracy using subject-independent training. They also address counting, using a matched filter and an HMM. This work does not address segmentation.

Gesture Spotting

A significant component of our contribution is exercise *segmentation*, or finding exercise amidst periods of non-exercise. A similar problem exists in gesture recognition: gestures are performed sporadically amidst non-gesture movement. The problem of isolating gestures is typically called "gesture spotting". Here we give an overview of techniques for motion-based gesture spotting, but we refer to [25] for a review of techniques based on computer vision.

Hidden Markov Models (HMMs) are commonly used for gesture recognition from inertial sensors; many approaches perform spotting by thresholding HMM probabilities [10,22]. To overcome the instability of HMM probabilities, [17] uses similarity to known examples as a preprocessing filter, [19] uses boosting to improve threshold models, and [1] proposes online threshold adaptation. Outside of HMMs, other proposed spotting approaches have been based on dynamic time warping [26] and string matching [32].

Autocorrelations and Periodicity

One of our key contributions lies in the use of the autocorrelation function to find regions of self-similar, repetitive exercise. This is inspired by work in other domains that has leveraged the autocorrelation for analyzing other highly periodic signals, e.g. for tracking the pitch of a musical signal [3] or for finding abnormalities in EKG signals [33]. A particularly close analogy to our work is the popular use of the autocorrelation function in speech analysis to separate speech from non-speech audio [2].

Continuous Activity Recognition

Coarse-grained classification of activities from inertial sensors has been widely studied in HCI [9,14] and health care [12]. In general, this work tries to classify typical activities (such as walking, sitting, running, driving, and sleeping), to estimate caloric expenditure and support context-aware computing. Precise boundaries between activities are generally less critical than in the present work. Surveys are available on activity recognition systems [21], healthcare applications [28], and classification techniques [29].

WHY IS AUTOMATIC EXERCISE ANALYSIS HARD?

Our goal is to track exercises from an arm-worn sensor, with no user-specific training, and no intervention from the user during a workout. In this section, we will discuss some of the challenges associated with this goal, motivating the specific algorithmic choices we describe in the next section.

Exercise looks very similar to non-exercise

Perhaps the most challenging problem in this space is that time spent actually exercising within the context of a workout session may be as little as 10% or as much as 100% of the total session. Between exercises, one walks around the workout space, socializes, stretches, rests, drinks water, selects and retrieves equipment, etc. The distinction between these non-exercise periods and actual exercise may be obvious to a human observing video, but to a wearable sensor, these distinctions are much less clear.

Magnitude alone is rarely informative; for high-velocity exercises like jumping jacks, motion magnitude may exceed typical non-exercise magnitude. These are the "easy cases" (Figure 1a), but such activities are the exception: most exercises are actually performed quite slowly, and exercisers often *strive* to avoid jerky movements that would yield high acceleration values. In practice, the amplitude of acceleration during exercise is generally consistent with that during non-exercise (Figure 1b), and non-exercise stretches may significantly *exceed* the magnitude of most exercises.

Some exercises, e.g. pushups, result in almost no translation of an arm-worn sensor, and when performed slowly, the slow rotation of a sensor is very close to the noise floor of most

gyroscopes. In fact, the primary observable phenomenon in these cases is not energy on any sensed axis, but a repetitive change in the gravity axis observed by the accelerometer, a phenomenon not observed at all for a non-rotational exercise like shoulder presses (lifting a weight straight overhead from the shoulders). This diversity in the fundamental phenomena that characterize exercises motivates our use of a machine learning approach to segmentation.

Fortunately, one intuition does allow us to separate most exercise from most non-exercise: *exercise is typically more periodic (i.e., repetitive) than non-exercise*. Consequently, many of our features are based on the *autocorrelation* of our signals, from which we can derive metrics of repetitiveness.

However, this leads to another challenge: *walking* is the most common non-exercise activity performed during a workout. Walking is *extremely* periodic, and is similar in amplitude to many exercises. In fact, it's almost impossible to heuristically describe systematic differences between walking and exercise, which further motivates our use of a machine learning approach to segmentation, with a strong emphasis on walking in our training data collection.

Furthermore, *dynamic stretching* – repetitive movements designed to loosen joints or muscles, but not intended as exercises per se – is quite common during a workout. This presents a tremendous challenge to robust exercise segmentation: separating "exercise" from "dynamic stretching" is almost a question of semantics, but one that significantly impacts user experience. Fortunately, we observed throughout our data collection that it is *extremely* rare for an individual to consistently perform the same dynamic stretching movement – without changing orientation – for more than a few seconds, which supports our use of self-similarity as a core of our feature set, and motivates the temporal smoothing approach we will describe in the next section.

Though challenging, we consider robust segmentation *critical* to the practicality of automatic exercise analysis. At the highest level, false positives (when the system tracks an exercise that the user did not actually perform) or false negatives (when the system fails to "credit" the user for an exercise) are potentially disastrous from a user experience perspective. Furthermore, even when segmentation is "correct", the boundaries of exercises identified by our segmenter need to be precise to enable robust performance at subsequent stages of our pipeline. In particular, reliable counting relies heavily on accurate segmentation to ignore preparatory and post-exercise movements, such as lying down to perform pushups, or putting weights down after biceps curls.

Variability in form

Since we aim to require no user-specific training of our system, variability in users' interpretations of exercise descriptions, and their ability to consistently execute a particular form, has a tremendous impact on recognition accuracy. Even an exercise like pushups, which has a consistent definition to most potential users, exhibits wide variation in arm

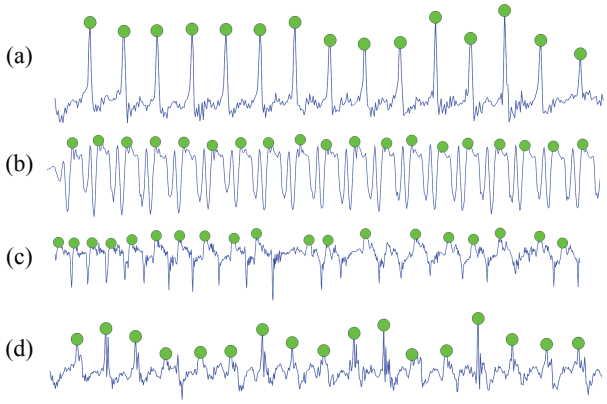

Figure 2: Counting challenges. (a) Consistent large peaks. (b) Each repetition contains multiple similar peaks. (c) Irregular timing and form. (d) Irregular amplitude. Circles indicate repetitions, correctly identified by our algorithm.

posture, pace of repetition, and the temporal "shape" of the movement. And less familiar exercises often exhibit an entirely more challenging level of variability, where users interpret the fundamental form of the exercise differently.

In developing RecoFit, we assumed that end-users would have *some* familiarity with the available exercises, but that users would not typically watch a proscriptive video or have access to a coach who would refine their form. Consequently, we believe there is no way to address the problem of variation in form other than large-scale training data collection with enough flexibility to elicit such variation. Therefore, users in both our training data collection and our evaluation study were given instructions – which they were not required to read – that would simulate "reasonable familiarity", but allow enough interpretation to elicit natural variation. Instructions contained an illustrative image and a high-level description for each exercise, and experimenters did not coach or correct form during data collection.

Temporal Irregularities

Counting exercise repetitions is sometimes straightforward,

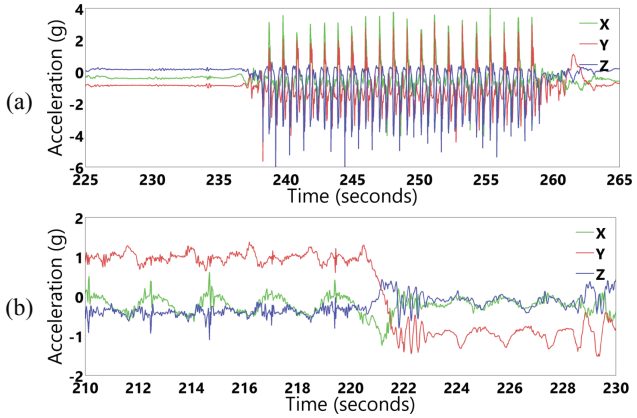

Figure 1: Challenges separating exercise from non-exercise. (a) The (rare) easy case: a participant goes from a relatively motionless state to rapid movement. (b) A typical case: a participant finishes a set, puts weights down, and stretches a bit; exercise and non-exercise amplitude are almost identical.

such that a simple peak-picking or zero-crossing approach will yield an accurate count (Figure 2a). Typically, this is only true of high-amplitude activities like jumping jacks. Figure 2b is a typical set of squats, with two bursts of acceleration per repetition, resulting in a "double peak" for each count, motivating our use of autocorrelation-based period estimation combined with peak counting. Figures 2c and 2d highlight very challenging cases.

Unpredictable device orientation

Finally, with our specific goal of an arm- or wrist-worn inertial sensor, the form factor itself presents a challenge: in real-world use, an arm-worn sensor will naturally rotate differently around different users' arms, as a consequence of preference and natural fit. Therefore, RecoFit assumes that it can "trust" the axis pointing *along* the arm (a watch, for example, always has its face pointed out, in a readable orientation), but that the device might rotate arbitrarily *around* the arm.

TRAINING DATA COLLECTION

The previous section discussed the challenges associated with exercise analysis, motivating a large-scale training data set exhibiting real-world behavioral variability.

Data Collection Hardware

Data was collected from an armband (Figure 3) worn on the right forearm, containing a SparkFun "Razor IMU" inertial sensor, which includes a 3-axis accelerometer and a 3-axis gyroscope. The armband also included a battery and a Bluetooth radio, which transmitted sensor values to a PC at 50Hz.

"Natural" Environment and Procedure Design

We cannot stress enough the importance of encouraging natural variability in training data. Anecdotally, we began this project with a smaller data set (30 participants), collected in a space-constrained laboratory environment that did not aesthetically resemble a gym, with clear instructions regarding sequencing and form. Cross-validation on this data set was unrealistically encouraging: segmentation, recognition, and counting were all close to perfect. We used this data to train our system and deployed a real-time prototype in a more realistic environment, and it quickly became clear that early success was the result of "robotic" behavior among training participants: segmentation dropped to precision/recall levels close to 50% when users were in a more natural environment.

In other words, it became clear that real-world deployment required a data set that was both *larger* and *more natural*. The practicalities of labeling activities and scaling to over 100 participants prevented us from operating in an actual gym, so we retro-fitted a large lab space to resemble a home gym, with appropriate décor (wallpaper, curtains, etc.), video and audio entertainment under participants' control, a couch for rest periods, and no computers or experimenters visible to participants.

The data collection procedure changed as well: participants were given a set of exercises to perform, akin to a workout assigned by a trainer, but were instructed that they could perform them in any order. A wide range of repetitions was suggested for each exercise, but participants were not required to complete any particular number of repetitions. In order to diversify the non-exercise activity in our training data, participants were instructed to take a break of about five minutes somewhere near the middle of the session, during which they could watch TV, check email, etc. Participants were also asked to spend five minutes stretching during the session. Sessions were typically about 45 minutes, and participants had no contact with the experimenter once the session began.

Figure 3: Data collection and evaluation hardware.

We provided instructions that included an illustration and a short textual description of each exercise. In practice, most participants looked at the images and did not carefully read the text. This was consistent with our goal of simulating the "reasonable familiarity" with the exercises that one might have when using a system like RecoFit.

Walking

Although the primary goal was a natural workout, we also needed to address the problem of *walking* specifically. We addressed this to some extent during the workouts themselves, by making the space large enough to facilitate walking, and by positioning water, entertainment, and weights in realistically-spaced positions around the room to encourage movement. However, the proportion of walking data during sessions was not enough to fully address this problem, so at the *end* of approximately 25 of the sessions, participants were asked to "walk around the room" for five minutes.

Labeling

An experimenter watching from another room labeled all exercises and intervening periods of non-exercise – with the type of exercise, the repetition count, and the start/end times of that exercise – using Noldus Observer XT software. Sessions were also video-recorded; missing or uncertain labels were corrected from video post-hoc. Video data was synchronized with the sensor data by having the participant tap on the sensor unit at the end of the session.

It is almost impossible to label most exercise boundaries from video with our desired precision (~200ms). Thus, boundaries of exercise regions were further refined based on visual analysis of the accelerometer traces.

Participants and exercises

94 participants (28 female), ages 18-58 (μ=34.2) from within our institution provided 126 sessions of training data. Sessions averaged 38min, about half of which was "non-exercise" time. Participants were of widely varying fitness levels,

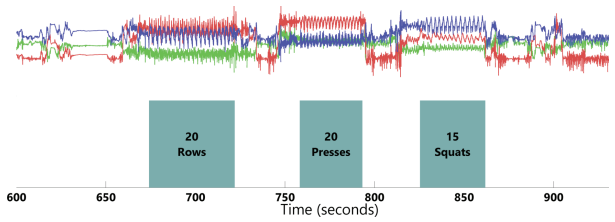

Figure 4: Segmentation overview. The segmenter infers exercise periods (bottom) from accelerometer data (top).

but some self-selection bias toward higher-than-random fitness levels no doubt occurred; all participants were willing to exercise for a study and felt that they could safely complete the exercises. Training data participants were compensated with dining coupons for nearby restaurants.

We collected data for 26 exercises (Table 1), plus walking and stretching. Consequently, no participant could complete all exercises in a single session, and the 126 sessions spanned four lists of about twelve exercises each, with some overlap. The number of collected sets of any exercise ranged from 16 to 47 (some variation was introduced as participants were allowed to skip or repeat exercises for preference).

THE RECOFIT SYSTEM

System Overview

RecoFit's input is 6-axis data at 50Hz, which feeds the *segmentation* stage. The segmenter is essentially a binary state machine that indicates whether the user is exercising. Subsequent stages do not operate during non-exercise periods. When the segmenter detects exercise, it feeds the segmented region to the *recognition* stage, which labels the region with a particular exercise type. The data and the label are then fed to the *counting* stage for repetition counting.

Segmentation

The "segmentation problem" is summarized in Figure 4, which shows an accelerometer trace in the top row; the bottom row indicates when this participant was actually exercising. Our goal is to infer the blue boxes in the bottom row from the raw data in the top row.

Segmentation can be broken down into four stages: preprocessing, feature computation, classification, and aggregation.

Segmentation: Preprocessing

Accelerometer and gyroscope data are smoothed with a Butterworth low-pass filter (-60dB at 20Hz), then windowed into 5-second windows sliding at 200ms (i.e., each 5s window shares 4.8s of data with the previous window).

Segmentation: Feature Computation

Each 5-second window is transformed into 224 features that we use to characterize exercise. These features can be described as 28 features computed over each of 8 one-dimensional "signals" – also five seconds long – computed from the raw accelerometer/gyroscope axes. We will describe those 8 signals, then the 28 features that we compute for each signal. We use the convention that the X axis of our sensor corresponds to the vector along the user's arm; as discussed above, this is the only axis with a known interpretation in physical space, because the sensor may rotate arbitrarily around the arm due to movement or preference.

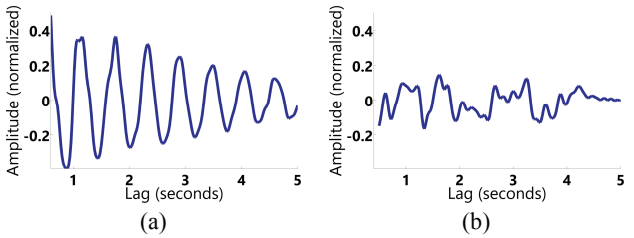

Figure 5: Autocorrelation samples. (a) Autocorrelation of repetitive activity is smooth with prominent peaks. (b) Autocorrelation of non-repetitive activity is jagged with weak peaks.

Segmentation signals:

1) **aX**: the X-axis accelerometer signal
2) **aXmag**: the magnitude of the accelerometer signal at each sample, i.e. sqrt(ax^2+ay^2+az^2).
3) **aPC1**: the projection of the three-dimensional accelerometer signal onto its first principal component. This is the movement along the axis that demonstrates the most variance within this window, or – anecdotally – the most "interesting" rotation of the window.
4) **aYZPC1**: the projection of *only* the Y and Z axes onto the first principal component of those two axes. This captures *movement perpendicular to the arm*, which allows us to derive information from the Y and Z axes despite the unknown rotation of the armband.

Each of these signals is repeated for the gyroscope, yielding eight 5-second signals for each window.

Segmentation features (computed for each signal):

Our segmentation feature set aims to capture the intuition that exercise is usually more repetitive than non-exercise, for which we leverage the *autocorrelation* function, i.e. the cross-correlation of a signal with itself. Each value L in the autocorrelation is the convolution of a signal with a version of itself lagged (shifted) by L samples. If a signal has a strong periodic component at frequency *f*, this will appear as a peak in the autocorrelation at lag 1/*f*. Consequently, the autocorrelation of white noise shows no peaks at all (no periodicity). In practice, autocorrelations of real accelerometer data are somewhere in between, and this set of features is intended to estimate the periodicity of the signal from the autocorrelation. Representative examples of exercise and non-exercise autocorrelations are shown in Figure 5.

We compute the autocorrelation of each signal, normalize it to have value 1 at zero lag, exclude lags less than 0.5s, and compute the following five features:

- **Number of autocorrelation peaks** (note that this feature does not vary monotonically with repetitiveness; too many peaks is indicative of irregular movement, but too few is also consistent with a noisy signal)

- **Prominent peaks**: The number of autocorrelation peaks that are larger than their neighboring peaks by a threshold and are more than a threshold lag away from their neighboring peaks. In general, more prominent peaks are present in exercise than non-exercise.
- **Weak peaks**: The number of autocorrelation peaks that are within a threshold height of their neighboring peaks and are less than a threshold lag away from their neighboring peaks. In general, more weak peaks are present in non-exercise than exercise.
- **Maximum autocorrelation value**: In general, a higher maximum value (other than the initial peak at zero lag, which we do not consider) is consistent with exercise.
- **Height of the first autocorrelation peak *after* a zero-crossing**.

We also compute 15 features not derived via autocorrelation:

- **RMS**: The root-mean-square amplitude of the signal. The intuition here is that faster motion is more likely to correspond to exercise than non-exercise.
- **Power bands**: The magnitude of the power spectrum in 10 bands spaced linearly from 0.1-25Hz (10 features).
- **Mean, standard deviation, and variance (3 features)**.
- **Integrated RMS**: The root-mean-square amplitude of the signal after cumulative summation (which roughly takes us from "acceleration" space to "velocity" space).

Finally, in order to improve the precision of our segmentation boundaries, we compute the RMS, mean, standard deviation, and variance for the *first half* and *second half* of each window. This results in a smoother transition in probability during windows that overlap the beginning or end of exercise, and a stronger tendency for this probability to cross a reliable threshold just at the exercise boundary. This yields 8 features, bringing the total number to 28 features per signal.

Segmentation: Classification

Every 5-second window (200ms step) in our training data is labeled "exercise" or "non-exercise" from the ground truth information (walking is considered "non-exercise"). For each window, we compute these 224 features, and the resulting feature matrix is used to train an L2 linear support vector machine (SVM) [18], which predicts either "exercise" or "non-exercise" for each 5-second window.

Segmentation: Aggregation

Even if the classifier is performing well, it is common for predictions to disagree with the actual exercise/non-exercise state for short periods, e.g. for brief dynamic stretches during non-exercise, or pauses in the middle of an exercise. The output of the system should not reflect these brief transitions, as they do not match a user's model of periods of exercise and non-exercise. The "aggregation" step smoothes these inconsistencies to produce a coherent set of exercise periods.

Inevitably, this involves trading some amount of latency for robustness to these minor inconsistencies. We cannot start counting repetitions immediately after the first prediction of "exercise", since it may well be that the aggregation stage discovers this was not really the beginning of an exercise. Our assumption is that falsely starting counting would be disruptive to any user experience. This is not an issue for offline operation (e.g. post-workout review).

We employ an "accumulator" strategy for aggregation. The system starts in the non-exercise state, and every time we see a prediction of "exercise", we increment a value (the *accumulator*). Each time we see a prediction of non-exercise, we decrement the accumulator. When the accumulator reaches a threshold, the system enters the "exercise" state. This threshold is set to the equivalent of 6 seconds. We maintain a history buffer, so that when the system enters the "exercise" state, it can provide the entire window of exercise to the recognition and counting systems. This process is simply performed in reverse to detect the end of an exercise.

Recognition

Once the segmenter has identified a period of exercise, the recognition stage assigns a label to that exercise from a set of possible exercises, or "circuit". A circuit could be any subset of the 26 exercises for which we have training data, though accuracy declines with large circuits, and it's extremely rare for a workout program to give an exerciser 26 options at any one time. Consequently, practical use would likely involve circuits of four to eight exercises.

Recognition: Preprocessing

RecoFit's recognition stage operates over the same filtered data windows described above. There is no inherent reason why the filtering or windowing need to match the segmentation stage, but it provides an implementation convenience (a single stream of filtered data can be shared for both stages).

Recognition: Feature Computation

Each 5-second window will be transformed into a feature vector, as with segmentation. Again, we described the features as a set of 20 features computed over each of 3 derived signals. The three signals are the **aX, aYZPC1**, and **gPC1** used above in segmentation.

Recognition features (computed for each signal):

- **Autocorrelation bins**: 5 evenly-spaced bins of the 5-second autocorrelation – summed per bin (5 features).
- **RMS**: The root-mean-square amplitude of the signal.
- **Power bands**: The magnitude of the power spectrum in 10 bands spaced linearly from 0.1-25Hz (10 features).
- **Mean, standard deviation, kurtosis, interquartile range** (4 features).

Recognition: Classification

Features are computed for each window in every training instance of each activity in the circuit; windows that only par-

tially include an activity are not included. As with segmentation, an L2-SVM is trained on this feature/label matrix; in this case, rather than a single binary SVM, a "multiclass SVM" (a series of binary SVMs in a most-predictions-wins configuration) is trained to discriminate among the activities. We train a multiclass SVM for each unique circuit.

Recognition: Voting

For an exercise set that lasts 20 seconds, we may have several predictions that disagree about which activity is being performed. In an offline context, it is straightforward to take the most common prediction over the entire exercise, but this does not apply to online use: we assume that it would be confusing to a user if RecoFit predicts "pushups", then – based on a change in the plurality vote – changes its prediction to "situps" later in the same set. Therefore, we want to make a single prediction and "stick with it". Empirically, we observed that the first second or two of exercise is often inconsistent with the rest of the set, due to rebalancing of weights, etc. On the other hand, waiting too long to commit to a label comes at a price in latency for real-time applications. Therefore, for online recognition, we use *a single window starting three seconds into the set* for recognition. For offline applications, we use the most common prediction over all windows from three seconds to the end of the exercise. We will discuss both metrics in our results.

Counting

Our counting algorithm assumes we have already performed segmentation and recognition, so it depends on the label (e.g. "jumping jacks"), and the raw accelerometer data corresponding to an exercise. Empirically, we did not find the gyroscope helpful for counting.

The counting stage itself has two components. First, we process the three-axis data into a one-dimensional (1d) signal. Second, we count peaks in this signal using the autocorrelation to eliminate peaks that do not correspond to repetitions.

Counting: Signal Computation

Our first goal is to extract a 1d signal over which we can count repetitions. Ideally, each exercise repetition should correspond to a single cycle or peak.

First, an elliptical bandpass filter (0.15Hz – 11Hz) removes high- and low-frequency components. We then subtract the mean from the data, apply Principal Component Analysis (PCA), and project the data onto its first PC. By projecting onto this axis – the axis of highest variation during this exercise – we simplify counting repetitions to the problem of counting peaks on a 1d signal.

Counting: Peak Detection

Figure 2a highlights an "easy" example: each repetition corresponds to a strong peak, all peaks are similar, and peaks occur at a constant rate. Counting peaks in this signal is straightforward, but few signals look like this. Depending on the exercise and the user's form, there may be multiple peaks per repetition (Fig 2b), variation in the repetition rate (Fig 2c), or variation in peak shape and amplitude (Fig 2d).

To overcome these difficulties, we use local period estimation to reject peaks that do not correspond to actual repetitions. This approach captures two key intuitions: peaks that are significantly "off-schedule" with respect to the current repetition rate are likely to be false repetitions, as are peaks that are significantly smaller than the typical peaks in the set.

The method assumes estimates of the minimum and maximum *reasonable* times needed for one repetition (*minPeriod* and *maxPeriod*). For example, it is nearly impossible for a situp to take less than 0.75s or more than 4s.

We first compute a set of candidate peaks (local maxima). We sort these peaks based on amplitude and loop through this sorted list, accepting a candidate peak so long as it is at least *minPeriod* away from the closest already-accepted peak. The intuition here is that if we see two peaks in the signal that are very close together (e.g. 200ms apart), one of them is not a real repetition of the exercise.

minPeriod is an estimate of the minimum *possible* time needed to perform one repetition (based on the fastest repetition we've *ever* seen of that exercise), so in most cases it is much smaller than the actual repetition time. Consequently, we often have multiple candidate peaks per actual exercise repetition. We attempt to estimate the actual exercise period around each peak, and we use this to refine the set of candidates. Specifically, for each candidate, we compute the autocorrelation in a window centered on the peak. We find the largest autocorrelation value within the range of lags [*minPeriod*,*maxPeriod*]; the lag corresponding to this value is our estimate of the exercise period *P* for this candidate. We repeat the process of sorting and filtering peaks, this time rejecting peaks that are closer to neighbors than 0.75**P*.

Finally, we filter remaining candidates using amplitude statistics. We again sort all of the candidate peaks based on amplitude and find the peak at the 40th percentile; we reject all peaks smaller than half the amplitude of this peak, which nearly always correspond to sub-repetition movements. We have found this much more robust than absolute thresholds. The number of remaining peaks is our final repetition count.

Counting: Online Considerations

We have described this process in the context of counting repetitions over a complete window of exercise; in an online scenario, we simply keep a buffer of the filtered signal corresponding to an exercise and repeat this process every 200 milliseconds. This window exists only to bound computation time; conceptually, we could re-count the entire exercise window with each arriving sample (unlike segmentation and recognition, which fundamentally require windowing). So once an exercise has been recognized, there is no fundamental latency in counting new repetitions.

END-TO-END STUDY

In order to evaluate our complete system, we conducted an experiment after all parameters were fixed on the training set.

Study A (4-class)	Squat, crunch, pushup, shoulder press
Study B (4-class)	Curl, jumping jack, triceps extension, dumbbell row
Medium (7-class)	Squat, curl, crunch, pushup, triceps extension, dumbbell row, jumping jack
Large (13-class)	Crunch, row, punch, jumping jack, kettlebell swing, triceps extension, pushup, rowing machine, Russian twist, back fly, shoulder press, squat, curl

Table 1: Circuits used in our study and cross-validation.

Participants

We recruited 20 participants (8 female), ages 25-53 (μ=35.1) from the area surrounding our institution. Participants varied in weight from <115lbs to >350lbs, and although they self-identified as "exercise at the gym at least once a week", participants were of widely varying fitness levels.

Procedure

We created two four-exercise circuits (Table 1) from the exercises collected during our training data. Each participant performed two rounds of one of these four-exercise circuits, for a total of eight exercises. Since we wanted to ensure adequate data for segmentation precision/recall analysis, participants were asked to take at least 30 seconds between sets, and to take a five-minute break between circuits. Prior to the session, participants were given a brief overview of the exercises and were instructed to select weights and repetition counts that they could complete safely and consistently. As with the training data collection, participants were given a wide range of suggested repetition counts (not enforced).

RESULTS

Segmentation Results

In evaluating our segmenter, we define *precision* as the fraction of predicted exercise sets that correspond to actual exercise, and *recall* as the fraction of actual exercise sets predicted as exercise. We further define three notions of precision and recall: *traditional*, *close*, and *tight*.

Traditional precision requires only that a predicted set uniquely overlap with a ground truth set, i.e. that the segmenter said "a set happened here".

Close precision requires *both* boundaries of a predicted set to fall within five seconds of the corresponding ground truth.

Tight precision requires *both* boundaries of a predicted set to fall within two seconds of the corresponding ground truth set.

Recall levels are defined in the same way. Table 2 presents precision/recall values for our study, along with leave-one-out cross-validation results on our training data (each participant's data was analyzed using a segmenter trained on all other participants). In our study, we achieve nearly 98.8% precision and recall by the "close" definition, corresponding to only two errors. In both error cases, a segmented region uniquely matched a ground truth label: one end of the segmented region was within one second of ground truth, but the other was off by about six seconds.

	Precision	**Recall**
End-to-End Study		
Traditional	100%	100%
Close	98.8%	98.8%
Tight	85.6%	85.6%
Leave-One-Out (Training Data)		
Traditional	99.1%	98.3%
Close	91.4%	91.0%
Tight	86.8%	86.9%

Table 2: Segmentation Results.

Recognition Results

Recognition accuracy is assessed in the context of a circuit, and inevitably the choice of circuit affects accuracy. A larger number of activities or high similarity among activities will reduce accuracy. Comprehensive analysis of all possible circuits is prohibitive, so we present results from the two circuits used in our study, along with leave-one-out cross-validation results from our training data for two reasonable circuits of different sizes, to demonstrate the effect of circuit size on accuracy. The circuits are described in Table 1.

We also note that an offline analysis, e.g. for post-workout summarization, can benefit from looking at an *entire set* before making a classification; a real-time system is expected to provide feedback within a reasonable amount of time at the beginning of a set (see "voting" above). Consequently, we present both "offline" recognition accuracies (using the entire set) and "online" recognition accuracies (using only the first 8 seconds of a set).

Table 3 shows the recognition results from our end-to-end study. Only three errors were made in the offline case, all three labeled "dumbbell rows" as "biceps curls", and visual inspection confirmed that participants were almost completely upright during this set, in which case rows and curls become almost indistinguishable. Interestingly, two of these were classified correctly in the online case, resulting in slightly *better* performance (one error total in 160 sets).

Critically, *recognition results for our study are identical whether we use the ground truth boundaries or the automatic boundaries produced by our segmenter*, confirming that recognition is robust to minor errors in segmentation.

Table 4 shows the same circuits analyzed using leave-one-out cross-validation analysis on our training data, along with the two larger circuits from Table 1. Even with the 13-class "large" circuit, post-workout (offline) accuracy is 96.0%.

Counting Results

Table 5 shows counting results from our end-to-end study and our training data. Using the ground truth activity boundaries and labels, counts are within 1 count of correct 97% of the time (93% using the automatic segmenter and recognizer). Results for the training data set (Table 5, third row) use actual event boundaries, across all 26 activities.

Circuit	Online	Offline	Chance
A	100%	100%	25%
B	98.8%	96.3%	25%
Mean	99.3%	98.2%	25%

Table 3: Recognition accuracies from our end-to-end study. Results are the same using ground truth activity boundaries and automatically-segmented boundaries. Chance is 25%.

DISCUSSION AND FUTURE WORK

Intensive Strength-Training and Periodicity Breakdown

One of our core intuitions is our use of self-similarity as a segmentation metric. However, self-similarity may break down in intensive strength-training scenarios. For this reason, more validation of intensive weightlifting is important future work. However, we highlight that we only rely on short-term periodicity: if the 10th repetition is different than the first, segmentation will not suffer, as long as it's somewhat similar to the 9th. This addresses the bulk of cases we've observed, where breakdown from fatigue is gradual. Furthermore, our aggregation step allows even a sudden shift in form (such as reversing grip) to be handled gracefully, so long as self-similarity resumes.

Such scenarios also may yield exercises longer than our 5-second windows. This window is tailored to minimize latency, but one may use longer windows for circuits with slow exercises, or use a short window in the non-exercise state and a longer window when looking for the *end* of slow exercises.

Mechanical and Form Factor Considerations

Our selection of the forearm as a location for our evaluation was primarily for experimental reasons: it was easier to develop a prototype that was stable and comfortable for the forearm than for the wrist. The wrist, however, is an appealing target for sensor placement, given the emergence of sensor-rich "smart watches" with the ability to transmit data to smartphones for computation, storage, and interaction. Anecdotal evidence from a preliminary wrist-based data set suggests that although we will need to train new models for wrist data, our approach will translate trivially.

Interestingly, many runners and walkers choose to keep their smartphones in armbands during exercise. This suggests an alternative route to deployment: a smartphone already contains the sensors, display, and computation required to run the complete RecoFit system as an application.

Finally, we assume a single inertial sensor, expecting that multiple devices would be prohibitively cumbersome for some users. However, our techniques generalize naturally to more sensor streams, particularly the use of PCA for dimensionality reduction. We expect that sensors on both arms and one or more legs would both improve accuracy and enable a wider variety of activities – such as leg-centric exercises – with straightforward adaptation of our methods.

Circuit	Online	Offline	Chance
Study A	99.4%	100%	25%
Study B	98.6%	99.3%	25%
Medium	96.2%	98.1%	14.3%
Large	93.8%	96.0%	7.7%

Table 4: Recognition accuracies from cross-validation analysis on our training data (using actual activity boundaries).

	Mean	Exact	Within 1	Within 2
Actual boundaries	0.26	0.5	0.97	0.99
Segmenter	0.52	0.77	0.93	0.97
Training data	0.49	0.70	0.93	0.97

Table 5: Counting accuracy (end-to-end study). The "mean" column refers to the mean absolute error across sets; the other columns indicate the fraction of sets (over 160 sets in our study) that are counted exactly, within 1, and within 2.

User Experience Considerations

A real-time application leveraging our approach is analogous to the "digital running coach" that a GPS device provides to a runner, allowing goal tracking and competition. However, for real-time applications, even the few seconds of latency incurred by our aggregation presents a challenge to making a system feel "responsive", as users expect counting to begin immediately. We have explored several mechanisms for addressing this limitation, for example using animations that respond directly to movement, even when the segmenter does not detect exercise, analogous to a moving waveform used in voice search applications indicate that "the system is listening". Furthermore, the aggregation value itself can be surfaced through color and animation to communicate progress toward detecting exercise. These design issues are critical to the persuasive and motivational goals of a fitness tracking system, and will be explored in future work.

We have developed a real-time embodiment of RecoFit (see video figure), which allowed us to confirm its computational feasibility (it runs easily on a phone-class processor or a high-end embedded processor) and will enable us to explore these important user experience questions.

Implications for Generalized Activity Recognition

We conclude our discussion by summarizing three key lessons derived from our work that we believe have implications for sensor-based activity recognition, even outside of the exercise space:

(1) When analyzing periodic signals, the use of independent learned models for periodicity identification and activity recognition can increase robustness.

(2) Dimensionality reduction can increase robustness to variation in device placement and behavioral orientation.

(3) We provide specific novel features to capture self-similarity for human motion applications, relevant to fitness, pedometry, physical therapy, etc.

ACKNOWLEDGEMENTS

Thanks to the incredible efforts of the data collection team, and to Sumit Basu, Desney Tan, Siddharth Khullar, and Harsh Vathsangam for advice.

REFERENCES

1. Amft, O. Adaptive activity spotting based on event rates. *Proc IEEE Sensor Networks, Ubiquitous, and Trustworthy Computing 2010*.

2. Atal, B., Rabiner, L. A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *IEEE Trans Acoustics, Speech and Sig Proc* 24(3), 201-212, 1976.
3. Boersma, P. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proc Inst Phon Sci* 17, p97-110, 1993.
4. Bravata, D. M., et al. Using pedometers to increase physical activity and improve health. *J Amer Med Assoc* 298(19), 2296-2304, 2007.
5. Centers for Disease Control and Prevention. Adult Participation in Aerobic and Muscle-Strengthening Physical Activities: United States, 2011. *Morbidity and Mortality Weekly Report*, 62(17):326-30, 2013.
6. Centers for Disease Control and Prevention. Physical activity recommendations for adults. cdc.gov/physicalactivity/everyone/guidelines/adults.html, retr 9/2013.
7. Chan, C. B., Ryan, D. A., Tudor-Locke, C. Health benefits of a pedometer-based physical activity intervention in sedentary workers. *Prev Med*, 39(6), 1215-22, 2004.
8. Chang, K-H., Chen, M., Canny, J. Tracking free-weight exercises. *Proc ACM Ubicomp 2007*.
9. Consolvo, S. et al. Activity sensing in the wild: a field trial of ubifit garden. *Proc ACM CHI 2008*.
10. Deng, J., Tsui, H. An HMM-based approach for gesture segmentation and recognition. *Proc IEEE Pattern Recognition 2000*.
11. Deslandes, A. et al. Exercise and mental health: many reasons to move. *Neuropsychobiology* 59.4:191-8, 2009.
12. Dobkin, B., Xu, X., Batalin, M., Thomas, S., Kaiser, W. Reliability and validity of bilateral ankle accelerometer algorithms for activity recognition and walking speed after stroke. *Stroke*, 42(8), 2246-50, 2011.
13. Hamer, M., Stamatakis, E., Steptoe, A. Dose-response relationship between physical activity and mental health: The Scottish Health Survey. *Brit J Sports Med*, 43:1111-14, 2009.
14. Harrison, B., Consolvo, S., Choudhury, T. Using multimodal sensing for human activity modeling in the real world. In *Handbook of Ambient Intelligence and Smart Environments*, 463-78. Springer, 2010.
15. Haskell, W.L. et al. Physical activity and public health: updated recommendation for adults from the American College of Sports Medicine and the American Heart Association. *Circulation*. 28;116(9):1081-93, 2007.
16. Janssen, I., LeBlanc, A.G. Systematic review of the health benefits of physical activity and fitness in school-aged children and youth. *Intl J Behavioral Nutrition and Physical Activity* 7.40: 1-16, 2010.
17. Junker, H., Amft, O., Lukowicz, P., Tröster, G. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognition*, 41(6), 2010-24, 2008.
18. Keerthi, S. DeCoste, D. A Modified Finite Newton Method for Fast Solution of Large Scale Linear SVMs. *J Mach Learn Res* 6, 341-361, 2005.
19. Krishnan, N., Lade, P., Panchanathan, S. Activity gesture spotting using a threshold model based on adaptive boosting. *Proc IEEE Multimedia & Expo (ICME) 2010*.
20. Kruger, J., Blanck, H. M., Gillespie, C. Dietary and physical activity behaviors among adults successful at weight loss maintenance. *Intl J of Behavioral Nutrition and Physical Activity*, 3(1), 17, 2006.
21. Lara, O., Labrador, M. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3), 1192-1209, 2013.
22. Lee, H., Kim, J. An HMM-based threshold model approach for gesture recognition. *IEEE Trans Pattern Analysis Machine Intellligence* 21(10), 961-973, 1999.
23. Merom, D. et al. Promoting Walking with Pedometers in the Community: The Step-by-Step Trial. *Am J Prev Med*, 32.4, 290-7, 2007.
24. Muehlbauer, M., Bahle, G., Lukowicz, P. What can an arm holster worn smart phone do for activity recognition? *Proc IEEE ISWC 2011*.
25. Murthy, G., Jadon, R. A review of vision based hand gestures recognition. *Intl J Inf Tech and Knowledge Management*, 2(2), 405-410, 2009.
26. Nguyen-Dinh, L., Roggen, D., Calatroni, A., Troster, G. Improving online gesture recognition with template matching methods in accelerometer data. *Proc Intel Sys Design Appl (ISDA) 2012*.
27. O'Donovan, G. Blazevich, A., Boreham, C. et al. The ABC of Physical Activity for Health: a consensus statement from the British Association of Sport and Exercise Sciences. *J Sports Sci* 28.6: 573-91, 2010.
28. Plasqui, G., Bonomi, A., Westerterp, K. Daily physical activity assessment with accelerometers: new insights and validation studies. *Obesity Rev*, 14(6):451-62, 2013.
29. Preece, S., Goulermas, J., Kenney, L., Howard, D., Meijer, K., Crompton, R. Activity identification using body-mounted sensors—a review of classification techniques. *Physiological Measurement*, 30(4), 2009.
30. Schutzer, K.A., Graves, B.S. Barriers and motivations to exercise in older adults. *Prev Med* 39.5, 1056-61, 2004.
31. Seeger, C., Buchmann, A., Van Laerhoven, K.. myHealthAssistant: A Phone-based Body Sensor Network that Captures the Wearer's Exercises throughout the Day. *Proc ACM Body Area Networks*, 2011.
32. Stiefmeier, T., Roggen, D., Tröster, G. Gestures are strings: efficient online gesture spotting & classification using string matching. *Proc Body Area Networks 2007*.
33. Thakor, N. V., Zhu, Y. S. Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection. *IEEE Trans Bio Eng*, 38.8, 785-94, 1992.
34. Tomporowski, P.D. Effects of acute bouts of exercise on cognition. *Acta Psychologica* 112.3: 297-324, 2003.