Poster: When Wearable Sensing Meets Arm Tracking

Yang Liu, Chengdong Lin, Zhenjiang Li
City University of Hong Kong, Hong Kong, China

Zhidan Liu, Kaishun Wu
Shenzhen University, Shenzhen, China

ABSTRACT

In this poster, we present our recent work, a wearable system for achieving real-time 3D arm skeleton. We have coped with the major challenge that the skeleton of each arm is determined from the locations of the elbow and wrist, whereas a wearable device only senses a single point from the wrist. Result shows that the potential solution space is huge. This underconstrained nature fundamentally challenges the achievement of accurate and real-time arm skeleton tracking. In this study, we propose Hidden Markov Model (HMM) state reorganization and hierarchical search two methods to improve the heavyweight computation of the state-of-art arm tracking model and achieve real-time tracking even on mobile phone.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**.

KEYWORDS

arm tracking, mobile sensing

ACM Reference Format:
Yang Liu, Chengdong Lin, Zhenjiang Li and Zhidan Liu, Kaishun Wu. 2019. Poster: When Wearable Sensing Meets Arm Tracking. In *The 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '19), June 17–21, 2019, Seoul, Republic of Korea.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3307334.3328603

1 INTRODUCTION

We propose a wearable system for comprehensively understanding and analyzing the detailed arm motions of people. This system can track the 3D skeleton (posture) of the entire arm of a user in **real time** on mobile phone, *e.g.,* the locations of the *elbow* and *wrist* with respect to (*w.r.t.*) the body [2]. In addition, it uses motion sensors from a wearable device only on the user's wrist (*i.e.,* accelerometer and gyroscope from a smart watch or wristband) instead of attaching multiple sensors on the user's entire arm.

To develop the aforementioned system, we take advantage of kinematic studies [1], wherein ArmTrak [2] recently makes a remarkable contribution to recover user's arm motions from a single watch, to achieve the aforementioned skeleton tracking design; however, the shortcoming is long recovery latency, *e.g.,* a t-time activity requires approximately $10 \times t$ times to recover even on a desktop PC [2][1], which is due to the inherent hardness problem, and limits the proposed solution to off-line analysis.

Skeleton recovery is essentially a search problem (for unknown elbow and wrist locations within a huge space). We observe that the search space can be carefully diminished (without impairing tracking accuracy), and "unlikely" candidates can be intelligently excluded as early as possible to considerably accelerate the search. In particular, we propose HMM state reorganization and hierarchical search two methods to improve the heavyweight computation of the state-of-art arm tracking model [2]. By these design, skeleton tracking occurs in real time on a mobile phone, and can promisingly achieve a higher accuracy than ArmTrak [2]. In addition, recovery errors are not accumulated over time.

2 SKELETON TRACKING DESIGN

2.1 Design Principle

Arm skeleton model. An **arm skeleton** refers to an arm's 3D geometric relation, which is uniquely determined by: elbow and wrist locations, wrist orientation in the torso coordinate system [2].

Design principle of ArmTrak [2]. Via the arm skeleton model, determining an arm's skeleton essentially confirms two parameters: 1) loc_{elb}: the relative position of the *elbow*, and 2) ori_{wrs}: the orientation of the *wrist* in the torso coordinate system. Once loc_{elb} and ori_{wrs} are determined, the wrist location also becomes available because the arm is a rigid object [1]. In particular, the ori_{wrs} can be *indirectly measured* from the watch's gyroscope data ori_{watch} [2]. Thus, the remaining task is determining loc_{elb} through two phases:

1) *Off-line phase*: From the kinematic model, ArmTrak observes that all possible elbow locations, loc_{elb}, are within a limited range given one measured wrist orientation ori_{wrs}, denoted as point cloud. In the off-line phase, for each wrist orientation ori_{wrs} (of several degree granularity), ArmTrak builds a library to store its corresponding point cloud for the user, which is a one-time effort.

2) *Recovery phase*: When a user's arm is moving, the acceleration of the elbow $acc_{elb}(t)$ can be converted from the reported acceleration $acc_{watch}(t)$ from the smart watch. After T time stamps, we have T point clouds (based on $ori_{wrs}(t)$). We can generate a feasible moving trace of the elbow by selecting one location from each point cloud and infer the corresponding acceleration trace, $\{\overline{acc}_{elb}(t)\}_{t=1}^{T}$. Then, we find the elbow location trace generating $\{\overline{acc}_{elb}(t)\}_{t=1}^{T}$ which best matches the indirectly measured $\{acc_{elb}(t)\}_{t=1}^{T}$.

Complexity. The problem can be formulated using the HMM with Viterbi algorithm within $O(S^2T)$ [2], wherein the search space size is S, and the total time step is T. To fulfill the HMM formulation, each circle in Figure 1(a) represents a point cloud at time t of size $O(N)$. Each HMM state of ArmTrak is defined as a pair of elbow locations among two consecutive time stamps. The search space of each

[1]A fast version of a simple weighted average design is proposed in [2]; however, it is inaccurate and insensitive to arm motions.

ACM ISBN 978-1-4503-6661-8/19/06.
https://doi.org/10.1145/3307334.3328603

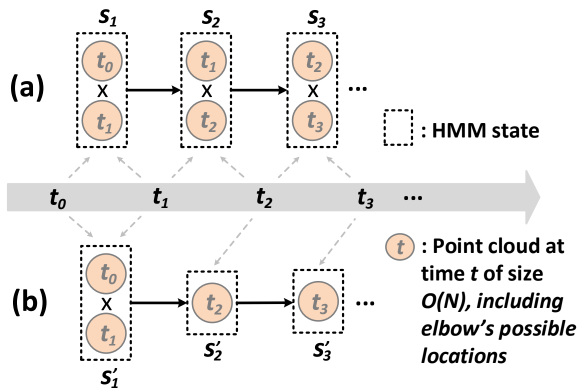

Figure 1: HMM state constructions of size (a) $O(N^2)$, and (b) $O(N)$ for all the states except the first one [3].

HMM state is $O(N^2)$, and the solution complexity is $O(N^4T)$, which can be reduced to $O(N^3T)$ by leveraging the location continuity constraint [2]. To further accelerate the search, T is downsampled to 5 Hz from default 50 Hz.

2.2 Accelerating Skeleton Recovery

2.2.1 HMM State Reorganization. The transition from s_1 to s_2 in Figure 1(a) can be approximated as:

$$s_1' = < loc_{elb}(0),\ loc_{elb}(1) > \quad \rightarrow \quad s_2' = < loc_{elb}(2) >,$$

which also preserves all possible acceleration values cross the first three point clouds (Figure 1(b)). The advantage of this update is that starting from the fourth point cloud (after t_3 in Figure 1(b)), all the remaining states can be defined as $s_t =< loc_{elb}(t) >$, which essentially reduce the possible paths from the states in Figure 1(a). Only the first state has size $O(N^2)$, whereas the sizes of all the remaining states are $O(N)$, *e.g.,* the overall search space S is nearly $O(N)$. Therefore, search complexity is decreased to $O(N^2T)$.

2.2.2 Hierarchical Search. In the original Viterbi search, we need to explore within a large search space for each time step, but only one location is the correct solution, which implies that most computations are consumed ("wasted") to calculate the likelihoods for all "incorrect" locations so that they can be eventually excluded. Our core idea is thus to exclude incorrect locations as early as possible to minimize computational waste for acceleration.

To this end, we propose to conduct the search in a hierarchical manner. In particular, we first conduct downsampling for point clouds with a ratio of $\frac{1}{n_1}$, *i.e.,* we group every n_1 nearby locations in each cloud into $O(\frac{N}{n_1})$ regions and use the centroid of each region to form a coarse-level search space. After performing the first round of search on this coarse-level search space, the most likely region can be determined for each time step, as shown in Figure 2. The complexity for completing this round of search is $O(\frac{N}{n_1})^2T$.

We can immediately launch the next round of search after selecting four regions from the first round (nearly in parallel) while focusing on these selected regions, *i.e.,* the outputs from the first round. In this round, the effective point clouds are merely "shrunk"

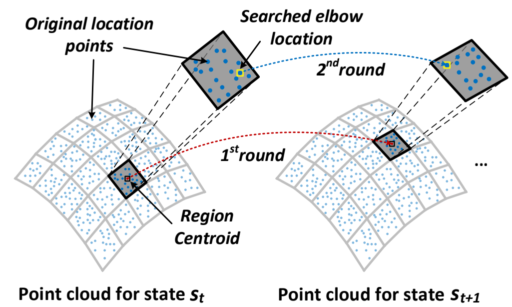

Figure 2: Illustration of the hierarchical search [3].

to these selected regions of size $O(n_1)$, as shown in Figure 2. In principle, we can further divide each selected region into sub-regions again. In our implementation, we adopt a two-layer search, thus the time complexity of the second round search is $O(n_1)^2T$.

Thus, complexity decreases from $O(N^2T)$ to $O((\frac{N}{n_1})^2T + (n_1)^2T)$, where $n_1 \ll N$. This result also indicates that complexity reduction mostly results from the first round of downsampling.

2.3 System Performance

We develop our system using LG watches (with Invensense MPU-6515 six-axis motion sensors) and SAMSUNG Galaxy S7. The ground truth is collected from Kinect 2.0. In our implementation, an activity with duration t requires $t \times 0.47$ times to be recovered by our system on the mobile phone. Moreover, we evaluate our system to traceback at the current time stamp in the HMM search to report the instant locations every second since our system can run in real time on mobile phone. Overall, the median errors of elbow and wrist from our system are 10.53 and 12.94 *cm*, respectively.

3 CONCLUSION

In this poster, we propose novel techniques to enable real-time 3D arm skeleton on mobile phone through HMM state reorganization and hierarchical search two methods to improve the heavyweight computation of the state-of-art arm tracking model.

ACKNOWLEDGEMENTS

This work is supported by the ECS (CityU 21203516) and GRF (CityU 11217817) grants from Hong Kong RGC. It is also supported by NSFC Grant (No.61802261 and No.61872248), NSF Grant of Shenzhen University (No.2018061), Tencent "Rhinoceros Birds" - Scientific Research Foundation for Young Teachers of Shenzhen University, Guangdong NSF 2017A030312008, Fok Ying-Tong Education Foundation for Young Teachers in Higher Education Institutions of China (No.161064) and GDUPS (2015). Corresponding author is Zhenjiang Li. This poster is related to our recent work [3].

REFERENCES

[1] Nancy Berryman Reese and William D Bandy. 2016. *Joint Range of Motion and Muscle Length Testing-E-Book.* Elsevier Health Sciences.
[2] Sheng Shen, He Wang, and Romit Roy Choudhury. 2016. I am a Smartwatch and I can Track my User's Arm. In *Proc. of ACM MobiSys.*
[3] Liu Yang, Li Zhenjiang, Liu Zhidan, and Wu Kaishun. 2019. Real-time Arm Skeleton Tracking and Gesture Inference Tolerant to Missing Wearable Sensors. In *Proc. of ACM MobiSys.*