

AUGMENTED CONDUCTOR

John Anderson Mills III¹, Maria Astrinaki¹, Ricardo Chessini Bose², Loïc Rebousière¹, Todor Todoroff¹

¹ Laboratoire de Théorie des Circuits et Traitement du Signal (TCTS), Université de Mons (UMONS), Belgique

² Service Electronique et Microélectronique (SEMI), Université de Mons (UMONS), Belgique

ABSTRACT

In order to provide better integration of realtime signal processing for live sound generated from acoustic instruments based on the control gestures of a conductor, the *Augmented Conductor* project sought to place sensors on the conductor's arm(s). This project presented many challenges in both hardware and software including creating chains of sensor nodes, improving on previous sensor designs, capturing and processing the sensor data, and gesture recognition. We used sensors and gesture recognition algorithms from previous Numediart projects.

This project required three parallel research paths: sensors, control data processing, and gesture recognition. The previous sensor nodes were improved by reducing their size while increasing their sensitivity using new gyroscope sensors. The data was processed from raw accelerometer, gyroscope, and magnetometer data into more useful arm angles. The gesture recognition software was improved by adding gesture editing capabilities using both video and sensor values. The integration of all of these research paths allowed the improved sensor data to generate more useful sensor information which could then be used to make the gesture recognition more robust. The entire package was delivered to the composer for composition, rehearsal, and, finally, a performance using the sensor nodes and corresponding algorithms.

KEYWORDS

Sensors, gesture recognition, electroacoustic music, musical control

1. INTRODUCTION

The integration of live acoustic instruments and realtime audio signal processing is a long-standing tradition in electroacoustic music. The search for new methods of controlling the processing has led to the invention of a plethora of sensors, controllers, and data processing techniques. The Augmented Conductor project sought to create another new method to control signal processing.

1.1. Motivation

The idea of this project was to use sensors and gesture recognition technology to track the motion of an orchestra conductor. This motion was used to control compositional triggers and sound processing in an electroacoustic composition which mixes live orchestra and live electronic synthesis and sound processing.

The sensor technology involved included previously built sensor devices containing accelerometers, gyroscopes, and magnetometers. Figure 1 shows how these sensors worn by the conductor to provide information for arm movement including the orientation of the upper arm (humerus), orientation of the lower arm (ulna/radius), and orientation of the hand/wrist.



Figure 1: Alex Schubert with attached sensors.

Aside from the sensor system for such data, another challenge in this project was the representation of the raw data for usability and gesture recognition. Because the conductor performed both customary and artificial motions, part of the challenge was to recognize the artificial gestures amongst the more customary conductor movements. In this project, the customary gestures are not of interest for this project, but effectively decrease the signal-to-noise ratio in the gesture recognition, as the two sets of motion can occur in rapid succession.

The major challenges of this project are summarized:

- creating the sensor configuration,
- capturing the sensor data,
- translating the sensor data into a usable format,
- gesture recognition from the sensor data,
- communication with the final composition environment, Max/MSP[3], and
- realtime latency for all data processing.

The main motivation for this project was a musical performance. Alexander Schubert (seen in figure 1), a collaborator and composer experienced with sensor-based composition, received an invitation to create an orchestral piece for the Sonic Visions festival in Karlsruhe, Germany on March 5th, 2011. The piece will be performed by the "Württembergischen Philharmonie." His composition includes artificial gestures made by the conductor for compositional triggers and sound processing.

1.2. Background

There have been many projects which attempt to translate the customary movements of a conductor into useful information. A survey of conducting gesture systems was conducted by Paul Kolesnik in 2004 which identified more than 20 conducting gesture systems including:

- baton-like devices - ex. Radio Baton by Max Mathews[10],
- sensor-based devices - ex. Conductor's Jacket at MIT[11][9],
- video analysis software - ex. a HMM based system by Kolesnik himself[8], and
- software for performance control - ex. prerecorded score performance[4].

Other technologies also apply to this project including hand gesture controllers like Michel Waisvisz's "Hands"[17], and the hand, arm, and leg gesture control system, DIVA[5].

1.2.1. Numediart: FireTraSe

In the Numediart project *FireTraSe* [15], Todor Todoroff et al., created impressive, state-of-the-art sensor devices which include different sensor technologies together in one sensor node with a communications protocol to incorporate all of the sensors nodes. These sensor nodes were the heart of our sensor system.

1.2.2. Alexander Schubert: Laplace Tiger

In his previous performance piece, Laplace Tiger[13], Alexander Schubert used the sensors from a Wii nunchuk game-controller to send a bluetooth stream to a program called OSCulator[2]. OSCulator was used to transmit the sensor data to Max/MSP using the Open Sound Control (OSC) protocol. Custom software was then used to analyze the data and calculate orientation information about a drummer's arms. This data was then used for compositional triggering and sound processing. This experience directly influenced the work on the Augmented Conductor project.

2. THE AUGMENTED CONDUCTOR

Research was conducted in three parallel tracks which were synthesized at the end of the project:

- sensors,
- control data processing, and
- gesture recognition.

The sensors include both the individual sensors as well as how they are connected in chains. The controls and heuristic triggers describes the ways in which the raw data from the sensors was processed in order to gain usable control data. The gesture recognition describes the improvements made to the discrete time warping (DTW) algorithm and the gesture recording software we developed. All software was implemented in the Max/MSP software platform, as this was the target platform for the sensor usage and the performance platform of the composer.

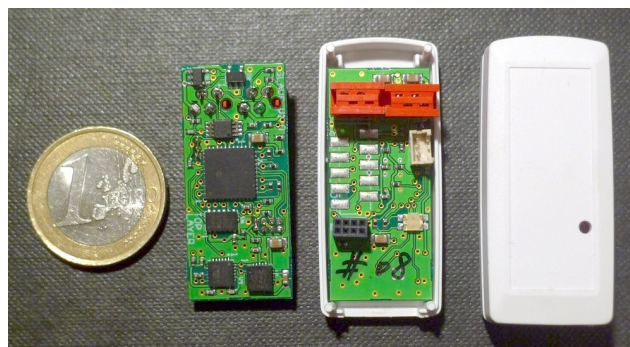


Figure 2: The sensor nodes. The items shown from left to right: 1 Euro coin, top PCB view, bottom PCB view in the lower part of the USB key box, showing Micromatch red connectors for I2C Bus and power supply (empty pads can be used to solder up to 6 additional analog inputs directly on the PCB or using an optional 10-contact Micromatch connector), and box closed with a hole to monitor the state given by a bicolor LED.

2.1. Sensors

The sensor architecture consists of a master and several sensor nodes connected a sensor chain. The master sends information to a host computer while controlling synchronized acquisition of a number of sensor nodes and recovering their data own using an on-body digital bus. The details of the communication paths are described in a previous numediart project, *FireTraSe* [15]. The improvements made to the individual sensor nodes are described in section ??.

2.1.1. Sensor Improvements

The sensor nodes have been redesigned with the help of Multitel Research Centre for larger production. As mentioned in the *FireTraSe* project [15], we were waiting for the availability of the new InvenSense ITG-3200 3-axes digital gyroscopes to make a smaller version of our initial prototype that relied on analog gyroscopes. The resulting design fits on a 17 x 38 mm PCB, which fits inside a small USB key box (see figure 2). As a new boxed sensor node weights only 5 grams, it can closely represent the movements of the limb it is attached to without adding significant additional momentum.

The sensor provides six optional ADC channels, which can be used to connect various additional analog sensors (pressure, flexion, light, etc.). Micromatch connectors are used to easily chain sensors with a standard 4-wire flat cable carrying the power supply line, the ground reference, and the bi-directional I2C clock (SCL) and data (SDA) links.

The final production version of the wireless WiFi master was not completed by the end of the Augmented Conductor project, but will include a Lithium battery and integrated charger. A wired USB version of the master was used in this project when testing the new sensors.

2.1.2. Results

The received data is decoded by the same Max/MSP external object used for *FireTraSe*, which has been updated to accept the increased 16-bit resolution gyroscope data. The user can define a name space for the sensors and calibrate the gains and offsets of the

accelerometers, magnetometers, and gyroscopes by sending messages to that external. The data values are then scaled according to the calibration results (or according to gains and offsets given manually by the user) and the value of each axis is made available through a named Max/MSP receive object. The data values of the sensors are given in meaningful units: g for the accelerometer, Gauss for the magnetometer and deg/s for the gyroscope. Compared to the previous analog version, the increased gyroscope resolution and stability opens new possibilities for analysis.

2.2. Control Data Processing

The data from the sensors in its raw form was inappropriate to use for control signals. The data needed to be processed first to create control signals for use by the composer. These control signals could then be used with value thresholds and time sequencing to create compositional triggers.

2.2.1. Absolute Angle

It has been a broad challenge to combine sensor data from accelerometers, gyroscopes, and magnetometers to effectively acquire the absolute orientation of a circuit board containing these sensors. Several different methods have been employed to varying degrees of success. We chose to implement a self-correcting version of the same accelerometer-gyroscope complementary-filter algorithm [16] used by wheeled inverted pendulum robots.

Accelerometers can be used to determine the direction of gravity when the mounting board is still or moving slowly, but suffer when the mounting board is moved quickly, that is, with high acceleration. Gyroscopes can be used to determine a rotational speed, but because it is necessary to integrate the results in order to determine absolute angle, this angle is inherently flawed over time. The new method employs a similar technique as the wheeled inverted pendulum algorithm.

The wheeled inverted pendulum algorithm mixes the data from the accelerometers and gyroscopes through a crossover filter, using the low frequency data from the accelerometers and the high frequency data from the gyroscopes. For a wheeled inverted pendulum, a crossover frequency of 0.5 to 2.0 Hz is appropriate to keep a robot stable and upright. Unfortunately this frequency range is inappropriate for the type of movements demonstrated by a conductor, and moving the crossover frequency higher causes more problems as the accelerometer data causes late changes to the control values. This late change create a visual-audio disparity that is unappealing for our purposes.

A middle ground solution for us was to follow the gyroscopes unless the gyroscope velocity was under some tunable threshold. In this conditions, the current absolute angle value is changed to meet the accelerator value at a tunable rate. Through the tuning of this heuristic approach, appropriate behavior for the control values was achieved.

During the augmented conductor project, the combined absolute angle algorithm was developed only in 2-D, though as seen in section 2.2.2, this 2-D plane can be chosen relative to the conductor.

For smooth flow of the angle data, a phase unwrapping algorithm was used. This algorithm simply adds the arccosine of the angle of the provided by the cosine between the last two angle samples to the previous angle value.

2.2.2. 3D transformations

By using a series of two 3D rotations, the absolute angle data from the sensor nodes can be converted from the sensor reference into user reference. This process is not perfect, as the accelerometer data is only effectively available in two dimensions of earth reference, but it does create usable, limited control data. The 3D rotation matrix [6] is shown in figure 3.

The process for calibrating a sensor chain from sensor reference to user reference is as follows. The user extends their arm directly forward, and the first calibration is performed. This calibration orients the -Z axis (downward) for all three sensor nodes to gravity according to the accelerometers. The user then extends their arm upwards, and performs the second calibration. This calibration orients the -X axis (forward) for all three sensor nodes toward gravity according to the accelerometers. The conductor can then rotate about a vertical axis with no change to the absolute angle calculated by the combined accelerometer-gyroscope algorithm. The absolute angle of each sensor node is calculated in the X-Z plane relative to the conductor. This method will be sensitive to rotation of the sensors around the arm, but that is acceptable in our performance case.

2.2.3. results

From absolute angle, several other angles can be determined, for example, the angle between two sensors in the chain. The bend of the elbow is an example of a possible control value, as it can be used independently of the absolute orientation of the arm. Also, given the distance between articulations, a sensor node's position relative to another sensor node can be determined. For our purpose, however, angles were of more interest.

2.3. Gesture Recognition

The gesture recognition is performed by the implementation of the DTW algorithm first developed for the *Dancing Viola* [14] project as a Max/MSP external object. Details of this implementation can be found in [14] and [1]. The feature we wanted to add in the *Augmented Conductor* project was a visualization and editing tool for the gestures. A first step towards this feature was done during the *Bodily Benchmark* project [7], and includes updates to the DTW external.

In that project, our purpose was to build a reliable data acquisition tool where the different types of data were time-synchronized (sensors, front and top video recordings, and video analysis data). The Sound Description Interchange format (SDIF) file format was chosen as a way to store the time-tagged values.

Our purpose for the DTW external was to allow the performer to record a video of a gesture at the same time as recording sensor signals, and then to edit the gesture with the corresponding video. Three main elements are then part of the editing tool:

- video playback,
- sensors signals visualization, and
- refinement of the beginning and ending point of the gestures.

The two main benefits of this tool are an easy way for the performer both to replay and remember the recorded gesture, and to refine the beginning and ending points of the gestures. The refinement of the beginning and ending points prevents the algorithm from making useless calculations on sensor values which are not part of a gesture.

$$R = \begin{bmatrix} \cos \theta + u_x^2 (1 - \cos \theta) & u_x u_y (1 - \cos \theta) - u_z \sin \theta & u_x u_z (1 - \cos \theta) + u_y \sin \theta \\ u_y u_x (1 - \cos \theta) + u_z \sin \theta & \cos \theta + u_y^2 (1 - \cos \theta) & u_y u_z (1 - \cos \theta) - u_x \sin \theta \\ u_z u_x (1 - \cos \theta) - u_y \sin \theta & u_z u_y (1 - \cos \theta) + u_x \sin \theta & \cos \theta + u_z^2 (1 - \cos \theta) \end{bmatrix}$$

Figure 3: 3D rotation (θ) matrix around an axis (u_x, u_y, u_z).

2.3.1. Method

For the *Augmented Conductor* project we did not need to synchronize many different sources of media. Only the playback of the sensor values needed to be synchronized to the video of the gesture. As all the sensor values were synchronized together during the gesture recording process of the DTW external, a time-tagged file format, as SDIF, was not needed, and a custom DTW external file format was chosen.

A Java external object was written to read and retrieve the content of the DTW reference gesture files. We used the FTM[12] framework (as in the *Bodily Benchmark* project) as it provided the tools needed to store matrices of values and retrieve these values easily.

Once a DTW reference gesture file is loaded in the external, the file is read, and all of the data is output to matrices. The global file information is stored in one matrix, the gesture information in another matrix, and values of each gesture in a dictionary of matrices. Once this step is completed, the desired gesture can be selected from a list, and the sensor values and the corresponding video are displayed. Only one sensor node's axis is shown at a time in order to avoid display clutter. By dragging the slider object above the video, beginning and ending points can be set. The final step is then to send the points to the DTW object, so that the algorithm can take them into account. In order that these points are used by the DTW algorithm several improvements of the external were necessary. The gesture recording tool can be seen in figure 4.

2.3.2. DTW Max/MSP object updates

We made several improvements to the DTW external Max/MSP object developed for the *Dancing Viola* project. In order to allow a much greater flexibility in the use of the object, all global and local constraint parameters (*epsilon i1*, *epsilon j1*, *lambda min*, *lambda max*) may now be modified in real time as all memory (re-)allocations are now done on the fly whenever a parameter is changed. The (re-)allocations used to happen only when reading a reference gesture file.

This change also is in effect for the hop size of the multi-grid implementation and for two new parameters: *Jmin* and *Jmax*. Those parameters were added so that the beginning and ending points of each reference gesture may be dynamically adjusted with the help of the visualization tools described in section 2.3. These parameters are stored with the reference gesture file.

We felt there was a need for better tools to organize the database of reference gestures, therefore, the previous message *store_in_ref_gesture i*, which could only store the last recorded gesture in a memory location, has been replaced by the more flexible *copy_ref_gesture i j* message that allows one to copy any reference gesture, including the last recorded one, from one memory location to another. And as the name suggests, the message *swap_ref_gestures i j* swaps the content of two memory locations.

The combination of both messages allows the reorganization of the data base at will. *copy_ref_gesture i j* can also be used to run the DTW algorithm with identical reference gestures with different start and end points, giving the possibility to test which segmentation gives the best recognition rate.

Reference gesture recordings are now tagged with date and time. More flexibility has been given to the handling of reference gesture files, as the number of input dimensions (sensing axes or derived analysis) may now differ between the Max/MSP external settings and the file being read. The extra axes are discarded and the missing ones are filled with zeros.

2.3.3. Results

This first version of the editing and visualization tool for recorded gesture has been made as a Max/MSP external for speed of development. In further developments, we plan to integrate it more directly into the actual DTW object for ease of use. The primary purpose of using redefined beginning and ending points was to prevent the algorithm from performing useless computation. It has to be noted that no study on how much the use of redefined beginning and ending points helps the recognition has been done yet.

2.4. Integration

During a final workshop in Hamburg, Germany which took place 15–17 December, 2010, the different systems were integrated together. The final sensor configuration of one master and two sensor nodes was built. The data from these sensors were collected and processed by custom Max/MSP patches to create control data. This control data was then provided to the DTW gesture recognition externals in order to create reproducible user-referenced gestures.

3. RESULTS

At the end of the project the following were delivered to Alexander Schubert in order that he could to create, practice, and perform his composition:

- one chain of one master and two sensor nodes
- sensor translation algorithms
- gesture recognition algorithms

Unfortunately, reasearch and development of prototype sensor nodes and corresponding algorithms does not necessarily produce a chain of sensor nodes which has the robustness to be used in public concert settings. Our desire was, of course, that the newest sensors with the newest algorithms would be able to be used, but due to reliability issues, Alexander chose to work with our older, but more reliable, set of sensors.

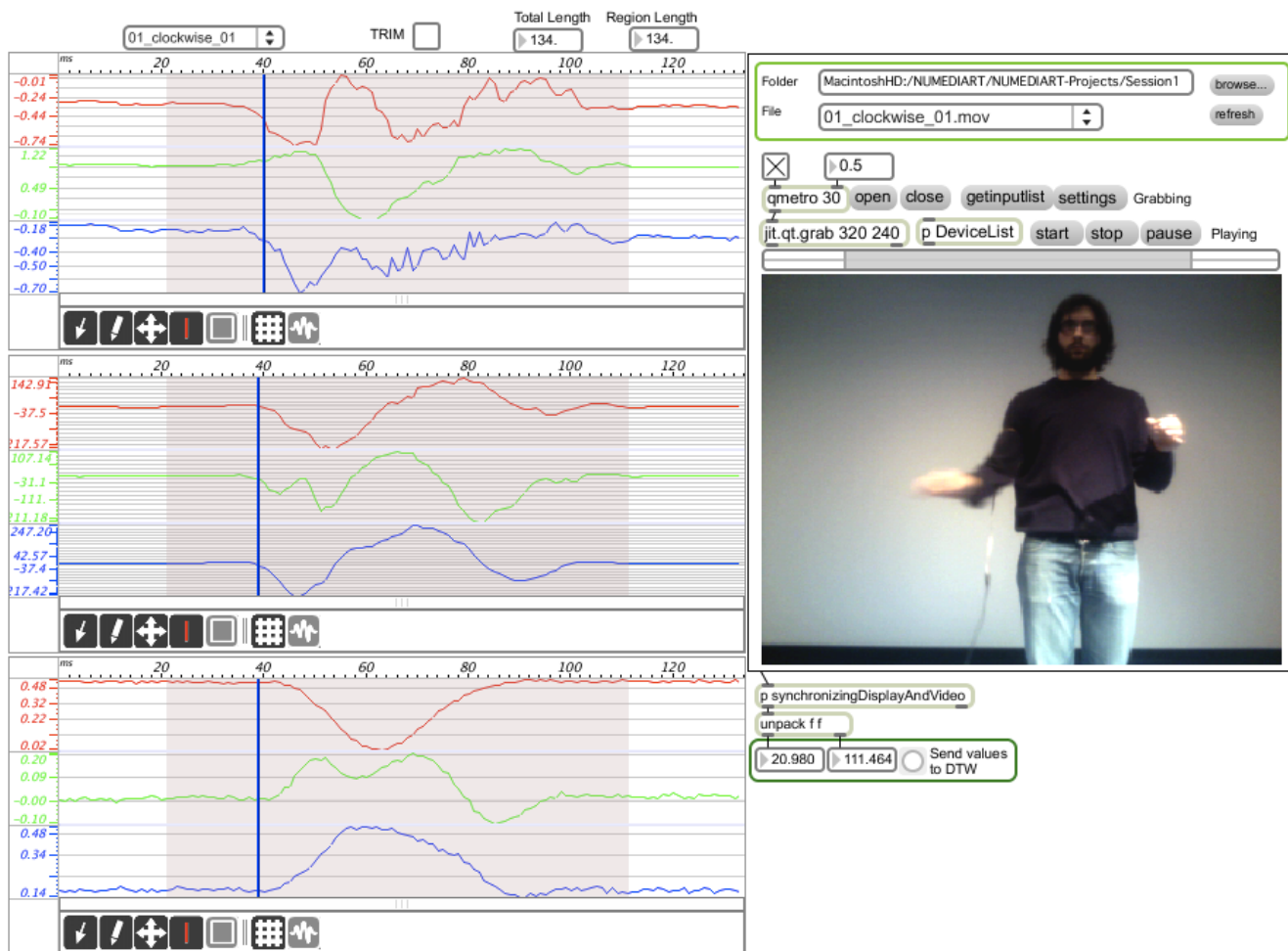


Figure 4: Gesture recording tool.

4. CONCLUSIONS AND FUTURE WORK

The *Augmented Conductor* project did help to improve the body sensor nodes developed at Numediart. The project provided several Max/MSP patches which translated the incoming raw sensor data into usable information about the orientation of the conductor's arms and corresponding differences in angles. The project also improved on the DTW gesture recognition algorithms developed earlier, especially with regard to editing of gestures and providing user-referenced data to the DTW algorithm. Due to the impressive design of the sensors and the power of modern processors, latency has not been a significant concern during the project. The development of the trigger algorithms and the mapping of control values was left to the composer.

The sensor nodes are still being developed, with improvements to sampling rate and reliability. The sensor masters which have rechargeable batteries and WiFi are also still in development. We will continue to improve the data processing and DTW gesture recognition algorithms as well. It should be noted that no testing was done yet on the accuracy in separation of the customary and artificial gestures, and this is a matter for further study.

5. ACKNOWLEDGMENTS

numediart is a long-term research program centered on Digital Media Arts, funded by Région Wallonne, Belgium (grant N°716631).

6. REFERENCES

6.1. Software and technologies

- [1] Frédéric Bettens and Todor Todoroff. "Real-time DTW-based Gesture Recognition External Object for Max/MSP and PureData". In: *6th Sound and Music Computing Conference (SMC 2009)*. Porto, Portugal, July 2009. URL: <http://smc2009.smcnetwork.org/programme/pdfs/321.pdf>. P.: 73.
- [2] Andy Chung. "OSCulator". URL: <http://www.osculator.net/>. P.: 72.
- [3] Cycling'74. "Max/MSP". URL: <http://www.cycling74.com>. P.: 71.

- [4] Roger B Dannenberg and Kenneth Bookstein. “Practical Aspects of a Midi Conducting Program”. In: *1991 International Computer Music Conference*. International Computer Music Association, 1991. Pp. 537–540. P.: 72.
- [5] Sidney Fels, Bob Pritchard, and Allison Lenters. “For-Touch: a Wearable Digital Ventriloquized Actor”. In: *New Interfaces for Musical Expression (NIME)*. 2009. Pp. 274–275. P.: 72.
- [6] James D. Foley et al. *Computer Graphics: Principles and Practice*. 2nd ed. Boston, USA: Addison-Wesley, 1997. ISBN: 978-0201848403. P.: 73.
- [7] Christian Frisson et al. “Bodily Benchmark: Gestural/Physiological Analysis by Remote/Wearable Sensing”. In: *QPSR of the numediart research program*. Ed. by Thierry Dutoit and Benoît Macq. Vol. 2. 2. numediart Research Program on Digital Art Technologies. 2009. Pp. 41–57. URL: http://www.numediart.org/docs/numediart_2009_s06_p3_report.pdf. P.: 73.
- [8] Paul Kolesnik. “Conducting Gesture Recognition, Analysis and Performance System”. MA thesis. McGill University, 2004. P.: 72.
- [9] Teresa Marrin and Rosalind Picard. “Analysis of Affective Musical Expression with the Conductor’s Jacket”. In: *XII Colloquium for Musical Informatics*. Press. Page, 1998. Pp. 61–64. P.: 72.
- [10] Max V. Mathews. “The Radio Baton and Conductor Program, or: Pitch, the Most Important and Least Expressive Part of Music”. In: *Computer Music Journal* 15.4 (1991). Pp. 37–46. P.: 72.
- [11] T.M. Nakra. “Inside the Conductor’s Jacket: Analysis, Interpretation and Musical Synthesis of Expressive Gesture”. PhD thesis. Massachusetts Institute of Technology, 2000. P.: 72.
- [12] Norbert Schnell et al. “FTM — Complex data structures for Max”. In: *International Computer Music Conference (ICMC)*. Barcelona, Spain 2005. URL: <http://articles.ircam.fr/textes/Schnell105a/>. P.: 74.
- [13] Alexander Schubert. *Laplace Tiger*. <http://www.alexanderschubert.net/>. Feb. 2011. P.: 72.
- [14] Todor Todoroff et al. “Dancing Viola”. In: *QPSR of the numediart research program*. Ed. by Thierry Dutoit and Benoît Macq. Vol. 1. 4. numediart Research Program on Digital Art Technologies. 2008. Pp. 129–146. URL: http://www.numediart.org/docs/numediart_2008_s04_p2_report.pdf. P.: 73.
- [15] Todor Todoroff et al. “FireTraSe: Stereoscopic camera tracking and wireless wearable sensors system for interactive dance performances - Application to “Fire Experiences and Projections””. In: *QPSR of the numediart research program*. Ed. by Thierry Dutoit and Benoît Macq. Vol. 3. 1. numediart Research Program on Digital Art Technologies. 2010. Pp. 9–24. URL: http://www.numediart.org/docs/numediart_2010_s09_p3_report.pdf. P.: 72.
- [16] Jr. W. T. Higgins. “A Comparison of Complementary and Kalman Filtering”. In: *IEEE Transactions on Aerospace and Electronic Systems* AES-11.3 (May 1975). Pp. 321–325. P.: 73.
- [17] Michel Waisvisz. “The Hands”. URL: <http://crackle.org/TheHands.htm>. P.: 72.