Nick Collins* **and Fredrik Olofsson**†
*Centre for Science and Music
Faculty of Music
11 West Road, Cambridge CB3 9DP UK
nc272@cam.ac.uk
†Boxhagener Straße 16
10245 Berlin, Germany
www.fredrikolofsson.com
f@fredrikolofsson.com

klipp av: Live Algorithmic Splicing and Audiovisual Event Capture

Recent new media concerts feature a trend toward the fuller integration of modalities enabled by close audiovisual collaborations, avoiding the sometimes artificial separation of disc jockey (DJ) and video jockey (VJ), of audio and visual artist. Integrating audio and visual domains has been an artistic concern from the experimental films of such notaries as Oskar Fischinger and Norman McClaren earlier in the 20th century, through 1960s happenings, 1970s analog video synthesizers, and 1980s pop videos, to the current proliferation of VJing, DVD labels, and live cinema (Lew 2004). The rise of the VJ has been allied with the growth in club culture since the 1980s, with Super-8 film and video projectionists at early raves now replaced by "laptopists" armed with commercial digital VJ software like Isadora, Aestesis, Motion Dive, and Arkaos VJ. (An extensive list is maintained at www.audiovisualizers.com.)

In much current practice, where a VJ accompanies fixed (pre-recorded) audio, correlation in mapping is usually achieved via a simple spectral analysis of the overall output sound. Graphical objects can be controlled using a downsampled energy envelope in a frequency band. Yet this is a crude solution for live generated audio; fine details in the creation of audio objects should be accessible as video controls. Analogous to the sampling culture within digital music, source material for visual manipulation is often provided by pre-prepared footage or captured live with digital cameras. Synthesis also provides an option for the creation of imagery, and generative graphics are a further staple. Modern programs integrate many different possible sources and effects processes in software interfaces, with external control from MIDI, Open Sound Control (OSC; Wright and Freed 1997), and Universal Serial Bus (USB) devices.

Computer Music Journal, 30:2, pp. 8–18, Summer 2006

Live performance has seen the development of MIDI-triggering software for video clips such as EBN's Video Control System and Coldcut's VJamm (www.vjamm.com), and turntable tracking devices applied as control interfaces to video playback like Final Scratch (www.finalscratch.com) and MsPinky (www.mspinky.com). The influential audiovisual sampling group Coldcut performs live by running precomposed or keyboard-performed MIDI sequences from Ableton Live as control inputs to their VJamm software, triggering simultaneous playback of video clips with their soundtracks. They have not explored, however, the use of captured audio and video, nor the real potential of algorithmic automation of such effects. Techniques will be described in this article that make this natural step.

Customizable graphical programming languages like Max/MSP (with the nato.0+55, softVNS2, and Jitter extensions), Pure Data (Pd, with GEM, PDP, and GridFlow extensions), or jMax (with the DIPS, or Digital Image Processing with Sound, package of Matsuda et al. 2002) can cater to those who see no a priori separation of the modalities and wish to generate both concurrently, from a common algorithm or with some form of internal message passing. Other authors choose to define their own protocols for information transfer between modality-specific applications (Betts 2002; Collins and Olofsson 2003), perhaps using a network protocol like OSC to connect laptops.

The heritage of the VJ is somewhat independent of another tradition that has combined music and image, namely film. It is important to be cautious about the direct application of film music theory, particularly wherever the subservience of non-diegetic music is trumpeted. The emotional underscoring of narrative concerns in typical orchestral film music is certainly not the state of play in a club environment! As Nicholas Cook notes (1998), it may be more correct to speak of music film in many cases of multimedia, and the emphasis is certainly this way around in the history of VJ performance. In an artistic sphere, however, the marriage of sound and vision provides potential not just for direct "mickey mousing" (the absolute coincidence of sound and visual action, originally pertaining to animated characters), but also more subtle synchronizations, contrasts of pacing, and even combative opposition of domains.

The mapping between modalities can be a central concern of audiovisual collaboration, even as a theme of live improvisation.

klipp av (Swedish for "cut apart") is an audiovisual performance duo that has been experimenting intensely with audiovisual capture technology since 2002. In particular, advances in the algorithmic manipulation of audio and real-time event analysis have been applied multi-modally to video splicing and event capture. The project was founded to investigate providing evidence for live audio-splicing algorithms by a further correlated visual component, but expanded its remit with the realization that audio processing could inform new video processing effects. A clear technical theme of this work has become the exploitation of event-sensitive audio analysis to provide equivalent video analysis. This is commensurate with recent work in multimedia content analysis, where audio descriptors play a role in the segmentation of video (Wang, Liu, and Huang 2000).

Video Processing from Audio Analysis

Inspiration for much live-electronic video-splicing work can be traced to the "collision editing" (Chion 1994, p. 152) of pioneering scratch and pop-video experiments (Snider 2000; Heap 2003; Hyman 2005): the seminal video for Herbie Hancock's *Rockit* (1983), the stutters of Max Headroom, or the audiovisual collage of Steinski's *The Motorcade Sped On* (1985). A move to real-time manipulation of digital media was pioneered in early live-electronic scratch-video performance by such ensembles as Emergency Broadcast Network and Coldcut/Hex in the 1990s. In this section, we describe extending the process to live, algorithmically composed cut-ups and show further dividends of technology for real-time audiovisual analysis.

Video Scratching for Free: Applying the BBCut Library to Video

Digital editing practice in the constructions of such popular music styles as jungle (Goldie, Metalheadz), drill and bass (Aphex Twin, Squarepusher), and breakcore (Venetian Snares) is a laborious process of cutting source drum loops and placing segments in highly elaborate patterns, often valuing fast repetitions above 15 Hz that become pitched as audio-rate events (the "drill" in drill and bass). Preparatory work is reduced with such loop segmentation software as Recycle, which tags samples with MIDI triggers at transients; a MIDI sequencer can then be used to devise splicing patterns. Automation for the creation of patterns themselves would seem a natural application of algorithmic composition. Moreover, because this music often explores rhythmic motifs unplayable by humans, live generation is only possible through such routines. Splicing algorithms can have handles for interactive control of pattern generation and the synthesis of the edits, further avoiding any need to pre-render audio and thus showing a compressive benefit.

BBCut (Collins 2002) for the SuperCollider audio programming environment (McCartney 2002) is an advanced architecture to tackle algorithmic audio splicing. The composition of a sequence of cuts is separated from the synthesis of those cuts. Each algorithmic-cutting procedure can work on various possible splicing targets, encouraging software reusability. Cuts can be applied both to fixed audio loops and to continuous audio streams, enabling processing of live audio. By adopting a simple hierarchy of phrase/block/repeat (see Figure 1), "cut-aware" effects can be applied that respect that particular structure, allowing many novel synthesis effects. For example, a filter's cutoff frequency might be reduced for each successive repetition within a block, then reset for the next block.

BBCut contains a variety of splicing algorithms,

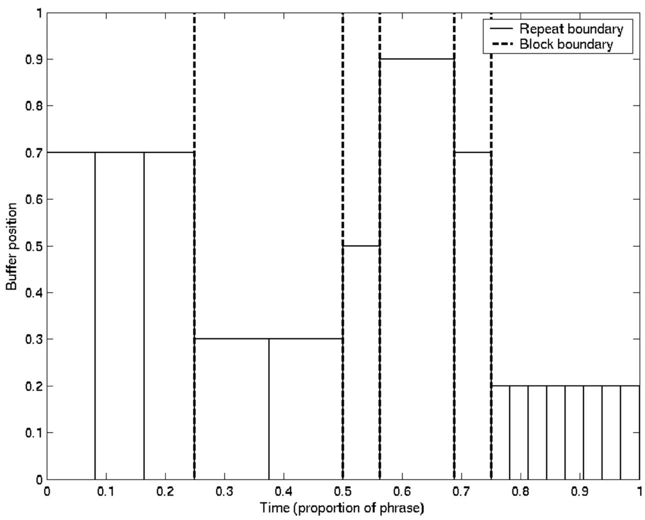


Figure 1. A phrase shown segmented into six blocks, each referencing a given buffer position as a common start position of component repeats (individual rendered cuts). The repeats are shown as successive rectangles of the same height (the y-axis corresponding to initial buffer playback position), and block boundaries are indicated by dashed vertical lines.

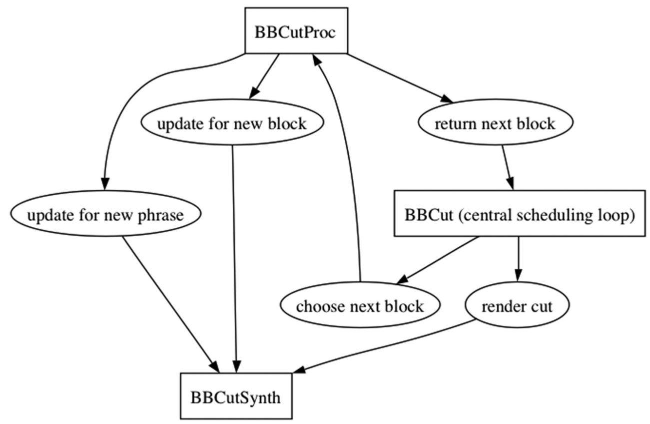


Figure 2. Message passing between core BBCut classes. Messages are in ellipses; classes are in rectangles. With this architecture, new cut synthesizer classes are derived from BBCutSynth, and new cutting algorithms are derived from BBCutProc.

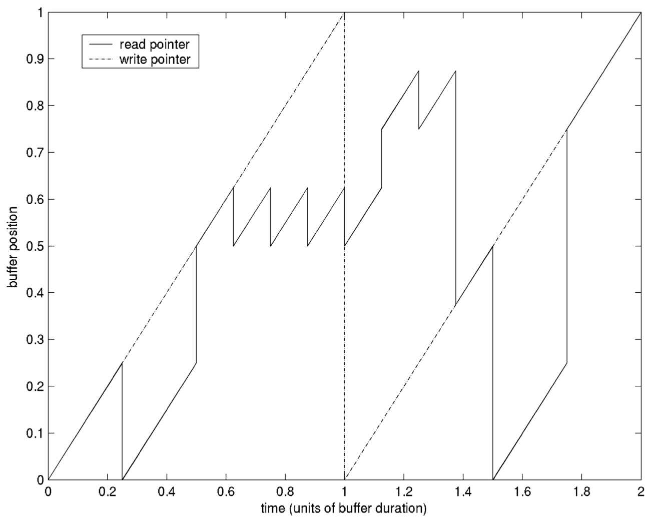
inspired by different popular music practices including jungle (BBCutProc11), drill and bass (WarpCutProc, named after the influential Warp Records), thrash drumming (ThrashCutProc), and the work of Squarepusher (SQPusher1). This latter is based on an analysis of Tom Jenkinson's drum programming, with various production rules for material, including a small database of templates for n-tuplet fill patterns in Jenkinson's manic jazz drummer style. Alongside such constructions are more abstract and experimental procedures. BBCut includes mathematical permutation spaces from arbitrary functions and change-ringing patterns, iterative cutting algorithms based on fractals and other recursions, and more generalized procedures whose functions can be adjusted in real time using SuperCollider as an interpreted programming language. BBCut is flexible enough to cope with rapidly changing sequences of time signatures, and the structure assumptions do not impose a 4/4-centric view of the musical world, though obviously such assumptions underlie some specific cutting algorithms in a given style.

Because the synthesis of cuts is independent of the composition of possible cut sequences, it seemed natural to try co-synthesizing an output in the visual modality. A rendering class was written in SuperCollider simply to send OSC messages to a third-party application conveying splicing information in real time. A simple protocol (Collins and Olofsson 2003) follows the basic object-messaging system used internally in BBCut to communicate with other applications. Figure 2 shows details of this message passing; the OSC messaging class is a sub-class of BBCutSynth. To support more than one cutter at once, each OSC messaging object is assigned its own instance identification (ID) tag.

On a local network, the latency is on the order of only milliseconds, equivalent to the delay in sending synthesis instructions from the SuperCollider language to the separate SC Server synthesizer. In practice, this provides a perceptually equivalent rendering time in audio and visual modalities. Given a sufficient frame rate in the video application, this yields well under the 30 msec multimodal integration time suggested by Pöppel (1997).

Analogous to fixed-buffer and circular-buffer (loop and stream) cut synthesizers for audio, fixed-video-buffer and continuous-video-stream synthesizers are built into the visual application. Loops are sourced from pre-existing material or captured live. Video streams come from disk or from continual camera capture in a venue. Stuttering repetitions generated by BBCut translate into jumps back to a stored read pointer position in a circular buffer. Figure 3 demonstrates this mechanism for a circular streaming buffer of video with a time map diagram. The write position in the buffer progresses constantly, whereas the read pointer is seen to jump back and forth to pertinent start points for cuts. As long as these maneuvers

Figure 3. Demonstration of a possible cut sequence in terms of read pointer positions into a continuously recording circular buffer. Blocks are seen as a jagged comb where each repetition returns to a common starting position.



do not allow the write head to overtake the read head (which could happen if there are playback-rate manipulations), no discontinuity will occur.

By leveraging the existing message passing of the BBCut class hierarchy, video-processing effects were obtained cheaply, and audiovisual synchronization was automatic. The communication protocol was expanded to allow capture synchronization such that recorded video buffers could correlate exactly to captured audio. The compositional applications were immediate, with the ability to cut-up footage of dancers in synchrony with overt aspects of the music to which they danced. Music could be manipulated as audiovisual sources, showing the causal source of an audio snippet alongside its processing.

Video Event Analysis for Free: Audiovisual Event Capture Based on Real-Time Audio Segmentation

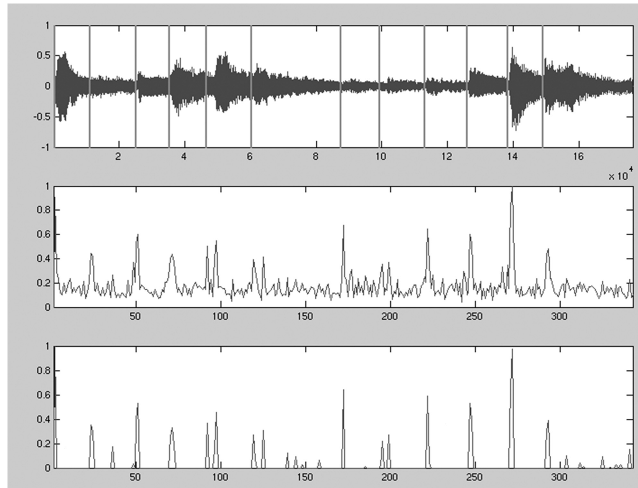
Splicing algorithms can be most effective when the source material is segmented into perceptually relevant events, typically of the order of 100–500 msec in duration. These events correspond to such sonic entities as drum hits, discrete pitched note events, and speech phonemes (dubbed *note/phone* events by Rossignol et al. 1999) and are the rhythmic-rate building-blocks of a sound stream. Tristan Jehan (2004) has written of event analysis/synthesis on this time scale. In terms of splicing audio, knowledge of the location of these events underlies a sensitive handling of a source rather than the imposition of an external partitioning grid.

For live performance in which audio is collected in real time, segmentation should take place in real time as well, and as soon as possible (Brossier, Bello, and Plumbley 2004; Collins 2004). Buffer locations of captured events can be stored into a database for compositional reference; however, this event-tagging information may only be valid as long as a circular buffer does not overwrite itself. It is convenient to take a relatively large buffer size or to copy event data into long-term storage.

If events are being analyzed in the audio domain, we thought that we should also pass this information to the visual domain. With a video-streaming buffer of equivalent size, time tags for beginnings and endings of events are equally valid and provide a direct video-event analysis. Where the audio source is causally linked to a visual source, audiovisual event capture is possible. For instance, a musician's output can be segmented into note objects, each tagged with the appropriate video snippet showing the production of that event. Algorithmic use of these events can be accomplished as both audio and video playback. Such a segmentation strategy has been proposed in the context of multimedia content analysis (Wang, Liu, and Huang 2000) for semantic description of broadcast material. The concept is extended here to the online situation of live performance work acting on musical events.

Various possible audio segmentation algorithms were compared (Collins 2005a). The most successful for percussive audio files was an adaptation of a psychoacoustically motivated transient-detection method introduced by Anssi Klapuri (1999). A 1,024-point Fast Fourier Transform (FFT) on 44,100 Hz, 16-bit audio with a hop size of 512 samples is taken, and the FFT bin energies are combined according to 40 equivalent rectangular band (ERB) scale bands (Moore, Glasberg and Baer 1997). The detection function (DF) is defined band-wise based on a difference of log energy across successive frames. It was found propitious to adjust the log intensities as decibels according to equal-loudness

Figure 4. Segmentation of a sitar source signal. The top signal view shows the chosen segmentation points as vertical lines. In the middle is the detection function DF, and the peak-picking algorithm output (before final thresholding) is shown on the bottom.



curves to mimic human perceptual bias. International standard ISO226-2003 provided the contour data.

An important issue is how to manage peak picking on this function. One option as shown in the following equations is a generalization of the local maxima test. Equations for the detection function *DF* at frame *n* and a generalized peak-picking function *PP* are given by

$$E_n(k) = f[10 \log_{10}(I_n(k))] \qquad k = 1 \; . \; . \; . \; 40 \tag{1}$$

$$DF(n) = \sum_{k=1}^{40} max \; (E_n(k) - E_{n-1}(k), 0) \tag{2}$$

$$PP_s(n) = (DF(n) - DF(n-s)) + (DF(n) - DF(n+s)) \tag{3}$$

Here, function *f* represents the equal-loudness contour correction, index *k* is the ERB scale band, and *s* is a spread parameter. $I_n(k)$ represents the average intensity of the FFT in ERB band *k*. This was extended to a more complicated algorithm than shown in Equation 3, detailed here in pseudo-code:

```
for i= 1 to length(DF)
    sum=0;
    spread=3;
    //consider local indices up to spread
    units away
    for s= -spread to +spread
        diff= DF(i)-DF(i+s);    //difference
        //negative scores are exaggerated
        if (diff<0) diff= 10*diff;
        sum=sum + diff; //cumulative score
    end
    PP(i)=sum;
end
```

This algorithm scores the "superiority" of peaks with respect to their location. The output *PP* is normalized and compared to a fixed threshold. Figure 4 shows this. Double hits have also been removed by only allowing the onset detector to trigger once every 40 msec; double hits could be segmented as legitimate events by dropping this condition, or an average position of close multiple detections may be taken. The early peaking of the Klapuri function is evident.

The steps required for the audiovisual event capture are now outlined as natural-language pseudo-code. The specific callback structure is avoided for clarity.

1. Run detection function and peak picking on audio stream continuously.
2. If a peak is detected, find the most recent zero crossing as a glitch-free start point; store this adjusted start time.
3. Wait for one of three ending cues: a new peak, a maximal elapsed duration, or a drop-off of intensity by a significant factor. Adjust the end to the nearest zero crossing.
4. Store this information in the event database along with an identification number. IDs may overwrite old IDs where the event is known to have been lost owing to the continual recording of the circular buffer.
5. The start and end times of the event are passed via OSC to the video application, along with the event-identifier number. The times can be converted to appropriate frame locations in the video buffer. On playback of this audio event, the identifier can be sent to the video application to trigger the corresponding video data.

This process requires a delay of at least the length of an event to complete. It should be noted that this event-segmentation technology allows audio-splicing algorithms to show greater sensitivity to their targets and empowers various event resynthesis effects (Jehan 2004).

Although the MIREX2005 Contest's evaluation of onset detection algorithms supported the effectiveness of this segmentation strategy for percussive onsets (this algorithm was the top-performing for drum and bar/bell sounds), its general applicability was questioned, and in particular, performance was severely degraded for cases involving the singing voice and sustained strings. Different techniques for slow-attacking, vibrato-rich signals are necessary (Collins 2005b), and complex polyphonic audio of course provides a further challenge. However, in a modular system, alternative onset detectors can be substituted as required.

Audiovisual Mappings

The techniques presented previously involve refined one-to-one mappings between audio and visual modalities. It was seen as important to gain control of such synchrony so that any deviations from correlation could be more carefully controlled. However, injective mappings are not the only possibility in artistic practice. Cook's model of multimedia (Cook 1998) proposes three mapping strategies between audio and visual information: conformance (one-to-one correlation), complementation (of broad meaning but not events), and contest. Of these, Cook claims that only the second two are true exemplars of multimedia. We would argue, however, that contest and complementation are most convincing as options in VJ practice when synchronization can also be demonstrated. Perhaps it is inevitable, however, that the development of new technologies for conformance of sound and visual events encourage one-to-one mappings in current artistic practice while these techniques are fresh.

As a salutary reminder to those who would avoid controlling connections between audio and visual, the anti-correlation experimental films of Oscar Fischinger in the 1930s show the willingness of human observers to impose points of synchronization on independent visual and auditory streams, a human tactic that has an evolutionary basis in our causal expectations of scenes. Lipscomb and Tolchinsky (2005) call this *accidental synchronization,* and Chion proposes the term *synchresis* in the context of a propensity to merge (synthesize) events presented in synchrony (Chion 1994, p. 63). A cognitive basis for such integration is explored by Phillips (1999) in a theory of audiovisual scene analysis, after Bregman (1990). She is forced into such a position in her study of film music by the realization that the combination of audio and visual information may transfigure information from either modality alone, and that conventional accounts of film music lack a firm scientific basis for such investigation.

Lipscomb and Tolchinsky (2005) actually propose a continuum between simple stimuli (such as Fischinger's animated shapes) where synchronization points dominate, to complex stimuli (such as video footage), where association (meaning) predominates. In the narrative-free, information-overload jump cutting of the VJ, consensual meaning is somewhat left behind, so that the rapidity of the onslaught itself takes on the central role, and the pacing of synchronization points is the chief concern. Chion notes that "what becomes significant . . . [are] . . . changes in . . . tempo or appearance" (Chion 1994, p. 163).

The equality of modalities in an audiovisual collaboration may be a powerful dream, but it is unrealistic with respect to the different processing rates and attentional status (Posner, Nissen, and Klein 1976; Pöppel 1997) of visual and aural information. Typical sampling rates for music and film give this away: 44,100 Hz versus 24 Hz! Chion (1994, pp. 10–11) notes the differences in speed of perception and praises the added value of audiovisual union, the spotting of visual events by the more rapidly perceived audio modality. This enables an illusion of a greater rate for the union than that provided to the eyes.

Multimedia experiments are often traced back to Castel's proposed *clavecin oculaire* (1725/35), Rimington's color organ (1893), Scriabin's Prometheus (1911), and other synesthetic landmarks. In an interesting position statement, Amy Alexander (2004) has recently suggested that in a mature culture confronting non-causal image and sound, avoiding simple synesthetic correlations gives rise to a deeper artistic statement. In this regard, she undermines those who deplore the loss of gestural correlation in laptop music and particularly real-time programming practice and stands in favor of contrast and conflict as compositional factors for audiovisual work. While acknowledging it cannot stand alone, we do not renounce correlation however, believing that such conformance in respect of live capture restores a certain gestural connection that audiences may appreciate as an entry point to electronic media work. Alexander's position also has the danger of being more likely susceptible to individual interpretation.

klipp av Mappings

For klipp av, a core part of their performance is the selection of mappings controlling the exchange of information between their two laptops, one geared to sound and the second to video and computer graphics. Both computers are able to capture current environmental information, the former via microphone, the second by video camera. Figure 5 reveals the performance set-up and shows the flow of information. The richest event information arrives from the audio stream, so we shall concentrate here mainly on the perspective of the visual artist and how they may exploit this data.

For the BBCut information, a first selection is performed by ear, selecting significant sound sources (cutter instances) to track. The video application's user interface gives an overview of cutters that are currently providing data and allows for quick filtering via an instance ID tag. Other pages display incoming data to help make the selection. The option exists to use all or just some of the different parameters of each sound source. Different functions can be filtered out or muted, for example, disregarding the very fast repeats in Figure 1 and following only block boundaries.

For hard-synced, one-to-one correlation, only one sound source at a time is tracked—typically the most dominant one based on subjective listening, or verbal agreement between the artists, who play side by side. Tracking more than one cutter actually establishes a counterpoint that is increasingly tricky to integrate as an audiovisual correlation. The combined rhythms may form a new and separate gestalt unless the streams are carefully segregated by the use of spatial position (perhaps split screens) or constituent visual object properties.

Owing to the slower visual sampling rate, filtering is necessary to compensate for faster-than-frame-rate audio. Given an integration time of 30 msec (Pöppel 1997), a 33-Hz frame rate is probably optimal. Owing to aliasing, only rhythmic events up to 16.5 Hz are representable; this covers the range of rhythmic (haptic) events but will not track audio-rate pitched stutters, for instance. In practical terms, however, it is sufficient to convey an impression of correlated energy, so if the video frame alternates at its maximal rate while the audio goes much faster, the integrated percept will remain convincing. Figure 6 portrays this situation; the reaction time delay of up to a frame and the correlation of "strobing" activity level to audio-rate repetitions is readily seen.

For the video-scratching techniques discussed previously, klipp av has devised a few successful strategies of mapping data to fixed and circular buffers of video. Different approaches offer more or less correlation, and this is also a parameter under control of the user. The most obvious one would be a direct mapping of the read-buffer position of the sound source to the video buffer's read position. Here, audio and video captured at the same time will stay exactly in sync, and "digital scratching" or cut-ups in the audio domain follow over to the video. A variant of this technique is to let every other stutter message change the playback direction of the video. This works particularly well with captured footage of people dancing. A more abstract mapping strategy is to detect when the audio stutters and then independently pick an offset for the video buffer. Now the audio and video use the same overall phrasing and stutters but differ in source material and order of events.

To add to this, cut-sensitive effects can enhance or disrupt the feeling of audiovisual correlation. Especially effective are strobes that invert the video and different kinds of mirror effects that can flip parts of the picture with each repetition. If these are

Figure 5. Simplified depiction of OSC message-passing between audio and visual machines in the klipp av live setup. Input sources are in octogons, modules are in rectangles, and message categories are in ellipses. "Composer" here refers to human users, their interface, and their choice of algorithms, thus encompassing creative decision-making regarding mappings.

Figure 6. Individual video frames shown against audio events.

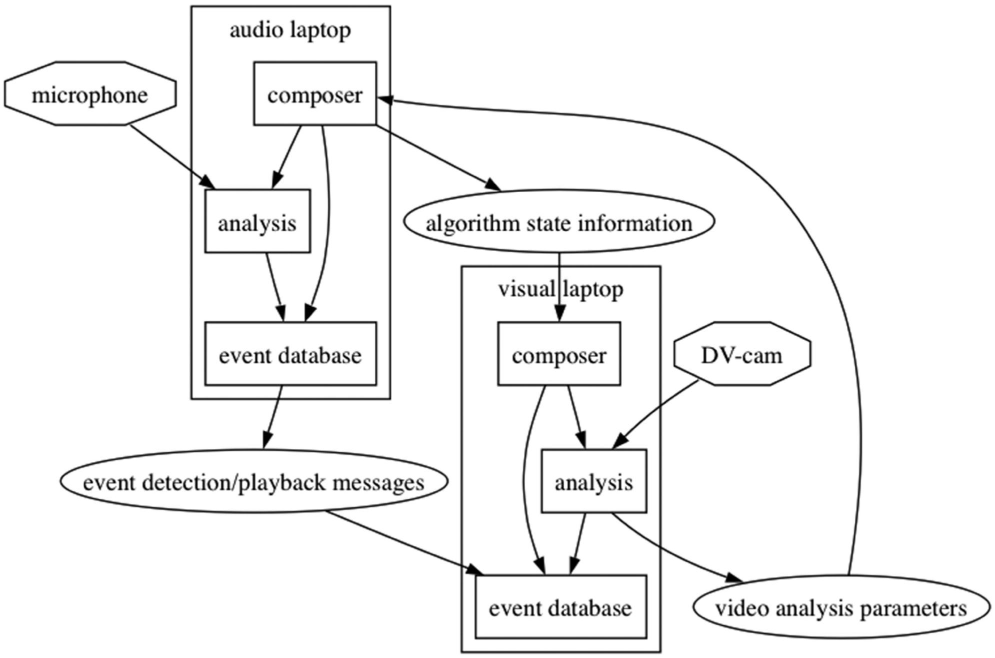


Figure 5

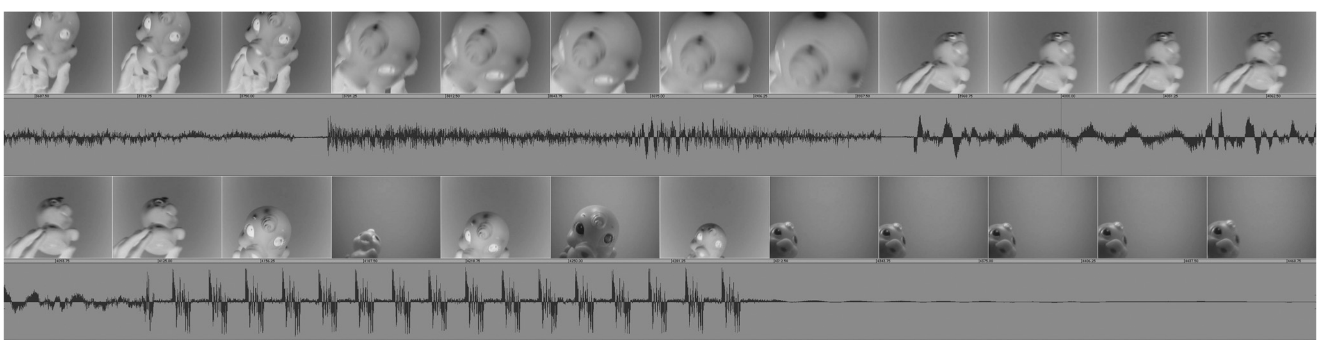


Figure 6

mapped to stutters and used in combination with the jumping-offset strategy mentioned above, a very harsh and evident feeling of one-to-one synchronization is obtainable.

In addition to the cutting data, the video application receives data about discrete captured events and their playback. Again, how directly this information is used forms part of the mapping strategy improvised by the visual performer. A performer might decide to reverse the video segment associated with a given event on playback, or substitute a synthesized visual event of appropriate or inappropriate duration. Such decisions determine very quickly whether the audience observes a tight synchronization or an abstract or absent conformance.

Alongside video, klipp av has experimented with the control of generative graphics. Perspective shifts and shape and position changes have been based on cut sequences. Particularly successful were explosions of activity during rolls; by tracking the number of repetitions in a block, the extreme parameters that most modify the vertex information are reached at the extremes of activity.

The directness of any mapping is up to the specific composition code, and each partner can choose to ignore or pervert their input data. In the klipp av system, the data sent between applications can be interpreted and transformed in numerous ways. During performance, the video artist in particular is highly involved in modifying and filtering this data and also actively chooses which control information is mapped to which parameters in the visual domain. This allows instant changes between Cook's three mapping strategies, and the amount of explicit synchronization between audio and visuals becomes a very interesting performance parameter. It is perfectly possible to emphasize more suppressed musical elements or to ignore, transform, or record data for later "contests." Such strategies challenge the musician by exploring an abstracted parallel or opposite direction to the obvious mappings.

A circle of information flow has been completed by passing control messages from the video stream to the audio based on methods such as RGB-analysis of the incoming or outgoing visuals (see Figure 5). The output control data is indirectly perturbed by the input control data on both machines. This facility actually empowers audiovisual feedback; the video stream is perturbed by the rendering state of audio algorithms themselves under video control. Such abstract feedback is a complex factor of the performance setup as a dynamical system and may act on a continuum of abstruseness. Such aspects of distributed and negotiated control were first exploited by early network bands such as The Hub (Dean 2003).

Through these means, klipp av has made the act of mapping a central concern of the improvisational process. Various demonstrations and extracts of live performances are available from www.klippav.org.

Conclusions

There are many difficulties in allowing the general public to directly control musical and visual output (Ulyate and Bianciardi 2002), but klipp av are supportive of the use of randomly selected participants as audio and video sources in live capture. One of our favorite moments from a live show involved a press photographer who tried to come onstage to photograph us and found himself the subject of audiovisual cut-up. We have also developed technologies that make working with acoustic musicians (in an event-sensitive way) a natural step. Photographs from two recent performances are shown in Figures 7 and 8.

An important technical theme of this article has been the exploitation of audio analysis to inform video analysis. Event segmentation in the audio domain was leveraged to segment video events. This led to a notion of audiovisual events as manipulated entities in performance. But we have also discussed completing the cycle of information transfer to encompass video analysis informing audio composition, and closing the cycle leads to "audiovisual feedback."

All of this work is to support the improvisational scope of electronic media performance. As detailed herein, decisions about mappings play a vital role in the adaptation to a particular environment. Algorithmic composition respecting live source material and its embodiment in multiple modalities is a broad field within which to explore new possibilities of performance.

References

Alexander, A. 2004. "Live Coding is Not Synaesthesia." Available online at www.toplap.org/index.php/LivecodingIsNotSynaesthesia.

Betts, T. 2002. "Pixelmap." *Proceedings of Cybersonica 2002.* London: ICA.

Bregman, A. 1990. *Auditory Scene Analysis.* Cambridge, Massachusetts: MIT Press.

Brossier, P., J. P. Bello, and M. D. Plumbley. 2004. "Fast Labeling of Notes in Music Signals." *Proceedings of the 5th International Conference on Music Information Retrieval.* Barcelona: Universitat Pompeu Fabra, pp. 331–336.

Chion, M. 1994. *Audio-Vision.* New York: Columbia University Press.

Collins, N. 2002. "The BBCut Library." *Proceedings of the 2002 International Computer Music Conference.*

Figure 7. Photograph from a live klipp av performance at State51, London.

Figure 8. Photograph from a live klipp av performance at Tank, New York.

Figure 7

Figure 8

San Francisco, California: International Computer Music Association, pp. 313–316.

Collins, N. 2004. "On Onsets On-the-Fly: Real-time Event Segmentation and Categorisation as a Compositional Effect." Paper presented at Sound and Music Computing, Paris, October 20–22.

Collins, N. 2005a. "A Comparison of Sound Onset Detection Algorithms with Emphasis on Psychoacoustically Motivated Detection Functions." Paper presented at the 118th Audio Engineering Society Convention, Barcelona, 28–31 May.

Collins, N. 2005b. "Using a Pitch Detector as an Onset Detector." Paper presented at ISMIR 2005, London, 11–15 September.

Collins, N., and F. Olofsson. 2003. "A Protocol for Audiovisual Cutting." *Proceedings of the 2003 International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 99–102.

Cook, N. 1998. *Analysing Musical Multimedia.* Oxford: Oxford University Press.

Dean, R. 2003. *Hyperimprovisation: Computer Interactive Sound Improvisation.* Middleton, Wisconsin: A-R Editions.

Heap, M. 2003. "The History of VJ." Available online at www.kether.com/SFPCS/Heap-historyVJ.html.

Hyman, J. 2005. "Scratch Video and TV Sampling." Available online at www.jameshyman.com/blog/archives/000089.html.

Jehan, T. 2004. "Event-Synchronous Music Analysis/Synthesis." Paper presented at the 2004 Digital Audio Effects Workshop, Naples, 5–8 October.

Klapuri, A. 1999. "Sound Onset Detection by Applying Psychoacoustic Knowledge." Paper presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Phoenix, 15–19 March.

Lew, M. 2004. "Live Cinema: Designing an Instrument for Cinema Editing as a Live Performance." Paper presented at the 2004 Conference on New Interfaces for Musical Expression, Hamamatsu, 3–5 June.

Lipscomb, S., and Tolchinsky, D. 2005. "The Role of Music Communication in Cinema." In D. Miell, R. MacDonald, and D. Hargreaves, eds. *Music Communication.* New York: Oxford University Press, pp. 283–304.

Matsuda, S., et al. 2002. "DIPS for Linux and Max OS X." *Proceedings of the 2002 International Computer Music Conference.* San Francisco, California: International Computer Music Conference, pp. 317–320.

McCartney, J. 2002. "Rethinking the Computer Music Language: SuperCollider." *Computer Music Journal* 26(4):61–68.

Moore, B. C. J., B. R. Glasberg, and T. Baer. 1997. "A Model for the Prediction of Thresholds, Loudness, and Partial Loudness." *Journal of the Audio Engineering Society* 45(4):224–240.

Phillips, N. 1999. "Audio-Visual Scene Analysis: Attending to Music in Film." Ph.D. thesis, University of Cambridge.

Pöppel, E. 1997. "A Hierarchical Model of Temporal Perception." *Trends in Cognitive Sciences* 1(2):56–61.

Posner, M., M. Nissen, and R. Klein. 1976. "Visual Dominance: An Information-Processing Account of Its Origins and Significance." *Psychological Revie*w 83(2):157–171.

Rossignol, S., et al. 1999. "Automatic Characterisation of Musical Signals: Feature Extraction and Temporal Segmentation." *Journal of New Music Research* 28(4):281–295.

Snider, H. 2000. "Scratch Video: A Mutant Hybrid of Scratch DJ Music and Guerilla TV." Master's thesis, University of Concordia. Available online at www.artengine.ca/scratchvideo.

Ulyate, R., and D. Bianciardi. 2002. "The Interactive Dance Club: Avoiding Chaos in a Multi-Participant Environment." *Computer Music Journal* 26(3):40–49.

Wang, Y., Z. Liu, and J. Huang. 2000. "Multimedia Content Analysis Using Both Audio and Visual Clues." *IEEE Signal Processing Magazine* 17(6):12–36.

Wright, M., and A. Freed. 1997. "Open Sound Control: A New Protocol for Communicating with Sound Synthesizers." *Proceedings of the 1997 International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 101–104.