Steven Lewis

CV Drum Tracker: A Introductory User Manual

The CV Drum Tracker has four modes of use:

1) Drum (Performance) Mode - Three mic inputs from the drums serve as input to the record~ object

2) Motion Mode - sound files serve as inputs as opposed to live mic input. Sound files are chosen from a [polybuffer~] and manipulated with motion data generated by target sensor activity inside specified zones.

3) Synth Mode - a combination of audio analysis and camera tracking control the performance of a Soft Synth

4) Hybrid Mode - using the drums to initially record a performance in Drum Mode, then using that recording as a sound file in something that resembles Motion Mode.

Tab Controls:

Understanding the key commands for the tab controls is essential for minimizing any functional movement required for a performance. These controls enable the performer to change interfaces with a number of simple two stroke combinations.

To change interface tabs, press Shift + the following keys:

Shift + C = CV_DrumTracker (root)

Shift + G = grabber

Shift + T = targetSensors

Shift + A = audioSystem

Shift + M = motionMode

Shift + S = synthMode

It is highly recommended to learn and use these key commands (shortcuts), not only in performance but in practice as well. This tab system is one way to compartmentalize the many dimensions to the patch's interface, to only view the info needed to see at any given time. It is an attempt at minimizing the amount of time searching for parameters during a performance. However, achieving this economy of time and motion is predicated on spending a considerable amount of time working with the interfaces and knowing the aforementioned key-commands.

When the shift key is pressed, a toggle will be triggered on the CV_DrumTracker interface[1].

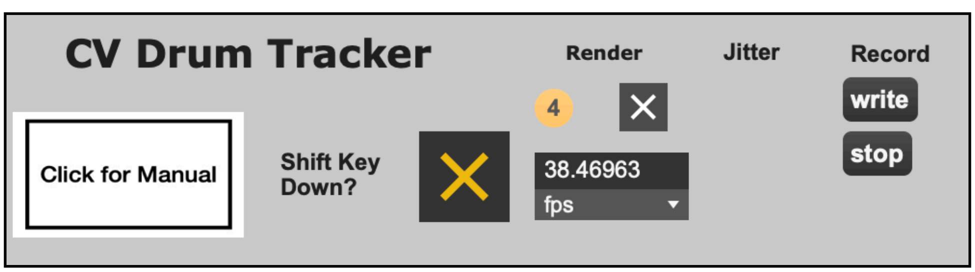

Figure 1: Is the shift key down (yes, it is…)?

[1] The key commands will always switch tabs, but this feature will only work if jitter rendering is on.

Start Instructions

Instruction steps correspond to the numbers included in the patch. Be mindful that these numbers run across multiple tabs. The steps below do not privilege any distinct mode, but are designed to turn on the entire patch. If followed, the user will have the full range of modes at their disposal, controls by the master control sends (to be demonstrated).

1. Turn on ADC, or press the "start" message. Set audio inputs and output.
2. Check signal levels for all three individual audio channels.
3. Adjust the threshold of *Audio Analysis* according to the incoming signal amplitude.
4. Turn on Jitter Render.

— Go to *grabber* tab — (Shift + G)

5. Press "getvdevlist" message. Select the desired camera input.
6. Press "open" to turn on the camera.

The camera input will look something like these two examples[2]:

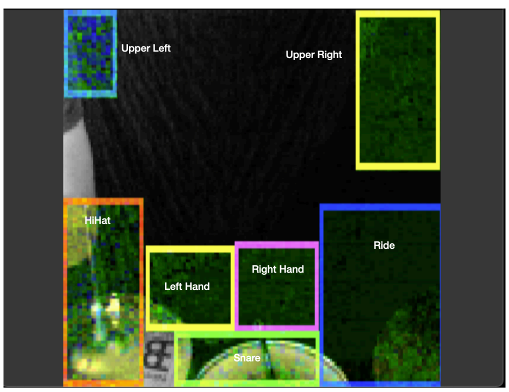

Figure 2: Camera Configuration, Drum Mode

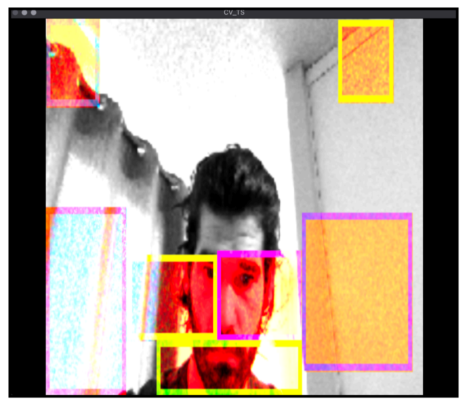

Figure 3: Camera Configuration, Different Camera Input

— Go to *targetSensors* tab — (Shift + T)

7. Adjust thresholds for Frame Differencing, Centroid/Perimeter Tracking. Once again, this will vary greatly based on each individual performance setting, and it will have to be done for every zone. Thresholds have already been preset if thew user doesn't wish to burden themselves with these details, but these thresholds will have a profound impact on the sensitivity of the system's responsiveness to motion.

8. Determine speed amount of the cpu timer applied to jit.gl.bfg.

9. Determine feedback amount for jit.gl.bfg.

---

[2] It is at this point where the user will most likely have to take some time to adjust their camera configuration and drum/instrument set up relative to the preset target sensing, or adjust the matrix position of the each target sensor to align with their natural playing position (to do this, go to the *targetSensors* tab). Many factors will influence these decisions, so it is best to merely suggest a number of best practices that were discovered while writing this manual, and leave the rest up to the individual.

Audio should be turned on, signal should be coming through all 3 audio (drum) inputs, and all nine target sensors should be capturing video.

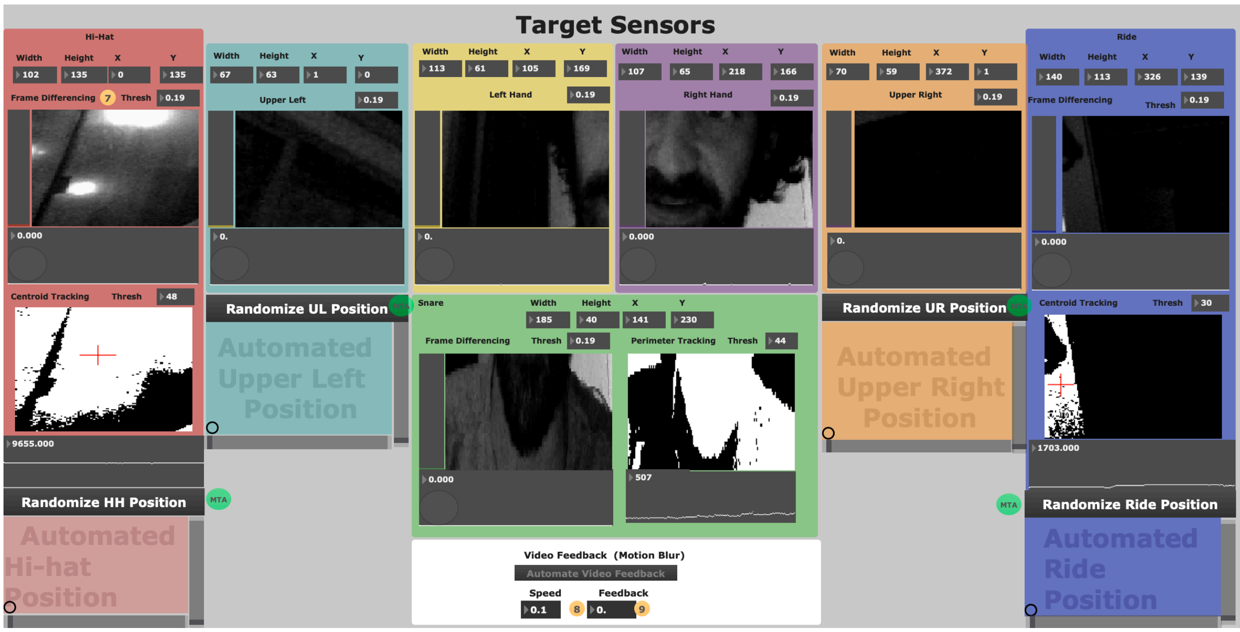

Figure 4: Target Sensor Set Up. Note the different thresholds for each of the two camera inputs using centroid tracking This is, as always, is due to lighting differences, spatial position in the room relative to lighting, and a host of other factors.

— Go to motionMode tab — (Shift + M)

10. Drop sound file folder into the yellow rectangle (this will take some time, depending on the size of your folder). Select which file to play beneath the yellow rectangle.

11. Decide how to cue new files to play: the space bar, Upper Left Frame Differencing, or Random Select on Loop).

12. Decide whether to engage in Motion Mode by selecting the “Turn On” and “Start Over” buttons.[3]

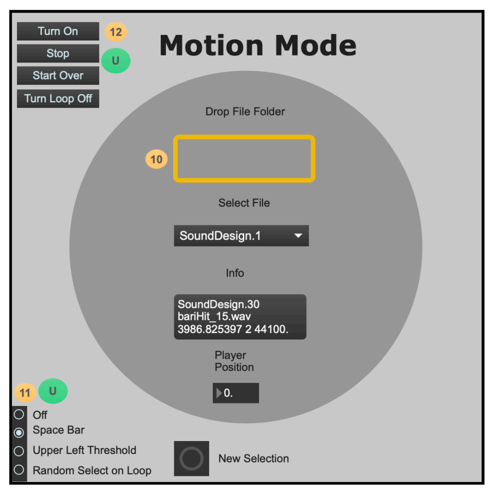

Figure 5. Motion Mode

[3] The dry send in the *audioSystem tab* will still have be given a non-zero number to hear the audio from Motion Mode

— Go to *synthMode* tab — (Shift + S)

13. Set Frame Differencing Threshold for Pitch and Velocity Selection.

14. Set the key, mode, or random size for the pitch range.

15. Decide whether to automate SubOsc, Speed, and ModWave amounts.

16. Send signal to DAC (when/if desirable).

— Go to audioSystem tab — (Shift + A)

17. Make sure the desired Recording Inlet is selected[4].

Figure 6. Synth Mode

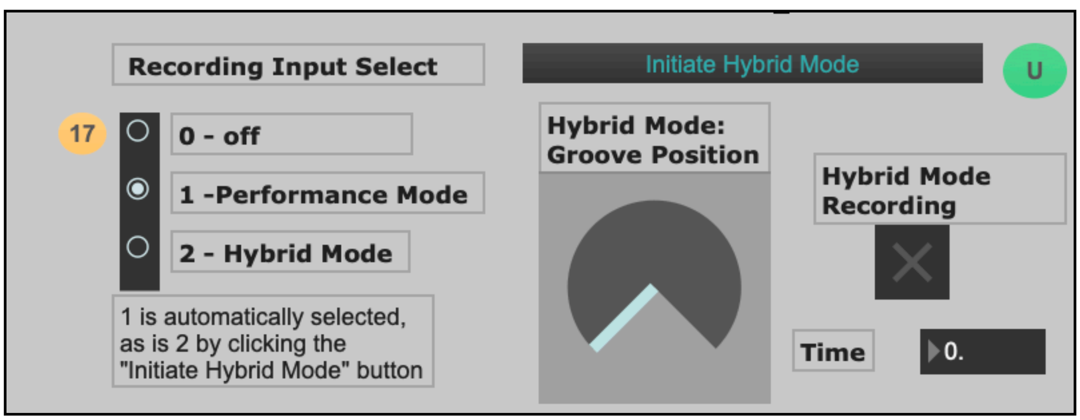

Figure 7. Initiating Hybrid Mode

Hybrid Mode never has to be manually selected here. Pressing the "Initiate Hybrid Mode" Button will automatically switch the audio outlets, as well as initiate all the operations necessary to transition into this mode. The recording in Hybrid Mode will only last five seconds (the length of the buffer being recorded into), then all live inlets to the [record~] are closed (0 is automatically selected). The five seconds recording will now continuously (on loop) record into the same buffer used as the input for feedback and other audio processing operations.

18. Choose to/not to automate Comb Filter feedback and feedforward.

---

[4] Drum Mode is automatically selected when the software is opened.

19. Choose to/not to automate Bit Crusher filter cutoffs.

20. Choose to/not to unlink playback buffer~ time from delay time.

21. Choose to/not automate feedback into recording signal.

22. Choose between frame differencing or perimeter sensing for the snare drum target zone.

23. The gain dials applied to each signal control the total output.

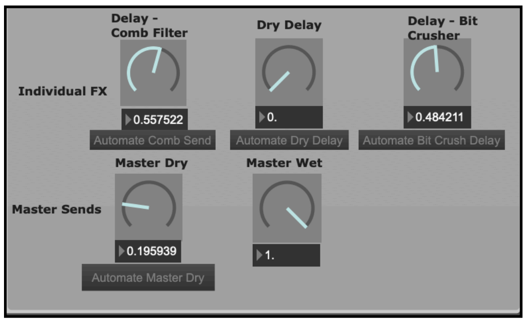

Figure 8. Main Gain Controls

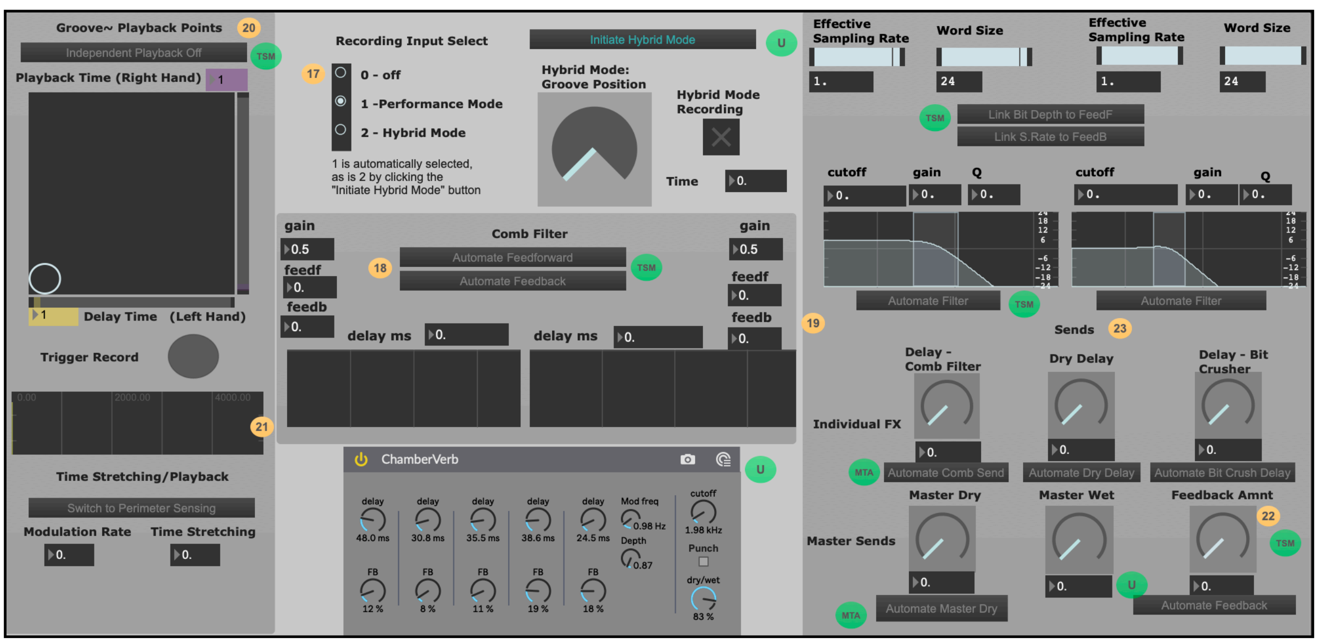

Figure 9. Audio Processing

Advanced Video Tracking

The user is able to move four of the six zones present in the camera feed: the Upper Left, Upper Right, Hi-hat, and Ride. The rate of movement and position range are not up to the user at this time, but zones can always be reverted to their original position.

Figure 10. Randomizing Target Sensors. Select the button at the top of the [pictslidr] to randomize target position.

The moving target zones are given saturation effects to contrast with the grayscale video feedback of the entire camera feed, which results in a hybrid interface-video art piece that obscures any motion or physical activity occurring outside the specified target zones. This is an attempt to fuse both the technical and artistic aspects of this project. Machine listening techniques yield amplitude tracking data, which is mapped to the double feedback effect being applied to the background video feed. The discreet sections upon which motion data is being measured conceptually translates to the *Oligopticon* in the sense that there is only brief moments in or time (or vantage points) where the target resolution gives full clarity to the underlying image The user, nor the potential audience observes for full resolution of the original source environment.

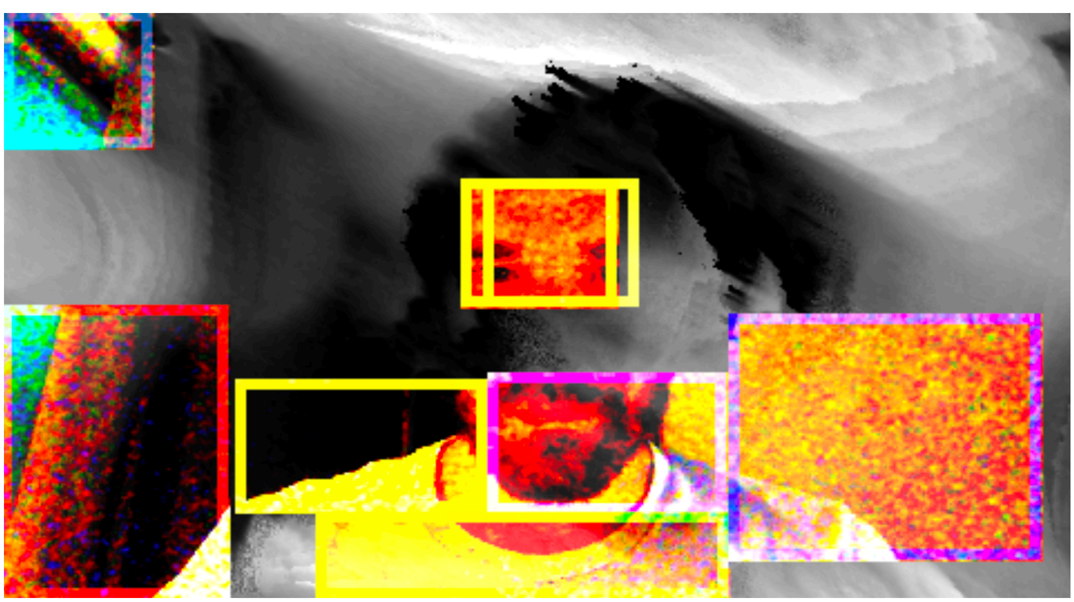

Figure 11. Moving Targets: A Roaming Target Sensor (Upper Right)

A Survey of Non-Automated and Automated Parameter Mappings

<u>Mapping: Non-Automated User Controls</u>

Figure 12. Designation of Non-Automated user Controls on Patch

These controls have to be specified by the user at all times. They are designed with a green button labeled “U”. To be clear, every parameter that can be automated can also be controlled manually, but this probably not the preferable workflow, since the musical output of the system is motion-controlled.

Automated Mappings

There are three distinct, yet interdependent mapping strategies:

1. Audio - Reactive Mappings
2. Target Sensor Mappings
3. Moving Target Mappings

1. Audio - Reactive Mappings

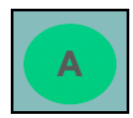

Figure 13. Designation of Audio-Reactive Mappings on Patch

The audio amplitude and spectral centroid from Drum, Hybrid, and Motion Modes are all continuously tracked and compared against a threshold, which when exceeded, triggers the following mechanisms to trigger:

- Re-recording of audio into the buffer (this is continuous).
- Playing and looping of the buffer.
- Randomization of specified, queried parameters of the Bassline Synth.
- Changes the Scale, Zoom, and the Noise Function for the jit.bfg objects used in the grayscale camera background.
- Scales the “double feedback” amount applied to the grayscale camera background

The running amplitude tracker is also mapped (and scaled) as the incoming MIDI velocity value for the next synth cue, which can then vary 15% up or down relative to the incoming root.[5]
These mapping are critical to understand while performing with the system. So many operations in the visual and audio system are contingent on dynamic levels (or rather, amplitude readings). As this is more clearly understood, certain drum stokes sounds can become audio visual cues. Essentially through the judicial use of dynamics and selective orchestration, the drummer can not only control much of the visual output of the system, but can also establish audio-visual relationships to which an observer can correlate as a larger compositional framework. One of the most effective ways of accomplishing this is to use the amplitude of the bass as a trigger for all of the processes mentioned above. The bass drum (which is actually a modified floor tom that has been converted into a makeshift bass dum) is the loudest of the

---

[5] A technique first learned in Todd Winkler’s *Composing Interactive Music* .

drums, and is the only area of the drum set not being tracked by any camera.[6] This combination makes it the ideal candidate to link amplitude with triggering a host of musical and visual changes within the performance. This is not to say that other instruments on drum set cannot exceed any set amplitude threshold, but every other drum and cymbal is in some way being monitored by a Target Sensor. If the goal is to reset of the processes listed above without involving the effects mapped to any targets through motion tracking, then performing ass if they are linked to bass drum note onsets is a prudent strategy.

2. Target Sensor Mappings

Target Sensor Mapping Automations are labeled with the “TSM” marking in the patch.

Figure 13. Designation of Target Sensor Mappings on Patch

These are the most consequential mappings into term of audio processing. Camera Vision techniques of Frame Differencing, Centroid Tracking, and Perimeter Sensing are all used to control audio processing in the “wet” signal chain.

The different options for each channel can either be substituted as the input for the default Frame Differencing. In some cases, this decision has already been made for the user. For example, the alternative measuring options for the Hihat and Ride - Centroid Tracking - is already mapped filters. Both measuring techniques are already assigned, so it may be best to reference this guide when performing or re-mapping any of the incoming data.

	Frame Diff	Centroid Tracking	Perimeter Sense
Upper Left	Pitch Select (Synth Mode)	-	-
Upper Right	Feedback Amount	-	-
Hi-hat	Feedfoward (Comb Filter)	Bit Crush Cutoff Filter L	-
Ride	Feedback (Comb Filter)	Bit Crush Cutoff Filter R	-
Left Hand	Delay Time (Record Time)	-	-
Right Hand	Playback Time	-	-
Snare	Time Stretching	-	Time Stretching

Figure 14: Camera Measurement Techniques, Organized By Zone

[6] Plans are currently being designed that will integrate camera tracking on the bass drum into future versions of the Cv Drum Tracker

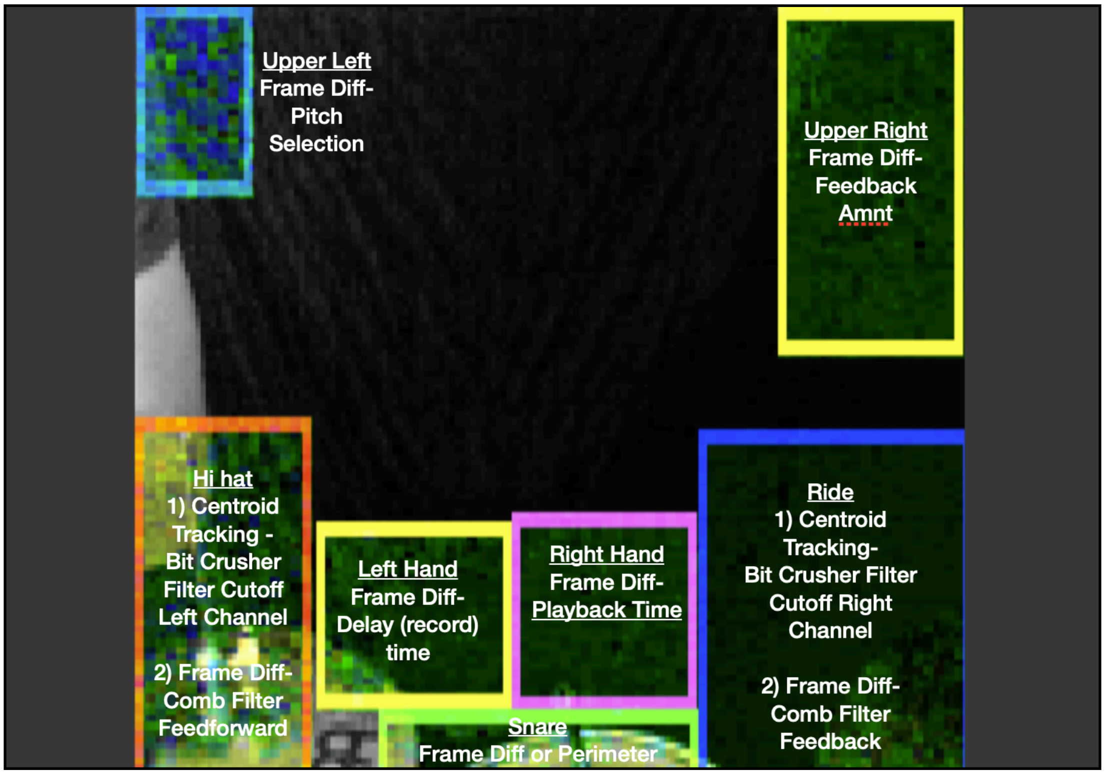

Figure 15: Camera Configuration, Mapping Measuring Techniques to Audio Processing Parameters

The following signal flow covers audio processing that comes as a result of Motion Target Sensor tracking. This is provided in the hope that it may elimination any residual confusion or inherent lack of clarity in how these distinct technologies of camera vision and digital audio form a cohesive, initial prototype for what will eventually materialize in a motion-responsive installation-performance system. That is not to say that this software does not already qualify for this description, but the broader vision for this project involves the camera not only tracking a drummer's movement, but also serving an the input to a machine learning-based model that makes musical decisions by referencing stored material (a previous, "learned" performance) to this incoming stream of motion data. The Bassline Synth will be replaced by a proprietary tone generator which will be controlled entirely by the camera data, as opposed to the user having to deliberately cross an arbitrary threshold to create a new pitch. However, such plans are not included in the following signal flow chart. It includes the Bassline Synth, which is acting as a substitute while this aforementioned, automated tone generator (next page).

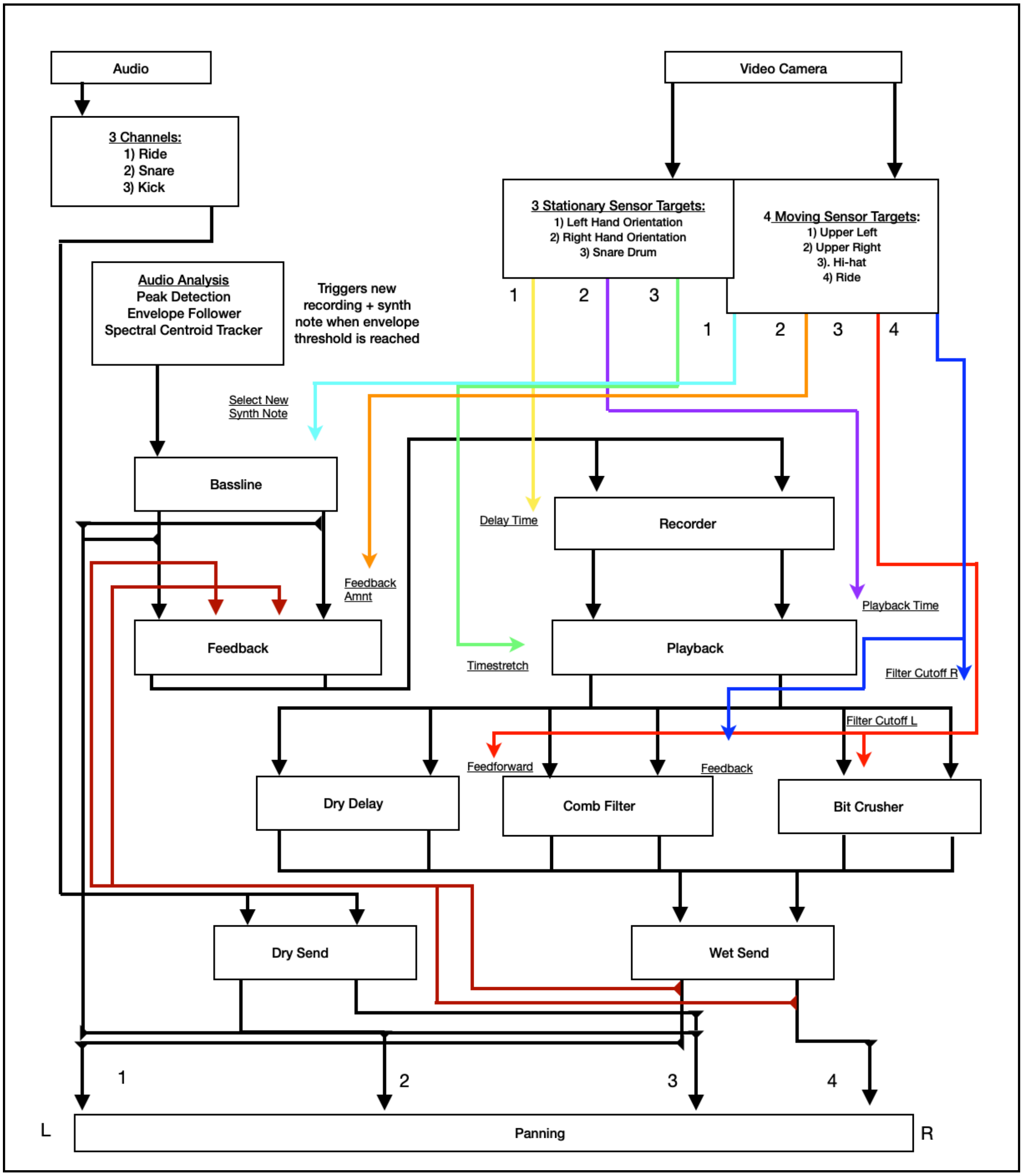

Figure 16. Signal Flow of Audio Inputs, Processing, and Target Sensor Mappings. Colors correspond to panel design in the Target Sensor Interface Tab, and are indexed from left to right (meaning, the light blue line coming out of “4 Moving Target Sensors” box corresponds to the Upper Left Moving Target Sensor

3. Moving Target Mappings

Moving Target Mapping Automation is labeled with MTA.

Figure 16.
Designation of Target
Movement Mapping
Automation

Moving Target Mappings: Upper Right

Moving Target Mappings control 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 (see Figure 10). Once this process is initiated, any movement of the Upper Right target zone along the horizontal axis will control the panning of four front speakers. The user can always revert the target zone back to its original position, which will also revert any automated Moving Target Mappings back to the Manual setting.

Figure 17: Workspace Space Set Up, Speaker Configuration.
Channel 1 is the left-most speaker.

While the horizontal movement controls the panning of dry and processed sounds, the vertical movement (when automated) manages the balance between the individual effects, which include the Comb Filter, Bit Crusher, and the Dry Delay, as well as the Master Dry Send. This is accomplished by dividing the overall horizontal and vertical movements of the Upper Right relative to pixel location each by half so that the overall video dimensions can be split into four individual quadrants. Each of these quadrants assigned an audio effects. Once the Upper Right Target Zone moves into a quadrant its respective gain level increases, just as it will decrease to zero as the zone proceeds to transition into another quad. This causes a dynamic balance between the different audio effects and the Master Dry Send.

Refer to Figure 18 above to follow this mapping example (next page):

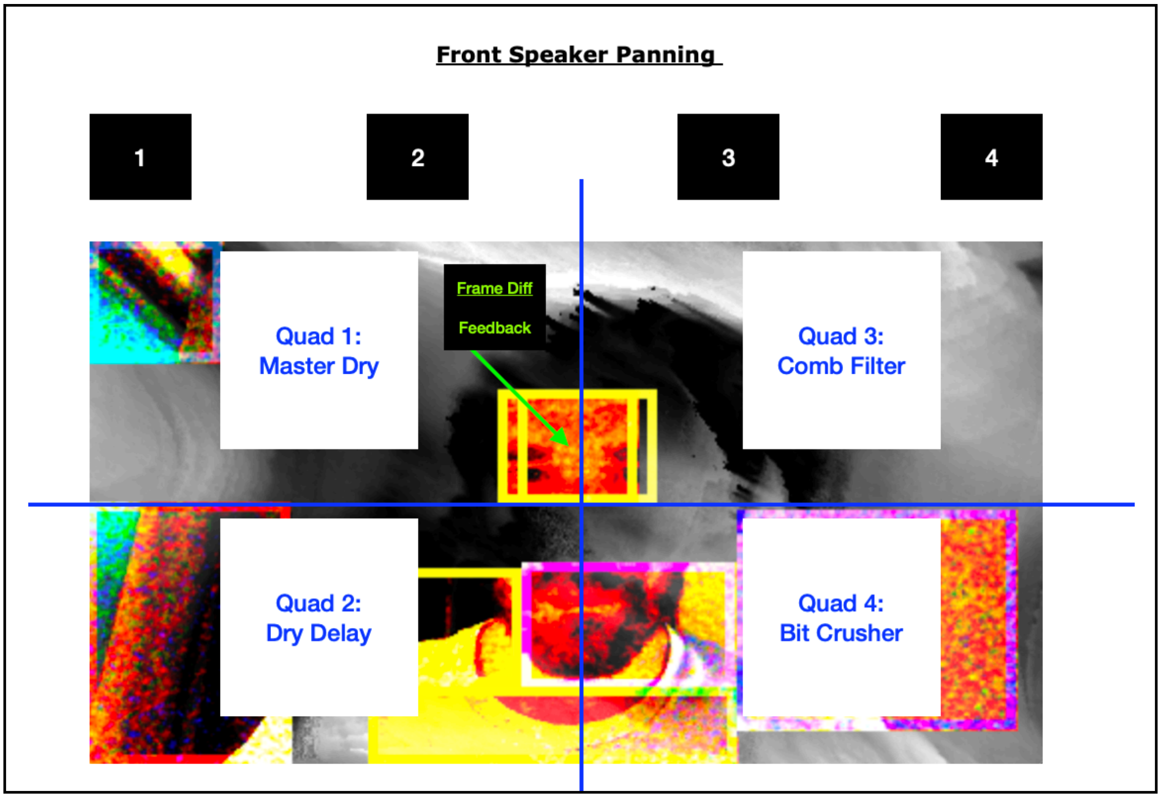

Figure 18: A Mapping Example

When automated, the randomized Upper Right Target Zone is currently moving between Quadrants 1 and 3. This means that there is gain being applied to the Comb Filter and the Master Dry Send, the exact amount scaled based upon the Target Zone's position within each respective quadrant. The horizontal movement controls panning. The Upper Right Target Zone is currently moving through an area that corresponds to non-zero panning values being assigned to speakers 2 ad 3. Whatever audio signal is routed through these two speakers will be audible, while any signal going to channels 1 and 4 is momentarily muted. However, it is important to keep in mind that Moving Target Mappings control balance, not processing. The Frame Differencing amount in the moving Upper Right Target Zone will directly translate to the amount of feedback is applied back into the recording signal chain.

Moving Target Mappings: Upper Left

Horizontal Movement in the Upper Left Target Zone controls the panning of the Synth Mode.[7] The vertical position can control the overall gain level sent to the DAC and the feedback system. Just like Upper Right Movement, Moving Target Mapping for the Upper Left has no effect on the Frame Differencing, which still triggers the pitch selection of the Bassline Synth when the activity exceeds a specified threshold. While the Upper Right and Left target Zones have been isolated for demonstration purposes, they can both be simultaneously randomized in practice.

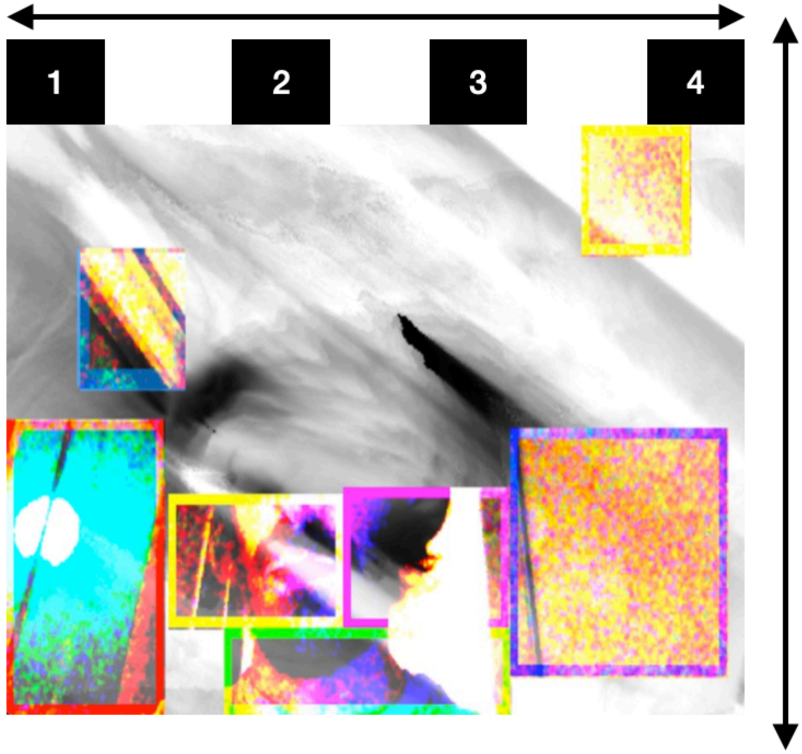

Figure 19: Moving Target Mapping for the Upper Left Target Zone. Horizontal Movement Controls Panning Balance. Vertical Movement Controls Gain Level

To prevent any unwanted automation or lack of personal control over the processing, panning or level balance, each of the Moving Target Mappings needs to be instantiated by the user individually next to the primary gain controls. The only gain control without automated capabilities is the Master Wet Send. If this were to be automated, it could unintentionally mute the individual effects sends, as they all go through the Master Wet Send[8]. This must be given some non-zero amount to send any processed audio into the DAC.

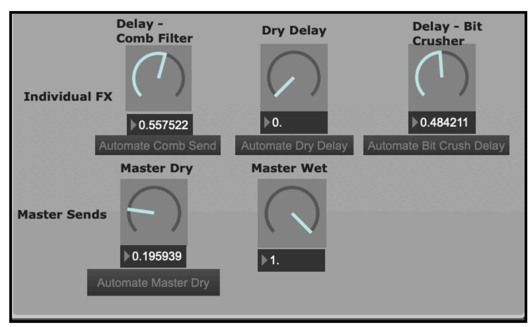

Figure 20. Main Gain Controls

Other Moving Targets

The Ride and Hi-hat Target Zones can also be randomized, although within a significantly smaller range the Upper Right or Upper Left Zones. While neither the horizontal nor vertical movement control the balance of any panning or amplitude levels, slight positional alterations within these areas give more comprehensive motion data, as they can account for physical activity in the shoulders, arms, and torso area that may be salient to executing a particular musical gesture.The positioning of four of these seven Target Zones can be continuously randomized at once, with the user choosing the specified parameters that will get automated based upon this movement.

---

[8] It is important to note that the only effect that is running through the ChamberVerb is the Comb Filter.

Emergent Interdependence of Mapping Relationships

From the combination of Target Sensors and Moving Target Mappings, layered relationships between user motion and audio processing emerge. While the emergence of these relationships are rather organic, and can vary based upon user input, light sourcing, and the Mode of choice, the experience of interacting with the software was always envisioned to be highly transparent, where the majority of the user's (performer) movements would represent a musical choice, rather than a function of achieving moving faders, pads, or a host of other interfaces. The camera cannot discern musical gestures from functional movement, so any activity is going to translate to some sort of processed sonification. The design implications for such an instrument are immense, as it means that the primary interface need not be any computer monitor, MIDI controller, or off-camera touch screen, but the subject in camera's path: the performer. The camera input, with the additional processing in Jitter, seemed like one of the most streamlined ways of constructing a transparent interface, and is the method utilized here in this current iteration of the project.

The intention was to eliminate as much functional movement as possible by automating some of the parameters that were related yet exclusive to the internal workings within the separate audio processing units (this is also the reason why here are so many steps in starting the software - so that most fo these functional movements have been made before the performance is initiated). Target Sensor Mappings were already assigned to their designated sound parameters, and the decision was made that this relationship would remain unaltered. The user would need to discern some relationship between their movements within these targets and sound, and splitting the primary processing of audio to different modes of interactivity seemed like it would confuse the user rather than lead to any sort of intrigue. As a result of this consideration, the Moving Target Mappings were assigned to panning and gain levels.

Refer once more to the initial, stationary positioning of the Target Sensor Zones:

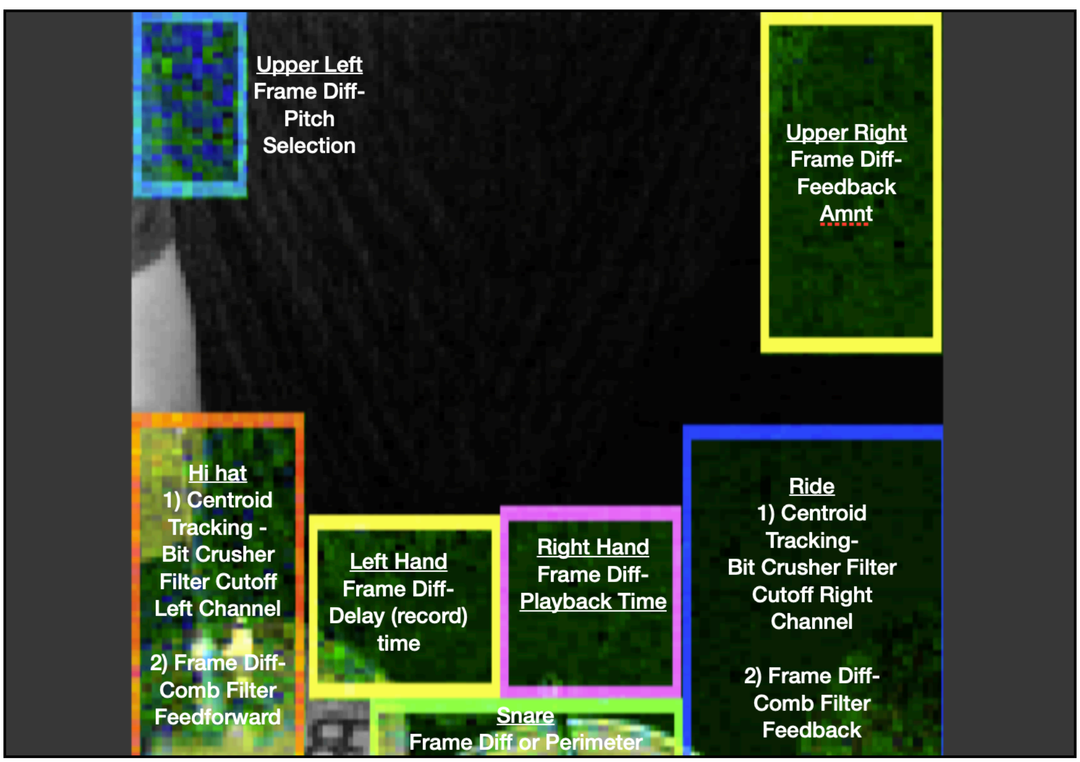

Figure 21: Initial Positioning of Target Zones, Corresponding Target Sensor Zone Mapping Assignments

In this stagnate state, the software retains its functionality, yet the drum set performer will always have to make a gesture far removed from any motion related to or necessary for playing the drums. This could be problematic is a performance, as these functional movements to the Upper Left and Right Target Zones limit the gestural possibilities (which is to say, musical possibilities) at a drummer's disposal once they engage with these regions. This illuminated a contradiction regarding the software's interface design. If one of the design goals was to eliminate as much functional movement (having to constantly and manually change faders or dials to control audio processing, as an example) - why would the interface be designed in such a way that made it possible to initiate certain audio processing parameters without the use of functional movement? This needed to be rectified, as it not only presented a design issue, but engaging in a target zone led to moving in a way that was physically uncomfortable and musically unsatisfying. The decision to include a positional randomization feature on four of the zones was made so that the drummer could choose how they wanted to interact with these target - either manually choosing discreet locations or having the position of the target zone within the camera feed to be continually updated for them. This decision to select between this binary has direct musical consequences. Automating the positioning for the Upper Left and Right target zones will provide a way to activate the mappings already established in the Target Sensor Zones while eliminating excess functional movement, but the randomization meant that the drummer could not determine the position, nor the rate of movement for these zones themselves once the featured was initiated. Since the amount of feedback sent to the [record~] object was mapped to the frame differencing data measured within this target zone, the drummer may not have full control over whether or not feedback is inserted into the signal chain while they are executing the physical gestures necessary for the performance of drum set vocabulary. In fact, it became apparent that this was the case for any Target Sensor Mapping tied to the same Target Zone that were simultaneously used for the purposes of Moving Target Mappings. Consider the following example of the Upper Left Target Zone operating in Automation Mode:

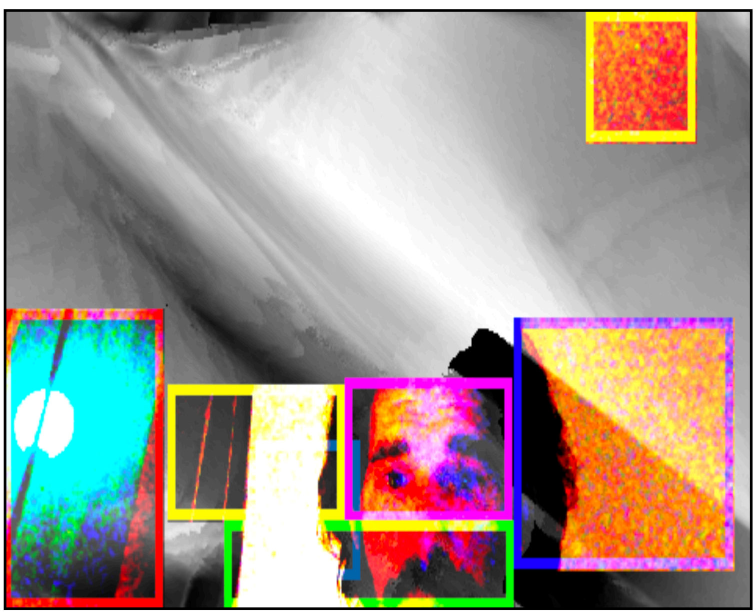

Figure 22: Upper Left Target Zone Operating in Automation Mode, Barely Seen in side 3 other Target Zones

This still shot is taken from the performer using the system in Motion Mode. The Upper Left Target Zone (faint blue) is moving further away from its initial position, and into the input range that is already occupied by three different Target Zones. So, if the performer meant to alter an audio effect mapped to the Left Hand target Zone, they would most likely also trigger the Bassline synth (if turned on, of course) to choose a new incoming pitch and velocity value pair. This context highlights the fact that there is some sonic indeterminacy built in the system due to the automated mapping strategies that are currently being implemented. This sonic indeterminacy is not only limited to just triggering MIDI values, but can occur with feedback gain values being sent to the [record~], which have been assigned (mapped) to the Upper Right Target Sensor Mapping Zone. While this could be considered a negative quality of the design, it could also be thought of as a musical context where there exists an emergent interdependence of mapping relationships, between that of the position of the Moving Targets and the sonification of Target Sensor Zone data. Panning is another optimal example that highlights the potential of these emergent relationships. Though seemingly random at time, their emergence provides musical constraints that can be learned and worked within over periods of practice, a dynamic that is familiar yet certainly more defined for practicers of a well-established acoustic musical instrument within the Western Music.

These randomized, indeterminate qualities in the visual (and by extension, audio) design also highlights the Sound Installation-centric aspects of the experience . These emergent relationships are not always apparent, nor are they consistent, because they are based on the random positioning of the Target Zones within the camera input. In effect, there is something to "figure-out" within the experience. Yes, someone is playing the drums (or any instrument), but there are moments of confusion, silence, chaos, and beauty that derives from the combination of these mappings, a real-time performance and the non-linear accessing of previously played musical material. These aforementioned emerging relationships create a game-like environment similar to that designed by Michalakos for *Icarus*, where the uninitiated user - who still cannot see the full real-input input from the camera - is constantly experimenting with their movements in an effort to discern how the consequentiality of their past gestures have on real-time and future decisions. Similar to *Icarus* the user must use the drum as controller, to determine the a present method of navigating these sonic exploration while having an obfuscated and highly processed curation of their own appearance fed back to them as an interface. The system does not only process gesture for musical purposed, but utilizes them as the primary subject of a multimodal art piece.

The game metaphor can be expanded upon even further. The Moving Target Mappings control panning and the balance between effect sends. Each of these distinct factors are controlled by different axes, and dry and wet sounds do not all feed into the same DAC inputs (Wet sounds go to DAC inputs 1 and 4, Dry ones to 2 and 3). This means that detains effects can be sent to the DAC based on horizontal movement, but based on a Moving Target Mapping Zones' vertical movement, the processed sound will not be audible. The user does not only have to familiarize themselves with the distinct and related contingencies with respect to Target Sensor and Moving target Mappings, but they also have to be aware of when a particular movement will represent the most meaning as a musical gesture.

Conclusion

This project only represents the first iteration of the CV Drum Tracker. Since there are few examples of interactive drum performance/installation systems to reference that also use Camera Vision as its designated input for motion tracking, this project is thought to be a working prototype upon which more improved versions will be released later. As a result of this consideration and an iterative workflow of programing, performing , and analysis, this project eventually became more centered around issues in digital instrument design than it was focused on any musical output. That is not to say that the musical results are of no importance in the least, but designing the CV Drum Tracker is equivalent to the act of building a new instrument - or at the very least - deriving a completely new way to interface with a of non-linear drum (audio) processor. This endeavor is at the stage where designing the ergonomics of hybrid instrument is more crucial if it is to be considered an effective musical tool by both both experiences and uninitiated users.

The initial interest in Camera Vision was to build utilities that could provide drummer with tools needed for motion tracking, so that they may better analyze how their physicality could be relating to musical gesture. However, the current system is a personalized one, created for an installation-performance hybrid. In an effort to make a system that is more utilitarian in its flexibility, the next iteration of experiments needs to involve other drummers, all deriving from a multiplicity of cultural backgrounds and musical experiences. Observing how other musicians react to the system in practice would also offer vital insight into how these timbral and temporal changes in drum gesture influence their real-time interaction and decision making processes. Collecting feedback from these diverse experiences would aid in identifying whatever programmatic and artistic biases may have been built into the system.

With that being said, this version of the software will persist as an experiment into developing a system capable of recognizing, indexing, and summarily making musical decisions that are influenced by the drum input or physical gesture. The ultimate artistic goal is to transition from a reactive model that triggers defined pitches from a the Bassline Synth into a generative model that creates its own melodic material through a stochastic process. Over time, the system will learn the body motions of a user and be able to associate MIDI pitch, duration, and velocity values in relation to them. These learned gestures could translated into the system generating new visual "Modes" that could accompanies the many differing musical modes that are now being prototyped, similar to the way *Icarus* has five levels that act as distinct musical modalities.