User-Centric Exergaming with Fine-Grain Activity Recognition: A Dynamic Optimization Approach

Bobak Mortazavi
Wireless Health Institute
Computer Science
Department - UCLA
Los Angeles, CA 90095 USA
bobakm@cs.ucla.edu

Sunghoon Ivan Lee
Wireless Health Institute
Computer Science
Department - UCLA
Los Angeles, CA 90095 USA
silee@cs.ucla.edu

Majid Sarrafzadeh
Wireless Health Institute
Computer Science
Department - UCLA
Los Angeles, CA 90095 USA
majid@cs.ucla.edu

UbiComp'14 Adjunct, September 13–17, 2014, Seattle, WA, USA
Copyright ACM 978-1-4503-3047-3/14/09...$15.00.
http://dx.doi.org/10.1145/2638728.2638806

Abstract
Exergaming, the use of activity, exercise, and information in video games, has been a growing field for the promotion of wellness and for preventing and treating obesity. Realistic exergaming requires movements that are adapted from detailed, fine-grain motions. An appropriate, active exergame requires a user-centric design, allowing for accurate motion recognition as well as a real-time responsiveness, often balancing accuracy with latency. This paper presents a framework for such an exergaming system, specializing on human interaction. This system includes a method for dynamically altering the algorithm to analyze the trade-off between classification accuracy and real-time responsiveness, allowing for a unique, tailored, interactive experience.

Author Keywords
Exergaming; Wireless Health; Dynamic Windowing; Derivative Free Optimization; User Experience; User Interfaces; User-Centered Design; Classification Accuracy;

ACM Classification Keywords
I.5.4 [Pattern Recognition]: Applications: Signal Processing; H.5.2 [User Interfaces]: User-centered Design

Introduction

Accurate physical activity monitoring of human motions has importance in many applications. These applications include medical monitoring systems [10], personal training [9], and personal gaming [1]. The latter, personal gaming, is of particular interest, since it spans the realm of different forms of human activity monitoring. Such monitoring can be on general movements [5], or detailed movements, such as those in exergaming, which is activity monitoring for gaming with targeted health applications [1]. This latter application realm is a target solution for childhood and adult obesity, a growing epidemic [14] and economic burden [4]. Indeed, work in [4] estimates that, over the next two decades, there will be a 33% increase in obesity and 130% increase in severe obesity in the United States. That work also proposes methods to save almost $550 billion in medical expenditures over that time. Exergaming has potential to help both the health epidemic as well as relieve some of the health care economic burden. User adherence to a game, however, drops with noticeable latency [3]. Latency can alter the user experience in a negative fashion in many application realms, for example, in user performance in gaming consistency and experience [15]. Ultimately, this shows the importance of analyzing each aspect of any interactive system and the importance of the human element in that loop.

This paper presents a system and optimization approach for playing a realistic soccer exergame, based on a system developed in [8]. That game consists of a sensor system for user input, a classification method for fine-grain activities, and a visual interface on a mobile tablet for gameplay and feedback. This paper dynamically adjusts the algorithm presented in [8] through the use of a derivative free optimization approach in order to define a unique exergaming experience for each individual user.

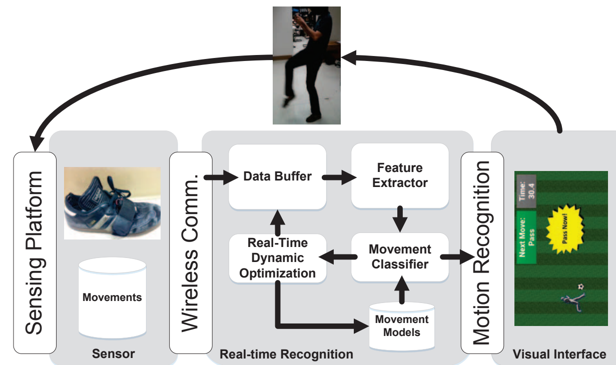


Figure 1: System Architecture of User Playing Soccer Exergame

Related Works

General Activities of Daily Living Monitoring

In [12], hereafter known as RDML, a method is presented that uses a combination of the mean of accelerations, the standard deviations, the energy or power expenditure, and a correlation between the channels of an accelerometer worn on the body for general activity monitoring of standing, walking, running, climbing up stairs, climbing down stairs, sit-ups, vacuuming, and brushing teeth. Features are extracted from windows of size 256, which is five seconds of repetitive, cyclical movement data. Fine-grain movements that fit in much smaller windows could be problematic in such a scenario. Ultimately, these calculations do not properly classify fine-grain movements[8], but are still presented as applicable to

Move	Description
Back Heel	Kick Backward
Behind Foot Pass	Pass left
Chip/Lob	Lift ball
Fake Shot	Fake kick
Flick Pass	Flick pass
Full Swing Shot	Full shot
Laces Shot	Mid shot
Quick Shot	Low shot
Curved Shot	Placed shot
Through Pass	Diagonal pass
Pass	Pass left
Step Over Move	Swing around ball
Side Step	Step right
Run	Step
Sprint	Step 2x

Table 1: Movements Captured

the dynamic optimization approach presented here.

Exergaming
There exist many vision-based exergaming approaches. The Microsoft Kinect camera system and SDK for human motion monitoring has proven effective [11] at monitoring activity. Skeletal gestures are combined principal component analysis (PCA) [11] to classify movements. The goal of this work is to develop a general system that can be used in a mobile setting that approaches the classification of many movements and delay in a similar fashion to those presented in [11] that use specialized hardware for processing.

Derivative Free Optimization
Dynamic optimization problems encompass the range of optimization problems where the constraints themselves often require complex solutions or simulations [13], and indeed, might be learned or sampled instead of known. Often, work involves evaluating different solvers for smooth or noisy constraint functions [7] or interpolation and approximation [6]. This paper takes the latter approach, by sampling the function for the constraint instead of knowing it outright. [2] uses a case based dynamic window to improve results and this work will take a similar approach for dynamically adjusting the window size, not for rule refinement, but for real-time responsiveness.

System Architecture

Exergaming Overview
This section describes the full system architecture presented in Figure 1. This figure shows the user playing the exergame. It starts with a motion from the user, follows to the sensing platform and movements required. From here, data is wireless transmitted to a computing device (e.g., a tablet computer). This computational device contains the recognition engine as well as dynamic optimization tools. When movements are detected they are passed to the game and the information is returned to the user via a visual interface. This loop needs to happen in a responsive fashion for an appropriate, interactive user experience.

Soccer Exergaming
Work in [8] develops a mobile, interactive gaming system for classifying fine-grain movements. The system developed in this paper incorporates an optimization approach to such a system, in order to develop a more responsive, unique exergaming experience. The classification engine in that game is based on twenty four users collecting the movements listed in Table 1, all referenced from the right foot wearing a three-axis accelerometer and three-axis gyroscope sampled at 100 Hz. A nearest-neighbor classifier based upon reconstructions from a principal component analysis are used to find the appropriate movements, based upon a 330 point sliding window, roughly 3 seconds, the full length of the largest move, and a source of the classification delay.

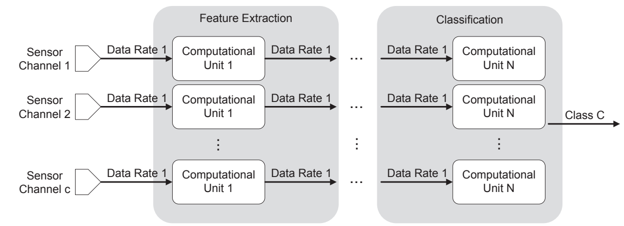


Figure 2: Sensor System and Computational Units

Dynamic Optimization

Problem Formulation

At its basis, the trade off between classification accuracy and latency can be modeled as an optimization problem. This work will use the Micro F1 score to measure overall accuracy, as it accounts for the precision and recall of a system. Using accuracy alone could bias a solution since a data set this large might result in high accuracy even if it classifies 0 movements because the number of true negatives would have been high. The F measure is calculated as follows:

$$F = 2 \times \frac{P \times R}{P + R} \tag{1}$$

where P is the precision and R is the recall rate of the learning model, and the factor of 2 puts the end result back in the familiar 0 to 1 range of precision, recall, and accuracy. Implicitly, F can be defined as $F(\delta)$ where δ is a delay factor for the system. This is because the more data present the better the system can classify; however, the more data used before a classification is made, then the longer the feature extraction steps take before classification. If one considers a system layout, as in Figure 2, where each sensor channel (from possibly different sensors) provides data at a given rate to different computational modules, an eventual classification is made when considering the combination of the data provided to each computational unit and the delay with which its computation proceeds. These values are, however, very application dependent and need to be learned. In fact, delay can be considered as the following:

$$\delta = \max_c \{ \sum_i (\alpha_i \times r_i \times k_i) - \beta \} \tag{2}$$

where r_i is the data rate to a given module, k_i is the computational delay for that operation, such as feature extraction, α_i is a scaling factor for a given module's importance to the overall delay, β is the real-time length of the motion (e.g. two seconds to shoot a soccer ball, delay only matters after the motion is completed at the point the ball should leave the foot) and the delay is the maximum such delay over all the sensor channels necessary to calculate the class C. From this, F as a function of δ becomes more clear and this presents the optimization problem of:

$$\min \delta \tag{3}$$

subject to

$$F(\delta) \geq \tau \tag{4}$$

where τ can be some predefined accuracy threshold. The difficulty comes in determining both the $F(\delta)$ and τ because the former is learned, and the latter is application specific. Figure 3 shows how the F measure changes as the window size changes. In particular, window size is one of the features that represents overall delay, and it seems here that the window size can range from about 150 to 330 samples with similar accuracy results. If a move takes 2 seconds (or 200 samples) then the most accurate point also represents the highest delay, a trade off the user might not want, when the 0 delay point has reasonable accuracy.

Dynamic Adjustment

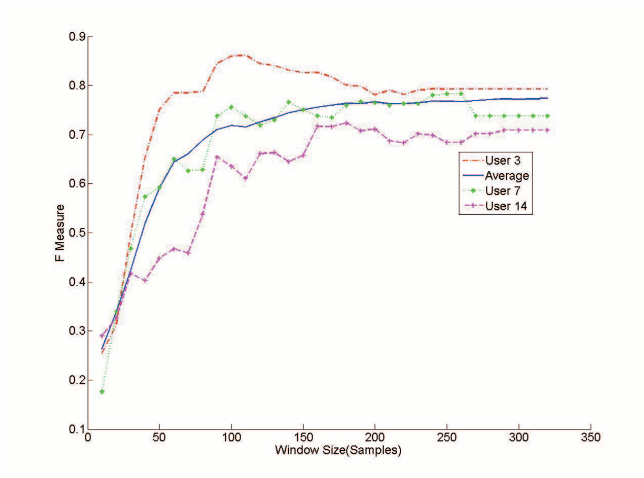


Figure 3: F Measures as scaled with window size for average and for different users as test subjects

As shown in Figure 3, different curves result from leaving specific users out of the training in a cross-validation setting of the data set in [8], using different window size each time. In other words, the way in which an individual user might perform particular actions either conforms to the model or shows their necessity in the training set for a more robust model for other users. The right hand side of the constraint then becomes $\tau(\delta)$, which is also sampled and learned. From this, a dynamic solution must be derived. A method to alter the delay must be chosen that is adaptable while in use. In particular, a dynamic window-size algorithm is chosen, where the constraint optimization problem tells us the left hand side of the constraint and the minimization goal wants to shrink the window size as much as possible while the right hand, the constraint threshold, limits this shrinking.

The learned model has a maximum classification accuracy at the full window size of 330 points. The dynamic windowing compares the classification result at the smaller sizes with the full size that is most accurate on the largest number of subjects, because no ground truth is known while a user plays with the system, and so the best known result must be used and compared. If a series of matches are found, where reduced window sized classification matches full window sized classification, the window size is shrunk. If a series of mismatches are found, the window size is increased again. In both cases, the threshold to allow for re-sizing can be adjusted per application use. In this case, since the latency is of greater concern, a series of only five matches in a row will be enough to shrink the window size (a number deemed an aggressive shrinking factor that still clearly represents a pattern of correct results). This aggressive shrinking must then be robust enough to regrow the window if it shrank in error (e.g., the user only performed one action correctly ten times and it was a small window sized action). Thus, five misses in a row will set a lower boundary on the possible window size and the window is grown, half way back to the previous set. As a result, a form of binary search results in eventually settling on an operating window size that is user specific. If the accuracy is of greater concern, the system can be altered to shrink less aggressively and re-grow more aggressively based on $F(\delta)$.

Results

A leave-one-subject-out cross-validation was used in order to interpret the results of this interactive system, and was applied to an algorithm for fine-grain motions as well as one designed for general daily living to show the adaptability of such a method. This cross validation was developed to simulate the algorithm behavior in an online setting by providing movements over time. The data set is that of [8].

RDML

The dynamic windowing algorithm was run a data set of moves from RDML to show a slow shrinking result in a system where accuracy is considered more important than delay, needed a greater number of correct matches. This shows the system is adaptable to general daily living and can be used to convert such a method to a real-time one. Figure 4 shows that this algorithm can slowly shrink to what is even less than the 256 point window size that is chosen by RDML.

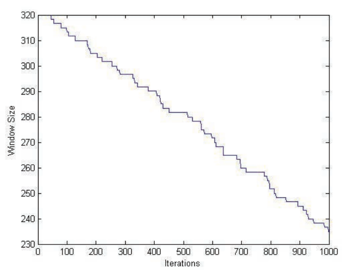


Figure 4: Dynamic windowing of RDML algorithm

Soccer Exergaming

Figure 5 shows four such runs of four users run to dynamically adjust the window sizes (where iterations are classification queries, including those that result in no movement, as defined in [8]). This shows each individual user has a different response to the model. Since this value is learned and sampled instead of defined, each user receives a tailored experience with a particular-sized window fit for that individual's performance. Note that Figure 5(a) settles on a much larger size than might have been picked with Figure 3, while Figure 5(b) is much smaller. Figures 5(c) and 5(d) show an average window size of 160 points. However, all show that in some cases this algorithm picks a much smaller window size than would be expected from general model, giving a user-centric model.

Future Work

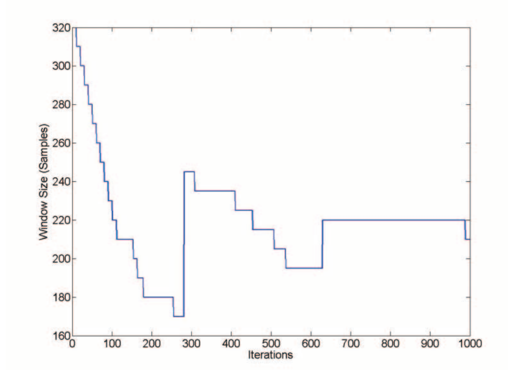
The system described in this paper shows a soccer exergame, with a dynamic optimization algorithm, developed on a tablet using a wearable sensor on a user's foot. The formulaic representation of the problem allows for adjustments to other parameters, including data rate, sampling rate, features, and/or power considerations to fine tune the human interaction. A more automated way to determine the dynamic adjustment and threshold parameters should be defined. Finally, some discussion should surround situations in which the smaller, optimized system provides higher accuracy than the full model, and thus certain misclassifications are desired and should be kept.

Conclusion

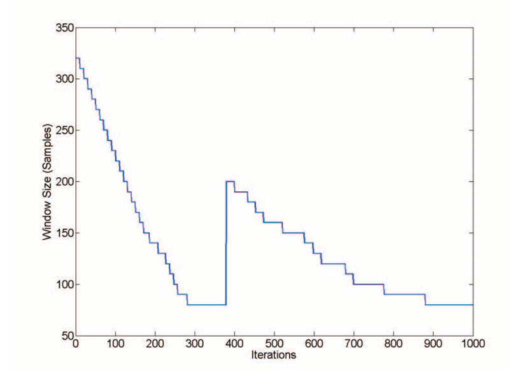
This paper presents an interactive exergame system with a method for optimizing the detection of fine-grain physical activity of the human body in a real-time environment for each individual user. The derivative-free optimization metrics can be adjusted on a per-application basis with the trade off of delay versus accuracy in mind. This gives each user a unique experience in a generalized exergame. Results run on data sets show that previous exergaming systems can be tailored to provide a lower-latency classification system with high accuracy during gameplay, giving each the best possible system of gameplay accuracy and gameplay latency.

References

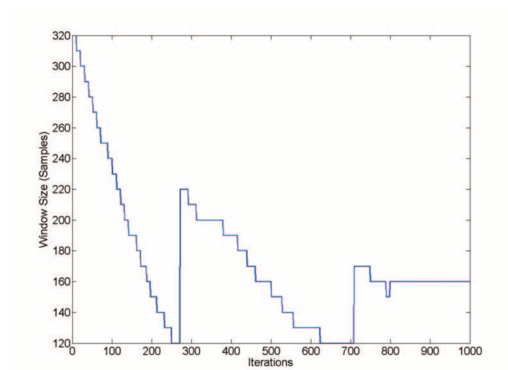
[1] Berkovsky, S., Coombe, M., Freyne, J., Bhandari, D., and Baghaei, N. Physical activity motivating games: virtual rewards for real activity. In *Proceedings of the 28th international conference*



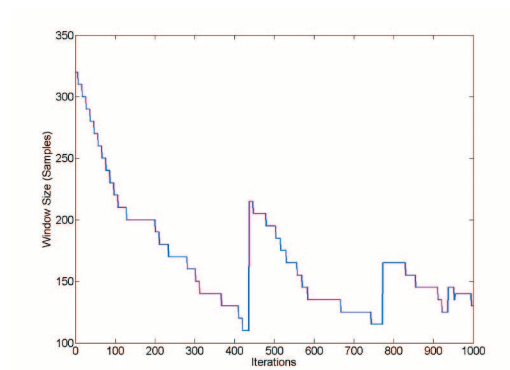
(a) A user with accuracy results needing a large window size than the average F-Measure shows



(b) A user with a much smaller window size than the average F-Measure shows



(c) A user with a window size indicated by the average F-Measure



(d) Another user with a window size indicated by the average F-Measure

Figure 5: Dynamic windowing of Soccer Exergame for (a) a below average user (b) an above average user (c) an average user with expected window size and (d) another user with an expected window size

on Human factors in computing systems, ACM (2010), 243–252.

[2] Chang, P.-C., Liu, C.-H., Lin, J.-L., Fan, C.-Y., and Ng, C. S. A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications 36*, 3 (2009), 6889–6898.

[3] Chen, K.-T., Huang, P., and Lei, C.-L. How sensitive are online gamers to network quality? *Communications of the ACM 49*, 11 (2006), 34–38.

[4] Finkelstein, E. A., Khavjou, O. A., Thompson, H., Trogdon, J. G., Pan, L., Sherry, B., and Dietz, W. Obesity and severe obesity forecasts through 2030. *American journal of preventive medicine 42*, 6 (2012), 563–570.

[5] Henpraserttae, A., Thiemjarus, S., and Marukatat, S. Accurate activity recognition using a mobile phone regardless of device orientation and location. In *Body Sensor Networks (BSN), 2011 International Conference on*, IEEE (2011), 41–46.

[6] Kramer, O., Ciaurri, D. E., and Koziel, S. Derivative-free optimization. In *Computational Optimization, Methods and Algorithms*. Springer, 2011, 61–83.

[7] Moré, J. J., and Wild, S. M. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization 20*, 1 (2009), 172–191.

[8] Mortazavi, B., Nyamathi, S., Lee, S., Wilkerson, T., Ghasemzadeh, H., and Sarrafzadeh, M. Near-realistic mobile exergames with wireless wearable sensors. *Biomedical and Health Informatics, IEEE Journal of 18*, 2 (March 2014), 449–456.

[9] Oliver, N., and Flores-Mangas, F. Mptrain: a mobile, music and physiology-based personal trainer. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, ACM (2006), 21–28.

[10] Pansiot, J., Stoyanov, D., McIlwraith, D., Lo, B., and Yang, G. Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, Springer (2007), 208–212.

[11] Raptis, M., Kirovski, D., and Hoppe, H. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM (2011), 147–156.

[12] Ravi, N., Dandekar, N., Mysore, P., and Littman, M. Activity recognition from accelerometer data. In *Proceedings of the national conference on artificial intelligence*, vol. 20, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2005), 1541.

[13] Rong, A., Akkerman, R., and Grunow, M. An optimization approach for managing fresh food quality throughout the supply chain. *International Journal of Production Economics 131*, 1 (2011), 421–429.

[14] Wang, Y., Beydoun, M., Liang, L., Caballero, B., and Kumanyika, S. Will all americans become overweight or obese? estimating the progression and cost of the us obesity epidemic. *Obesity 16*, 10 (2008), 2323–2330.

[15] Wu, W., Arefin, A., Huang, Z., Agarwal, P., Shi, S., Rivas, R., and Nahrstedt, K. Im the jedi!-a case study of user experience in 3d tele-immersive gaming. *Proc. IEEE ISM* (2010).