Making Gestural Input from Arm-Worn Inertial Sensors More Practical

Louis Kratz[1,2], T. Scott Saponas[1], Dan Morris[1]
lak24@drexel.edu ssaponas@microsoft.com dan@microsoft.com

[1]Microsoft Research, Redmond, WA
[2]Department of Computer Science, Drexel University, Philadelphia, PA

ABSTRACT

Gestural input can greatly improve computing experiences away from the desktop, and has the potential to provide always-available access to computing. Specifically, accelerometers and gyroscopes worn on the arm (e.g., in a wristwatch) can sense arm gestures, enabling natural input in untethered scenarios. Two core components of any gesture recognition system are *detecting* when a gesture is occurring and *classifying* which gesture a person has performed. In previous work, accurate detection has required significant computation, and high-accuracy classification has come at the cost of training the system on a per-user basis. In this note, we present a gesture detection method whose computational complexity does not depend on the duration of the gesture, and describe a novel method for recognizing gestures with only a single example from a new user.

Author Keywords

Gesture Recognition, Hidden Markov Models

ACM Classification Keywords

I.5.5 Pattern Recognition: Interactive Systems

INTRODUCTION

Gestural input has the potential to greatly expand scenarios in which interacting with computing is convenient, feasible, and fun. Gesture recognition has already been widely deployed using a variety of sensors including touch screens, pen trackers, hand-held motion controllers, and depth-camera-based skeleton trackers. Another type of gestural input with a wide variety of applications is arm movement detected by inertial sensors embedded in wearable devices (e.g., a wristwatch). This modality has several attractive properties: it can be performed one-handed, worn continuously, leaves the hands free, and does not require sensors in the environment. In order for gesture recognition from arm-worn inertial sensors to be practical, however, systems must be able to efficiently and accurately recognize gestures with as little training from new users as possible.

Two specific problems in gesture recognition are *detecting* when a gesture is occurring and *classifying* which gesture a person has performed. A key challenge in gesture detection is duration ambiguity: users naturally perform gestures at different speeds, making it impossible to know how many sensor measurements compose an instance of a gesture. This ambiguity introduces significant computational cost, since the entire range of possible gesture durations must be evaluated.

Another significant challenge in gesture recognition is to reduce or eliminate the need for new users to train the system. Previous approaches [5, 11] suffer from significant drops in accuracy when presented with a user not in the training set. Variations in gestures occur among users due to physical (such as the user's height or arm length) or stylistic (such as the the shape of the gesture) differences. These variations, however, often result in only a linear transformation of the input space. For example, two users of different heights performing the same gesture will move their arms at different speeds, resulting in a scaling of the motion derivatives.

In this note, we present an approach to gesture recognition that addresses these issues of duration ambiguity and new user training. Our key insights are 1) by representing each gesture using a hidden Markov model (HMM), we may efficiently evaluate the range of possible durations by exploiting redundant computations and 2) we can learn a user's unique variations from a *single example* and use it to *map* their input space to a known user in the training set. We evaluate our method through an eight-person experiment where participants perform gestures while wearing a three-axis accelerometer and three-axis gyroscope on their forearm. Our results show that our user mapping vastly improves detection accuracy and achieves classification accuracy similar to models trained separately for each user.

RELATED WORK

Other detection methods search the entire range of gesture durations. Feature similarity search [1, 2, 6] compares a templated set of features to one computed over all possible gesture durations at each time step. Wilson et al. [10] use HMMs, as we do, but compute the likelihood of each possible subsequence. The computational complexity of these methods, however, depends on the range of durations over which a gesture can occur. As such, detection can be computationally unreasonable, especially if the sensor has a high sampling rate or the range of possible gesture durations is large.

Many vision-based methods does not evaluate the range of possible durations at all. Lee and Kim [4] and Peng and

Qian [7] use non-gesture models in an HMM network to perform detection. Deng and Tsui [3] compute the likelihood of all possible subsequences of observations, not just the ones within a known range. Such methods perform well for vision-based detection, but do not retain any notion of the gesture duration and understandably suffer from false positives.

User-invariant gesture recognition has been largely unaddressed. Most detection work do not address it at all, and classification work suffers from a gap in accuracy. Lui et al. [5] report a drop in classification accuracy (almost 25%) when evaluating their method on users not in the training set. They mention that a larger training set should relieve such limitations, but capturing such a training set requires additional resources. Wu et al. [11] report a drop from 95% to 89%. This gap, though seemingly small, represents enough incorrect classifications to make an input system ineffective. In contrast, our approach matches new users to ones in the training set using a *single* example of the gesture. By doing so, we achieve accurate recognition without the need for a large training set.

RECOGNITION USING HIDDEN MARKOV MODELS

Hidden Markov models (HMMs) are a popular tool for gesture recognition due to their temporal nature. HMMs containing N states are represented by an $N \times 1$ vector $\boldsymbol{\pi}$ of initial state probabilities, an $N \times N$ matrix $\mathbf{A}$ of transition probabilities, and a set $\mathcal{B}$ of N emission densities. Inference is efficiently computed with the Forwards-Backwards algorithm [8]. We represent each gesture with a left-to-right HMM. In left-to-right HMMs, each hidden state has only two non-zero transition likelihoods: one to stay in the current state, and one to transition to the next. (see [8] for details).

A gesture is a subsequence of observations $O_{s:e} = \{o_s, \ldots, o_e\}$ with a duration $D = e - s$ (where s and e indicate the start and end time, respectively). In our scenario, each observation o_t at time t is a 6×1 vector containing 3 accelerometer and 3 gyroscope values. If s and e are known, the HMM that yields the largest probability is the inferred gesture.

Often, the start and end times are not known, and thus the gesture must be *detected.* Detection may be achieved by identifying samples with a high likelihood of being in the end state

$$\mathrm{p}(z_e = N, O_{s:e} | \lambda_g) \tag{1}$$

where z_e is the hidden state at the end time e, N is the number of hidden states, and λ_g is the HMM model for gesture g. If the starting time s is known, then Eq. 1 may be computed in $O(N^2 D)$ time using the forward-backward algorithm.

People naturally perform gestures at different speeds, however, and thus the duration D is not constant. A gesture duration can vary over a fixed range from D_{min} to D_{max}. Even with this assumption, however, the computational complexity is high since we would need to evaluate all possible sequences of length D_{min} to D_{max} at each time instance t. In the worst case, $D_{min} = 1$ and evaluating all subsequences takes time

$$O(N^2 \sum_{j=1}^{D_{max}} j) = O(N^2 D^2). \tag{2}$$

Fast Detection Over a Range of Subsequences

To perform detection, we wish to evaluate Eq. 1 for all possible subsequence lengths from D_{min} to D_{max}. Let $t_m = t - D_{max}$ and $t_r = t - D_{min}$. We consider a gesture spotted if

$$\sum_{k=t_r}^{t_m} \mathrm{p}(z_t = N, O_{k:t} | \lambda_g) > \mathcal{T}_g \tag{3}$$

where $\mathcal{T}_g$ is a threshold selected empirically. Eq. 3 represents the likelihood that a gesture within the desired range ended at time t. After the end time is known, we identify the start state using Viterbi backtracking.

Deng and Tsui [3] compute the sum of all subsequences from $t = 1$ to t by assuming that each subsequence is independent:

$$\sum_{k=1}^{t-1} \mathrm{p}(O_{t_k:t} | \lambda_g) = \mathrm{p}(O_{1:t} \cup \ldots \cup O_{t-1:t} | \lambda_g). \tag{4}$$

They compute Eq. 4 using a vector

$$\boldsymbol{\phi}_t(i) = \mathrm{p}(z_t = i, O_{1:t} \cup \ldots \cup O_{t-1:t} | \lambda) \tag{5}$$

that is defined sequentially by

$$\boldsymbol{\phi}_1(i) = \boldsymbol{\pi}_i b_i(o_t) \tag{6}$$

$$\boldsymbol{\phi}_t(i) = \left(\boldsymbol{\pi}_i + \sum_{j=1}^{N} \boldsymbol{\phi}_{t-1}(j) \mathbf{A}_{ji} \right) b_i(o_t)\ , \tag{7}$$

where $\mathbf{A}_{ji}$ is the HMM's transition likelihood, $\boldsymbol{\pi}_i$ is the initial probability of state i, and $b_i(o_t)$ is the probability of observation o_t given hidden state i. They consider a gesture spotted when the following is greater than some threshold.

$$\boldsymbol{\phi}_t(N) = \sum_{k=1}^{t-1} \mathrm{p}(z_t = N, O_{t_k:t} | \lambda_g). \tag{8}$$

Note the difference between Eq. 8 and Eq. 3: Deng and Tsui compute the likelihood of *all* possible subsequences, while we examine the likelihood of only a *range* of subsequences. This is a key difference: Deng and Tsui assume subsequences that are too short or too long to be real gestures will have low likelihoods as modeled by the HMM. We show later this can result in a larger number of errors.

We compute Eq. 3 using two instances from Deng and Tsui's method: one that starts at time t_m and one that starts at $t_r + 1$

$$\begin{aligned} \sum_{k=m}^{r} \mathrm{p}(O_{t_k:t} | \lambda_g) = \ & \mathrm{p}(O_{t_m:t} \cup \ldots \cup O_{t-1:t} | \lambda_g) \\ & - \mathrm{p}(O_{t_r+1:t} \cup \ldots \cup O_{t-1:t} | \lambda_g). \end{aligned} \tag{9}$$

The key to efficient computation is to update Eq. 9 at time $t+1$ without re-running the entire algorithm. We do so according to:

$$\begin{aligned} \sum_{k=m+1}^{r+1} \mathrm{p}(O_{t_k:t+1}) = \ & \mathrm{p}(O_{t_m:t+1} \cup \ldots \cup O_{t:t+1}) \\ & - \mathrm{p}(O_{t_r+1:t+1} \cup \ldots \cup O_{t:t+1}) \\ & + \mathrm{p}(O_{t_r+1:t+1}) - \mathrm{p}(O_{t_m:t+1}) \end{aligned} \tag{10}$$

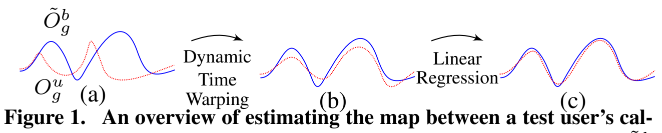

Figure 1. An overview of estimating the map between a test user's calibration gesture O_g^u (red) and a training user's canonical example $\tilde{O}_g^b$ (blue). We temporally align the data (b) using dynamic time warping, and estimate the map (c) using linear regression. We use the map to transform inputs between the test user u and the training user b, enabling accurate recognition with only a single calibration gesture.

(we have dropped the λ_g for convenience). Note that the first two terms on the right hand side of Eq. 10 are computed by applying the update step in Eq. 7 to the values computed at time t, and takes $O(N^2)$ time. The last two terms $\mathrm{p}(O_{t_r+1:t+1})$ and $\mathrm{p}(O_{t_m:t+1})$ are simply the likelihoods of a single sequence, and may be computed in $O(N^2D)$ using the forward-backward algorithm.

Under specific numeric conditions we may improve our algorithm to run in $O(N^3)$, completely *independent* of the duration of the gesture or range of possible subsequences, by efficiently computing $\mathrm{p}(O_{t_r+1:t+1})$ and $\mathrm{p}(O_{t_m:t+1})$ using values at the previous time instance. For the interested reader, we have included this proof in supplementary material[1].

Refining Gesture Candidates

Each time the value of Eq. 3 goes above $\mathcal{T}_g$ a gesture is considered detected. Often, Eq. 3 is greater than the threshold for several samples near the end of the gesture, resulting in multiple detections. Stack-based approaches [3, 4] keep either the longest or most likely detected gesture. We use similar logic: candidates are buffered until Eq. 3 drops below the threshold, and we return the candidate with the highest likelihood. In addition, we disregard candidates whose duration is less than D_{min} or greater than D_{max} to reduce errors.

USER-INVARIANT GESTURE RECOGNITION

We represent the physical and stylistic differences between users as a linear transformation of the accelerometer and gyroscope measurements. Given a new user u, we assume there exists a similar user, or "buddy", b in the training set. For each gesture g, we map the observations o_t from the new user to the buddy's input space using a linear transformation f_g^b

$$f_g^b(o_t) = \mathbf{K}o_t + \mathbf{w}\ , \qquad (11)$$

where $\mathbf{K}$ is a 6×6 diagonal matrix and $\mathbf{w}$ is a 6×1 vector. Note that $\mathbf{K}$ and $\mathbf{w}$ are unique to each user-buddy-gesture triple.

If buddy b and map f_g^b are known, then the new user's gestures may be recognized by transforming their input before computing the likelihood. Eq. 1 becomes

$$\mathrm{p}(z_e = N, f_g^b(o_s), \ldots, f_g^b(o_e)|\lambda_g^b)\ , \qquad (12)$$

where λ_g^b is the HMM trained for gesture g and user b.

Fig. 1 illustrates our method for estimating the map f_g^b between a new user and a given buddy. We compute a canonical example $\tilde{O}_g^b$ of the buddy's gesture g by temporally aligning all of their training samples using dynamic time warping [9] and averaging the results. We align a single calibration gesture O_g^u (red) from the user the buddy's canonical example

[1]http://research.microsoft.com/cue/chi12gestures

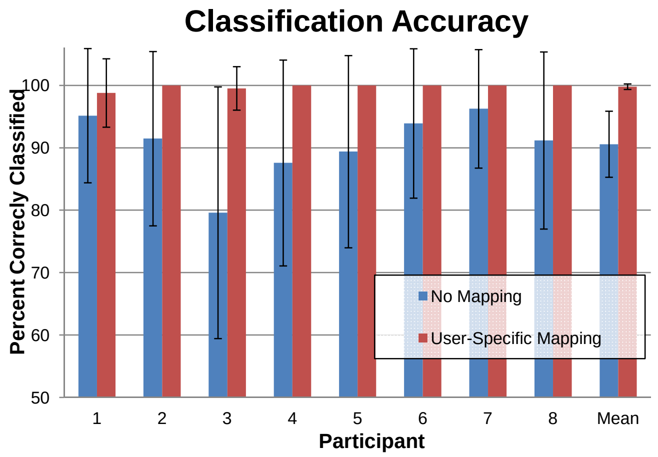

Figure 2. Gesture classification accuracy with (red) and without (blue) our user-specific mapping method. Error bars show standard deviation.

$\tilde{O}_g^b$ (blue) using dynamic time warping. To do so, however, requires similar amplitudes from both examples, which is exactly the function of f_g^b. We address this by normalizing the amplitude of both $\tilde{O}_g^b$ and O_g^u, and using the difference of absolute values as the dynamic time warping distance. We then use the offset from dynamic time warping to temporally align the *original* calibration gesture and the canonical example. Finally, we use least-squares linear regression to estimate the parameters to f_g^b.

Note that the linear transformation f_g^b is applied to the amplitude alone, and recognition is performed using Eq. 12 *without* temporal alignment (i.e., no dynamic time warping is used during recognition). HMMs are robust to such temporal variations (such as speed), but not to amplitude differences, and thus temporally aligning the query samples is unnecessary.

Selecting The Best Buddy

We wish to know which training user will provide the best recognition for a test user. For classification, we pick b as the training user whose models provide the highest likelihood. For detection, however, we achieve better results by selecting different buddies for each gesture, and do so empirically. We plan to explore optimal buddy selection as future work.

RESULTS

We evaluate our method on a data set of eight different gestures collected from eight participants. The gesture set included circle shapes in the air (clockwise and counter-clockwise), a wave of the hand, swatting (up, left, and right), pointing, and touching the head. A three-axis ST Micro LIS331DLH accelerometer and a three-axis ST Micro L3G4200D gyroscope (both sampled at 15Hz) were attached to the participant's forearm. Data was sent to a PC via Bluetooth. Each participant performed ten examples of each gesture for training, then ten more for testing.

Gesture Classification

First, we evaluate our user-invariant method on gesture *classification* by performing a leave-one-out test: for each user, we use the other seven users as a training set. To increase ro-

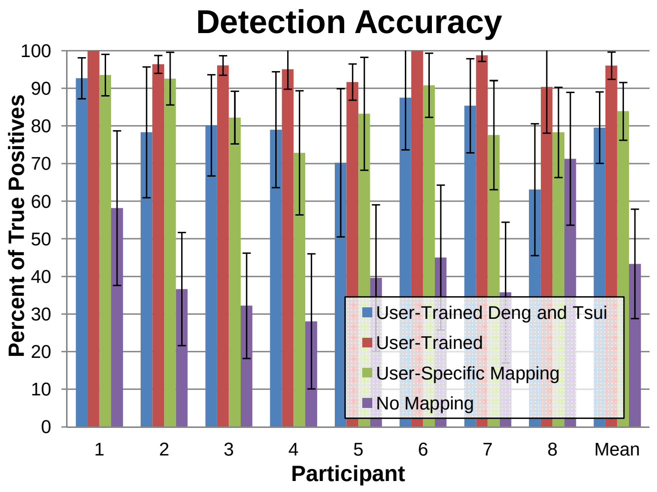

Figure 3. Detection accuracy using our (red) and Deng and Tsui's [3] (blue) methods with user-trained models (i.e., each user provides their own training data). Error bars show standard deviation. Our method achieves a higher accuracy by evaluating gestures within a known range of possible durations. Models not including training data from the test user (purple) have poor detection accuracy. Using our mapping (green) vastly increases the detection accuracy, above even the user-trained method of Deng and Tsui.

bustness, we map the entire training set to the buddy b before training the model λ_g^b. This increases the number of training examples for each model.

Fig. 2 shows the classification accuracy for all eight users without mapping (blue) and using our method (red). Classification accuracy is the ratio of correctly classified gestures to the total number of gestures performed. An average accuracy of 90% (sd 5.29) is achieved with models trained without mapping. Other methods [5, 11] report similar results when evaluating on users not in the training set. By learning the transformation between user spaces, our method achieves over 98% accuracy for all users and a mean accuracy of 99.8% (sd 0.4).

To measure the statistical significance of our method, we compared the distributions of the classification results for the user-specific mapping and non-mapped results using a two-sample t-test; the distributions differ with with $p < 0.001$.

Gesture Detection

We hand-label the start and end of each gesture to qualitatively evaluate our method. The detection accuracy is the ratio of correct spots to all detection results (false negatives, true and false positives).

Fig. 3 shows the detection accuracy of our approach (red) and that of Deng and Tsui [3] (blue) using models trained on each user. Since we only consider gestures that occur within a specific range, our log-likelihood is more responsive and results in fewer errors. Overall, we achieve an average of 96% accuracy (standard deviation of 3.6), compared to 80% from Deng and Tsui (standard deviation of 9.5). A two-sample t-test shows that the detection results are from different distributions with $p < 0.001$.

Fig. 3 also shows detection accuracy when the user is not in the training set (purple) and after applying our mapping (green). Without mapping, the accuracy is quite poor (average 43%, sd 14.5). Our approach achieves an average of 84% accuracy (sd 7.7), higher than even the user-trained models from Deng and Tsui's [3] (blue). A two-sample t-test at shows that the detection results are from different distributions with $p < 0.001$. User-specific mapping provides a clear improvement when faced with users not in the training data set.

CONCLUSION

Gesture recognition has been too limited by user-specific variabilities and computational challenges for use in real-world systems. We have presented an accurate, efficient method that improves both gesture *detection* and gesture *classification* by transforming the input space to a user already in the training set. In addition, our method efficiently searches the range of possible gesture durations by exploiting redundant computations. Our mapped models classify gestures with accuracy comparable to those trained by the specific user, but does so with only a single calibration example. Our method vastly improves detection accuracy over previous approaches, especially for users not in the training set.

REFERENCES

1. Amft, O. Adaptive activity spotting based on event rates. In *Proc. of the IEEE Int'l Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing* (2010), 169–176.
2. Amft, O., and Troster, G. Recognition of dietary activity events using onbody sensors. *Artificial Intelligence in Medicine 42* (2008), 121–136.
3. Deng, J. W., and Tsui, H. T. An hmm-based approach for gesture segmentation and recognition. In *Proc. of the Int'l Conf. on Pattern Recognition* (2000).
4. Lee, H.-K., and Kim, J. H. An hmm-based threshold model approach for gesture recognition. *IEEE Trans on Pattern Analysis and Machine Intelligence 21* (1999), 961–973.
5. Liu, J., Wang, Z., Zhong, L., Wickramasuriya, J., and Vasudevan, V. uWave: Accelerometer-based Personalized Gesture Recognition and its Applications. In *IEEE Int'l Conf. on Pervasive Computing and Communications* (2009), 1–9.
6. Lombriser, C., Amft, O., Zappi, P., Benini, L., and Troster, G. *Benefits of Dynamically Reconfigurable Activity Recognition in Distributed Sensing Environments*. World Scientific Publishing, 2010, ch. 12, 261–286.
7. Peng, B., and Qian, G. Online gesture spotting from visual hull data. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 6 (2011), 1175–1188.
8. Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. of the IEEE* (1989), 257–286.
9. Sakoe, H., and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Trans. on 26*, 1 (feb 1978), 43–49.
10. Wilson, A. D., and Bobick, A. F. Learning visual behavior for gesture analysis. In *In Proc. IEEE Int'l. Symp. on Comp. Vision* (1995).
11. Wu, J., Pan, G., Zhang, D., Qi, G., and Li, S. Gesture recognition with a 3-d accelerometer. In *Proc. of the Int'l Conf. on Ubiquitous Intelligence and Computing* (2009), 25–38.