I see you: How to improve wearable activity recognition by leveraging information from environmental cameras

Gernot Bahle, Paul Lukowicz
Embedded Intelligence, DFKI, Germany
first.lastname@dfki.de

Kai Kunze, Koichi Kise
Dept. of Comp. Science, Osaka Prefecture Univ., Japan
lastname@m.cs.osakafu-u.ac.jp

Abstract

In this paper we investigate how vision based devices (cameras or the Kinect controller) that happen to be in the users' environment can be used to improve and fine tune on body sensor systems for activity recognition. Thus we imagine a user with his on body activity recognition system passing through a space with a video camera (or a Kinect), picking up some information, and using it to improve his system. The general idea is to correlate an anonymous "stick figure" like description of the motion of a user's body parts provided by the vision system with the sensor signals as a means of analyzing the sensors' properties. In the paper we for example demonstrate how such a correlation can be used to determine, without the need to train any classifiers, on which body part a motion sensor is worn.

1. Introduction

Cameras are increasingly becoming ubiquitous in public spaces. Continuous progress in computer vision has made it possible to spot persons track their motions in real life settings. Additionally, structured light systems such as the Kinect have brought simple 3D motion tracking capability into the private domain.

In this paper we investigated how such video based approaches can support personal, on-body sensor based activity recognition (AR). The basic vision is to facilitate "opportunistic" use of video based devices already there rather than the dedicated design of systems that combine on body sensors with video processing. Thus, we imagine a user with his on body AR system passing through a space with a video camera (or a Kinect), picking up some information, and using it to improve his system. This implies two basic assumptions:

1) For privacy (and sometimes bandwidth) reasons, transmission of the raw video stream is not accessible on wearable devices.
2) No activity and/or user specific training of the video based system can be assumed.

Thus, the best we can expect to acquire from the video system is a set of abstract, general purpose features. If these are abstract enough to not pose a privacy threat, cameras in public spaces could make them available to any device within the local network. But how can on body sensor based AR systems make use of them?

In this paper we propose to use such features for self-calibration of body worn motion sensors rather than for the enhancement of any specific AR task. Our approach is based on two observations:

- "Stick figure" like motion trajectories of body segments (limbs, torso, head etc.) are now easily derived from video based systems (e.g. the Kinect).
- These trajectories can be used as "ground truth" on the motion of individual body parts. By correlating it with the information provided by sensors attached to those body parts we can characterize, calibrate and/or optimize their performance.

These improvements would also persist outside the field of view of the video system, potentially having a long lasting effect on system performance.

1.1. Paper Scope and Contributions

We consider two concrete examples of correlating body motion information derived from a video signal with on body sensor information to improve the on body system: (1) determining the location of the sensors on the user's body and (2) recalibrating smart phone based inertial navigation systems. Since the focus of this work is *not on computer vision* but on methods for effective correlation of the different motion signals, we use a Kinect system to obtain the video based motion information.

Locating the sensors on the users body. On body sensing increasingly relies on sensors integrated in consumer devices (e.g. phones, watches). While attractive regarding wide acceptance, device placement concerns are always an issue. Regarding signal interpretation, whether a phone is in a pocket or an arm holster can make a big difference for a variety of sensors ([10], [22]). Previously, we have shown that a classifier can be trained [11] to recognize the on body location from acceleration signals. In this paper, we show how the location can be recognized *without the need for training* by correlating the signal received from an on body inertial sensor with video-based body part trajectories.

Recalibrating inertial navigation systems. . Using a smart phone in a pocket for indoor navigation (Pedestrian Dead Reckoning, PDR) is attractive and extensively studied [19]. While reasonably accurate over short distances, eventually all systems suffer from drift issues and unbounded accumulation of errors (since the path is the double integral over the acceleration, errors increase exponentially). Thus, PDR is

978-1-4673-5077-8/13/$31.00 ©2013 IEEE

always combined with recalibration using some sort of external information such as RF beacons, map analysis or collaboration between devices [23], [8]. Cameras, especially with a limited field of view, offer another recalibration possibility. Obviously, the user being visible constrains his position to the area covered by the camera. However, persons in the image must be matched to persons using the pdr app. In a public space, face recognition may be neither feasible nor desirable for privacy reasons. We show how to use the correlation between abstract body parts trajectories and motion sensor data from a smart phone to link a particular person in the image to a particular PDR system. This linking could be done on the user's phone, based on features made public by the camera, thus avoiding privacy issues.

1.2. Related Work

Several research work focuses on detecting the on-body placement/orientation of devices just using inertial motion sensors [7], [9], [10]. The most common way to deal with on-body placement issues is to use placement-robust features [13]However, these approaches are limited to very simple activity recognition problems (e.g. modes of locomotion).

Traditional computer vision approaches use one or multiple cameras to localize, track and identify persons [3], [15]. Due to lighting changes in real world environments etc., systems relying on cameras alone have their limitations [15].

Some works combine motion sensors with cameras to identify, localize and track people/objects [1], [20], [18] Plotz et. al. present a method to automatically synchronize sensor data streamed from accelerometers with video cameras [16]. Yet, they introduce special synchronization gestures the users have to perform. Most prominently, Teixeira et. al. identify and localize people by correlating the accelerometer signals from their mobile phones with video from CCTV cameras [21]. Their experiments and methods are quite impressive. They use a mesh of multiple CCTV cameras for localization, identification and tracking on a person level. In contrast, we are not aiming to create a system capable of tracking users. Rather, we try to aid established on body systems embracing both privacy and simplicity concerns.

The bulk of the research uses standard 2D cameras, making the inference task more error-prone. All of these papers just aim at identifying or tracking a particular person, they do not infer on which body part the motion sensor is mounted. They do not focus on using the camera for auto-calibration of a recognition system.

2. Locating sensors using Kinect

Many sensing modalities are very dependent on their on body position. Obviously, the signal of inertial motion sensors varies heavily with the on-body location [9], [10]. However, the on-body location of a device also affects other sensor modalities, from audio over wifi signal strength to GPS signals [6], [22]. Thus, it is of tremendous advantage to easily locate them without the need for previous training and with some robustness concerning their individual placement (i.e. some small rotation or translation should be tolerated). We demonstrate that this is indeed possible using a simple depth camera like the Microsoft Kinect.

We chose the Kinect as it is readily available and already includes a large set of libraries (e.g. translating actors into a skeleton of joints); it is feasible to use any of the multitude of commercially available computer vision systems to achieve the same results.

2.1. Experimental Setup and Dataset

Hardware: we used the XBus sensor platform with 5 connected XSENS sensors as inertial motion system and a Microsoft Kinect as a video system. The XSENS modules were placed at the upper and lower arm, the chest, the side of the lower torso (roughly corresponding to a coat pocket) and on the upper leg. To account for variance in sensor placement, 5 runs were performed, with randomly rotated (in steps of 45 degrees) and slightly translated (randomly moved by about 5cm) sensors. It is important to emphasize that we did not intend to create entirely new placements for each run. Rather, we wanted to demonstrate that one could deal with small shifts that might result from people putting on sensors themselves or sensors shifting during movement. The data recorded by the Kinect consisted of 3D-coordinates for 20 joints using the MS Kinect SDK. Given these joints, it is then possible to represent entire limbs as vectors spanned by subsets of points. The upper arm, e.g., can be represented as the vector between right shoulder and right elbow. From the time series of these vectors, angular velocity or acceleration can be estimated.

With our equipment, the Kinect skeleton model library processed about 30 fps, which provided enough detail for our analysis. While the XBus is capable of up to 100Hz, we set it to the same rate, both to avoid dealing with widely different sampling rates between data sources and to better simulate the capabilities of mobile devices.

In all 5 runs, each subject performed 4 different activities, namely walking, writing on a white board, climbing stairs and using a dishwasher (opening it, taking out and putting in a cup, closing it). All of these could be conceivably spotted by cameras mounted e.g. in hallways, stairwells or conference rooms. All activities were performed five times per run, for a total of 25 times per subject.

Our experiment included 7 subjects, 3 female, 4 male, ranging in age from 23 to 46.

Thus, in total, we recorded 700 individual actions (7 subjects x 5 runs x 4 activities x 5 repetitions) for every sensors.

2.2. Evaluation Approach

We built our evaluation around two premises:

1) Features and techniques used should be as generic as possible, ideally requiring no manual intervention (i.e. no "use this feature for that activity but the other one for ...").
2) Our system should not require training but should be usable as is, out of the box.

As an added difficulty, the data acquired from both sources is not comparable as is. Kinect delivers 3D-coordinates (and thus trajectories in space), while XSens yields acceleration and angular velocity. For the acceleration case, three possibilities present themselves: differentiate the Kinect trajectories 2 times, integrate the XSens acceleration 2 times or meet in the middle (1 differentation and integration).

All three approaches can lead to large errors because of the two linear transformations required. Using angular velocity, the situation is less error prone; one differentiation of the Kinect data leads to angular velocity data that can be directly compared to gyroscope data.

With the above in mind, we explored features ranging from frequency based approaches to matching trajectories in space to comparing changes in angles. Ultimately, the two features that worked best were the change in vertical angle and the variance in horizontal angle. For the Kinect: The vertical angle is given as as $ang_v(t) = \frac{|g(t)*limb(t)|}{|g(t)|*|limb(t)|}$ (with $g(t)$ the vector of gravity and $limb(t)$ a limb given by two Kinect joints) and the change in angle as $ang_v(t+1) - ang_v(t)$. It might seem obvious to use the change in horizontal angle as a second feature. In reality, however, it is a lot more difficult to determine the direction of forward (as compared to "down", i.e. gravity [12]) and thus align the two coordinate systems of Kinect and XSens. Resorting to the variance of horizontal angle eliminates this difficulty, since no absolute orientation is required for it.

For the XSens, both features could be gleaned directly from the gyroscope data, as each XSens has notions of down and forward. This is no contradiction to our previous statement: even though both systems have those notions, the "forward" axes are arbitrary; XSens defines it in relation to "North" gathered from compass data (fraught with error indoors). Kinect simply uses the direction its cameras are facing. While aligning them is theoretically possible, it is difficult and not necessary with our features. It should also be noted that while these two features do not use acceleration, it is very important in determining the vector of gravity [10].

Given these features, a suitable unsupervised technique of matching them between sensors and Kinect was needed. Simply comparing the signal frame by frame (subtraction / correlation) proved inadequate (e.g. due to timing issues). We therefore explored other options and settled on Dynamic Time Warping (DTW). DTW treats both of the signal time axes as dynamic; i.e. a point x_t of signal 1 may be matched to a point y_{t+j} of signal 2, where j is a parameter of the DTW method (the larger, the more distant points in time may be matched). A more detailed summary of DTW can be found in [17]. When applied to two signals, DTW yields both an absolute distance measure as well as the list of timestamps matched to one another. This list is useful on its own: it can e.g. be used to calculate a correlation between signals that incorporates timing issues.

Assembling the steps detailed above yielded this algorithm for each activity:

1) Calculate the change of angle features both for the five Kinect positions as well as for the XSens in question. Each XSens was considered on its own. Doing otherwise would render this a discriminative problem ("which of the 5 sensors is worn where"), which is a simpler subset of the problem we present here.
2) Calculate the DTW distance and timestamp matches for the XSens signal and each of the five Kinect positions.
3) Pick the winner according to the minimal distance found
4) Calculate the correlation between both signals according to the timestamp matches and square it. This serves as a normalized confidence measure later on.
5) Perform a majority decision on the 5 runs done and average the confidence measure.
6) After executing steps 1 to 5 for each activity, pick the one with the highest confidence as global winner.

A note on 4-6: distances vary quite a bit between activities. Correlation alone proved worse for picking a winner. Also, some activities are more suited to recognize some positions than others. Steps 4-6 combine the strength of both measures.

2.3. Results

	Pants	Coat	Chest	Upper arm	Lower arm
Walking	94	40	46	71	77
Writing	54	37	57	94	97
Stairs	94	43	43	63	74
Dishwasher	51	63	60	83	83
Fused	94	53	60	94	97

TABLE 1. Accuracy for each activity by sensor position (all values are percentages)

This algorithm was applied to the entire dataset, resulting in 35 runs (5 repetitions x 7 subjects) for each of the 5 sensor positions. Table 1 lists our results by sensor position. Accuracy is defined as correct results / all. Three locations can be identified very reliably. These are pants (94%), lower arm (97%) and upper arm (94%). For the pants, both walking and climbing stairs are very suited to recognize the position. For lower and upper arm, writing on a white board serves as a very distinguishing task. It is, however, also readily apparent that not all positions can be identified with high accuracy. Both the coat pocket at 53% as well as the chest at 60% perform quite a bit worse than pants, upper arm and lower arm. There are two reasons for this: first, the Kinect provides joints for knee and left / right hip as well as hand, elbow and shoulder. These match the sensor positions quite well. On the other hand, "resolution" is a lot more coarse for the central body, with only HipCenter, Spine and ShoulderCenter available. Second, there is also a lot less movement in the central body when compared to the extremities. Less motion overall necessarily increases noise compared to useful data. Both values, however, are still a fair bit above random guessing, which, for 5 possible sensor locations, would amount to a 20% chance of guessing right.

3. Identifying mobile sensors using Kinect

Next we need to identify the mobile sensors and matching them to other data, e.g. locations. We present a Kinect based system that is able to match data gathered from mobile phones to people passing in front of it.

3.1. Experimental Setup and Dataset

5 subjects (4 male, 1 female) walk randomly between four rooms at three speeds in an office space. A Kinect is mounted

on the ceiling of the hallway connecting these locations. Each subject carries a mobile phone in a pocket of their pants. The device logs acceleration and gyroscope data as well as runs a personal dead reckoning system also developed at our lab [8]. The entire experiment is recorded by video camera as ground truth. All five participants perform the experiment at the same time. Each subject does about 50 walks, for a total of 250 walks. 178 of them passed in front of the Kinect (see Figure 1).

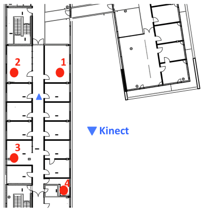

Fig. 1. Matching mobile phones to Kinect data - Setup

3.2. Evaluation Approach

A match between mobile device and Kinect is possible on two levels. Both raw data (acceleration and gyroscope) as well as trajectories generated by the pdr system are feasible candidates. We opt to match on the signal level first for two reasons:

1) Raw data is usually available from most devices; running a pdr requires a dedicated app.
2) Pdr systems become less reliable the longer they run without recalibration. Using positive matches by the Kinect in a feedback loop to the pdr app is interesting, but outside the scope.

We used the features described above and also DTW as a matching algorithm. To be thorough, we also tried to match the pdr and Kinect trajectories. As expected, results were mediocre: 46% (random chance: 20%) of Kinect traces were identified correctly. This showed, however, that there was valuable information in the pdr traces. To utilise it, we used coarse direction as a filter. With a very generous margin for error, we had the pdr system tell us if a person was moving across or along the hallway and if it was moving towards or away from the Kinect, resulting in this algorithm.

1) For each trace, isolate the raw signal and trajectory information from all 5 mobile devices based on timestamps
2) For each of the 5 devices, calculate the change of angle feature and match it by DTW to the Kinect signal.
3) In case there is an almost perfect match, pick that one as the winner. Else, continue.
4) For each of the 5 devices, analyse the pdr trajectory. If the device was not moving along the hallway in the right direction, eliminate it from the pool. Pick the winner based on step 3 distance from the remaining ones.

3.3. Results

Matching phones and Kinect on raw data alone achieved an accuracy of about 73% (random chance at 20%). For an unsupervised method, this is a fair result. Adding the pdr information significantly boosted accuracy to 92% (i.e. 164 of 178 traces were matched correctly).

4. Conclusion

We have presented two examples of leveraging a basic ”stick figure” like description of user motion of a video system to support on body sensing and AR. At the heart of our vision is the notion of ”opportunistically” using devices that happen to be in the users' environment without the need for dedicated training or transmission of privacy sensitive raw images. We believe that given the rising omnipresence of cameras such an approach can have significant benefits, in particular in conjunction with sensors in consumer devices such as phones, music players and watches.

References

[1] M. Aron, G. Simon, and M.-O. Berger. Use of inertial sensors to support video tracking. *Comp. Animation Vrtl. Worlds*, 18(1):57–68, Feb. 2007.

[2] G. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Applications of Comp. Vision, 1998. WACV '98. Proc., 4th IEEE Workshop on*, pages 214 –219, oct 1998.

[3] T. Franke, P. Lukowicz, K. Kunze, and D. Bannach. Can a mobile phone in a pocket reliably recognize ambient sounds? In *ISWC*, pages 161–162. IEEE, Jan. 2009.

[4] A. Henpraserttae, S. Thiemjarus, and S. Marukatat. Accurate activity recognition using a mobile phone regardless of device orientation and location. In *Body Sensor Networks*, pages 41 –46, may 2011.

[5] K. Kloch, P. Lukowicz, and C. Fischer. Collaborative pdr localisation with mobile phones. In *ISWC 2011*, pages 37 –40, june 2011.

[6] K. Kunze and P. Lukowicz. Using acceleration signatures from everyday activities for on-body device location. In *ISWC '07*. IEEE, Oct. 2007.

[7] K. Kunze and P. Lukowicz. Dealing with sensor displacement in motion-based onbody activity recognition systems. In *UbiComp '08*, pages 20–29, Seoul, South Korea, Sept. 2008. ACM.

[8] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster. Where am i: Recog. on-body pos. of wearable sensors. In *LoCA*, pages 264–275, 2005.

[9] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole. Which Way Am I Facing: Inferring Horizontal Device Orientation from an Accelerometer Signal. In *ISWC '09*, Linz, Austria, Sept. 2009. IEEE Computer Society.

[10] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. *Pervasive Computing*, pages 1–16, 2006.

[11] A. Pentland and T. Choudhury. Face recognition for smart environments. *Computer*, 33(2):50–55, 2000.

[12] T. Plotz, C. Chen, N. Hammerla, and G. Abowd. Automatic synch. of wearable sensors and video-cameras for ground truth annot. In *ISWC, 2012 16th Int. Symp. on*, pages 100 –103, june 2012.

[13] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Proc.*, (1):43–49, 1978.

[14] O. Shigeta, S. Kagami, and K. Hashimoto. Identifying a moving object with an accelerometer in a camera view. In *IEEE IROS 2008*, pages 3872 –3877, sept. 2008.

[15] U. Steinhoff and B. Schiele. Dead reck. from the pocket. In *PerCom, 2010 IEEE Int. Conf. on*, pages 162 –170, 29 2010-april 2 2010.

[16] Y. Tao, H. Hu, and H. Zhou. Integration of vision and inertial sensors for 3d arm motion tracking in home-based rehabilitation. *I. Journal of Robotic Research*, 26(6):607–624, 2007.

[17] T. Teixeira, D. Jung, and A. Savvides. Tasking networked cctv cameras and mobile phones to identify and localize multiple people. In *Ubicomp '10*, pages 213–222, New York, NY, USA, 2010. ACM.

[18] C. Vaitl, K. Kunze, and P. Lukowicz. Does On-body Location of a GPS Receiver Matter? . In *BSN*, pages 219–221. IEEE, Jan. 2010.

[19] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *UbiComp '08*, pages 114–123, NY, USA, 2008. ACM.