Motion-sound Interaction Using Sonification based on Motiongrams

Alexander Refsum Jensenius
University of Oslo, Department of Musicology
fourMs - Music, Mind, Motion, Machines
Oslo, Norway
Email: a.r.jensenius@imv.uio.no

***Abstract*—The paper presents a method for sonification of human body motion based on motiongrams. Motiongrams show the spatiotemporal development of body motion by plotting average matrices of motion images over time. The resultant visual representation resembles spectrograms, and is treated as such by the new sonifyer module for Jamoma for Max, which turns motiongrams into sound by reading a part of the matrix and passing it on to an oscillator bank. The method is surprisingly simple, and has proven to be useful for analytical applications and in interactive music systems.**

Keywords-sonification; motion; motiongram; jamoma.

I. INTRODUCTION

Motion and sound are closely linked in the real world, but not always so in interactive systems. Even though the awareness of sound has grown steadily since the early experiments on sonic interaction by, e.g., Gaver [1], [2], it is first in the last decade that the field of *sonic interaction design* has emerged as an established research field and design direction, as documented in, e.g., [3], [4].

A core challenge in sonic interaction design is to understand more about the relationships between action and sound, i.e., what types of sounds fit with what types of actions [5]. In the physical world, actions involving objects will always lead to some kind of sonic feedback dependent on the mechanical and acoustic properties of the actions and objects involved. Furthermore, there are countless examples of how motion and sound are part of a feedback cycle, where sound may again lead to action (e.g., dancing). In electronic devices, on the other hand, the sonic feedback (if there is one) is designed and constructed either mechanically or electroacoustically.

This paper will present one approach to understanding more about the interaction between motion and sound, and a method that can be used in the design process of interactive systems. The method is based on *sonification*, the representation of numerical data in an auditory form [6], of body motion captured using a regular video camera. Such an exploration of how it is possible to "translate" from motion to sound, or sound to motion, may give valuable insights into our multimodal cognition of both motion and sound, and may also be the starting point for explorations of systems using such relationships between motion and sound for various types of interaction.

The starting point for the paper was the observation that *motiongrams* (see Section III for an explanation) visually resemble spectrograms. I was therefore interested in exploring what would happen if motiongrams were turned into sound, as if they had been a spectrogram. The study has two aims:

- exploring how sound can be used in the analysis of music-related body motion
- exploring how sonification of body motion can be used in interactive systems

The paper starts with an overview of some related research. Then motiongrams are introduced, followed by an explanation of how motiongrams can be used to create sound. Finally, some examples of both analytical and interactive applications are presented and discussed.

II. BACKGROUND

My approach to turning video images of body motion into sound is based on what could be called an "inverse spectrogram" technique. This was most directly inspired by the work on image *scanning* and *probing*, as proposed by Yeo and Berger [7], where an image is transferred into sound with frequency on the Y-axis and time on the X-axis. These techniques were later developed into *raster scanning* and the creation of *rastrograms* in [8].

The idea of translating an image into sound is not new. The perhaps earliest example of using a spectrogram-like approach to sonification was the *Pattern Playback* machine built in the late 1940s by speech researcher Franklin S. Cooper [9]. This system made it possible to "draw" shapes that could afterwards be played back as sound. The UPIC system by Iannis Xenakis, developed in 1977, made it possible to use a digital pen to draw on a computer screen [10]. This approach is nowadays available in the Metasynth software [11], and a simplified version in the demo patch *Additive Synthesis* shipping with the graphical programming environment Max/MSP/Jitter. An augmented reality version of the same idea was used in Golan Levin's *Scrappler* [12], where objects put on a table are tracked using computer vision and used to control the sound synthesis.

Parallel to this development, and closer to my own approach, are the many attempts at creating systems for controlling sound through movement, e.g., in interactive art. An early example here is that of Erkki Kurenniemi's

 ISBN: 978-1-61208-177-9

electronic music instrument Dimi-O (1971), using video input for controlling the sound synthesis [13]. Other notable examples include David Rokeby's *Very Nervous System* (1982-1991) and SoftVNS, both of which have been used in a number of interactive installations and dance performances. In the last decade, the availability of graphical programming environments like EyesWeb, Isadora, and Max/MSP/Jitter, have made it possible for artists to easily set up interactive systems based on video input. Many of these systems use motion detection to control either sound synthesis or samplers in realtime. One such example is Pelletier's direct mapping of motion flow fields to sound [14], and subsequent motion-sound mappings using Gestalt-based feature extraction [15].

Also related to my work, but starting from a different premise, is the sonification of clarinetists' performance actions [16], [17]. These projects use data from a marker-based infrared motion capture system as the point of departure for the sonification, something which makes it possible to select specific points on the body/instrument to sonify. As such, it is a more specialized technique than what I am proposing, but it still shows some of the potential in a successful sonification process of motion to sound.

III. MOTIONGRAMS

An overview of creating a motiongram is shown in Figure 1. The process starts by reading a video stream and converting it into a greyscale image. In future research it would be interesting to also use the color information, but the current exploration has been done with greyscale images only. It may also be useful to do some simple image adjustments at this stage, e.g., changing the brightness and contrast, so that the video used for further analysis is as clear as possible (Figure 1.2).

The next step involves producing the *motion image* by calculating the absolute frame difference between subsequent video frames (Figure 1.3). Dependent on the quality of the original image, and the noise level in the image due to video compression, lighting, etc., it may be necessary to filter the motion image (Figure 1.4). This can be done through simple thresholding, or applying a noise removal algorithm to remove groups of few pixels. The motiongram is created by calculating the normalized mean value for each row in the motion image (Figure 1.5). This means that for each image matrix of size MxN, a $1xN$ matrix is calculated. Drawing these 1 pixel wide "stripes" next to each other over time results in a horizontal motiongram (Figure 1.6).

As opposed to a spectrogram in which the intensity in the plot is used to show the energy level of the frequency bands, a motiongram is simply a reduced display of a series of motion images. There is no analysis being done, the creation process is only based on a simple reduction algorithm. This has made the technique very useful in many different applications, as has been summarized in [5].

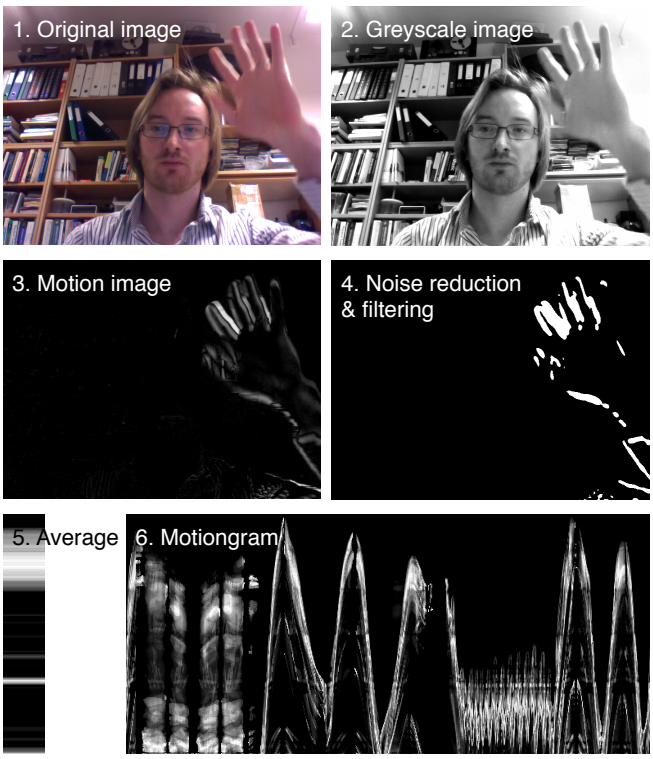

Figure 1. The steps involved in creating a motiongram: greyscale conversion (2), frame differencing (3), filtering (4), averaging (5) and plotting over time (6).

It is worth mentioning that a motiongram will only display motion in *one* dimension. Thus a horizontal motiongram visualizes only vertical motion, since all information about the spatial distribution of motion in the horizontal plane is represented by only 1 pixel for each row. When creating motiongrams it is therefore necessary to evaluate in which plane(s) the motion is occurring, before deciding whether to create a horizontal or a vertical motiongram (or both).

IV. FROM MOTION TO SOUND

Since motiongrams share many visual properties with spectrograms, I was interested to see how they could be used as the basis for sonification of motion. The most obvious way of doing this is by treating the motiongram as a spectrogram, as suggested by Yeo and Berger in their scanning approach mentioned in Section II [7]. This way we can create a direct mapping from motiongram to spectrogram, as illustrated in Figure 2.

A minimal implementation of such an "inverse spectrogram" technique in the graphical programming environment Max/MSP/Jitter can be seen in Figure 3. The implementation is based on reading one line at a time from the motiongram matrix and turning this into an audio signal using the **jit.peek~** object. This is then sent to an interpolated oscillator bank (**ioscbank~**), which does the additive synthesis. The result is a direct sonification of the motion, where lower

 ISBN: 978-1-61208-177-9

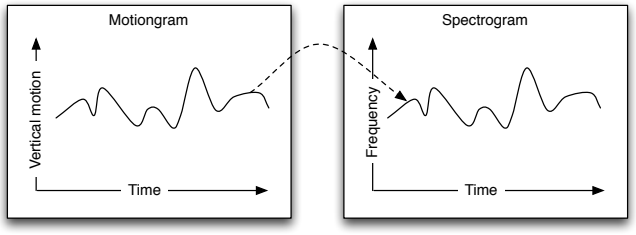

Figure 2. A direct mapping from motion data to spectral audio data.

sound frequencies are controlled by moving in the lower part of the image, and vice versa.

The sonification algorithm has been implemented in the module **jmod.sonifyer**∼ in the open framework Jamoma for Max [18]. Thus the module benefits from the extensive preset, mapping and cueing functionality present in Jamoma [19]. As for many other video modules in Jamoma, **jmod.sonifyer**∼ will adapt itself to any incoming matrix size, something which makes it easy to change between differently sized videos on the fly.

An example of how the sonifyer module may be used in conjunction with other Jamoma modules is shown in Figure 4, and a video tutorial of the functionality of the module can be seen in Video 1 (all video examples are available at www.arj.no/sonifyer/). The **jmod.input%** module gets video from the camera and passes it on to **jmod.motion%**, which calculates the motion image and does the noise reduction and filtering. The filtered motion image is passed on to **jmod.motiongram%**, which outputs a motiongram of the chosen size and direction (in this case horizontal). The right outlet of **jmod.motiongram%** passes the reference to the motiongram matrix on to **jmod.sonifyer**∼, while the left outlet passes on the message of the internal counter in the motiongram algorithm. This counter keeps track of the column number that the motiongram is currently outputting, and is used to control the speed of the "playback" of the motiongram to sound. For realtime applications this counter increases for each new frame received from the camera (typically at 25 fps), and for non-realtime applications it can be used to scan through the image as fast as possible.

V. Examples

Video 2 shows examples of sonification of some basic motion patterns: up–down, sideways, diagonal and circular. Here only a simple level of filtering and noise reduction is used, otherwise it is a direct translation from motiongram to sound. The examples show one of the largest problems with this approach to sonification: the motiongram's ability to only display motion in one direction. Thus the up and downwards motion is clearly visualized in the motiongram, and heard in the sound, but all the other motion patterns (sideways, diagonal and circular) are not represented equally well with only one dimension being sonified.

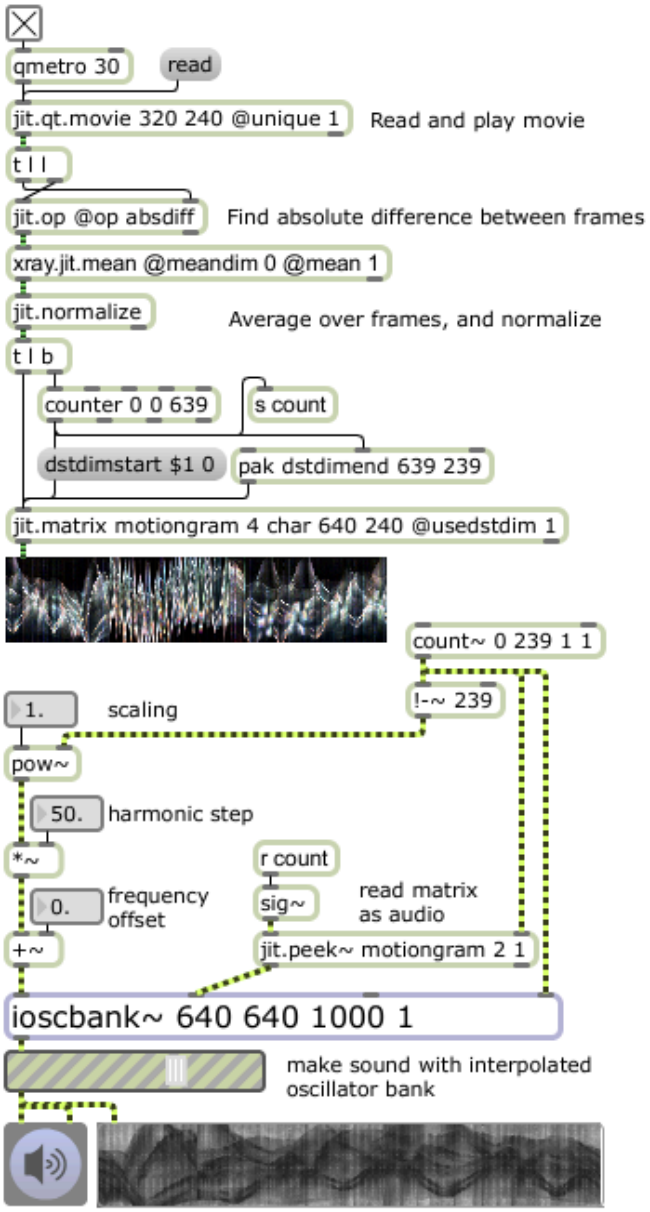

Figure 3. A minimal Max implementation for generating the sonification from video input. The oscillators are created in the **ioscbank**∼ object, and Jitter matrix data is converted to an audio signal with the **jit.peek**∼ object.

One attempt at sonifying the two axes at the same time is shown in Video 3. Here both horizontal and vertical motiongrams are created from the same video recording, and the sonifications of the two motiongrams have been mapped to the left and right audio channel respectively. While I originally thought this may be a good idea, the example shows that it does not work particularly well. Clearly more research is needed to find a better solution for sonifying the two dimensions.

Filtering and thresholding of the motion image is important for the final sounding result, as can be seen in

 ISBN: 978-1-61208-177-9

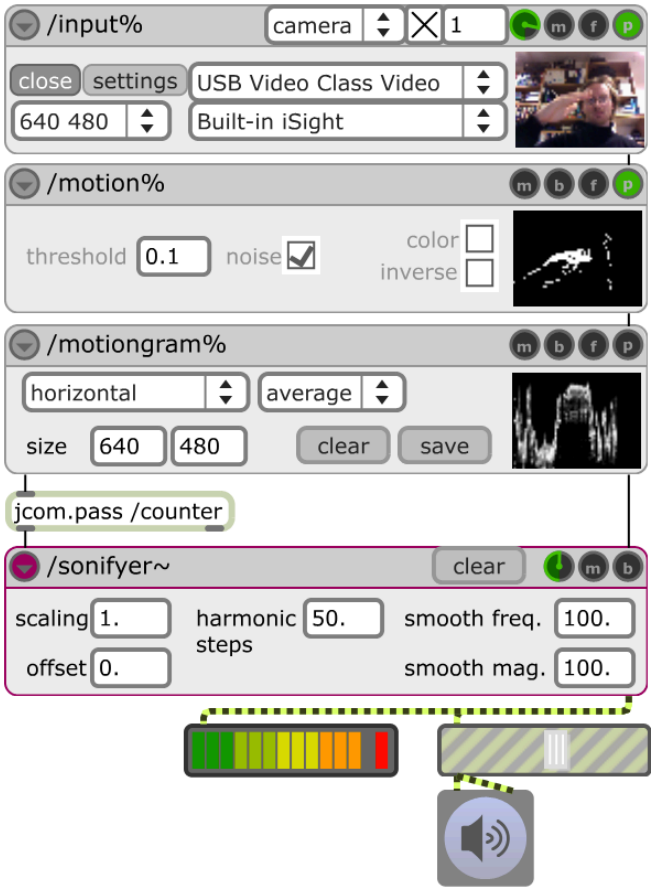

Figure 4. From the help patch of the **jmod.sonifyer~** module. The video input module is connected to a module creating the motion image and then to the motiongram module. Finally, the output motiongram is sent to the sonifyer module for the creation of sound.

an example of the sonification of a high-speed recording (200 fps) of a hand in motion in Video 4. Here three different types of filtering have been applied to show the different sonic results. When there is no thresholding and no noise reduction, all the details of the motion is shown in the motiongram and can also be heard in the sonification. Adding a binary threshold removes a substantial amount of pixels in the motiongram, and hence makes a cleaner sonification. Finally, adding a noise reduction algorithm further reduces the amount of pixels and sonifies only the most important part of the motion.

Video 5 shows an example of the sonification of a short violin improvisation. While the sonification manages to capture some of the details and temporal unfolding of motion over time, I generally find that a sonification of sound-producing actions tend to be confusing. This is probably because we expect that the sonified sound should be related to the sound-producing action. This, however, is not possible with such a generic sonification technique, which is based on translating all motion into sound without any prior knowledge about the content of the video material.

A more successful sonification of the motion of a performer can be seen in Video 6 of a French-Canadian fiddler. Here we are focusing mainly on the clogging pattern that is created in the feet. The rhythmicity of this pattern is sonified clearly, and the change of rhythmic figure and tempo is easily audible halfway throughout the excerpt. See [20] for a more detailed analysis of this performance.

An example of the sonification of dance motion is shown in Video 7. First the original recording is shown, where a dancer moves spontaneously to a short musical excerpt, followed by a sonification of the same motion. Here the sonification of the motion shows some clear similarities to the sonic qualities of the original sound. This, however, is a special case of a good correspondence between the original sound and the sonification result. In general I would argue that sonifications should not be evaluated against the original sound, but rather against the motion that they are sonifying. It is only in cases of sound-imitating motion that the sonification will be similar to the original sound.

The sonification module has also been tested in music performance. An excerpt from a performance of the piece *Soniperforma* at Biermannsgården in Oslo on 18 December 2010 can be seen in Video 8, and a screenshot from the performance patch in Figure 5. This piece is based on applying only video effects to change the sonic quality. This way it is possible to create for example delays in the sound by applying a motion blur function on the video image.

VI. Discussion

The sonification technique based on motiongrams presented in this paper is still in development. While the method works well for some examples, there are also several issues that will have to be explored further:

Dimensionality: The limitation of handling motion in more than one dimension was shown in Video 2. This limitation is based on the fact that motiongrams average over each row (or column) and therefore reduce motion in the video from 2 to 1 dimension. I will continue to explore how it is possible to handle multiple dimensions (2 and 3) in sonification, but with an aim to continue keeping the process simple, direct and intuitive.

Temporal resolution: A challenge when working with video as the source material for a sonification process is the poor temporal resolution as compared to audio. This is particularly apparent when working with direct mappings from video to sound. For this reason it will be interesting to explore how high-speed (200-1000 fps) video recordings will work as the basis for sonification. Such frequencies are still far lower than the possibilities of audio synthesis, but may reveal some possible future uses of this type of sonification approach.

 ISBN: 978-1-61208-177-9

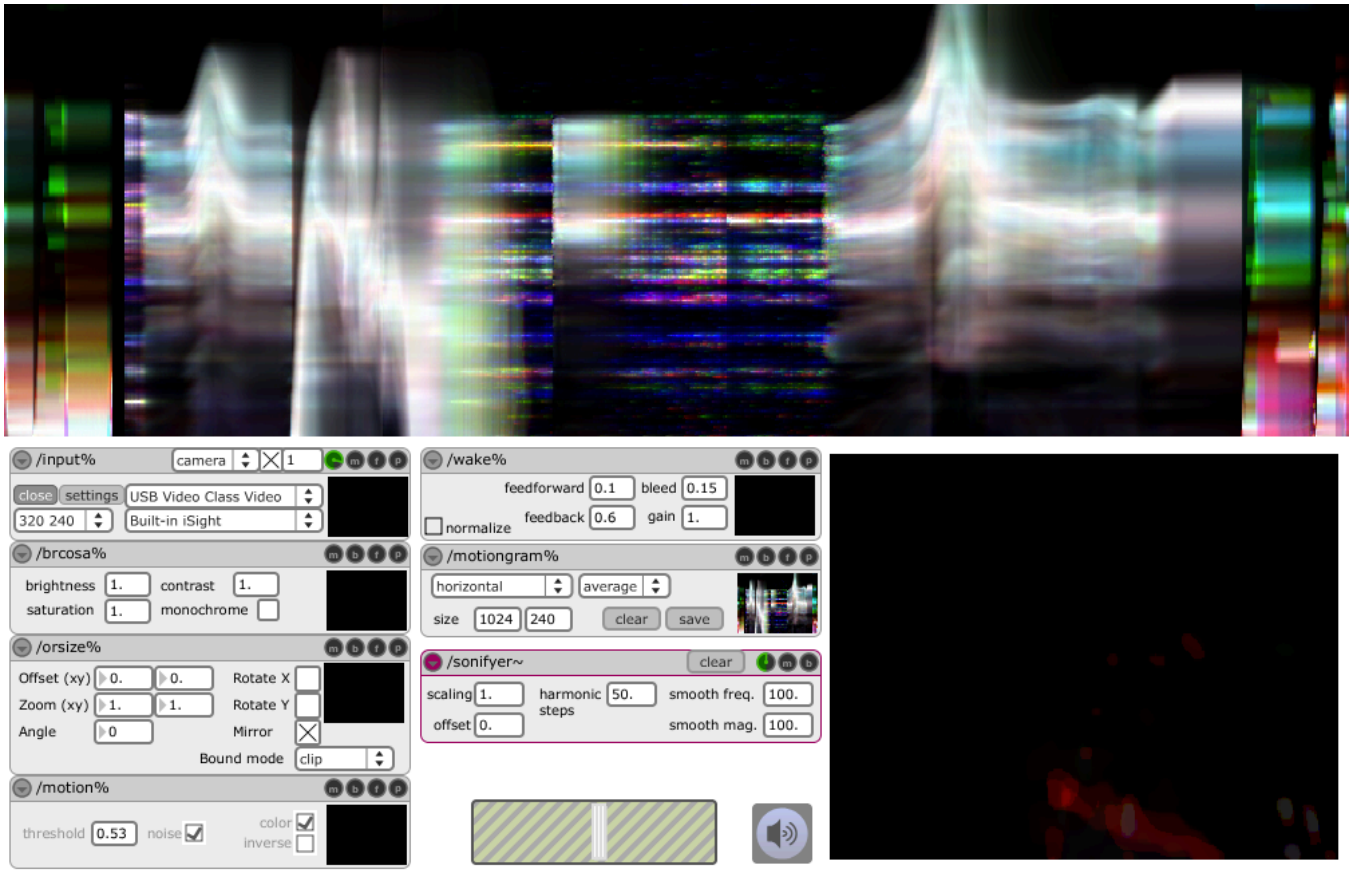

Figure 5. Performance patch using various types of video effects to modify the motiongram, and hence the output sound. The visual result from changing various video parameters can be seen in the motiongram at the top.

Analysis: The original idea of this sonification approach came from an analytic point of view: creating a tool to help in the analysis of various types of music-related motion. The exploration has shown that the largest potential of the method may be in the sonification of all sorts of non-sound-producing motion. If we were to create a sonification of the sound-producing action, it would be necessary to know where the excitatory parts of the instrument are in the image. This is a very different problem, and was never the intention of the project.

Sonic interaction design: I quickly realized that the current implementation probably has a larger potential in interactive than in analytic applications. The immediate and intuitive connection between motion and sound has opened for interesting sonic explorations in many different contexts, both in general human-computer interaction and for more creative applications. The method can be used as a tool to quickly create a sonification of body motion, which can later be used as the basis for designing a more complete sound design in an electronic device or system.

Music applications: As shown in Video 8, using video effects to modify sound in realtime has been a refreshing approach to sound creation. It has been fun to perform with, and audience members have commented that the link between projected image and sonified sound works well. At first such a performance setup may seem odd, but in fact it is quite similar to performing with a regular instrument. Since sound is only created when there is motion, moving one hand in front of the camera can be used to excite the "instrument," while the other hand can be used to modify the quality of the sound by changing video filters.

Scalability: The system has been tested on close-ups of hands, upper body, and full body video recordings. I have also done a test with a group of 20 students standing on the floor and being filmed from above. In such a setup it is possible to create a collaborative performance among the people making up "pixels" in the image.

Stability: The current implementation has been very reliable. The patch runs comfortably on a single laptop using a built-in camera, and can easily be extended to use any type of external camera. The video modules have been used in analytic and creative applications for the last 5 years, and have been adjusted so that they work well in all sorts of lighting conditions. Also, using the motion image as point of

 ISBN: 978-1-61208-177-9

departure means that the system does not rely on a particular type of background, as long as it is possible to make a separation between motion in the foreground and in the background.

VII. Future work

Issues to be addressed in future research include:

- optimizing the implementation so that it runs faster.
- implementing more interactive controls of the sonification parameters, e.g., based on extracted motion features.
- developing a non-realtime application. This would allow for creating more detailed sonifications of large motiongrams, e.g., of high-speed and high-resolution video material.
- exploring sonification of both horizontal and vertical motion, as well as from multiple cameras.
- exploration and user testing in many different contexts.
- exploring a similar approach to sonify data from infrared/inertial motion capture systems.

Acknowledgment

This research has been funded by the Norwegian Research Council through the *Sensing Music-related Actions* project, and has been carried out in the fourMs lab at the University of Oslo.

References

[1] W. W. Gaver, "Auditory icons: Using sound in computer interfaces," *Human-Computer Interaction*, vol. 2, pp. 167–177, 1986.

[2] ——, "The SonicFinder: An interface that uses auditory icons," *Human-Computer Interaction*, vol. 4, no. 1, pp. 67–94, 1989.

[3] D. Rocchesso and F. Fontana, *The Sounding Object*. Firenze: Edizioni di Mondo Estremo, 2003.

[4] D. Rocchesso, *Explorations in Sonic Interaction Design*. Logos, 2011.

[5] A. R. Jensenius, "Action–sound: Developing methods and tools to study music-related body movement," Ph.D. dissertation, University of Oslo, 2007. [Online]. Available: http://urn.nb.no/URN:NBN:no-18922

[6] S. Barrass and G. Kramer, "Using sonification," *Multimedia Systems*, vol. 7, no. 1, pp. 23–31, 1999.

[7] W. S. Yeo and J. Berger, "Application of image sonification methods to music," in *Proceedings of the International Computer Music Conference*, Barcelona, 2005.

[8] ——, "Application of raster scanning method to image sonification, sound visualization, sound analysis and synthesis," in *Proceedings of the 9th International Conference on Digital Audio Effects*, Montreal, 2006.

[9] F. Cooper, A. Liberman, and J. Borst, "The interconversion of audible and visible patterns as a basis for research in the perception of speech," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 37, no. 5, p. 318, 1951.

[10] G. Marino, M.-H. Serra, and J.-M. Raczinski, "The upic system: Origins and innovations," *Perspectives of New Music*, vol. 31, no. 1, pp. 258–269, 1993.

[11] "Metasynth [computer program]. available: http://www.uisoftware.com/metasynth/index.php (last checked: 08.11.2011)."

[12] G. Levin, "The table is the score: An augmented-reality interface for real-time, tangible, spectrographic performance," in *Proceedings of the International Conference on Computer Music 2006 (ICMC'06)*, 2006.

[13] M. Ojanen, J. Suominen, T. Kallio, and K. Lassfolk, "Design principles and user interfaces of erkki kurenniemi's electronic musical instruments of the 1960's and 1970's," in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 88–93.

[14] J.-M. Pelletier, "Sonified motion flow fields as a means of musical expression," in *Proceedings of the International Conference on New Interfaces For Musical Expression*, Genova, 2008, pp. 158–163.

[15] ——, "Perceptually motivated sonification of moving images," in *Proceedings of the International Computer Music Conference*, Montreal, 2009, pp. 207–210.

[16] O. Quek, V. Verfaille, and M. M. Wanderley, "Sonification of musician's ancillary gesutres," in *Proceedings of the 12th International Conference on Auditory Display*, London, UK, 2006, pp. 194–197.

[17] F. Grond, T. Hermann, V. Verfaille, and M. Wanderley, "Methods for effective sonification of clarinetists' ancillary gestures," *Gesture in Embodied Communication and Human-Computer Interaction*, pp. 171–181, 2010.

[18] T. Place and T. Lossius, "Jamoma: A modular standard for structuring patches in Max," in *Proceedings of the International Computer Music Conference*, New Orleans, LA, 2006, pp. 143–146.

[19] T. Place, T. Lossius, A. R. Jensenius, and N. Peters, "Flexible control of composite parameters in max/msp," in *Proceedings of the 2008 International Computer Music Conference*, Belfast, 2008, pp. 233–236. [Online]. Available: http://urn.nb.no/URN:NBN:no-20631

[20] E. Schoonderwaldt and A. R. Jensenius, "Effective and expressive movements in a french-canadian fiddler's performance," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, A. R. Jensenius, A. Tveit, R. I. Godøy, and D. Overholt, Eds., Oslo, Norway, 2011, pp. 256–259. [Online]. Available: http://www.nime2011.org/proceedings/papers/G12-Schoonderwaldt.pdf

 ISBN: 978-1-61208-177-9