



Tesi di diploma accademico di 1° livello in
TECNICO DI SALA DI REGISTRAZIONE

AMBIENTE ESECUTIVO PER PERFORMANCE AUDIO-VIDEO IN SISTEMI SURROUND

Diplomando: Simone Andolfato

Matricola: 0095

Relatore: Prof. Massimo Marchi



Quest'opera è stata rilasciata sotto la licenza Creative Commons **Attribuzione-Condividi allo stesso modo 2.5 Italia**. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-sa/2.5/it/> o spedisce una lettera a Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Prefazione

La costruzione di ambienti esecutivi che utilizzano tecnologie moderne ha portato negli ultimi anni ad un potenziamento, a livello di espressione e creatività, delle possibilità di esecuzione in tutte le arti performative. In questo progetto di tesi si è deciso di realizzare un software che permettesse, appoggiandosi a software già esistenti, maggiori potenzialità di creazione in un'ottica di espansione e sviluppo per creare nuovi strumenti. È proprio la possibilità offerta oggi dalle tecnologie nell'inventare nuovi mezzi di espressione che pone le basi di questo progetto.

Nel primo capitolo si è analizzata la storia della performance audio-video analizzando poi le motivazioni che hanno spinto alla creazione di questo software.

Nel secondo capitolo si è deciso di analizzare nello specifico le varie parti del software per quanto riguarda la parte video di questo progetto. Si è cercato di spiegare, più che il codice nel dettaglio, le modalità di funzionamento in modo da rendere il più comprensibile l'architettura del programma per facilitarne, con le dovute conoscenze informatiche, eventuali sviluppi futuri da parte di terzi.

Il terzo capitolo ha un approccio del tutto simile al secondo volto però ai plug-in disegnati per l'elaborazione della parte audio della performance.

Infine nell'ultimo capitolo si sono considerati eventuali sviluppi futuri ed altre applicazioni diverse dalla performance audio-video.

Ringraziamenti

Il progetto sviluppato in questa tesi è frutto di diversi insegnamenti che ho frequentato in questi tre anni di corso. Vorrei fare un ringraziamento generale a tutti i miei professori per quello che hanno saputo insegnarmi. Un particolare ringraziamento va al mio relatore Professor Massimo Marchi e al Professor Nicola Bernardini per avermi fatto scoprire molte potenzialità di un computer di cui non ero a conoscenza. Ringrazio anche Sampl e il Conservatorio C. Pollini per la disponibilità offerta nell'utilizzo di materiali. Un grande ringraziamento infine va al Professor Andrea Cera i cui consigli, riguardo i metodi di sincronia, sono applicati oggi in questo software e ne costituiscono un pilastro fondamentale.

Indice

Prefazione -----	p.1.
Ringraziamenti -----	p.2.
Indice -----	p.3.

Capitolo 1:

Storia della performance audio-video e strumenti software in commercio.

1.1	Introduzione -----	p.5.
1.2	Precursori -----	p.6.
1.3	Anni 70 -----	p.7.
1.4	Anni 80 -----	p.9.
1.5	Anni 90 -----	p.10.
1.6	Anni 2000 -----	p.16.
1.7	Software per vjing -----	p.20.

Capitolo 2:

Video for club environment.

2.1	Introduzione -----	p.26.
2.2	Ableton Live, MaxforLive e Max/Jitter -----	p.27.
2.3	MaxforLive plug-in -----	p.31.
2.4	Max/Jitter Pacth -----	p.38.

Capitolo 3:

Audio control surfaces

3.1	Introduzione -----	p.51.
3.2	Pan Surround -----	p.51.
3.3	Step.api -----	p.55.

Capitolo 4:

Sviluppi futuri ed altre applicazioni.

4.1 Sviluppi futuri -----p.58.

4.2 Altre applicazioni -----p.60.

Bibliografia -----p.61.

Sitografia -----p.61.

Capitolo 1

Storia della performance audio-video e strumenti software in commercio

1.1 Introduzione

Sin dagli albori l'esecuzione musicale dal vivo è stata spesso accompagnata da altre forme artistiche. Danza e coreografia hanno accompagnato l'esecuzione musicale dando vita a specifici generi musicali che risulterebbero incompleti in mancanza di queste.

Il teatro d'opera per esempio è l'unione di musica poesia e teatro e rappresenta forse il primo spettacolo dove più forme artistiche si fondono nel creare qualcosa di mai visto prima.

Andando a ritroso nella storia si potrebbe individuare un'enorme quantità di radici e collegamenti a quello che vuole essere l'oggetto di questa tesi.

Si è dunque deciso di analizzare l'argomento da un punto di vista tecnico-scientifico riportando i cenni storici che interessano direttamente le performance audio-video quali le evoluzioni tecnologiche e le trasformazioni culturali che ne permettono la realizzazione.

1.2 Precursori

Come già detto le radici storiche della performance audio-video possono essere ricondotte a numerose altre forme artistiche del passato.

Il primo antecedente che si è preso in considerazione è l'organo a colori in quanto primo strumento ad essere concepito per un utilizzo audiovisivo.

L'idea risale al cinquecento ma la sua realizzazione è stata resa possibile dall'evoluzione tecnologica solo alla fine dell'ottocento.

Bainbridge Bishop, che ha contribuito alla sua realizzazione, scrive in un libro che documenta il suo lavoro, di aver progettato un meccanismo elettromeccanico il quale collegato alla tastiera dello strumento genera luci colorate in rapporto con il suono.^[1]

Tra il 1919 e il 1927 Mary Hallock-Greenewalt, pianista solista, utilizza l'organo a colori dando origine ad una nuova arte tecnologica chiamata Nourathar (in arabo essenza della luce).

Durante gli anni sessanta nascono nei club e negli eventi privati in America le prime palle a specchi, proiezioni di diapositive e proiezioni di luci su fumo per dare alle persone nuove sensazioni. Alcune di queste erano solo decorative, altre in relazione con la musica. Presero il nome di Liquid Light Shows e grazie a collettivi artistici come The Joshua Light Show e Brotherhood of Light accompagnarono i concerti dei The Grateful Dead tra il 1965 e il 1966 a S. Francisco.

Nei due anni successivi Andy Warhol contribuisce alla fusione di musica e visual organizzando gli Exploding Plastic Inevitable legando queste forme artistiche nel contesto di una festa.^[2]

1.3 Anni 70

Alla fine degli anni 70 le performance audio-video erano meglio integrate l'un l'altra. Ai concerti alcune band cominciarono ad avere un addetto alle proiezioni che diventavano parte integrante dello spettacolo.

Sono sicuramente i precursori dell'attuale ruolo del vj. Alcuni locali cominciarono ad avere un vj resident durante i concerti.

Gruppi come i Cabaret Voltaire furono tra i primi ad utilizzare della strumentazione low cost per editare le diapositive o le pellicole da proiettare nei loro concerti.

Dal punto di vista tecnologico, dopo l'avvento del primo sintetizzatore audio, si era molto vicini all'invenzione del sintetizzatore di immagini.

Importante in questo settore fu il Experimental Television Center, creato a New York nel 1971, dove gli artisti avevano libero accesso all'hardware sperimentale utilizzato da video pionieri come Nam June Paik, Steve Rutt e Bill Etra.



Sempre nel '71 nacque il Sandin Image Processor (IP) il primo hardware per l'elaborazione analogica dell'immagine open source. Esso infatti veniva distribuito al prezzo dei componenti elettronici necessari per il montaggio con una guida contenente gli schemi elettrici e le istruzioni per essere montato. In questo modo l'utente poteva sperimentare diversi modi di montaggio ed eventualmente aggiungere componenti per sperimentare nuovi effetti. Grazie a questo processore low cost ed open source si è creata la possibilità di ricercare e sviluppare ottenendo man mano che le tecnologie diventavano più accessibili sempre più effetti video.

1.4 Anni 80

Lo sviluppo della musica elettronica, soprattutto la house e la techno, ha fatto sì che i dj club offrissero più possibilità ed attrezzature per accompagnare con visual le serate.



La televisione negli anni 80 è già il più grande mass media di tutti i tempi. Ecco che Mtv diventa il primo canale televisivo mondiale ad occuparsi esclusivamente di musica. I videoclip musicali dove le canzoni sono accompagnate dal video, sono disponibili sia in televisione sia su supporti VHS e molti locali scelgono di creare l'atmosfera del luogo anche attraverso la proiezione di questi.

All'interno di mtv nasce una nuova figura professionale: il Vj. Come un Dj per le radio il Vj di Mtv si occupa della programmazione dei video da mandare in onda.

Questa nuova figura non è però da confondere con la più attuale ed omonima professione del Vj che lavora durante gli spettacoli dal vivo.

Cominciano i primi contest che offrono premi in denaro per la realizzazione di videoclip musicali o comunque lavori video su VHS che offrono ad un vj molte più possibilità di lavoro rispetto al passato.

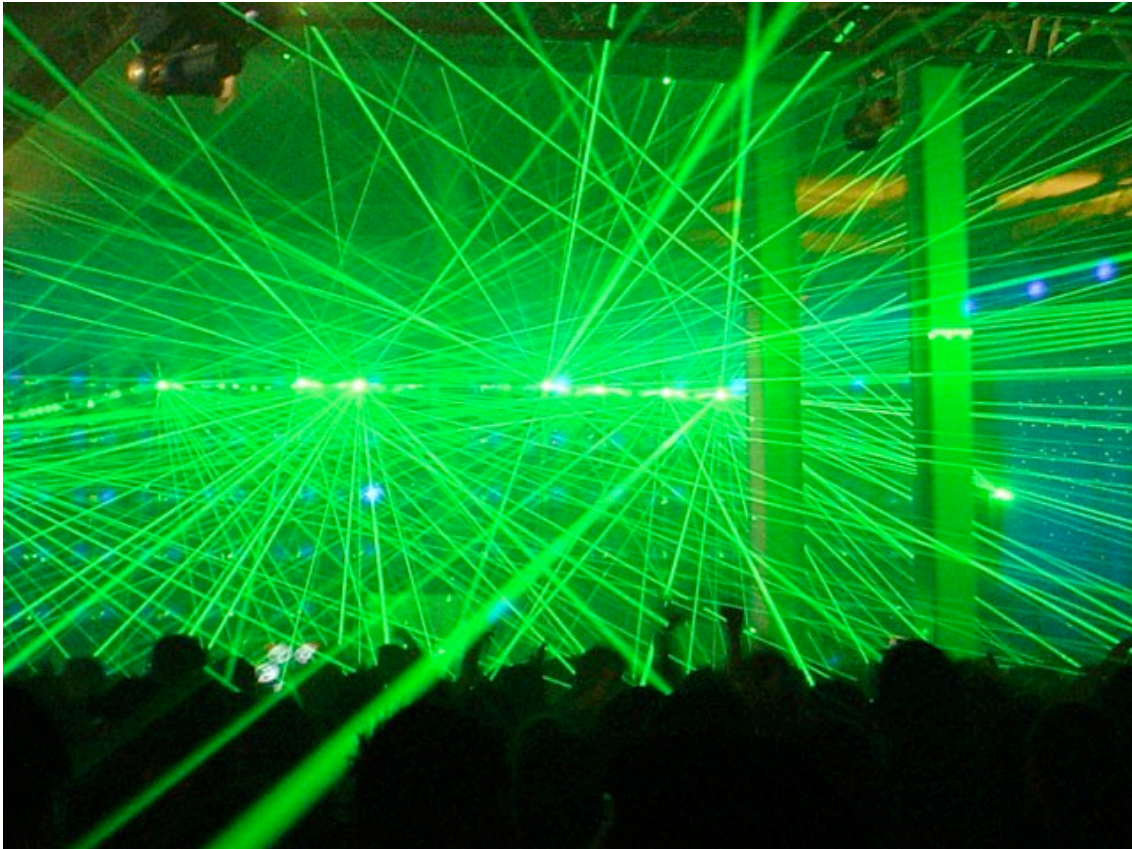
Gli sviluppi tecnologici degli anni 80 riguardanti la realizzazione di transistor e circuiti integrati più economici hanno reso possibile, da parte di un vj o di un gestore di locali, l'acquisto di hardware dedicato all'elaborazione in digitale del video.

Tra i primi sintetizzatori video distribuiti sul mercato nel 1981 viene prodotto il CEL Electronic Chromaspace e nel 1983 il Computer Video Instrument il quale permette l'elaborazione digitale in tempo reale di un qualsiasi input video.

Il Commodore Amiga introdotto sul mercato nel 1985 rappresenta una specie di rivoluzione nel campo: rende possibile la creazione di animazioni 2D o 3D da un comune PC. L'accessibilità dell'amiga non ha precedenti. Si possono quindi realizzare video per broadcast semplicemente dal pc di casa.

1.5 Anni 90

Gli anni 90 hanno segnato una grande svolta nella storia della performance audio-video. Si è passati dall'uso come scenografia nei club ad un utilizzo più vasto, sia a livello di numero di eventi sia di tipologia di performance: non più quindi solo scenografie di contorno allo spettacolo musicale nei locali ma bensì vere e proprie installazioni video anche interattive.

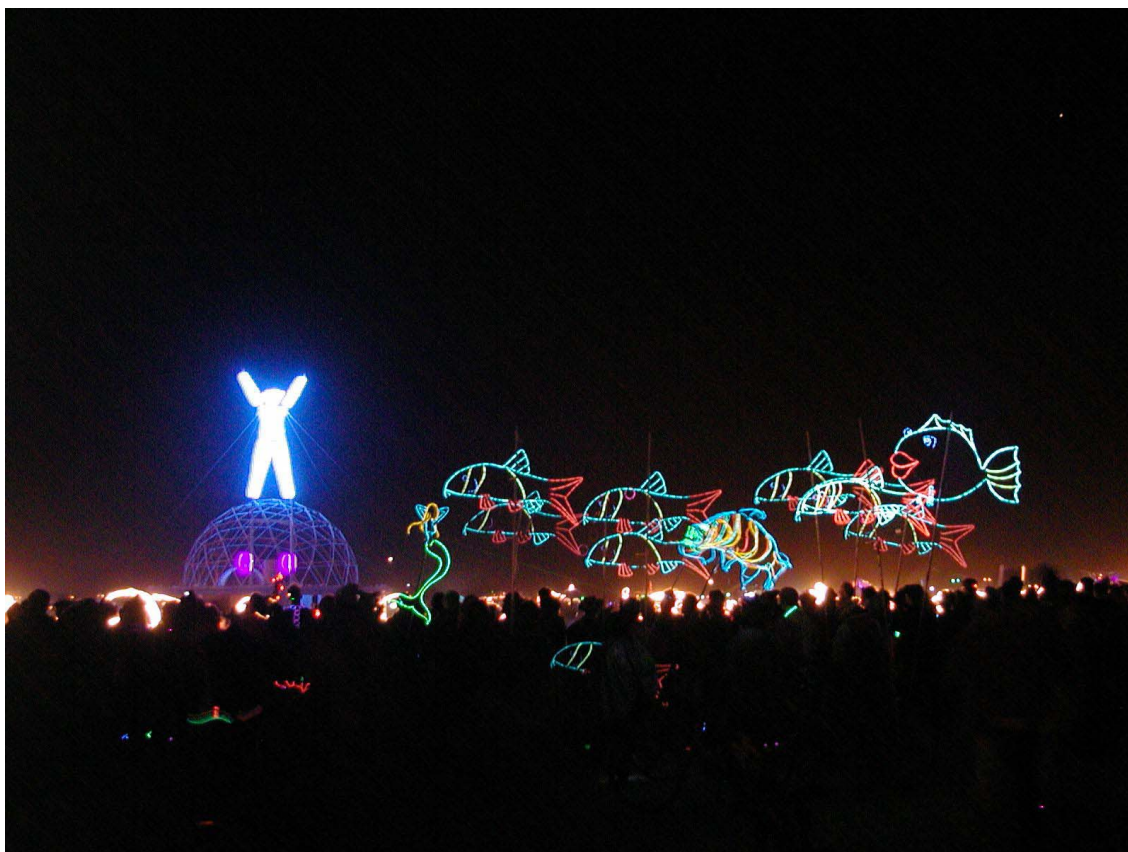


La rave culture sviluppatasi nei 90, ha dato un contributo senza precedenti all'evoluzione del vjing spostandone il baricentro dai nightclub ad una scena underground dove il vj poteva disporre di maggiori spazi, mezzi e poteva sviluppare la propria creatività in modo più libero e senza i vincoli imposti dai gestori di locali.

In America l'area di Chicago era molto attiva nell'organizzazione di rave party tra i quali il "Massive New Year Eve Revolution" che riscosse molto successo e fu seguito da numerosi altri eventi che ricevettero l'attenzione di Psychic TV. In questi rave fu molto importante l'operato del collettivo Optique Vid Tek (OVT) che ne realizzò i visual.^[3]

Dimension 7 fu un altro collettivo americano che si lavorò ai visual di molti festival nella rave scene della west coast cominciando a lavorare nel 93.

Tra il '96 e il '98 effettuano proiezioni al Burningman festival, uno dei più importanti festival di musica della durata di una settimana nel Black Rock desert nel nord Nevada.



Nel Regno Unito gruppi come The Light Surgeons e Eikon hanno trasformato i club e i rave combinando la vecchia tecnica delle Liquid Lightshows con proiezioni video. Un altro collettivo, Hex, ha lavorato con una vasta gamma di media, dai giochi per il computer alle mostre di arte diventando pionieri nella realizzazione di nuovi ibridi multimediali come le jam session audio-video, performance audio generate tramite algoritmi al computer, e nuovi strumenti interattivi dove il pubblico collabora alla realizzazione dell'opera. Il lavoro di questi collettivi è molto importante perché è l'inizio di una tendenza che continua tutt'ora: molti vj di oggi vedono la performance video

come un'installazione artistica e cercano un punto di incontro tra pubblico ed esecutori.

Gli anni 90 sono molto importanti anche dal punto di vista delle pubblicazioni. Un gran numero di pubblicazioni riguardanti il mondo dei VJ cominciano infatti ad essere disponibili sul mercato. Tra queste vanno ricordate le XMix Compilation che cominciano ad essere pubblicate nel '93, Lifeforms del duo Future Sound of London nel '94, Telecommunication Breakdown del collettivo Emergency Broadcast Network del '95 che vanta la partecipazione di artisti come Brian Eno, Timber dalla collaborazione di Coldcut e Hexstatic nel '97 e la The Mego Videos una compilation di lavori tra '96 e '98 pubblicata nel '99.

Da ricordare anche il CD Odyssey Through O₂ di Jean Michel Jarre che esce contenendo una copia del software per VJ ArKaos.

Nel '99 viene inoltre pubblicato VJ2000 di Daizaburo Harada uno tra i primi libri dedicati alla pratica del VJ.

Molte di queste pubblicazioni originariamente su supporto VHS vennero poi ristampate su CD-Rom allegando anche del materiale software.

Il lavoro degli artisti negli anni 90 è fondamentale per quanto riguarda l'evoluzione degli strumenti per creare performance audio-video. Sono infatti gli stessi artisti spesso che in mancanza di mezzi disponibili sul mercato si impegnano a costruirseli. Molti artisti lavorano all'interno di un collettivo sviluppando insieme idee e strumenti per le performance. In questi gruppi di lavoro spesso c'è chi studia per creare gli strumenti più adatti alle proprie esigenze, molti software per i visual infatti sono stati sviluppati da questi collettivi e distribuiti insieme ai lavori pubblicati su CD. La nascita della rave scene affiancata dalla maggiore disponibilità di tecnologie più convenienti per l'editing video home made, ha aiutato gli artisti nello sviluppo di ambienti esecutivi per le loro performance.

Ci sono quattro principali fattori che guidano lo sviluppo dell'audio-visual durante gli anni 90:

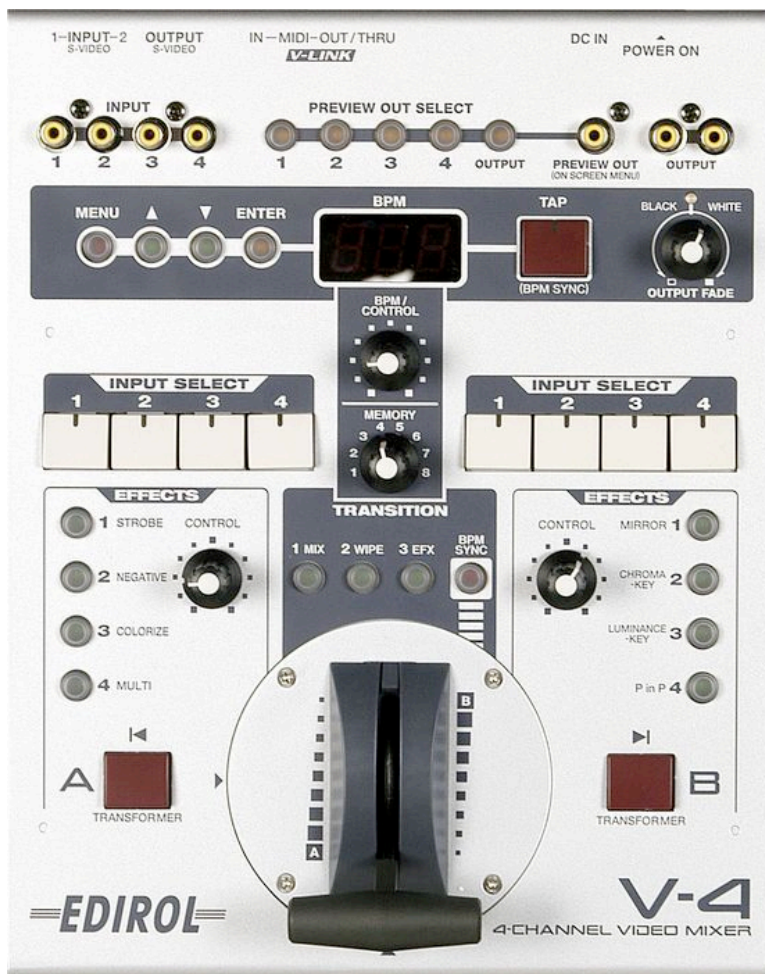
1. La nascita di computer portatili più potenti e affidabili.
2. La nascita di video mixer digitali.
3. La diminuzione del costo di proiettori e impianti per i locali.
4. La crescita di una scena musicale legata ai rave party e ai club a livello internazionale.

In questo decennio la tecnologia video è passata dall'essere una tecnologia per solo uso professionale all'interno di studi televisivi, quindi con prezzi proibitivi per il piccolo consumatore, all'essere sempre più accessibile.

I mixer, per esempio, prima erano utilizzati dai vj negli studi per preparare materiale pre-editato da proiettare durante gli spettacoli. Con lo sviluppo di mixer video come i Panasonic WJ-MX50 e WJ-MX12 e il Videonics MX-1 i vj non attendono a lungo per utilizzarli, aumentando così di molto le possibilità di generazione di materiale durante i live.

Nel 1998 la Roland/Edirol rilascia il V5 Video Canvas. Si tratta di un mixer video ibrido, provvisto di una memoria a stato solido che permette l'archiviazione di immagini e la possibilità di utilizzarle miscelandole con i classici input video. Questo prodotto segna un importante punto di transizione: le case produttrici di hardware musicale vedono l'affermarsi di un forte mercato anche per gli hardware dedicati alla performance video.

Il prodotto che seguì il V5 diventò parte fondamentale del setup base di un vj.



Come già visto i vj avevano già cominciato ad usare primordiali programmi per l'editing del video al computer come il New Tek Video Toaster per l'Amiga. Il primo software noto scritto appositamente per i vj fu Vujak e fu sviluppato nel 1992 da Brian Kane un artista del collettivo Emergency Broadcast Network.

Dalla metà degli anni 90 sono disponibili diversi software sia per la generazione che per la riproduzione di materiale video come MooNSTER, Aestesis, Advanced Visualization Studio, Motion Dive, ArKaos, e VJamm.

Altri software come Max/MSP, Macromedia Director e più tardi Quartz Composer cominciano ad essere disponibili e permettono all'artista un maggior controllo del proprio lavoro.

Una delle caratteristiche più rilevanti degli anni 90 è la costante evoluzione dei computer che hanno visto la propria capacità di calcolo ottenere sviluppi senza precedenti.

L'affidabilità e la potenza raggiunta dai computer negli anni 90 hanno fatto sì che il laptop diventasse elemento fondamentale ed insostituibile per il setup di una performance video e sarà utilizzato e sviluppato negli anni seguenti tramite lo sviluppo di software in grado di sfruttare sempre di più la continua evoluzione dell'hardware.

1.6 Anni 2000

Nel primo decennio del nuovo secolo la performance video ha trovato terreno fertile per poter essere praticata sempre più spesso.

L'evoluzione tecnologica con crescita esponenziale delle capacità di calcolo dei computer, l'affermazione sul mercato di proiettori, videocamere ed altre attrezzature video anche per il piccolo consumatore, ha reso possibile sviluppare questo tipo di performance in questi dieci anni, più di quanto non fosse stato possibile in tutta la storia del video.

La sempre maggiore disponibilità di internet per il singolo consumatore ha permesso ai vj la creazione di siti, forum o community dedicati al dibattito, sviluppo e condivisione di idee e software. Se prima i vj erano isolati in una professione tanto sconosciuta quanto rara, ora si confrontano e si scambiano materiali. Nascono i primi archivi on-line di samples e loop video, dove i vj possono condividere il proprio lavoro.

Molti furono i festival che diedero sempre più attenzione ai visual: il VideA a Barcellona dal 2000 al 2005, AVIT organizzato on-line su VJCentral.com la prima volta a Leeds nel 2002 seguito da Chicago e Brighton nel 2003, San Francisco nel 2004 e Birmingham nel 2005, 320x240 in Croazia nel 2003, Contact Europe a Berlino nel 2003.

Il Live Performers Meeting (LPM) nasce a Roma nel 2004 con il tentativo di dare una risposta al bisogno di creare un punto di riferimento per il lavoro nell'ambito della performance video. Dopo un'altra edizione in Messico nel 2008 LPM è oggi un ritrovo internazionale di professionisti, performers ed appassionati di visual e performance video live.

Il Mapping festival di Ginevra e il Vision'R di Parigi furono organizzati per la prima volta nel 2005 e furono tra i primi festival interamente dedicati ai Vj.



Pionieri nella ricerca del visual degli anni 90 come Grant Davis (VJ culture) e Jon Schwark dei Dimension 7 crearono Video Salon una raccolta mensile utile per creare un'identità per la community di Vj di San Francisco e California.

In questo decennio nascono inoltre numerosi contest e competizioni per vj, nelle quali gli artisti si confrontano allo stesso modo dei dj nei loro contest: veloci performance nelle quali sfruttare al massimo la propria abilità per generare il miglior mix di video.

Durante i 2000 si assiste ad una crescita senza precedenti di eventi dove si afferma la figura del vj. Le caratteristiche principali di questi anni sono la sempre maggiore disponibilità di community dedicate all'argomento che favorendo il confronto tra gli artisti accelerano, molto più rapidamente rispetto al passato, lo sviluppo di questa forma artistica. La diffusione di massa di internet spezza i confini e permette ad un genere artistico poco conosciuto di evolversi nonostante sul territorio vi siano ancora oggi pochi centri specializzati in materia.

Importantissima è in quest'ambito l'evoluzione tecnologica che, come già visto negli anni precedenti stabilisce il confine tra realizzabile o meno.

I limiti dettati dall'hardware infatti, hanno limitato le possibilità creative degli artisti sia dalla nascita della performance video. Costo di proiettori, attrezzature come mixer ma soprattutto mancanza fisica di supporti adatti all'utilizzo nei live hanno rallentato lo sviluppo in questo settore.

La sincronizzazione di una VHS o di diapositive in un live è impossibile e questo ha condizionato la tipologia di contenuti video utilizzati negli spettacoli.

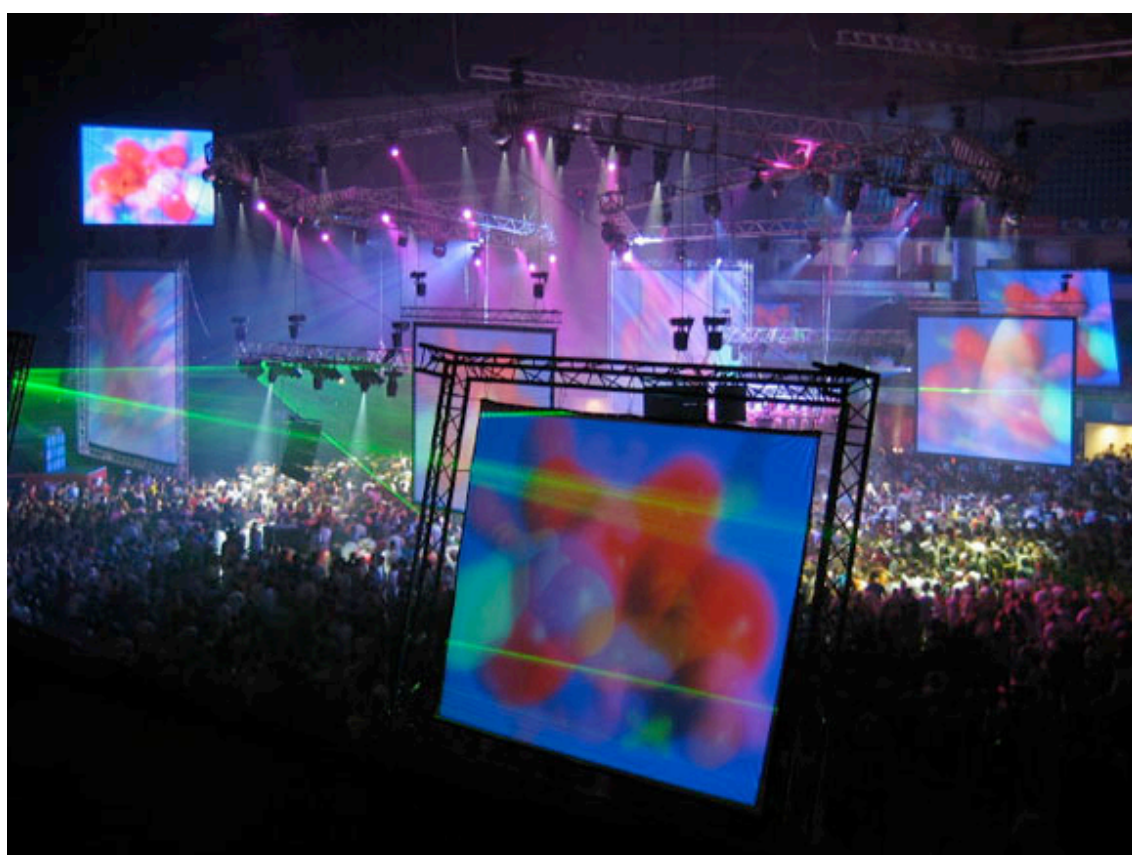


Nel corso dei 2000 si è raggiunto uno sviluppo tecnologico che consente la realizzazione di qualsiasi (fisica permettendo) performance.

Mixer digitali, DVJ player, proiettori ad alta luminosità, schermi led e al plasma e computer permettono oggi l'esecuzione live di video come uno strumento musicale permette allo strumentista di suonare.

La vera rivoluzione di questi anni è l'enorme crescita nella capacità di calcolo dei computer: è sufficiente infatti un normale computer di oggi per poter realizzare una performance per la quale dieci anni prima serviva molta strumentazione spesso costosa.

Accanto all'evoluzione hardware c'è uno sviluppo parallelo di software dedicati al vjing. Grazie a questi software il vj può preparare la performance sul proprio laptop e sarà sufficiente collegarlo all'impianto video nella location della performance.



Nel seguente paragrafo si analizzeranno alcuni tra i più utilizzati software per performance audio/video nelle loro caratteristiche tecniche.

1.7 Software per vjing

Come si è visto negli ultimi dieci anni il computer è diventato non solo parte integrante dell'ambiente esecutivo per performance video, ma soprattutto parte insostituibile e fondamentale per la sua realizzazione. L'evoluzione dell'hardware dell'ultimo decennio ha permesso lo sviluppo di software sempre più avanzati per la riproduzione ed elaborazione in tempo reale di materiale video. La quantità di software disponibile oggi a questo scopo è estremamente vasta ed è impossibile effettuare un'analisi dettagliata di tutti questi. Si è deciso quindi, in questo paragrafo, di trattare l'argomento senza entrare nel dettaglio del singolo software ma di analizzare funzioni basilari comuni alla maggior parte di questi software e di approfondire in modo più dettagliato solo i più utilizzati.

Molti di questi software nascono come già visto dagli stessi artisti che gli utilizzano e sono spesso il risultato del lavoro di più persone che collaborano allo sviluppo in ambienti open source. Con l'avvento di internet e di community dedicate al vjing gli artisti si confrontano e condividono il proprio lavoro creando software sempre più performanti. Il diffondersi di questo tipo di performance all'interno di locali attrezzati od eventi, dove sempre più il video comincia a far parte della coreografia, pone le fondamenta per poter commercializzare software dedicato al vjing. Nascono quindi software professionali che devono però accontentare molte esigenze diverse. Si viene a creare uno standard per quanto riguarda le funzioni principali che devono essere disponibili in un ambiente esecutivo per performance video. Il setup di base deve permettere la gestione di diversi input video contemporaneamente, deve permettere la modifica in tempo

reale e dare la possibilità di mixarli tra loro in almeno un'uscita video (proiettore). Questi media possono essere di vario tipo come una videocamera, un file video, un algoritmo che effettua sintesi o altro. Prima dei computer era necessario disporre di diversi mezzi (registratori VHS, strumenti di sintesi, effetti, mixer) per eseguire una performance. Oggi un computer da solo riesce a fare molto di più. I vari input possono essere collegati ad un computer (videocamere usb o firewire) o direttamente riprodotti dal suo interno (file video, algoritmi di sintesi ecc.). Non essendoci più la necessità di mixare esternamente le sorgenti video anche il ruolo di un mixer non è più fondamentale come in passato, tuttavia sono molti i vj che continuano ad utilizzarli per motivi di stabilità o semplicemente per espandere le potenzialità della performance utilizzando per esempio più di un computer. Sono evidenti i vantaggi che il computer ha portato in questo settore sia dal punto di vista praticità nel montaggio dell'ambiente esecutivo sia da quello economico.

Il software ha dovuto rimpiazzare tutte quelle funzioni effettuate prima da hardware esterni che sono poi diventate le caratteristiche di base e fondamentali di un software per vj.

Questo passaggio è reso possibile dalla conversione in digitale delle sorgenti video siano esse convertite in real time da schede di acquisizione video o in precedenza salvate in file. Una volta creato il file video le informazioni dei singoli fotogrammi sono contenute al suo interno sotto forma di matrici numeriche. Il numero di queste matrici dipende dalla tipologia di codec video utilizzato per la compressione del file e il suo ruolo è molto importante per quanto riguarda l'utilizzo di processore del computer. Un codec non è altro che una tipologia di compressione dati grazie alla quale è possibile salvare su file grandi quantità di informazioni utilizzando meno spazio in memoria fissa. Le caratteristiche del codec che interessano la performance video sono soprattutto la velocità di decompressione dei dati e

quindi l'utilizzo di CPU. L'utilizzo di file non compressi non è una soluzione perché oltre ad occupare molto più spazio si impiega molto più tempo a leggerli a causa della velocità in lettura da hard disk. L'ideale è un codec che effettui una buona compressione lossless in modo da dover leggere meno informazioni da HD e che sia di facile decompressione per il CPU. La soluzione più performante è un particolare codec chiamato DXV creato dai programmatori del software Resolume che utilizza il processore grafico della scheda video del computer (GPU) per decomprimere i frame diminuendo di molto il carico del CPU. Questo particolare codec appartiene alle librerie di Quick Time ed è dunque utilizzabile sia su Windows che su Mac ma necessita di una scheda video con GPU quindi non è compatibile con le schede integrate prive di acceleratori grafici.

Una volta decompresso il file video si presenta sotto forma di matrici numeriche di dimensione variabile in base alla risoluzione del video. Per esempio le matrici di un video con risoluzione 640x480 avranno 640 colonne e 480 righe. Il numero di matrici dipende dal colorspace, uno dei più utilizzati è l'argb che utilizza 4 matrici: la prima è la matrice alfa che determina la trasparenza del colore, la seconda è la red che determina la quantità di rosso, la terza green per i verdi e l'ultima blue per i blu. Ogni pixel del video è il risultato della combinazione delle celle che lo rappresentano in queste quattro matrici.

Si analizzeranno ora le funzioni principali di un software di oggi.

Tra queste sicuramente la possibilità di riprodurre più file multimediali contemporaneamente (file di video, immagini, vettoriali, audio). Questi file verranno chiamati clip multimediali, di ognuno di questi è possibile controllare diverse impostazioni come velocità e direzione di riproduzione, risoluzione in pixel della grandezza dei fotogrammi immagine da modificare, automatizzazioni di timeline o controlli esterni (midi, keyboard) per selezionare quando mandare in esecuzione il clip. In molti software per vjing

l'organizzazione dei diversi clip è organizzata in livelli. Un livello può essere considerato una traccia video quindi un gruppo o contenitore di più clip. Il numero di livelli o tracce video utilizzabili contemporaneamente varia di software in software, anche se, spesso sono i limiti hardware in capacità di calcolo dei computer a limitarne il numero. In ogni livello è possibile poi inserire una serie di plug-in per l'elaborazione in tempo reale della matrice video. Questi plug-in effettuano un calcolo matriciale per ogni frame del video attraverso funzioni matematiche più o meno complesse. Per esempio la moltiplicazione di ogni cella delle matrici utilizzando il colorspace argb per un valore variabile tra 0 e 1 ne determinerà una modifica della luminosità del video. Infatti in argb il colore nero è rappresentato con i valori 1 0 0 0 e il bianco con 1 1 1 1 (rispettivamente alfa red green e blue) mentre tutti i grigi si ottengono con i valori intermedi a questi due estremi, moltiplicando le matrici video per il bianco (quindi per 1) si ottiene il video originale mentre per valori minori si scurisce sino a diventare tutto nero (moltiplicazione per 0). Utilizzando funzioni più complesse si possono creare una vasta gamma di effetti video differenti:

1. Controllo su luminosità, contrasto e saturazione
2. Controlli di zoom (ingrandimento/diminuzione)
3. Controlli sui colori (bilanciamento colori, cambio colorspace, tinta)
4. Algoritmi di sintesi che utilizzano la matrice in input come variabile per generare nuovi pixel.

Una volta processato il livello nel modo desiderato, un software per vjing deve permettere il mixaggio dei diversi livelli in uno o più output video. Esistono molti metodi per mixare in digitale due o più matrici video in un'unica matrice sempre attraverso plug-in con algoritmi di tipo matematico. Qualsiasi operazione matematica può essere usata per unire più matrici come ad esempio l'addizione, la sottrazione, la moltiplicazione o

divisione, ma anche funzioni più complesse come maggiore minore e seno coseno tangente. Il valore argb di ogni singolo pixel in uscita è ottenuto dall'operazione matematica desiderata tra il corrispondente pixel nella prima matrice con quello nella seconda.

Oltre a queste operazioni matematiche esistono funzioni più complesse per il mixaggio di video che sono chiamate operazioni di keying il cui scopo principale è la sostituzione di pixel del primo video, aventi determinate caratteristiche, con i corrispondenti del secondo. Esistono due tipologie di keying: il Chromakey e il Lumakey. Il primo agisce sul colore del video, una volta impostati il colore da rimuovere e la tolleranza di scostamento da questo valore, rende i pixel che rientrano in questo range trasparenti andando a modificare la matrice alfa. Diventati trasparenti questi pixel sono poi sostituiti con i corrispondenti in coordinate x e y del secondo video. In modo del tutto simile il lumakey agisce sulla luminosità del video. Impostando un valore soglia di luminosità chiamato appunto threshold e la relativa tolleranza di scostamento da questo valore, questo plug-in rende trasparenti quei pixel che superano la soglia stabilita. Entrambi i keying possono essere utilizzati in modo inverso, mantenendo quindi quei pixel che rispecchiano le caratteristiche di colore o luminosità ed andando ad eliminare gli altri.

Ultima ma fondamentale funzione che un software per vjing deve avere è la sincronia con l'audio che deve accompagnare. Esistono però diverse metodologie per la sincronizzazione tra video e audio. Ad esempio alcuni programmi hanno una time line interna, della quale è possibile impostare la lunghezza del loop nel quale si lavora, e la velocità di esecuzione di questo loop in bpm. Questo metodo non è tra i più efficaci in quanto è difficile sincronizzare gli attacchi dei video con gli attacchi in musica anche quando i bpm sono uguali. Altri software usano un ingresso audio dal quale ricavano i bpm della traccia attraverso particolari algoritmi

di beat decoding. Questi software hanno il vantaggio di rendere bene la sincronia quando l'audio è ricco di percussioni o comunque eventi ritmici ma in mancanza di questi hanno difficoltà nel trovare i bpm e la sincronia può mancare. Un terzo modo per sincronizzare il software è lo standard midi clock. Questo standard è utilizzato da molti anni in tutti gli strumenti elettronici (synth, drum machines ecc) e in quasi la totalità di software audio. Grazie a questo protocollo è possibile sincronizzare il software per vjing con gli strumenti o gli altri software utilizzati per l'esecuzione musicale, ottenendo così un sincronismo molto preciso.

Capitolo 2

Video for club environment

2.1 Introduzione

In questo capitolo verranno analizzati i plug-in dell'ambiente esecutivo, creati per gestire il video durante le performance live.

Come si è visto in precedenza, esistono centinaia di software dedicati al vjing spesso nati dall'esigenza, da parte degli artisti, di avere nuovi strumenti che potessero permettere la personalizzazione a piacimento della propria performance. Per queste stesse esigenze si è deciso di sviluppare un software che permettesse un ampio controllo della propria performance, ma che allo stesso tempo fosse di facile utilizzo e non rendesse necessaria la presenza di una persona per controllarlo. L'idea nasce infatti con l'intento di creare un ambiente esecutivo dove chi suona genera anche la performance video, senza che questa limiti le possibilità di suonare. L'ambiente esecutivo progettato utilizza plug-in di MaxforLive per comandare tramite OSC la patch di Max/Jitter presenti su altri computer. L'utente riesce in questo modo ad avere il controllo di più software posizionati anche in posti differenti da un'unica postazione.

2.2 Ableton Live, MaxforLive e Max/Jitter

Per la realizzazione di questo progetto si è deciso di appoggiarsi al programma Live della Ableton in quanto è molto versatile e particolarmente adatto alle esigenze. Live è un sequencer progettato per essere utilizzato come strumento per suonare dal vivo, oltre ad avere una parte più classica per produzioni in studio. All'interno di Live è possibile creare tracce audio o midi e di ognuna è possibile deciderne il routing sia in input che in output. In ogni traccia audio è possibile caricare una vasta gamma di media in diversi formati (aiff, wav, mp3, mp4, mov, flac, ecc), utilizzare gli ingressi fisici della scheda audio per elaborare il suono in tempo reale oppure ricevere l'input da un'altra traccia per continuare a processarlo. In ogni traccia midi è possibile utilizzare effetti midi, caricare strumenti software come i plug-in vst o rtas di sintesi che modificano l'output della traccia da midi ad audio e successivamente altri plug-in per l'elaborazione in tempo reale dell'audio generato. Di ogni traccia midi è possibile deciderne l'input, mentre per quanto riguarda l'output dipende dalla tipologia di plug-in utilizzati al suo interno: se si utilizzano solo plug-in midi l'output sarà midi e quindi indirizzabile su qualsiasi uscita midi o altra traccia midi, se invece vengono utilizzati plug-in strumentali che convertono le note midi in segnale audio sintetizzato da particolari algoritmi l'output sarà audio quindi indirizzabile solo su uscite ed entrate audio.

Oltre alla enorme flessibilità di routing, Live presenta grandi vantaggi negli editor audio e midi. Tutti i media, siano essi file audio o video o midi, sono visualizzati in Live come clip colorati e questi sono editabili con un semplice doppio click.

L'editor audio ha le principali funzioni di un qualsiasi editor come copia, taglia, incolla, reverse, e offre anche funzioni di pitch shift e time stretching

utilizzabili in modo molto semplice e soprattutto in realtime quindi accessibili anche durante l'esecuzione.

L'editor midi di Live grazie alla sua versatilità rappresenta il fulcro principale per la realizzazione del progetto di questa tesi. Esso permette la creazione di clip midi di qualsiasi lunghezza e l'impostazione all'interno di questo di un loop di lunghezza variabile. La scrittura delle note è realizzata in modo molto intuitivo e semplice, con diversi shortcut da tastiera che ne rendono molto veloce la modifica di lunghezza della nota, pitch e velocity. Inoltre offre la possibilità di nominare le note di una determinata traccia midi rendendo così facilmente riconoscibile la funzione di quella nota se utilizzata diversamente dalla sua normale funzione di nota musicale. La funzione di modulation permette la scrittura all'interno del clip di automazioni di qualsiasi controllo midi, sia esso mod wheel, pitch bend o qualsiasi knob di Live.

Altra caratteristica di Live che lo rende particolarmente adatto per realizzare performance audio-video è la funzione di launch dei clip siano essi audio, midi o video. Ogni clip presente nel progetto infatti ha una serie di proprietà che permettono una vasta gamma di interazioni e automazioni riguardanti relazioni tra un clip e l'altro e quantizzazione di lancio. È impostabile, in un punto di timeline all'interno del clip misurato in battute e suddivisioni, un'azione successiva inerente al lancio di un altro clip una volta raggiunto questo punto. Il sistema di quantizzazione garantisce la sincronizzazione del clip lanciato facendolo partire alla successiva divisione o suddivisione della battuta impostata rispetto al global transport. Per esempio in un tempo di 4/4 se si imposta come quantizzazione 1 bar e si lancia il clip questo entrerà in play solamente quando arriverà la nuova battuta nel global transport indipendentemente che esso sia stato lanciato nel primo, secondo, terzo, o ultimo quarto di battuta.

L'utilizzo di Live come sequencer base per la realizzazione di questo ambiente esecutivo, oltre alle qualità evidenziate precedentemente, è dovuto alla collaborazione tra la software house che produce Live (Ableton) e la Cycling74 che produce Max/MSP Jitter. Dalla versione 8.0 di Live infatti è disponibile una particolare libreria di plug-in utilizzabili al suo interno chiamata MaxforLive. Questa libreria permette di creare un plug-in in una qualsiasi traccia di Live ed editarlo in Max/MSP. Esistono tre tipologie di plug-in MaxforLive:

- Max midi effect
- Max instrument
- Max audio effect

Il midi effect è un plug-in ideato per elaborare messaggi midi e restituirli alterati a Live. L'interfacciamento con Max/MSP avviene tramite gli oggetti midiin e midiout che rappresentano le entrate e le uscite del plug in e sono necessari per la comunicazione tra i due programmi.

L'instrument si interfaccia con un midiin in entrata che trasmette i messaggi midi da Live a Max/MSP ed un pluginout~ in uscita che restituisce a Live l'audio elaborato in Max. Questo plug-in permette la realizzazione di strumenti software come sintetizzatori o campionatori programmabili in Max/MSP.

L'ultima tipologia di plug-in di MaxforLive è l'audio effect che si interfaccia con un oggetto plugin~ in input, dal quale riceve il segnale audio da Live, e un oggetto pluginout~ tramite il quale lo restituisce. Con questo tipo di plug-in si possono costruire effetti audio per elaborare in realtime una qualsiasi traccia di Live.

Le differenze di questi tre plug-in sono, come appena visto, unicamente nell'interfacciamento tra i due programmi quindi le possibilità di programmazione al loro interno dipendono solamente dalle entrate ed uscite del plug-in. Quasi tutti gli oggetti di Max/MSP sono di fatto disponibili

all'interno del plug-in fatta eccezione per gli oggetti che gestiscono input e output audio-midi, in quanto il plug-in lavora all'interno di Live e quindi la gestione delle periferiche audio-midi è controllata da quest'ultimo. Per scambiare informazioni tra due o più plug-in si utilizza l'oggetto `send` funzionante solo per messaggi, l'oggetto `send~` non è funzionante in MaxforLive. La comunicazione tra il plug-in all'interno di Live e altri software può avvenire tramite la connessione di rete Ethernet. Esistono diversi oggetti di Max/MSP per la gestione dei protocolli di rete. Quelli utilizzati in questo progetto sono `udpsend` e `udpreceive` i quali permettono trasmissione di dati utilizzando lo standard OSC.

Per quanto riguarda la parte video di questo progetto Live funge solamente come controller della patch di Max/Jitter che genera il visual. Tutte le operazioni di riproduzione ed elaborazione in tempo reale dei media avvengono infatti all'interno di una patch di Max/Jitter controllata tramite OSC.

Max/MSP Jitter è un programma che permette di organizzare più oggetti, principalmente scritti in C++ o Java, facendoli interagire tra loro costruendo così dei veri e propri programmi. Può essere definito come un linguaggio di programmazione a blocchi, dove ogni oggetto è un blocco di codice che viene eseguito quando l'oggetto stesso è connesso tramite `inlet` ed `outlet` ad altri. I limiti di questo programma rispetto ad un linguaggio di programmazione classico sono soprattutto lo spreco di CPU utilizzato e la difficoltà nel controllo diretto delle periferiche hardware. Il maggiore utilizzo di CPU rispetto ad un altro linguaggio di programmazione classico è dovuto alla multifunzionalità di molti oggetti in quanto quasi sempre gli oggetti che si utilizzano hanno altre funzioni rispetto a quella desiderata e questo comporta una maggiore quantità di codice al loro interno che seppur non utilizzato rallenta il processore.

La scelta di Max/MSP Jitter come software per la programmazione di questo ambiente esecutivo nonostante i difetti sopra elencati è dovuta dalla semplice interfaccia e dall'immediato interfacciamento con Live grazie ai plug-in creati in MaxforLive.

2.3 MaxforLive Plug-in

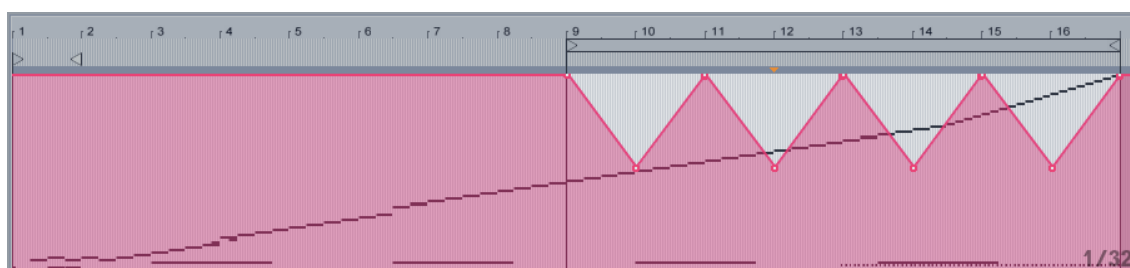
Per controllare i video nella patch stand alone di Max/Jitter vengono utilizzate tre tracce midi, una per il primo video, una per il secondo e l'ultima per gli effetti e il mixaggio dei due video. Ciascuna di queste ha diversi plug-in al suo interno che elaborano le note e le automazioni presenti all'interno dei vari clip di Live trasferendole poi alla patch di Max/Jitter tramite OSC.

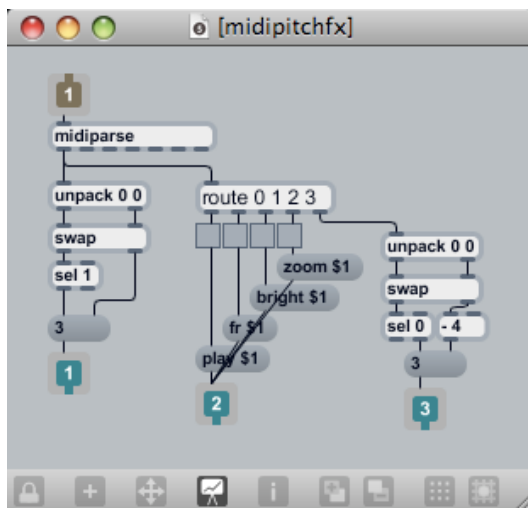
Di seguito saranno analizzati i plug-in creati per questo progetto:

- Video_Player



Il Video_Player è un plug-in che utilizza le note midi e le modulation di un clip per regolare la riproduzione di una traccia video della patch principale. Grazie a questo plug-in è possibile scegliere il video da eseguire, la posizione in frame alla quale saltare, la velocità e la direzione di riproduzione, la

[illegible][illegible]



Il midipitchfx è un oggetto realizzato con la concatenazione di diversi oggetti di max. Ha un unico ingresso dal quale riceve i messaggi midi da Live. Questo ingresso è collegato al midiparse che ha la funzione di dividere i messaggi midi nei 7 outlet che possiede. Dal primo outlet esce una lista di due numeri che rappresentano pitch e velocity di ogni nota del clip midi. Nella parte di sinistra vengono analizzate le velocity delle note e se una di queste ha valore 1 viene spedito in uscita dal primo outlet il valore di pitch di quella nota. Nella parte centrale viene effettuata una analisi riguardante il pitch delle prime 4 note midi, se una di queste viene suonata avrà sicuramente una velocity diversa da 0 e quindi accenderà il toggle corrispondente che a sua volta sostituirà il valore \$1 del messaggio a cui è collegato con il valore 1. Sullo stacco della nota si avrà una velocity 0 che andrà a spegnere il toggle prima acceso il quale restituirà in uscita il valore 0. I messaggi sono poi collegati al secondo outlet che invierà il contenuto del messaggio via OSC alla patch principale. Tutte le note diverse dalle prime quattro vengono invece processate e manderanno il proprio valore di pitch all'outlet 3 solo nel momento di attacco della nota. Questi valori saranno poi inviati alla patch principale e determineranno il frame al quale saltare all'arrivo della nota.

All'esterno del midipitchfx il primo outlet è collegato ad un oggetto tab il quale riceve il numero dell'elemento salvato al suo interno restituendo una stringa che verrà poi utilizzata dalla patch principale per leggere il file txt corrispondente contenente tutte le informazioni dell'omonimo video. Il secondo outlet, contenete i messaggi relativi alle prime 4 note, viene spedito direttamente via OSC. L'ultimo outlet viene concatenato insieme al messaggio rate e spedito a sua volta via OSC. I tre knob con i relativi messaggi spediscono il proprio valore per il controllo di zoom speed e bright. Il primo oggetto gate viene utilizzato con il relativo comando in input a sinistra per indirizzare tutti i messaggi del plug-in nei due oggetti sottostanti che aggiungono in prefisso V1 o V2 a seconda se si desidera che quel plug-in gestisca il primo o il secondo video nella patch principale. Analogamente il secondo gate indirizza i messaggi in uno dei tre send che rappresentano i tre indirizzi IP assegnabili.

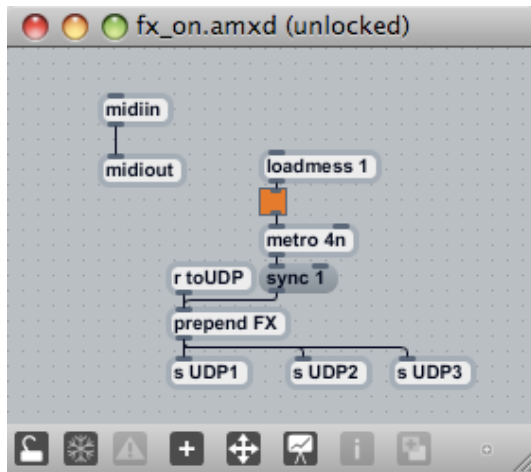
- Fx_Player



Fx_Player è un insieme di 22 plug-in di MaxforLive organizzati all'intero di un Drum Rack di Live. Il Drum Rack è uno strumento che permette internamente ad un'unica traccia midi la creazione di effetti midi ed audio differenti per ogni nota midi. Esso riceve in input le note dal clip midi e ne effettua la separazione in slot ed in ognuno di questi è possibile inserire plug-in midi e/o plug-in audio. Una volta inserito un plug-in all'interno dello slot, questo prenderà il nome del plug-in e di conseguenza anche il nome della nota nell'editor midi verrà modificato. Questo rende molto semplice la scrittura dei midi di controllo dalla patch principale perché da un feedback immediato di ciò che avviene quando si crea una nota.

I 22 plug-in interni al Drum Rack si differenziano solamente nel messaggio che mandano tramite send in output. La loro realizzazione è del tutto simile all'oggetto midipitchfx visto in precedenza. Se il plug-in è di attivazione dell'effetto, la velocity della nota collegata allo slot in cui è inserito comanderà un toggle il quale è collegato al messaggio di accensione di quel particolare effetto. Se invece è di rate, il note on ricevuto in quello slot invierà il messaggio di rate dell'effetto. Funzionano allo stesso modo i primi plug-in che gestiscono il mix aggiungendo solamente il prefisso mix al messaggio. Il primo plug-in del Drum Rack presenta due knob: V1 e V2. Questi knob gestiscono il mix se si utilizza l'operatore matematico come tecnica di mixaggio.

- Fx_On

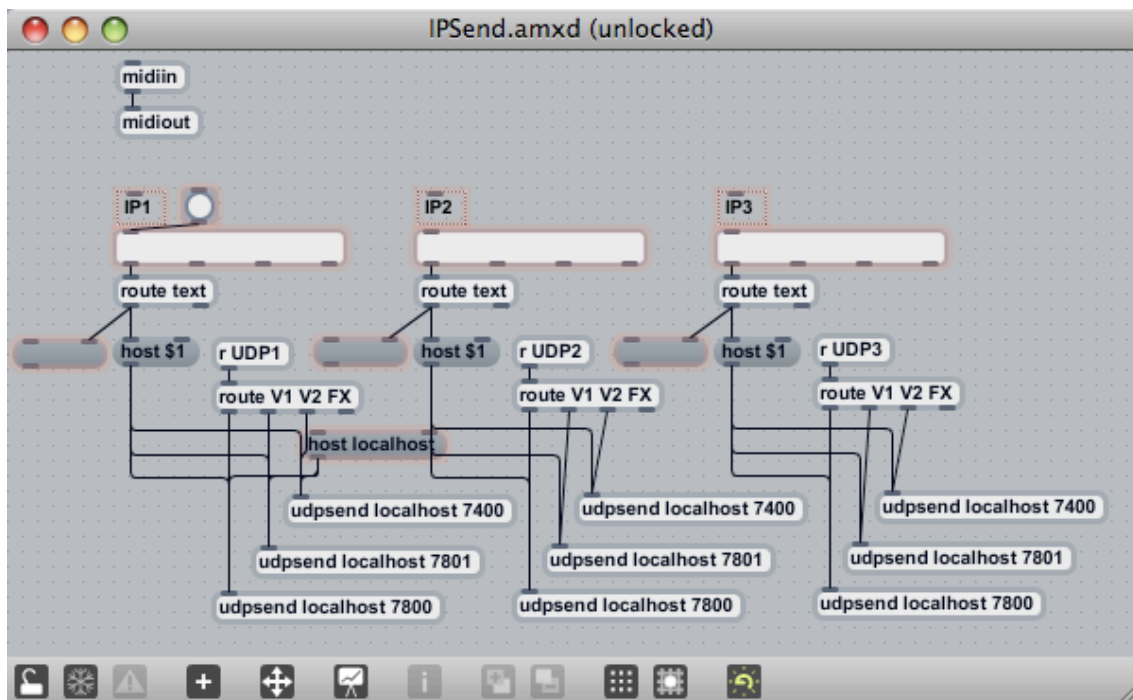


Questo plug-in riceve tutti i messaggi inviati dai 22 plug-in nel Drum Rack e li spedisce alle tre uscite udp aggiungendo il prefisso FX per identificare tutti i messaggi come messaggi effetto. Inoltre spedisce il messaggio sync 1 ogni quarto di battuta per permettere, nella patch principale, il calcolo in millisecondi dell'intervallo di tempo.

- IPSend



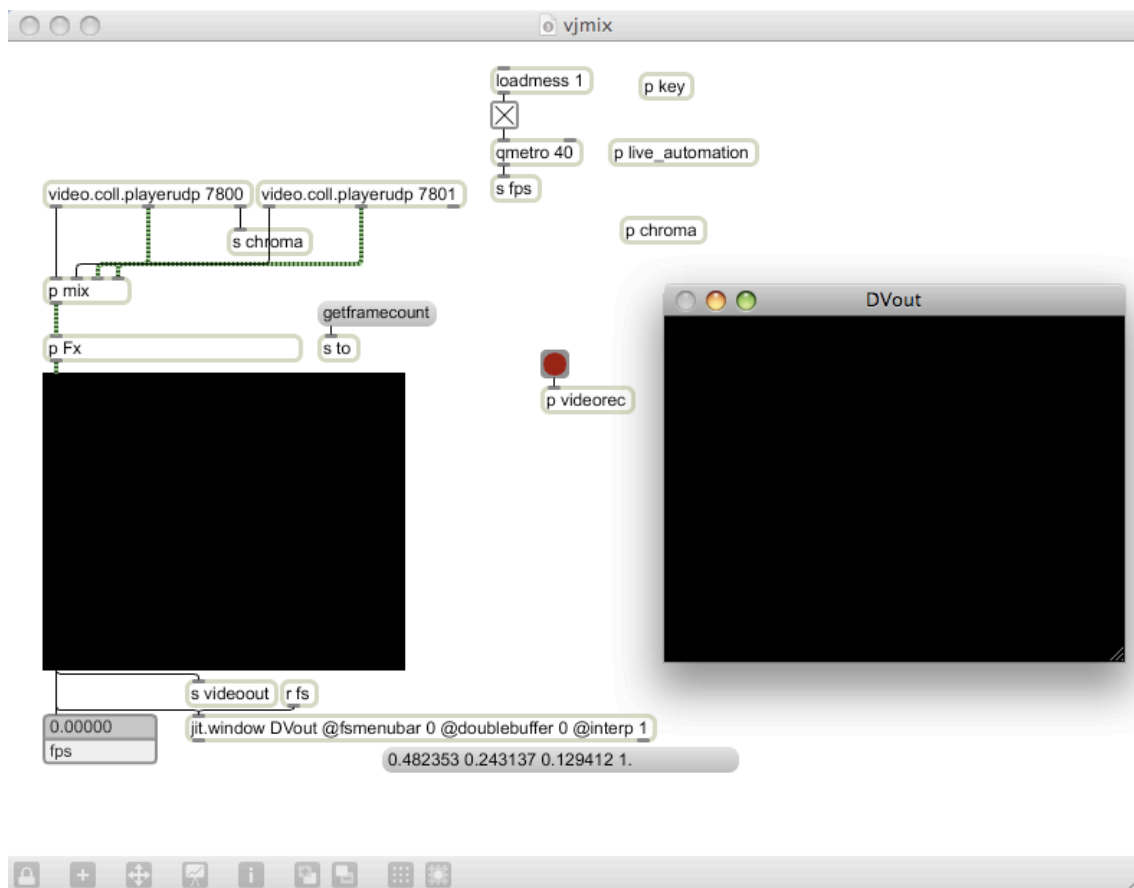
IPSend è il plug-in che invia tutte le informazioni ricevute tramite receive dagli altri plug-in via UDP alle patch di Max/MSP collegate in rete. L'interfaccia presenta tre textbox nelle quali si inseriscono gli indirizzi IP che ospitano la Patch principale.



Al suo interno il plug-in presenta un modulo di routing per indirizzare i receive UDP1 UDP2 e UDP3 all'udpsend corretto smistando i messaggi

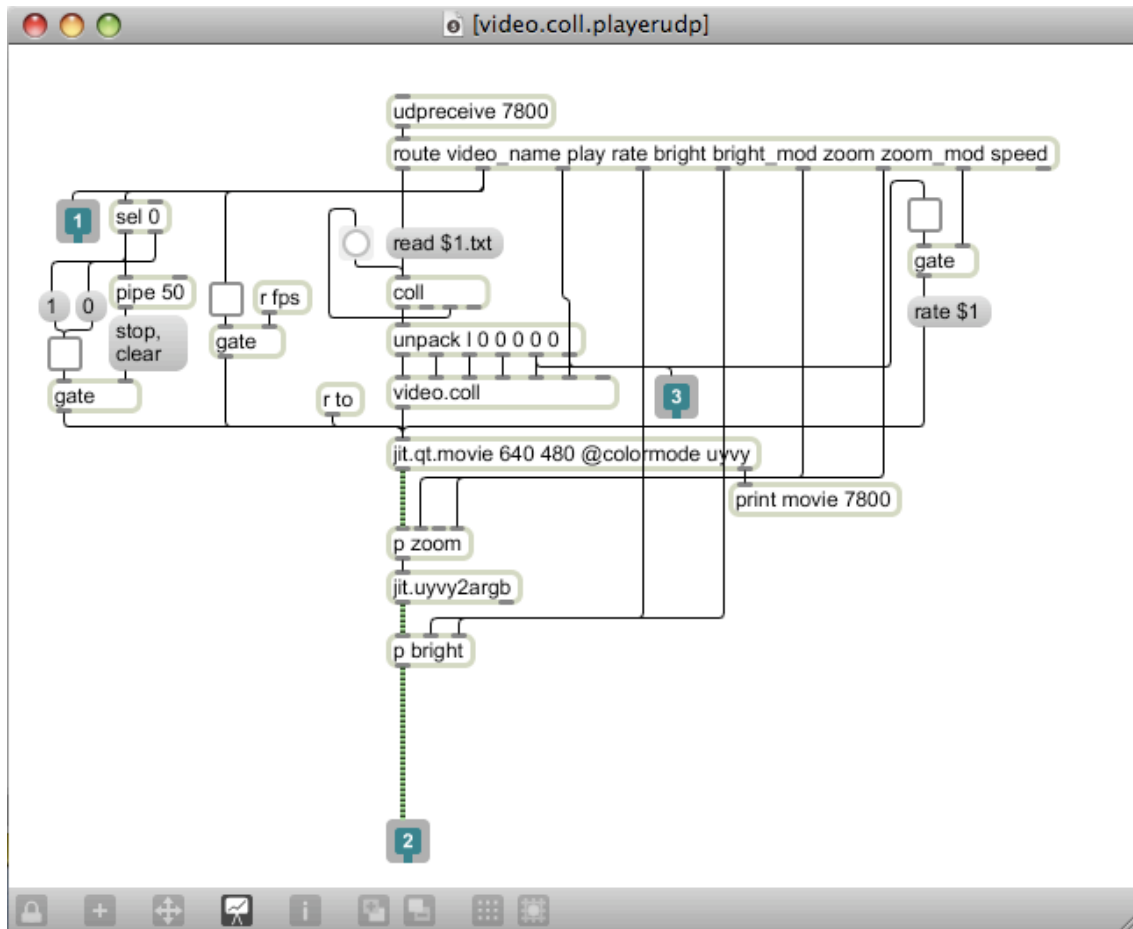
riguardanti il primo e secondo video e la catena di effetti verso porte di comunicazione differenti.

2.4 Max/Jitter patch



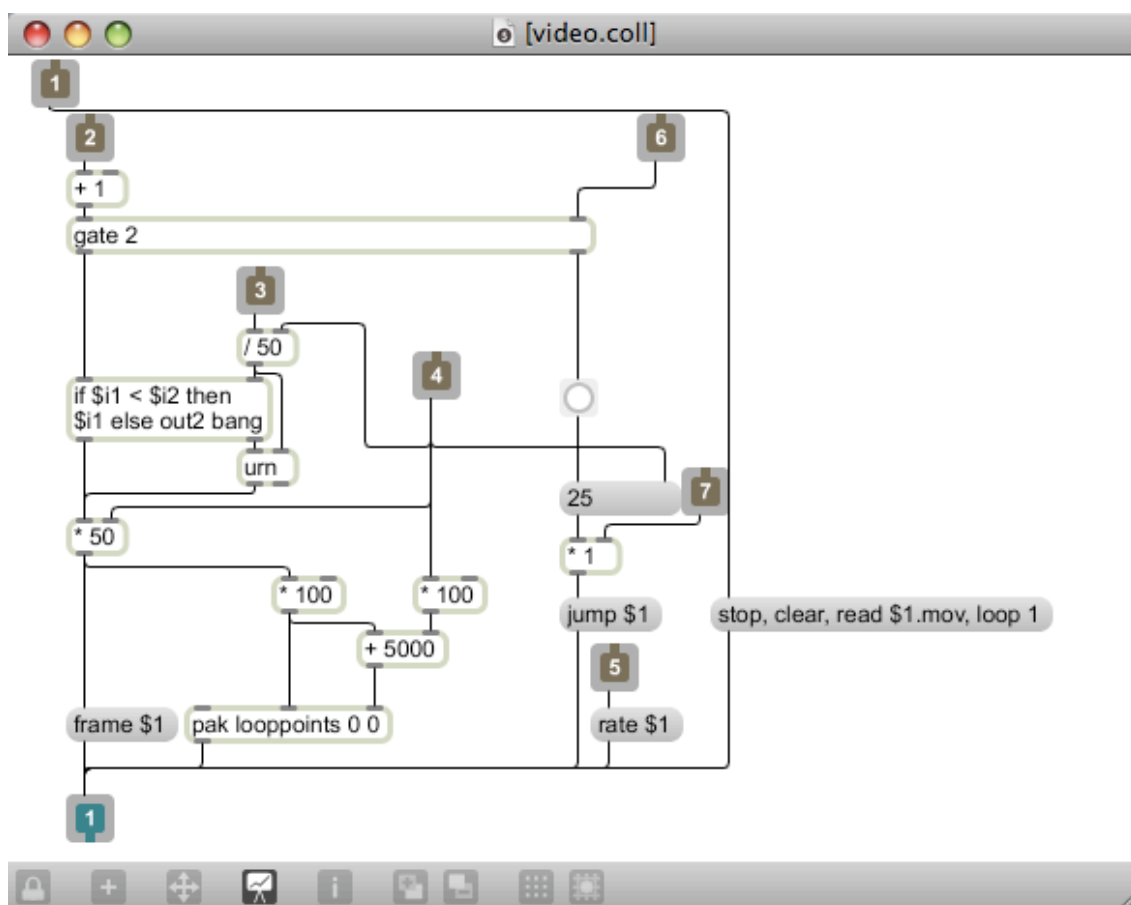
La patch principale si presenta con pochi oggetti, qualche patcher con funzione di contenitore e la quasi totale assenza di controlli. Gli unici comandi azionabili dall'interno di questa sono il full screen attivabile tramite il tasto escape della tastiera del computer, il record per registrare il video, il getframecount utile per avere il numero di frame dei video e, cliccando nella preview, si può ottenere il valore argb del pixel selezionato. La finestra DVout può essere spostata nell'uscita DVout del computer in genere collegata ad un proiettore.

Il primo oggetto che si andrà ad analizzare è il `video.coll.playerudp`:



Questo oggetto non ha inlet, in quanto riceve gli input direttamente tramite OSC dai plug-in di MaxforLive. Tutti i messaggi ricevuti vengono smistati tramite l'oggetto route, il quale indirizza agli output i messaggi che contengono il prefisso impostato come argomento. Da qui escono i messaggi riguardanti il video mandati dal plug-in video_player. Il primo è il nome del video, il secondo messaggio di play che indica se c'è un clip in esecuzione, il rate che indica lo spezzone di frame da eseguire, e dalle altre uscite arrivano i messaggi riguardanti la luminosità lo zoom e la velocità del video. Il nome del video viene utilizzato per leggere un file di testo tramite oggetto coll avente lo stesso nome del video. Questo file di testo al suo interno contiene tutte le informazioni riguardanti il video selezionato che verranno poi

inviare all'oggetto video.coll. Questo oggetto elaborerà le informazioni inviandole poi al jit.qt.movie che leggerà il video mandandone in uscita la matrice. Una volta ottenuta la matrice questa viene elaborata nei moduli successivi di zoom e bright che permettono un controllo individuale per ogni video dell'ingrandimento e della luminosità. La parte di destra di questo oggetto si occupa di controllare che ci sia un clip in esecuzione per quella traccia video. Quando il clip viene fermato la nota di pitch 0 che è assegnata al messaggio play e collegata ad un toggle manda il messaggio 0 che resetta il jit.qt.movie e blocca il flusso di messaggi a questo tramite una serie di gate. L'oggetto video.coll verrà ora analizzato più nel dettaglio per comprendere meglio il meccanismo di funzionamento base di questo progetto.



Questo oggetto presenta 7 inlet corrispondenti ai 7 valori contenuti sotto forma di lista all'interno del file .txt relativo al video selezionato. La sintassi all'interno del file .txt è di questo tipo:

```
1, (string)video_name (int)type (int)frame_lenght (int)jump_between_frames (int)speed;
```

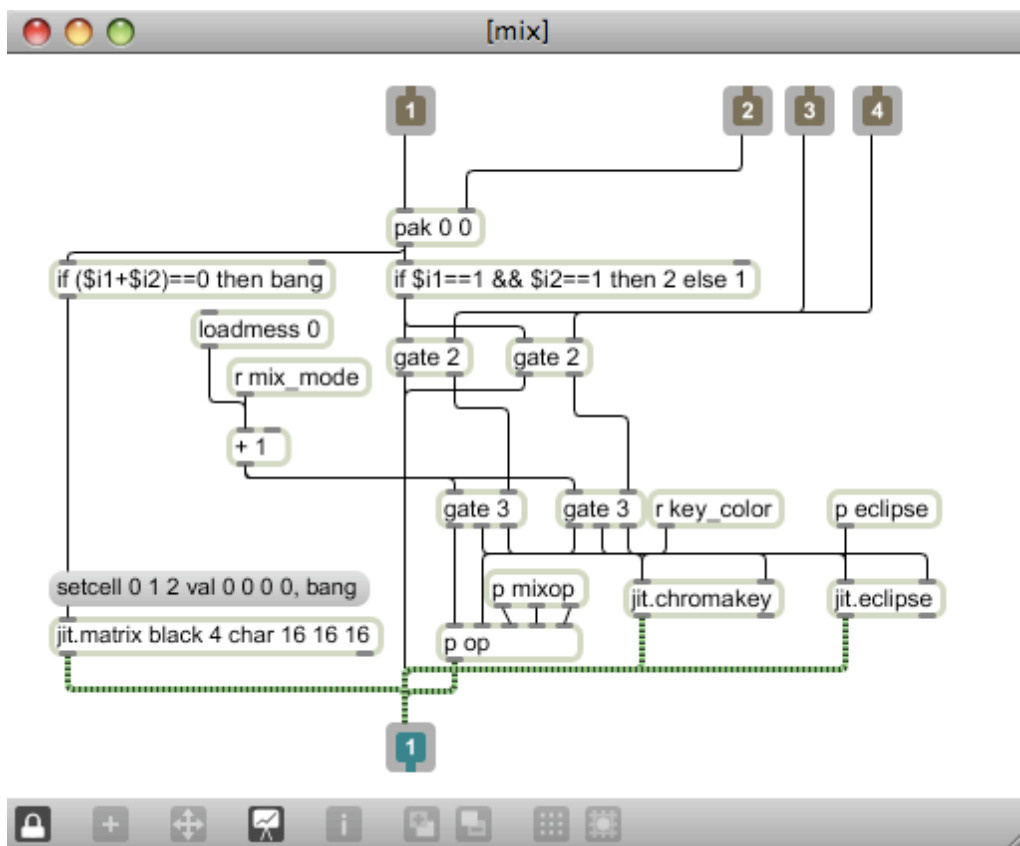
Dal primo inlet si riceve il nome del video in formato stringa che verrà spedito al jit.qt.movie insieme ai messaggi di stop, clear e loop. Il secondo inlet riceve il type con valore di 0 o 1 che verrà utilizzato per il routing dei messaggi di rate ottenuti dalle note del clip. Con il type 0 questi messaggi vengono mandati all'interno di un if che controlla il valore del rate moltiplicato per il numero di frame di ogni campione. Se questo è superiore alla lunghezza in frame del video verrà generato casualmente un valore che restituirà la posizione di un frame interno al video. Se è minore della lunghezza totale il valore di rate andrà ad impostare un loop della dimensione in frame pari al valore specificato nel file .txt e tramite il messaggio frame sposterà la riproduzione del video all'interno di questo loop. I video che utilizzano il type 0 devono essere preparati prima dell'utilizzo in quanto al loro interno dev'essere presente una struttura organizzata in una serie di spezzoni tutti di uguale durata. Il numero di frame di ogni spezzone deve essere impostato all'interno del file .txt alla voce jump_between_frames. Questo metodo di riproduzione garantisce molti vantaggi in quanto ogni filmato può essere visto come un insieme di tanti e la selezione al suo interno risulta rapida e leggera perché avviene in memoria RAM una volta che l'intero filmato è stato caricato. Il type 1 invece non richiede la preparazione del filmato a priori. In questo caso il valore di jump_between_frames verrà utilizzato per saltare avanti nel filmato ogni volta che arriva una nota di rate.

Il terzo inlet riceve la lunghezza totale in frame del video mentre il quarto il jump_between_frames. Il quinto la velocità di riproduzione utile per gestire un altro tipo di filmati dove ogni frame è un'immagine. Per utilizzare questi

filmati è necessario infatti impostare il valore di speed a 0 in modo che questo non avanzi di frame se non arriva una diversa nota di rate. In questo caso il `jump_between_frames` è impostato a 1 permettendo così la navigazione all'interno del video frame per frame dove ognuno di quest è rappresentato da un corrispondente valore di pitch della nota di rate.

Il sesto inlet infine riceve la nota di rate che verrà gestita come visto per la riproduzione del video. Il settimo inlet (scollegato in questo caso) è stato pensato per un possibile controllo direttamente dalla patch principale del salto di frame. In uscita il `video.coll` è collegato direttamente al `jit.qt.movie` al quale invia tutte le informazioni riguardanti la riproduzione.

Il `video.coll.playerudp` presenta tre outlet: il primo spedisce al patcher di mix il valore `play` che permette di sapere se quell'oggetto sta o meno riproducendo un video, il secondo spedisce la matrice video e l'ultimo il valore di `chromakey` utilizzato poi nella patch di mix.

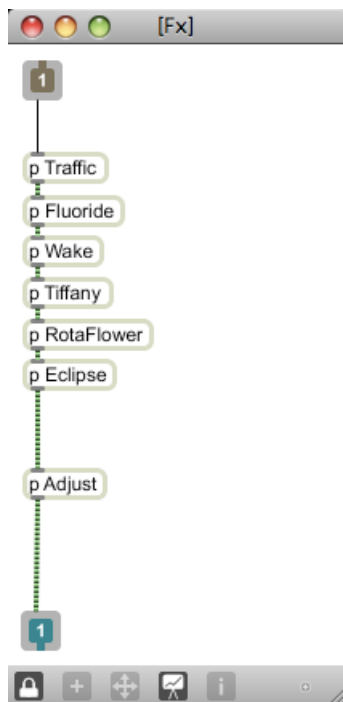


La patch di mix riceve in input i due valori di play provenienti dai 2 video.coll.playerudp nei primi due inlet e le matrici dei due video nei secondi due inlet. Viene effettuato un controllo incrociato dei messaggi di play che permette di bypassare l'intero algoritmo di mix qualora uno dei due video non sia in riproduzione. Tramite il receive mix_mode, spedito dal patcher live_automation, si seleziona la tecnica di mixaggio desiderata tra mix_op mix_eclipse e mix_croma. La prima effettua operazioni matematiche tra le matrici dei due video. La seconda è un particolare algoritmo di un oggetto di Jitter chiamato jit.eclipse che permette la duplicazione in righe e colonne del primo video, dando ad ogni cella la tinta di colore corrispondente alla stessa area di pixel del secondo video. L'ultima modalità implementata in questo progetto è l'utilizzo del chromakey: ogni video all'interno del file di testo ad esso corrispondente presenta un valore di argb che identifica il colore che si desidera eliminare. Il secondo video verrà inserito al posto di quei pixel che rispecchiano il valore argb impostato nel file .txt.

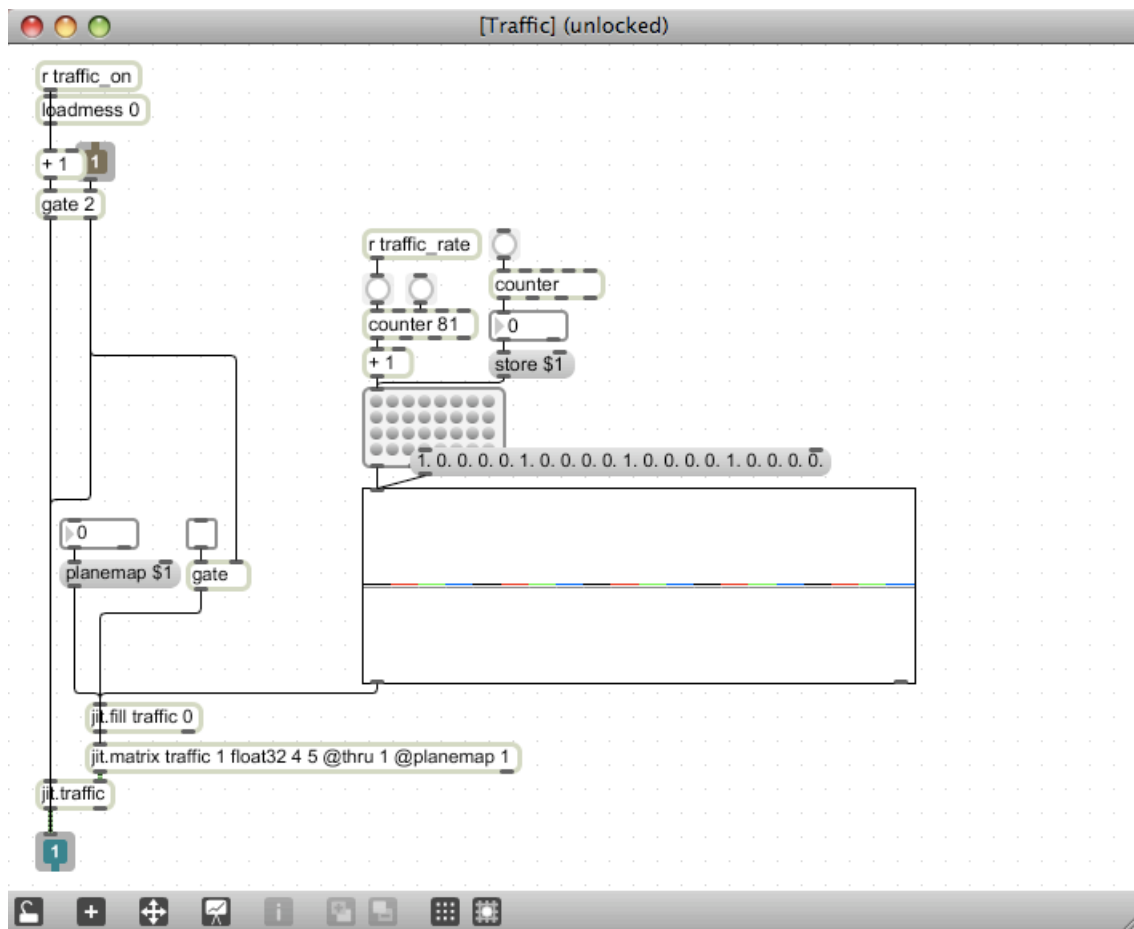
Il mix_op viene eseguito tramite l'oggetto di Jitter jit.op che riceve dal patcher live_automation le informazioni riguardanti l'operatore matematico utilizzato e il crossfade tra i due video.

Il mix_eclipse riceve in input il numero di righe e colonne da generare permettendo così la scelta di quale video rendere visibile.

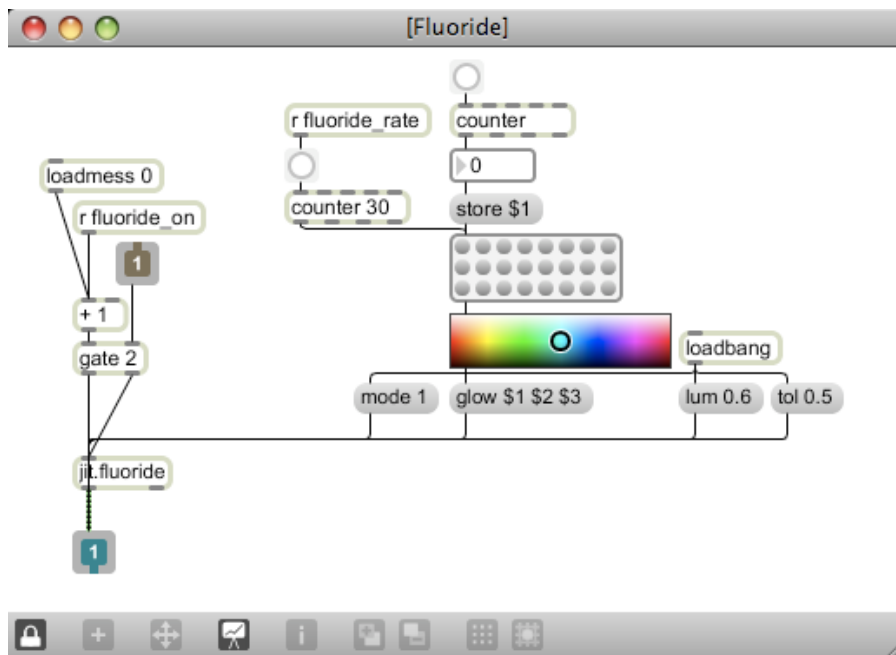
Uscita dal patcher mix la matrice video mixata entra nella catena di effetti all'interno del patcher Fx.



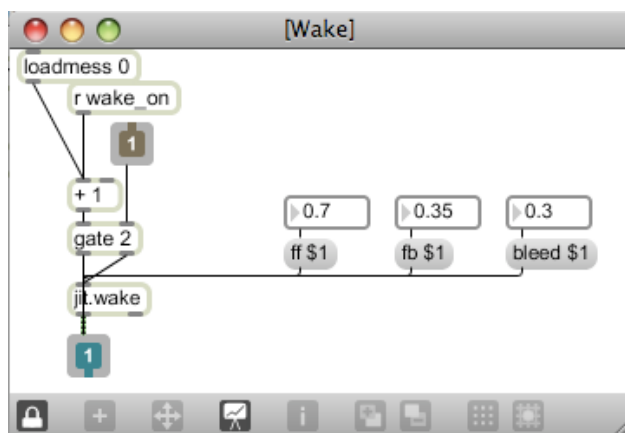
In questa sezione vengono effettuate elaborazioni in tempo reale della matrice video premixata.



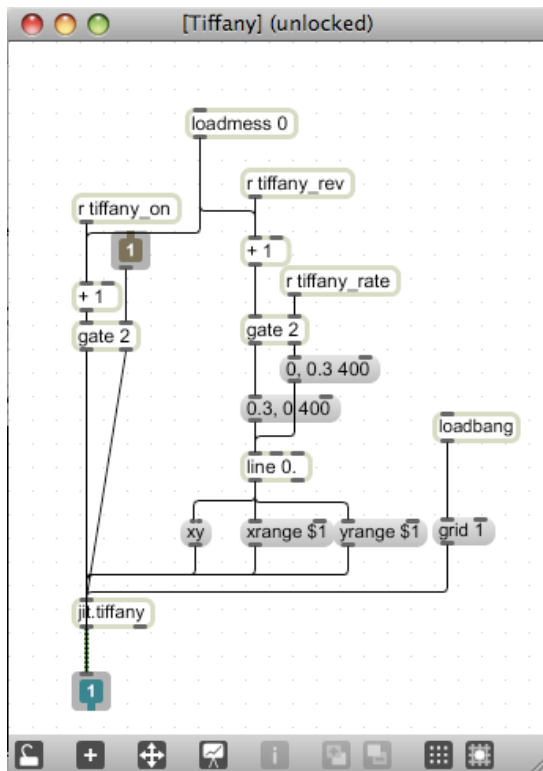
Il primo effetto è di color balance ed è realizzato dall'oggetto Jitter jit.traffic. Questo effettua un'operazione di tinta del colore nei toni scuri, medi e chiari modificando il colore di ogni range tramite una matrice contenente i coefficienti argb impostabili dall'utente. Come tutti gli effetti di questo progetto il jit.traffic riceve gli input da Live attraverso il patcher live_automation che ne comanda l'attivazione e il cambio di preset tramite le note del clip midi.



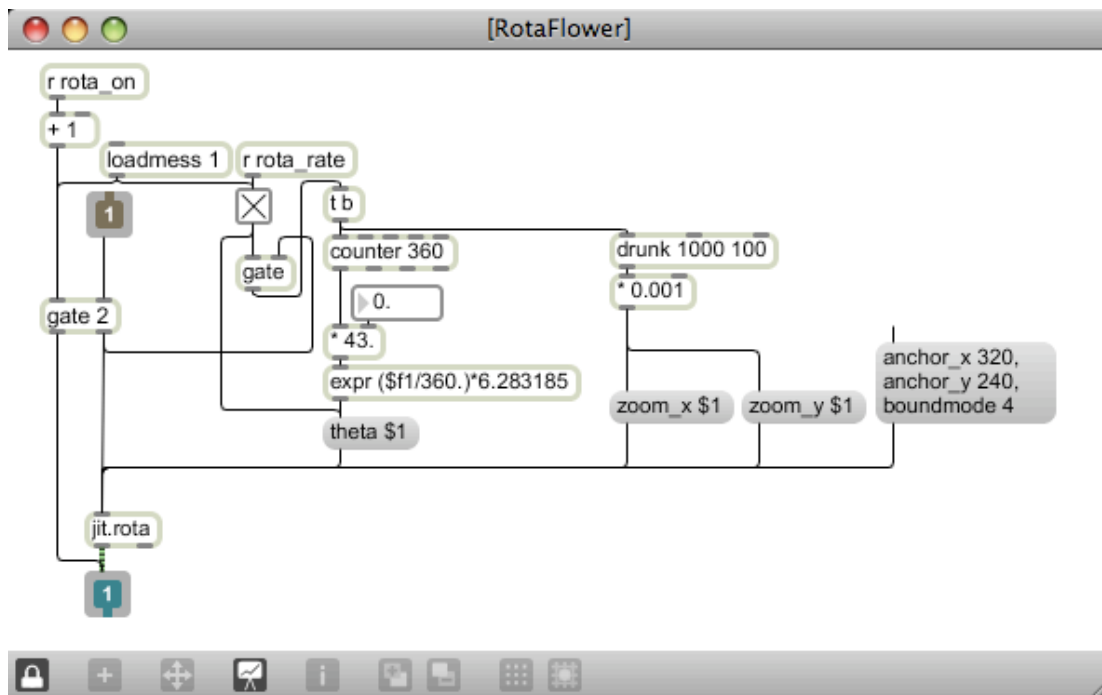
Il controllo dell'oggetto `jit.fluoride` è del tutto simile a quello del `jit.traffic`: le note midi relative a questo oggetto ne controllano l'attivazione e il cambio di preset. L'oggetto `jit.fluoride` permette la tinta in un colore desiderato di quelle aree di pixel che superano una certa soglia di luminosità.



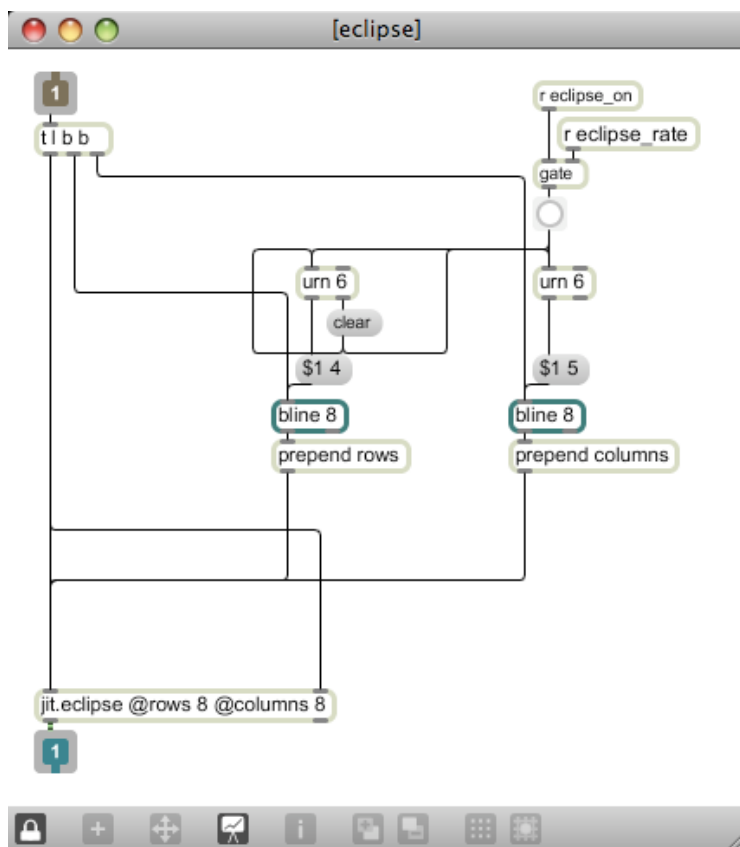
L'oggetto `jit.wake` non ha controlli al di fuori della sua attivazione. Esso effettua un delay sulla matrice in input regolabile con i parametri di flashforward, flashback e bleed che vengono inizializzati tramite `loadmess`.



L'oggetto `jit.tiffany` effettua la riduzione di risoluzione della matrice in input creando dei pixel ingranditi colorati in base alla media dei pixel originali di quell'area di video. I parametri modificabili sono l'attivazione e la grandezza dei pixel. Questa viene modificata attraverso un oggetto `line` che genera una rampa di valori compresi tra i valori di partenza e destinazione ogni volta che viene ricevuta la nota di rate relativa a questo effetto.



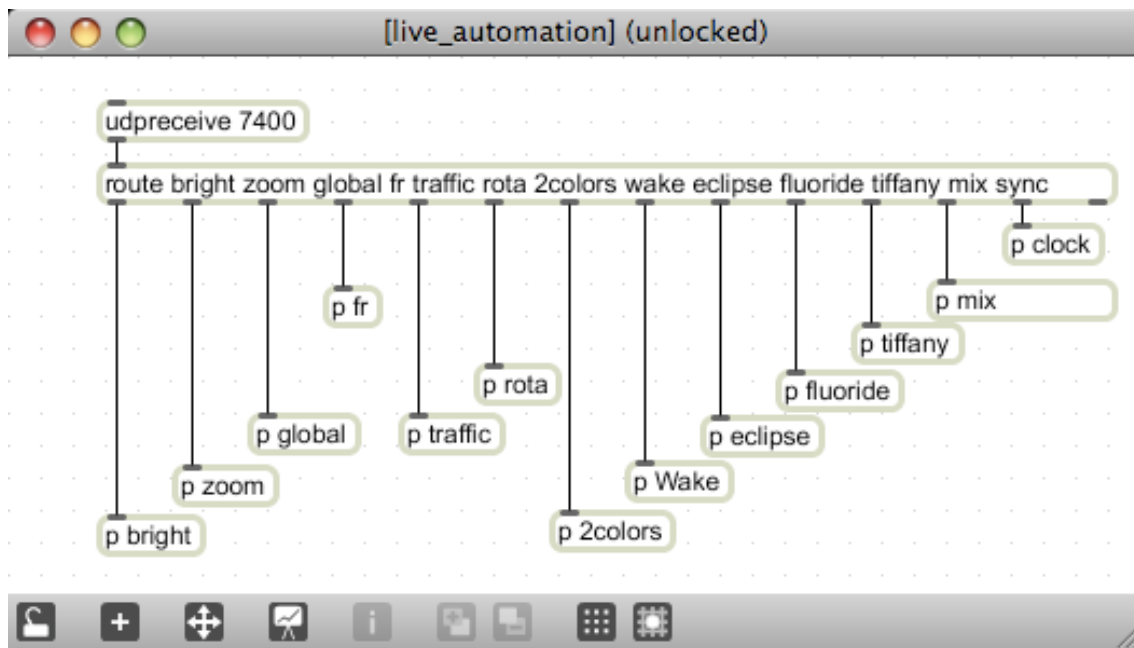
L'oggetto `jit.rota` permette una serie di operazioni differenti, in precedenza si è usato per l'effetto di ingrandimento dei due video. In questo caso viene utilizzata la combinazione di zoom e di rotazione, che è in grado di generare un effetto simile al caleidoscopio. Gli input ricevuti da live sono l'attivazione e il cambio preset.



L'oggetto `jit.eclipse` già utilizzato in precedenza nella patch di mix, viene utilizzato ora con la semplice funzione di moltiplicazione in righe e colonne dello stesso video. I parametri controllabili sono l'attivazione e il cambio di numero di righe e colonne generato random ogni volta che viene ricevuta una nota di rate.

L'ultimo effetto della catena è il `jit.brcosa` posto all'interno del patcher Adjust. Questo permette la semplice regolazione finale di luminosità contrasto e saturazione. È pensato per essere regolato all'inizio della performance e una volta regolato non ha bisogno di controlli esterni.

I parametri di controllo di tutta la catena di mix ed effetti vengono ricevuti nella patch principale all'interno del patche `live_automation`.



In questo patcher, i messaggi inviati dai plug-in della traccia di effetti da Live, vengono smistati nei diversi patcher che tramite send invieranno i controlli agli effetti e al mix.

Capitolo 3

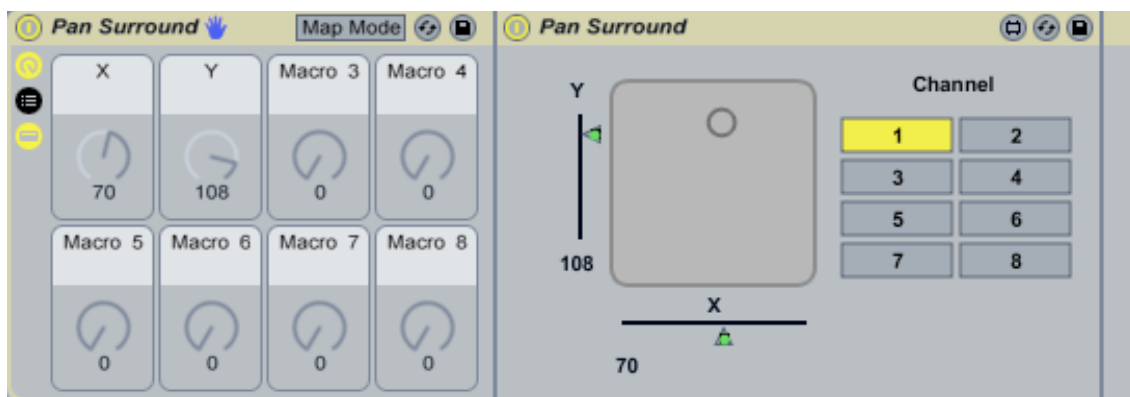
Audio control surfaces

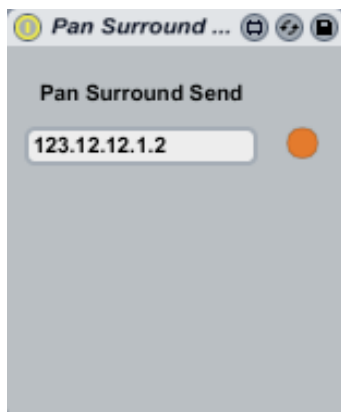
3.1 Introduzione

In questo capitolo verranno trattati i plug-in realizzati per ottenere un maggior controllo ed ampliare le possibilità di esecuzione durante una performance live. Questi plug-in sono stati progettati per sopperire ai limiti di Live riguardo la gestione di un audio surround, funzione non presente in Live, e per aumentarne le possibilità di modulazione in tempo reale di sorgenti sonore. Per la gestione del panorama surround si è deciso di appoggiarsi al mixer digitale Yamaha DM1000 che già implementa al suo interno diverse tipologie di surround.

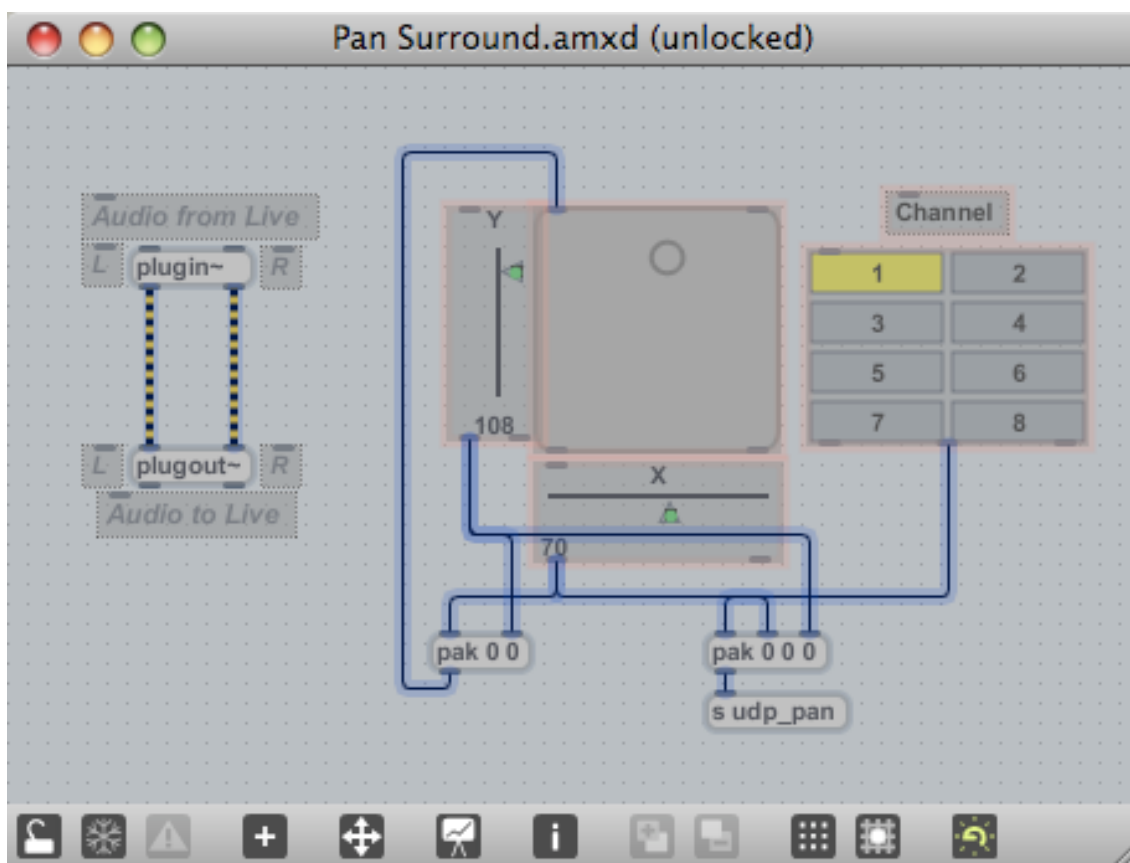
3.2 Pan Surround in MaxforLive

Per la gestione del pan surround del DM1000 sono stati realizzati due plug-in di MaxforLive e una patch di Max/MSP. I Plug-in di MaxforLive sono il Pan Surround e il Pan Surround Send i quali comunicano tra loro tramite oggetti send e receive di Max.



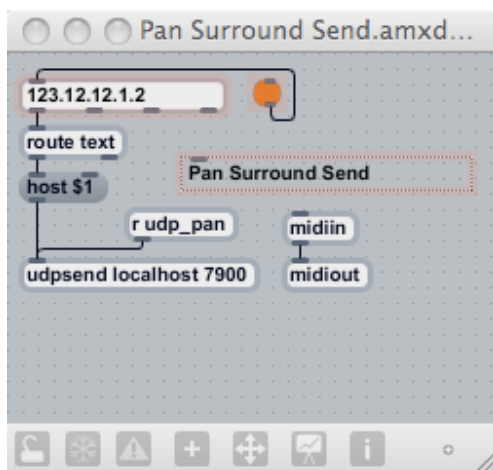


Il primo dev'essere posizionato all'interno della traccia che si desidera spazializzare, in questo modo sarà anche possibile disegnare all'interno di un clip presente in quella traccia la modulazione di pan voluta.

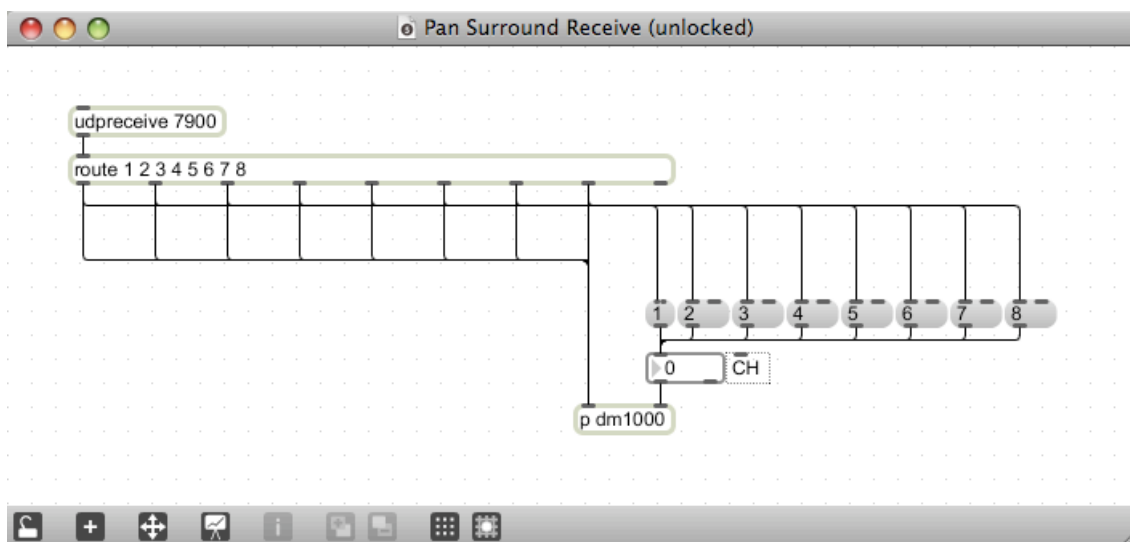


Il funzionamento di questo plug-in è molto semplice: l'audio non subisce elaborazioni mentre i valori dei due slider X e Y e l'indice del tab channel,

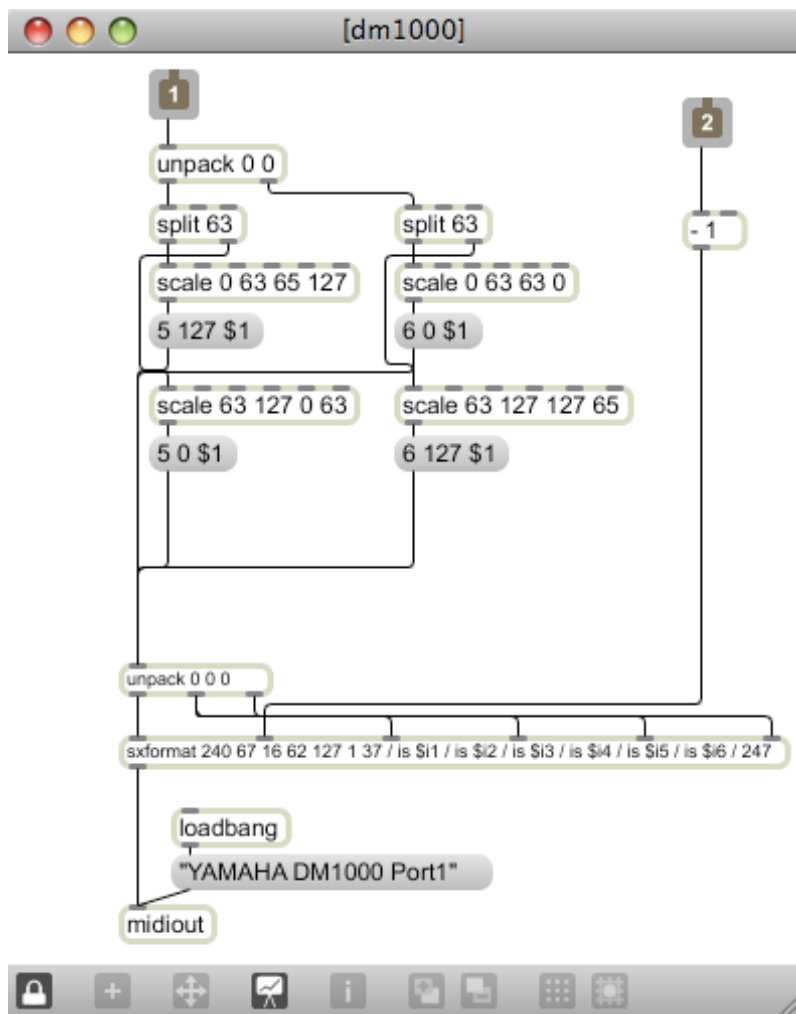
vengono inseriti in una lista spedita poi al plug-in Pan Surround Send. L'audio della traccia deve poi essere indirizzato ad un'uscita fisica della scheda audio e collegata al canale del mixer selezionato nel tab channel.



Ricevuta la lista il plug-in Pan Surround Send la spedisce tramite OSC alla patch di Max/MSP. L'unico controllo utilizzabile dall'utente in questo plug-in è la text box che permette l'inserimento dell'indirizzo IP di destinazione.

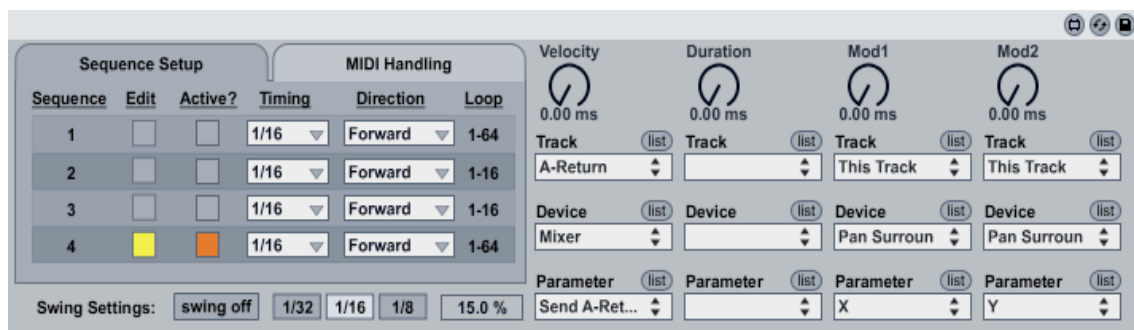
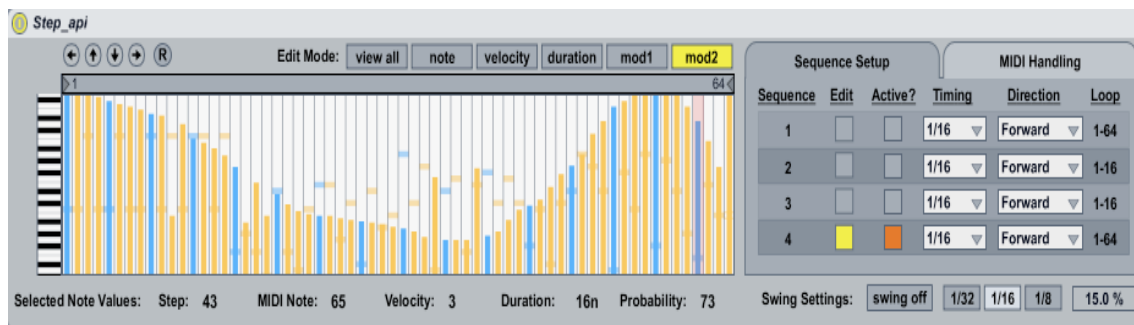


La patch Pan Surround Receive riceve la lista via OSC e ne utilizza i valori per controllare tramite messaggi midi di sistema esclusivo il DM1000.



I messaggi di sistema esclusivo del DM1000 sono generati all'interno del patcher dm1000. Tramite gli oggetti split e scale viene impostato il messaggio spedito al mixer dall'oggetto sxformat.

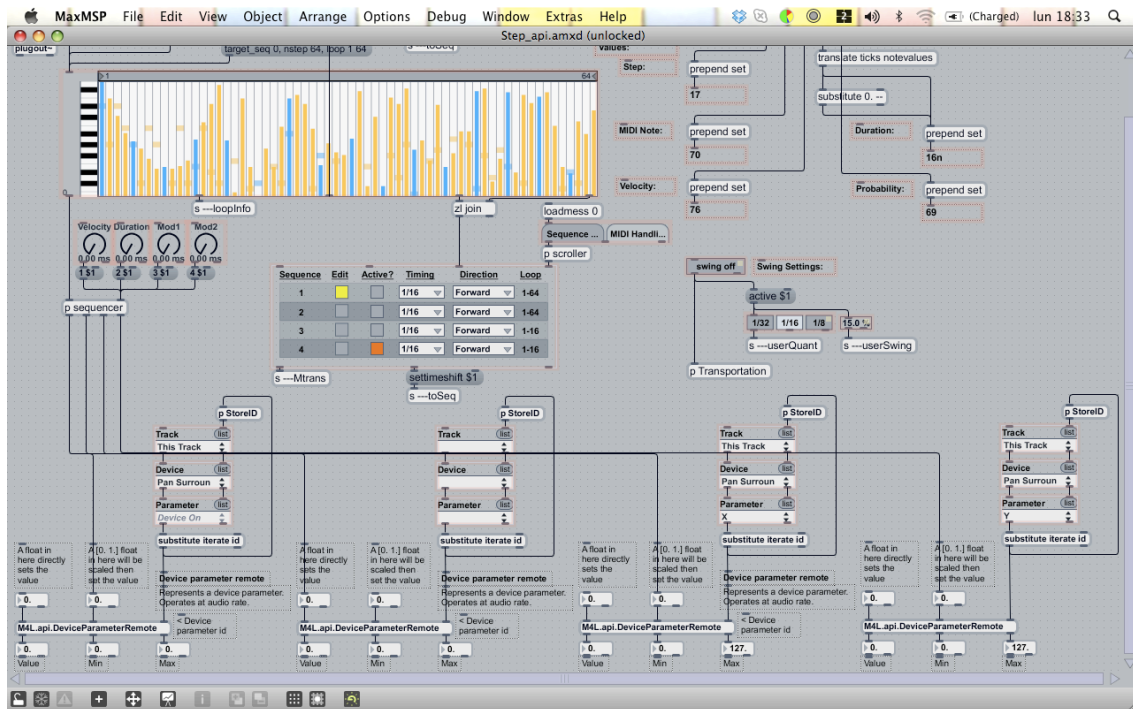
3.3 Step_api



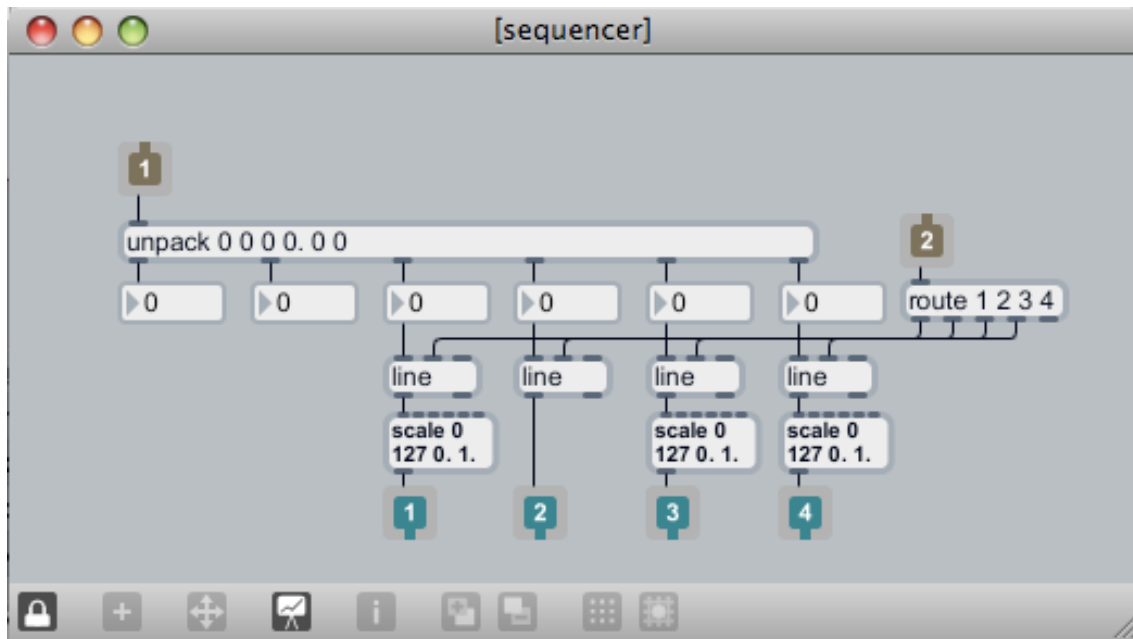
Lo Step_api è un plug-in di MaxforLive ideato per controllare qualsiasi parametro di Live o un qualsiasi parametro di un device di Live. È derivato dallo Step Sequencer midi presente nella libreria di MaxforLive di Live del quale utilizza tutte le funzioni di editing dei quattro sequencers presenti al suo interno. Nella parte di sinistra dell'interfaccia è infatti possibile effettuare modifiche al sequencer riguardo pitch delle note, velocity, duration e due ulteriori parametri chiamati Extra1 ed Extra2. I bottoni in alto a sinistra permettono di spostare tutti il contenuto del sequencer nelle direzioni indicate dalle frecce. La barra di colore grigio sopra le note identifica i punti di loop misurati in step e rende modificabile sia il punto di partenza del loo sia quello di arrivo. Sopra questa c'è un oggetto tab che permette la selezione del parametro di sequencing da editare. Nella parte centrale si trovano i controlli riguardanti la selezione del sequencer da

editare, l'attivazione di uno o più sequencer, l'unità di misura di ogni step, la direzione di percorrenza dei vari sequencer e il controllo di swing.

La parte di sinistra invece, contiene dei menu a tendina che consentono il routing dei parametri di velocity, duration, mod1 e mod2 a qualsiasi controllo presente in Live.



Le quattro scelte di routing sono realizzate tramite particolari bpatcher di MaxforLive che utilizzando gli oggetti live.path live.observer e live.object consentono di reperire informazioni riguardo il Live set. In questo caso sono usati per fornire un id rappresentante il parametro da modificare, all'oggetto M4L.api.DeviceParameterRemote il quale ne permette la modifica. I quattro knob collegati all'inlet di sinistra del patcher sequencer vengono utilizzati per impostare il tempo di rampa tra un valore di sequencer e il successivo, permettendo così una continuità nei valori.



All'interno del patcher sequencer la lista generata in output dall'oggetto `live.stepsequencer` viene divisa e i valori relativi ad ogni parametro opportunamente scalati ed inviati nell'inlet centrale degli oggetti `M4L.api.DeviceParameterRemote` i quali ricevono gli id impostati dall'utente ed effettueranno la modulazione del parametro desiderato.

Capitolo 4

Sviluppi futuri ed altre applicazioni

4.1 Sviluppi futuri

Lo sviluppo di quest'ambiente esecutivo cominciò circa due anni fa dall'esigenza di avere un software che permettesse la sincronizzazione di immagini con la musica. In particolare si desiderava creare uno strumento utilizzabile in live che utilizzasse una videocamera in input e ne elaborasse in tempo reale il segnale per poi proiettarlo dietro il palco ad uso di scenografia. L'idea era quella di rendere più partecipe il pubblico facendolo entrare a far parte dell'evento stesso a cui esso partecipava non solo in veste di spettatore. Ad oggi si sono sviluppate moltissime funzioni non preventivate la cui necessità è emersa nel corso della realizzazione. Molte altre funzioni non ancora sviluppate sono state prese in considerazione per eventuali sviluppi futuri. Di seguito un rapido elenco di funzioni non ancora realizzate che verranno implementate in futuro:

- Interfaccia e plug-in di controllo DMX512

Sono già presenti in Max/Jitter oggetti dedicati al controllo del protocollo DMX512 standard utilizzato per il controllo di luci di scena. L'utilizzo di plug-in di MaxforLive, simili a quelli creati per controllare l'ambiente esecutivo di questo progetto, applicati al controllo delle luci aumenterebbe le possibilità coreografiche di questo progetto.

- Render grafici per sintesi video

Le librerie grafiche OpenGL sono librerie open source che offrono una vasta quantità di funzioni riguardanti la generazione video. Sono crossplatform e compatibili con la quasi totalità di software dedicati alle performance video. Alcune di queste funzioni sono già state utilizzate all'interno di questo progetto nella sezione di effetti, esistono infatti molti oggetti di Jitter che si appoggiano a queste librerie per generare video. Sviluppi futuri potranno implementare ulteriori funzioni e la realizzazione di un editor per gestire gli algoritmi di sintesi. La realizzazione dell'editor all'interno di questo progetto sarebbe poi molto semplice da interfacciare a Live il quale garantirebbe un efficace metodo per inviare le variabili necessarie alla sintesi video in sincronia con l'audio.

- Plug-in interattivi utilizzabili dal pubblico

L'utilizzo di videocamere controllate dall'interno dell'ambiente esecutivo e dal pubblico permetterebbe la creazione di postazioni all'interno delle quali gli utenti potrebbero interagire con la coreografia.

L'attuale potenza di calcolo dei computer non è più un limite per la creatività degli artisti. Questo progetto dimostra che i limiti hardware possono essere superati suddividendo il lavoro tra più computer che, una volta collegati in rete, possono essere controllati remotamente da uno solo. Ecco che molti limiti imposti dalla tecnologia negli anni scorsi sono da qualche tempo ovviabili, disponibilità di attrezzature permettendo.

In conclusione questo progetto è da considerarsi un work in progress e le quasi infinite possibilità di evoluzione fanno sì che resterà tale finché si desidererà portarne avanti lo sviluppo.

4.2 Altre applicazioni

Sin dalla sua ideazione, si è deciso di sviluppare questo software in modo da renderlo il più versatile possibile. Infatti esso nasce con l'intento di unire più forme artistiche in un unico contesto di performance. All'interno delle arti performative trova quindi il suo habitat naturale e può, tramite un opportuno utilizzo, essere applicato a tutte queste. Per sua natura questo progetto aumenta le proprie potenzialità nell'interazione con il pubblico, trovando così diverse possibilità di utilizzo nell'ambito delle installazioni interattive, negli happening e nell'interactcion desing.

Bibliografia

- VJ Theory (2008). "VJam Theory: Collective Writings on Realtime Visual Performance". Falmouth:realtime Books.
- Faulkner, Michael / D-Fuse (ed), 2006. VJ: Audio-Visual Art and VJ Culture. London: Laurence King.
- Spinrad, Paul (2005). The VJ Book: Inspirations and practical Advice for Live Visuals Performance. Los Angeles: Feral House.

Sitografia

- www.vjingit.it
- www.vjcentral.it
- www.ableton.com
- www.cycling74.com
- www.vjforum.com
- www.vimeo.com/groups/vjing
- www.resolume.com
- <http://en.wikipedia.org/wiki/VJing>