# A Method for Outdoor Skateboarding Video Games


Jan Anlauff
Ambient Intelligence, CITEC

Erik Weitnauer
Neuroinformatics Group, CoR-Lab

Alexander Lehnhardt
Neuroinformatics Group, CITEC

Stefanie Schirmer
Practical Computer Science, CeBiTec

Sebastian Zehe
Ambient Intelligence, CITEC

Keywan Tonekaboni
Ambient Intelligence, CITEC

Bielefeld University, Germany
{janlauff,eweitnau,alenhard,szehe,sschirme,keywan}@techfak.uni-bielefeld.de



## ABSTRACT
Video games aimed at motivating players to exercises have gained popularity over the last few years, but most games are still designed for indoor scenarios. In this paper, we present a platform for a novel game concept: a mobile video game that is controlled by performing tricks on a real skateboard. The platform consists of two parts. A well-protected small wireless sensor module integrated unobtrusively into a skateboard and trick detection software that employs data mining techniques to classify skateboarding tricks from the raw data. We show the feasibility of the approach by presenting Tilt'n'Roll, a prototype skateboarding game application built on this platform.



## Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: Input devices and strategies, user centered design; B.4.2 [**Input / Output and Data Communications**]: Input / Output Devices; K.8 [**Personal Computing**]: Games

## Keywords
Mobile, Gaming, Sports, Digital Entertainment and Sports, New Gaming Audiences, HCI, Real-time Classification


## 1. INTRODUCTION
Video games controlled by player movements mimicking real-world moves gained a lot of popularity over the last few years. Apart from facilitating the players' immersion into the game, coupling physical activity and video games can also help to motivate the players to do physical exercise beneficial for their health. Most prominent movement-based games were developed for game consoles equipped with special controllers, such as the Nintendo Wii.

We present an approach that sets itself apart from current movement-based game concepts in that we use an ordinary piece of sports equipment as input device for an outdoor game. Our prototypical skateboarding game *Tilt'n'Roll* is played by riding a real skateboard and performing tricks to control the game. To this end, we have equipped a regular skateboard with sensors that allow monitoring its motion and detecting performed skateboarding tricks. The game engine runs on a mobile phone, so the game can be played wherever there is a suitable skateboarding spot.


**Figure 1: Our platform, consisting of a skateboard with a sensor module and custom riser pad (only one mounted for comparison) and a smartphone.**

The modifications of the skateboard were done with caution to not change its handling, so the skills acquired in the game are identical to their equivalents in real skateboarding and can therefore be used outside the game. This separates our approach from virtual world physical equipment games like Guitar Hero or Tony Hawk: Ride, which use a plastic guitar with five fret buttons respectively a plastic skateboard without wheels as input devices.

Pasch et al. [7] identified two distinct motivations of players in movement-based games: to *achieve* and to *relax*. Taking the Wii Boxing game as example, they found that players employed different movement strategies to either immerse in the game (relax) or to exploit the controller (achieve). However, when one succeeds in combining the real physical challenge of an outdoor sport like skateboarding with the fascination of video games the two motivations, achieving and relaxing, will blend into each other.

In the remainder of this paper, we first cover related work and then describe our skateboarding game platform consisting of a sensor-equipped skateboard and the software to classify tricks from sensor data. Then we briefly outline the prototype skateboarding game we developed as proof of concept and conclude with a summary and outlook.

# 2. RELATED WORK

There are several different attempts to categorize emerging new game types. Lundgren and Björk describe a number of game mechanics typically used in *computer augmented games* [5]. Florian Müller defines the notion of *exergames* or exertion games as games with an input mechanism that requires players to invest physical exertion [6]. Clearly, a computer augmented skateboarding game would also fall into this category. In the context of professional sports, research is conducted on equipping sports equipment or the athletes themselves with sensors. Reilly et al. present a taxonomy of such augmented sports systems and identify training, safety, and refereeing – but not gaming – as application areas [8]. Spelmezan et al. provide an example for such a system in [9]. Yim and Graham [10] categorize exercise games along the two axes 'user interface' and 'game world' and mark the category of equipment based physical interface games in an augmented reality setting as open for research. Our platform may be used to develop games closing this gap.

# 3. ARCHITECTURE

The underlying idea of the prototype Tilt'n'Roll skateboarding game is to augment the experience of real outdoor skateboarding with elements of modern video games. The skater couples the modified skateboard with a smartphone on which the game is running. Using the graphical game interface, he/she starts one of different game modes. After that, all interaction with the game is done only through the skateboard, so no visual attention is taken away from the actual skateboarding. The player receives points for the tricks performed and all significant information about his/her progress is provided via audio. The phone is placed in the pocket, which is both safer and more convenient, because he/she can use both arms freely. The skateboarding game relies on an architecture consisting of three functional layers:

**Backend** The base layer consists of all components related to sensor data acquisition and data transmission. A *sensor board* is attached below both front and back truck of the skateboard. It contains sensors for measuring linear and rotational acceleration and transmits the acquired data to the mobile phone via Bluetooth.

**Translation Layer** In this layer, the raw sensor data is transformed into game events by means of machine learning and signal processing techniques.

**Frontend** The user interface and game logic form the highest level layer. Game events generated by the translation layer are used to control the game. Generally, all components generating visual and auditory aspects of the game belong to this category.

The components of both translation layer and frontend should run on a mobile phone and should react to the movements of the skateboard. We used the Nokia N900 smartphone as a platform. It has a 600 Mhz ARM processor and 256 MB memory, runs a Linux operating system and is among the more powerful phones at the time of this writing. Still, it puts strong computational limitations on the trick detection and game. In the following, each of the three layer's components will be described in detail.

## 3.1 Sensor Board (Backend)

Modern smartphones incorporate motion sensors, such as accelerometers, which may be used directly in movement-based games. An attempt to detect skateboard moves using only the mobile phone's sensors is not feasible however, since it would require to attach the phone directly to a skateboard, which would not only expose it to physical damage, but also add significant, non-uniform weight that would impact the skateboard handling. Thus, a separate sensing hardware was developed. This allows for reduced the size and higher robustness of the hardware and the incorporation of additional sensing channels that are not provided by the phone's hardware. We decided to transfer the data from the sensor board to the phone via Bluetooth, which can be implemented reliably and is already present in most modern smartphones.

The motions of the skateboard and the skater are measured by several sensors: linear acceleration, rotational velocity and pressure distribution of the skater on the board. The necessary electronics are aggregated on a printed circuit board (PCB), the sensor board, which is small enough to fit underneath the *base plate* of the *trucks*. Two sensor boards are used per skateboard, one under each truck, instead of one, as the skateboard is often rotated around either truck. These additional input channels are beneficial for the trick classification.

The weight distribution of the skater is measured by a custom, paper-based force-sensing resistor (FSR) similar to the technique described by Koehly et al. [3]. The FSRs are built as two long rectangular strips and placed underneath the grip tape on top of the skateboard. They allow to sense if the skater is standing on the board and how he/she is shifting weight between his/her feet. This is important for determining whether the skater landed the trick.

All sensor data is collected by an Atmel 8-Bit microcontroller and transmitted to the smartphone via Bluetooth. The sensor board is powered by a small Lithium Ion Polymer rechargeable battery fitted into the hollow space of the truck base plate, providing a battery life of over two hours of continuous use.

The electronics and power supply have to be fixated and physically protected from impacts and mechanical shocks without significantly changing the skateboard handling. The only place on a skateboard that is not exposed to hits from the skater, various objects or the ground during tricks is between the trucks and the deck. *Riser pads* to increase the distance between the deck and the wheels, in order to improve board handling, exist for regular skateboards. We developed a custom riser pad that holds and protects the PCB and the battery, made out of polyvinyl chloride (PVC).

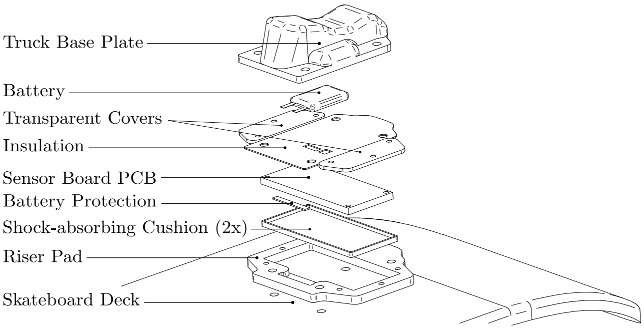


**Figure 2: Exploded assembly drawing of the custom riser pad with the sensor module.**

This mounting, shown in Figure 2, ensures minimal influence on the skateboard handling while providing good protection. As the spacing of the trucks mounting holes is standardized, any skateboard may be equipped with our sensor riser pad. A custom two-piece shock-absorbing cushion was made out of foam rubber to protect the PCB from high frequency and amplitude shocks. Transparent acrylic glass covers close the casing and allow to observe the status LEDs on the PCB.

## 3.2 Trick Detection (Translation Layer)

The trick detection is an essential part of our system, since it is responsible for mapping data from sensor space to skateboarding events like a successfully performed Ollie[1], which can be processed further by the frontend. The mapping task is a two-fold process consisting of detecting the time at which the skater starts to perform a trick (trick segmentation) and deciding on the type of trick after it was finished (trick classification). For the online detection of tricks on the phone Linear Discriminant Analysis (LDA) has been used. This method requires an initial training of the trick models based on sensor data recorded from a skateboarder performing the tricks.

### *3.2.1 Data Recording*

As an initial step, raw sensor values were collected from the skateboard while a skater was performing a set of predefined tricks. A Bluetooth connection between both sensor boards and a laptop was set up and data recording was started manually. Two different tricks (Ollie[1]and Ollie 180[2]) were chosen to be trained since these could be reproduced by our skater reliably. Each trick was repeated 20 times to gather enough training data. After the data collection phase, we ended up with two datasets, each containing 7-dimensional sensor data for the 20 tricks of the respective type. The first three dimensions reflect the the acceleration sensor's X, Y and Z axis, the next 3 dimensions correspond to the gyroscope's angular velocity around X, Y and Z axis while the last channel carries pressure information received from the FSRs underneath the grip tape. Sensors were sampled at approximately 70 Hz and thus acceleration changes up to 35 Hz could be measured.

[1]**Ollie**: Basic skate trick where the back of the skateboard is quickly pushed down, causing the nose to move upwards and resulting in a *jump*.
[2]**Ollie 180**: Same as a *Ollie* but in addition the skater rotates 180 degrees around the vertical body axis.

### *3.2.2 Segmentation*

One of the most crucial parts of the trick detection system is the data segmentation. During this step, the raw data is being processed to extract the characteristic trick acceleration patterns that can be used to identify trick onsets. Since the data consists of a continuous stream, with no markers that mark the onset of tricks being present, a semi-automated method was developed to heuristically detect these onsets. Once all onsets have been found, short time windows of 0.5 s beginning at each onset were extracted and passed on to the subsequent classifier training. To identify those points, channel 2 (corresponding to the vertical acceleration) was manually selected for the Ollie trick as it showed the most significant change when the trick was performed. A first order Coiflet wavelet was used to compute the time scale spectrum for each dataset [1]. Segment points were found by moving a 0.5 s window through the stream. Whenever a window contained two maxima peaks above a certain threshold and their distance to each other was less than 0.3 s, a segment point was added at the position of the first maximum in the window. To reduce the computational load for mobile devices, the wavelet based onset detection was only used for training where accurate onsets are required. When the game is running on the phone, we do not use explicit segmentation and instead regard every time step as potential start of a trick.

### *3.2.3 Training and Classification*

Classification was performed using Linear Discriminant analysis (LDA) [2] on a sliding window of 0.5 s sensor data. To compute a model of the trick, LDA requires to calculate the pooled *within scatter matrix*. For small sample sizes, it is known that the sample covariance estimation is inaccurate and even ill-posed in cases where the number of dimensions exceeds by far the number of observations as is the case with the skateboard tricks.

A popular method to improve the accuracy of the estimation is to use *shrinkage methods*. Covariance matrices were regularized using the lemma proposed by Ledoit and Wolf [4]. The projection onto the normal vector of the separating hyperplane found by LDA results in a scalar value whose sign is used to discriminate between two classes, i.e. Ollie vs. Ollie 180. Since the standard formulation of LDA only handles two-class problems, a *one-vs-rest* scheme is applied which essentially computes a separating hyperplane for each class vs. all other combined classes. In our prototype application's case, two tricks were trained resulting in the classes *Ollie*, *Ollie 180* and *No-Trick*.

When running in online mode, i.e. during a running game session, a 0.5 s window is shifted continuously through the received data stream on a per-sample basis where each single window is projected onto each computed LDA vector. A refractory time for classification outputs was introduced to avoid multiple classifications at trick onsets due to the sliding window approach. We tested the classifier performance in a 10-fold cross-validation using a training set of 20 segmented time windows for each trick class. The continuous data was preprocessed with the segmentation method to extract appropriate time windows for the classification procedure. Offline results for the two tricks versus the the background class (one-vs-rest scheme) are depicted in Table 1.

| Trick | Correct rate | Sensitivity | Specificity |
|---|---|---|---|
| Ollie | 0.96 | 0.97 | 0.95 |
| Ollie 180 | 0.86 | 0.9 | 0.79 |
| No-Trick | 0.86 | 0.78 | 0.96 |

**Table 1: Classification performance obtained in a 10-fold cross-validation.**

A *no-trick* class was created by generating 40 random segment points and taking time windows of 0.5 s starting at these points.

### 3.3 Prototype game application

A game that supplements skateboarding must take little attention away from skateboarding, while still keeping the player informed about its current state. Usually, both arms are used when performing tricks, so the phone cannot be hold in the hand. Our approach to these challenges is to rely on audio feedback to connect the player to the game. The graphical user interface is only needed before and after a game and was designed in a finger-friendly way for to be used through the smartphones touchscreen. So far, we have implemented the single player freestyle mode of the prototype game application Tilt'n'Roll. In this mode, the sensor boards under the skateboard trucks send sensor data to the phone, where the trick detector is running. When the skater performs a trick, the trick detector will ideally detect and classify it correctly and send an according event to the game. For each detected trick, the skater is rewarded with points. During the course of the game, the player enters new levels based on the current number of points. Audio feedback is triggered by several in-game events. On level rise, it reports the new level. On a detected trick, it reports the number of points, along with the trick name, and may report it to the web via a microblogging service such as Twitter. The game ends when the time counter has reached zero. If the score is among the top ten, the skater may enter his/her name for the high score list.

## 4. CONCLUSION

In this paper, we have demonstrated how robust embedded sensor hardware and machine learning methods can be used to realize a computer augmented sporting device. To show the feasibility of our approach, we implemented a working prototypical mobile skateboarding game called Tilt'n'Roll[3]. We consider a major achievement to be the sensor hardware, which is robust enough to withstand extreme forces that occur during skateboarding sessions. Out design also allows seamless integration of the sensors into existing skateboard trucks and does not affect the handling of the skateboard. Furthermore, the trick detection method achieved high recognition rates of more than 90% once enough trick samples had been recorded in the training session. The method is computationally efficient and can run in real-time on modern smartphones.

In future work, we plan to investigate alternative trick classification methods to allow training of the classifier to occur directly on the phone with less training samples, while maintaining or improving the current accuracy. Our Tilt'n'Roll game presents a basic application that allows us to test the underlying hardware and algorithms. We expect to see more advanced game concepts to be built on this platform in the future. These could involve story telling elements, detailed feedback about trick performance to support learning and multiplayer modes to allow players to compete against each other.

Furthermore, the hardware platform presented here can be easily transferred to other applications. Given appropriate mounting, other outdoor sports such as snowboarding or BMX biking may be turned into computer augmented sport games. Certainly, skateboarding is a promising sport to be combined with mobile augmented reality games and refining and extending Tilt'n'Roll according to the requirements of Yim and Graham [10] has a lot of potential. We are confident that this new kind of equipment-based exercise game and the platform presented in this paper can help to address new gaming audiences and may even help to motivate players to learn new skills and increase their physical exercise.

[3] `http://www.youtube.com/watch?v=SFTRoslKSOE`

## 5. ADDITIONAL AUTHORS

Nick Thomas, Sociable Agents Group, Bielefeld University and Florian Fusco, Weißensee Kunsthochschule Berlin.

## 6. REFERENCES

[1] I. Daubechies. *Ten lectures on wavelets*, volume 61. Society for Industrial Mathematics, 1992.

[2] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen*, 7:179–188, 1936.

[3] R. Koehly, D. Curtil, and M. Wanderley. Paper FSRs and latex/fabric traction sensors: methods for the development of home-made touch sensors. In *NIME '06 Proceedings*, pages 230–233, Paris, France, France, 2006. IRCAM.

[4] O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.

[5] S. Lundgren and S. Björk. Game mechanics: Describing computer-augmented games in terms of interaction. In *Proceedings of TIDSE*, 2003.

[6] F. Müller, M. Gibbs, and F. Vetere. Taxonomy of exertion games. In *Proceedings of the 20th Australasian Conference on Computer-Human Interaction: Designing for Habitus and Habitat*, pages 263–266. ACM, 2008.

[7] M. Pasch, N. Bianchi-Berthouze, B. van Dijk, and A. Nijholt. Movement-based sports video games: Investigating motivation and gaming experience. *Entertainment Computing*, 1(2):49–61, 2009.

[8] S. Reilly, P. Barron, V. Cahill, K. Moran, and M. Haahr. A general-purpose taxonomy of computer-augmented sports systems. *Digital Sport for Performance Enhancement and Competitive Evolution: Intelligent Gaming Technologies*, page 19, 2009.

[9] D. Spelmezan, A. Schanowski, and J. Borchers. Wearable automatic feedback devices for physical activities. In *BodyNets '09 Proceedings*, pages 1–8, ICST, Brussels, Belgium, 2009. ICST.

[10] J. Yim and T. C. N. Graham. Using games to increase exercise motivation. In *Future Play '07 Proceedings*, pages 166–173, New York, NY, USA, 2007. ACM.