EVALUATING HOW DIFFERENT VIDEO FEATURES INFLUENCE THE VISUAL QUALITY OF RESULTANT MOTIONGRAMS

Alexander Refsum Jensenius
University of Oslo, Department of Musicology, fourMs lab
a.r.jensenius@imv.uio.no

ABSTRACT

Motiongrams are visual representations of human motion, generated from regular video recordings. This paper evaluates how different video features may influence the generated motiongram: inversion, colour, filtering, background, lighting, clothing, video size and compression. It is argued that the proposed motiongram implementation is capable of visualising the main motion features even with quite drastic changes in all of the above mentioned variables.

1. INTRODUCTION

The last decade has seen a rapid growth of interest in studying *music-related body motion* [1–3]. Music-related motion is here used to describe all types of body motion that appear in a musical context, including those carried out by *performers* (i.e. musicians, conductors, dancers) and *perceivers* (e.g. in concerts, discos, on the bus). This includes a large variety of motion types, all of which may also occur in any type of location, e.g. concert halls, clubs, at home, in the street, or on the bus.

Having tools and methods for recording, visualising and analysing music-related motion are important for empirical music researchers. Various types of marker/sensor based motion capture systems excel at providing high quality data, which is useful for both qualitative and quantitative analyses. However, a major challenge with most such motion capture systems is that they require markers/sensors to be put on the body of the subject, something which makes them less ideal for recording, say, a musician in concert. Another problem with the data obtained from motion capture systems, is that they are focused on capturing the position of markers, or possibly body joints, and may not capture the global qualities of complex body motion satisfactorily. Here regular video recordings excel, albeit with a trade-off in terms of lower resolution/speed as opposed to motion capture systems.

All in all, I believe that a regular video recording is still among the most flexible, cheapest and most accessible solutions to recording music-related motion. Extracting useful information from regular video recordings is a challenge, however, and is often computationally heavy and based on many assumptions of the content of the video file. As opposed to *analysis*-based visualisation techniques, *motiongrams* is a simple and straightforward *reduction*-based approach to creating visual displays of continuous motion over time [4].

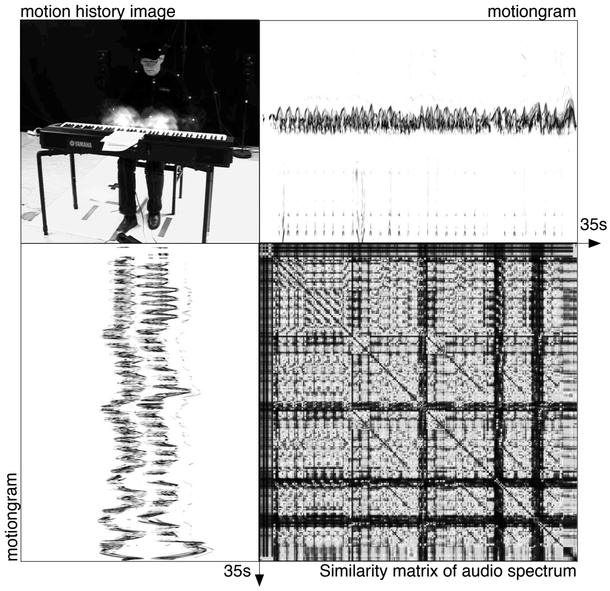

Figure 1. From a video recording of a pianist performing the opening of Beethoven's *Tempest Sonata*. Motion history image (top left), horizontal motiongram (top right), vertical motiongram (bottom left) and a similarity matrix of the audio spectrum (bottom right).

An example of how motiongrams may be used to study a performers' motion can be seen in Figure 1. Here the horizontal and vertical motiongrams represent vertical and horizontal motion, respectively. Since two motiongrams are shown, a similarity matrix of the audio spectrum is used so that it is possible to compare motion to sound in both dimensions. The vertical motiongram effectively visualises the phrasing in the transverse (horizontal) plane of the performer, while the horizontal motiongram displays the continuous attacks in the hands, as well as weight shifts in the legs, and the pedal activity of the right foot.

Motiongrams have been used for visualising many different types of music-related motion over the years [5], and even in studies of young infants with the risk of developing cerebral palsy [6]. The technique has proven to be flexible, scalable, and tolerable for changes in the input video files, but there has not yet been any systematic testing of the implementation. The aim of this paper has therefore been to study how different features of the input video files influence the final motiongrams. The paper starts with a brief overview of the motiongram technique, before the method is evaluated and discussed.

2. BACKGROUND

Motiongram as a technique is quite similar to that of *slit-scan photography* (see e.g. [7] for an overview), where a tiny slice of a video frame is captured at each time interval and added together to form a continuous display with time on one axis. Another related approach is the creation of "video traces," where the visual summary gives a sense of virtual presence [8]. The *Recreating Movement* application creates three-dimensional waterfall visualisations of body motion based on pre-keyed film sequences [9]. Yet another approach is the multi-camera system by Liu et al., where reduced silhouette images are used to study periodic patterns in analysis of gait [10].

My approach to creating motiongrams is in many ways similar to the above-mentioned techniques, but with some important differences. First, while slit-scanning takes only one slice of each video frame, motiongrams are based on averaging over the whole frame. Second, motiongrams are based on the motion image, which means that only motion will show up in the final display. Third, motiongrams are primarily meant as a visual display for (qualitative) analytical use, and are often used together with displays of sound (e.g. spectrograms) and symbolic music notation.

2.1 Motiongram overview

Figure 2 shows an overview of the steps involved in the creation of a motiongram. The method is based on calculating the absolute difference between each of the pixels in subsequent frames of a video stream. The end result is a new image, the *motion image*, where only the pixels that have changed between the frames are displayed [11]. It is possible to calculate a motion image based on the original image, but pre-processing the original image (e.g. by converting to greyscale and adjusting the brightness) often leads to motion images with less noise. It is also possible to do background subtraction before calculating the motion image, but this requires that the background has been recorded separately, which is not always the case.

The quality of the raw motion image depends on the quality of the video recording. Small changes in lighting (e.g. fluorescent lighting), camera motion, compression artefacts, etc., may all influence the final motion image. It is therefore often necessary to filter the motion image, for example by converting to a binary image and applying a low-pass filter to remove pixels below a certain threshold. It may also be useful to apply a "blob-based" noise removal algorithm, which leaves out single (or groups of few) pixels in the motion image.

The motion image may be used for creating an *average motion image* and a *motion history image* [12], as can also be seen in Figure 2. The motion history image is created by calculating the average motion image and laying this

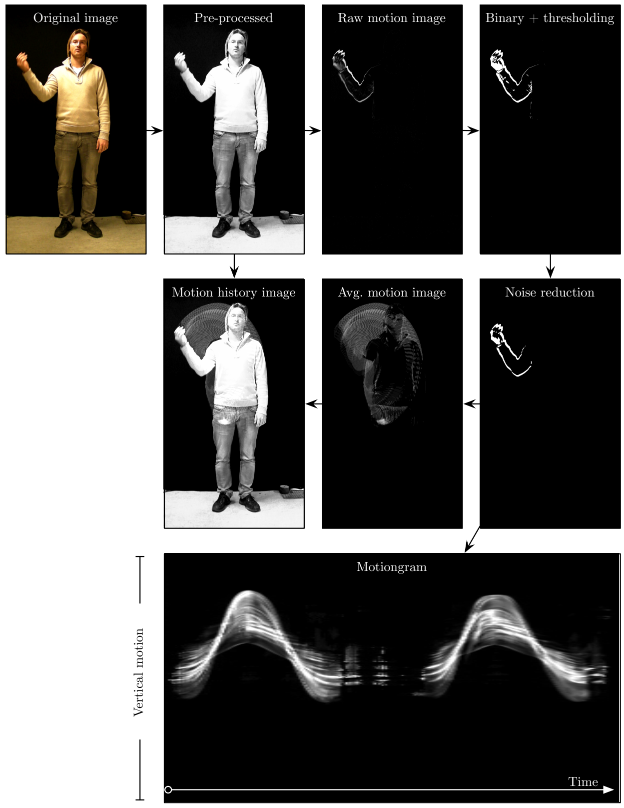

Figure 2. The process of creating a motion image, average motion image, motion history image and motiongram.

on top of a frame of the video recording. This makes it possible to get an impression of the spatial distribution of the motion, at least for motion sequences limited in time and/or space. More information about these visualisation techniques can be found in [5, 13].

A motiongram is created by adding up a time series of normalised mean values of the columns or rows from the motion image. Thus for each frame of the input video matrix (of size MxN), a $1xN$ or $Mx1$ matrix is calculated. Adding these 1 pixel wide or tall stripes next to each other results in either horizontal or vertical motiongrams.

2.2 Implementations

The motiongram technique was originally developed as part of the *Musical Gestures Toolbox* [14], which is now included in the open framework Jamoma [15]. [1] A screenshot of the help patch of the Jamoma motiongram module (*jmod.motiongram%*) is shown in Figure 3. This module takes a motion image as input, and outputs a motiongram of the chosen size and direction (horizontal or vertical). Besides regular Jitter objects, this module also uses the *xray.jit.mean* object [16]. Since the motiongram module only does the reduction and plotting over time, all filtering, choice of colour, etc., needs to be done in modules earlier on in the video chain.

An implementation of the motiongram algorithm is also implemented in the EyesWeb platform, in a module called

[1] `http://www.jamoma.org`

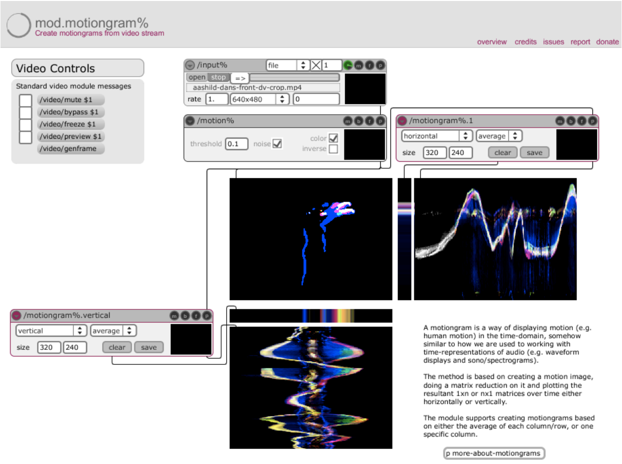

Figure 3. From the help patch of the *jmod.motiongram%* module in Jamoma for Max. The *jmod.input%* module is used for playing back a video file, a *jmod.motion%* module is used for calculating the motion image and do noise removal, and two *jmod.motiongram%* modules are used for creating the horizontal and vertical motiongrams.

videogram [17], and in Matlab through the *Motion Segmentation Toolbox* [18]. For people that are not inclined to do the programming themselves, I have developed two standalone applications for OSX and Windows: *AudioVideoAnalysis* and *VideoAnalysis*.[2] The former provides realtime plotting of screen-sized motiongrams, together with spectrograms of the audio. The latter is a non-realtime application that can batch export full-sized motiongrams of all video files in a folder, as well as outputting motion history images and text files with various quantitative features. Most figures in this paper are based on images exported from the *VideoAnalysis* application.

3. EVALUATION

This section will show how different video variables or external features may influence the resultant motiongrams: colours, image inversion, filtering, background, lighting, clothing, markers, video size and compression. For comparative purposes all examples are based on short video recordings of the same, short action: a circular motion with the right arm (as seen in Figure 2).

3.1 Colour

Motiongrams can be created in either colours or greyscale, and this is usually controlled when deciding whether the motion image matrix should be calculated with 1 plane (greyscale) or 4 planes (ARGB). Unless there are specific meaningful colours in the image, greyscale motiongrams may give the best visual result, with the added benefit of being much faster to compute.

In some cases, however, motiongrams in colours may be preferable, for example if the subject is wearing clothing with particular colours. An example of this can be seen in Figure 4, where it is possible to follow the trajectories

[2] `http://www.fourms.uio.no/software/`

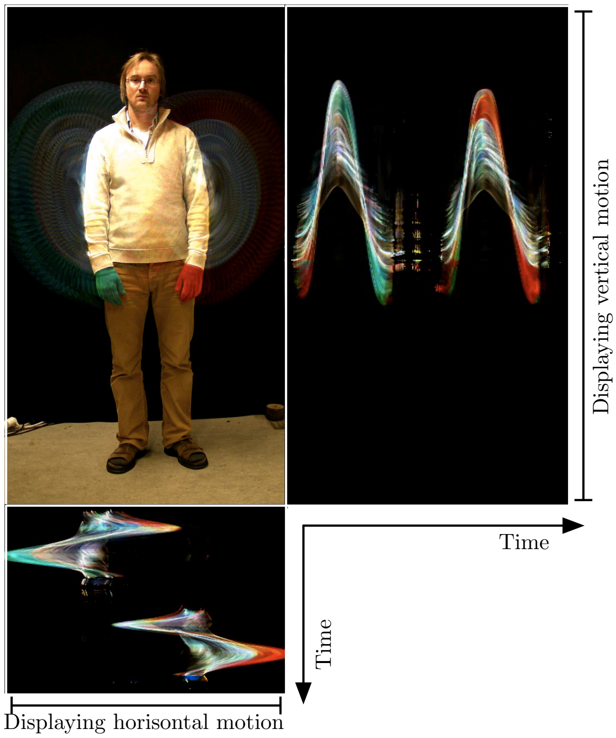

Figure 4. An example of how colours can be used to trace each hand separately in a motiongram. Here the right hand of the subject (wearing a green glove) is moving first, followed by the same action performed with the left hand of the subject (wearing a red glove). The motiongrams are (to a certain extent) able to visualise the colours separately.

of each of the coloured gloves through the colours in the motiongram, i.e. the hand with the green glove moved first, followed by the hand with the red glove.

3.2 Inversion

A normal motion image and motiongram will show motion as white traces on a black background. While this often works well on screen, it may be easier on the eye (and for the printer) to invert the motiongram so that it displays black "traces" on a white background instead. This can be done through a simple image inversion process, and the resultant motiongrams are otherwise identical to the originals. Examples of both normal and inverted motiongrams are shown in Figure 5.

3.3 Filtering

The types and levels of filtering applied to a motion image are important for the final looking result of a motiongram. Since the motiongram algorithm is only based on reduction, pre-processing of the original image, and filtering of the motion image, should be done in the earlier stages of the chain. In general, I find that adjusting the threshold level of the motion image is one of the most important parameters, as it decides how much information is visible in the output motiongram. A low threshold will result in more active pixels in the motion image, which again will result in more information/noise in the motiongram.

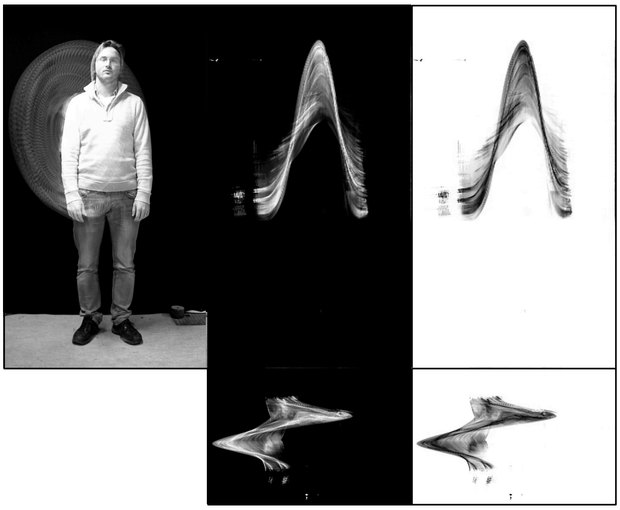

Figure 5. Motion history image (left), regular motiongrams (middle), inverted motiongrams (right). Except for the inversion, the motiongrams are otherwise identical.

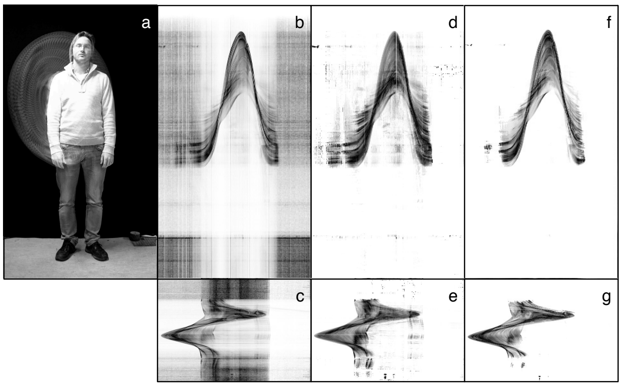

Figure 6. Different levels of the threshold (t) and noise removal (n) functions influence the resultant motiongrams: b/c: t=0.0, n=0. d/e: t=0.02, n=1. f/g: t=0.1, n=1.

The motiongrams in Figure 6 are based on applying different levels of filtering and noise reduction on the motion image. Using no filtering and no noise reduction results in a noisy motiongram. There are several reasons for this noise, the most important ones being changes due to fluorescent lighting and video compression artefacts. Using a low-pass filter and/or a noise reduction algorithm based on blurring the image before applying the filter, greatly improves the output motiongram. There is no rule of thumb as to how much filtering should be applied; this depends entirely on the video material and what the motiongram shall be used for. For example, using little filtering and noise reduction makes it possible to see smaller movements, but it will also introduce more noise in the image. Using noise reduction will make the largest movements stand out more clearly in the motiongram, but it will also remove smaller movements from the display.

The final motiongram can be filtered as well, but in general I find that pre-processing and filtering the image before creating the motiongram give the best results. Still, as the different motiongrams in Figure 6 show, filtering the image does not alter the main properties of the traces in the motiongrams, it merely enhances or removes certain parts of the display.

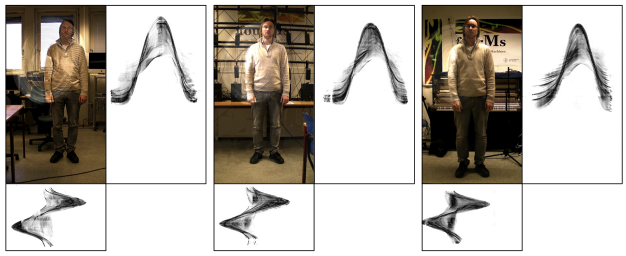

Figure 7. Different recordings of the same action: a circular motion of the right hand. Even with quite different background and lighting conditions the motiongrams show the main features of the motion.

3.4 Background and lighting

Separation of foreground and background is an important topic in computer vision and video analysis. Fortunately, the creation of motion images effectively removes the (unmoving) background, which means that the background is also effectively removed in the resultant motiongrams. In my experience, motion images and hence motiongrams are remarkably robust when it comes to differences in background and lighting conditions. Figure 7 shows examples of motiongrams created from recordings of the same action as in previous figures, but recorded in front of different backgrounds and with different lighting conditions. Even with such different recording conditions the corresponding motiongrams look quite similar.

3.5 Clothing

As mentioned above, the background is effectively "removed" in motiongrams due to the frame differencing. This, however, is not entirely true for cases where the foreground and background have the same or similar colour. In such cases, the similarity or difference between the background and the colour and luminosity of clothes and skin, will affect the final result. The rule of thumb is to use a background that is as different as possible to the foreground. But, as Figure 8 shows, wearing a black sweater in front of a black background still makes it possible to see the trace of the motion. This is mainly because the hand is clearly separable from the background. As such, deliberately choosing clothing with a similar colour to the background may help in enhancing salient parts of the body, e.g. hands and head.

3.6 Video size and compression

The pixel size of the input video stream, and the compression technique used for storing the video, also influence the resultant motiongram. Obviously, using large and uncompressed video files will result in motiongrams with the most detail. That said, I find that files with low to moderate compression (e.g. MPEG-4 H.264) often work well.

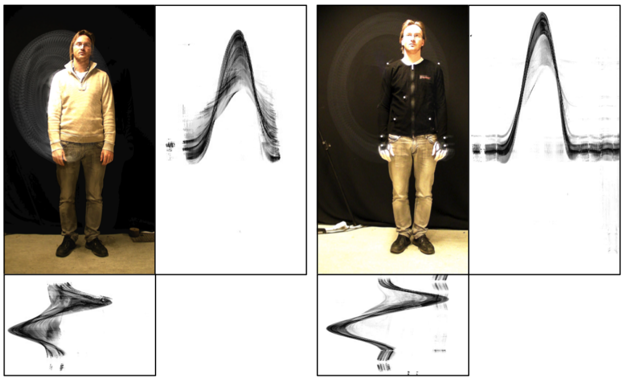

Figure 8. Two recordings with different clothing. When wearing clothes similar to the background, the motion of the moving hand is more apparent in the motiongram.

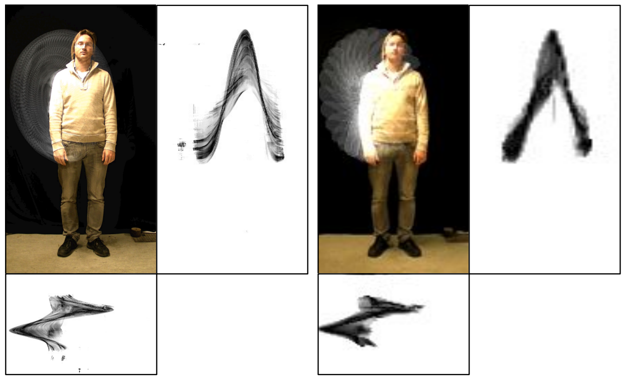

Figure 9. Motiongrams from a video with full frame rate and good compression (left) and a downsampled and highly compressed version of the same file (right).

One reason for this is that the filtering applied when compressing the file effectively clears up the salient parts of the image. Thus, less filtering has to be applied when creating the motiongram.

Motiongrams inherit the resolution of the input video file, so more pixels in the video will result in more pixels in the motiongram. For most purposes keeping the original resolution may be advisable. However, if the computational speed is important, or if the original video file has low resolution and/or is heavily compressed, it may be necessary to create reduced motiongrams. My experience is that even such low-resolution motiongrams may contain most of the salient information in the motion sequence. Figure 9 shows motiongrams made from a high quality video file (1920x1080 pixels, MPEG-4 H.264, 24 Mbit/s), compared to a downsampled and highly compressed version of the same file (176x99 pixels, MPEG-4 H.264, 82 kbit/s). The motiongrams created from the latter file are less rich in detail, but still manage to capture the overall shape of the action. In general, the method seems to be quite good at generating meaningful displays even from a very small and highly compressed source file.

4. CONCLUSIONS

The strength of motiongrams as a visualisation technique is the ability to represent both the temporal and spatial unfolding of motion based on a simple reduction process. As such, it is a very general technique, and is therefore also flexible when it comes to the original video input. Motiongrams can be used to represent both short and long motion sequences, anything from a few seconds to several hours. For short sequences, details in the motion patterns may be studied. Longer motiongrams may be used to quickly get an overview of structural elements of the recordings. This makes them useful as motion summaries, for navigation and in comparative studies.

As the paper has shown, the method is robust and flexible: motiongrams can be created in colours or greyscale, be inverted, and filtered at different levels, without affecting the motion trajectories. Also, having different backgrounds, clothing, video size, and video compression will not change the final visual display to any larger extent. One weakness, though, is that a motiongram will not display external motion (e.g. camera panning or zooming) particularly well, unless it is this external motion that is the focus of the study.

There are a number of issues that can, and will, be improved in future research, including:

- implementing the technique in other programming environments, e.g. PureData and Octave.
- improving the processing speed by creating motiongrams based on sampling frames progressively, so that a preview of the motiongram can be presented more quickly.
- exploring more advanced tracking techniques as the source material for creating motiongrams, e.g. optical flow. Some preliminary testing in Matlab shows that this may enhance the results, albeit at a higher CPU cost [19].
- exploring how it is possible to create three-dimensional motiongrams to display both horizontal and vertical motion in one image, and make it possible to control them interactively by the user, e.g. based on the ideas presented in [9].
- exploring development of visualisation techniques that complement novel audio visualisation techniques, e.g. as presented in [20].
- further exploration of motiongrams as the basis for sonification of motion, e.g. starting with some of the approaches presented in [21, 22].
- developing visualisation techniques for 3D/6D data from motion capture systems that can be used together with motiongrams.

Acknowledgments

This work was supported by the Norwegian Research Council through the projects *Musical Gestures* (2004–2007) and *Sensing Music-related Actions* (2008–2012).

5. REFERENCES

[1] C. Cadoz and M. M. Wanderley, “Gesture – Music,” in *Trends in Gestural Control of Music [CD-ROM]*, M. M. Wanderley and M. Battier, Eds. Paris: IRCAM, 2000, pp. 71–94. [Online]. Available: http://www.music.mcgill.ca/~mwanderley/Trends/

[2] A. Gritten and E. King, Eds., *Music and Gesture*. Hampshire: Ashgate, 2006.

[3] R. I. Godøy and M. Leman, *Musical Gestures: Sound, Movement, and Meaning*. New York: Routledge, 2010.

[4] A. R. Jensenius, “Using motiongrams in the study of musical gestures,” in *Proceedings of the 2006 International Computer Music Conference*. New Orleans, LA: Tulane University, 2006, pp. 499–502. [Online]. Available: http://urn.nb.no/URN:NBN:no-21798

[5] ——, “Action–sound: Developing methods and tools to study music-related body movement,” Ph.D. dissertation, University of Oslo, 2008. [Online]. Available: http://urn.nb.no/URN:NBN:no-18922

[6] L. Adde, J. L. Helbostad, A. R. Jensenius, G. Taraldsen, and R. Støen, “Using computer-based video analysis in the study of fidgety movements,” *Early Human Development*, vol. 85, no. 9, pp. 541–547, 2009. [Online]. Available: http://dx.doi.org/doi:10.1016/j.earlhumdev.2009.05.003

[7] G. Levin, “An informal catalogue of slit-scan video artworks,” 2005. [Online]. Available: http://www.flong.com/texts/lists/slit_scan

[8] M. Nunes, S. Greenberg, S. Carpendale, and C. Gutwin, “What did I miss? visualizing the past through video traces,” in *Proceedings of the Tenth European Conference on Computer Supported Cooperative Work*. Springer, 2007, pp. 1–20. [Online]. Available: http://dx.doi.org/10.1007/978-1-84800-031-5_1

[9] M. Hilpoltsteiner, “Recreating movement - tools for analyzing film sequences,” Diploma thesis, University of Applied Sciences, Wuerzburg, Germany, 2005.

[10] Y. Liu, R. Collins, and Y. Tsin, “Gait sequence analysis using frieze patterns,” in *Computer Vision — ECCV 2002*, ser. Lecture Notes in Computer Science, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Springer Berlin / Heidelberg, 2002, vol. 2351, pp. 733–736. [Online]. Available: http://dx.doi.org/10.1007/3-540-47967-8_44

[11] A. Camurri, I. Lagerlöf, and G. Volpe, “Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques,” *International Journal of Human-Computer Studies*, vol. 59, no. 1-2, pp. 213–225, July 2003. [Online]. Available: http://dx.doi.org/10.1016/S1071-5819(03)00050-8

[12] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, March 2001.

[13] A. R. Jensenius, “Some video abstraction techniques for displaying body movement in analysis and performance,” *Leonardo*, 2012 Forthcoming.

[14] A. R. Jensenius, R. I. Godøy, and M. M. Wanderley, “Developing tools for studying musical gestures within the Max/MSP/Jitter environment,” in *Proceedings of the International Computer Music Conference, 4-10 September, 2005*, Barcelona, 2005, pp. 282–285. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2005.178

[15] 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. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2006.032

[16] W. H. Smith, “xray.jit (max5 objects),” 2010. [Online]. Available: http://moniker.name/worldmaking/?page_id=17

[17] A. Camurri, P. Coletta, G. Varni, and S. Ghisio, “Developing multimodal interactive systems with EyesWeb XMI,” in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 305–308.

[18] S. F. Olsen, “Automatisk segmentering av bevegelsesdata,” Master’s thesis, Norwegian University of Science and Technology (NTNU), Department of Engineering Cybernetics, 2010. [Online]. Available: http://ntnu.diva-portal.org/smash/record.jsf?searchId=2&pid=diva2:360262

[19] H. Kirkerød, “Optical flow applied to infant movement,” Master’s thesis, Norwegian University of Science and Technology (NTNU), Department of Engineering Cybernetics, 2010. [Online]. Available: http://ntnu.diva-portal.org/smash/record.jsf?searchId=1&pid=diva2:348932

[20] K. Siedenburg, “An exploration of real-time visualizations of musical timbre,” CNMAT, University of California, Berkeley, Tech. Rep., 2009. [Online]. Available: http://cnmat.berkeley.edu/publication/exploration_real_time_visualizations_musical_timbre

[21] J.-F. Charles, “A tutorial on spectral sound processing using Max/MSP and Jitter,” *Computer Music Journal*, vol. 32, no. 3, pp. 87—102, 2008.

[22] A. R. Jensenius, “Motion-sound interaction using sonification based on motiongrams,” in *Proceedings of ACHI 2012: The Fifth International Conference on Advances in Computer-Human Interactions*. IARIA, 2012, pp. 170–175. [Online]. Available: http://urn.nb.no/URN:NBN:no-30588