BRIX - An Easy-to-Use Modular Sensor and Actuator Prototyping Toolkit

Sebastian Zehe
Bielefeld University
Ambient Intelligence Group
Bielefeld, Germany
szehe@techfak.uni-bielefeld.de

Tobias Grosshauser
ETH Zurich, Collegium Helveticum
Electronics Laboratory
Zurich, Switzerland
grosshauser@ife.ee.ethz.ch

Thomas Hermann
Bielefeld University
Ambient Intelligence Group
Bielefeld, Germany
thermann@techfak.uni-bielefeld.de

***Abstract*—In this paper we present *BRIX*, a novel modular hardware prototyping platform for applications in mobile, wearable and stationary sensing, data streaming and feedback. The system consists of three different types of compact stackable modules, which can adapt to various applications and scenarios. The core of BRIX is a base module that contains basic motion sensors, a processor and a wireless interface. A battery module provides power for the system and makes it a mobile device. Different types of extension modules can be stacked onto the base module to extend its scope of functions by sensors, actuators and interactive elements. BRIX allows a very intuitive, inexpensive and expeditious prototyping that does not require knowledge in electronics or hardware design. In an example application, we demonstrate how BRIX can be used to track human body movements.**

***Keywords*-sensor; actuator; toolkit; sensor network; wearable; physical computing; ad-hoc**

I. Introduction

Rapidly decreasing costs along with simultaneously increasing availability of complex sensors and powerful microcontrollers allow their seamless integration into our everyday lives. The omnipresence of this technologies encourages the fusion of stationary computers, mobile devices, wearables and intelligent surroundings towards the concept of *ubiquitous computing* [1]. Smart objects that enrich this computerized world rely on hardware that is capable to sense, act, compute and communicate representing the bridge between software and real world. Though the general concept may occur to remain similar, novel applications often require custom electronics hardware that can not be purchased off the shelf and has to be developed and optimized in order to fit the designated scenario. The final revision of such custom hardware is the result of a time consuming and costly prototyping stage including several revisions. This blocks the flow of a creative development process, especially if application and hardware are not designed by the same person. In order to overcome these difficulties in the prototyping phase, we developed a system that allows researchers to build and modify the hardware for their application regardless of their level of experience in electronics and hardware design.

With the BRIX system we introduce a user-friendly, runtime-configurable, modular hardware prototyping platform for applications in ambient intelligence and ubiquitous computing. BRIX features basic inertial sensors, a wireless interface and slots for extensions in an intuitive-to-use and robust package. As a proof of concept, we have developed eight different extension modules equipped with sensors or actuators that enrich the functional range, making the BRIX kit a powerful and flexible rapid prototyping tool for developers and users. Potential fields of use for the BRIX system range from pure wireless data acquisition over intelligent surroundings to standalone interactive applications.

This paper is structured as follows: In Section II we collect requirements for a prototyping system and compare them to selected present systems. After that, we describe BRIX in detail in Section III, regarding design aspects and technical information. Subsequently in Section IV we present one of our BRIX applications, focusing on how our system adapts to human body motion tracking scenario, followed by a short conclusion of our presented work and a prospect on an improved version of BRIX.

II. System Requirements and State of the Art

In this section, we define requirements for a hardware rapid prototyping system from a user's perspective and present selected existing systems whose scope of functions overlap with these requirements.

A. Requirements for a modular prototyping platform

For users without experience in physical computing, it is hard to develop hardware for a designated application. To address this problem, we summarized requirements for a modular prototyping system that is capable to simplify and accelerate hardware development in fields like human computer interfaces, ambient intelligence and ubiquitous computing.

The ideal system should be in the first place *(a) easy to use* to not unnecessarily restrict the community of users. Secondly, it has to be *(b) modular and easily extensible* to adapt to a wide range of scenarios. In order to be used in mobile and wearable applications it has to be *(c) small and compact* and offer a *(d) wireless interface*. To allow easy integration into existing projects, the system requires a *(e) platform independent host software*. Especially ambient intelligence applications may require a greater number of sensors and actuators widely distributed in a room, therefore the system should be *(f) low-cost and well reproducible*.

B. State of the Art

Many existing systems that are capable of sensing, feedback and data processing already meet a subset of these requirements. In the following, we review selected hardware platforms that are designed for application areas like motion sensing, automation in mechatronics, embedded prototyping, wearable computing, motion sensing as well as performance arts and compare them to the requirements defined above.

1) Arduino: Arduino was developed as an open source project in 2005 to be a low-cost alternative to established physical computing prototyping platforms [2]. It consists of a printed circuit board (PCB) with an *Atmel* microcontroller, a USB interface and connectors that allow to access the controller's input and output pins. An easy-to-use software development kit (SDK) enables even unexperienced programmers to write firmware for the platform, using a C-dialect. Many different libraries written by users and distributed over the Internet simplify the programming process. Since the Arduino itself can only perform calculations, external hardware can be attached either directly to the connectors or as so-called *shields*, PCBs that can be stacked on top of the platform. Different companies offer shields that contain sensors, drivers or communication interfaces. If the scope of functions is increased by these shields, the size of the structure rapidly grows, complicating an integration into mobile or wearable applications.

2) Lego NXT: The NXT was developed by Lego in 2006 as a controller for electromechanical assembly sets [3]. It is a small embedded computer with an ARM central processing unit (CPU) that allows to connect and control several external components such as sensors and actuators. All these components are mechanically compatible to other Lego products. This allows to easily build electromechanical devices that can be either standalone or controlled by a PC using a USB or a Bluetooth interface. A variety of sensors and actuators like servo motors can be connected to the NXT. Several SDKs with different levels of complexity allow users with all levels of programming experience to build their own NXT software. The NXT system is designed for education in mechatronics and robotics and is often used even in engineering classes. All NXT parts are optimized to match the rest of the Lego system, which complicates the mechanical integration for example into smart objects or clothing. Apart from that, the NXT controller is too big and not compact enough for mobile applications.

3) SunSpot: This platform was developed in 2007 by *Sun Labs* [4] as a compact, wireless physical computing platform. SunSpot consists of stackable elements that form a small and powerful wireless prototyping device. The processing board includes an ARM CPU, external flash and SRAM memory as well as an RF transceiver. An optional sensor board is equipped with a 3-axis accelerometer, ambient sensors for temperature and light as well as solder pads that allow the connection of external components. The SDK and the hardware are both designed to handle elaborate tasks, which results in a complex system that is not feasible for simpler projects or for unexperienced users.

4) Ethos: The system was developed by the *Wearable Computing Group* of the ETH Zürich in 2010 [5] and is designed to monitor the orientation of human body segments over a long period of time. It features a 9-degree-of-freedom (DOF) inertial measurement unit (IMU) consisting of an accelerometer, a gyroscope and a magnetometer as well as a PIC microcontroller and an ANT RF transceiver. An integrated SD-card slot allows the continuous recording of sampled data for four days. Different custom shaped cases like a bracelet case optimize the system for wearability. Ethos is a very accurate and highly efficient motion sensing device but its compact design complicates extensions, which makes it hard to adapt the system to applications it was not originally intended for.

5) Eowave Eobody3: The Eobody3 presented by Eowave in 2011 [6] is a modular "plug & play" system that focuses on applications in music and dance performances. On the hardware side, the system consists of a box with multiple inputs and outputs that allows the connection of external sensors and relays, forming a bridge between the sensors/actuators and a PC. The EoBody3 editor allows the user to map the channels of the Eobody3 to MIDI messages, allowing to either control e.g. a virtual musical instrument with the sensors or control the relays with a MIDI device. An older wireless version of the system [7] addresses especially mobile applications like tracking the motion of a dancer. The limitation on the MIDI standard and the mandatory use of the EoBody3 editor complicates the integration of the system into existing projects. Apart from that, all external components have to be connected with cables which results in a setup that is not useful for mobile or wearable applications.

While the existence of a huge variety of hardware platforms clearly demonstrates the need and interest in flexible toolkits, the presented systems do not sufficiently unify the requirements we summarized above. In consequence we developed with BRIX a new design concept which is presented in the following sections.

III. The BRIX System

In this section, we provide a description of the design ideas, the concepts and the technical details of the BRIX modules.

A. Design Concept

Our goal is to abstract complex embedded electronic systems into easy-to-handle modular physical blocks, allowing users to rapidly build a hardware that fits their sensor or actuator application. We chose to embed our electronics into Lego bricks, thereby suggesting to users that they can reconfigure modules as simple as stacking Legos. This metaphor encourages to experiment, discover and motivates a creative prototyping process [8].

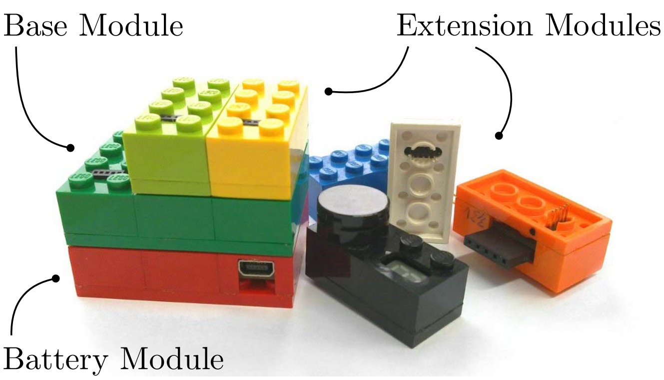


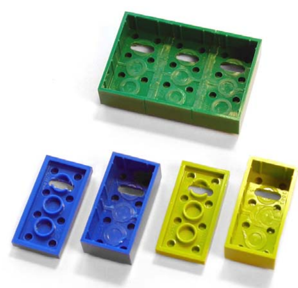
Figure 1: A BRIX stack example and extension modules.

B. System Structure

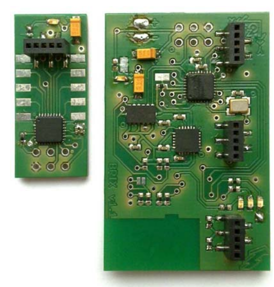
BRIX consists of three different types of modules: base module, battery module and extension modules. These individual parts can be attached to each other, resulting in a fully functional unit, a BRIX stack, see Figure 1.
The base module represents the core of each stack and contains basic motion sensors, a microcontroller and a Bluetooth interface. Furthermore the module offers a connector at the bottom to connect to the battery module as well as three slots on the top side to attach extension modules.
A base module attached to a battery module forms a basic functional BRIX stack, capable of 6-DOF wireless motion sensing. In order to increase the set of functions, the user only needs to attach the corresponding extension module to any of the three slots of the base module, see Figure 1.
We decided to separate battery and base module to allow the user to replace an empty battery module, charge it separately while continuing with a fresh one.

C. Modular Cases for Modular Electronics

Modular systems are a great challenge for the mechanical design of cases and interconnections. We evaluated different connection types like snap-in, Velcro, magnets and friction type connections, which turned out to be most feasible. They are mechanically simple, easy to part and durable. Producing a well working friction type connection system requires a lot of trial and error in order to find the correct machining parameters. Lego has optimized friction type connections to a high level [9], which is hard to achieve with common rapid prototyping techniques like 3D-printing or CNC milling [10] that were available to us. As Lego bricks are very cheap, widely available and easy to modify, we decided to use them to build the BRIX cases. They consist of three basic bricks along with a corresponding plate to close the bottom side. All of these parts are glued together and hollowed partially, which results in a light and robust case with a volume of $48 \times 32 \times 13.5$ mm^3, see Figure 2a (top row).
Cases for single-brick extension modules are even easier to construct, since they require just two Lego parts and a minimum of one cutout, see Figure 2a (second row). This way, we designed inexpensive cases that are simple to make and therefore reproducible with little effort.



(a) Cases made of Lego bricks.



(b) PCB of an extension module and a base module.

Figure 2: BRIX cases and PCBs.

D. Base Module

The base module is the fundamental component of each BRIX stack, containing circuitry for voltage regulation, wireless communication, 6-DOF motion sensing, extension slots and data processing integrated on a single, 44×28 mm^2 PCB, see Figure 2b (right).
We use an *Analog Devices ADXL345* 3-axis accelerometer to measure linear acceleration forces with a maximum precision of 256 LSB/g at a maximum adjustable range of ± 16 g with selectable sampling rates up to 3.2 kHz. To measure angular velocities, we use an *Invensense ITG-3200* 3-axis gyroscope. This device has a maximum measurement range of ± 2000 °/s, a maximum resolution of 16 LSB per °/s and a maximum internal sampling rate of 8 kHz. For self-calibration, the ITG-3200 includes a temperature sensor with a resolution of 280 LSB/°C, which can also be read externally.
Data processing and communication coordination is performed by an *ATmega168* 8-bit microcontroller. The controller is equipped with 16 kB internal flash memory, 1 kB internal random access memory (RAM) and is clocked at

7.3728 MHz. The firmware running on the microcontroller organizes the communication with the internal sensors of the base module as well as the communication with the extension modules and the host system. Additionally it implements a small server application that allows further communication and remote access to configuration features, see Section III-G. The wireless connection to the host system is established by a *Roving Networks RN-41* Bluetooth module, which abstracts the Bluetooth connection to a standard EIA-232 connection. The RN-41 features baud rate speeds up to 921 kbit/s and is rated up to 100 m range, which reduces to around 20 m in most actual application scenarios due to electromagnetic interferences and obstacles.
Three 5-pin female fine pitch pin headers on the base module PCB allow the connection of extension modules that communicate via an I^2C bus. These headers also provide the voltage supply for the extension modules. The base module weights 14.5 g including the PCB and the case.

E. Battery Module

The battery module has the same size as the base module. It contains a 400 mAh single-cell lithium-ion polymer battery with a nominal voltage of 3.7 V, a *Microchip MCP73871* Li-Polymer battery charge management controller and a battery protection circuit. A female pin header sticks out on the upper side of the case and connects to the base module. The battery can be charged via a mini-USB connector on any computer or powered USB hub. Without any connected extension modules, the basic BRIX stack runs 5-8 hours on a single battery charge, depending on the usage conditions. The battery module weights 19.5 g including battery, PCB and the case.

F. Extension Modules

Extension modules are the most important feature of the BRIX system, as they allow the user to rapidly change the capabilities of the stack in order to precisely match it to the desired application, see Figure 1 (right). For our first prototypes, we developed eight different extension modules, providing more sensing capabilities, feedback of different modalities (auditory, visual, tactile) and an interface for attaching external components.
Three sensor modules equipped with either a magnetometer, a barometric pressure sensor or a temperature sensor allow advanced motion sensing as well as ambient sensing. Three feedback modules can be connected to provide the user with visual (full color LED), auditory (piezo speaker) or haptic (vibration motor) feedback, covering the most important modalities.
Furthermore, we developed an interface module that allows the user to change certain parameters of the application at runtime directly on the stack. The module is equipped with a rotary switch and seven-segment display, allowing to manipulate and to display single digit values related to an application parameter.
A special extension module is the I/O-module that enables the user to connect external hardware for instance switches, LEDs, etc. to the BRIX stack with minimal effort. It features 4 analog inputs and a total of 8 general purpose I/Os, as well as a freely accessible programming connector to update the firmware, allowing a quick and fluid prototyping progress.
The extension modules connect to the base module via a 5-pin male fine pitch pin header at their bottom. An optional female header on top allows to stack them onto each other. Our extension modules generally include an *ATmega88* microcontroller that handles two primary tasks on the module: First, it is responsible for reading data from the internal sensors or controlling actuators that are embedded in the module. Secondly, it wraps all of the module functions and forms a consistent interface to the system bus, regarding communication protocol, address spaces and register addresses. This abstraction layer allows the base module to handle every extension module with the same routines, regardless of its function.
We impose minimal constraints on the design of new extension modules to encourage the development of new modules in order to further expand the BRIX set. First, an extension module PCB has to match the Lego grid regarding its size and the position of upper and lower pin header, see Figure 2b (left). This allows to embed it in one of our custom cases and attach it mechanically to the BRIX stack. Our prototype extension modules fit into a single standard Lego brick, but future extension modules could also have a size which is a multiple of that. For instance, should volume be an issue, the designer may as well build an extension module into a case made of two bricks.
Secondly the microcontroller firmware of the extension module has to implement the abstraction layer to the system bus. It is easy to program a new extension module by using a firmware template and adding the desired new functions.

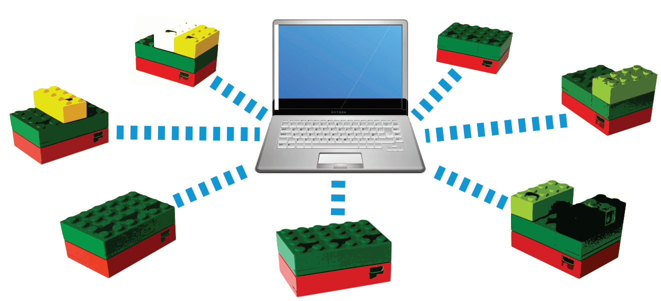


Figure 3: Multiple BRIX stacks communicating with a single host.

G. Communicating with BRIX

All BRIX stacks of an application have to connect to a host system like a PC or a smartphone via Bluetooth. This leads to one or more star-shaped networks, see Figure 3.

BRIX connected via Bluetooth can be handled by the host as a virtual serial port, independent of the operating system of the host. This allows the user to freely choose a programming language for the host application as long as it supports serial ports.
The base module of each BRIX stack offers different modes of operation: For plain data acquisition, all sensor data can either be streamed or polled via the virtual serial port. In the streaming mode, we achieve a sampling rate of around 140 Hz (accelerometer and gyroscope data, human readable)
An interactive server running on the base module, can be accessed via a serial terminal emulator, offering the user to send commands to the extension modules, read sensor values and comfortably configure the BRIX stack. Configuration commands allow to adjust internal features like sampling rates and to activate or deactivate the sensors in the base module. Extension modules can be accessed either manually by sending read or write commands to the server or automatically by modifying the so-called *read vector*. This construct is basically a list of different sources in the BRIX stack to read data from. It contains the accelerometer and gyroscope by default and can be modified using special commands of the server, allowing the user to add for instance a certain register of an extension module. Each time a dataset is either requested by the host or streamed out automatically, it contains all elements of the read vector. Read vector configurations can be stored in the non-volatile flash memory of the microcontroller and are automatically restored at power-on.
The communication interface of BRIX is designed to be on the one hand very simple for applications like pure data acquisition, on the other hand complex enough to allow the precise configuration of the BRIX stack in order to adapt to a desired application.

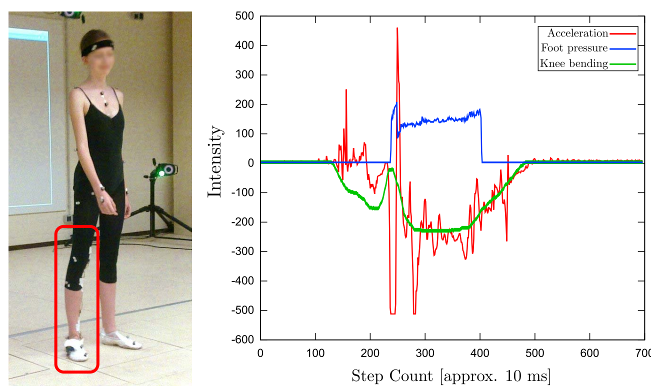
H. Creating BRIX Application Software

Our current early version of BRIX is not designed to allow the user to place own software directly onto the devices. The reason for this is on the one hand the narrow hardware resources of the microcontroller that would unnecessarily limit user software, see III-D. On the other hand, writing firmware for a microcontroller requires special knowledge about the internals of the chip, which would limit the potential user community. A drawback of this is that a BRIX application cannot run standalone and always requires the presence of a host system such as a PC or a smartphone. When developing software for BRIX, the user is free to choose any programming language as long as it supports the use of serial ports, see III-G. Therefore the effort for designing an application depends only on the selected language and the programming experience of the user.

IV. Application

In this section, we present a selected sample application from the field of motion sensing in order to demonstrate BRIX ability to adapt to a designated application.

Dancing requires precise coordination of different movements at the same time. To evaluate the quality of a performance visually with regard to coordination, typically slow motion video recordings are used, because the low temporal resolution of the human eye does not allow to capture rapid movements in sufficient detail. Furthermore it is difficult for humans to focus on multiple distributed motion details at the same time, for instance to rate the synchrony of movements. In light of this disadvantages, we applied BRIX to allow real-time auditory display (sonification) of the motion and pressure data.
In our setup we use specific sensors and actuators for a predefined question regarding the coordination of one leg and foot during a jump sequence. We used a goniometer to measure the knee angle, a pressure sensor fixed on the downside of the foot to measure its contact with the floor as well as a single BRIX stack mounted on the upside of the dancer's foot to measure acceleration data, see Figure 4a.



(a) Dancer equipped with a BRIX stack.

(b) Plot of a data sample recorded during a single jump.

Figure 4: Dancer equipped with BRIX, plot of sensor data.

The two external sensors are connected to the BRIX stack using the analog inputs of the I/O extension module, see Section III-F. Since both sensors are resistance-based, a voltage divider circuit is used to convert the varying resistance to a corresponding voltage. The data are recorded for further offline examinations such as visualizations. Figure 4b depicts a plot of the actual sensor data fragment during a jump. An audio feedback rendered in real-time is provided, based on the live data stream. This allows time effective training and improvements of the movement during the execution by the dancer or the teacher [11].
This experiment shows that BRIX adapts to applications like detailed timing analysis of complex human body motions without much effort regarding time and costs, cf. [12].

The straightforward human readable data format makes it seamless to integrate data streams into other visualizations and processing steps.

V. Conclusion and Future Development

In our sample application, we showed that BRIX is capable to adapt to mobile data acquisition tasks, motion and gesture analysis. Apart from that countless other potential usage scenarios exist in a wide variety of fields. Ambient intelligence for example relies on distributed sensors and actuators in order to create sensitive and responsive smart environments. Transforming a room into such an environment usually involves great effort regarding time, money and knowledge. The BRIX system can be used to equip any room with sensors and actuators within minutes, allowing rapid prototyping of ambient intelligence and ubiquitous computing applications. If a sufficient setup for the desired application has been created, the developer can either continue to use BRIX or design custom hardware based on the experiences gained with the prototype.
Wearable computing applications also profit from the BRIX system. As the sample application presented above shows, the system can be used wirelessly in a highly mobile application and is robust enough to easily withstand rough conditions such as being mounted to the foot of a dancer. This qualifies BRIX to serve as the hardware basis for prototypes of intelligent clothing or smart objects. Such objects can run as a standalone application or interact with other mobile or even stationary BRIX that are for example distributed in the room.

The current BRIX prototype demonstrates the capabilities of a modular prototyping platform. Certainly there is space for further improvements. In future versions, a new smaller case is going to replace the current one in order to increase the mobility and wearability of the system, without sacrificing the modularity and flexibility. To allow applications with a higher complexity we are going to improve the processing power and memory size of the BRIX modules as well as optimize the software framework towards higher usability. This includes on the one hand the possibility to store custom software on the devices in order to run standalone applications without the need for a host computer. On the other hand, we plan to provide an SDK that enables even unexperienced users to write software for the BRIX system.

Acknowledgment

We thank the German Research Foundation (DFG) and the Center of Excellence 277 Cognitive Interaction Technology (CITEC) that enabled this work within the German Excellence Initiative.

References

[1] M. Weiser, “The computer for the 21st century,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, pp. 3–11, July 1999. [Online]. Available: http://doi.acm.org/10.1145/329124.329126

[2] Arduino. (2011) Arduino homepage. [Online]. Available: http://www.arduino.cc/

[3] LEGO. (2011) Lego mindstorms homepage. [Online]. Available: http://mindstorms.lego.com

[4] Oracle. (2011) Sunspot world homepage. [Online]. Available: http://www.sunspotworld.com

[5] H. Harms, O. Amft, R. Winkler, J. Schumm, M. Kusserow, and G. Troester, “Ethos: Miniature orientation sensor for wearable human motion analysis,” in *Sensors, 2010 IEEE*, nov. 2010, pp. 1037 –1042.

[6] E. Gallin and M. Sirguy, “Eobody3: A ready-to-use pre-mapped & multi-protocol sensor interface,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011.

[7] Eowave. (2007) Eobody2 hf wireless sensor interface homepage. [Online]. Available: http://www.eowave.com/products.php?prod=57

[8] T. S. McNerney, “From turtles to tangible programming bricks: explorations in physical language design,” *Personal Ubiquitous Comput.*, vol. 8, pp. 326–337, September 2004. [Online]. Available: http://dx.doi.org/10.1007/s00779-004-0295-6

[9] G. Christiansen, “Toy building brick,” US Patent 3 005 282, October 24, 1961.

[10] C. K. Chua, K. F. Leong, and C. S. Lim, *Rapid Prototyping: Principles and Applications (Book & CD Rom)*. World Scientific Publishing Company, 2010.

[11] T. Hermann, A. Hunt, and J. G. Neuhoff, *The Sonification Handbook*. Logos Verlag, 2011.

[12] T. Hermann and S. Zehe, “Sonified aerobics - interactive sonification of coordinated body movements,” in *The 17th Annual Conference on Auditory Display, Budapest, Hungary 20-24 June, 2011, Proceedings*, 2011.