Exploring and Constructing Video in Improvisational Manner

Paul Nemirovsky
The Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: pauln@media.mit.edu

Gene Shuman
The Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: gshuman@mit.edu

Abstract

How can machines help us to manipulate and structure audiovisual media in ways that are always novel and are uniquely ours? How can such construction happen in real time, with no precise planning or guidance given by the user? The Emonic Environment (EE), the system described in this paper, enables improvisational construction and navigation of media space, both by individuals and by groups. Participants either control the system directly (e.g., real-time recording, processing, and performance of audio, video, and text, or exchange with remote users and online databases), or provide only a higher-level structural guidance, letting the underlying genetic algorithms control the low-level details. The system's behaviour and content is controlled utilizing keyboard/mouse, as well as microphones, cameras, sensors, MIDI controllers, and cell phones.

This paper focuses on the new aspect of the EE: a capability to manipulate video, and the methods by which the system's control structures, assisted by genetic algorithms, make it possible for the participants to manipulate video without having to attend to the minute parametric details.

Characteristics of improvisational action are described as is the rationale for our particular design. Two architectural notions aimed at encouraging real-time structural thinking in the creative context are introduced: these of *content abstraction* and *structural control*. This is followed by a description of a relevant subset of the EE's implementation: temporal and new video elements. We conclude with remarks on future work.

1. Introduction

The Emonic Environment (EE) is a system for media creation and exploration based on characteristics of improvisational action, detailed below and in [1]. The EE's architecture allows its users to manipulate video, audio and text in a nearly identical fashion. Additionally, the EE utilizes genetic algorithms to introduce mutations of the media space (the content and its controlling structures) that might be of interest to its users. The EE's aim is: to draw people into exploring structural behaviours (strategies) rather than individual aspects of the media space.

One of the primary characteristics of improvisational action is *content abstraction*. In the context of human-computer interaction, content abstraction means presenting users with tools for structural control of the media space (i.e., control of its density, intensity, and behaviour over time), while making the particulars of media manipulation controllable by the machine (or other users). Two conditions are necessary for content abstraction: (1) functions used to control media need to be as medium-independent as possible (video, sound, and text controlled in the same fashion) and (2) it should be possible to easily make links between these functions and an abstract neural network that provides the structural control. In other words, the EE's premise is to abstract away from the low-level control of particular media properties and towards a higher-level functional view of the media space, leading the participants to consider such configurations of the space that they would not arrive at on their own. To facilitate such functional view of the space, its components need to be modular, easy to interconnect, and be capable of manipulating the entire spectrum of media. The previous version of the EE was limited in that regard – it worked with audio only. In this paper we describe our attempt at bringing the EE closer to the idea of content abstraction by incorporating video manipulation.

The rest of the paper is as following: we start by presenting an updated description of characteristics of improvisational action, essential for understanding the paper. We then discuss the rationale for our design – the notions of *content abstraction* and *structural control*. This is followed by a description of a relevant subset of the EE's implementation: temporal and new video elements. We conclude with remarks on future work.

2. Characteristics of Improvisational Action

Improvisational action, defined here in the context of human-computer interaction, implies that no predefined rules, plans, or objectives of the action exist. Instead, the media space being explored is regarded as an evolving structure, its configuration guided by real-time genetic algorithms and human feedback to continuously restructure itself. The ten characteristics that follow are inspired by a mix of non-idiomatic improvisational and experimental music traditions ([2], [3]).

1. **Changing, Multi-level Focus.** Whether deciding what to do next, or reflecting on past actions, improvisers employ different levels of abstraction simultaneously. Switching between representations, they attend to the minute details at one moment, only to shift to looking at an overall structural development (e.g., a climax) a second later.

2. **Dynamic Structural Rules.** Improvisation, like composition, has rules concerning elements' interrelationship in terms of time, volume, and other perceptual characteristics. Unlike in composition, however, these rules (and the structure they comprise) need not be predetermined. Instead, they are often created and modified on the spot, defining the improvisation's character. For instance, an improviser may spontaneously decide to repeat a given motive every few seconds, or increasingly desynchronize two ongoing motives (regardless of what the actual motives are at the moment).

3. **Absence of Static Plan.** Choosing the path to follow in exploring and structuring media is a dynamic process that happens as the improvisation unfolds. Instead of first creating the structure and then filling the content in the blanks, improvisational plans are generated and evaluated on the spot. As a result, improvisers are not too concerned with following an existing framework in a perfect manner, focusing instead on creating new plans and learning from unintended mistakes and unexpected successes.

4. **Absence of Authoritative Score and Price of Mistakes.** Improvisational performance, unlike that of a composed piece, cannot be quantitatively compared with a pre-existing deterministic score. As a result, the notion of 'mistake' is modified from that of a non-compliance with predetermined solutions to a higher-level non-compliance with aesthetic expectations. As a result, initiating an unorthodox action becomes less threatening, and risk-taking is encouraged, due to the fact that unlike a performance of a composed piece

evaluated against an authoritative score, aesthetic expectations of improvisation cannot be evaluated on a note-per-note basis. Generating new actions becomes easier than following predetermined ones, with the improvisers being free to exercise as much control over the creative action as they desire yet avoid the responsibility that conformance to or creation of a deterministic score implies.

5. **Process, not Artefact Production, as the Goal.** An improviser, unlike a feature-film cinematographer, a Western composer, or a product designer, is not concerned with producing a final viable artefact – a movie, a sonata, a pop song, or a chair. While improvisation might be recorded and, as such, seen as a fixed construct, improvisation is primarily a process of exploration, contextualizing and interrelating memories, perceptions, and actions. Improvisers weave together an array of 'sketches,' which gain relevance and meaning only as the improvisation unfolds. The importance of individual elements lessens in favour of that of the paths by which these elements appear, become significant, and disappear – that is, the strategies for exploring the overarching structure. Improvisers employ these strategies to find structure within chaos - only to break it again a moment later, and start looking anew.

6. **Relevance of Context.** Improvisers do not follow a score; as a result, their decision-making is guided both by explicit actions (their own and others') and their perception of the moment in its entirety. In other words, improvisation is not formed in a vacuum; it merges between explicit decision-making and implicit context-gathering from the environment in which the improvisation is being created.

7. **Distributed Responsibility and Control.** In an improvisational performance, no fixed contract specifying responsibilities of control (i.e., a balance of power) exists between the performers. The degree of control assumed over the improvisation by each participant is set dynamically, following implicit and explicit negotiations. Improvisers are always free to renegotiate what and how they control during the course of improvisation, thus freeing them from preoccupation with every aspect of the creative action, and making experimentation easier.

8. **Audience as a Participant.** From the passive audience of linear storytelling systems to the nearly equally passive audience of multiple-choice interactive environments, a strict giver / taker dichotomy has been enforced between the consumer (the audience) and the producer (the performer). In the context of improvisation such a distinction is obsolete;

anyone can co-improvise, so long as the effect of his activity is seen or heard in some way by the other performers. As a result, any audience becomes a pool of potential participants which, even when not actively participating in the act of media creation, are regarded as a part of the improvisational circle.

9. Timeframe and Obligation to Participate. In the compositional paradigm, performer's participation is to last as long as is required for the performance of the piece to be completed. Walking out in the middle of the creative action means that the action is stopped until the active participants' return. With time, social norms have emerged to prevent such walkouts, as manifested in events such as concerts of classical music: events that force the participants into a highly ritualized act of performance with no escape until the end of the action is reached. Improvisational setting, on the other hand, allows for a more relaxed participatory mode: people are no longer forced to a particular length of non-stop participation and are free to commence / end their participation whenever desired.

10. Immediacy of Feedback. Unlike the compositional paradigm, with its sequential two-layered creative process (compose, then perform), improvisational action blends the two. As a result, both structures and content are evaluated and incorporated (or rejected) in real time, whether coming from one of the improvisers or from an external source.

3. Design and Rationale

3.1. Time-based Content Abstraction

Media systems today are typically limited in their treatment of time and context[1], when considering placement and relevance of a given media content. In traditional media systems, content elements follow each other in a fixed, predetermined order. In loop-based systems, content elements occur at set intervals for the duration of the loop condition, with temporal relationships defined between individual pieces of content. In both cases, the focus is on the content rather than on the structure. As a result, temporal characteristics are hard to abstract (e.g., instead of arranging events on a timeline, specify that element A is to occur *m* times for each *n* occurrences of element B). The bigger problem however is

[1] The EE incorporates an extensive array of tools for sampling the context; discussion of these capabilities is beyond the scope of this paper.

that the temporal structures (e.g., the relationship between elements A & B at any given point of the performance) are fixed, in their location and interrelationship with other temporal elements unless a manual change is performed by their users. In designing content abstraction into the EE's temporal elements, our first requirement therefore was to provide a unified temporal treatment for audio, video, and text. The elements had to be modular and capable of dynamically changing their placement in the events chain.

Media content is often viewed as pattern-able (e.g., endless available libraries of drum loops). However, learning, making apparent and manipulating *action patterns* (elements' interrelationships over time) is still absent from real-time interactions in computer context.

Our second requirement in for content abstraction was making the EE's temporal elements responsive to real-time evolutionary process and users' feedback.

Finally, synchronization of media is also typically limited to a within-component synch. For instance, audio and video may be synchronized within a video clip, yet the 'behaviour' of their synchronization – that is, the change in synch between the two over time – is rarely scriptable on its own. Our third requirement was to make construction of such action patterns possible, both by users and by the machine.

3.2. Structural Video Manipulation

Nowadays most people with no expertise on sound and video editing see them as qualitatively different. Sound is seen as fluid and open to manipulation (anybody can create audio effects with their own mouth) while video is seen as fixed and its manipulation is considered to be a complex and cumbersome affair. By incorporating video within the Emonic Environment we hope to show that video can be manipulated very similarly to sound: real-time and unconstrained by the initial shape.

Today most of the off-the-shelf video editors regard video manipulation as an *editing process*, performed in stages and aimed at producing a fixed result: first shoot, then digitize, then position on a timeline, then cut into fixed pieces and connect by fixed transitions. The end output is similarly fixed, and, intriguingly, almost always square in shape. More than twenty years after the release of interfaces such as Fairlight [4], real-time video manipulation interfaces mostly remain complex, expensive, and without the type of instantaneous plug-camera-and-manipulate interaction that would be appealing to many of today's computer users. Systems that do allow such interaction require custom hardware or a solid understanding of visual manipulation principles (e.g., VJamm [5]).

The EE is currently focused on sample-based media, rather than on a purely algorithmic processing. The rationale for that is quite simple – people love seeing and hearing themselves and may possess a more intuitive understanding of the core elements behind sample-based media (i.e., photos, videos) than of the parameters behind a purely generative art. Additionally, operating on samples provides us with the added benefit of being able to utilize the vast amounts of sampled media currently available in free domain.

Some of the concepts mentioned in this paper are familiar to users of the more novel video editors (e.g., Jitter [6]), which allow, effect-wise, video manipulation similar to and beyond the one described in this paper. The core difference between such editors and the Emonic Environment is the added possibility of *structural control*, assisted by genetic algorithms. Structural control implies that the participants can focus on higher-level control strategies, leaving the manipulation of component-specific parameters to the machine (or collaborating users). In that way, media contributed by a participant can be changed in ways its contributor would not consciously consider or imagine. To effect change, the user only needs to interact with generalized structural controllers (unless he desires a precise function control).

Prevailingly, a media space (e.g., a video clip or an audio database) is viewed today as consisting of discrete elements upon which media manipulations are performed. Integration of structural control and content abstraction is aimed at presenting a network view of the media space; a network of structural elements controlling media behaviour over time, regardless of the particulars (media type, content, etc). If we succeed in getting people closer to realizing that the same structure can be used to control differing content we bring our users a step closer to becoming creators thinking about context and structural development.

We believe structural control and content abstraction to be a powerful way to encourage users to think structurally. Traditional editors require first learning abstract concepts and then applying these to media content in order to design finished structures (pieces). In the EE, users play with content while figuring out how it constitutes a structure. By browsing through disjoint content elements and manipulating how the content is controlled over time, users end up creating structural control networks (or use the ones suggested by the machine).

4. System Architecture and Implementation

The EE is written in Java with auxiliary C libraries and contains over forty thousand lines of code, manipulating audio, video, and text, in shared and individual contexts, and utilizing sensors, microphones, cameras, and cell phones as input devices. As such, the features described below represent only the fraction of the EE relevant for understanding the video manipulation capabilities presented later in this paper. We start by describing the overall architecture, and follow by describing a number of elements that constitute the core of medium-independent processing.

The EE consists of two main processing layers, *Perceptual* and *Structural*, each represented as an independent network of interconnected elements, and an auxiliary *Mediated* layer, connecting the two main layers. Simply put, the Perceptual layer defines *what* and *how* we hear, see, and read, while the Structural layer defines *when* and *why*. The operation of the system is purely real-time, with no offline processing.

The participants interact with each layer in the following three ways: (1) directly setting the individual components' properties within each layer, (2) providing feedback to the built-in genetic algorithms that are evolving the network's state, and (3) contributing and exploring media and defining its interrelationships with the control structures.

4.1. Perceptual Layer

The Perceptual layer is populated with *emons*[2] of various types, each with its own set of features. The emons control how the media is generated, modified, and played back (e.g., speed or volume of a sound, rotation angle of a video frame, semantic relationships of a piece of text). Emons' modular architecture (ability to interconnect with other emon types) allows for the creation of nested processing structures. Overall, emons can be thought of as a media-processing engine where, by combining multiple types of emons, the Perceptual layer can be built to taste and reconfigured in real time. Viewed sequentially, the processing can be seen to originate from a single, repetitive

[2] Emon: a media-processing functional primitive; combined together, emons form interconnected structures for generation, modification and presentation of media.

beat; propagating through tempo adjusters to become faster or off-beat; propagating again through filters which alter the overall pattern of emons' processing; and finally ending with a collage of temporal signals interpreted by effect producers, retrieval mechanisms, and audio/video/text players, with the result being output into one or more physical environments.

4.2. Structural Layer

The Structural layer is populated with *nodes*, structural constructs entwined to create a network providing the participant with a higher-level abstracted instrument for observing and influencing the ongoing activity within the EE. The layer is modelled as a recurrent neural network, allowing evaluation of events' propagation over time and concerned solely with change of its elements' activities. As a purely abstract system for controlling objects, it plays no role unless connected to the Perceptual layer. Each node has its own activation level, its value continuously decaying. The nodes communicate by sending stimuli, which can originate at any point in the network. When a node is triggered, implying that its activation level is beyond a propagation threshold, it sends stimuli of proportional strength to all the connected nodes. The growth and decay in the nodes' activation levels can be controlled individually or en masse, by participants, other nodes, MIDI controllers, sensors, or the ongoing evolutionary process.

4.3. Mediator

The Mediated layer maps the neural activity of the Structural layer onto the media processing activity of the Perceptual layer. The mapping is reconfigurable in real time and unifies the layers into a framework for improvisational action. The Mediator allows each Structural *node* to have an unlimited amount of thresholds that indicate that some action is to be taken. Each time a node passes one of its thresholds, a corresponding *Mediating Action* is performed. Mediating Actions consist of one or more functions that use the information received from the associated node to modify one or more property of the Perceptual emons. Mediating Actions may be very precise, controlling low-level particulars of a given emon (e.g., 'change the strength of the green channel of the associated video file to the next within the provided envelope of data points') or high-level (e.g., 'desynchronize the ongoing audio events a bit more'). Mediated Actions are also tradable with other participants within the EE. The Mediator insulates the two main layers, allowing for their independence and thus making it potentially possible to replace either layer with a different type of controller or media system if desired. Our method for navigating (evolving) media networks utilizes the Mediator to access the properties of components located at both Structural and Perceptual layers. The genetic algorithms that facilitate the navigation mutate the properties, biased by the participants' feedback and currently active constraints on the evolutionary process.

4.4. Emons

Multiple emon types exist within the EE. For brevity sake, we only describe two categories of emons: (1) system emons, which deal with temporal properties of the EE and apply equally to audio, video, and text, and (2) video emons.

All the emons can be activated by (1) connected Tempo emons, (2) Actions designed by the participants or (3) ongoing evolutionary process driven by built-in genetic algorithms and users' feedback.

Emon :: Master Beat (MB). MB is the emon that provides the pulse to be used by any other elements within the Perceptual network as a synchronization reference. On each of its beats, it fires signals to all of the directly connected emons, thus, prompting action within the rest of the system. MB's tempo can be changed at any time, with a slower tempo resulting in a network in which numerous actions occur within each beat, while a faster tempo contributing to a fast-paced but less intricate network. Changing the MB's tempo can have an effect on the entire 'attitude' of the network.

Emon :: Tempo Cycle (TC). TC emons depend on a supplied 'parent beat', breaking it into smaller, more distinct bits of time. By nesting TC emons, participants create complex polyrhythmic temporal structures without having to understand concepts of music theory and composition that deal with time. TC emons can also introduce a time delay, thus propagating a beat off-synch from the original beat. Such time variance can be utilized to create an echo or stagger effect and make asynchronous behaviours possible.

Emon :: Action Filter (AF). No system is complete without the ability to ignore the directives provided by others. Such 'resistance' of an element within a network can be formalized using the concept of a mask or a filter. The Filter emon implements a mutable 'pattern of resistance', propagating events at its own discretion, filtering the through-coming data in accordance with a preset or a dynamically defined mask. The filter is a binary array, signalling which events should and should not propagate. It operates in a circular fashion, looping through the array in synch with the input events it is filtering. Filters are useful in media performance scenarios, allowing for variation in frequency of the output. For example, instead of a simple repetitive chain of events playing an audio sample every beat, the filter allows for a structured but non-monotonous repetition.

Emon :: VisualSample (VS). VS allows playback of video and still pictures. As such, it parallels the functionality of AudioSample and TextSample emons. To play a picture or a video, participants associate the emon with a video or an image, stored locally or remotely, that they would like to manipulate. Using Quicktime [7], individual frames of the video are extracted and passed to JOGL [8], an OpenGL library we use for video manipulation. VS emon's properties include
(1) Start / Stop cues defining the limits at which the playback begins and ends; (2) individually controllable prominence of the video's red, blue, and green channels; and (3) opacity control defining the video's prominence within the Master Visual Player emon's display space – its visibility given multiple videos layered on top and within each other.

Emon :: Master Visual Player (MVP). MVP emon is responsible for the overall video playback, resulting from the sum of all the ongoing visual manipulations. As such, it parallels the Master Audio Player and the Master Text Player emons. Its properties include (1) window size; (2) display ID to allow multi-display scenarios with the network placed on one screen and the visual result on another; (3) coordinates on screen (modifiable by Actions and evolutionary parameters to make the video move on screen); and (4) layering of active VisualSample emons.

Emon :: Visual Subdivide (VSD). VS emon divides and replicates its input (a VisualSample or another VSD emon). VSD emon's properties are two arrays of ratios defining how the original size of the incoming image will be divided on X and Y axes, with each division becoming a clone of the original image. For example, setting the X property to [1, 2, 3] will result in three horizontal copies of the source signal, arranged side by side, with the ratio of their relative sizes being 1 : 2 : 3. Such subdivision of space can be seen as analogous to subdividing time by Tempo emons.

Emon :: Spatial Position (SP). The SP emon takes its input and performs a spatial positioning transform on it. In other words, it takes a video or a picture, and reshapes it according to the 3D space coordinates (X, Y, and Z). SP emon's properties are (1) the X, Y, and Z coordinates that define (translate) how the input signal is transformed in 3D, and (2) TH, PHI, and PSI that define the input signal's rotation about the X, Y, and Z axes respectively. Multiple SP emons can be chained off each other, easily creating complex transformations of the source signal.

Emon :: Visual Mask (VM). VM emon allows the participants to define sub-areas within a particular video that will be played (with the rest of the image ignored). The rationale for the existence of the VM emon is that sometimes only a particular part of the image presents interest, or, alternatively, placing only a particular part of the image in a given context results in the construction of a new meaning. VM emon's properties are a set of coordinate points (presented as a black-and-white mask) that define the boundaries of the region that will be processed.

Emon :: Time Scale (TS). TS emon takes an incoming video stream, and slows it down or speeds it up by a requested factor. To exemplify the modularity of emon construction let's consider the possible scenarios:

1. The participant makes three clones of VisualSample emon playing the same video sample and defines each of them to display only one of the RGB channels. Inserting the TS emon in the pipeline between only one of the clones and the MasterVisualPlayer emon, results in one of the channels being desynchronized from the others by a fixed amount.

2. The participant takes the configuration described in #1, and adds a VisualMask emon in between the VisualSample and TS emons. Now, only a part of this particular channel will be visible as it is scaled, thus allowing for time-scaling of subparts of a single video stream.

Emon :: Visual Input Grabber (VIG). VIG allows real-time acquisition of video for subsequent manipulation, storage and/or broadcast. The source of the video can be either a video camera attached to the computer or a web source (a streaming file). VIG emon is useful for introducing continuously changing media source into the performance and parallels the AudioInputGrabber emon. The number of VIG emons is limited only by the amount of live video sources available.

Emon :: OutputStream (OS). Sometimes it is useful to aggregate everything that happens in a given network in terms of the audio, video, and text manipulation, and forward it, sans any structural information, somewhere else. For example, if one participant creates networks that consist solely of percussive rhythms, another participant may want to use these as a rhythmic basis for his own creations. This is where the OutputStream emon comes in. Any participant may connect his MasterPlayer emons into the OS emon, thus indicating that anything that happens to be played or displayed by the MasterPlayer emons is to be made available for broadcast. With the OS emon activated, anyone who uses Audio/VideoInputGrabber emon in their network may point it to the address of the broadcast computer and get the stream as the input.

5. Conclusions

This paper discusses the Emonic Environment, a system built on the principles of structural control and content abstraction. We argue that these two principles are essential in a media system if we are to effect change in how people think about creating and exploring media.

We focus on describing our work at extending the EE's improvisational structure to accommodate video manipulation. We present a solution that (1) allows such manipulation in real-time, (2) falls in line with the currently available audio and text manipulation capabilities of our system and (3) allows participants to move between being intimately involved in the details of the control or conversely, attend only to the high-level details of what is going on.

The next steps in the development of the EE include making it more accessible for novice users as well as conducting experiments that utilize new video capabilities.

6. Acknowledgements

The authors thank Ariadna Quattoni for her help.

7. References

[1] Nemirovsky & Guillaume

[2] Nyman, M. (1999) Experimental music: Cage and beyond, Cambridge University Press, Cambridge.

[3] Cage, J. (1966) Silence; lectures and writings, MIT Press, Cambridge.

[4] FairLight Computer Video Instrument, audiovisualizers.com/toolshak/vidsynth/fair_cvi/fair_cvi.htm

[5] VJamm, http://vjammpro.com/vjammpro/

[6] Jitter, http://www.cycling74.com/products/jitter.html

[7] Quicktime, http://quicktime.apple.com

[8] JOGL, https://jogl.dev.java.net