The Cybernetic “Trap” Kit: Interface Design Considerations for Virtual Augmentation

Steven Lewis
Integrated Composition, Improvisation and Technology
University of California, Irvine
smlewis2@uci.edu

ABSTRACT

The CV Drum Tracker is a cybernetic extension of the drumset that aims to reshape the performer's interaction and musical relationship with the acoustic instrument. It prompts users to embrace a systems-oriented, iterative method of design and performance, to explore the gestural and musical potentialities in a space where they serve as mediators between the physical instrument and its virtual components. In digital design, affordances actualized by the end-user correspond to the inherent constraints integrated within the system. The establishment of these constraints transpires through parameter mapping, and the perceptual affordances of a given system manifest as the user scrutinizes and navigates its predetermined boundaries. Each design phase culminating in the current iteration is presented and discussed through an analytical framework centered on constraints and affordances. This framework serves to highlight the dynamic relationship existing between these two crucial components in any interactive digital music software, and the procedural nature by which these design relationships formulate for both the programmer and performer of the system.

1. INTRODUCTION

In response to the rapid expansion of technological progress during the nineteenth and mid twentieth centuries in the United States, the "contraption kit," underwent a process of standardization into the modular assemblage of mechanized devices and fabricated physical materials referred to as the modern drumset. At that exact point in the decades-long process of this standardization, the drumsets were not considered revolutionary in terms of their designs, nor did they challenge a drummer to change their physical relationship to the instrument. This process towards a uniform instrument design was a necessary step in being able to realize what the drummer already wanted to do during live performance. However, the path leading up to this point in standardization included countless individual efforts of remarkable ingenuity and mechanical invention, each of them contributing towards the evolution of the instrument’s construction, and by extension, its vocabulary.

The system presented in this paper, referred to as the *CV Drum Tracker,* is a cybernetic extension of this standardization, one which represents a singular effort towards reorienting the performer’s physical relation with a hybridized version of the instrument. It encourages a potential user to think in terms of how computer programmers would conceptualize and procedurally apply a systems-oriented design method, which then emboldens their performance to explore the gestural possibilities and musical potentialities within a space that situates them as the mediator between the physical instrument and its cybernetic components. Through the design of computational systems, the implementation of multimodal data inputs into the system, and the leveraging of computer-vision technologies into a sensor-controlled visual interface, this approach to design can be thought of as a virtual augmentation of the drumset. This technological mediation, and the iterative nature of programming based on process-oriented outcomes forms the basis for what will be referred here to as *Procedural Aesthetics*. It is through this concept that the *CV Drum Track Tracker* exists in its current iteration. The desired outcome of the system is not focused on completing one particular piece of music, but on establishing a computational approach towards crafting an entire environment or infrastructure through which the combining of multimedia formats and multi-modal performance techniques could be achieved. The following sections will explore the design considerations encountered while programming performance systems for camera-controlled, computer vision technologies and the virtually augmented drumset.

2. DESIGN CONSIDERATIONS

2.1. Designing for Transparency

The drums are a physically demanding instrument. Players must possess coordination between four limbs, and utilize these skills according to the temporal and dynamic fluctuations occurring in the music. Depending of the musical style and the performance model of the ensemble, the drum set player may be responsible for outlining the structural form of the overall piece while simultaneously executing the micro rhythmic phrases that comprise a macro-musical interaction with a soloist. Within a free jazz context, these physical movements can be thought of as “hypergestures” [1] as each individual musicians’ gestural activity can “invent their own creative musical trajectories” while simultaneously influencing the way in which other players in ensemble create trajectories of their own. The skill required to perform such a balancing act is premised on practice and experience of course, but can be accelerated through the means of acquiring tactile and kinetic muscle memory in combination with critical, audible feedback [2].

With that in consideration, knowing a physical gesture "feels good" (or physically familiar) is an important factor in having the confidence to perceive that a musical gesture "sounds right" within the context of a broader creative trajectory being created in real-time. If the drum surface is altered, or if the player is required to be tethered to some device that they are otherwise unaccustomed to wearing, this may alter the way the player moves during a performance. Moreover, placing nearly anything on the surface of the drum will change its natural resonance, timbre, and the tactile sensation of playing the instrument. Depending on the placement of these materials, the drummer may have to alter their physical approach to the instrument in meaningful ways [3]. This could dissuade drummers from experimenting with electro-acoustic musical practices altogether, as previous research has revealed that "improvisors who perform simultaneously on acoustic instruments and electronic devices are likely to bring different means to their use of electronic gestures" [4].

An indelible challenge in constructing an interactive system for the drum set is in designing it to be as transparent as possible during live performance. The system's design must function so as to not impede on the motions through which a drum set player executes a musical thought, nor on the sensory experience of playing the electro-acoustic instrument. In order to design an effective 2D screen-based musical interface, the programmer must apply their understanding of what perceptual psychologist J.J. Gibson refers to as his Theory of Affordances [5]:

"The *affordances* of the environment are what it *offers* the animal, what it *provides or furnishes*, either for good or ill. The verb to *afford* is found in the dictionary, but the noun affordance is not. I have made it up. I mean by it something that refers to both the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment."

This definition of affordances was made in the context of how mammals perceive the physical environment around them. Designer Don Norman coined the phrase "Perceptual Affordances" [6] to refer to the options unique to virtual, screen-based environments. To Norman, there is distinction to be made between the affordances provided by physical components that comprise the computer (or any real-world tool or environment) - the keyboard, the trackpad, built-in camera, to name a few - compared to the options provided to the user through the design of computer software. This distinction informs his approach to human-computer interaction, which he refers to as "human centered, natural-based design" [7]. This is an approach to designing environments where humans do not sense any type of resistance or struggle when interacting with a piece of virtual technology, and instead solely focus on achieving the desired end results of using any given tool. Often referred to as "natural design," Norman's design prioritizes the need to maximize the transparency involved in the interactions between humans and technology, to the point where humans ultimately forget that they are entering a technologically-mediated space at all [8]. Such design principles aim to enhance the transparency of use to the point where the experience feels unmediated through a particular technology at all.

As music technologist Thor Magnusson has stated, conceptual simplicity, and common use-functions are derived from an interaction model premised on transparency of use [9]. A well designed interface establishes the framework of an interaction mode that enables the user to recognize the gestural options and utility afforded to them. The affordances programmed into virtual devices are more imperceptible than those designed into physical tools [10], be it a saw, a steering wheel, a violin, or a snare drum. These virtual, screen-based interfaces also disembody the musicians from the tactile familiarity musicians acquire through years of development. In the case of the *CV Drum Tracker*, a clearly defined interaction model between subject (human) and object (virtual, screen-based software), is an absolute necessity.

2.2. Designing for Transparency

Ensuring that a piece of technology does not cause a severe disruption to naturally playing the instrument is a concern which exists at the forefront of designing the *CV Drum Tracker.* However, completely adhering to the principles of Natural Design can parameterize the creative potentialities of a particular technology, especially when it is being appropriated as the means for musical interaction. After all, enabling Facial Recognition features to log into a phone is a completely different relationship to Computer Vision technology compared to using it to gesturally manipulate incoming audio and visual signals in real-time.

In digital design, the counterpart to the affordances realized through the user are the constraints built into the system [10]. Constraints are established through parameter mapping, and the perceptual affordances of any system are realized through the user exploring the limits of its programmed constraints. Similarly, the physical and design constraints of any musical instrument (virtual or otherwise) define the way in which musicians can physically interact with its material dimensions. The expressive scope of this interactivity defines the constraints, which in turn, reveal the affordances in any system. This is to say that in any digitally-designed musical system, creative activity is assessed by how the users navigate its constraints while exploiting the affordances provided to them. Whether these features fall under Gibson's idea of "environmental" affordances or Norman's "perceptual" definition is determined by the constraints, not in spite of them.

Robert Rowe defines an Interactive Music System as a system "whose behavior changes in response to a musical or physical input" [11]. If a user does not change their behavior in response to the constraints and affordances in the system, then it is not truly interactive. Similar to human interaction within improvisatory musical environments, reciprocity must go both ways. Limiting the drummer through the principles of Natural Design alone would prevent the performer from realizing the full expressive, gestural, or musical potential that the system could afford them. This would limit the ways in which the system could react to user input, which would ultimately and severely parameterize the expressive scope of the instrument. Operating from such a stringent framework, the system designer would have to superimpose a layer of virtual technologies onto the drummer and the instrument in such a way that would negate the need for any interactivity during performance, thereby potentially eliminating the possibility of taking advantage of the efficiencies, constraints, and affordances provided by an interactive system.

Designing an interactive system that leverages buffer recording, machine listening and motion tracking technologies affords the performer access to musical material from a previous point in the performance. This temporally elastic, non-linear relation to their own performance affords the drummer to further process sounds by using expressive gestural motions that exist in contrast to the functional movement associated with playing an acoustic drumset vocabulary. Implementing this design consideration can expand the drummer's gestural motions from the instrument's acoustic timbre and material dimensions into a virtual space, permitting the performer to extricate themselves from learned rhythmic patterns, and to center their attention on deriving trajectories of motion intended for the explicit purposes of generating sustain on an instrument whose natural sonic profile mainly consists of short, densely layered and repeated impulses.

Simply stated, an approach to digital, screen-based instrument design based on transparency alone leads to a difficulty in perceiving affordances, while careless mapping leads to an inability to recognize constraints. From the user's perspective, ease of motion, flexibility, and clarity in design are of paramount importance as well. The inability to recognize constraints translates into a sloppily designed tool where its expressive scope is never fully realized. This makes style and creativity within a hybrid physical-virtual environment difficult to define, and challenging to evaluate.

3. MULTIMODAL DRUM EXTENSIONS.

3.1. Experiments in Creating a Virtual Drumming Environment with Motion Tracking

The *Virtual Drum Simulator* uses Computer Vision to build a "drum system that can be played using a webcam and a computer system alone," [12] by defining sensor zones and identifying the oft-changing contours of a drummer's hands while simulating a playing motion. The video feed acts as the interface for the user by highlighting every designed target with a colored frame, each one tethered to a distinct virtual drum or cymbal that would trigger if the user entered into its specified area. The OpenCV library was used to detect whether or not the contour of the hands inside the target rectangles was correct enough to trigger sound ("correct" being defined as whether the recognized position of the hands correspond to the motion of sticking that particular drum or cymbal in an acoustic setting). With such accuracy required to trigger each drum or cymbal, this software is intended to serve as a virtual replacement of an acoustic drumset, or an effective teaching or practicing tool.

Similar to the *Virtual Drum Simulator*, the *Air Drums* use Computer Vision to simulate the sensation of playing an acoustic drumset [13]. Using the same OpenCV library as the *Virtual Drum Simulator,* the *Air Drums* use color tracking for Object Detection/Tracking, Event Detection, and its subsequent Drum Synthesis. Makeshift sticks wrapped in colored paper, along with placing a color sticker on the user's left thigh substitute for the hands and bass drum, respectively. By using Blob Detection, *Air Drums* can use the largest blob to run a *By Points Comparison* and an *Acceleration Comparison* on a frame-by-frame basis, thereby triggering note onsets through a prediction model based on present stick position in comparison to the previous data acquired from the last two video frames.

In the tradition of incorporating a wearable sensor for the purposes (among many) of monitoring XYZ positional data - a concept similar to Max Matthews' Radio Baton [14], The *Airstick Drum* [15] integrates virtual percussion instruments alongside an acoustic drumset. Bluetooth sends data from the drummer's sticks to a computer which is then transferred into MIDI messages based on specific stick positions. Salient MIDI messages for note onsets, velocity, and duration are determined through attaching gyroscope accelerators to the sticks. In *Drum* [16], a Microsoft Kinect is used in conjunction with the sensing framework OpenNI to track drumming movements within specific sensing zones. Each sensing zone is mapped to a .WAV file that triggers whenever the amount detected activity within these regions exceeds a certain threshold.

3.2. Current Practices in Multi-Modal Augmented Drum Performance

Using the drums as an integral component in a multi-sensory, immersive experience is certainly not a novel idea, nor is the idea of using sensors to capture drumming gestures. Regarding the latter, this project distinguishes itself by the type of motion capture used and its subsequent artistic application.

From a design perspective, the conceptual approach most emulated in the *CV Drum Tacker* is a combination of the *Pragmatic Motion Capture System* [17] the *Digitally Active Drum* [18] and Pras's *Digital Musical Instrument* (DMI) for drummer Jim Black [4]. As designers Peter Williams and Daniel Overholt state, The *DAD* was constructed around the sentiment that the "natural resonances" of the head should be respected, that the drums should be designed for an allowance of "different techniques, nuanced control, and co-location of sound and instrument…digital augmentation should not obstruct choice of technique". Furthermore, the synthesis technique or degree of processing applied to the acoustic should not privilege any particular playing technique over another. The DAD assigned off-center locations on the snare drum to act as triggers for synthesis, where the audio amplitude tracking of each snare stroke is mapped to the amplitude envelope of the enacted subtractive synth sound. The snare drum head was separated into two primary regions: the Central and Accentric Regions. While the Central Region preserved the acoustic sound from the middle of snare drum, notes inside the Accentric Region triggered the aforementioned synthesis processed, among other user-specified, time-based audio effects.

Aptly named, The *Pragmatic Motion Capture System* [17] relies on a portable camera being angled to strictly capture only the drum strokes, drum surface and the neutral background of the wall. Simultaneously to this data acquisition, a dynamic microphone is used as a spot mic on the snare drum, quickly facilitating the designer to visualize drum gestures in two distinct mediums at once.

As varied as these methods are, nearly all of them attempt to capture the highly personalized and idiosyncratic nuances associated with drumming motions by closely tracking the variances in any combination of quantifiable attributes - timing, velocity, amplitude, and spectral centroid measurements - through modifying the drum's surface, body, or the drumstick [3]. Nearly all of the projects listed above simultaneously utilize multiple techniques and technologies to exact enough data in real-time to accurately account for the numerous physical and musical dimensions of a drummer's performance. Maintaining a sustained level of transparency in acquiring data is another factor contributing towards the frequent implementation of multimodal measurement systems, as they tend to be less intrusive to the performer [18].

When looking for similar projects to reference, the search included audio-visual experiences where the drums were to be conceptualized as a control source for sound-responsive data exchange. The artistic endeavors that are most aligned with the *CV Drum Tracker* can be found in Christos Michalakos's *Icarus* project [19]. *Icarus i*s a hybrid interactive game/performance system designed for Michalakos's *Augmented Drumset* [20], an electro-acoustic drum set built from electronic drum triggers and contact microphones. In *Icarus,* the drummer has to navigate five distinct game environments, all of which are visualized through was Michalakos describes as "light art," [19] which is an interactive light system that alters its color, tone and direction based on the data acquired through actively running the acoustic audio signal through machine listening techniques. Audio signal processing and real-time performance tracking leverage the drum set into a controller for the video game, effectively improvising an electro-acoustic soundtrack that will differ upon each iteration of the experience.

The similarities between Michalakos's work and the *CV Drum Tracker* are primarily rooted in how the drums are positioned within their respective multimodal technologies, and in conceptualizing the instrument as transducer for these interactive experiences. There are also parallels between structuring the different levels of *Icarus* as distinct musical modes that the performer can autonomously navigate through and the different performance modes included in the *CV Drum Tracker* (these are discussed below). Of all the conceptual congruencies between *Icarus* and the *CV Drum Tracker,* the most consequential of them is the structural freedom the performer experiences when interacting with each system. While there are no gameplay consequences to the *CV Drum Tracker*, there exists a temporal elasticity that fuses the hybrid performance-installation into a dynamic experience that is at once emergent and indeterminate upon each engagement with the system.

From the plethora of projects and research listed above, it is obvious that the mere combined use of computer vision, machine listening, and multimodality with the drumset is not some sort of singular distinction for the *CV Drum Tracker.* However, the *CV Drum Tracker* distinguishes itself from the aforementioned projects by using transparent technologies of surveillance to preserve the act of playing the drums in a completely improvisatory context while also affording the performer to free themselves of these technical rigors in an effort to explore more elongated or alternatively expressive gestures. Based on the Free Jazz Performance Model [21] of Computer-Human Interactivity, this design consideration centers the compositional process around the performer itself, as their gestures become "the technical tool of communication and creative flow" [1]. Thus, the system can be considered a performance-driven software program, as there is no anticipation or realization of any pre-programmed musical score [11]. Transformative response methods process the incoming audio signal from the acoustically augmented drumset by simultaneously measuring pixel-to-pixel differentiation in the gestural movement monitored in the incoming input matrix.

While the process of data acquisition and feature extraction is completely virtual, the experience of playing the drums remains intact. In contrast to the *Virtual Drum Simulator* [12] or *Air Drums* [13] projects, computer vision techniques are not used to replace the drumset with a virtual facsimile of itself, nor it is meant as a practice replacement for the acoustic set, or to function as an educational tool for entry level drummers. Rather, that same transparency the computer vision tools provide in these projects functions as the same means to acquire gestural data in the *CV Drum Tracker*, guaranteeing that the necessity of measuring the drummer's movement during a performance does not impede on the tactile sensation of playing the acoustic instrument.

4. DESIGNING CV DRUM SYSTEMS

4.1. Phase One

The current design of *CV Drum Tracker* is a result of an iterative process, one which can be separated into three phases. Phase One represented the first attempt at designing the screen-based interface. The incoming video image was separated into smaller sub matrices, which were referred to as Target Sensors. The user was able to move four of the seven zones present in the camera feed: the Upper Left, Upper Right, Hi-hat, and Ride Zones. The user was not able to change the rate or range of their movement, but the Moving Target Zones (MTZs) could always be reverted to their original position. Saturation effects were applied to each submatrix as a way to contrast with the grayscale video feedback of the entire camera feed. This made a clear distinction between the motion occurring inside and outside the specified target zones. The Machine Listening techniques yielded amplitude tracking data that controlled the amount of feedback that was applied to the background video feed.

Computer Vision techniques of Frame Differencing and Centroid Tracking were used inside the Target Sensors to control audio processing in the signal chain. These mapping techniques were programmed into the system, unable to be changed by the performer.

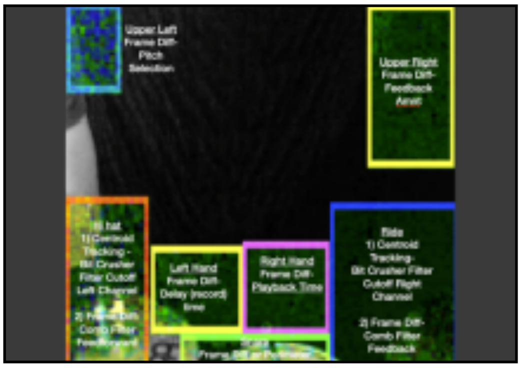

Image 1. Video Screen Interface, Phase One[1]

The video feed is captured through a Logitech C920 Webcam. Using a camera detached from the computer enabled the performer to position the camera where it could best capture their own body position without manipulating the angle of the screen. All audio and visual processing software was built in Cycling 74's Max/Msp~/Jitter, using Jean-Marc Pelletier's cv.jit external package [22]

5.1.1. Parameter Mapping

Parameter mapping relationships could be established between the position of each MTZs and the gestures occurring inside of them. These relationships were referred to as Moving Target Mappings (MTM). MTM controlled the spatial and amplitude balance between dry and processed sounds. This is done by randomizing the position of the Upper Right or Upper Left Target Zones. Once this process is initiated, any movement of the Upper Right target zone along the horizontal axis controls the panning of four front speakers. While the horizontal movement controls the panning of dry and processed sounds, the vertical movement (when automated) manages the amplitude (level) balance between the individual effects, which include the Comb Filter, Bit Crusher, and Multi-Tap Delays. This is done by dividing the overall horizontal and vertical movements of the Upper Right (relative to pixel location) by half so that the overall video dimensions can be split into four individual quadrants. Each of these quadrants are assigned an audio effect. Once the Upper Right Target Zone moves into a quadrant, its gain level increases, just as it will decrease to zero as the zone transitions into another quadrant.

5.1.2. Evaluation

While these mapping constraints yielded some interesting sonic results, the affordances provided by these mapped constraints were a complete mystery to the user, which made them seem even more opaque to an observer. As Image 2 indicates, so little instruction was provided. There is no initial information provided on the user interface that indicates what these submatrices do, or how the user is supposed to interact with them. As a result, instead of interacting with the software as a musical counterpart the user has to develop these connections in real-time. While making connections between the gestures and sonic outcomes was supposed to be somewhat of an emergent process, the design simply proved to add to the difficulty of discerning the mapped relationships. The system afforded the performer very little in terms of exercising their own creativity within the very stringent constraints imposed upon them.

The MTZs were intended to extend the gestural possibilities at a drummer's disposal. However, since the gestural movement inside the submatrices was responsible for processing audio and not the movement of the matrices themselves, the user was forced to perform gestures that were completely separate from any physical motion required to play the acoustic drumset. This was especially true if the drummer wanted to augment the instrument's sound in real-time. This option to augment gestures was intended to be optional but by no means compulsory. There were far too many submatrices than were necessary, and the positioning of the sub-matrices made it difficult to expand gestural vocabulary beyond those motions inherent to playing the acoustic drums without a complete disruption of these drum-centric movements altogether. The drummer had to choose between one set of gestures or another. Furthermore, the performer was not able to designate any mappings between their own Moving Target Zones, gestures, and sonic outcomes. This proved to be too much of a constraint on a system which was capable of producing more outcomes than one specific mapping could possibly anticipate. In addition to a more flexible mapping strategy, more modularized processing software was needed to maximize the sonic potential of the system.

In summary, an ambiguous interface design lacking in any data or text-based information provided an interesting visual component, but it did not translate to the user having any sort of clarity in knowing how to interact with the software. The movement of the matrices produced too much of a game-like environment, where the user felt like they were competing against the software instead of mediating through it. The time it took to discover the few constraints and affordances of the system was unnecessarily long, and perhaps impossible to do if the user was not also its programmer and designer. Perhaps most crucially, the technological limitations of using only Frame Differencing and Centroid Tracking as the means for audio processing produced results that were deemed to be sonically unsatisfactory. It became increasingly apparent that the amount of Computer Vision-based techniques in

---

1

the system needed to be expanded while the interface itself needed to be clarified.

4.2. Phase Two

Phase Two has proven to be most crucial in determining the future development and design of the software, and focused on providing clarity to the user. Every submatrix was eliminated but one. No longer did multiple submatrices move across the screen, which eliminated the Moving Target Mappings from Phase One (along with the potential confusion that stemmed from their presence). The remaining single submatrix was still able to move around a section of the larger matrix.

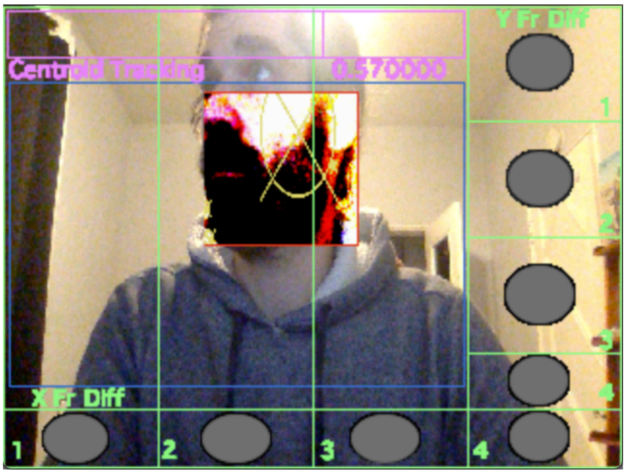

Image 2. Video Screen User Interface, Phase Two

The number of Computer Vision technologies at the user's disposal was expanded. These additions included Blob-Centroid Tracking and Optical Tracking. Rather than using Frame Differencing inside seven different submatrices, the performer could rely on other tracking techniques that provided more useful data for processing audio. Frame Differencing is effective for measuring extreme changes in motion on a frame-to-frame basis, but is not conducive for providing gradually changing data. In order to enact audio processing with Frame Differencing, either a very demonstrative gesture, long periods of complete inactivity, or increasing the distance between the performer and the camera feed, were required. Other techniques, such as measuring for concentrated areas of light inside the submatrix and tracking how these areas change over time, were more effective for generating data for processing audio over long periods of time. Switching to Blob-Centroid Tracking and Optical Tracking facilitated these subtleties in data acquisition inside the single submatrix, resulting in more fluid experimentation with gesture.

Frame Differencing was still used, although in a completely different manner than in Phase One. The video screen interface was separated into 16 sections (4 rows and 4 columns), all of equal size. Frame Differencing operations could be performed on each section. The incoming measurements were continuously evaluated against a set threshold, and virtual buttons were used on the video screen interface to indicate when a gesture exceeded this value. If the incoming motion exceeded the threshold, then the activity would trigger the process or event that was mapped to that particular region of the screen. For instance, Frame Differencing values on the rightmost column of the video screen would simultaneously trigger the playback of an audio file and switch between the primary camera view and feedback effect. The two leftmost, bottom regions triggered virtual synths.

The increase in computer vision technologies afforded the performer to use the software to both generate and process sounds. Virtual synths could be triggered by Frame Differencing, routed to an audio device, and processed with a mapped parameter coming from one of the video modules. The system was designed to afford the user to route any available video data point to a specified parameter in an audio processing device. Instead of having to discover these relationships in real-time during a performance, the user had to construct many of the constraints that they were to encounter. It became the standard practice to design this mapping procedure for each of the system's sound processing parameters.

5.1.2. Evaluation

The user interface on the video screen provided some salient measurements on the screen for the performer to reference. The options at the users disposal were far greater in Phase Two than in Phase One. While there were some data points provided on the video screen. The software required far more monitoring than before. This meant that the user had to perform with the interface on the video screen and simultaneously monitor the system's software in real-time. Experimenting with gestures became easier, but operating the software became increasingly difficult. Rather than freely playing the drums and interacting with the video screen, I often found myself thinking like a programmer or manager of the system during performance. Moreover, the buttons were placed in locations on the screen that made it nearly impossible to not trigger the processes associated with them. I could not simply put my hand into the submatrix to use these techniques of Optical Tracking and Blob Tracking without cueing a synth or triggering a visual effect. This made live performance quite difficult. It was decided that the complexities did not to be visible to the performer. Observing more than one screen during performance was superfluous and distracting. All of the information needed to monitor the data produced by the video-human interactions needed to be on the standalone video screen itself.

Additionally, the system had no way to organize time on a structural level. There was no way to transition to different presets or reconfigure the signal chain without manually clicking on the screen. Clicking on the screen proved to be incredibly difficult to do during performance, especially when the user was monitoring video data points from the software interfaces and their own motion in the video-screen interface. From a design perspective, this needed to be rectified. This design was also limiting the sonic capabilities of the system. At this point in development, the *CV Drum Tracker* represented a potentially interesting tool for generating and manipulating sound, but could not be considered an automated system for organizing improvisation any further than to interact with it through one preset. While the tool provided the user the affordance of generating and manipulating sound, the constraints built into the system did not make it a particularly effective tool for organizing time. Timed events were needed to both transition to a next signal chain of audio software, and to change the chosen presets for those modular devices. The user would have to compose the structure by thinking of each signal chain as a section of the piece and map the video data points to audio processing parameters for each device prior to the performance. Once this macro form was established, they could then improvise their way through each section by interacting with the video screen alone, without the need to organize time or create parameter mappings in real-time.

5.3. Phase Three

Phase Three represents the most technologically and conceptually sophisticated design of the system to date. More Computer Vision techniques have been implemented into the design, yet the complexity involved in monitoring the software during performance has been greatly reduced. The techniques of Blob Bounds Tracking, Blob Rotation Tracking, and Face-Tracking have been added into the system. This data provides far more control to the user and more options for parameter mapping. These measurement provide the size, movement, and rotation of each blob tracked in the submatrix. From a technological perspective these are the most consequential additions to the system to date, as the movement of this tracking data best resembles the way a performer would move a dial or slider on an analog interface.

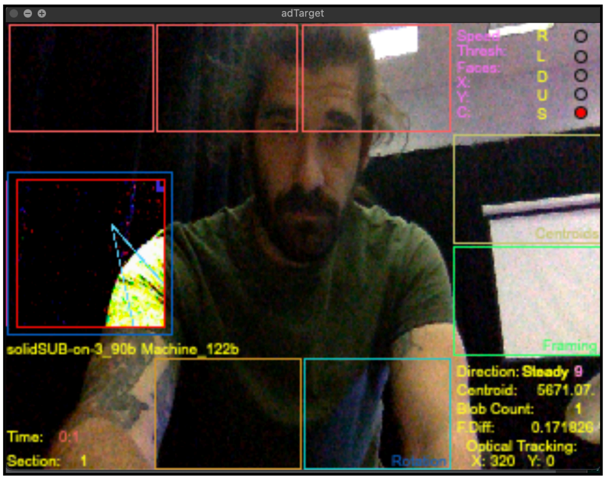

Image 3. Video Screen User Interface, Phase Three

Virtual buttons that were impossible to avoid triggering in the previous phase have been eliminated. Those spaces on the bottom left and right side provide a point of entry into the screen without enacting any musical or visual outcome. The submatrix no longer moves at all. This was a feature that was rarely used, so it seemed somewhat superfluous to keep including it. Eliminating the movement of the submatrix meant that there was more space to include salient data points that needed to be monitored. Instead of having to reference an expansive, complicated piece of software during performance, the user is provided the information for nearly all the CV techniques included in the design. The user can also now also monitor the direction of their motion and how long they have held steady in the submatrix. They are also able to switch between different tracking modes (which CV techniques can be monitored) by using the frame differencing techniques on the right of the screen. These buttons (which represent sections of the screen) were previously mapped to trigger visual effects and spoken word files.

Three more visual effects were added to the system: a texture generator, a color streaker, and a video glitcher. The visual effects were still triggered by the same technique, but the buttons and regions that cue these processes are located at the top of the interface. These buttons are placed in the topmost section of the screen, which virtually eliminates the possibility of the visual effects being inadvertently triggered. Situating these buttons at the bottom of the screen would make the user have to cross them to access any other part of the interface. This particular hindrance was discovered in performance during Phase Two and had to be corrected as soon as possible.

The updated design of the video interface meant that the need to click or touch the software at all during the performance had been nearly eliminated. In Phase Two, the user had to monitor up to 4 video images at once. Providing so much information on the video screen interface meant that there was no need to monitor a huge piece of software while performing, which led to an overall improvement and simplicity in the appearance of the user interface. This simplicity in design proved invaluable during performance. There was far less technology to be cognizant of while trying to play the drums, allowing the performer to simply focus on using the software as they desired during improvisation. No longer did the performer need to simultaneously assume the roles of "performing musician" and "system technician."

Turning the system on was now an automated process. Every component of the screen interface was automatically rendered without user input. All the performer would have to do was simply instantiate the software and the program ran by itself. Other new features in Phase Three included the ability to use a video as input source instead of the built-in video camera feed, and having the ability to choose between three main output modes for the matrix feed: the primary camera feed (or movie file), the Frame Difference Mode, and the Effects Mode. The Effects Mode had four different settings unto itself, giving the user up to 7 different visual outputs to choose from. The submatrix feed could be switched between 5 different outputs as well. Each video processing module has its own user interface embedded in the system. In order to access the individual CV modules, the user simply clicks (or assigns a MIDI controller to open them) on each individual button. However, this information does not necessarily have to be monitored by the performer, especially during improvisation.

In Phase Three, the system interface has been re-organized into two distinct sections. Any controls dealing with video input or data acquisition are located at the top of the interface. Any user-input that organizes audio processing is located at the button half. The most substantial addition to the software in Phase Three was the automated score control and audio signal routing system (and the addition of four more audio processors). This feature afforded the user to create audio signal routings that would change based on an automated timer. Each "section" of the performance can be thought of as individual signal routing. This timer feature manages the transition from one section to another without any user input during the performance. However, this automation is by no means compulsory, as the user does have the option to pause the timing of events, as well as skip to any of the pre-programmed sections (signal routings) during performance. Each audio processing module can be individually opened in the same manner in which each of the video modules can be accessed.

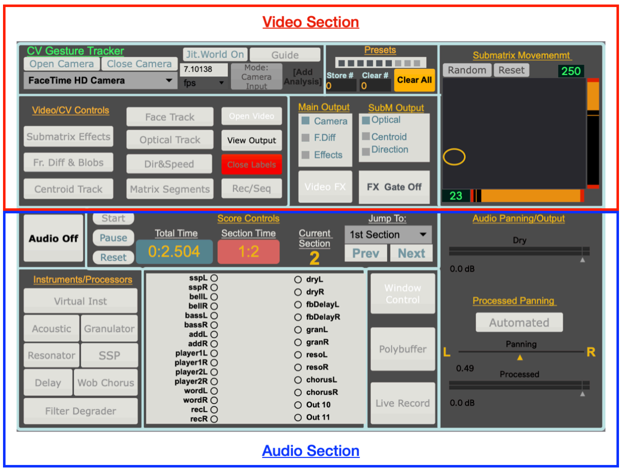

Image 4. Software User Interface, Phase Three

5.1.2. Evaluation

It is at this point in the development and creative process that the system represents a technology-mediated strategy for what composer and professor Sandeep Bhagwati refers to as "comprovisation" [23]. He describes his neologism as "An approach to creation in time-based arts predicated on an aesthetically relevant interlocking of context-independent and contingent performance elements. Comprovisation often uses unique constellations of oral, written, animated and interactive scores that can accommodate the scoring paradigms of many traditions and practices."

Musical form is decided by the transitions between user-defined signal routings, but the interaction model between the human and computer within these sections is based on the performer discovering these emergent gestural-to-sound (and visual) relationships through improvisation. Is it at Phase Three in the design process that the *CV Drum Tracker* represents an integrative approach to assimilating both its formal elements and indeterminate aspects into its technological infrastructure. The constellation of material that determines its multimodal output is embedded within the virtual machine itself. Far too often in electro-acoustic practices, the technology involved in the composition and improvisation is assigned a unidimensional, auxiliary, or tangential role in the creative process. The intention behind *CV Drum Tracker* is to conceptualize a musical process, evaluate its outcomes, and ultimately, define a personalized style through a holistic mediation with technology.

The design method described over the course of these three phases represents a procedural, systems-based, integrative approach to composition, where the comproviser is at once the programmer, designer, performer, listener, observer, and mediator of this hybrid environment. Such an orientation requires iterative prototyping, and the need to re-conceptualize the creative environment and the resulting aesthetic outcomes based upon the instant auditory and visual feedback yielded from the processes involved in this particular mediation with virtual and sensor-based technologies. Therefore, the term *Procedural Aesthetics* refers to both the iterative nature of how design choices are made and the dynamic, unfinished programming environment from which this technological mediation takes place. Time is organized and sound is generated and manipulated within a system made from an open programming environment (Max/Msp~/Jitter), not a Digital Audio Workstation. To hear the outcome of their design choices, the user does not have to play back a virtual tape machine, but has to rely on recursive operations that organize the playback of musical ideas. Observing musical and visual processes happening right at the point of programming changes the way the comprovisor evaluates and responds to the last decision that was made.The way time becomes organized and how sound is generated feels systematic in preparation, yet dynamic in performance.

5. CONCLUSIONS

Rowe distinguishes two types of computer-mediated interactive music systems based upon the degree to which the computer's output affects the actions of the human performer. The two classifications can be broadly determined by whether the objective is to either create a virtual musical instrument which acoustically augments a real-time performance by analyzing human gesture, or if the goal is to construct a cybernated improviser with their own performance behaviors that exist nearly independent from the user input [11]. The current design of the *CV Drum Tracker* clearly places it in the former category. Future development phases will be dedicated to leveraging more machine learning technologies so that the system interaction with the human improviser will exhibit the behavior of a semi-autonomous, cybernetic performer. This will be done by utilizing more computer vision techniques, such as Depth Data and Gesture Recognition, to construct histograms that analyze and subsequently respond to human activity.

Perhaps most important for the future design of the *CV Drum Tracker* is to distribute the technology into the hands of other drummers. Ultimately, the long term goal for the *CV Drum Tracker* is to construct a semi-autonomous, interactive environment that does not preclude the use of any current drumming vocabularies, but instead inspires a diverse range of drummers to explore the outer limits of their potential within this emergent physical and virtual assemblage.

6. REFERENCES

1. Mazzola, G., & Cherlin, P. B. (2008). Flow, gesture, and spaces in free jazz: Towards a theory of collaboration. Springer Science & Business Media.

2. Dolhansky, B., McPherson, A., & Kim, Y. E. (2011, July). Designing an expressive virtual percussion instrument. In *Proceedings of the 8th Sound and Music Computing Conference, Padova, Italy* (Vol. 69).

3. Tindale, A. R., Kapur, A., Tzanetakis, G., Driessen, P., & Schloss, A. (2005, May). A comparison of sensor strategies for capturing percussive gestures. In *Proceedings of the 2005 conference on New interfaces for musical expression* (pp. 200-203).

4. Pras, A., Rodrigues, M. G., Grupp, V., & Wanderley, M. M. (2021). Connecting Free Improvisation Performance and Drumming Gestures Through Digital Wearables. *Frontiers in Psychology*, 1001.

5. Gibson, J. J. (1977). The theory of affordances. *Hilldale, USA*, *1*(2), 67-82.

6. Norman, D. A. (1999). Affordance, conventions, and design. *interactions*, *6*(3), 38-43.

7. Norman, D. (2013). The Design of Everyday Things: Revised and Expanded Edition. Basic books.

8. D'Errico, M. (2022). Push: Software Design and the Cultural Politics of Music Production. Oxford University Press.

9. Magnusson, T. (2006, October). Affordances and constraints in screen-based musical instruments. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles* (pp. 441-444).

10. Magnusson, T. (2010). Designing constraints: Composing and performing with digital musical systems. *Computer Music Journal*, *34*(4), 62-73.

11. Rowe, R. (1992). Interactive music systems: machine listening and composing. MIT press.

12. Bering, S. R. F., & Famador, S. M. W. (2016, August). Virtual Drum Simulator Using Computer Vision. In *The 4th IIAE International Conference on Intelligent Systems and Image Processing* (Vol. 2016).

13. Tolentino, C. T., Uy, A., & Naval, P. (2019, August). Air Drums: Playing Drums Using Computer Vision. In *2019 International Symposium on Multimedia and Communication Technology (ISMAC)* (pp. 1-6). IEEE.

14. Boulanger, R. (1997, September). The 1997 Mathews radio-baton and improvisation modes. In *ICMC*.

15. H. Kanke, Y. Takegawa, T. Terada, and M. Tsukamoto. Airstic Drum: a Drumstick for Integration of Real and Virtual Drums. In *Proc. The International Conference on Advance in Computer Entertainment Technology (ACE2012), pp. 57–69*, 2012.

16. Sarang, P., More, A., Gaikwad, A., & Varma, T. (2015). Air drum using Kinect and Arduino. *International Journal of Computer Science and Information Technologies*, *6*(2), 1153-1155.

17. Van Rooyen, R., & Tzanetakis, G. (2015, May). Pragmatic drum motion capture system. In *NIME* (pp. 339-342).

18. Williams, P., & Overholt, D. (2021). Design and evaluation of a digitally active drum. *Personal and Ubiquitous Computing*, *25*, 783-795.

19. Michalakos, C. (2020, July). Icarus: a game/ performance for the augmented drum-kit. In *8th Conference on Computation, Communication, Aesthetics & X* (pp. 393-395). Universidade do Porto.

20. Michalakos, C. (2012). The augmented drum kit: an intuitive approach to live electronic percussion performance. In *ICMC 2012: International Computer Music Conference* (pp. 257-260). Michigan Publishing.

21. Winkler, T. (2001). Composing interactive music: techniques and ideas using Max. MIT press.

22. Pelletier, J. M. (n.d.) *Cv.jit: Computer Vision for jitter*. Jean-Marc Pelletier. Retrieved March 10, 2023, from https://jmpelletier.com/cvjit/

23. Bhagwati, S. (2018). Glossaire raisonné. *Circuit*, *28*(1), 15–22.