Jason Freeman

MTRX (2012)

for laptop orchestra

Performance Guide

About the Piece

In MTRX, the musicians in the laptop orchestra use a text-based performance interface to create, share, and transform rhythmic motives. These motives are built from a collection of sounds recorded throughout the city of Middletown, Connecticut by community members and Wesleyan University students. The sounds, captured between April and October 2012, were recorded using smartphones.

MTRX was commissioned by the Center for the Arts, Wesleyan University, Middletown, CT. LOLC, the software used by the laptop musicians, was developed by myself and my students Akito Van Troyer, Sang Won Lee, Andrew Colella, and Shannon Yao, with support from the National Science Foundation (NSF CreativeIT #0855758). UrbanRemix, the software used by Middletown residents to capture and share sounds with their smartphones, was developed by myself, Carl DiSalvo, Michael Nitsche, and many of our students, with support from Google, Turner, the Georgia Tech Foundation, and Georgia Tech's GVU Center.

Duration

Duration is flexible but expected to be approximately 15 minutes.

MTRX can also be performed in a looped form of indefinite length.

Instrumentation

MTRX requires at least 4 laptop musicians. There is no upper limit on the number of musicians in the ensemble.

Technical requirements

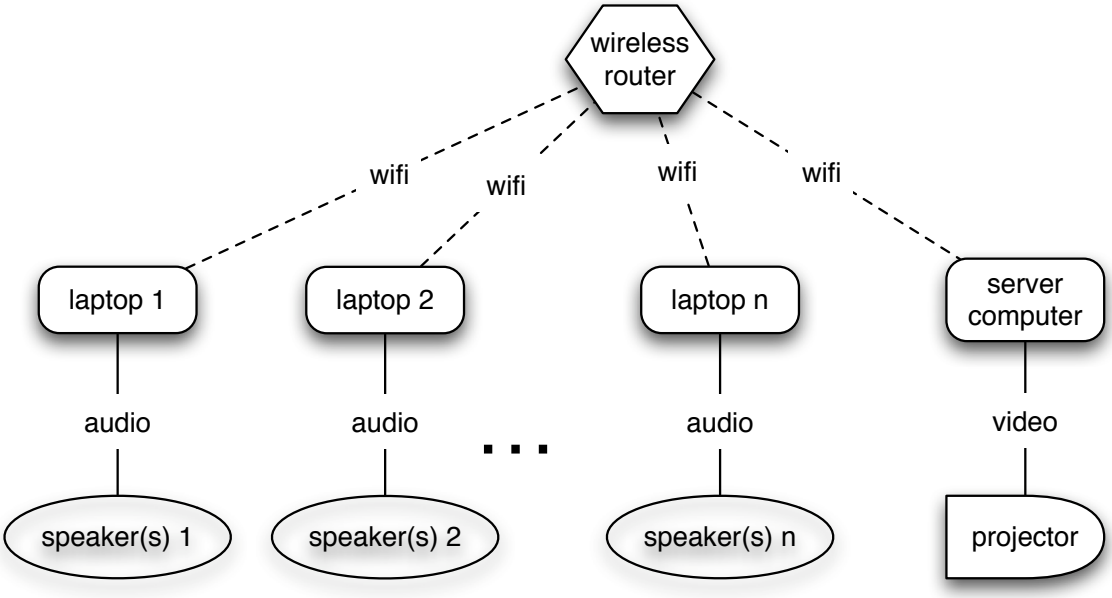
Each musician in the ensemble must have:

- A Mac or Windows laptop
- One or more speakers connected to her laptop in a mono or stereo configuration
- An external audio interface (optional)
- Headphones for audio previewing (optional, requires a multi-channel audio interface)

The following additional equipment is required:

- A wireless network router. The router need not be connected to the Internet.
- A Mac or Windows computer that acts as the server. I prefer that this be a separate machine, but if need be, one of the laptops in the ensemble can also operate as the server.
- A projector (optional but strongly recommended).

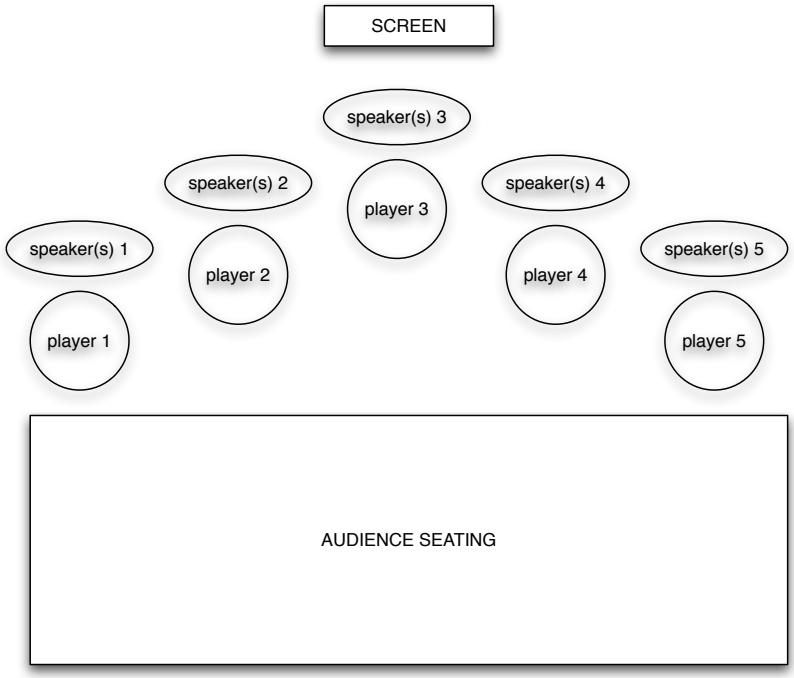
Wiring diagram



Performance Layout

The arrangement of musicians in performance is flexible, but each musician should be located as close to her respective speaker(s) as possible.

My preferred seating arrangement is a semicircle on stage, with the projection behind the musicians:



Installing the LOLC Software

All musicians in the ensemble must run a special MTRX build of the LOLC Client software on their laptops. The server computer must run a special MTRX build of the LOLC Server software.

To install and run this software, follow these steps:

1. Download the LOLC software from http://www.jasonfreeman.net/mtrx/.
2. Double-click the download to decompress it.
3. Install the JMSL license:
    a. In the JMSL folder, double click the JMSL_License_installer.jar file.
    b. Drag the JMSL.lic file onto the gray area to install the license.
4. Double-click the client or server application files to launch. It may take a minute for the software to launch; please be patient.

If you do not have a JMSL license, please contact me at http://www.jasonfreeman.net/contact/.

Learning LOLC

Performing MTRX requires you to learn how to use LOLC. The best way to learn LOLC is to read through the LOLC tutorial and try out the examples it includes. The LOLC tutorial is included with the LOLC client. After launching LOLC, choose Help->Tutorial from the menu. The MTRX build includes a special tutorial with examples specific to the sounds for MTRX.

The application also includes a quick command reference (Help->Reference).

Configuring LOLC

Individual Practice

When using LOLC by yourself, you must launch both the server and client on your machine:

1. Double-click the LOLC server.
2. Set tempo to 96, beats per measure to 4, beat unit to quarter, and mode to sound. Then click the start server button.
3. Double-click the LOLC client.
4. Choose your audio output device from the pop-up menu, and (if a multi-channel device) the channels you wish to use for main output and for headphone previewing. Then click the ok button.
5. In the Network dialog box, leave the IP address as “localhost” and the other parameters as they are. Click the connect button.

Ensemble Performance

When using LOLC in an ensemble, follow these steps:

1. Connect all computers (client and server) to your local wireless router.
2. Note the IP address of the server computer. The IP address is listed in System Preferences->Network on Macs.
6. Launch the LOLC server on the server computer. Set tempo to 96, beats per measure to 4, beat unit to quarter, and mode to sound. Then click the start server button.
7. Launch the LOLC client on each of the musicians' computers. Choose your audio output device from the pop-up menu, and (if a multi-channel device) the channels you wish to use for main output and for headphone previewing. Then click the ok button.
3. In the network dialog box in each LOLC client, enter the server's IP address (see step 2), a username of your choice, and (if using projection) a unique identifying color for your user. Then hit connect.
4. Verify the configuration by looking at the visualization window on the server. There should be a circle for each connected user. (Hit the space bar to toggle full screen visualization.)

If an LOLC client cannot connect to the server, try these troubleshooting tips:

- Make sure that the client machine is running on the same wireless network as the server.
- Make sure that you entered the IP address of the server correctly.
- Make sure that you are running the same version of LOLC as the server.

LOLC client Options and Features

Chat Pane

The chat pane is the main interface through which to interact with LOLC in performance. You can type in LOLC pattern definitions and scheduling operations in the bottom-right text field. To execute them, hit enter. To enter a chat message that the other musicians can read, hit shift-enter.

The chat pane includes several features to help you type faster. Clicking on any previous message in the chat view will copy it into the text field so you can quickly edit and re-execute it. Pressing the up arrow will cycle through a history of the previous code you have typed. And as you type, LOLC auto-completes your words: just click on a suggested word or hit tab to accept the default suggestion.

Pattern List Pane

The pattern list helps you to understand the patterns created within the ensemble and to borrow and transform material created by others.

The Pattern List shows all currently-defined patterns. The colors of blocks indicate the sound files they play, their brightness indicates their dynamic, the lengths of the blocks indicate the rhythmic duration of each event within the pattern, and their height indicates spectral centroid (how high or low the sound is). Since this list can quickly become long, numerous options on the "Organize" tab help you to sort and filter the list of variables to find specific patterns based on their author, the sounds they use, when they were created, and other criteria.

Audio Configuration Dialog

The Audio Configuration dialog, which is displayed on startup, lets you assign audio channels to the main audio output and to the preview output.

I encourage you to set up preview functionality so that you can hear patterns on headphones before scheduling them for playback over speakers. You can do this if you use a multi-channel audio interface.

Note that to change your audio settings, you must restart LOLC.

Console Pane

The console displays status and error messages. Pay special attention to the console for syntax errors when you execute pattern definition and scheduling expressions.

Sounds

The MTRX build of LOLC ships with a set of 71 sounds recorded by Middletown residents and Wesleyan students on their smartphones. More details about the sounds and their pre-defined names are below.

LOLC server Options and Features

Visualization

The main window of the server shows a visualization of LOLC performers, status, code, collaboration, and chat messages. I strongly encourage you to project this screen so that audience members can see it during performance.

To show the visualization in full-screen mode, hiding the dock and menu bar, hit the space bar.

In performance, you should configure LOLC so that the placement of each performer on screen corresponds to her physical location in the performance space. To do so, edit "stageLayout.xml" inside of the main LOLC folder on Windows (or right-click the application to Show Package Contents->Contents/Resources/lib/code on Mac). Follow the example file to create an entry for each performer that identifies their user id and x/y placement on screen.

Console

The console shows server status, error messages, and network connection activities (e.g. client connects and disconnects).

Config

The config dialog, which is displayed on startup, provides control over basic server functions. You can change the tempo (bpm), meter (beats per measure), and beat unit (by default a quarter note).

Menu

"Reload All Variables" will clear all patterns out of the server and reload the default sounds.

"Dump actions to file" will log all code and chat messages to a text file. Select it once to begin recording and a second time to stop recording. The file is stored in lib/stash/.

MTRX Sounds

MTRX includes 71 sounds. These sounds were recorded between April 2012 and October 2012 as part of Wesleyan University Center for the Arts' "Middletown Remixed" project. Wesleyan students and Middletown residents recorded and shared over 500 sounds from all around Middletown during this time, using the UrbanRemix smartphone software to record geo-tagged sounds with their smartphones.

The complete set of sounds is available at:
http://urbanremix.gatech.edu/content/middletown

I listened to all of those sounds and chose 58 sounds I felt had that greatest potential for use in MTRX:

http://urbanremix.gatech.edu/uploads?group_nid%5B%5D=5207&tid=laptop&username=&title=

These sounds chosen for MTRX were recorded by WESU, AHoyle, Mannetta, Cassandrasdis, bstolz, carterpeterson, kathylee, tsul, Jules Berman, diabonte, deveneciaj, Malyn, aasgarian, Ezralittlewood, Acannon, Mikawvawn, Gabebeaudoin, lhess, Marroyo, tgallivan, Eschatonic, Derek and Greg, ThatKidsFine, Kroosa, Mcost, Kikimoon7, mt, AEG, Brock and Matt, wblee, eibarra, dasullivan, Blindsay, and abrudnick.

I then chose 1, 2, or 3 excerpts from each of those 58 sounds to create the full set of 71 short, one-shot, percussive sounds for MTRX.

I then divided the 71 sounds into 8 classification categories based on their abstract sonic characteristics: music, klink, hollow, gesture, pitch, slam, two-step, and voice.

Within MTRX, I then created two complementary ordered sets of the sounds within each category. The first ordering is based on relative pitch (low to high). The second ordering is based on geographical location (loosely following a path from south to north).

To view the MTRX sounds on a map, visit:

http://urbanremix.gatech.edu:8080/urbanremix-webapp/?projectid=5207&latitude=41.5569&longitude=-72.6575&zoom=15

On the filter view on the right side, click on tags and select “laptop” to view all the MTRX sounds on the map, or click on one of the category names to view only the sounds in that category. Then click the view results button to see those sounds on the map.

To facilitate quick typing, MTRX defines sounds using a shorthand format. The first letter of the sound name represents the category (m=music, k=klink, h=hollow, g=gesture, p=pitch, s=slam, t=twostep, and v=voice). If the second character in the name is a ‘p,’ then it is part of a geographic path-based ordering. If the second character is a number, then it is a pitch-based numbering. In other words, k1, k2, k3, k4, and k5 represent the five klink sounds ordered by pitch from lowest (k1) to highest (k5), while kp1, kp2, kp3, kp4, and kp5 represent the same five klink sounds ordered by geographic path roughly from south (k1) to north (k5).

Here is the complete list of sound file definitions for MTRX:

m1 : "stair_railing__20120401_025656-low.wav"
m2 : "stair_railing__20120401_025656-high.wav"
m3 : "symphonic__20120923_081424.wav"

mp1 : "stair_railing__20120401_025656-low.wav"
mp2 : "stair_railing__20120401_025656-high.wav"
mp3 : "symphonic__20120923_081424.wav"

k1 : "Trash_-2012-09-25-22-44-43-2.wav"
k2 : "Trash_-2012-09-25-22-44-43-3.wav"
k3 : "Trash_-2012-09-25-22-44-43.wav"
k4 : "ding_20120923_051455.wav"
k5 : "chisel_20120921_123823.wav"

kp1 : "chisel_20120921_123823.wav"
kp2 : "Trash_-2012-09-25-22-44-43-2.wav"
kp3 : "Trash_-2012-09-25-22-44-43-3.wav"
kp4 : "Trash_-2012-09-25-22-44-43.wav"
kp5 : "ding_20120923_051455.wav"

h1 : "dumpster_plywood-2012-09-25-16-58-26.wav"

h2 : "night_basketball_north_end_20120925_070637.wav"
h3 : "movie_theater_20120924_015509-low.wav"
h4 : "movie_theater_20120924_015509-high.wav"
h5 : "Trash_-2012-09-25-22-44-43-4.wav"
h6 : "solo_on_fountain_20120923_042542-low.wav"
h7 : "solo_on_fountain_20120923_042542-high.wav"

hp1 : "solo_on_fountain_20120923_042542-low.wav"
hp2 : "solo_on_fountain_20120923_042542-high.wav"
hp3 : "movie_theater_20120924_015509-low.wav"
hp4 : "movie_theater_20120924_015509-high.wav"
hp5 : "Trash_-2012-09-25-22-44-43-4.wav"
hp6 : "dumpster_plywood-2012-09-25-16-58-26.wav"
hp7 : "night_basketball_north_end_20120925_070637.wav"

g1 : "dooropening_20120401_133738.wav"
g2 : "atm_usdan_20120924_040521.wav"
g3 : "coin_in_dryer_20120401_035253.wav"
g4 : "soundu_20120919_033338.wav"
g5 : "samcoins2-2012-09-21-18-48-36-low.wav"
g6 : "samcoins2-2012-09-21-18-48-36-high.wav"
g7 : "Farm_Hill_Swingset_20120920_034030-low.wav"
g8 : "Farm_Hill_Swingset_20120920_034030-high.wav"

gp1 : "samcoins2-2012-09-21-18-48-36-low.wav"
gp2 : "samcoins2-2012-09-21-18-48-36-high.wav"
gp3 : "Farm_Hill_Swingset_20120920_034030-low.wav"
gp4 : "Farm_Hill_Swingset_20120920_034030-high.wav"
gp5 : "dooropening_20120401_133738.wav"
gp6 : "atm_usdan_20120924_040521.wav"
gp7 : "coin_in_dryer_20120401_035253.wav"
gp8 : "soundu_20120919_033338.wav"

p1 : "bells4_20120921_121619.wav"
p2 : "crosswalk_to_home_20120924_073756.wav"
p3 : "Car_horn-2012-04-01-13-46-31.wav"
p4 : "restaruant_2_20120924_020208.wav"
p5 : "_campus_sound_1_20120921_122511.wav"
p6 : "elevator__20120401_025420.wav"
p7 : "touchingnature_20120922_072646-res.wav"
p8 : "parking_lot_2_20120924_015808.wav"
p9 : "rung_up_at_gas_man_20120405_114602-2.wav"
p10 : "Murmers_of_Check-Out_Desk_20120922_124628.wav"
p11 : "car_20120923_033114.wav"
p12 : "open_door_20120401_031149-2.wav"
p13 : "riteaid_20120923_030816.wav"

p14 : "rung_up_at_gas_man_20120405_11460-1.wav"
p15 : "crosswalkbutton_20120401_135056.wav"
p16 : "traffic_button_20120922_030654.wav"

pp1 : "rung_up_at_gas_man_20120405_114602-2.wav"
pp2 : "rung_up_at_gas_man_20120405_11460-1.wav"
pp3 : "touchingnature_20120922_072646-res.wav"
pp4 : "riteaid_20120923_030816.wav"
pp5 : "restaruant_2_20120924_020208.wav"
pp6 : "parking_lot_2_20120924_015808.wav"
pp7 : "Car_horn-2012-04-01-13-46-31.wav"
pp8 : "elevator__20120401_025420.wav"
pp9 : "bells4_20120921_121619.wav"
pp10 : "_campus_sound_1_20120921_122511.wav"
pp11 : "traffic_button_20120922_030654.wav"
pp12 : "Murmers_of_Check-Out_Desk_20120922_124628.wav"
pp13 : "open_door_20120401_031149-2.wav"
pp14 : "crosswalk_to_home_20120924_073756.wav"
pp15 : "car_20120923_033114.wav"
pp16 : "crosswalkbutton_20120401_135056.wav"

s1 : "fridge_20120923_012136.wav"
s2 : "scilidoor2_20120922_060831.wav"
s3 : "skaaaaate_20120924_033506.wav"
s4 : "door_slamming__20120401_035525.wav"
s5 : "gasman_station_20120923_045051.wav"
s6 : "Rolling_Chair_20120924_033224.wav"
s7 : "open_door_20120919_032908.wav"
s8 : "fountain_foosball_20120925_124403.wav"
s9 : "sign_slap_20120922_021713.wav"
s10 : "Rim_Grab_20120922_100916.wav"

sp1 : "Rim_Grab_20120922_100916.wav"
sp2 : "fountain_foosball_20120925_124403.wav"
sp3 : "scilidoor2_20120922_060831.wav"
sp4 : "fridge_20120923_012136.wav"
sp5 : "gasman_station_20120923_045051.wav"
sp6 : "open_door_20120919_032908.wav"
sp7 : "door_slamming__20120401_035525.wav"
sp8 : "skaaaaate_20120924_033506.wav"
sp9 : "Rolling_Chair_20120924_033224.wav"
sp10 : "sign_slap_20120922_021713.wav"

t1 : "jogging_20120401_011949.wav"
t2 : "walkingonrocks-2012-09-24-00-09-04-low.wav"
t3 : "walkingonrocks-2012-09-24-00-09-04-high.wav"

t4 : "freeman_door_2_20120921_015933.wav"
t5 : "car_clacks_20120922_020159.wav"
t6 : "mailbox_20120401_011448.wav"
t7 : "open_door_20120401_031149-1.wav"
t8 : "sound_20120922_095712.wav"
t9 : "touchingnature_20120922_072646-thud.wav"

tp1 : "mailbox_20120401_011448.wav"
tp2 : "car_clacks_20120922_020159.wav"
tp3 : "jogging_20120401_011949.wav"
tp4 : "open_door_20120401_031149-1.wav"
tp5 : "freeman_door_2_20120921_015933.wav"
tp6 : "walkingonrocks-2012-09-24-00-09-04-low.wav"
tp7 : "walkingonrocks-2012-09-24-00-09-04-high.wav"
tp8 : "sound_20120922_095712.wav"
tp9 : "touchingnature_20120922_072646-thud.wav"

v1 : "Long_Lane_Rugby_Practice-2012-09-25-18-11-58.wav"
v2 : "walk_light_is_now_on_20120922_032502.wav"
v3 : "sound_20120401_124854.wav"
v4 : "dog_2_20120401_124948.wav"
v5 : "chickens_20120922_011757.wav"
v6 : "dog_bark_20120922_022330.wav"
v7 : "hen_house2-2012-09-18-08-48-15.wav"
v8 : "court_20120922_022908.wav"
v9 : "long_lane_farm-2012-09-24-19-08-37.wav"
v10 : "hoophouse_owl_20120922_012526.wav"
v11 : "night_intruders_20120915_081010.wav"
v12 : "sound__night_intruders_20120827_081128.wav"

vp1 : "hen_house2-2012-09-18-08-48-15.wav"
vp2 : "chickens_20120922_011757.wav"
vp3 : "long_lane_farm-2012-09-24-19-08-37.wav"
vp4 : "Long_Lane_Rugby_Practice-2012-09-25-18-11-58.wav"
vp5 : "hoophouse_owl_20120922_012526.wav"
vp6 : "sound_20120401_124854.wav"
vp7 : "dog_2_20120401_124948.wav"
vp8 : "walk_light_is_now_on_20120922_032502.wav"
vp9 : "dog_bark_20120922_022330.wav"
vp10 : "court_20120922_022908.wav"
vp11 : "sound__night_intruders_20120827_081128.wav"
vp12 : "night_intruders_20120915_081010.wav"

MTRX Structure

MTRX is largely improvised, but there is a general formal structure that guides that improvisation in performance. This section describes the structure and should be used as a guide in performance and rehearsal.

Some important notes and explanations:

- The entire performance should last approximately 15 minutes, which is 360 measures at 96 bpm in 4/4 time.
- Always borrow and transform the patterns created by others, using whatever operations are allowed within the current section.
- Unless indicated otherwise, it is up to you to decide how to coordinate movement from section to section, and whether (and when) transitions should be sudden or gradual.
- The example code fragments are deliberately simple; do not feel restricted to using similarly simple operations and expressions in your performance.
- None of these guidelines are strict, and any specifics of the structure can be ignored or changed for good musical reasons, either in the moment of a performance or as planned through rehearsal.
- During performance, use text chat to coordinate among the players and to share that discussion with the audience.
- You may wish to appoint a "leader" within the group whose responsibility is to indicate, via text chat, when the group should move from one section to the next. Or you may want to assign this role to different players at different points in the piece.

Description	Example Code Fragments

Pitched, Metrical, Additive
Moving from left to right in the semicircle, each player loops a new pattern based on a single pitched (pp) sound. There is at least a two-measure delay between successive entrances. Each pattern's length is an integer multiple of 4 beats. The first player uses pp1, the second pp2, the third pp3, and so on.
	`ex1 : pp1[q.f,q,q,e,e]`
	`loop ex1 @ m`
	`ex2 : pp2[e,q.p,eh]`
	`loop ex2 @ m`
	`ex3 : pp3[wh.ff, h.p]`
	`loop ex3 @ m`

Transform and Combine
Players may use any operations they want to transform and combine patterns or create new ones. But all patterns must continue to use only pitched (p or pp) sounds and must have a length that is an integer multiple of 4 beats. Gradually increase intensity and density.
	`tc1 : shuffle(cat(ex1, ex2, ex3))`
	`loop tc1 ~4`
	`tc2 : merge(tc1, cat(pp4, pp5, pp6))`
	`loop tc2 @ 73`
	`tc3 : spread(ex3, 4, 7, 0.5)`
	`loop tc3 ~8`

More Categories
Players may also use klink (k or kp), hollow (h or hp), and slam (s or sp) sounds. When creating new patterns, players are encouraged to use geographically proximate sounds (e.g. hp2, hp3, hp4). Total pattern duration must still be a multiple of a whole note. Gradually increase intensity and density to first peak.
	`mc1 : cat(hp2[h], hp3[eq], hp4[e])`
	`play mc1 @ m`
	`mc2 : kp3[e,e,e,s,s,qe,t,t,t,t]`
	`play mc2 @ m`
	`mc3 : alternate(mc1, mc2)`
	`loop mc3`

Voice, Phased, Additive
Moving from right to left in the semi-circle, each player loops a new pattern based on a single voice (vp) sound. There is at least a two-measure delay between successive entrances. Patterns can have any length and need not line up with measure boundaries. The first player uses vp1, the second vp2, the third vp3, and so on. As soon as the first player has started their vp1 loop, all other players issue a kill * command to immediately stop playback of material from the previous three sections.
	`kill *`
	`vv1 : vp1[ww.f]`
	`loop vv1`
	`vv2 : vp2[h,hs.p]`
	`loop vv2`
	`vv3 : vp3[qe,qs.pp]`
	`loop vv3`

<u>Transform, Combine, More Categories</u>
Players may use any operations they want to transform and combine patterns or create new ones. In addition to voice sounds, players may also use twopart (t or tp) and gesture (g or gp) sounds. When creating new patterns, players are encouraged to use geographically proximate sounds (e.g. tp2, tp3, tp4). Gradually increase intensity and density.
	`tcm1 : amplify(vv1, 1.5)`
	`tcm2 : rotate(cat(vv3, vv2), 1)`
	`tcm3 : merge(tcm2[w,h,e], cat(tp1, tp2, tp3))`

<u>Anything Goes</u>
Players may use any sounds and operations they wish. Gradually increase intensity and density to second peak.
	`ag1 : s5[u,u,u,u,u,w.n]`
	`ag2 : spread(cat(tp4,tp5,tp6), 3, 7, 0)`
	`ag3 : rotate(ag1, 2)`

<u>Ending</u>
Some of the players introduce new patterns created solely with m1 and m2. These are quiet enough that they are initially inaudible within the larger texture. Then, gradually or suddenly, all other patterns (not using m1 and m2) are killed, leaving only the m1 and m2 patterns remaining. These are killed, gradually or suddenly, and the piece is over.
	`kill * @ 250`
	`kill *`
	`end1 : m1[q,qe,qs]`
	`loop end1`
	`end2 : cat(m1[w], m2[h])`
	`loop end2`

Looping MTRX

MTRX is intended primarily for concert performance, but the structure can be looped so that a performance in another setting can continue indefinitely.

To loop MTRX, the left-most player should simply begin another iteration of the structure (Pitched, Metrical, Additive) while the m1 and m2 sounds (Ending) are still sounding. Once the new iteration begins, the m1 and m2 sounds should be killed.

For extremely long performances, it may be desirable to divide the laptop orchestra into two groups that take turns (to avoid fatigue). Each group should have at least 4 players. When group 1 reaches the ending, they leave the stage to take a break while the m1 and m2 sounds are still playing. Group 2 takes their place on stage and begins the new iteration before the m1 and m2 sounds have finished.

Common Questions

Everything seems to work fine but I do not get any sound. What's going on?

Be sure you followed the installation instructions above and installed the JMSL license.

I am connecting my laptop's built-in audio output to my speakers — rather than an external audio interface — and I hear no sound. What's going on?

In some cases, you may need to switch the playback channel in Lolc-client's Audio Configuration window to channel 1 (instead of the default channel 0) when using built-in audio.

The visualization on LOLC Server looks funny. What's going on?

The visualization needs to be in full-screen mode to display properly. Hit the space bar to toggle between the small view and the full-screen view.

I have a mixed ensemble of Macs and PCs, and the Windows machines are not in sync with the Mac machines.

Some Windows machines introduce additional (significant) latency at the sound driver level, causing them to play at a consistent delay behind the Mac machines in the ensemble. To fix this problem, when you launch the LOLC client, set the driver latency time (in milliseconds) to compensate for this additional latency.

Why doesn't LOLC work on Linux?

LOLC may indeed work on Linux, but we have not tested it and do not support it.

Getting Help and Getting in Touch

Please do not hesitate to contact me with technical questions, bug reports, or other issues. I would also love to hear about your experiences with MTRX.

I can be reached via:

http://www.jasonfreeman.net/contact/

Most importantly, have fun with MTRX!

Jason Freeman

Atlanta, Georgia

November 4, 2012