RWTH Aachen University
Media Computing Group
Prof. Dr. Jan Borchers

Post-Desktop User Interfaces
WS 2006/2007

Real-time motion and activity recognition with sensors

Tim Hemig and Moritz Wittenhagen

November 9, 2006

Advisor: Daniel Spelmezan

Contents

1 Sensors

In the following paragraphs we will explain why we need sensors for real-time activity and movement recognition. After that several kinds of sensors and their current abilities are presented.

1.1 Why Sensors?

First of all there is a necessity for sensors, otherwise the computer or machine that is supposed to process data automatically cannot ”see” what is happening. With the help of sensors side effects of movement or activity in general can be measured electronically and transformed into digital values. Now a CPU can process the data according to the goal of the application.

1.2 Different sensor types

Since there are some major different side effects of movement, there are also different ways to measure them. The natural approach to that would be an optical sensor like a standard digital camera or webcam. This works well, and also comes up with good results, if you use cameras with high framerates and resolution. But you are limited to lab conditions because the imageprocessing still forces the user to stay in a static environment, and because all distances, angles and relations depend too hard on the context or point of view of the camera.

What else can be measured? Every movement is based on accelerating a body part to have it moved into a new position. Acceleration sensors can measure such movement if thy are attached to that body part. If calibrated it can measure the current orientation of the device by measuring the omnipresent gravitation field of earth. Each body part of interest would need at least three such sensors to cover all three dimensions and to have a picture of all body parts at the same time. Current technology provides us with relative small sensor units, so they can be put into wristwatches, cloths and other wearables we are already carrying with us. Unfortunately, the acceleration of human movement may be bigger than the maximum values of most acceleration sensors of adequate size. A sensor should detect up to 12g (1g is one time earth's gravity as reference), but we only found sensors with a maximum range of +-10g (see [7]). In the case of faster movement it is possible that our program gets confused because of wrong values for too fast movements.

Next candidate would be a magnetic sensor. Today's magnetic sensors have a much larger range than needed to measure the exact values of earth's magnetic field. With three orthogonal sensors a vector to the northern pole can be calculated and in comparison to an independent sensor the absolute orientation can be estimated. But not only earth has a magnetic field, many other things in our environment generate a field, or have a natural one, so it comes that magnetic sensors are easily confused by metallic objects or electromagnetic devices like a CRT-Monitor.

Gyroscopes are unaffected by gravity, acceleration or magnetic noise - they can tell you a very accurate information about the orientation everytime, but cannot measure the movement without changing the orientation of that bodypart. E.g., a gyroscope attached to the wrist cannot detect a straight punch without pitch or roll change.

Last known candidate we want to present is the electromyogram, which measures weak electric flow caused by the muscles to detect e.g., a moving arm. Currently this sensors are not perfect too, they get confused by noise from nearby muscles that cannot be distinguished from the actual measured muscle. For activity recognition this may be unimportant, but in general the values fluctuate depending on the user's training (muscle-man vs. geek) and the conductivity of the user's skin/body. At last this sensor gives perfect information about rhythm and exact timing of movements, because it measures the direct cause of the movement.

1.3 Technical details

After we learned what kind of sensors exist we want to lose some words about their hardware details. Starting with the cameras we found a very high level of technology with cameras able to get 1280x1024 pixel sized images with a rate of 500 frames per second. Since these cameras are affordable and meet the requirements - we do not need to talk about high tech machines that can record up to 33000 frames per second (see [8]) - using the 500fps-camera (see [9]) makes it possible to process the images of 6 cameras of that type the same time by a current gaming PC (needs graphic-power to show the results).
Saving and presenting these data is no problem, the amount of data can be processed. Each sensor can be read by over 20-50 times per second, which is enough for human motion. The acceleration sensor uses in general small piezoelectric crystals (see [13]) that build up electrical tension if they are bend physically. This tension may be amplified, but is already a useable signal for usage.
In a magnetic sensor a HAL-sensor (see [10]) is used to measure the magnetic field in one direction. Imagine a cube of metal that has a flow of electrons through it from the left to the right. Now, if there is a static magnetic field e.g., directing from top to bottom, the flowing electrons are influenced by the Lorentz Force and are pressed into your direction, so the cube is polarized and generates a measurable tension between the side facing you and the opposite one. This is our signal for further use.
Gyroscopes are based on the principle of conservation of angular momentum ([11]). According to earth's rotation a gyroscope finds the northpole (the one of earth's rotation axis). So it can provide you with information about the orientation of the attached body part in correlation to the direction of earth's rotation axis.
Last but not least the electromyogram ([12]). Electrical tension is measured between multiple electrodes on the skin near to the meant muscle. When the muscle is tensed, the surrounding neurons fire an electric impulse to animate their contraction. This impulse can be detected by the sensor as small electrical current between two or more electrodes.
We got all important data into a computer, and are able to work with them now.

2 Activity Recognition

2.1 Introduction

For the production of smart devices, knowledge about the current state of a user is important. With this knowledge and a set of user-defined rules a device can assess whether to interrupt the user's current activity. The knowledge can also be used to assist the user in critical situations. For example, there could be a device that calls for help automatically if it registers a car crash and an unconscious driver.
To realize such a system the current state of the user has to be determined from some sensor data, making activity recognition a classification problem. Although classification is a well-studied topic, for example in industrial surface inspection, the different sensortypes and individual differences between users create new problems.
These problems are adressed in several studies leading to very interesting and promising results. Although the studies coincide in their basic strucutre, they differ in their focus. L. Bao and S. Intille for example made a lot of effort to gain realistic data sets to train their classifiers whereas N. Ravi focused on the classification itself and took a good look at several classifier and possibilities to enhance these. The basic structure of the studies looks like this.

- Collection of training data
- Classifier training
- Testing

2.2 Data Collection

To gain useful data for the classifier, the collection of training data has to be considered carefully. The easy way, to just collect data from yourself or some friends in a lab, may be enough for a proof of concept, but it is not sufficient for a significant study. This is due to the difference in peoples behaviour when aware of their actions. Even simple task like "walk" can result in unrealistic data sets because firstly people suddenly start to make very distinctive steps or uncommon arm movements. Secondly, it is also not possible to recreate real-world situations like traffic flow or the need to avoid objects on the street in a lab.
On the other hand, the best way, which would be to collect real data from unaware subjects (naturalistic data), is impossible to achieve. Apart from the obvious problem to place the sensors without people noticing, a researcher would have to follow a subject all the time to note start and stop times of activities. Furthermore, there is no guarantee that all activities occur during the observation. The possibility of providing a questionaire for the day also leads to bad results because of recall errors and imprecise start and stop times.
To solve these problems L. Bao and S. Intille introduced semi-naturalistic data. It is not collected in a lab, and the subject are not under surveillance either. Instead the subjects get the necessary sensors (which should be as inconspicuous as possible) and a list of activities to be performed during the day. Everytime the subjects follow an instruction they have to note their start and stop times. The list does not neccessarily contain the activities of interest but instructions implying these activities. For example the list in the

Bao/Intille study used the instruction "use the web to find out what the world's largest city in terms of population is" instead of "work on a computer".
Figure 1 clearly shows the effect of testing on the training data. In the first row the results show a 95.8% accuracy for lab data (training and testing). In the second row the results dropped to 66.7% after using naturalistic data, either implying a need for different classification techniques or the impossibility to gain accurate results from real data.

[10]	95.8%	ambulation, posture, typing, talking, bicycling	24	L	4	chest, thigh, wrist, forearm
[10]	66.7%	ambulation, posture, typing, talking, bicycling	24	N	4	chest, thigh, wrist, forearm

Figure 2.1: Accuracy drop after training data change (taken from [3])

2.3 Data Processing

After collecting the data it is used to train one or several classifiers. In some cases the data has to be preprocessed before use to improve its quality. So when it was labeled by hand the data around the start and stop times may be cut about several seconds to remove some artifacts occuring because of the user standing still and writing.
Before a classifier can be applied the beginning (or end) of an activity has to be recognized. Usually this is done by detecting differences in the sensor values using Fast Fourier Transformation (FFT).
To solve classification problems there are two different categories of classifiers.
The first category are base-level classifiers. These classifiers work directly on the input.
Some base level classifiers are:

- k-nearest neighbour - a very simple classifier that takes the k nearest samples from the training data and returns the most numerous set. The training of this classifier only consists of storing the training data.

- naive Bayes - this classifier uses Bayes' theorem to compute the propability for an activity given the sensor data. To train the naive classisfier the propabilities of sensor data(-range) given an activity has to be computed from the training data.

- decision trees - a decision tree is a tree where all nodes but the leafs consist of "questions". These can be either simple questions like "Is the sun shining?" or, more interesting for activity recognition, questions like "In which range does the sensor data fall?". The leafs of the tree contain the final classification. A very simple decision tree deciding whether to get up in the morning and go to a lecture could look like in figure 2.2:

  The training process of a decision tree has to decide which question to use on which node. This is done by evaluating the quality of a question by computing it's entropy from the training data. In the example above it would be better to put the second question in the first node if there was a lot of rain during training and an equal distribution of lectures with videos. This is because the data indicates that it does not rain that often, implying that in the example above the algorithm would have

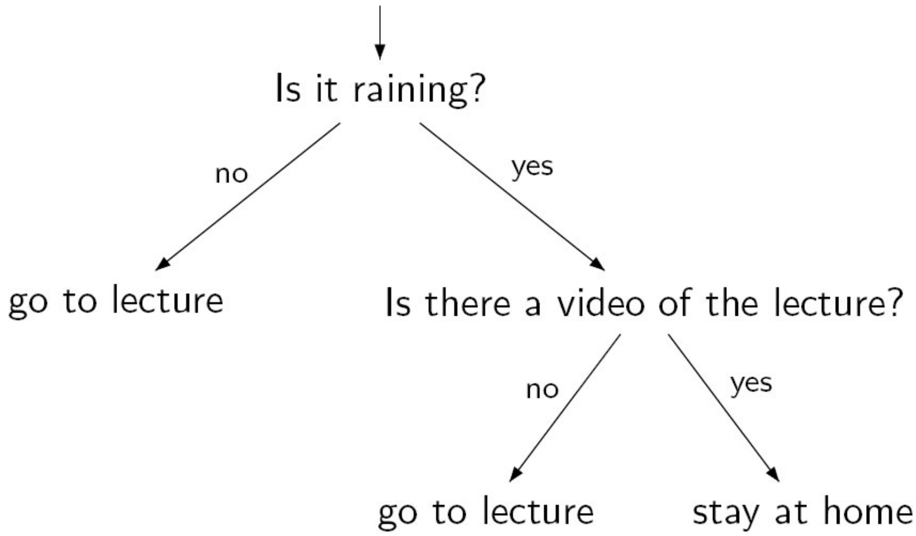

Figure 2.2: A simple decision tree

to traverse two levels of the tree in most cases. The formula to compute the entropy is $H(X) = -\sum_{i=1}^{|Z|} p_i \cdot \log_2(p_i)$ where Z represents the set of possible classification results and p_i the distribution for the current question X.

According to the studies descision trees and the k-nearest neigbor classifier work best for activity recognition.
The second classifier category are meta-classifiers. These classifiers do not work directly with the data but they improve the performace of base-level classifiers. This is either done by enhancing one classifier or by combining several classifier into one better classifier. Some meta-classifiers are:

- bagging - This technique enhances one classifier by creating different data sets from the training data (draw without replacement) and use these in several instances of this classifier. The resulting classifiers are more robust because artifacts are removed in most instances.
- boosting - Another possibility to enhance one classifier. Boosting uses the same learning technique repeatedly and assigns a weight to each sample in the data set depending on whether it was classified correctly.
- plurality voting - This is an easy way to combine several different classifiers. The result of every classifier is computed and the most numerous is taken as overall result.
- meta decision trees - This meta classifiers decides which classifier should be used to yield the best result.

For activity recognition the most accurate meta-classifier is plurality voting which is about 5% more accurate than decision trees. Although meta-classifiers yield better results in general, they are not always applicable in activity recognition. This is because you only have limited computation power and energy on a mobile device. A meta classifier that has to invoke several base level classifiers may be a too large strain on the ressources. Also, base-level classifiers are doing a pretty good job themselves. So the gain of using meta-classifiers might be small.

2.4 Studies

Several studies have been performed in the field of activity recognition. As mentioned above the main difference between the studies is their focus on different aspects of the recogniton process. Other differences include number of subjects or number of activities under examination. Despite these different focuses, the studies lead to the same results. These are that activity recognition is possible and that the decision tree classifier is the best base-level classifier for this problem. Figure 2.1 shows a comparison of several studies and their results: Studies on the subject have been performed by:

- Ref [1]: R. DeVaul, S. Dunn in '01. The study "Real-Time Mition Classification for Wearable Computing Applications" dealt with achieving low energy requirements, which is important on mobile devices. They discovered that easy questions like "Is the user moving?" can be answered by classification of only 1.34 second data windows. For more detailed information on the users activity the results of the first classification can be used to train another classifier.

- Ref [2]: N. Ravi, N. Dandekar, P. Mysore, M. Littman in '05. Their study "Activity Recognition from Accelerometer Data" focused on several different classifiers and classification techniques. Their activity set contained 8 different activities. One triaxial accelerometer worn on the hip was used to collect the data. The study shows that plurality voting is the best meta-classifier, the energy of the movement is less important feature than mean, standard deviation and correlation and that activities can be recognized using only one accelerometer with a missclassification rate about 14%.

- Ref [3]: L.Bao and S.Intille in '04. "Activity Recognition from User-Annotated from Acceleromter Data" dealt a lot with data collection, but they also considered several classifiers and the feature computation. The data was collected from 20 subjects performing 20 activities using 5 accelerometers. Testing was done by training the classifier with the data from 19 subjects and applying this classifier to the last person's data. The Results from this study are that decision trees have an an average accuracy of 84%, but user-specific training might be necessary for individual activities like, for example, "stretching".

Aspects that are missing in the studies include:

- Where to position the sensors and how many sensors are nessecary? This question is adressed in several studies in the way that the data of some sensors is ignored in one or more analyses. The results indicate that even one triaxial acceleromenter worn on the hip might be enough to distinguish most activities. Although at least some information can be derived from the existing data, a study focused on these questions can yield some interesting new information, especially concerning sensor placement.

- How much improvement can be expected from userspecifc data? L. Bao and S. Intille tried to use some of an user's data to train a user-specific classifier. This resulted in better results than from the general data set. The problem is that the amounts of data to train the different classifier was very different. So the validity of this conclusion is limited.

- Are accelerometers really the best data source? The success of accelerometers indicate that they are a pretty good data source, but there is no direct comparison of acceleromters with other sensors. Another question could be whether a mixture of different sensors (like accelerometers and gyroscopes) would lead to a significant increase in classification accuracy.

2.5 Conclusion

Although there is some work to be done to establish activity recognition, the existing studies show promising results. When the training data is collected carefully the classifiers are able to make sense from real world situations. For simple activities generalized training data sets are enough to distinguish activities like walking or vacuuming with about 90% accuracy. For activities that do not include a lot of movement, user specific training might be necessary to achieve similar accuracy rates. The best classifier for the problem are decision trees, which can be used well on mobile devices because they need a lot of computation power for training but not for execution.

3 Motion Recognition

3.1 Motivation

In contrast to activity recognition, which focuses on a series of movements and the interpretation of these movements, motion recognition is mostly interested in one particular movement. In many special fields of user interaction with computers or group interaction, a computer knowing the exact movement of one or all persons can be very useful. Dancers are able to record and review their performance, training martial arts could be supported by a computer giving corrections to executed moves. Basically a computer needs to know what exactly the current movement of a performer is. It needs sensors to capture these movements, either for directly interpreting the data or just for recording it for later use. People can do that by looking at other people and interpret the seen pictures. A computer can do the same procedure by using cameras and algorithms that can detect specific colors. To work around the problems of cameras we mentioned in the sensor-chapter, sensors that are able to produce data everytime and everywhere are used. The best place for sensors is the body of the performing user. The positioning of all or some moveable body parts are most interesting features, so that it seems logical to create or use one type of sensor that can give information about the current orientation, movement (acceleration) and the direction of that movement. With these data from all interesting body parts a computer could create a model of a human representing the current state of movement. The described sensors from chapter 1 are able to provide the data, so we only have to focus on the three following fields of motion recognition:

- Motion transfer
- Motion comparison
- Motion detection

3.2 Motion transfer

Transferring motion to virtual systems is the most basic way to use data from motion capturing systems as described above. For example most movie productions use systems with cameras and suits with LED-markers on it to transfer human gestures and behaviour to animated movie characters (see "interesting projects" later). A model of the character is generated with a basic skeleton similar to the human skeleton. Next step is to find a suitable translation from "human" values to animated values if your character has different proportions than the actor or is presenting a completely different lifeform. Now you can capture scenes with an actor wearing the suit. The recorded data is used to generate movements of the animated character, which can be included into a movie. With the same procedure you can display a real-time avatar of yourself that can even give you additional information like where to go or how to behave in several situations. That way an avatar could correct your movement during a Thai-Chi session by displaying the correct overlay to your own posture, but this is leading towards motion comparison. Motion transfer only includes transferring your movement into another system, which works with it and does not care about the meaning of the movement itself or it's correlation to another known movement.

3.3 Motion comparison

In movement comparison we have a first look on the signal itself and try to interpret some new information automatically to enlarge the amount of data. If we want to compare two or more live acting performers like a group of dancers or a master of martial arts and his student executing a synchronous series of movements, we have to compare the incoming signals to each other on the fly. Assuming that similar or equal movements are executed at almost the same time, we can observe a certain time window of e.g., the last two seconds. Delays of more than two seconds would be obvious to the observant and performer itself and we would not need a computer to tell us that something went wrong during performance. Because of the processing time for one window we cannot really evaluate every window.
We can define a step time after that a new comparison with a new time window is started. In conclusion we now have two or more signals of the corresponding sensors of the same length of time to compare and find similarities between them. Aylward & Paradiso (see [7]) submitted to the NIME '06 the idea of taking the crossvariance of each pair of signals to calculate a similarity. The crossvariance of two signals f and g is defined as

$$(f \star g)(x) \equiv \int f^*(t) g(x+t)\, dt.$$

This operation can be seen as one signal sliding over the other. For every difference of time (step of sliding) we count the similar parts through using the integral properties of the signals. For better visualisation we maybe have a look at an example of using the same principle for Strings:

```
Vector Element|W0(Biology,1Biologv)|W4(...)     |W6(Biology,1Biologv)|
--------------+--------------------+------------+--------------------+
              |           Biology  |   Biology  |      Biology       |
Word Alignment|   1Biologv         |1Biologv    |     1Biologv       |
              |           0        |   00100    |      1111110       |
--------------+--------------------+------------+--------------------+
Score         |           0        |0+0+1+0+0=1 |  1+1+1+1+1+1+0=6   |
```

Figure 3.1: Calculating scores for similarity

As you can see in figure 3.1, the two words "Biology" and "1Biologv" are compared by one sliding under the other character by character. We can see the steps 0, 4 and 6. Each time one letter equals the other one where two letters are one upon the other we add the score of 1. The highest similarity is achieved in step 6, when effectively the substrings "Biolog" are upon each other, which are the most common part of both words. Translating this procedure to signals we would score analog to this example by calculating differences of numeric rows instead of continuous signals. The result is a new signal consisting of the series of scores indicated by the difference of time the two origin signals were shifted. Now we can locale the maximum of this signal and decide if the similarity is high enough to have similar movements. Those signals could look like the ones in the figure (figure 3.2).
The time difference according to the maximum tells us the delay of the first subject's movement (if it gets negative, the second movement is too early instead of too late),

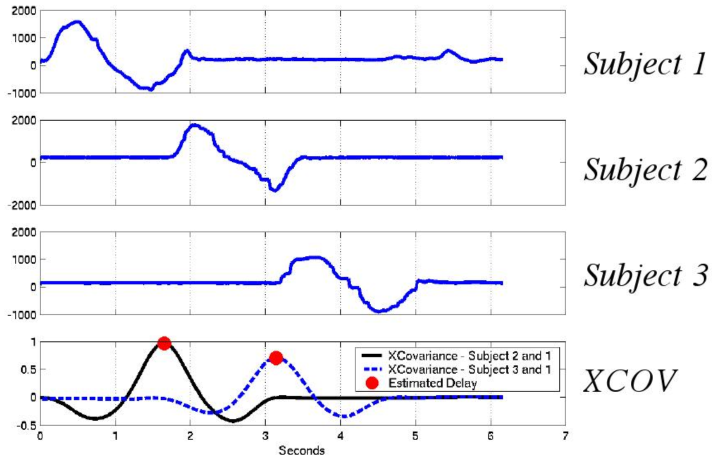

Figure 3.2: Raw data for hands raised and lowered in sequence (taken from [7])

which can be stored, displayed, or communicated to the performers by sound. Mostly such data is needed for analysing bigger performances afterwards like in big groups of dancers achieving synchronous behaviour for all members. Since a human cannot observe so many people, the computer is quite useful here.

3.4 Motion detection

For motion detection we need a definition of similar or equal movements to save known movements. Of course humans of different size and constitution will move exactly the same way, and will never generate equal data with a capturing device, so we need other characteristics of a movement. Measuring the distance between body parts does not work due to known reasons, but an equal angle between each two sensors attached to body parts indicates a similar posture as good as a similar burst of acceleration data could indicate a similar movement. Acceleration sensors provide you with both information in regular circumstances (extreme values can be polished by taking other kind of sensor values into account if the setting needs it). Next we need a algorithm to compare this data to others. While looking after patterns with more or less variance to a particular pattern of movement, such a pattern has to be defined, too. Assuming a single motion consists of three parts like an entrance posture, a gesture, and a ending posture of the body, we may look for values starting and stopping to change with relative high values per time. With a certain threshold for δt we mark the start posture of the movement quite before that value (e.g., last value) and start recording the motion between the two postures. After δt falls back under the threshold we stop recording and take the current values (or one of the next values to give the sensors time to calm down) as the ending posture. Now we gained three important values for comparing signals of movement. Two data sets of all sensors from the postures (static chunks) and the signal of the movement in between (dynamic chunk). All three chunks are used by Kwon&Gross (ETHZ) for analysing movement (see [14]). Following figure 3.3 shows such a series of chunks.

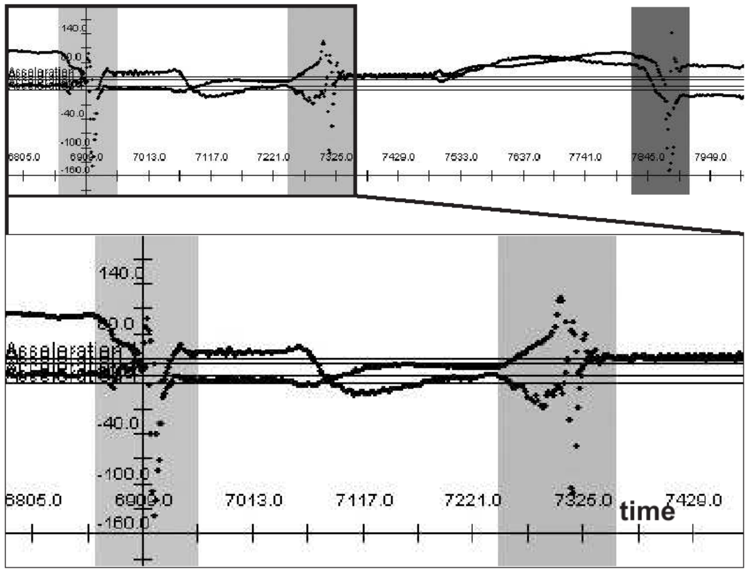

Figure 3.3: Structure of a motion (taken from [14])

With the help of this characteristics and allowing some variance to this values we may look around to other disciplines and search for a suitable solution to the occurring classification problem. Speech recognition is also based on signals. Detecting movements instead of words or phonemes should be the same way to process our data. Since the camera data had no signals like the other sensors yet, we assume that the x-y-coordinates can be represented as two signals over time, and one movement would never consist of only one signal. A complete chunk would consist out of several short signals of the same time window, detected like described before. The Hidden Markov Models (HMM) are used in speech recognition and are meant to solve the problem of single movement classification. A series of movements can be discussed later based on the HMM's results. HMM can be understood as a state based finite automaton with transitions meaning a stochastic event like the comparison of two motion chunks. Like the cross-variance before in motion comparison the stochastic process of the HMM scores the similarity of observed features of the signal (like position of local maximum and minimum in relation to start and end time). Taking the postures as states we now have a complete automaton based recognition algorithm that can recognize a single motion as a transition of one specific state into another by a high scored transition or even a series of movements as a path in this set of states.

To avoid timing differences (e.g., a motion is performed slower than the other but similar, if we do not look on execution time for that case), the signal can be adjusted to the length of the original one. To store some movements we just need to train several movements with perspective to the chosen features of one chunk, and we are able to recognize this motion in the future. Currently such a comparison primary tells you something about similarity, but if you want to know the kind of error your motion had compared to the original one, the difference of signals can be used for avatar-like correction view if you split the information of the body parts and the differing signals to interpret this context. E.g., if a motion met almost all features but the temporal length, the avatar could motivate you to perform this movement faster.

3.5 Interesting projects

At last we want to give a short overview about existing projects using this technique.

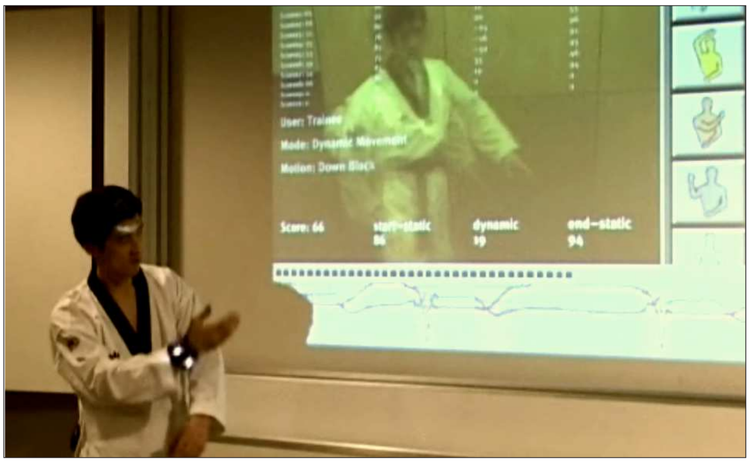

Figure 3.4: Motion training system (taken from [14])

Training Avatar for martial arts by Kwon & Gross (ETHZ) [14]
The group combined cameras and acceleration sensors to have a projected video image for visual feedback. Posture training by comparing the video image and gesture training by using image and acceleration data from the leading wrist and the hip is very useable to perfect the sport. Through the usertesting it came out that users want to use such a system on a regular basis. It helps to concentrate on precise body movement and people were able to achieve training effects after using the system for while. Since this project based on low cost technologies they would be able to create individualized systems with individually trained data.

The "Moven" motion capture suit by Xsens (Enschede, NL) [15]
Xsens constructed a suit (see figure 3.5) with distributed acceleration sensors all over it on key positions for human movement. All sensor units consists of 3D magnetic, angular and acceleration sensors to measure forces, orientation and magnetic field with a samplingrate up to 512 Hz. The whole suit also contains a single sender to broadcast the data to the receiver connected to the software by USB. Everything is running on a standard PC, so with a mobile PC you are just limited by the wireless range of the hardware (50-100m). Recording data to simulate the movement of a body abstracted to 23 movable body parts would support professional training, health-applications (rehabilitation), animation (Movie, PC-Games), and of course you can write you own programmes by using the provided interface class by Xsens.

Figure 3.5: Suit with integrated sensors

Figure 3.6: Camera and LED-marker used by Meta Motion

"Motion Captor" by Meta Motion [9]
Motion Captor is a commercial system of cameras creating real-time 3D data by observing up to two actors dressed with LED marked suits (see figure 3.6). This kind of setting it is meant for lab conditions, 6 cameras have to be arranged and calibrated. Mostly this system is also used for movie or game productions, but also TV stations use it to have animated characters going on in daily shows, or similar - even for live acts. The success is based on the high productivity with real-time motion capturing.

Mobile phones with motion detecting interface by Samsung (Korea) [16]
A complete different field of motion recognition is explored by Samsung. They equipped their mobile phone with angular rate sensors and acceleration sensors to have the user interact with it by moving the phone or just tilting it over into one direction. As example you can enter phone numbers by waving the number into the air (see figure 3.7 and 3.8).

"Sketch Furniture" by Front (SWE) [17]
The Front-team created an environment, where you can paint your idea of an furniture into

Figure 3.7: The SPH-S4000 by Samsung

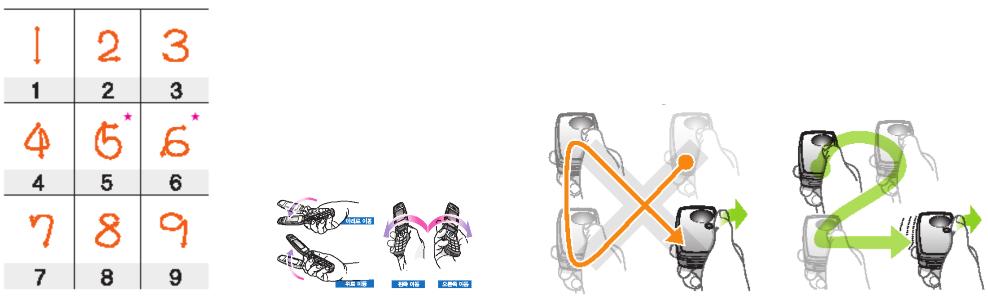

Figure 3.8: Different kinds of interaction with the SPH-S4000

the air. Cameras capture your pointing finger and create a 3D-model of your movement (see figure 3.9). With this data the current state of work can be drawn into the current video-stream to have visual feedback. Afterwards your furniture can be created physically exactly according to your drawing.

3.6 Future & Conclusion

Motion recognition is broadly used by mostly commercial applications. Although sensors already meet the requirements they still are a topic of research, because they can get cheaper, more robust, more exact, smaller, and some of them need to capture a bigger range of values than today. If sensors get available for everyday usage there also will be a need for a software framework to cut down the duration of development. Currently motion recognition is a nearly solved problem that can be understood as a transfer, signal comparison, or classification problem. Surly this problems may have better solutions, but today's know how fits the requirements well.

Figure 3.9: Project setting of "Sketch Furniture" with example results

References

1. Richard W. DeVaul and Steve Dunn, Real-Time Motion Classification for Wearable Computing Applications, Technical report, MIT Media Laboratory

2. Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman, Activity Recognition from Accelerometer Data, Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9 13 2005, Pittsburgh, Pennsylvania, Published by The AAAI Press, Menlo Park, California

3. L. Bao and S. S. Intille, Activity recognition from user-annotated acceleration data, Proceedings of PERVASIVE 2004, vol. LNCS 3001, A. Ferscha and F. Mattern, Eds. Berlin Heidelberg: Springer-Verlag, 2004, pp. 1-17.

4. Scott E. Hudson, James Fogarty, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang, Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study, HCI Institute, Robotics Institute, and School of Design, Carnegie Mellon University, Pittsburgh, PA 15213, USA

5. Entscheidungsbï¿$\frac{1}{2}$me / Decision trees, *http : //de.wikipedia.org/wiki/Entscheidungsbaum*

6. Decision trees and ID3, *http : //dms.irb.hr/tutorial/tut_dtrees.php*

7. Ryan Aylward, Joseph A. Paradiso, Sensemble: A Wireless, Compact, Multi-User Sensor System for Interactive Dance, submission to NIME 06, MIT Media laboratory

8. Olympus Cameras, *http : //www.olympusindustrial.com/*

9. Meta Motion, *http : //www.metamotion.com/*

10. The Hall-Effect sensor, $http://en.wikipedia.org/wiki/Hall_e ffect_s ensor$

11. Gyroscope, *http : //en.wikipedia.org/wiki/Gyroscope*

12. Electromyography, *http : //en.wikipedia.org/wiki/Electromyography*

13. Piezoelectric Sensor, $http://en.wikipedia.org/wiki/Piezoelectric_s ensor$

14. Doo Young Kwon, Markus Gross, Combining Body Sensors and Visual Sensors for Motion Training, ETH Zurich, Switzerland

15. Moven, Full-body motion capturing suit, *http : //www.xsens.com/*

16. Samung SPH-S4000, http:*//www.samsung.com*

17. Sketch Furniture by Front Design, *http : //www.frontdesign.se/sketchfurniture/*