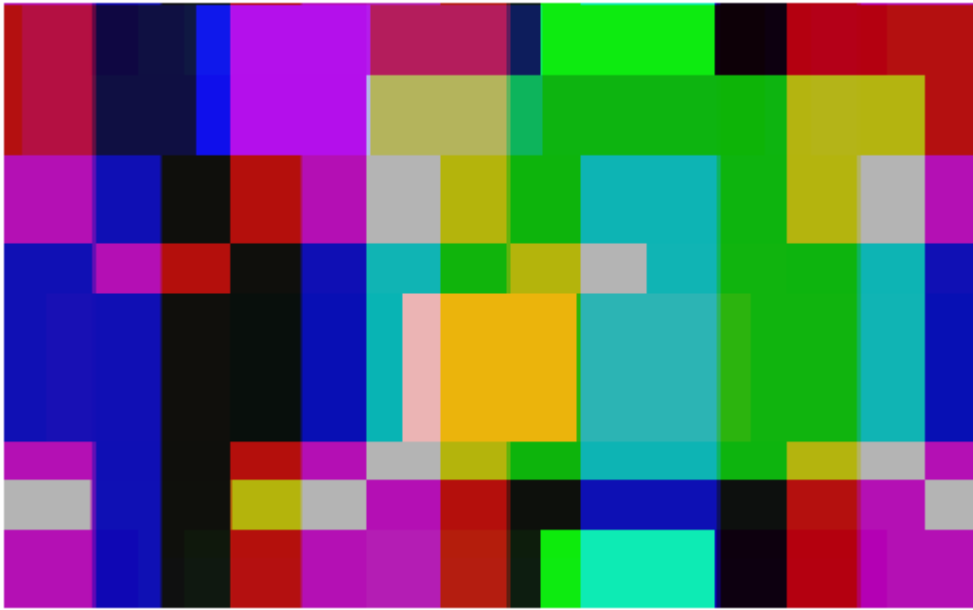


Δημιουργία οπτικοακουστικών διαδραστικών  
εγκαταστάσεων με χρήση MAX/MSP jitter.



διπλωματική εργασία  
Μπούκης Βασίλης

Επιβλέπων καθηγητής: Διαμαντόπουλος Ταξιάρχης

Ρέθυμνο 2008

Τμήμα Μουσικής Τεχνολογίας & Ακουστικής - ΤΕΙ Κρήτης

# ΠΕΡΙΕΧΟΜΕΝΑ

## Εισαγωγή

### Κεφάλαιο 1: Θεωρητικό πλαίσιο

<b>1.1 Γενικά στοιχεία</b> .....	<b>1</b>
1.1.1 New Media Art .....	1
1.1.2 Η διάδραση (interaction) στην τέχνη .....	2
1.1.3 Computer Vision (μηχανική όραση) .....	3
<b>1.2 Jitter-Γενικά στοιχεία</b> .....	<b>5</b>
1.2.1 Jitter και MAX/MSP .....	5
1.2.2 Ψηφιακή εικόνα -Γενικότερα .....	5
<b>1.3 Ο τρόπος λειτουργίας της ψηφιακής εικόνας στο Jitter</b> ....	<b>6</b>
1.3.1 Απο τον πίνακα στην εικόνα .....	7
1.3.2 Χρωματική αντιπροσώπευση .....	7
1.3.3 Τι είναι ένα επίπεδο; .....	8
1.3.4 Αριθμητική αντιπροσώπευση τιμών .....	9
1.3.5 Δημιουργία πίνακα Jitter .....	9
1.3.6 Αναπαράγοντας ένα video αρχείο .....	11
1.3.7 Πως επικοινωνούν τα jitter αντικείμενα .....	12
1.3.8 Δουλεύοντας με live video .....	13
1.3.9 Ο προσδιορισμός @ .....	14
1.3.10 jit.op .....	15
<b>1.4 Τεχνικές ανίχνευσης</b> .....	<b>18</b>
1.4.1 Image segmentation .....	19
1.4.2 Feature extraction .....	21
1.4.3 Shape recognition .....	22
1.4.4 Εξαγωγή πληροφοριών εικόνας .....	23

## **Κεφάλαιο 2: Πρακτικό μέρος-Υλοποίηση**

<b>2.1 Poly-reveal</b> .....	<b>26</b>
2.1.1 Περιγραφή .....	<b>26</b>
2.1.2 Αλγόριθμος ανίχνευσης .....	<b>27</b>
2.1.3 Αλγόριθμος σύνθεσης εικόνας.....	<b>28</b>
2.1.4 Αλγόριθμος σύνθεσης ήχου .....	<b>31</b>
<b>2.2 Μικρές ιστορίες πόλεων</b> .....	<b>33</b>
2.2.1 Περιγραφή .....	<b>34</b>
2.2.2 Αλγόριθμος ανίχνευσης .....	<b>36</b>
2.2.3 Διάταξη πινάκων στο χώρο .....	<b>39</b>
2.2.4 Ιεραρχία ροής δεδομένων.....	<b>40</b>
2.2.5 Υλοποίηση .....	<b>43</b>
2.2.6 Αλγόριθμος σύνθεσης ήχου .....	<b>44</b>
2.2.7 Ηχητικός σχεδιασμός .....	<b>45</b>
2.2.7.1 1 <sup>ο</sup> Επίπεδο-Εισαγωγή .....	<b>45</b>
2.2.7.2 2 <sup>ο</sup> Επίπεδο-animation .....	<b>46</b>
2.2.8 Μελλοντικές προεκτάσεις .....	<b>47</b>
<b>2.3 Διαδραστικά μπουκάλια</b> .....	<b>48</b>
2.3.1 Περιγραφή.....	<b>48</b>
2.3.2 Υλοποίηση.....	<b>49</b>
2.3.2.1 Background subtraction .....	<b>49</b>
2.3.2.2 IR .....	<b>52</b>
2.3.3 Ηχητικός σχεδιασμός-Μελλοντικές προεκτάσεις .....	<b>57</b>
<b>3 Συμπεράσματα</b>	
3.1 Μέτρηση απόδοσης μηχανήματος.....	<b>58</b>
3.2 Η ζωή δεν είναι μονόδρομος .....	<b>61</b>
<b>Επίλογος</b> .....	<b>62</b>
<b>Βιβλιογραφία</b> .....	<b>63</b>



## Εισαγωγή

Οι ψηφιακές τεχνικές και η πληροφορική τεχνολογία μεταβάλλουν τον τρόπο που η σύγχρονη κοινωνία προσεγγίζει τον υλικό χώρο, τον τρόπο που επικοινωνεί και αναπαριστά τα βιώματά της. Η "τέχνη και τεχνολογία" συζητείται όλο και περισσότερο ως ένα αυτόνομο είδος τέχνης, που χαρακτηρίζεται από τη χρήση των εργαλείων και μέσων που προσφέρει η νέα τεχνολογία.

Η παρούσα εργασία αποτελεί ένα είδος έρευνας σε πρακτικό επίπεδο, όσο και σε ένα στοιχειώδες θεωρητικό, γύρω από τη δημιουργία συστημάτων αλληλεπίδρασης ανθρώπου-υπολογιστή (HCI-human-computer interaction) και την αξιοποίησή τους στο χώρο της τέχνης. Για τη συγκεκριμένη έρευνα, δημιουργήθηκαν τρία παραδείγματα-εφαρμογές οπτικοακουστικής διαδραστικής εγκατάστασης με σκοπό να εξεταστούν, οι τεχνικές που χρησιμοποιούνται στο HCI και ειδικότερα το Computer Vision ως μέσο διάδρασης. Πιο αναλυτικά, με τη χρήση συγκεκριμένου λογισμικού (MAX/MSP) και σε συνδυασμό με τα στοιχεία του ήχου και της κινούμενης εικόνας δημιουργείται ένα διαδραστικό περιβάλλον το οποίο διαπραγματεύεται το ρόλο του θεατή-χρήστη όχι ως απλό αποδέκτη του εκάστοτε καλλιτεχνικού νοήματος αλλά ως αναπόσπαστο κομμάτι του ίδιου του έργου.

Σε αυτό το πειραματικό επίπεδο εργασίας χρησιμοποιήθηκε, για την υλοποίηση αλγορίθμων στο πρακτικό μέρος της έρευνας, το λογισμικό (software) της Cycling74 MAX/MSP σε συνδυασμό με το Jitter. Το Jitter είναι το σύνολο Max αντικειμένων για την επεξεργασία video. Επεκτείνοντας τις δυνατότητες του Max καθιστά εφικτή την επεξεργασία ήχου και video, μέσα σε ένα ενοποιημένο πλαίσιο εργασίας και προγραμματισμού. Η παραμετροποίηση του ήχου έγινε με τέτοιο τρόπο ώστε εννοιολογικά να συμβαδίζει με την αλλαγή της εικόνας. Οι χρήστες μπορούν να δουν και να ακούσουν το αποτέλεσμα της κίνησης τους.

Η προσέγγιση του αντικείμενου καθιστά την εργασία απευθυνόμενη κυρίως σε δημιουργούς που δίχως να διαθέτουν ισχυρή τεχνολογική κατάρτιση μπορούν να χρησιμοποιήσουν αυτή την τεχνολογία στη δημιουργία καλλιτεχνικού έργου. Όσον αφορά στην κατανόηση του περιβάλλοντος διάδρασης από το θεατή, επιδιώκεται η αμεσότητα ώστε η εγκατάσταση να προσφέρεται για χρήση σε οποιονδήποτε χωρίς να απαιτούνται ιδιαίτερες τεχνολογικές γνώσεις.

Ένα τέτοιο περιβάλλον διάδρασης θα μπορούσε να χρησιμοποιηθεί εκτός από το χώρο της τέχνης, στο χώρο της εκπαίδευσης (από εκπαιδευτικούς που

θέλουν π.χ. να δώσουν σε παιδιά να καταλάβουν μια μουσική έννοια), αλλά ακόμα και στο χώρο της διαφήμισης(από διαφημιστές που θέλουν να δημιουργήσουν συνθήκες εντυπωσιασμού γύρω από το προϊόν τους).

Στη συνέχεια θα εξετάσουμε κάποιες βασικές έννοιες της ψηφιακής εικόνας στα πλαίσια της λειτουργίας του Jitter και διάφορες μεθόδους ανίχνευσης της κίνησης με μια συνηθισμένη κάμερα. Στο δεύτερο κεφάλαιο θα περιγράψουμε την υλοποίηση των τριών εφαρμογών και θα αναλύσουμε τον τρόπο που χρησιμοποιήθηκαν οι διάφορες τεχνικές για τη δημιουργία της οπτικοακουστικής διαδραστικής εγκατάστασης. Πρέπει να αναφερθεί σε αυτό το σημείο πως η υλοποίηση της δευτερης εγκατάστασης «Μικρές Ιστορίες Πόλεων» είναι προϊόν συνεργασίας με την εικαστικό δημιουργό Χρηστού Ευαγγελία. Στην παρούσα εργασία θα συζητηθεί η προσωπική μου κατάβολη στο έργο που αποτελείται από τη δημιουργία του λογισμικού, το σχεδιασμό της διάδρασης και τη σύνθεση της μουσικής.

# Κεφάλαιο 1: Θεωρητικό Πλαίσιο

## 1.1 Γενικά στοιχεία

### 1.1.1 New media art

Τι είναι τα νέα μέσα? Ένας τρόπος να απαντήσει κανείς στο ερώτημα είναι να παραθέσει μία λίστα με τις κατηγορίες των μέσων διάδοσης της πληροφορίας που περιλαμβάνει η ορολογία. Internet, ιστοσελίδες, παιχνίδια υπολογιστών, cd-roms dvd και εικονική πραγματικότητα. Αποτελούν όμως αυτά το σύνολο του ορισμού new media? Συμπεριλαμβάνει για παράδειγμα μια τηλεοπτική εκπομπή που έχει μαγνητοσκοπηθεί από ψηφιακή κάμερα? Σε μία άλλη περίπτωση συμπεριλαμβάνει εκτυπωμένες εικόνες που έχουν δημιουργηθεί ή επεξεργαστεί από υπολογιστή? Ο όρος γενικότερα χαρακτηρίζεται από τη χρήση του υπολογιστή. Είναι δόκιμος όμως όταν ο υπολογιστής αποτελεί το μέσο διάδοσης της πληροφορίας ή όταν ο υπολογιστής χρησιμοποιείται στο στάδιο παραγωγής των νέων μέσων? Ο προβληματισμός ξεφεύγει του νοήματος αυτής της εργασίας. Αξίζει να σταθούμε όμως στο γεγονός ότι η πολιτισμική τεχνολογία και πληροφορική οδηγεί τόσο στην δημιουργία νέων μορφών διάδοσης του πολιτισμού, όσο και στον επαναπροσδιορισμό παλαιότερων μορφών όπως ο κινηματογράφος και η φωτογραφία.<sup>1</sup>

Στο χώρο της τέχνης η τεχνολογική ανάπτυξη πάντα εναγκαλίζονταν από τους καλλιτέχνες στα πλαίσια της αναζήτησης νέων τρόπων έκφρασης. Από τις εκτυπώσεις του Albrecht Dürer το 16ο αιώνα, μέχρι τις ιστορικές πρωτοπορίες των κινημάτων του 1960 (fluxus, happenings, performance art, conceptual art), οι καλλιτέχνες ενσωμάτωναν όλο και περισσότερο τις έρευνες των νέων τεχνολογιών, προάγοντας την ιδέα για "μια τέχνη που δεν θα είναι διαφορετική από τη ζωή, αλλά θα είναι μια δράση μέσα στη ζωή, θα υπόκειται στο τυχαίο και στους συνδυασμούς του, στον αυτοσχεδιασμό και στη μη προμελετημένη δράση."<sup>2</sup>

Το 1994 η εξέλιξη του internet σε δημοφιλές μέσο δημιούργησε τον καταλύτη της έξαρσης ενός παγκόσμιου κινήματος τέχνης που ξεκίνησε να εξερευνεί τις κοινωνικές, πολιτιστικές και αισθητικές δυνατότητες των νέων αυτών τεχνολογιών της επικοινωνίας όπως το internet, οι κάμερες παρακολούθησης, τα ασύρματα τηλέφωνα, οι φορητοί υπολογιστές κ.α. Η new

---

1 Γενικά για New media art, βλ. L. Manovich, *The Language of New Media*, Cambridge Mass, 2001· M. Rush, *New Media in Late 20th-Century Art*. London, Thames & Hudson, 1999.

2 C. Gintz, "Happening". Στο M. Ferrer και M.-H. Colas-Adler (Επιμ.), *Ομάδες, Κινήματα, Τάσεις της Σύγχρονης Τέχνης μετά το 1945*, Αθήνα, Εξάντας, 1999, σελ. 139.

media art περιέχει συχνά τη συνεργασία και την ελεύθερη διακίνηση ιδεών και τεχνοτροπιών και πολύ συχνά σχολιάζει τις επιπτώσεις της αύξησης της τεχνολογίας σε κοινωνικά ζητήματα όπως η ταυτότητα, ο εκφυλισμός του ιδιωτικού βίου, η παγκοσμιοποίηση κ.α.

### 1.1.2 Η διάδραση (interaction) στη τέχνη

Η διάδραση είναι μία κατάσταση ανταπόκρισης στην οποία η δράση και η αντίδραση είναι ενωμένες σε μία διμερή κατάσταση επηρεασμού. Το νόημα της λέξης είναι ότι η δράση ενός αντικειμένου προκαλεί μια παράλληλη δράση σε ένα άλλο αντικείμενο και δεν υπακούει στη λογική της δράσης-αντίδρασης. Ένα παράδειγμα διαδραστικού συστήματος μπορεί να είναι δύο άνθρωποι που κάνουν μια συζήτηση.

Η εμπειρία της τέχνης είναι πάντοτε ενεργητική και κατά μία έννοια διαδραστική αποτελούμενη από το περιβάλλον, την αντίληψη και τη δημιουργία νοήματος στο μυαλό του θεατή. Παρ'όλα αυτά, με την εξέλιξη των μεθόδων διάδρασης με υπολογιστή, ένα νέο είδος καλλιτεχνικής εμπειρίας έχει δημιουργηθεί. Σε έργα βασισμένα σε αυτή την τεχνολογία η εμπειρία δεν είναι μόνο ψυχολογική αλλά δημιουργείται μέσω υλικών ανταλλαγών που συμβαίνουν μεταξύ του θεατή και του υπολογιστή. Ο άνθρωπος και το μηχάνημα δουλεύουν μαζί σε διάλογο για να δημιουργήσουν ένα μοναδικό έργο που διαφέρει ανάλογα με τον εκάστοτε θεατή. Στις αρχές του 1970 ο Ernest Edmonds και ο Stroud Cornock εξέφρασαν μια νέα έννοια σχέσης μεταξύ του καλλιτέχνη, του θεατή και του αντικειμένου (υπολογιστής). Περιέγραψαν μία δυναμική καλλιτεχνική κατάσταση στην οποία αναφέρονται ως η μήτρα (the matrix). Όλα τα στοιχεία της μήτρας (ο θεατής, το αντικείμενο και ο καλλιτέχνης) στα οποία αναφέρονται ως το «σύστημα τέχνης» (art system) συμμετέχουν ενεργά στην κατασκευή του καλλιτεχνικού νοήματος. Το νόημα προκύπτει μέσω της διαδικασίας της ανταλλαγής και η διάδραση η ίδια είναι το καλλιτεχνικό μέσο.<sup>3</sup>

Αναφέραμε προηγουμένως ότι η διάδραση σε αυτή την εργασία επιτυγχάνεται μέσω ανίχνευσης της κίνησης από κάμερα. Αυτή η τεχνική άντλησης δεδομένων μέσω ψηφιακής εικόνας δεν είναι καινούργια στην τεχνολογία και ως γενικότερη τεχνοτροπία ονομάζεται Computer Vision. Στα ελληνικά η ορολογία έχει μεταφερθεί ως μηχανική όραση αλλά, στη συνέχεια, θα αναφερόμαστε σε αυτή απλά ως cv.

---

3 E. Edmonds, *Explorations in Art and Technology*, London, Springer, 2002.



### 1.1.3 Computer Vision (μηχανική όραση)

Το computer vision είναι η επιστήμη και η τεχνολογία των μηχανών που έχουν τη δυνατότητα οπτικής αναγνώρισης. Σαν επιστημονικός κλάδος, η μηχανική όραση ασχολείται με την τεχνολογία τεχνητών συστημάτων που λαμβάνουν πληροφορίες από εικόνες ή πολυδιάστατα στοιχεία. Οι πληροφορίες, όπως καθορίζονται από τον Shannon<sup>4</sup> είναι αυτές που επιτρέπουν τη λήψη μιας απόφασης. Δεδομένου ότι η αντίληψη μπορεί να ερμηνευτεί ως η εξαγωγή των πληροφοριών από αισθητήρια σήματα, το cv μπορεί να χαρακτηριστεί αλλιώς ως η επιστήμη των τεχνητών συστημάτων αντίληψης εικόνας ή πολυδιάστατων στοιχείων. Ως τεχνολογικός τομέας, η μηχανική όραση επιδιώκει να εφαρμόσει τις θεωρίες και τα πρότυπα της στην κατασκευή αντίστοιχων συστημάτων. Τα παραδείγματα εφαρμογής συστημάτων cv περιλαμβάνουν τα εξής συστήματα:

1. Ρομποτική (π.χ. ένα βιομηχανικό ρομπότ ή ένα αυτόνομο όχημα)
2. Ανίχνευση πληροφοριών οργάνωσης γεγονότων (π.χ. για την οπτική επιτήρηση σε μία κάμερα παρακολούθησης)
3. Οργάνωση γεγονότων (π.χ. για την ανάλυση βάσεων δεδομένων των εικόνων και ακολουθιών εικόνας)
4. Αντικείμενα διαμόρφωσης ή αλληλεπιδρόμενα περιβάλλοντα (π.χ. βιομηχανική επιθεώρηση, ιατρική ανάλυση εικόνας ή τοπογραφική διαμόρφωση)
5. Αλληλεπίδραση (π.χ. ως εισαγωγή δεδομένων σε μια συσκευή για εφαρμογές αλληλεπίδρασης ανθρώπου-υπολογιστή) HCI - human-computer interaction.

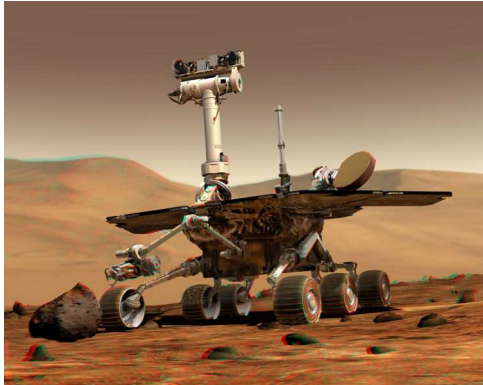
Οι υποκατηγορίες του computer vision περιλαμβάνουν την οπτική αναδημιουργία (visual reconstruction), την ανίχνευση γεγονότων (trace of events), την ανίχνευση στόχων, την αναγνώριση αντικειμένων (object recognition), την αυτοεκμάθηση υπολογιστικών συστημάτων, την κατηγοριοποίηση ακολουθιών εικόνας, την αυτόνομη κίνηση (ρομποτική) και την αποκατάσταση εικόνας (image restoration).

Ο τομέας της όρασης υπολογιστών μπορεί να χαρακτηριστεί ως πρώιμος. Αν και οι τεχνικές που έχουν ενσωματωθεί σήμερα σε αυτή την τεχνολογία υπάρχουν αρκετό καιρό, χρειάστηκε να περιμένουν την ανάλογη τεχνολογική

---

4 C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*. July and October (1948).

ανάπτυξη στον τομέα της επεξεργασίας δεδομένων (επεξεργαστική ισχύ), λόγω του μεγάλου όγκου πληροφοριών που πρέπει να διαχειριστεί ένα σύστημα για να εξάγει τα χρήσιμα δεδομένα. Η μελέτη του τομέα άνθισε στη δεκαετία του 1970 όταν μπόρεσαν οι υπολογιστές να διαχειριστούν την επεξεργασία των μεγάλων συνόλων στοιχείων όπως οι εικόνες.



*Εικ.1 ROVER ON MARS*

*Παράδειγμα ενός τηλεκατευθυνόμενου επίγειου οχήματος. Παρατηρούμε τις στερεοφωνικές κάμερες πάνω στο όχημα*

## 1.2 Jitter – Γενικά στοιχεία

### 1.2.1 Jitter και max/MSP

Το Jitter επεκτείνει το περιβάλλον γραφικού προγραμματισμού max/msp, για να υποστηρίξει το, σε πραγματικό χρόνο, χειρισμό 3d γραφικών, την επεξεργασία video και άλλων πακέτων δεδομένων μέσα από ένα ενοποιημένο τρόπο επεξεργασίας.

Επειδή το Jitter, όπως και το max/MSP, καλύπτει ένα ευρύ φάσμα οπτικο-ακουστικών εφαρμογών, προσφέρει δυνατότητες για τη δημιουργική εξερεύνηση video επεξεργασίας αλλά και ανάπτυξης διαδραστικών μέσων για ήχο και εικόνα. Ταυτόχρονα η χρήση του αφορά προγραμματιστές όλων των επιπέδων, χάρη στον απλά δομημένο τρόπο επεξεργασίας δεδομένων.

Το Jitter είναι ενσωματωμένο στο γραφικό περιβάλλον προγραμματισμού της CYCLING 74 MAX, το οποίο μας επιτρέπει να συνδέουμε οπτικά τα αντικείμενα επεξεργασίας δεδομένων μέσω patchcords για να δημιουργήσουμε εφαρμογές, κατά έναν παρόμοιο τρόπο με τους αναλογικούς συνθετητές. Μέσα σε αυτό το οπτικό πλαίσιο δημιουργούνται τα οπτικά εφέ, οι οπτικοί αναμίκτης σε πραγματικό χρόνο (video mixing), ακουστικά visualizers, αλγοριθμικές γεννήτριες εικόνας, προγράμματα μετατροπών batch επεξεργασίας κ.α.

### 1.2.2 Ψηφιακή εικόνα – Γενικά

Μια ψηφιακή εικόνα αποτελεί τη δισδιάστατη αναπαράσταση μίας εικόνας μέσω ενός πεπερασμένου αριθμού ψηφιακών τιμών, τα εικονοστοιχεία (pixels). Τα εικονοστοιχεία αποτελούν το μικρότερο δομικό στοιχείο της ψηφιακής εικόνας. Το κάθε ένα συγκρατεί την πληροφορία για την αντιπροσώπευση του χρώματος και της φωτεινότητας (brightness) του. Κάθε εικόνα έχει συγκεκριμένο αριθμό σειρών και στηλών για τα εικονοστοιχεία. Συνήθως τα pixels αποθηκεύουν την πληροφορία τους στο πλαίσιο του raster image, ένα δισδιάστατο πίνακα ακεραίων. Το raster image είναι ένας τρόπος δόμησης της ψηφιακής πληροφορίας που αναπαριστά ένα ορθογώνιο πλαίσιο από pixels που μπορεί να απεικονιστεί σε μια οθόνη υπολογιστή ή σε εκτυπωμένο χαρτί. Σε αυτό το σημείο δεν θα εμβαθύνουμε στους τρόπους συμπίεσης αυτής της πληροφορίας ή στις αρχές της ψηφιακής εικόνας γενικότερα αλλά στον τρόπο που αυτή η πληροφορία χρησιμοποιείται στο MAX/MSP Jitter.

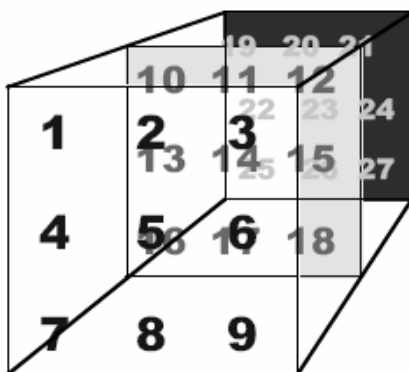
### 1.3 Ο τρόπος λειτουργίας της ψηφιακής εικόνας στο Jitter.

Ένας πίνακας είναι ένα πλέγμα, που αποτελείται από θέσεις οι οποίες περιέχουν κάποιες πληροφορίες. Παραδείγματος χάρη, μια επιφάνεια σκακιού είναι ένας πίνακας αποτελούμενος από τετράγωνα. Κάθε τετράγωνο περιέχει ένα συγκεκριμένο στοιχείο πληροφοριών: την ύπαρξη ενός ιδιαίτερου κομματιού σκακιού ή την έλλειψη ενός κομματιού σκακιού. Ας υποθέσουμε ότι οι "πληροφορίες" σε κάθε θέση σε ένα πίνακα είναι αριθμητικές τιμές. Παρακάτω παρατηρούμε ένα πίνακα με έναν αριθμό σε κάθε θέση πλέγματος.

0.56	7.65	47.89
23.48	12.47	5.24
4.56	3.22	3.21

Θα ονομάσουμε κάθε οριζόντια γραμμή στοιχείων σειρά και κάθε κάθετη γραμμή στήλη. Το Jitter αριθμεί κάθε θέση πλέγματος με τον αριθμό στήλης και σειράς, όπου η θέση(0,0) είναι η πάνω αριστερά. Με αυτό τον τρόπο μπορούμε να γνωρίζουμε την πληροφορία που περιέχεται σε κάθε θέση του πίνακα(π.χ. στη θέση πίνακα (1,2) έχουμε τον αριθμό 3.22)

Η εικόνα που παρουσιάζεται παραπάνω είναι ένα παραδείγμα πίνακα με δύο διαστάσεις, πλάτος(οριζόντια) και ύψος(κατακόρυφος). Στο Jitter, ένας πίνακας μπορεί να έχει οποιοδήποτε πλήθος διαστάσεων από 1 έως 32. Αν και είναι λίγο δύσκολο να απεικονίστει σε χαρτί, κάποιος θα μπορούσε να φανταστεί ένα πίνακα με τρεις διαστάσεις, ως κύβο που έχει μήκος, πλάτος και ύψος. Σε αυτή την εργασία θα περιοριστούμε σε δισδιάστατους πίνακες(2d).



Εικ.2 Ένας 3X3X3 πίνακας

### 1.3.1 Από τον πίνακα στην εικόνα

Μια ψηφιακή εικόνα είναι ένας τύπος πίνακα στις θέσεις του οποίου αποθηκεύονται τα εικονοστοιχεία που την αποτελούν (pixels). Ο καθένας που χρησιμοποιεί οθόνη υπολογιστή, θα έχει συναντήσει τον αριθμό που δείχνει την ανάλυση της, που είναι συνήθως κάποιο μέγεθος όπως 1024 εικονοστοιχεία πλάτος x 768 εικονοστοιχεία μήκος, ή ίσως 800x600 ή 640x480.

Παρατηρούμε ότι σε όλες αυτές τις περιπτώσεις ο αποκαλούμενος λόγος διάστασης του πλάτους στο ύψος είναι 4:3. Με το ευρύτερο σχήμα DV, ο λόγος διάστασης είναι 3:2, και η εικόνα είναι 720x480. Η υψηλής ευκρίνειας τηλεόραση (HDTV) παρέχει ακόμα μια πτυχή αναλογίας, 16:9. Σε αυτή την εργασία χρησιμοποιούμε ένα λόγο διάστασης 4:3.

Σε ένα πλαίσιο τυποποιημένου video (δηλαδή μια ενιαία τηλεοπτική εικόνα σε μια δεδομένη χρονική στιγμή) μεγέθους π.χ 640x480 έχουμε 307.200 εικονοστοιχεία. Κάθε εικονοστοιχείο επιδεικνύει ένα χρώμα. Προκειμένου να αντιπροσωπευθεί το χρώμα κάθε εικονοστοιχείου αριθμητικά, με αρκετή ποικιλία για ρεαλιστική απεικόνιση, χρειαζόμαστε μια μεγάλη γκάμα διαφορετικών πιθανών τιμών χρώματος.

### 1.3.2 Χρωματική αντιπροσώπευση (colorspace).

Είναι ευρέως γνωστό, πως, στη ζωγραφική, μπορούμε να δημιουργήσουμε οποιοδήποτε χρώμα από τα βασικά χρώματα μπλε, κόκκινο και κίτρινο. Αυτά τα τρία χρώματα συνιστούν άρα ένα **χρωματικό χώρο (colorspace)**. Στην ψηφιακή εικόνα χρησιμοποιούνται χρωματικά μοντέλα για την αντιπροσώπευση του χρώματος. Ένα χρωματικό μοντέλο είναι ένα μαθηματικό μοντέλο που περιγράφει το τρόπο που θα αντιπροσωπευθεί το χρώμα από μία σειρά αριθμών συνήθως τριών ή τεσσάρων τιμών. Αν σε αυτό το μαθηματικό μοντέλο κάνουμε μία αντιστοίχιση σε κάποια χρώματα-αναφορά από ένα χρωματικό χώρο δημιουργούμε έναν ψηφιακό χρωματικό χώρο.

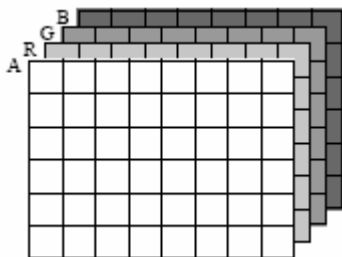
Ένας τυποποιημένος τρόπος στους υπολογιστές είναι να αναλυθεί το χρώμα σε τρία διαφορετικά τμήματα χρωμάτων - κόκκινο, πράσινο, μπλε (red - green - blue, rgb), καθώς και σε ένα πρόσθετο τμήμα διαφάνειας (opacity), γνωστό ως κανάλι άλφα (alpha channel). Τα περισσότερα προγράμματα υπολογιστών επομένως αποθηκεύουν το χρώμα ενός εικονοστοιχείου με τέσσερις χωριστούς αριθμούς-κανάλια, που αντιπροσωπεύουν τις τιμές επί του άλφα, και των συστατικών χρωμάτων κόκκινου, πράσινου, και μπλε. Αυτό το σχήμα αντιπροσώπευσης χρώματος καλείται συνήθως ARGB ή RGBA, ανάλογα με το πώς τα εικονοστοιχεία τοποθετούνται στη μνήμη.

Αντίστοιχα λοιπόν στο Jitter, στη κάθε θέση ενός πίνακα αποθηκεύονται τέσσερις αριθμητικές τιμές (άλφα, κόκκινο, πράσινο, και μπλε). Επακόλουθα, ένα καρέ του video αντιπροσωπεύεται στο Jitter ως ένας δισδιάστατος πίνακας. Κάθε στοιχείο (pixel) του πίνακα πρέπει να συγκρατεί αφενός την πληροφορία της θέσης του (συντεταγμένες X,Y) στο δισδιάστατο επίπεδο, και αφετέρου την RGBA τιμή που αντιστοιχεί σε αυτήν.

Προκειμένου να διαχειριστεί καλύτερα τις παραπάνω αριθμητικές πληροφορίες (τιμές RGBA και συντεταγμένες στο δισδιάστατο χώρο), οι οποίες είναι ουσιαστικές για την σωστή απεικόνιση της ψηφιακής εικόνας, το jitter εισάγει την έννοια των επιπέδων (planes).

### 1.3.3 Τι είναι ένα επίπεδο;

Η έννοια του επιπέδου (plane) στο jitter, δεν είναι άλλη από μία ομαδοποίηση των πληροφοριών σχετικά με τα τέσσερα βασικά συστατικά που αποτελούν ένα pixel. Προκειμένου να παρακολουθηθούν οι διαφορετικές τιμές που συγκρατούνται σε ένα εικονοστοιχείο, το Jitter χρησιμοποιεί την καθεμία σε ένα χωριστό επίπεδο (plane). Έτσι μπορούμε να σκεφτούμε ένα πλαίσιο εικόνας ως ένα δισδιάστατο πίνακα τεσσάρων επιπέδων.



Εικ. 3

Χρησιμοποιώντας αυτό το εννοιολογικό πλαίσιο, μπορούμε να μεταχειριστούμε κάθε επίπεδο, και έτσι κάθε κανάλι των πληροφοριών χρώματος, χωριστά όταν χρειαστεί. Παραδείγματος χάρη, εάν θέλουμε να αυξήσουμε την ερυθρότητα μιας εικόνας, μπορούμε απλά να αυξήσουμε όλες τις τιμές στο κόκκινο επίπεδο ενός πίνακα και να αφήσουμε τις τιμές των άλλων επιπέδων αμετάβλητες. Η συνήθης περίπτωση για το video στο Jitter είναι ένας πίνακας 2 διαστάσεων με τέσσερα επίπεδα του στοιχείου ARGB. Τα επίπεδα είναι αριθμημένα από 0 έως 3, έτσι το άλφα κανάλι είναι στο επίπεδο 0, και τα RGB κανάλια στα επίπεδα 1, 2 και 3 αντίστοιχα.

### 1.3.4 Αριθμητική αντιπροσώπευση τιμών

Οι υπολογιστές χρησιμοποιούν διαφορετικά `formats` για την αποθήκευση των αριθμών. Γενικά ο πίνακας του `jitter` μας δίνει τη δυνατότητα να χρησιμοποιήσουμε τους περισσότερους από τους πιο γνωστούς τύπους `format`. Σε αυτή την εργασία θα χρησιμοποιήσουμε μόνο τον `char` τύπο τιμών που αφενώς μας καλύπτει στο προγραμματισμό και αφετέρου είναι ο μικρότερος σε μέγεθος μνήμης. Όπως είναι γνωστό ο `char` τύπος τιμών καταλαμβάνει 8 bit αποδίδοντας ένα εύρος τιμών από 0 έως 255 ( $2^8=256$ ). Ένα τέτοιο εύρος τιμών είναι ικανοποιητικό για να αντιπροσωπεύσει οποιοδήποτε χρώμα στο `rgb`. Στην περίπτωση αυτή οι 256 διαφορετικές πιθανές τιμές αφορούν στην αριθμητική τιμή του `ARGB` και όχι σε κάποιον αλφαριθμητικό χαρακτήρα. Εφόσον ένα καρέ του βίντεο περιέχει τόσα πολλά `pixels` και το καθένα είναι πιθανό να κρατάει 4 τιμές, καταλαβαίνουμε γιατί χρειάζεται το `Jitter` να κάνει οικονομία στον αποθηκευτικό χώρο όταν επεξεργάζεται τέτοια δεδομένα. Για μονοχρωματικές (ασπρόμαυρες) εικόνες ή `video` ένα μόνο επίπεδο τύπου `char` είναι αρκετό.

### 1.3.5 Δημιουργία πίνακα `jitter`

Το βασικό αντικείμενο του `jitter`, στο οποίο στηρίζεται όλη η λειτουργία του, δεν είναι άλλο από το αντικείμενο `jit.matrix`. Αυτό δημιουργεί ένα πίνακα, δηλαδή έναν αποθηκευτικό χώρο στη μνήμη, στον οποίο μπορούμε να αποθηκεύουμε και να ανακτούμε αριθμητικά δεδομένα. Με χρήση άλλων αντικειμένων μπορούμε να εξάγουμε αυτά τα δεδομένα ή να τα εμφανίσουμε οπτικώς.

```
jit.matrix 1 char 16 12
```

Δημιουργία ενός 16\*12 αποθηκευτικού χώρου για τύπου `char` τιμές.

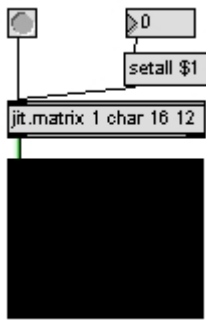
Τα βασικά ορίσματα που δέχεται ένας πίνακας είναι:

1. όνομα (το οποίο δεν περιέχεται στο παραπάνω παράδειγμα)
2. αριθμός επιπέδων (`planes`): το πλήθος των αριθμητικών τιμών που αντιπροσωπεύουν το χρώμα σε κάθε `pixel`
3. τύπος δεδομένων: πόσα `bytes` θα χρησιμοποιηθούν για κάθε μία τιμή

#### 4. διαστάσεις σε pixel: το συνολικό μέγεθος του πίνακα

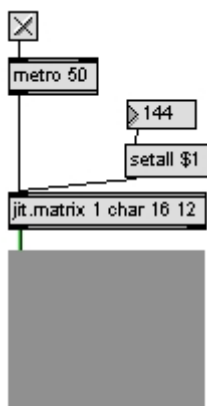
Με βάση το παραπάνω παράδειγμα έχουμε δημιουργήσει ένα πίνακα με ένα επίπεδο(plane), δηλαδή μία τιμή κρατείται σε κάθε pixel, χρησιμοποιώντας δεδομένα τύπου char, και με διαστάσεις 16 pixels σε οριζόντιο επίπεδο και 12 σε κάθετο(συνολικά 192 pixels). Έτσι, ο πίνακας μας είναι ικανός να αποθηκεύσει 192 αριθμητικές τιμές με εύρος από 0 έως 255.

Ας δούμε τώρα μερικούς τρόπους που μπορούμε να χρησιμοποιήσουμε για να γεμίσουμε τιμές τα εικονοστοιχεία ενός τέτοιου πίνακα. Η απεικόνιση του πίνακα γίνεται με το αντικείμενο jit.pwindow. Αυτό έχει τη δυνατότητα να επιδεικνύει αριθμητικές αξίες ως χρώμα. Η λειτουργία του θα γίνει πιο κατανοητή με το παρακάτω παράδειγμα.



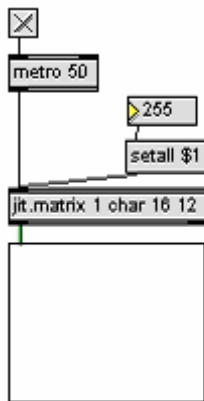
Εικ. 4

Με το μήνυμα(message) setall αποθηκεύουμε μία τιμή σε όλες τις θέσεις του πίνακα. Στο παραπάνω παράδειγμα η τιμή αυτή είναι το 0 το οποίο οπτικά αντιστοιχεί σε μαύρη εικόνα(εικ.4).



Εικ. 5





Εικ. 6

Θέτοντας τιμή 144 έχουμε δημιουργήσει μια γκριζα εικόνα(εικ.5). Η μέγιστη αξία που μπορεί να δεχτεί ένας πίνακας τύπου char είναι το 255 και οπτικά αυτό αντιστοιχεί στο άσπρο χρώμα(εικ.6). Επειδή ο πίνακας που επεξεργαζόμαστε είναι ενός επιπέδου, βλέπουμε απεικονίσεις του πίνακα μόνο στις διαβαθμίσεις του άσπρου.

### 1.3.6 Αναπαράγοντας ένα video αρχείο

Όταν εισάγουμε video δεδομένα στο jitter αυτά μετατρέπονται αυτόματα σε πίνακα Jitter. Το αντικείμενο του Jitter, για αναπαραγωγή ενός αρχείου video είναι το **jit.qt.movie**. Αυτό παρέχει μία σουίτα ανάλογη της αρχιτεκτονικής του quicktime για την κωδικοποίηση του video. Περιέχει στις λειτουργίες του όλες τις δυνατότητες που έχει το σύνηθες περιβάλλον του quicktime player: αναπαραγωγή video, επεξεργασία, εισαγωγή και εξαγωγή αρχείου ήχου και εικόνας, δημιουργία εφέ καθώς και έξοδο video σε πραγματικό χρόνο. Με την εντολή read ανοίγουμε ένα browser ώστε να επιλέξουμε ένα αρχείο video. Μόλις αυτό επιλεγεί, ανοίγει το αρχείο στο αντικείμενο και αρχίζει να το διαβάζει.



Εικ. 7

Παρατηρούμε πως ενώ το video έχει ανοίξει και διαβάζεται από το αντικείμενο δεν μπορούμε να το δούμε. Για να καταλάβουμε γιατί συμβαίνει αυτό, θα εξηγήσουμε τις δύο βασικές εσωτερικές λειτουργίες του jitter που πρέπει να λάβουν χώρα ώστε να απεικονιστεί το video:

1. Μεταφορά του αρχείου από το σκληρό δίσκο στη ram
2. Ανάγνωση του αρχείου από τη ram και απεικόνιση σε χρωματισμένα pixel στην οθόνη.

Την πρώτη λειτουργία εκτελεί το αντικείμενο `jit.qt.movie` και τη δεύτερη το αντικείμενο `jit.pwindow`. Για να ξέρει όμως το `jit.pwindow` τι να απεικονίσει πρέπει τα δύο αυτά αντικείμενα να επικοινωνήσουν μεταξύ τους.

### 1.3.7 Πως επικοινωνούν τα jitter αντικείμενα

Όπως τονίστηκε προηγουμένως, η πρωταρχική λειτουργία που εκτελεί το `jit.qt.movie` είναι να ανοίξει ένα video αρχείο και να αποθηκεύσει το κάθε καρέ στη ram για γρήγορη πρόσβαση από άλλα jitter αντικείμενα. Για να το πετύχει αυτό καταλαμβάνει ένα συγκεκριμένο χώρο στη μνήμη τον οποίο χρησιμοποιεί συνεχώς. Για να συγκρατήσει αυτές τις θέσεις στη ram, το jitter αποδίδει στην κάθε μία, μία εσωτερική ονομασία, που εμείς δεν χρειαζόμαστε απαραίτητα να γνωρίζουμε. Αυτή η ονομασία, που αναφέρεται σε ένα πίνακα και που αντιστοιχεί σε μία συγκεκριμένη θέση στη μνήμη, έχει ως αποτέλεσμα την επικοινωνία μεταξύ των αντικειμένων. Το μήνυμα που εξάγεται από κάθε jitter αντικείμενο είναι η λέξη `jit_matrix` ακολουθούμενη από ένα κενό και το όνομα του πίνακα όπου βρίσκονται αποθηκευμένα τα δεδομένα. Το μήνυμα αυτό μεταφέρεται από ένα jitter αντικείμενο στο επόμενο, μέσω καλωδίωσης (patchcord). Η διαδικασία αυτή είναι όμοια με αυτή του `max`, με τη διαφορά ότι τα καλώδια του jitter που μεταφέρουν τα στοιχεία ενός πίνακα έχουν πράσινο χρώμα. Το αντικείμενο που δέχεται αυτό το μήνυμα στην είσοδο του, αποσπά τα δεδομένα από τη συγκεκριμένη θέση μνήμης, τα επεξεργάζεται και στέλνει στην αριστερή έξοδο του τη νέα θέση των επεξεργασμένων δεδομένων.

Σε πολλές περιπτώσεις, ένα jitter αντικείμενο θα δημιουργήσει ένα μοναδικό όνομα για τον πίνακα και θα ανήκει μόνο σε αυτό. Σε άλλες περιπτώσεις είναι πιθανό και εύλογο να ορίσουμε για το αντικείμενο ένα κοινό όνομα πίνακα, ώστε να χρησιμοποιείται και από άλλα αντικείμενα στο patch, εξοικονομώντας έτσι μνήμη.

Για να προκαλέσουμε όμως τη ροή των δεδομένων ανάμεσα στα jitter αντικείμενα χρειαζόμαστε ένα άλλο μήνυμα. Αυτό είναι το `outputmatrix` ή

απλώς ένα **bang**. Αυτό το μήνυμα μεταφέρεται με τη ροή των δεδομένων από το ένα αντικείμενο στο άλλο οπότε δεν χρειάζεται να το στέλνουμε στο κάθε αντικείμενο ξεχωριστά. Αν δηλαδή, δώσουμε αυτό το μήνυμα στο πρώτο αντικείμενο της αλυσίδας των αντικειμένων, μόλις αυτό τελειώσει την επεξεργασία των δεδομένων που του έχουμε ορίσει, στέλνει αυτόματα ένα `jit_matrix` μήνυμα ενημερώνοντας τα άλλα αντικείμενα για τη θέση μνήμης που περιέχει τα καινούρια δεδομένα. Ας δούμε ένα παράδειγμα που θα κάνει πιο κατανοητή αυτή τη λειτουργία:



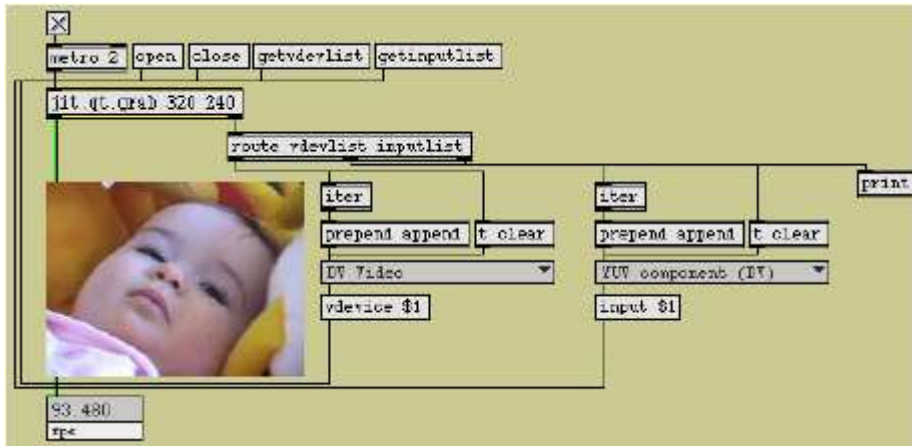
Εικ. 8

Το `jit.qt.movie` αναπαράγει το video αποθηκεύοντας το τρέχον καρέ, αλλά το `jit.pwindow` θα απεικονίσει το video μόνο όταν λάβει το `jit_matrix` μήνυμα από το `jit.qt.movie` και αυτό θα συμβεί όταν το τελευταίο δεχτεί το μήνυμα **bang**. Σε αυτό το χρονικό σημείο, το `jit.pwindow` θα απεικονίσει οποιοδήποτε καρέ του video τυχαίνει να αναπαράγεται τη δεδομένη στιγμή. Το **metro** αντικείμενο που χρησιμοποιούμε, ενεργοποιεί τη μεταφορά των δεδομένων και είναι ανεξάρτητο από το frame rate του αρχείου. Αν έχουμε όμως προς αναπαραγωγή ένα αρχείο με frame rate 25 fps, αυτό έχει συχνότητα εναλλαγής καρέ τα 40 m/s. Αν η συχνότητα, που το metro αντικείμενο στέλνει bang μηνύματα είναι μεγαλύτερη από 40 m/s, τότε το αποτέλεσμα στην οθόνη θα είναι να έχει το video μικρότερη τιμή fps.

### 1.3.8 Δουλεύοντας με live video

Το `jit.dx.grab` είναι ένα περιβάλλον εργασίας που ανήκει σε μία από τις ρουτίνες του quicktime για επικοινωνία με εξωτερικές συσκευές απεικόνισης. Με το μήνυμα **getvdevlist**, το αντικείμενο θα μας δείξει όλες τις, συμβατές με το quicktime, συσκευές που βλέπει συνδεδεμένες στον υπολογιστή. Μένει απλά να επιλέξουμε την επιθυμητή συσκευή και με την εντολή **open** θα αρχίσει να εισάγεται η ακολουθία εικόνων με τρόπο παρόμοιο με του προηγούμενου παραδείγματος. Ουσιαστικά αυτή η εντολή

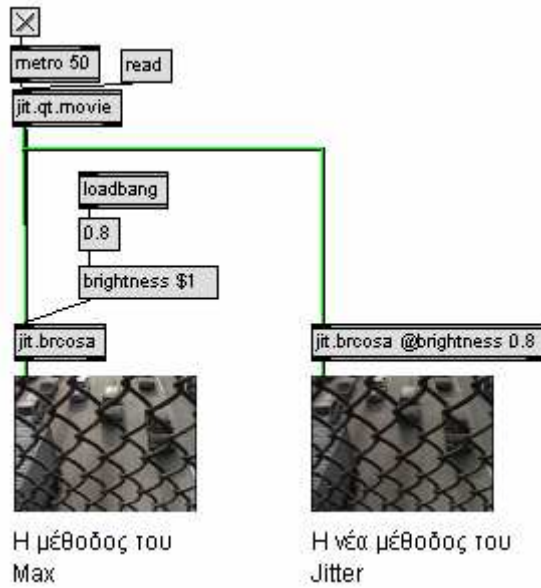
λέει στο object να συνδεθεί με τη συσκευή μας. Και εδώ φυσικά οδηγούμε τη ροή με metro αντικείμενο.



Εικ.9 Το Jitter σε οικογενειακές στιγμές

### 1.3.9 Ο προσδιορισμός @.

Πριν προχωρήσουμε παρακάτω θα αναφερθούμε στη μέθοδο που χρησιμοποιεί το Jitter, σε αντίθεση με το MAX, για τον καθορισμό των ιδιοτήτων των αντικειμένων. Τα αντικείμενα του jitter μπορούν να έχουν πολλές διαφορετικές καταστάσεις εσωτερικής λειτουργίας. Στο jitter, σε αντίθεση με το max που χρησιμοποιεί το **loadbang**, για να δωθούν κάποιες αρχικές τιμές μπορούμε να δηλώσουμε μια παράμετρο εσωτερικής λειτουργίας με τον χαρακτήρα @ ακολουθούμενο από το όνομα της ιδιότητας χωρίς κενό και μετά από ένα κενό το προσδιορισμό της παραμέτρου.



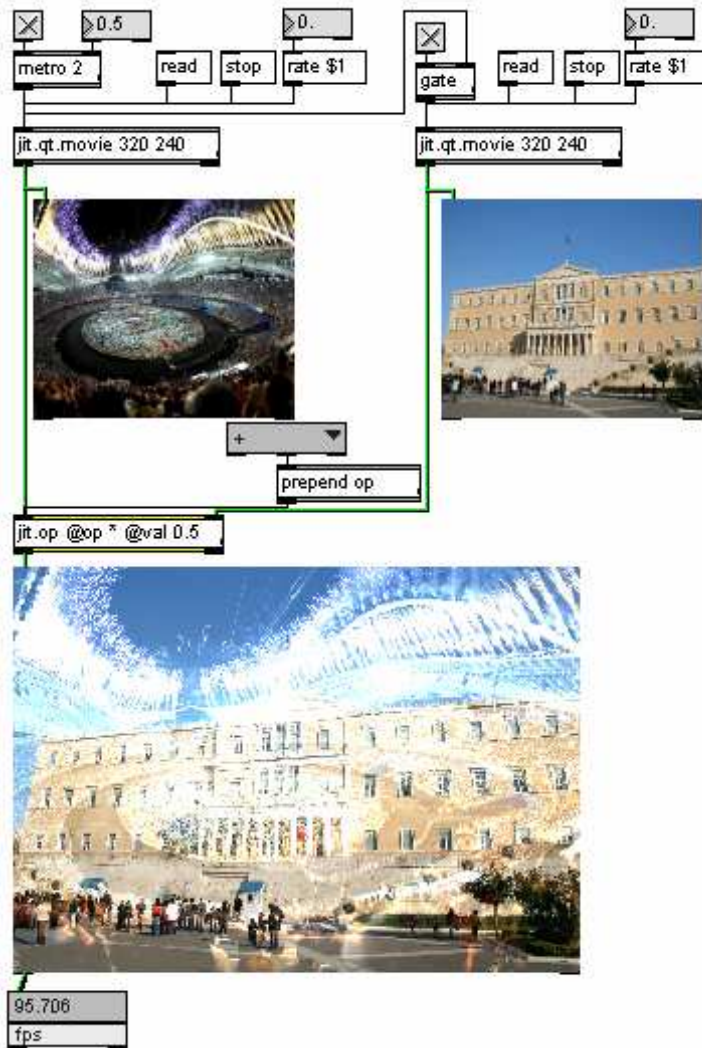
Εικ.10

### 1.3.10 jit.op

Το αντικείμενο **jit.op** εκτελεί μαθηματικές πράξεις στο σύνολο των δεδομένων του πίνακα ή σε κάθε επίπεδο(plane) ξεχωριστά. Οι μαθηματικές αυτές πράξεις είναι ανάλογες της γραμμικής άλγεβρας.

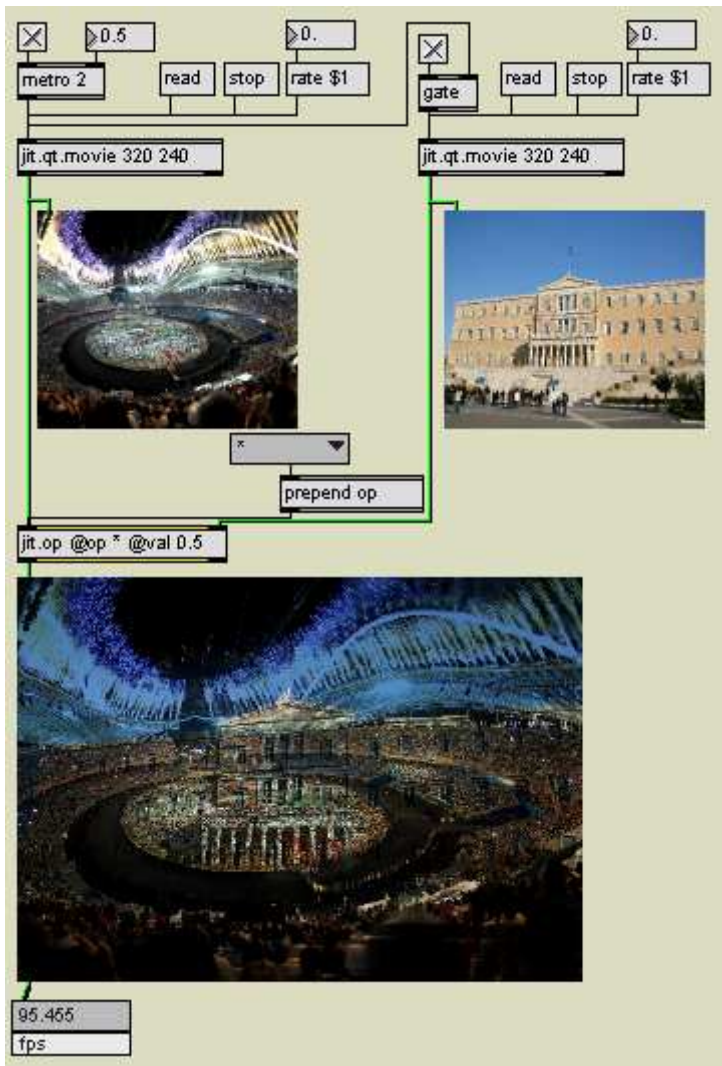


Βλέπουμε ότι το object έχει δύο εισόδους. Μπορούμε να εκτελέσουμε πράξεις μεταξύ δυο πινάκων ή και μόνο σε ένα. Ας δούμε ένα παράδειγμα που θα μας βοηθήσει να το κατανοήσουμε καλύτερα.



Εικ.11 Εθνική πόρωση με jit.op

Στο παραπάνω παράδειγμα έχουμε εκτελέσει την πρόσθεση των τιμών των δύο πινάκων. Μπορούμε να το καταλάβουμε κοιτώντας τις δύο εικόνες ξεχωριστά. Στα μαύρα σημεία της πρώτης εικόνας, η δεύτερη εικόνα έχει μείνει αμετάβλητη ενώ στα σημεία με μεγαλύτερες τιμές βλέπουμε έντονα άσπρα σαν αποτέλεσμα της πρόσθεσης μεγάλων αξιών.



Εικ.12

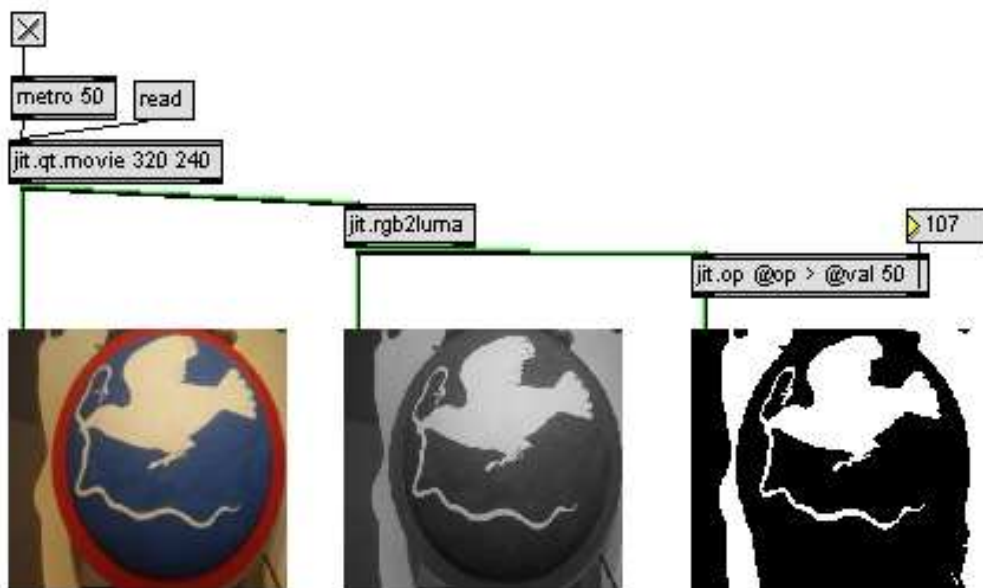
Εδώ μπορούμε να παρατηρήσουμε το αποτέλεσμα του πολλαπλασιασμού. Αν παρατηρήσουμε τα σημεία που η πρώτη εικόνα είναι μαύρη θα δούμε πως έχει παραμείνει έτσι καθώς πολλαπλασιάζουμε την αντίστοιχη τιμή της δεύτερης εικόνας με το 0.

## 1.4 Τεχνικές ανίχνευσης

Σε προηγούμενη παράγραφο εξετάστηκε το ευρύτερο πλαίσιο του computer vision. Σε αυτή την παράγραφο, θα ερευνηθούν συγκεκριμένες μέθοδοι ανίχνευσης και πως υλοποιούνται προγραμματιστικά στο jitter.

Ένας συνηθισμένος τρόπος για να ανιχνεύσουμε την κίνηση ή τη θέση ενός αντικειμένου στο χώρο είναι η ανίχνευση χρώματος(color tracking). Σε αυτή τη μέθοδο εισάγεται στο σύστημα ένα χρώμα(και ένα πιθανό εύρος απόκλισης από αυτό) και έτσι παρακολουθείται η θέση του χρώματος στο δισδιάστατο επίπεδο.

Το μεγαλύτερο μέρος όμως των μεθόδων του cv δεν χρησιμοποιεί χρώμα αλλά μετατρέπει την εικόνα που εισάγουμε στο σύστημα από την κάμερα σε boolean πληροφορία(on-off) για περαιτέρω ανάλυση. Η δυαδική αυτή εικόνα(binary image) απεικονίζεται σε πληροφορία δύο τιμών μόνο(άσπρου και μαύρου), και δεν έχει να κάνει με την ασπρόμαυρη εικόνα όπως την ξέρουμε, όπου παρουσιάζονται διαφορετικές διαβαθμίσεις φωτεινότητας. Στη δυαδική εικόνα επιλέγουμε ένα κατώτατο όριο φωτεινότητας πάνω από το οποίο θα μας εμφανίζονται άσπρα τα pixels.



Εικ.13 Παράδειγμα μετατροπής μίας εικόνας πρώτα σε ασπρόμαυρη και μετά σε δυαδική



Στο `jit`, η δυαδική εικόνα ορίζεται ως πίνακας τύπου `char` ενός επιπέδου. Αυτός ο τύπος εικόνας, είναι εξαιρετικά χρήσιμος στις εφαρμογές `cv` για την ανάλυση σχήματος και περιορίζει κατά πολύ τον αριθμό των πράξεων που εκτελούνται στον επεξεργαστή.

Αφού μετατρέψαμε την εικόνα μας σε δυαδική, θα αναλύσουμε συγκεκριμένες μεθόδους που χρησιμοποιούνται σε τέτοιου τύπου εικόνες για να αποσπάσουμε πληροφορίες από αυτές, οι οποίες είναι χρήσιμες για εφαρμογές αλληλεπίδρασης. Στις περισσότερες εφαρμογές έχουν χρησιμοποιηθεί τα αντικείμενα από τη freeware βιβλιοθήκη `cv.jit` που προσαρμόζουν τις γνωστές τεχνικές `cv` για χρήση από το `Max/Msp jit`.

#### **1.4.1. Image segmentation (κατάτμηση εικόνας) .**

Στη μέθοδο κατάτμησης της εικόνας (`image segmentation`) αναγνωρίζεται και απομονώνεται ένα μέρος της εικόνας που αποτελείται από τα ίδια χαρακτηριστικά. Αυτή η περιοχή ονομάζεται `blob` που μεταφράζεται στα ελληνικά ως σταγόνα. Το `blob` είναι μία περιοχή απο συνεχόμενα `on pixels`, δηλαδή πιο περιγραφικά, ένα νησί μέσα σε μια θάλασσα από `off pixels`.

Στο `jit`, τα αντικείμενα για ανίχνευση `blobs` προέρχονται από την κατηγορία `connected components` του `cv.jit`. Στο παράδειγμα που ακολουθεί χρησιμοποιούνται δύο αντικείμενα από αυτή την κατηγορία.

#### **`cv.jit.label`**



Αυτό το αντικείμενο αναγνωρίζει όλα τα `blobs` σε μια δυαδική (`binary`) εικόνα και αποδίδει στο καθένα μία μοναδική τιμή (`ID`).

Είσοδος: Μονό επίπεδο (`plane`) `char`

Έξοδος: Μονό επίπεδο `char`

Το χρησιμοποιούμε σε συνδυασμό με άλλα αντικείμενα της ίδιας κατηγορίας για να ανιχνεύσουμε τις συντεταγμένες των `blobs` στο χώρο.

## cv.jit.blobs.centroids

cv.jit.blobs.centroids

Δέχεται στην είσοδο του την έξοδο του cv.jit.label. Εξάγει:

1. Τις συντεταγμένες του κεντρικού σημείου για το κάθε blob.
2. Το μέγεθος σε pixels της επιφάνειας που καταλαμβάνει το κάθε blob στο δισδιάστατο επίπεδο.

Είσοδος: Μονό επίπεδο char

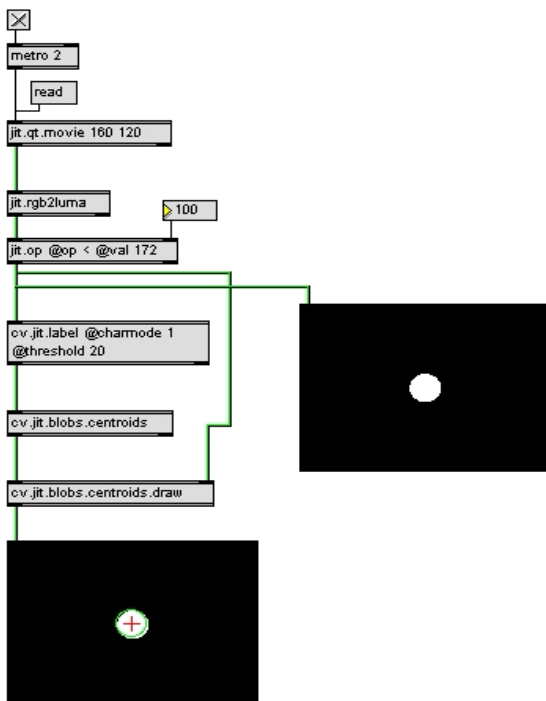
Έξοδος: Πίνακας τριών επιπέδων, μονή σειρά float32 εκ των οποίων:

Επίπεδο 1: Συντεταγμένη X που αντιστοιχεί στο κέντρικό σημείο κάθε blob

Επίπεδο 2: Συντεταγμένη Y που αντιστοιχεί στο κέντρικό σημείο κάθε blob

Επίπεδο 3: Συνολική επιφάνεια στο δισδιάστατο επίπεδο για το κάθε blob.

Το μέγεθος του πίνακα είναι όσο και ο αριθμός των blobs που αναγνωρίστηκαν.



Εικ.14 Παρατηρούμε τη δυαδική εικόνα ενός κύκλου και πως έχει αυτή ανιχνευτεί από το σύστημα ως blob με το πράσινο περιθώριο, σημαδεύοντας το κέντρο με ένα σταυρό.

#### 1.4.2. Feature extraction (Ανίχνευση χαρακτηριστικών)

Ο όρος feature στο cv χρησιμοποιείται για να δηλώσει σημεία στην εικόνα που διαφέρουν από τον περίγυρο τους. Το αντίθετο του feature είναι ένα pixel που το περικυκλώνουν όμοια pixels. Η εν λόγω τεχνική επιτυγχάνεται στο jitter μέσω του αντικειμένου:

#### **cv.jit.features**



Είσοδος: Μονό πλάνο char

Έξοδος: Δύο επίπεδα float32 μονής σειράς

Στο πρώτο επίπεδο περιέχεται η συντεταγμένη X και στο δεύτερο η συντεταγμένη Y του feature. Το μέγεθος του πίνακα είναι και ο αριθμός των features που ανιχνεύθηκαν.

#### **Edge detection (ανίχνευση άκρων)**

Πολύ δημοφιλής τακτική αποτελεί η μέθοδος edge detection. Σκοπός της είναι να αναγνωρίσει τα σημεία σε μία ψηφιακή εικόνα, στα οποία η ένταση της φωτεινότητας αλλάζει απότομα. Αυτές οι ξαφνικές αλλαγές αντανακλούν πολλά δρώμενα στο ζωντανό κόσμο. Μερικά απο αυτά είναι:

- 1: Ασυνέχειες στο βάθος της εικόνας στο 3d επίπεδο.
- 2: Ασυνέχειες στη κατευθυντικότητα μίας επιφάνειας.
- 3: Ενναλλαγές στο φωτισμό μίας σκηνής.

Αυτή η τεχνική ανήκει στο γενικότερη μέθοδο της feature extraction. Στο παρακάτω παράδειγμα χρησιμοποιείται το αντικείμενο cv.jit.canny που έχει πάρει το όνομα του απο τον John Canny ο οποίος δημιούργησε τον συγκεκριμένο αλγόριθμο το 1986.<sup>5</sup>

#### **cv.jit.canny**

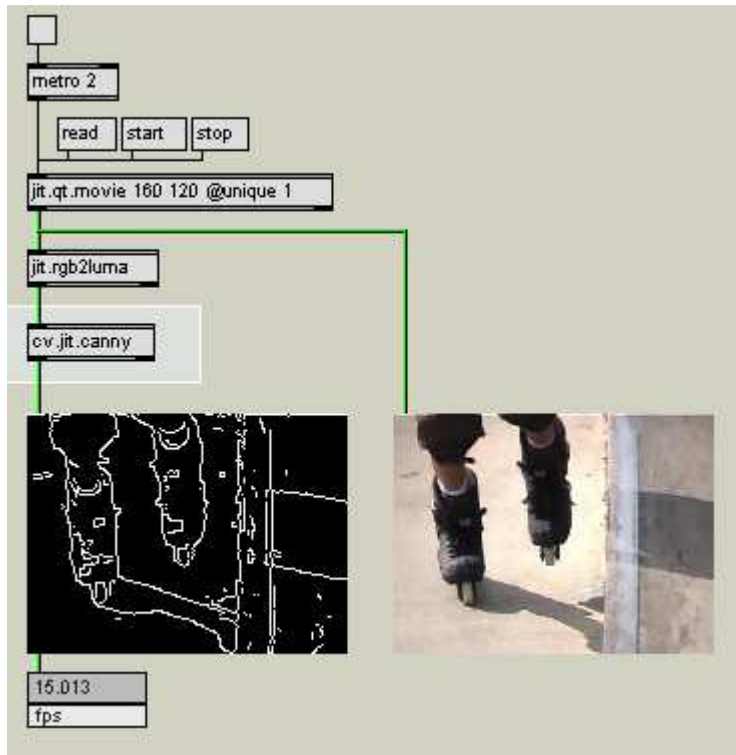
---

<sup>5</sup> Canny, J. A. 1986.

cv.jit.canny

Είσοδος: Μονό επίπεδο char

Εξοδος: Μονό επίπεδο char



Εικ.15 Χρήση αλγόριθμου Canny στο περιβάλλον του jitter. Παρατηρούμε πως επιδρά ο αλγόριθμος στην εικόνα.

#### 1.4.3. Shape recognition (αναγνώριση σχήματος)

##### Find faces

Πρόκειται για αλγόριθμο που ψάχνει σε μια ασπρόμαυρη εικόνα να βρεί περιοχές που αναπαριστούν εμπρόσθια όψη ανθρώπινου προσώπου. Επειδή ο αλγόριθμος είναι ιδιαίτερα απαιτητικός, από άποψη επεξεργαστικής ισχύως, δουλεύει καλύτερα σε μικρότερες εικόνες. Στο cv.jit:

cv.jit.faces

Είσοδος: Μονό επίπεδο char.

Εξοδος: Τεσσάρων επιπέδων char μίας σειράς.

Ο αλγόριθμος αυτός έχει μια ευελιξία που δεν φαίνεται με την πρώτη ματιά. Στην ανοιχτή βιβλιοθήκη OpenCV υπάρχουν αρχεία που μπορούν να φορτωθούν στο εν λόγω αντικείμενο ώστε να ανιχνεύει διαφορετικά σχήματα. Αυτά τα αρχεία είναι σε μορφή xml.

Υπάρχουν πολλοί αλγόριθμοι στο Jitter για διαφορετικά συστήματα ανίχνευσης όπως tracking αλγόριθμοι(κίνηση), ή αλγόριθμοι στατιστικής αναλύσης εικόνας. Ακόμα περισσότεροι υπάρχουν σε μορφή λογισμικού στην ανοιχτή βιβλιοθήκη **OpenCV**<sup>6</sup>. Σε αυτή την εργασία έχει χρησιμοποιηθεί για ανίχνευση, μόνο η τεχνική image segmentation στην εφαρμογή «Μικρές ιστορίες πόλεων». Στην εφαρμογή «polyreveal» δεν έχει χρησιμοποιηθεί κάποιος απο τους αλγόριθμους αυτούς και η ανίχνευση έγινε απευθείας με την εξαγωγή των δεδομένων ενός jit.matrix.

#### 1.4.4 Εξαγωγή πληροφοριών εικόνας

Αφού έχει γίνει κατανοητός ο τρόπος δόμησης της εικόνας στο jitter και έχουν αναλυθεί κάποιες βασικές τεχνικές ανίχνευσης, απομένει να ερευνήσουμε τρόπους που θα μας δώσουν τα δεδομένα σε αριθμητική μορφή όπου με τη χρήση max/msp θα μπορέσουμε να τα μετατρέψουμε σε μεταβλητές ρύθμισης του ήχου. Όσον αφορά στη διάδραση με την εικόνα, τις περισσότερες φορές δε θα χρειαστεί να μετατρέψουμε τις εξαγωγίμες τιμές αφού αυτές χρησιμοποιούνται εσωτερικά στο jitter.

Το jitter μας δίνει δύο τρόπους να μετατρέψουμε τον πίνακα jitter (jit.matrix) σε λίστες. Από αυτό το σημείο και μετά, ένας χρήστης του max/msp εξοικιωμένος με τις λίστες μπορεί να χρησιμοποιήσει τα δεδομένα κατά βούληση. Γνωστά αντικείμενα για επεξεργασία λίστας είναι το **zl** αντικείμενο με πολλές διαφορετικές ιδιότητες, το **pack** και πολλά άλλα, κυρίως όμως πρέπει να αναφερθούμε στα **Lobjects** που παρέχονται δωρεάν και περιέχουν πολλά διαφορετικά εργαλεία για τέτοιες επεξεργασίες.

##### 1) **jit.spill**



Είσοδος: Όλων των τύπων πίνακες

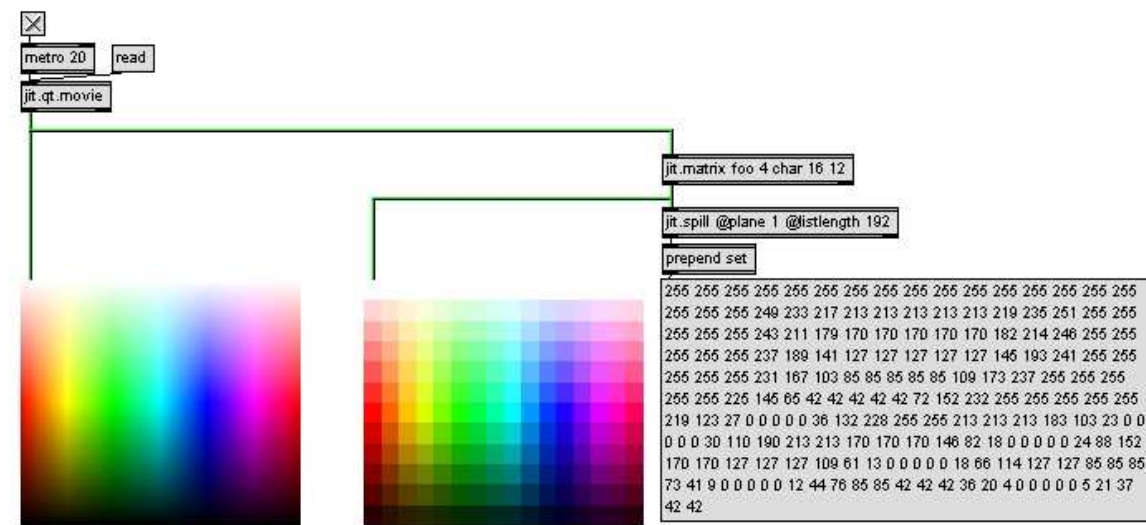
Εξοδος: Λίστα έως 128 τιμές ενός επιπέδου(plane) του πίνακα εισόδου.

---

<sup>6</sup> <http://opencvlibrary.sourceforge.net/>

Αυτό το αντικείμενο αποδίδει σε λίστα max τα περιεχόμενα ενός πίνακα jitter.

Δέχεται ορίσματα τόσο για το επίπεδο(plane) το οποίο θα αναλύσει όσο και για το μέγεθος της λίστας που θα εξάγει. Στο παρακάτω παράδειγμα, έχουμε ορίσει στο αντικείμενο να εξάγει τις πρώτες 192 τιμές του επιπέδου 1(κόκκινο στο rgb). Στα αριστερά παρατηρούμε την εικόνα στην κανονική της ανάλυση. Στα δεξιά είναι η ίδια εικόνα σε ανάλυση 16\*12 την οποία και αναλύουμε. Άρα οι 192 τιμές αποτελούν το σύνολο των τιμών του πίνακα.



Εικ.16

Καθώς αναλύουμε το κόκκινο κανάλι, παρατηρούμε ότι ο πίνακας παίρνει τη μέγιστη τιμή 255, στο πάνω μέρος του χάρτη το οποίο αντιστοιχεί στο λευκό χρώμα, όπως επίσης και στις πλευρές αριστερά και δεξιά του χάρτη στις οποίες αντιστοιχεί το ερυθρό χρώμα.

2) **jit.iter**



Είσοδος: Όλων των τύπων πίνακες.

Εξοδος: Λίστα ενός επιπέδου(plane) του πίνακα εισόδου.

Το `jit.iter` αναλύει όλα τα `pixel` του `jit.matrix` και στέλνει από την αριστερή του έξοδο, μια τιμή ή μια λίστα για κάθε ένα. Από τη μεσαία έξοδο δίνει τις συντεταγμένες του εκάστοτε `pixel` που αναλύει.



Το σύνολο των αξιών για κάθε κανάλι

## Κεφάλαιο 2: Πρακτικό μέρος-Υλοποίηση

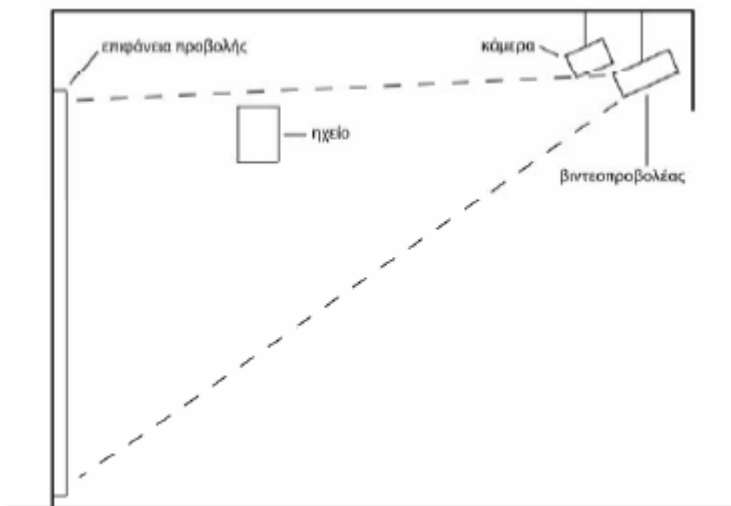
Σε αυτό το κεφάλαιο θα προσπαθήσουμε να αναλύσουμε την υλοποίηση των διαδραστικών εφαρμογών "Poly-reveal" και "Μικρές ιστορίες πόλεων". Μέσω των αλγορίθμων θα εξετάσουμε τις τεχνικές του Jitter που χρησιμοποιήθηκαν.

### 2.1 Poly-reveal

#### 2.1.1. Περιγραφή

Σε αυτό το παράδειγμα-εφαρμογή θα δούμε πως μπορούμε να χρησιμοποιήσουμε τα δεδομένα του jitter πίνακα για παράλληλη διάδραση εικόνας και ήχου. Για την εικόνα γίνεται χρήση του jit.op και για τον ήχο εξαγωγή αριθμητικών τιμών.

Η εγκατάσταση έχει ως εξής. Προβάλλουμε μια εικόνα με το projector σύμφωνα με τη διάταξη που βλέπουμε παρακάτω(Εικ. 17). Ο θεατής καλείται να περάσει μπροστά από αυτό το projection. Καθώς περνάει μπροστά από την προβολή σκιάζει την εικόνα με το σώμα του.



Εικ. 17

Τα σημεία της εικόνας που έχουν σκιάσει από το σώμα ανιχνεύονται από την κάμερα προκαλώντας την εμφάνιση μίας άλλης εικόνας σε αυτά τα σημεία σαν να «κρύβεται» από πίσω.





Εικ. 18



Εικ. 19

Η εικόνα 18 είναι η αρχική. Η εικόνα 19 είναι αυτή που θέλουμε να εμφανίσουμε ως κρυμμένη.



Εικ. 20

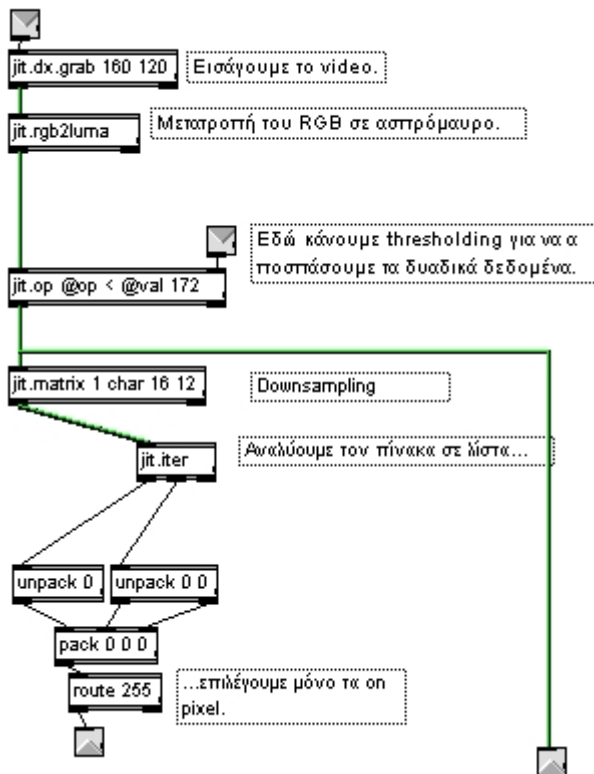


Εικ. 21

Η εικόνα 20 είναι το αντικείμενο της ανίχνευσης που σε πραγματικές συνθήκες εγκατάστασης είναι τα σημεία που σκιάζονται στην εικόνα από το θεατή. Στη συγκεκριμένη περίπτωση αυτό είναι το video ενός κύκλου που κινείται τυχαία στο δισδιάστατο επίπεδο. Στην εικόνα 21 διακρίνεται η εμφάνιση της δεύτερης εικόνας μέσα από την πρώτη στο σημείο που βρίσκεται ο κύκλος και τα σημεία που έχει περάσει. Αυτά τα σημεία δεν παραμένουν δείχνοντας τη δεύτερη εικόνα αλλά επανέρχονται στην αρχική μετά από συγκεκριμένο χρονικό διάστημα. Η αλλαγή της εικόνας δεν γίνεται σε πλήρης ανάλυση αλλά σε μία διάταξη 16\*12 τετραγώνων όπως παρατηρείται στο pixelation στην εικόνα 21. Κάθε ένα νέο τετραγωνάκι που αλλάζει παράγει ένα μουσικό τόνο.

### 2.1.2 Αλγόριθμος ανίχνευσης

Ας δούμε όμως πως υλοποιείται ένα τέτοιο περιβάλλον διάδρασης προγραμματιστικά. Θα ξεκινήσουμε από τον αλγόριθμο της ανίχνευσης.

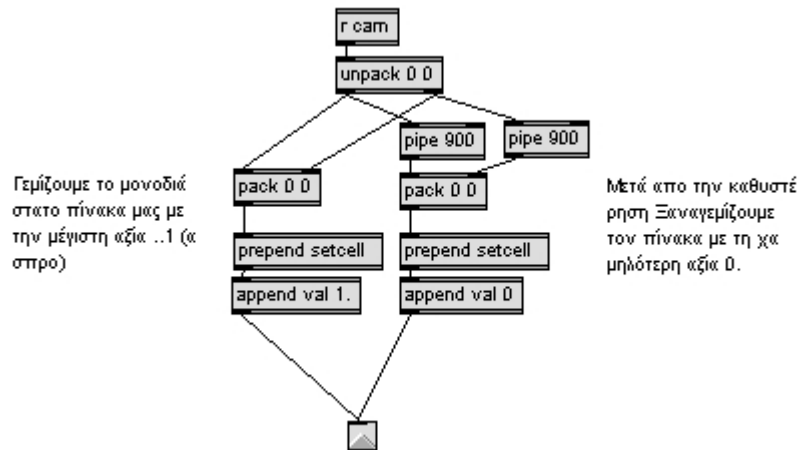


Εικ. 22

Το παραπάνω υπο-patch δέχεται στην είσοδο του το σήμα από την κάμερα. Κατά τρόπο παρόμοιο με αυτόν που σχολιάσαμε στις τεχνικές ανίχνευσης μετατρέπουμε την εικόνα σε δυαδική, μειώνουμε το μέγεθος του πίνακά για λόγους οικονομίας επεξεργαστικής ισχύως (downsampling), και τον αναλύουμε σε λίστα με το `jit.iter`. Με το αντικείμενο `route` έχουμε καταφέρει να εξάγουμε μία λίστα δύο αριθμών που είναι οι συντεταγμένες X και Y του κάθε νέου pixel που γίνεται 0η και αντιστοιχούν στην κίνηση του υποκειμένου. Αυτές τις δύο τιμές τις χρησιμοποιούμε και για το κομμάτι του ήχου στο `msp`.

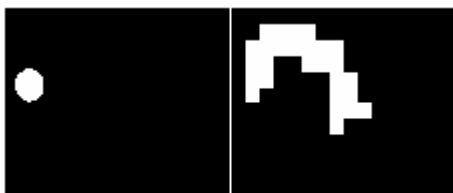
### 2.1.3 Αλγόριθμος σύνθεσης εικόνας

Θα χρησιμοποιήσουμε αυτή τη λίστα για να συνθέσουμε πάλι ένα πίνακα. Με το μήνυμα `setcell` μπορούμε να χρωμάσουμε συγκεκριμένα pixel σε ένα πίνακα, δίνοντας τις συντεταγμένες του pixel και την τιμή με την οποία θέλουμε να το χρωμάσουμε.



Εικ. 23

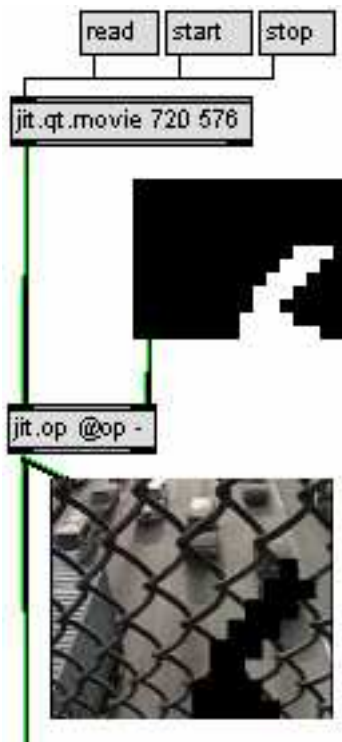
Στο αριστερό μέρος του παραπάνω υπο-patcher χρησιμοποιούμε το `setcell` δίνοντας του ως συντεταγμένες τη λίστα που αποσπάσαμε από τον αλγόριθμο ανίχνευσης και ως τιμή το 1 που αντιστοιχεί οπτικά στο άσπρο χρώμα. Στο δεξί μέρος του patch χρησιμοποιούμε την ίδια λίστα για να χρωματίσουμε τα ίδια pixels με την τιμή 0, που οπτικά αντιστοιχεί στο μαύρο χρώμα, αλλά με μία καθυστέρηση 900ms που προκαλεί το αντικείμενο **pipe**. Με αυτό τον τρόπο όποιο pixel χρωματίζεται ως άσπρο στο πίνακα που συνθέτουμε, παραμένει άσπρο έως τη στιγμή που θα επέλθει το δεύτερο `setcell` για να το μετατρέψει σε μαύρο.



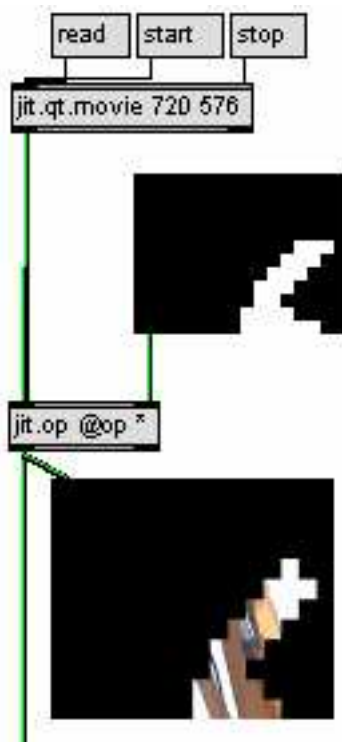
Εικ. 24

Εικ. 25

Στην εικόνα 24 βλέπουμε το video του κύκλου που χρησιμοποιήθηκε και παραπάνω. Στην εικόνα 25 είναι ο πίνακας που ανασυνθέσαμε. Σύμφωνα με το παραπάνω υπο-patcher, όσο κινείται ο κύκλος διαγράφεται η διαδρομή του με άσπρο χρώμα. Αυτή ακριβώς η διαδρομή χρωματίζεται με μαύρο μετά την καθυστέρηση. Άρα, το ίχνος που βλέπουμε, χαράζει τη διαδρομή του κύκλου για τόσο χρόνο πριν, όσο η καθυστέρηση που προβάλλαμε με το αντικείμενο `pipe`. Στο έξης θα αναφερόμαστε στον πίνακα αυτό ως πίνακα σύνθεσης.



Εικ. 26



Εικ. 27

Εδώ μπορούμε να δούμε πώς με τη χρήση του πίνακα σύνθεσης επιτυγχάνεται το εφέ. Από την εικόνα που θέτουμε ως αρχική αφαιρούμε **@op -** τον πίνακα σύνθεσης (Εικ. 26). Τα pixel του πίνακα σύνθεσης που έχουν τη μέγιστη τιμή (άσπρο) εφόσον αφαιρούνται από τα αντίστοιχα τους στην εικόνα του `jit.qt.movie` θα εμφανιστούν μαύρα στην έξοδο του `jit.op`.

Στο `jit.qt.movie` της εικόνας 27 βρίσκεται η "κρυμμένη" εικόνα που θέλουμε να εμφανίσουμε. Σε αυτή την περίπτωση χρησιμοποιούμε τον πίνακα σύνθεσης ώστε να γίνουν ορατά τα ίδια pixel που στην προηγούμενη περίπτωση έγιναν μαύρα. Το operation που χρειάζεται είναι ο πολλαπλασιασμός **@op \***. Στο `jit.pwindow` παρατηρούμε το αποτέλεσμα του πολλαπλασιασμού. Τα pixel του πίνακα σύνθεσης με την τιμή μηδέν, πολλαπλασιαζόμενα με τα αντίστοιχα τους στον πίνακα του `jit.qt.movie` φυσικά γίνονται μηδέν στην έξοδο του `jit.op`, αφήνοντας τα υπόλοιπα ανεπηρέαστα αφού η τιμή με την οποία πολλαπλασιάζονται είναι η τιμή 1.

Σε αυτό το σημείο έχουν δημιουργηθεί δύο πίνακες. Ο πρώτος περιέχει όλες τις τιμές του πίνακα του αριστερού `jit.qt.movie` εκτός από αυτές που μηδενίστηκαν από τον πίνακα σύνθεσης. Ο δεύτερος αντίστροφα περιέχει μόνο τις τιμές του δεύτερου `jit.qt.movie` που δεν μηδενίστηκαν από τον πίνακα σύνθεσης.

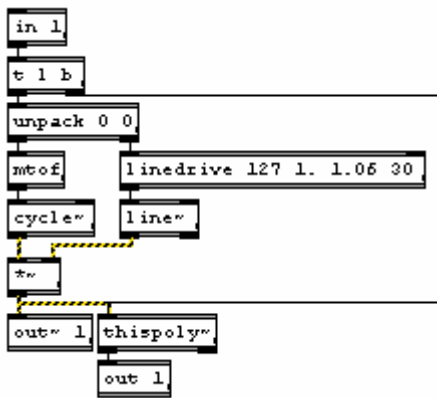


Εικ. 28

Με ένα τρίτο operation **@op max** καταφέρνουμε, από τις δύο εικόνες να φαίνονται μόνο όποια pixel έχουν μεγαλύτερη τιμή στο ένα video από τα αντίστοιχά τους στο άλλο. Παρατηρείται στην έξοδο του jit.op @op max ο τελικός πίνακας.

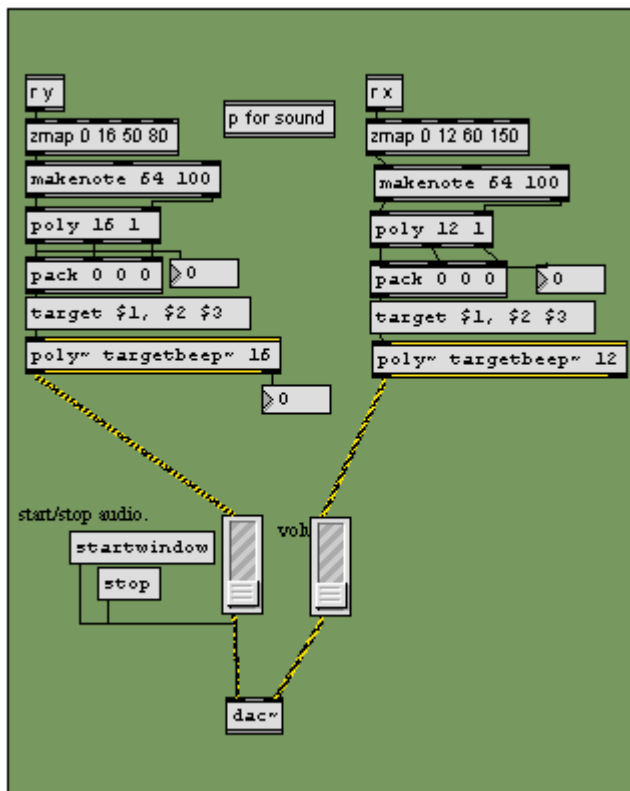
#### 2.1.4 Αλγόριθμος σύνθεσης ήχου

Για τη δημιουργία του ήχου, χρησιμοποιηθήκε η λίστα που εξάγεται από τον πίνακα ανίχνευσης. Αυτή η λίστα περιέχει τις συντεταγμένες της κίνησης που ανιχνεύθηκαν σε ένα εύρος 16 τιμών για την X συντεταγμένη και 12 τιμών για την Y συντεταγμένη. Κάθε νέα τιμή που εξάγεται διεγείρει δύο ημιτονοειδής γεννήτριες, μία για κάθε X και μία για κάθε Y, που παράγουν τόνους των οποίων η τονικότητα εξαρτάται από τη συντεταγμένη στην οποία ανιχνεύθηκε η κίνηση. Ακολουθεί ο αλγόριθμος αυτής της γεννήτριας.

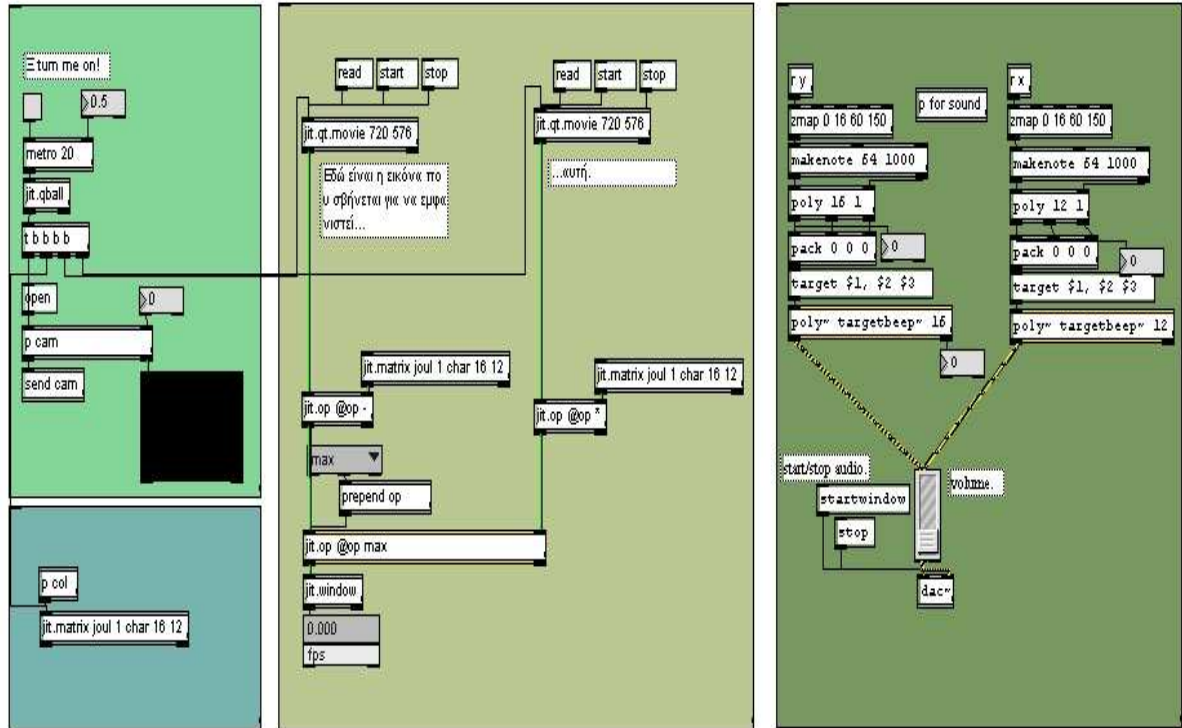


Εικ. 29

Για να αντιστοιχίσουμε την κάθε τιμή συντεταγμένης σε μία τονικότητα, εισάγουμε τη γεννήτρια αυτή σε δύο **poly~** αντικείμενα. Ένα για τις X συντεταγμένες (16 φωνές) και ένα για τις Y (12 φωνές). Με αυτό τον τρόπο είναι δυνατό, σε περιπτώσεις γρήγορης κίνησης του υποκειμένου, να ακουστούν όλες οι νότες μαζί παράγοντας συγχορδίες.



Εικ. 20



Εικ.21 Ολόκληρο το patch της εφαρμογής

## 2.2 Μικρές ιστορίες πόλεων

Οι “Μικρές ιστορίες πόλεων” αποτελούν μια οπτικοακουστική διαδραστική εγκατάσταση για το αστικό τοπίο με αφορμή την τέχνη του δρόμου. Η σύλληψη της ιδέας, η θεματολογία και ο τρόπος παρουσίασης της στο έργο καθώς και η παραγωγή του οπτικού υλικού είναι δουλειά της τελειόφοιτου φοιτήτριας του τμήματος ΤΠΤΕ Χρήστου Ευαγγελίας, της οποίας η εγκατάσταση και το καλλιτεχνικό της υπόβαθρο αποτελεί μεταπτυχιακή διατριβή. Στο πλαίσιο της υλοποίησης της εγκατάστασης μου ζητήθηκε να αναλάβω τον ηχητικό σχεδιασμό και την υλοποίηση των αλγορίθμων: ανίχνευσης, σύνθεσης εικόνας και αναπαραγωγής του ήχου.

Το οπικόακουστικό υλικό που παρουσιάζεται στην εγκατάσταση είναι αποτέλεσμα συνεργασίας, από πλευράς μου στη σύνθεση του ήχου και της συνεργάτιδος μου ως δημιουργό video-animation. Θα εξεταστεί εδώ η δική μου καταβολή στο έργο, που απασχολεί αυτή την εργασία, και που αποτελείται από το τεχνικό κομμάτι και τον ηχητικό σχεδιασμό. Το καλλιτεχνικό υπόβαθρο της εγκατάστασης επεξηγείται αλλού και δεν ανήκει σε μια εργασία που ασχολείται με το προγραμματιστικό κυρίως μέρος τέτοιων εγκαταστάσεων.

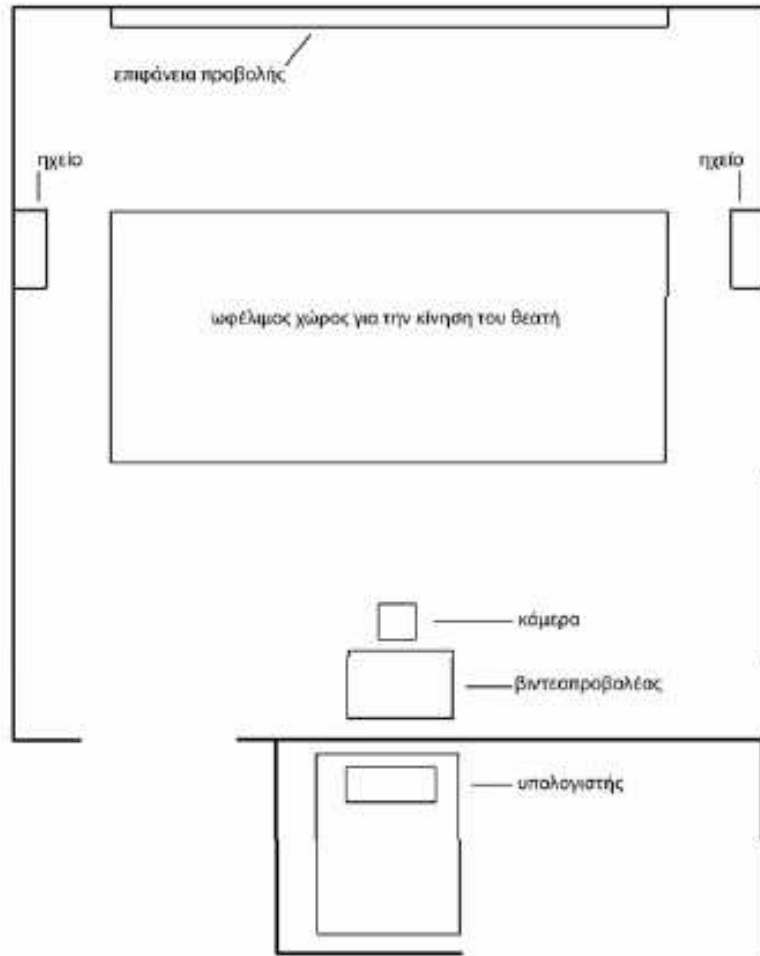
### 2.2.1 Περιγραφή

Στην εικόνα 22 παρατηρούμε τη κάτοψη του χώρου της εγκατάστασης. Κατά την είσοδο του στο χώρο της έκθεσης, ο θεατής έχει αμέσως τη δυνατότητα να δει μια εικόνα video που προβάλλεται στον τοίχο. Στην ουσία, η εικόνα αυτή είναι ένα animation που εξελίσσεται πολύ αργά στο χρόνο. Παράλληλα με την κινούμενη εικόνα, στο χώρο διαχέεται κι ένας ήχος-βόμβος, που σε συνδυασμό με αυτήν, λειτουργούν ως το πρώτο επίπεδο του έργου.

Πλησιάζοντας ο θεατής προς το κέντρο του χώρου, μια κάμερα, που βρίσκεται τοποθετημένη αντικριστά από την επιφάνεια της προβολής λαμβάνει την κίνηση, και πιο συγκεκριμένα, την κίνηση της σκιάς του θεατή. Ο χώρος είναι "μοιρασμένος" σε τρία ίσα μέρη-συντεταγμένες.

Κινούμενος ο θεατής σε συγκεκριμένα σημεία - κόμβους, τα οποία καθορίζονται σε σχέση με την κάμερα, ενεργοποιεί το/τα animation, τα οποία προβάλλονται πάνω στην αρχική εικόνα και παράλληλα τον ήχο που συνοδεύει το κάθε ένα. Τα animation είναι τρία και το καθένα προβάλλεται σε συγκεκριμένη θέση πάνω στην αρχική εικόνα, σαν τρίπτυχο (Εικόνα 23). Αν στο χώρο υπάρχει μόνο ένας θεατής και βρίσκεται ακίνητος, ενεργοποιεί ένα animation ανάλογα με τη θέση που βρίσκεται. Αν ο θεατής αρχίσει να κινείται στο χώρο παράλληλα με την προβολή, τότε ενεργοποιεί και το αντίστοιχο animation ανάλογα με τη θέση που βρίσκεται κάθε φορά. Αν οι θεατές είναι πάνω από δύο, ενεργοποιούνται και τα τρία animation.





Εικ. 22



Εικ. 23

Συνοπτικά, θα μπορούσαμε να πούμε ότι βασικό στοιχείο για την ενεργοποίηση των animation δεν αποτελεί μόνο η φυσική παρουσία του θεατή στο χώρο, αλλά και η περιπλάνηση του σε αυτόν. Ο θεατής καλείται να περιπλανηθεί στο χώρο για να ανακαλύψει τις τρεις “Μικρές ιστορίες πόλεων” και, αφού τις ανακαλύψει, μπορεί να επιλέξει - πάντα με την κίνηση του στο χώρο - ποια θα παρακολουθήσει. Στην περίπτωση που οι θεατές είναι περισσότεροι από δύο, τότε καλούνται όχι μόνο να ανακαλύψουν και να επιλέξουν, αλλά και να συνθέσουν τις ιστορίες μεταξύ τους εφόσον δίνεται η δυνατότητα να προβάλλονται δυο ή τρεις παράλληλα. Ο ήχος των animation είναι σχεδιασμένος έτσι ώστε ο καθένας από μόνος του να λειτουργεί αυτόνομα ως μουσικό κομμάτι, αλλά και οι δύο ή οι τρεις μαζί να συνθέτουν έναν καινούριο ήχο.

Τα ζητήματα που προκύπτουν στη συγκεκριμένη εγκατάσταση για τον προγραμματιστή είναι ο συγχρονισμός του ήχου με το video για το κάθε ένα ξεχωριστά αλλά και για το σύνολο καθώς η μουσική έχει έντονα ρυθμικά στοιχεία. Με αυτό το παράδειγμα θα έχουμε την ευκαιρία να δούμε πως μπορούμε να προβάλουμε ένα πίνακα πάνω από έναν άλλο, την ταυτόχρονη ενεργοποίηση του ήχου(στο MSP) και πως με τη μέθοδο blob detection κάνουμε πράξη το διαδραστικό κομμάτι. Θα εξετάσουμε το κάθε στάδιο του αλγόριθμου υλοποίησης ξεχωριστά καθώς η τελική του μορφή είναι αρκετά περίπλοκη.

### **2.2.2 Αλγόριθμος ανίχνευσης**

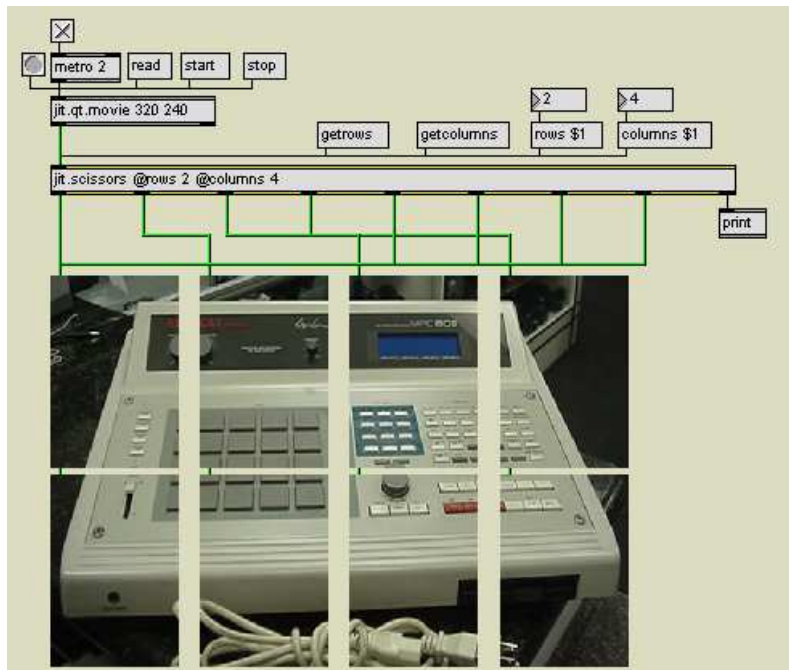
Ας ξεκινήσουμε από το στάδιο της ανίχνευσης. Θέλουμε να απομονώσουμε τρεις περιοχές στο X άξονα, ακριβώς κάτω από την περιοχή που αναπαράγονται τα τρία video, και να διαπιστώνουμε κάθε στιγμή αν αυτός ο χώρος περιέχει υποκείμενο-θεατή. Σαν αποτέλεσμα για μετέπειτα εύκολη διαχείριση ζητούμε να εξάγουμε τρία flags που θα υποδηλώνουν την ύπαρξη ή μη του θεατή. Ας δούμε πως επιτυγχάνεται αυτό με τη μέθοδο κατάτμησης της εικόνας.

Τα ξεχωριστά, για το εγχείρημα, αντικείμενα jitter που θα χρησιμοποιήσουμε είναι:

### **jit.scissors**

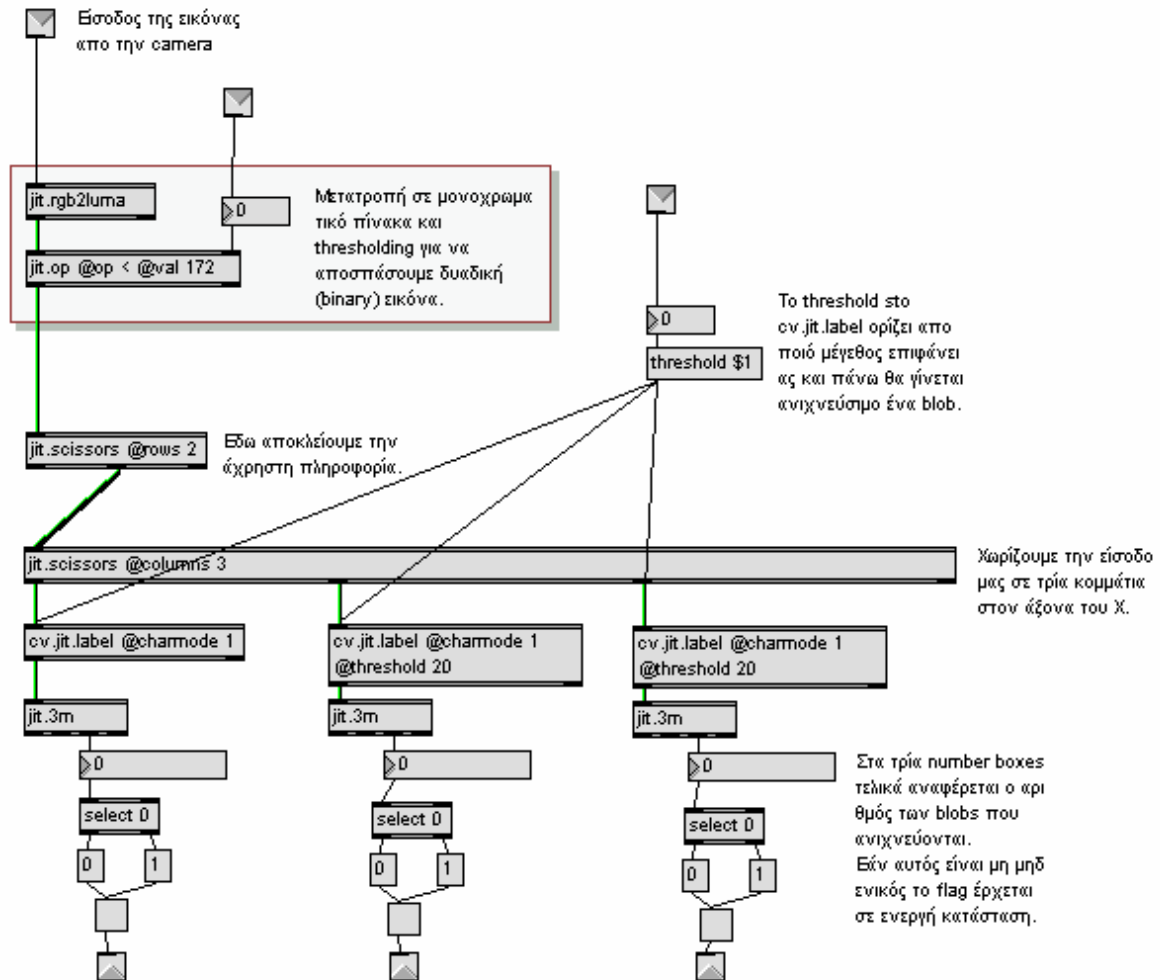


Το jit.scissors δέχεται στην είσοδο ένα πίνακα δύο διαστάσεων (2d) και τον χωρίζει σε συγκεκριμένο αριθμό στηλών και σειρών ίδιων διαστάσεων. Ο αριθμός των εξόδων είναι όσος και οι υποπεριοχές του πίνακα εισόδου που έχει οριστεί στο αντικείμενο να χωρίσει.



Εικ.24 Το mpc σε κομμάτια

Το αντικείμενο αυτό χρησιμοποιείται στον αλγόριθμο για τμηματοποίηση των περιοχών που εξετάζουμε. Η κάμερα λαμβάνει όλη την περιοχή του projection. Με το πρώτο jit.scissors αποκλείουμε το σήμα από τη μέση και πάνω όπου αναπαράγονται τα video. Με το δεύτερο χωρίζουμε την ωφέλιμη περιοχή σε τρία κομμάτια στον άξονα του X ώστε να εξετάσουμε το κάθε ένα ξεχωριστά για blobs.

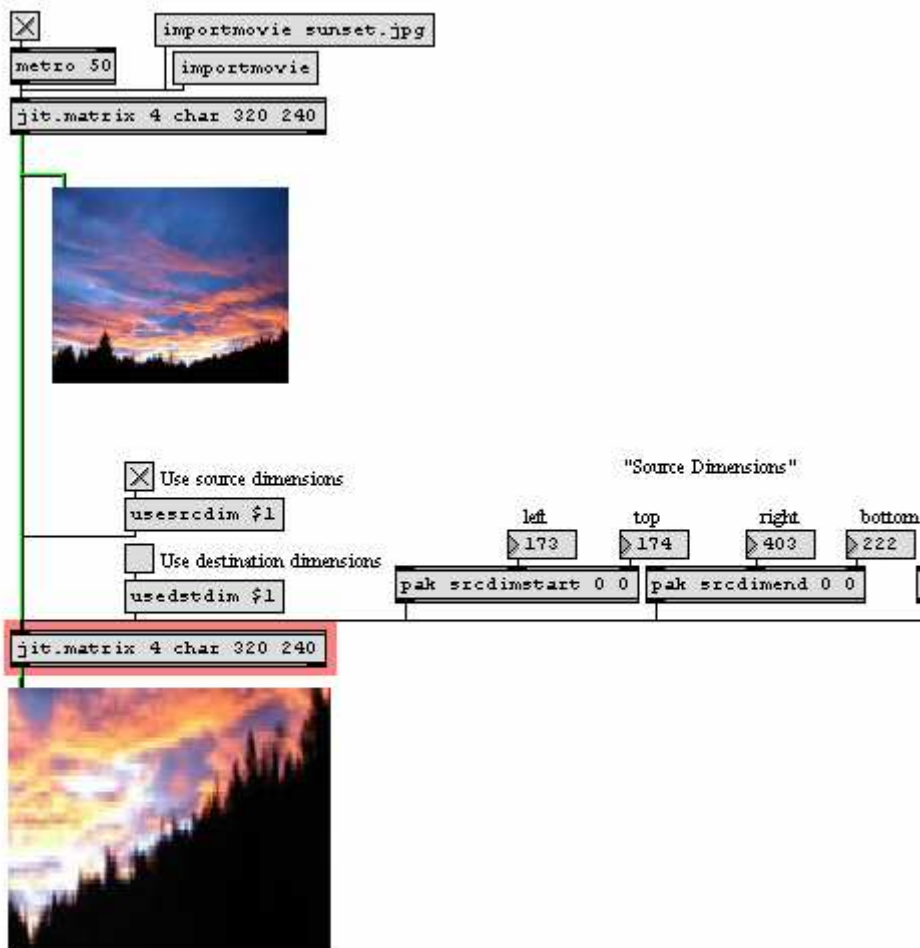


Εικ.25

Στη συνέχεια θα δούμε πως μπορούμε να προσδιορίσουμε τη διάταξη των video πάνω στην επιφάνεια της τελικής μας εικόνας. Θα εξηγηθούν οι τεχνικές σε ξεχωριστά patch για λόγους απλότητας και ευκολότερης κατανόησης αυτής της πιυχής του jitter που είναι λίγο πιο σύνθετη απ'ότι φαίνεται.

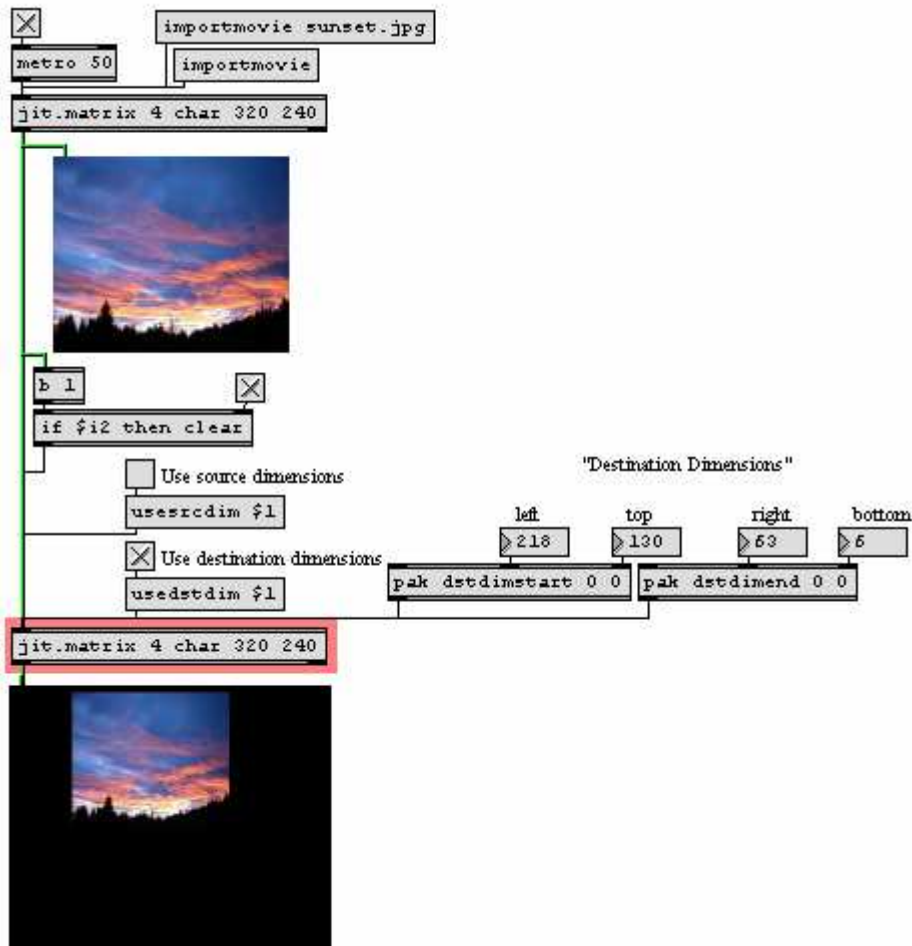
### 2.2.3 Διάταξη πινάκων στο χώρο

Ας δούμε εξ αρχής πως μπορούμε να πάρουμε ένα πίνακα ή μέρος του και να τον επανατοποθετήσουμε σε έναν άλλο πίνακα. Το jitter δεν έχει κάποιο συγκεκριμένο object για αυτή την επεξεργασία αλλά χρησιμοποιεί εσωτερικές μεταβλητές στο ίδιο το jitter.matrix. Αυτές οι μεταβλητές είναι το **srcdim** και το **dstdim**. Το srcdim ως λέξη προέρχεται από το source dimension (διάσταση πηγής). Το dstdim από το destination dimension (διάσταση προορισμού). Όπως μπορεί να καταλάβει κανείς, το πρώτο αναφέρεται σε ένα κομμάτι του πίνακα εισόδου που αποτυπώνεται στην έξοδο σε όλη τη διάσταση του πίνακα και το δεύτερο χρησιμοποιεί όλο τον πίνακα εισόδου και τον επανατοποθετεί σε ένα μικρότερο κομμάτι του πίνακα εξόδου. Το παράδειγμα που ακολουθεί θα κάνει πιο σαφή τα παραπάνω.



Εικ. 26

Βλέπουμε στο παραπάνω παράδειγμα (Εικ.26) πώς με τη χρήση του `srcdim` έχουμε προσδιορίσει μια περιοχή του πίνακα εισόδου να εμφανιστεί σε όλη τη διάσταση του πίνακα εξόδου. Αυτό δημιουργεί ένα εφέ `pixelation` λόγω του ότι ο πίνακας που βλέπουμε προσπαθεί να απεικονίσει μια υποπεριοχή του πίνακα εισόδου με τις ίδιες διαστάσεις στον πίνακα εξόδου.



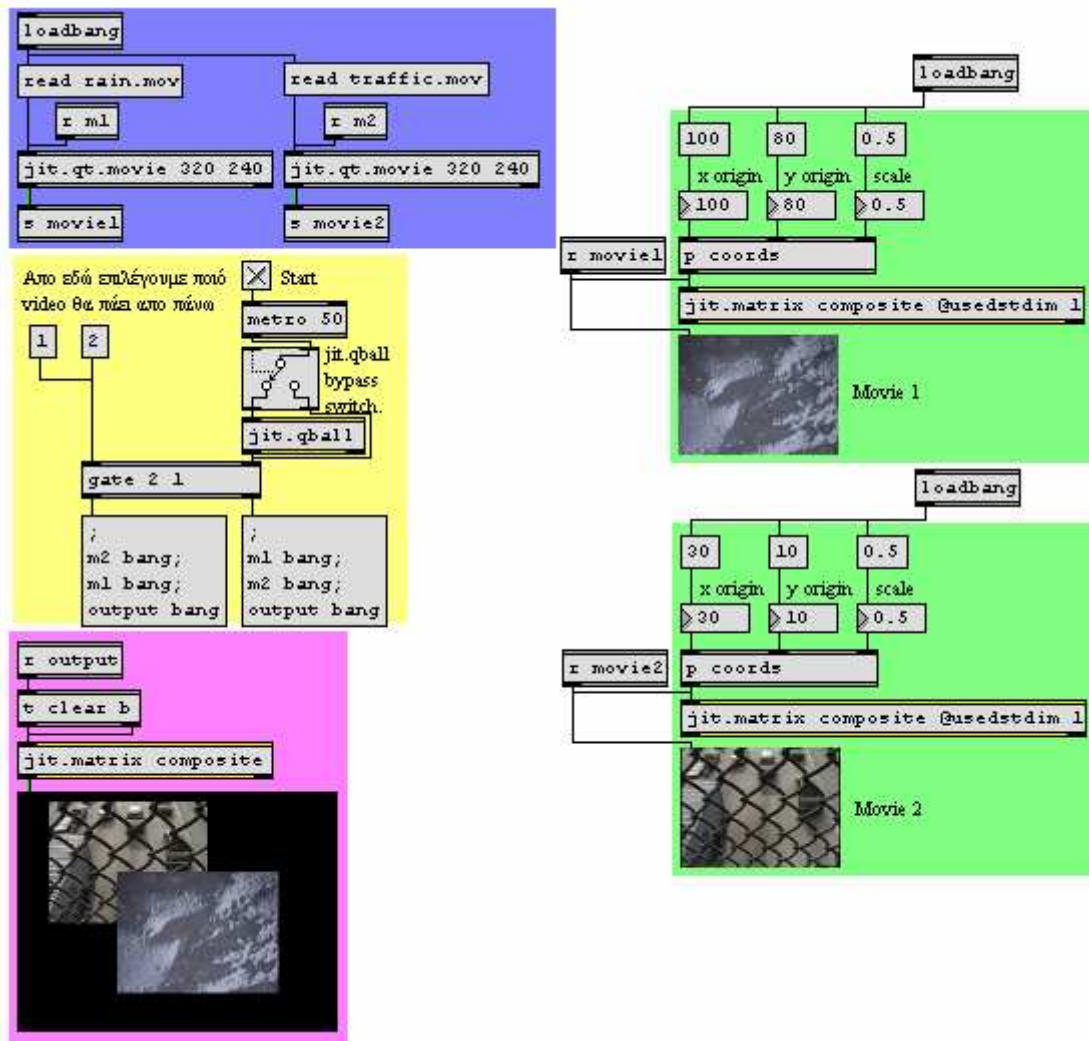
Εικ. 27

Σε αυτή την περίπτωση(Εικ.27) βλέπουμε πως λειτουργεί το `dstdim`. Χρησιμοποιεί τον πίνακα εισόδου και τον ξαναζωγραφίζει σε μια περιοχή μέσα στον πίνακα εξόδου.

#### 2.2.4 Ιεραρχία ροής δεδομένων

Η δεύτερη τεχνική που χρησιμοποιείται σε αυτή την εγκατάσταση θα μας δείξει πως μπορούμε να χρησιμοποιήσουμε τη μεταβλητή ονόματος του `jit.matrix` για να αποθηκεύσουμε στοιχεία από δύο ή παραπάνω πίνακες σε

έναν τελικό. Θα χρησιμοποιηθεί ένα patch από τα tutorials του jitter για την ευκολότερη κατανόηση αυτής της τεχνικής.



Εικ. 28

Στην μπλε περιοχή του patch διαβάζονται τα video από τα δύο jit.qt.movie αντικείμενα. Σε αυτή την περίπτωση βλέπουμε ότι μπορούμε να μεταφέρουμε video δεδομένα ασύρματα(χωρίς patchcord).

Στην κίτρινη περιοχή έχουμε το metro αντικείμενο που ενεργοποιεί την ροή των δεδομένων σε όλο το patch. Το **toggle** αντικείμενο είναι σε ενεργή κατάσταση και μπορούμε να δούμε το αποτέλεσμα στο jit.pwindow της ροζ περιοχής. Βλέπουμε ότι το metro αντικείμενο δεν είναι απευθείας συνδεδεμένο σε κάποιο jit.qt.movie αντικείμενο αλλά οδηγείται σε ένα **gate** αντικείμενο που το οδηγεί σε ένα από τα δύο message boxes.

Όποιο από τα δύο message boxes είναι επιλεγμένο αλλάζει τη διάταξη των video στον τελικό μας πίνακα(Εικ. 29).



Εικ. 29

Τα δύο object boxes στέλνουν τα ίδια τρία bang μηνύματα στα αντίστοιχα τρία receive αντικείμενα. Αυτό που αλλάζει είναι η σειρά με την οποία τα στέλνουν. Το αριστερό message box στέλνει το μήνυμα bang στο receive αντικείμενο m2, μετά στο receive αντικείμενο m1 και τέλος στο receive αντικείμενο output.

Αυτή η διαδικασία οδηγεί το δεξί αντικείμενο jit.qt.movie να εξάγει ένα πίνακα, ακολουθεί το αριστερό αντικείμενο jit.qt.movie και τέλος το jit.matrix στη ροζ περιοχή του patch δέχεται ένα bang που προκαλεί την εξοδό του πίνακα του.

Όταν τα jit.qt.movie αντικείμενα δεχτούν ένα bang, στέλνουν μέσω των send και receive αντικειμένων, τον πίνακα τους στα ονοματιζόμενα jit.matrix αντικείμενα.

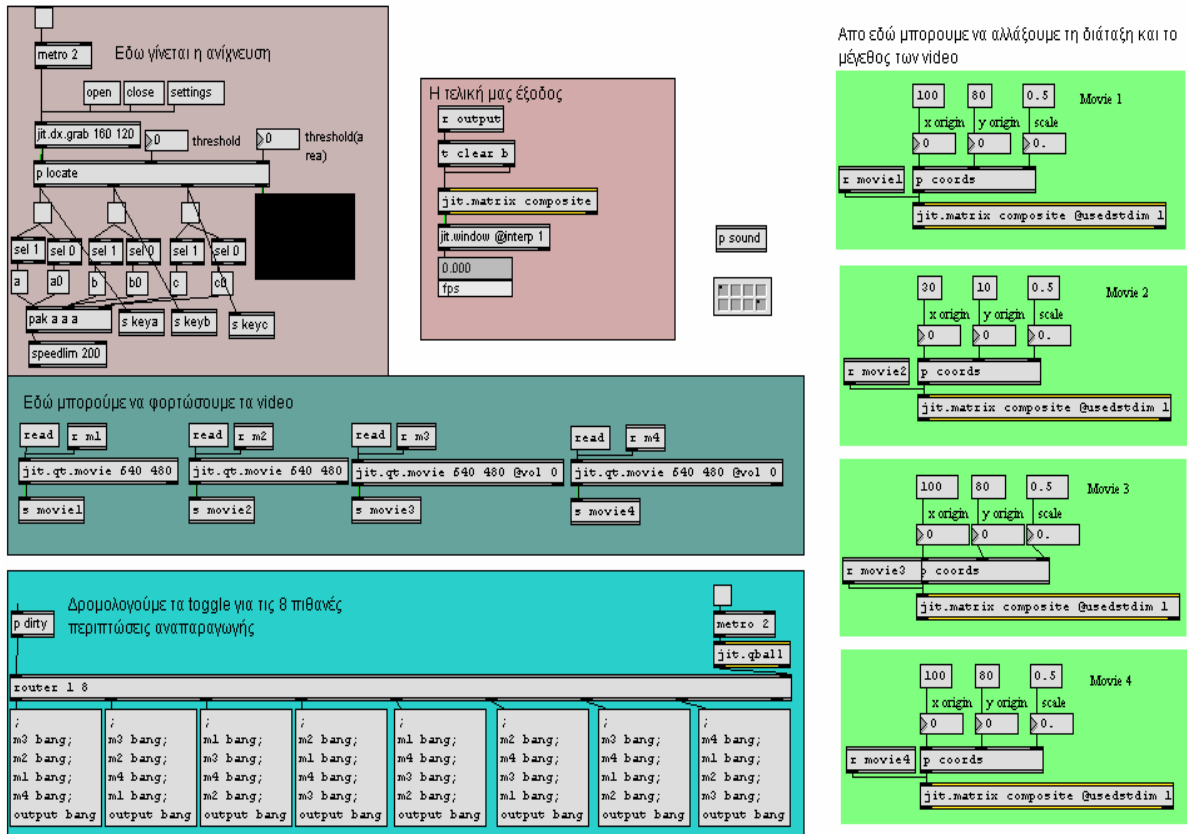
Βλέπουμε ότι, και τα τρία jit.matrix αντικείμενα έχουν το ίδιο όνομα "composite". Το αποτέλεσμα αυτής της διαδικασίας είναι και τα τρία αυτά αντικείμενα να περιέχουν τα ίδια στοιχεία πίνακα.

Εφόσον τα τρία matrix αντικείμενα αποθηκεύουν δεδομένα στον ίδιο πίνακα μπορούμε να καταλάβουμε γιατί έχει σημασία η σειρά των bang μηνυμάτων. Αν το αριστερό jit.qt.movie στείλει πρώτο τα δεδομένα του, αυτά θα γραφτούν στον πίνακα "composite". Στη συνέχεια τα δεδομένα του πίνακα του δεξιού jit.qt.movie θα γραφτούν στον ίδιο πίνακα. Αν τα δεδομένα από δύο πίνακες μοιράζονται τα ίδια pixels στον ίδιο πίνακα, αυτά που γράφονται τελευταία θα αντικαταστήσουν τα προηγούμενα.



## 2.2.5 Υλοποίηση

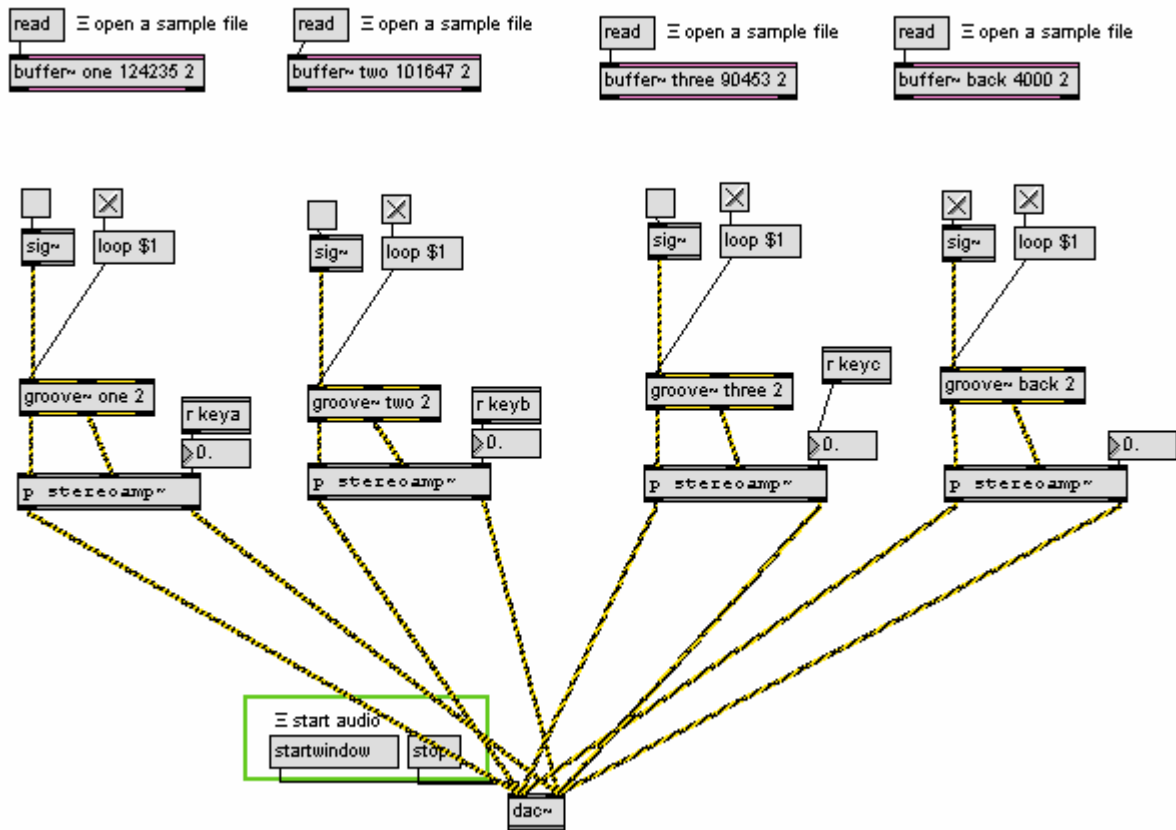
Παρακάτω βλέπουμε τον τελικό αλγόριθμο, που από το προϊόν της ανίχνευσης και μέσω των δύο τεχνικών που ερευνήθηκαν φέρνει εις πέρας τα ζητούμενα της εφαρμογής.



Εικ. 30

Αφού επιλεγούν τα αρχεία video και ανοίξουμε το metro αντικείμενο, αρχίζουν να αναπαράγονται τα video μαζί με τον ήχο. Στη παρούσα κατάσταση το patch προβάλλει το background με τον ήχο του. Μόλις ο αλγόριθμος ανιχνεύσει αντικείμενο αλλάζει τη σειρά που στέλνει τα bangs (στο κάτω αριστερά τμήμα του patch) για να κάνει ορατά τα video που αναλογούν στην ανίχνευση. Με αυτό τον τρόπο όταν ξεκινήσουμε το patch ενεργοποιείται η αναπαραγωγή σε όλα τα αρχεία μας (video και ήχου) και περιμένουν την εντολή για να «βγούν μπροστά» χωρίς να χάνουν το εσωτερικό συγχρονισμό τους καθώς παίζουν σε επανάληψη (loop).

## 2.2.6. Αλγόριθμος σύνθεσης ήχου



Εικ. 31

Τα τέσσερα **buffer~** αντικείμενα περιέχουν τα αρχεία ήχου προς αναπαραγωγή. Το κάθε ένα είναι δημιουργημένο για το αντίστοιχο video αλλά με τέτοιο τρόπο ώστε να μπορεί να αναπαράγεται και σε οποιοδήποτε συνδυασμό με τα άλλα αρχεία ήχου. Το τέταρτο, περιέχει το αρχείο που είναι προορισμένο να αναπαράγεται με το background όταν δεν έχει ανιχνευτεί υποκείμενο-θεατής. Όταν ενεργοποιείται το patch ξεκινάει η αναπαραγωγή σε όλα μέσω του αντικειμένου **groove~**. Με το μήνυμα **loop** εξασφαλίζουμε τα αρχεία να παίζουν σε επανάληψη. Χρησιμοποιούμε τα τρία flags απευθείας στην ένταση θέτωσης την είτε σε κατάσταση 1 (on-παρουσία θεατή) είτε σε κατάσταση 0 (off-απουσία θεατή).

### **2.2.7. Ηχητικός σχεδιασμός**

Ο σχεδιασμός της εγκατάστασης "Μικρές ιστορίες πόλεων" ήταν αρκετά περίπλοκος δεδομένου ότι τα στοιχεία της εικόνας, του ήχου και της διάδρασης ήταν άρρηκτα συνδεδεμένα μεταξύ τους κι έπρεπε να αντιμετωπιστούν με την ίδια βαρύτητα. Ένα επιπλέον επίπεδο δυσκολίας υπήρξε το γεγονός ότι το κάθε animation με τον ήχο του, έπρεπε να σχεδιαστεί με τέτοιο τρόπο, ώστε και να μπορεί να λειτουργήσει αυτόνομα αλλά και σε συνδυασμό με τα άλλα δύο animation. Ο σχεδιασμός ξεκίνησε από τα γραφικά του animation background και πριν ολοκληρωθούν, άρχισε να δημιουργείται και ο ήχος. Στα animation του 2ου επιπέδου, γραφικά και ήχος δημιουργήθηκαν σχεδόν παράλληλα. Οι πρώτες εικόνες έδωσαν το έναυσμα για την αρχική σύνθεση του ήχου ο οποίος με τη σειρά του καθόρισε την τελική μορφή των animation.

Ο ήχος στη συγκεκριμένη εγκατάσταση, αντιμετωπίστηκε ως αντικείμενο καθ' αυτό, πέρα από τα όρια της "μουσικής" με τη συμβατική έννοια. Ως πολυδιάστατο δυναμικό φαινόμενο στον χώρο που εξερευνά νέους τρόπους ερμηνείας και ως συγκινησιακός φορέας έκφρασης. Ως εκ τούτου, αποτέλεσε πεδίο πειραματισμού για την εξερεύνηση του ρόλου του.

#### **2.2.7.1.1° Επίπεδο-Εισαγωγή**

Πρόκειται για τον «πυρήνα» του έργου και την αρχική επαφή του χρήστη με την εγκατάσταση. Το βίντεο αυτό αποτελείται από τρεις στατικές εικόνες όπου η κάθε μία αντιστοιχεί σε μια πόλη (Αθήνα, Λονδίνο, Βερολίνο). Η κάθε εικόνα έχει δημιουργηθεί από φωτογραφικό υλικό που προέρχεται από την κάθε πόλη αντίστοιχα. Ένα είδος κολάζ συνθέτει διαφορετικά στοιχεία της κάθε πόλης για να δημιουργήσει την τελική εικόνα αυτής. Αυτές οι τρεις εικόνες εναλλάσσονται σταδιακά, δίνοντας οπτικά το αποτέλεσμα ενός μεταβαλλόμενου αστικού τοπίου. Σκοπός αυτής της μεταβολής δεν είναι η ανάδειξη της κάθε πόλης αλλά της γενικότερης έννοιας του αστικού τοπίου. Στο χώρο, σε συνδυασμό με το μεταβαλλόμενο αστικό τοπίο, διαχέεται ένας υπόκωφος βόμβος. Εδώ ο σκοπός είναι, ο ήχος να μην έχει κατευθυντικότητα και να γίνεται αντίληπτος ως περιβάλλον. Για να επιτευχθεί αυτό στη στερεοφωνική εικόνα, το ηχητικό περιεχόμενο του βόμβου βρίσκεται κυρίως στις χαμηλές συχνότητες. Επιπρόσθετα έχουν αντιστραφεί οι φάσεις μεταξύ του αριστερού και του δεξιού καναλιού που περιέχουν τον ίδιο ήχο.

#### 2.2.7.2.2ο Επίπεδο-Animations

Τα τρία animation που ενεργοποιούνται με την κίνηση του θεατή στο χώρο, διαπραγματεύονται και αυτά την έννοια του αστικού τοπίου αλλά αναφέρεται το κάθε ένα σε μια συγκεκριμένη πόλη. Τα στάδια ανάπτυξής του ήχου ήταν περίπλοκα. Σε συνδυασμό με τα στοιχεία της εικόνας αναπτύσσονταν, τόσο ο κάθε ήχος μόνος του όσο και σε συνδυασμό με τους άλλους. Η γενικότερη μορφή ολόκληρου του έργου, επηρέαζαν και διαμόρφωναν επαναλαμβανόμενα τη μορφή, τη δομή και το ύφος του.

##### A) Αθήνα

Για τη δημιουργία του animation της πόλης της Αθήνας, επιλέχτηκαν δύο έργα του ίδιου καλλιτέχνη (Αλέξανδρος Βασιμουλάκης). Η παρουσίαση των δύο αυτών έργων στο animation της εγκατάστασης γίνεται μέσα από μια μαύρη οθόνη με κινούμενα τετράγωνα μικρών διαστάσεων που λειτουργούν ως "μάσκα", αφήνοντας έτσι να αναπτυχθούν ζωγραφικές λεπτομέρειες και αξίες των δύο έργων. Η μέθοδος που χρησιμοποιήθηκε για τη δημιουργία του ήχου σε αυτό το animation ταιριάζει συνειρμικά με το animation. Ένα ηχητικό δείγμα κιθάρας που παίζει συγχορδίες με μεγάλες αξίες, και κομμένο σε κομμάτια (slices), αναπαράγεται συγχρονισμένα με το video σε μικρότερα ή μεγαλύτερα slices αφήνοντας τον ακροατή να ακούσει όλη τη διάρκεια ή ένα μικρότερο κομμάτι των συγχορδιών αυτών. Το ηχητικό αποτέλεσμα εδώ πρέπει να είναι πιο εύηχο ή ευχάριστο ωστέ να δημιουργεί αντίθεση, αλλά και να δουλεύει σε συνδυασμό, με αυτό του Λονδίνου.

##### B) Λονδίνο

Αφορμή για τη δημιουργία της συγκεκριμένης ιστορίας αποτέλεσε ένα stencil-graffiti επάνω στον τοίχο του Λονδίνου (Banksy) που διαπραγματεύεται μια παραλλαγή του γνωστού μύθου του Αισώπου "Ο μέρμηγκας κι ο τζίτζικας". Στην ουσία πρόκειται για μια αλληλουχία εικόνων που τόσο ο επαναλαμβανόμενος τρόπος συνύπαρξης τους όσο και ο ρυθμός που προβάλλονται δημιουργούν το οπτικό αποτέλεσμα μιας στρεσογόνας κατάστασης. Ο ήχος του Λονδίνου πλησιάζει περισσότερο στο να είναι αυτόνομο μουσικό κομμάτι ωστέ να υποστηρίξει τη λογική της αφήγησης μιάς ιστορίας με αρχή, μέση και τέλος. Εδώ το ηχητικό αποτέλεσμα πρέπει να είναι στρεσογόνο και να προκαλεί την αίσθηση της εγγρήγορης. Έχουν χρησιμοποιηθεί πολλά μικροδείγματα από θορύβους ενορχηστρωμένα σε γρήγορο τέμπο και αποδομένα σε ένα μουσικό στυλ τύπου drum and bass. Η μουσική αυτή πηγάζει από το αστικό τοπίο και ο συνειρμός που προκαλεί βοηθάει στη μετάδοση του νοήματος.

Γ) Βερολίνο

Στο συγκεκριμένο animation για την εικόνα επιλέχτηκαν δύο έργα ανώνυμων καλλιτεχνών που εναλλάσσονται με ρυθμούς άλλοτε γρήγορους και άλλοτε αργούς, δίνοντας μια αίσθηση συνέχειας και ασυνέχειας. Μαύρα καρτέ παρεμβάλλονται μεταξύ τους ως παύσεις-σιωπές. Οι αντιθέσεις του ασπρόμαυρου - χρωματιστού, αρσενικού - θηλυκού, δημιουργούν ένα είδος διάλογου. Ο ήχος της πόλης του Βερολίνου προέρχεται από αποσπάσματα συζήτησης ενός ζευγαριού. Τα δείγματα έχουν υποστεί επεξεργασία κυρίως τύπου pitch-shifting ή time stretching ώστε να δημιουργηθούν χροιές που ενώ παραπέμπουν σε ανθρώπινη φωνή το συναίσθημα απορέει όχι από την κατανόηση του λόγου αλλά από την πολυδιάστατη φύση του ήχου της ανθρώπινης φωνής. Το τονικό παιχνίδι του λόγου καθοδηγεί εδώ την αφήγηση.

#### **2.2.8.Μελλοντικές προεκτάσεις**

Μια μελλοντική προέκταση, θα μπορούσε να αφορά σε δύο σημεία.

Αρχικά, στην παρουσίαση, και κατ' επέκταση, αναδιαμόρφωση του έργου σε διαφορετικούς χώρους, όπου θα μας έδιναν τη δυνατότητα ανάπτυξης διαφορετικών περιβαλλόντων. Ο χώρος παίζει καθοριστικό ρόλο στη διαμόρφωση του έργου όχι μόνο εξαιτίας των φυσικών χαρακτηριστικών του, που επηρεάζουν τη μορφολογία του έργου, αλλά και λόγο των διαφορετικών σχέσεων - δικτύων που αναπτύσσονται στον εκάστοτε χώρο.

Σε μια δεύτερη προσέγγιση, θα μπορούσαν οι δημιουργοί να είναι περισσότεροι και να αποτελούν ένα δίκτυο. Η κάθε "μικρή ιστορία" για μια πόλη να προέρχεται από καλλιτέχνη ίσως της ίδιας πόλης. Στην περίπτωση αυτή, η αναδιαμόρφωση του έργου θα κρίνονταν απαραίτητη, καθώς η θέση του κέντρου του έργου θα εναλλάσσεται μεταξύ συμμετεχόντων. Το ερέθισμα το οποίο επιλέγεται για να ξεκινήσει ο διάλογος, διαφοροποιείται και καλεί κάθε ενδιαφερόμενο σε επικοινωνία. Σε αυτή την προσέγγιση, πρόκειται για τη συνειδητή συγκρότηση μιας ομάδας, ένα ενεργό δίκτυο συνεργασίας που αποφασίζει να εργαστεί από κοινού για ένα στόχο που δεν θα μπορούσε να πραγματοποιηθεί μεμονωμένα από κάθε μέλος της. Τα συλλογικά σχήματα παρέχουν ένα πλαίσιο συμπληρωματικών δεξιοτήτων σε διαφορετικά μέσα και τομείς γνώσης, δυνατότητες δράσης σε διαφορετικά πεδία πολιτισμού καθώς και δυνατότητες ταχύτατης απόκρισης στα νέα δεδομένα που διαμορφώνει η αστική κινητικότητα.

### 2.3. Διαδραστικά μπουκάλια

Αυτή η τρίτη και τελευταία εφαρμογή δεν προσφέρεται για διαδραστική σύνθεση εικόνας και ο χρήστης έχει τη δυνατότητα να διαδράσει μόνο με τον ήχο. Περιέχει όμως κάποιες τεχνικές για την ανίχνευση, πολύ χρήσιμες σε εφαρμογές cv, που αξίζει να εξετάσουμε στο περιβάλλον του jitter. Η πρώτη λέγεται **απάλειψη background** (background subtraction) και παρέχει τη δυνατότητα για ανίχνευση με κάμερα μπροστά από οποιοδήποτε στατικό background. Η δεύτερη τεχνική που θα εφαρμοστεί χρησιμοποιεί το **υπέρυθρο (ir)** φως για να δημιουργήσει ένα περιβάλλον ανίχνευσης ανεπηρέαστο από εξωτερικούς παράγοντες φωτισμού.

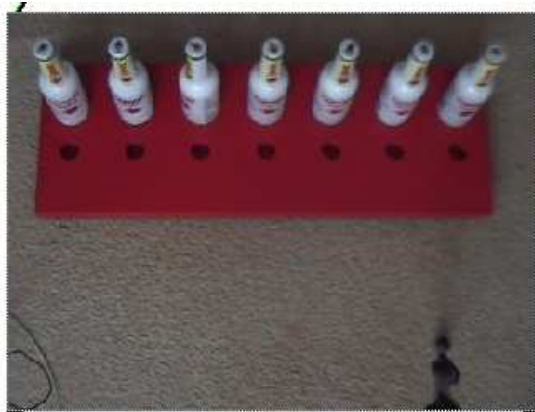
#### 2.3.1. Περιγραφή

Η ιδέα της εφαρμογής προέκυψε για λογαριασμό της **Freeframe productions** σχετικά με τη δημιουργία ενός διαδραστικού παιχνιδιού για τη διαφημιστική καμπάνια του Gordon's Space. Η ιδέα που επικράτησε ήταν να παραταχθούν 7 μπουκάλια σε ευθεία. (Εικ.32)



Εικ. 32

Το κάθε μπουκάλι αντιστοιχεί σε έναν ήχο. Ο χρήστης τοποθετώντας το χέρι του πάνω από το μπουκάλι προκαλεί την αναπαραγωγή αυτού του ήχου. Κατ' επέκταση, και λόγω της εμπορικότητας της εφαρμογής, ο κάθε ήχος αποτελεί το ένα από τα 7 κανάλια ενός μουσικού κομματιού που ο χρήστης έχει τη δυνατότητα να αναμίξει ώστε να δημιουργήσει τη δική του εκδοχή δομής του κομματιού. Θα τοποθετήσουμε την κάμερα πάνω και προς τα μπουκάλια για να ανιχνεύσουμε την ύπαρξη ή μη του χεριού. (Εικ.33)



Εικ. 33

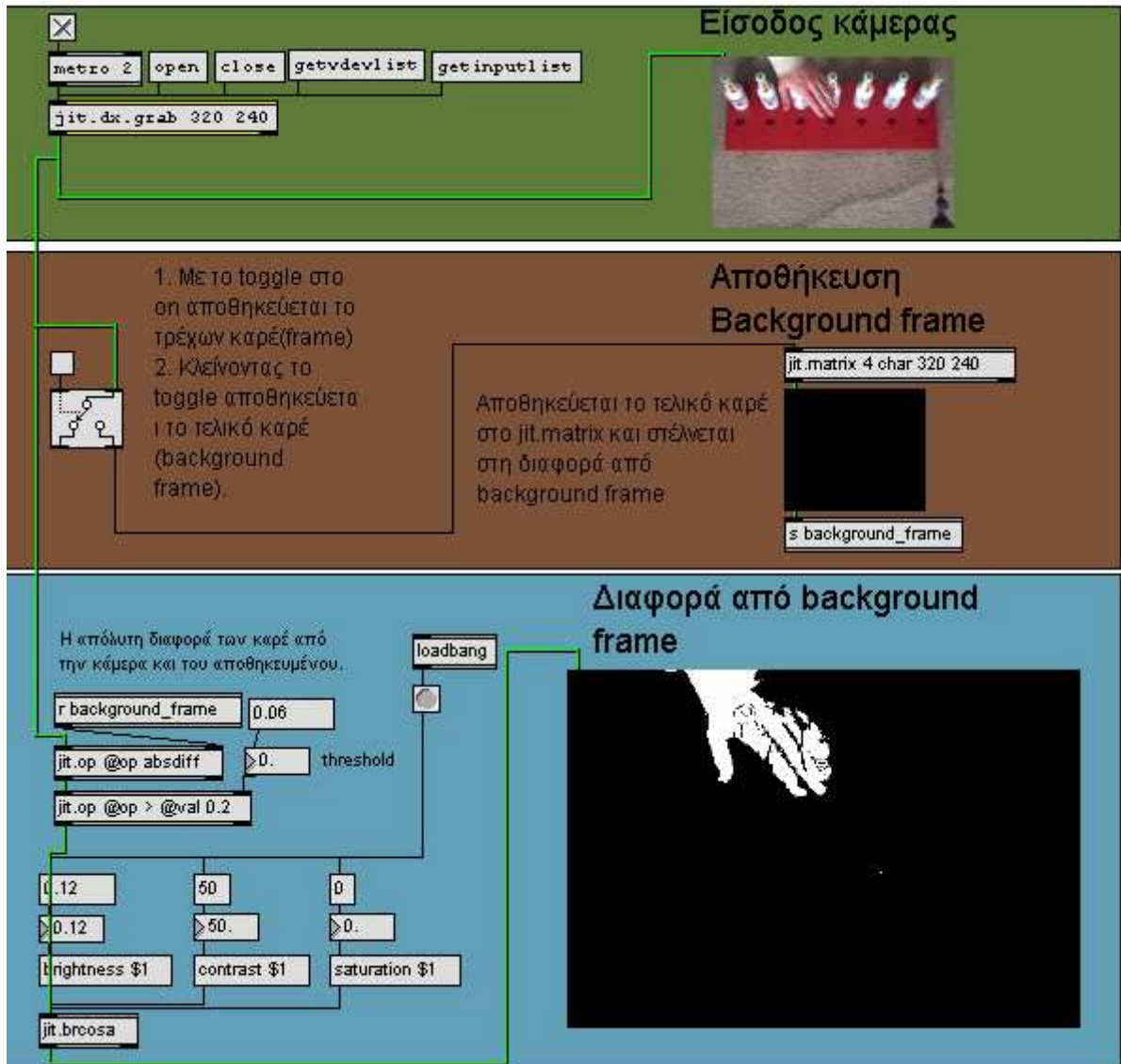
Στις προηγούμενες εφαρμογές το background ήταν το ίδιο το projection, που έχει μεγάλη τιμή φωτεινότητας, και με thresholding ανιχνεύσαμε τη σκιά που άφηνε το υποκείμενο. Εδώ, το κάδρο της κάμερας από το οποίο θα ανιχνεύσουμε, δεν έχει μεγάλες αποκλίσεις στη φωτεινότητα, με την είσοδο ή μη του χεριού, και χρειάζεται να εφαρμόσουμε άλλη επεξεργασία στο σήμα από την κάμερα πριν εφαρμόσουμε κάποια τεχνική ανίχνευσης. Επειδή η εφαρμογή υλοποιείται τελικά ως μια ενιαία κατασκευή εύκολα μεταφερόμενη, πρέπει να διασφαλιστεί ότι δουλεύει σε όλες τις φωτιστικές συνθήκες χωρίς πρόβλημα.

### **2.3.2.Υλοποίηση**

Η υλοποίηση του διαδραστικού μέρους της εγκατάστασης θα γίνει με δύο τρόπους που φέρνουν το ίδιο αποτέλεσμα. Θα εξετάσουμε τις δύο ξεχωριστά μέχρι το σημείο της ανίχνευσης γιατί ο αλγόριθμος της αναπαραγωγής και του συγχρονισμού του ήχου είναι όμοιος με αυτόν της εφαρμογής «Μικρές ιστορίες πόλεων».

#### **2.3.2.1. Background subtraction**

Η τεχνική της αφαίρεσης του background είναι πολύ διαδεδομένη σε εφαρμογές cv. Μας επιτρέπει να εφαρμόσουμε τις τεχνικές του cv σε οποιοδήποτε στατικό κάδρο της κάμερας αφαιρώντας το background και ανιχνεύοντας μόνο τα στοιχεία που μπαίνουν επιπρόσθετα σε αυτό. Ας δούμε όμως στην πράξη πως μπορεί να χρησιμοποιηθεί αυτό στην εφαρμογή μας.

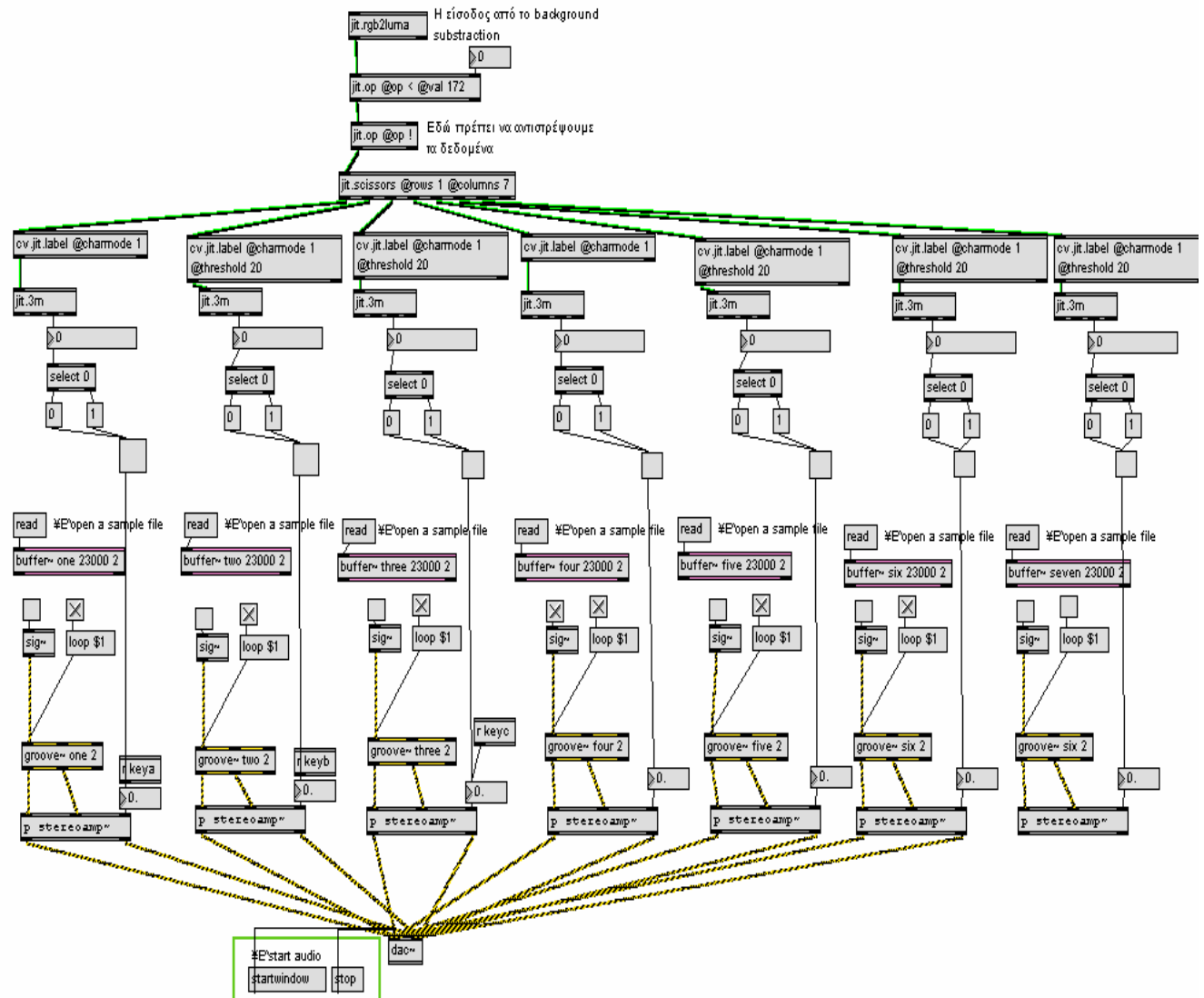


Εικ. 34

Στο παραπάνω patcher μπορούμε να δούμε πως γίνεται το background subtraction στο περιβάλλον του jitter με χρήση του jit.op. Αποθηκεύεται το καρέ(frame) που επιλέγεται για background και συγκρίνεται μέσω του **jit.op @op absdiff** (absolute difference), με τα καρέ που εισέρχονται από την κάμερα αποδίδοντας τη μεταξύ τους διαφορά. Το δεύτερο jit.op σε κατάσταση **>** (**bigger than**) μας προσφέρει μία δυνατότητα thresholding ώστε να ρυθμίζουμε το κατά πόσο πρέπει να αλλάξει η εικόνα για να έχουμε έξοδο. Φαίνεται στο jit.pwindow της μπλε περιοχής πως στο κάδρο υπάρχει μόνο το χέρι έτοιμο προς ανίχνευση της θέσης του.



Τώρα μένει να ανιχνεύσουμε τη θέση του χεριού στο x άξονα και να αντιστοιχίσουμε 7 σημεία σε 7 ήχους. Παρατίθεται ο αλγόριθμος χωρίς ανάλυση καθώς είναι ο ίδιος με αυτόν της εγκατάστασης «μικρές ιστορίες πόλεων». Χωρίζουμε τον άξονα x σε 7 σημεία και εφαρμόζουμε blob tracking.



Εικ. 35

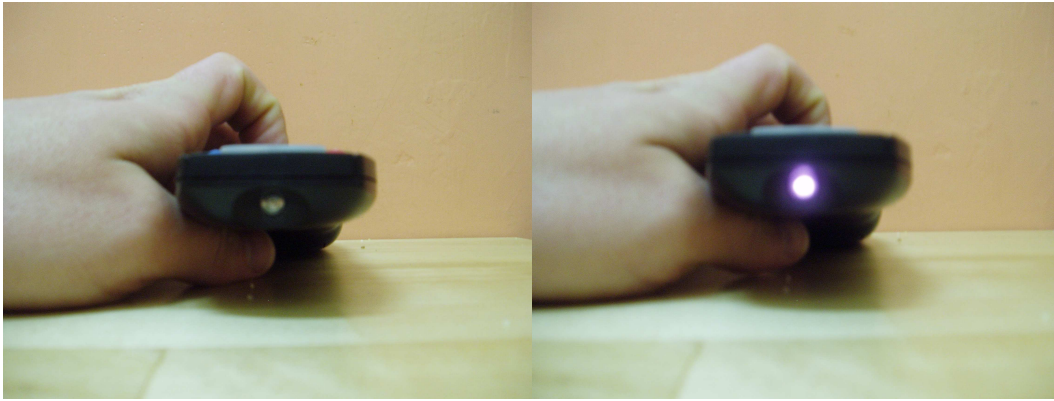
### 2.3.2.2.IR

Στην υπέρυθρη φωτογραφία, το φιλμ ή ο αισθητήρας εικόνας γενικότερα είναι ευαίσθητος στο υπέρυθρο φως. Το μέρος του χρωματικού φάσματος που χρησιμοποιείται αναφέρεται ως **near infrared** σε αντίθεση με το **far infrared** που χρησιμοποιείται στο τομέα της εικονογράφησης της θερμότητας (thermal imaging). Το μήκος κύματος του φωτός που χρησιμοποιεί η υπέρυθρη φωτογραφία κυμαίνεται από 700nm μέχρι τα 900nm. Συνήθως χρησιμοποιείται ένα φίλτρο που επιτρέπει το υπέρυθρο φως να περάσει στην κάμερα αλλά εμποδίζει όλο ή το μεγαλύτερο μέρος του ορατού φωτός. Οι φωτογράφοι χρησιμοποιούν αυτή την τεχνική, σε συνδυασμό με τα ευαίσθητα σε υπέρυθρο φως φιλμ, για να αποτυπώσουν φωτογραφίες που μοιάζουν με ασπρόμαυρες αλλά έχουν ασυνήθιστο χρώμα και ονειρική εμφάνιση. Αυτό το εφέ στη φωτογραφία λέγεται «the wood effect» από τον Robert Wood που πρώτος εφάρμοσε την τεχνική το 1910.



Εικ.36 Παράδειγμα υπέρυθρης φωτογράφισης

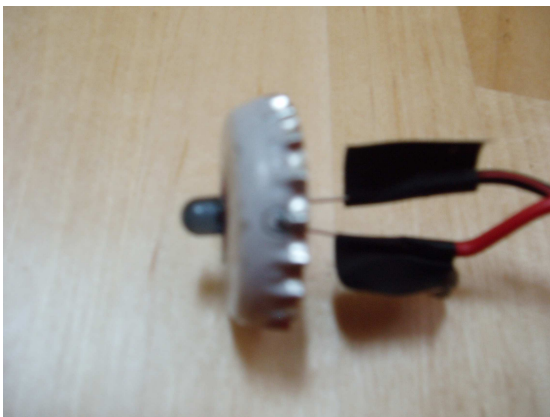
Είναι κοινώς γνωστό ότι το IR φως δεν μπορεί να το δει το ανθρώπινο μάτι. Αν όμως το κοιτάξει κανείς μέσα από μία κάμερα θα παρατηρήσει ότι φαίνεται ως ένα ανοιχτό μωβ.



Εικ.37

Στην αριστερή εικόνα βλέπουμε το αναμμένο ir led ενός συνηθισμένου τηλεκοντρόλ όπως φαίνεται από το γυμνό μάτι. Στα δεξιά βλέπουμε πως αυτό το φως φαίνεται στην κάμερα. Αυτό ισχύει για τις περισσότερες κάμερες με εξαίρεση αυτές που με φίλτρο εμποδίζουν τη διέλευση των υπεριώθρων.

Θα εκμεταλλευτούμε αυτό το φαινόμενο στην ανίχνευση, τοποθετώντας ένα ir led στο καπάκι κάθε μπουκαλιού. (Εικ.38)



Εικ.38

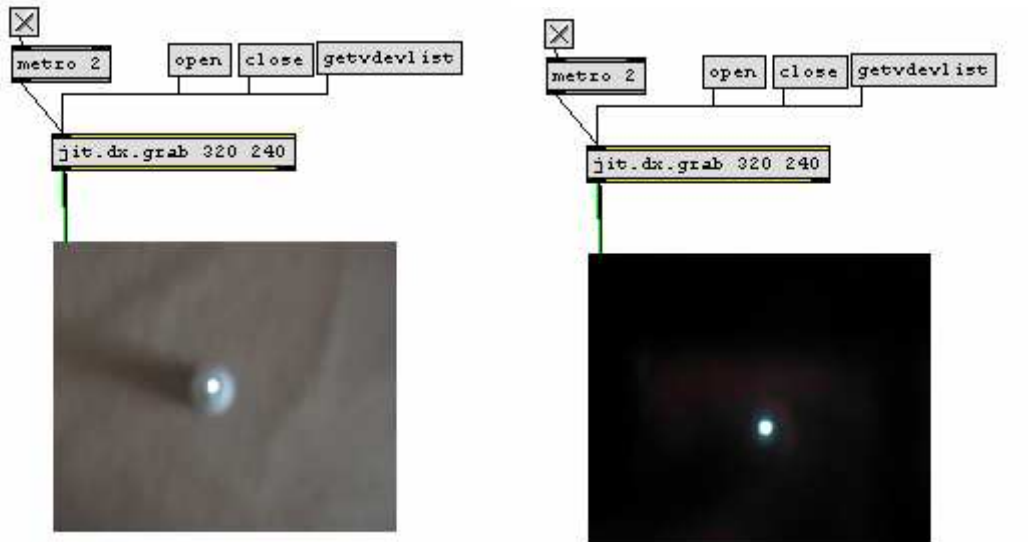


Εικ.39 Τροφοδοσία 1,5volt

Έχουμε μετατρέψει τώρα το καπάκι του μπουκαλιού σε μια πηγή υπέρυθρου φωτός. Αυτό που μένει είναι να εφαρμόσουμε στη κάμερα ένα φίλτρο που θα επιτρέπει τη διέλευση μόνο του υπέρυθρου φωτός. Αυτό το φίλτρο δεν είναι άλλο από το αρνητικό ενός φιλμ αναλογικής φωτογραφικής μηχανής. Χρειαζόμαστε από το αρνητικό όμως, μόνο αυτό το μέρος που είναι μαύρου χρώματος και βρίσκεται στην αρχή ή στο τέλος του φιλμ. Αυτό το φίλτρο όπως είπαμε εμποδίζει τη διέλευση όλου του φωτός πλην του υπέρυθρου.

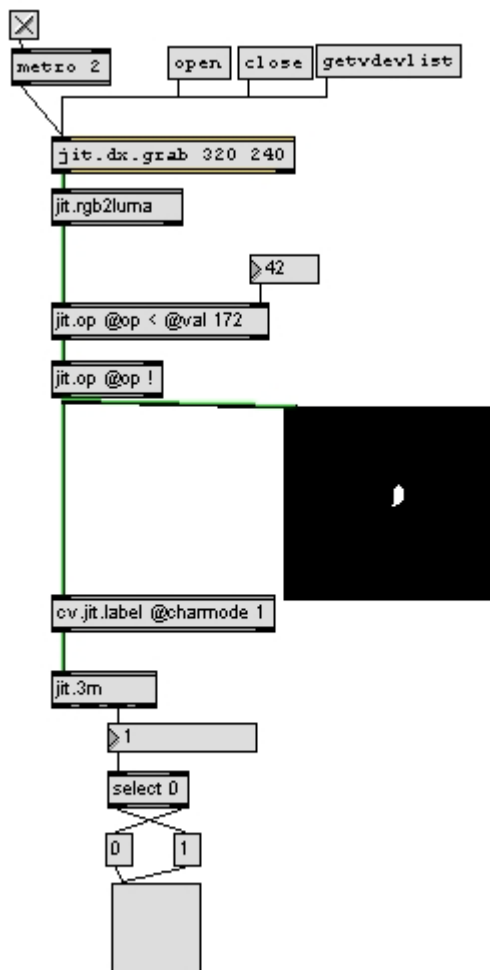


Εικ.40 Εφαρμόζουμε το φίλτρο και με μαύρη ταινία (tape) διασφαλίζουμε ότι δεν περνάει άλλο φως.



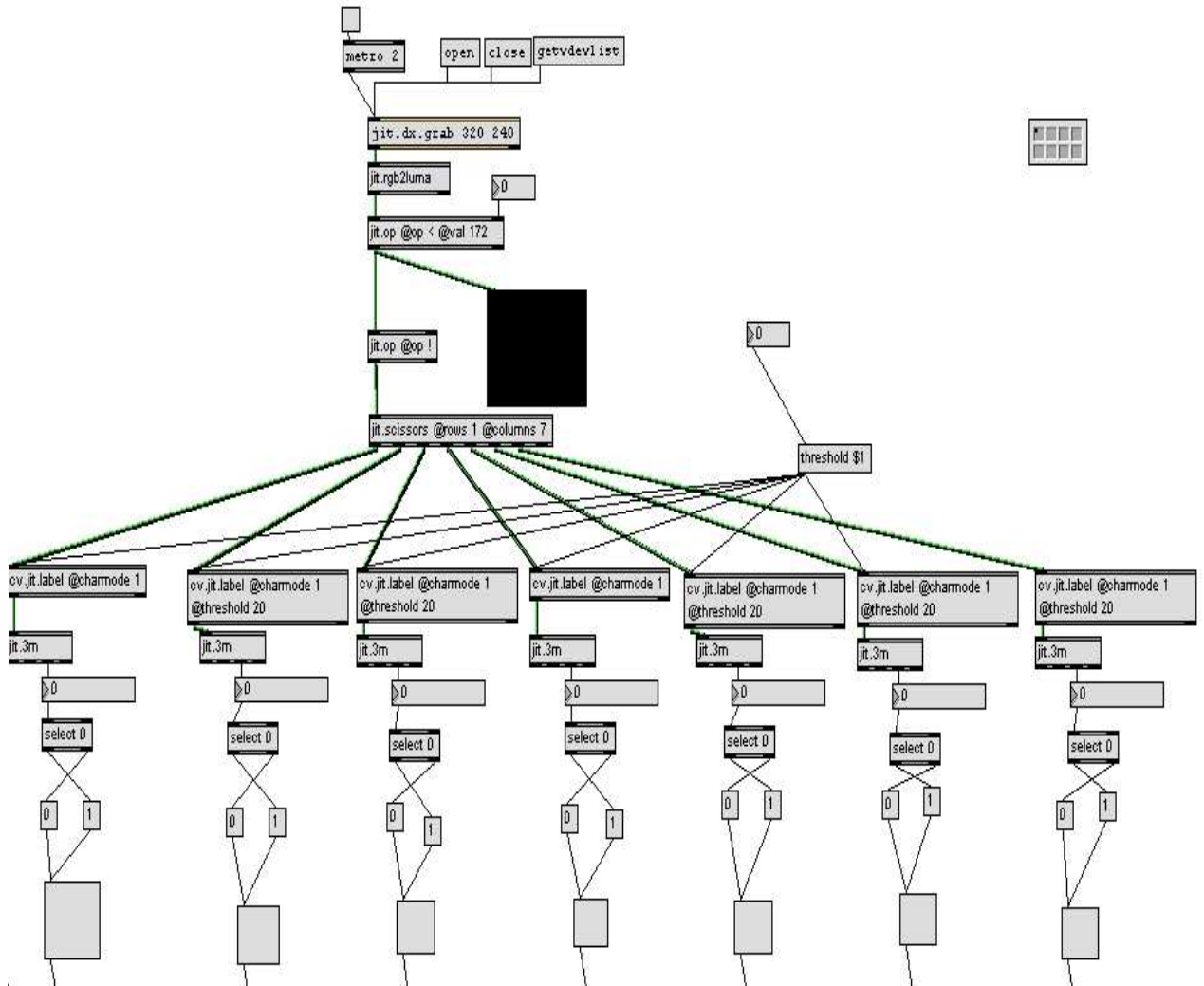
Εικ.41 Πριν και μετά

Στην παραπάνω εικόνα βλέπουμε πως λαμβάνεται η εικόνα του μπουκαλιού πριν και μετά την εφαρμογή του φίλτρου. Όλο το φως πλην του υπέρυθρου έχει φιλτραρισθεί καθιστώντας ορατό μόνο το led. Ο αλγόριθμος που ακολουθεί εφαρμόζει blob tracking στην φιλτραρισμένη εικόνα.



Εικ.42

Στην έξοδο του jit.3m, στο number box αναφέρεται ο αριθμός των blobs που ανιχνεύθηκαν και που τώρα είναι 1 λόγω του led. Όταν ο χρήστης σκεπάσει το led με το χέρι του ο αριθμός των blobs θα γίνει 0 θέτωντας το toggle σε 1. Για να ολοκληρωθεί ο αλγόριθμος θα χωρίσουμε το σήμα από την κάμερα σε 7 περιοχές, μία για κάθε μπουκάλι.



Εικ. 43

Κάνουμε χρήση του `jit.scissors` για να χωρίσουμε το σήμα από την κάμερα σε 7 περιοχές ανιχνεύοντας blobs σε κάθε μία. Τα toggle ενεργοποιούν τον ήχο με παρόμοιο τρόπο που εξετάσαμε και σε προηγούμενες εφαρμογές.

### 2.3.3. Ηχητικός σχεδιασμός-μελλοντικές προεκτάσεις

Το σκεπτικό πίσω από την εγκατάσταση δεν είναι κάτι καινούργιο και έχουν δημιουργηθεί πολλά παρόμοια όργανα από καλλιτέχνες με χρήση διάφορων ειδών αισθητήρων (sensors). Η καινοτομία εδώ είναι ο τρόπος με τον οποίο επιτυγχάνεται η διάδραση και όχι η ίδια η διάδραση. Τη χρησιμότητα του background subtraction σε πλήθος εφαρμογών μπορεί να την αντιληφθεί κανείς εύκολα. Η χρήση όμως των υπερύθρων σε συνδυασμό με το cv είναι καινούρια σχετικά τεχνολογία. Παραδειγμα εφαρμογής cv σε συνδυασμό με υπέρυθρο φως , με μεγαλύτερο όμως επίπεδο δυσκολίας, αποτελεί το **Multitouch interface** που χρησιμοποιείται στους υπολογιστές και επιτρέπει στο χρήστη να διαδρά με την εικόνα μέσω πολλαπλών σημείων επαφής πάνω στην οθόνη.

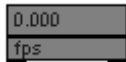
Όσον αφορά τον ηχητικό σχεδιασμό, στα πλαίσια της εφαρμογής και δεδομένου της διαφημιστικής χρήσης της, έχει δημιουργηθεί ένα κομμάτι εμπορικού χαρακτήρα. Η απεύθυνση της εγκατάστασης δεν προορίζεται σε μουσικούς και πρέπει να βοηθηθεί ο χρήστης ώστε να δημιουργήσει κάτι εύηχο που να μην έχει την πιθανότητα λάθους. Οι ήχοι έχουν χωριστεί σε επτά κανάλια. Ο τρόπος διάδρασης επιτρέπει ως επί το πλείστον να ακούγονται δύο κανάλια τη φορά οπότε το κυρίως μέλημα ,πέρα από το αισθητικό κριτήριο, είναι το κάθε κανάλι να περιέχει πλούσιο ηχητικό περιεχόμενο και να μη διακόπτει τη ροή του κομματιού εάν ακούγεται μόνο του. Οπότε το πρώτο κανάλι περιέχει, εκτός από τη μπότα, και το hi-hat. Το δεύτερο κανάλι περιέχει το snare με ένα ρυθμικό delay. Στο κανάλι του μπάσου υπάρχει και ένα δεύτερος ηλεκτρονικός ήχος με arpeggiator. Είναι προφανές ότι ο κάθε μουσικός θα μπορούσε να φτιάξει ένα κομμάτι προς εκτέλεση σε αυτό το όργανο για διαφορετικές χρήσεις.



## Κεφάλαιο 3: Συμπεράσματα

### 3.1 Μέτρηση απόδοσης μηχανήματος

Ο τρόπος δόμησης της εικόνας στο Jitter ως ένας πίνακας αριθμητικών δεδομένων. Είναι μια έξυπνη λογική και μπορεί να γίνει κατανοητή από ένα χρήστη εύκολα ώστε να αρχίσει να φαντάζεται τρόπους που θα μπορούσε να υλοποιήσει τις ιδέες του. Αυτό ακριβώς όμως, είναι και το μειονέκτημα σε αυτόν τον τρόπο λειτουργίας του Jitter. Όλες οι πράξεις που φέρνουν σαν αποτέλεσμα την εικόνα εκτελούνται στον επεξεργαστή(cpu), με αποτέλεσμα οι μεγάλες σε διαστάσεις εικόνες να καθιστούν τα patch πολύ απαιτητικά σε υπολογιστική ισχύ και ακόμα και οι δυνατότεροι υπολογιστές να μην μπορούν να τα τρέξουν. Για να γίνει κατανοητό αυτό θα κάνουμε χρήση ενός utility του jitter, το αντικείμενο **jit.fpsgui**.



Αυτό είναι ένας μετρητής της απόδοσης του μηχανήματος. Στην εσωτερική κατάσταση που θα το χρησιμοποιήσουμε θα μας δείξει την συχνότητα που προλαβαίνει να ανανεώσει τον τελικό πίνακα σε fps(frames per second) καρέ ανά δευτερόλεπτο. Το μηχάνημα που θα κάνει τη συγκριτική δοκιμή έχει επεξεργαστή dual-core intel 6600.



Εικ. 44

Σε περιβάλλον μίας απλής αναπαραγωγής video σε ανάλυση 320\*240 το μηχάνημα αποκρίνεται πολύ ικανοποιητικά καθώς ανανεώνει την εικόνα 254 φορές το δευτερόλεπτο.



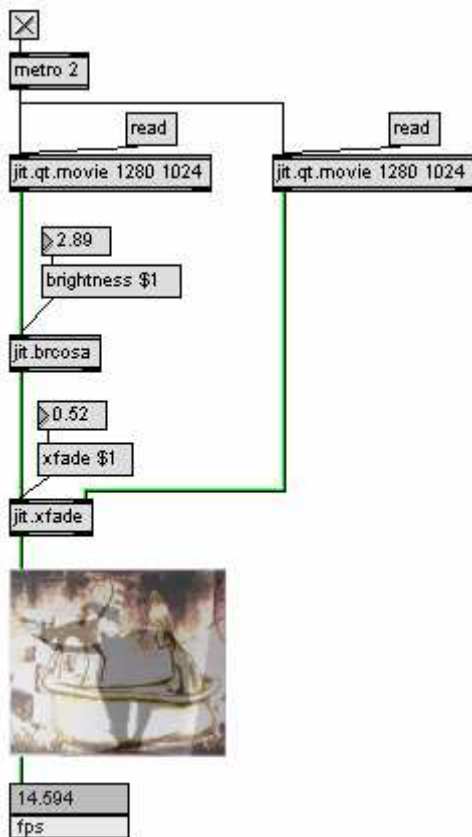
Εικ. 45

Αν διπλασιάσαμε την ανάλυση σε 640\*420, παρατηρούμε πως η απόδοση του μηχανήματος έχει πέσει περίπου στο μισό. Παρ'όλα αυτά αποκρίνεται πολύ ικανοποιητικά.



Εικ. 46

Σε αυτό το παράδειγμα χρησιμοποιούμε μία ανάλυση εικόνας 1280\*1024, που καλύπτει οποιοδήποτε είδος προβολής, ακόμα και προβολή εικόνα από projector σε ένα επαγγελματικό περιβάλλον. Το μηχάνημα ακόμα ανανεώνει τον πίνακα περισσότερες φορές από το μέσο όρο του τυπικού video (25 fps). Ας δούμε τώρα πως θα συμπεριφερθεί το μηχάνημα μετά από δύο στάδια επεξεργασίας πολύ τυπικά στις εικόνες.



Εικ. 47

Το `jit.brcosa` αντικείμενο παρέχει ρυθμίσεις για τη φωτεινότητα (`brightness`), την αντίθεση (`contrast`) και τον κορεσμό (`saturation`) του χρώματος. Το `jit.xfade`, αναμιγνύει οπτικά δύο εικόνες. Η έξοδος του μηχανήματος έχει πέσει λίγο κάτω από το ελάχιστο επιτρεπτό όριο, όπου το ανθρώπινο μάτι μπορεί να αντιληφθεί την εναλλαγή των καρτέ (15fps).

Σε ένα πραγματικό περιβάλλον προγραμματισμού `Jitter` οι επεξεργασίες που θα θέταμε σε ένα πίνακα θα ήταν πιο πολλές και ίσως πιο δύσκολες για τον επεξεργαστή. Σε αυτή την περίπτωση θα πρέπει να μειώσουμε το μέγεθος του πίνακα μας, εκτός αν έχουμε να διαθέσουμε πολλά χρήματα σε επεξεργαστή. Ο προγραμματιστής `Jitter` πρέπει να ξοδεύει πολύ χρόνο ωστέ να βρει τις παραδοχές που πρέπει να γίνουν σε ένα `patch` ώστε να γίνει λειτουργικό και εκτελέσιμο στο `hardware` που διαθέτει. Για αυτό το λόγο θα αναφερθούμε σε άλλα `software` που ενώ παρέχουν πολλές δυνατότητες για τη δημιουργία διαδραστικών περιβάλλοντων (`interactivity`) έχουν αναπτυχθεί με άλλη λογική από αυτή της απεικόνισης αριθμητικών τιμών πίνακα.

### **3.2 Η ζωή δεν είναι μονόδρομος.**

#### **vnnv**

Το vnnv είναι μία ανοιχτή βιβλιοθήκη εργαλείων σύνθεσης Video (Video synthesis) σε πραγματικό χρόνο, με γραφικό περιβάλλον προγραμματισμού. Μπορεί να επεξεργαστεί πολλών ειδών media που χρησιμοποιούνται στους υπολογιστές όπως ήχο, video, internet, 3d animation και πολλά άλλα. Στηρίζει τη λειτουργία του στα spreads, κάτι ανάλογο με τις λίστες του max αλλά με πολύ πιο βολικό για το video τρόπο. Έχει επίσης μεγάλη προσαρμοστικότητα σε μεγάλη γκάμα εξωτερικών συσκευών και πρωτοκόλλων όπως είναι τα midi, dmx, osc, http κι άλλα. Το μεγάλο του πλεονέκτημα σε σχέση με άλλα προγράμματα είναι ότι χρησιμοποιεί την κάρτα γραφικών (gpu) για τη δημιουργία και επεξεργασία εικόνας, πετυχαίνοντας κατά μεγάλη διαφορά μεγαλύτερες αναλύσεις και fps.

#### **EyesWeb**

Το EyesWeb είναι μία ανοιχτή πλατφόρμα λογισμικού για το περιβάλλον των Windows. Δημιουργήθηκε αρχικά για να υποστηρίξει την έρευνα γύρω από πολυμεσικά διαδραστικά συστήματα αλλά χρησιμοποιείται τώρα ευρέως για διαδραστικές εφαρμογές με χορό ή/και μουσική. Επειδή δημιουργήθηκε για αυτό το σκοπό, διαθέτει γραφικό περιβάλλον προγραμματισμού και στα εργαλεία του περιέχονται αντικείμενα για απευθείας αντιστοίχιση του ήχου σε video ή το ανάποδο.

#### **ISADORA**

Το isadora της εταιρείας troika tronix είναι ένα γραφικό περιβάλλον προγραμματισμού που παρέχει διαδραστικό έλεγχο σε ψηφιακά μέσα (media) με ιδιαίτερη έμφαση στην επεξεργασία του ψηφιακού video σε πραγματικό χρόνο. Έχει την ίδια λογική με αυτή του jitter όσον αφορά στη χρήση αντικειμένων ως δομικών στοιχείων στη δημιουργία αλγορίθμων.

#### **VNS**

Το very nervous system αναπτύχθηκε από τον David Rokeby σαν μέσο για τη δημιουργία interactive sound installations (διαδραστικές ηχητικές εγκαταστάσεις) μόνο για το macosx. Χρησιμοποιεί αισθητήρες, κάμερες, επεξεργαστές εικόνας, υπολογιστές και synthesizers ώστε να δημιουργήσει ένα χώρο μέσα στον οποίο οι κινήσεις κάποιου δημιουργούν ήχο ή και μουσική. Δουλεύει και ως plug-in στο jitter.

## Επίλογος

Η ανάπτυξη της τεχνολογίας στον τομέα της επεξεργαστικής ισχύως είναι αυτή που έχει καταστήσει τη χρήση της κάμερας, ως διαδραστικό μέσο, προσιτή στον μέσο χρήστη και αποτελεί εως ένα σημείο, μιά εναλλακτική των διαφόρων ειδών αισθητήρων που υπάρχουν στο εμπόριο. Το μεγαλύτερο μέρος των εφαρμογών που κάνουν χρήση αισθητήρων απαιτούν, εκτός από πολλά χρήματα, ίσως μια καλή γνώση των ηλεκτρονικών για μετατροπή της αναλογικής πληροφορίας σε ψηφιακά πρωτόκολλα που να μπορεί να χρησιμοποιήσει ένας new media καλλιτέχνης. Με τη χρήση λογισμικών η πληροφορία που προέρχεται από την κάμερα είναι άμεσα μετατρέψιμη σε πολλούς τύπους δεδομένων.

Στην εργασία αυτή, αναλύθηκαν κάποιοι τρόποι δημιουργίας γραφικών. Από την έρευνα στα πλαίσια της εργασίας έχει βγει το συμπέρασμα ότι η δημιουργία γραφικών στα πλαίσια του πίνακα jitter είναι περιορισμένη σε δυνατότητες λόγω της επεξεργαστικής ισχύως που απαιτεί. Κάποιος που θέλει να εστιάσει σε αυτόν τον τομέα θα πρέπει να εμβαθύνει σε τρόπους να χρησιμοποιεί την κάρτα γραφικών (gpu-graphics processor unit) έναντι της cpu για τη δημιουργία γραφικών. Μέσω του jitter μπορούν να γίνουν τέτοιες επεξεργασίες με τη βιβλιοθήκη αντικειμένων jitter OpenGL. Ο προγραμματισμός σε OpenGL αποτελεί ένα μεγάλο κεφάλαιο στο Jitter και η ανάλυση του δεν ανήκει σε αυτή την εργασία.

Όσον αφορά στον τομέα του ήχου, το msp είναι γνωστό για τις δυνατότητες του και χρησιμοποιείται όλο και περισσότερο από μουσικούς.

Η όλη εργασία παρουσίασε ενδεχομένως ένα τρόπο δημιουργίας διαδραστικού περιβάλλοντος προσιτό σε καλλιτέχνες που δεν τους απασχολεί τόσο το τεχνικό κομμάτι όσο η άμεση πραγματοποίηση της ιδέας τους. Ενώ περιγράφει συγκεκριμένο λογισμικό και μεθοδολογία εργασίας, το λογισμικό αυτό όπως και πολλά σύγχρονα λογισμικά, έχει προσαρμοστικότητα ώστε να μπορεί με πολλούς τρόπους να επικοινωνήσει με άλλα λογισμικά ή εξωτερικές συσκευές. Ενθαρρύνει άρα, τη συνεργασία και μπορεί να αποτελέσει κεντρικό κόμβο επικοινωνίας μεταξύ καλλιτεχνών και προγραμματιστών.

## Βιβλιογραφία

- Baecker, R. M., *Readings in Human-Computer Interaction: Toward*, 1997  
<http://books.google.com/books?hl=en&lr=&id=gjm6FpMUTXgC&oi=fnd&pg=RA2-PR11&dq=hci&ots=RgbytcvQmU&sig=hSl4uicaDQZCc3fxKp3wWUNz5AE#PRA2-PR11,M1>
- Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 1986, 8:679-714.
- Cornock, S., Edmonds, E. A.. *The creative process where the artist is amplified or superseded by the computer*. Leonardo, 1973, 6, 11-16  
<http://links.jstor.org/sici?sici=0024-094X%28197324%296%3A1%3C11%3ATCPWTA%3E2.0.CO%3B2-%23&access=1&origin=JSTOR-accessOptions>
- Edmond, E., *Explorations in Art and Technology*. London:Springer, 2002
- Gintz, C., "Happening". Στο Μ. Ferrer και Μ.-Η. Colas-Adler (Επιμ.), *Ομάδες, Κινήματα, Τάσεις της Σύγχρονης Τέχνης μετά το 1945* (σελ. 139-141). Αθήνα, Εξάνιας, 1999
- Levine, S., Ehrlich, S., *The Freestyle System: A design perspective*. In Klinger, A. (Ed.), *Human machine Interactive systems*, Plenum, 1991, 3-21.
- Manovich, L., *The Death of Computer Art*, 1996  
<http://www-apparitions.ucsd.edu/~manovich.text/death.html>
- Manovich, L., *The Language of New Media*, Cambridge Mass, USA, MIT Press, 2001
- Martin, S., *Video art*. Bonn: Taschen, 2006
- Muller, L., Edmonds, E. *Living Laboratories: Making and Curating Interactive Art*, Creativity and Cognition Studios, University of Technology, Sydney  
<http://www.siggraph.org/artdesign/gallery/S06/paper2.pdf>
- Ritter, D., *E's for Artists: a handbook for New Media Artists*. New York City, 2001
- Russ, J. C., *The Image Processing Handbook*, 2nd ed. Florida: CRC Press, 1995
- Rush, M., *New Media in Late 20th-Century Art*. London: Thames & Hudson, 1999
- Sebel, N., *The State-of-the-Art in Human-Computer Interaction*. Faculty of Science, University of Amsterdam, The Netherlands, 2004  
[http://staff.science.uva.nl/~nicu/publications/hci04\\_edit.pdf](http://staff.science.uva.nl/~nicu/publications/hci04_edit.pdf)
- Shannon, C. E., *A mathematical theory of communication*. Bell System Technical Journal. July and October, 1948

Stephenson, G., *An Introduction to Matrices, Sets, and Groups for  
Science Students*. Toronto: Dover, 1986  
*The Jitter tutorials*

## Ευχαριστίες

Η ολοκλήρωση της παρούσας εργασίας δε θα μπορούσε να πραγματοποιηθεί χωρίς τη συμπαράσταση και καθοδήγηση μίας σειράς ανθρώπων, στους οποίους θα ήθελα να απευθύνω ολόψυχα τις ευχαριστίες μου. Ειδικότερα, θα ήθελα να ευχαριστήσω: Τα μέλη της επιτροπής για την εξέταση της εργασίας μου, και συγκεκριμένα τον επιβλέποντα καθηγητή κ. Ταξιάρχη Διαμαντόπουλο , για την εμπιστοσύνη που μου έδειξε και την υπομονή του καθώς και για την καθοδήγηση του στο σύνολο της εργασίας και τη κα. Τζεδάκη Κατερίνα για την καθοδήγηση της στα τελευταία στάδια της εργασίας.

Τη Χρήστου Ευαγγελία που το έργο της παρουσιάζεται εδώ για την πολύ επικοδομητική συνεργασία και τη πολύτιμη βοήθεια της σε όλα τα επίπεδα της εργασίας μου.

Το Μόσχο Κωσταντίνο για τις γόνιμες συζητήσεις και τις ιδέες του αλλά και για την βοήθεια με τα απαραίτητα hardware.