



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Κοτσιράκης Μιχαήλ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ

**ΣΥΝΘΕΣΗ ΚΑΙ ΜΕΤΑΜΟΡΦΩΣΗ ΗΧΟΥ
ΜΕ ΧΡΗΣΗ ΗΜΙΤΟΝΟΕΙΔΟΥΣ ΜΟΝΤΕΛΟΥ**

εξεταστική επιτροπή:

Κουτσάκης Πολυχρόνης (επιβλέπων)

Ποταμιάνος Αλέξανδρος

Πετράκης Ευριπίδης

Αφιερώνεται στην οικογένειά μου
και ιδιαίτερα στην αγαπημένη μου αδερφή Αθηνά.

ΠΕΡΙΛΗΨΗ

Σκοπός αυτής της διπλωματικής εργασίας είναι η ανάπτυξη ενός αλγορίθμου μεταμόρφωσης ήχου και πιο συγκεκριμένα μουσικού σήματος. Ο αλγόριθμος που ονομάζεται Athena αναλύει δύο ήχους A και B χρησιμοποιώντας ένα ημιτονοειδές μοντέλο και μέσα σε ένα καθορισμένο από το χρήστη χρονικό διάστημα εξομαλύνει τις διαφωνίες ανάμεσα στους δύο ήχους δημιουργώντας την εντύπωση ότι ο ήχος A μεταμορφώνεται στον B. Έχει αναπτυχθεί για να μεταμορφώσει ήχους που είναι δυσαρμονικοί μεταξύ τους ενώ για να οριστεί αυτή η δυσαρμονία χρησιμοποιείται το μοντέλο ERB που προσεγγίζει τη λειτουργία του ανθρώπινου αυτιού με υπερτιθέμενα ακουστικά φίλτρα περιορισμένου εύρους ζώνης. Οι ήχοι μοντελοποιούνται σύμφωνα με το ημιτονοειδές μοντέλο MQ πάνω στις παραμέτρους του οποίου παρεμβάλλει ο αλγόριθμος Athena. Εξάγονται τα διαφορετικά κανάλια που αποτελούν τους υπό επεξεργασία ήχους και η αρμονική ανακατασκευή αφαιρείται από το αρχικό σήμα ώστε να εξαχθεί ο θόρυβος. Τα κανάλια που συνηχώντας θα προκαλέσουν δυσαρμονία ταιριάζονται μεταξύ τους και δημιουργούν νέα κανάλια με ενδιάμεσες τιμές. Τα υπόλοιπα κανάλια εισάγονται ή σιωπούν σταδιακά ώστε να επιτευχθεί η μετάβαση. Στη συνέχεια ο επεξεργασμένος θόρυβος από τους δύο ήχους προστίθεται στο συνθετικό αποτέλεσμα και αποτελεί την έξοδο του εφέ.

Τα αποτελέσματα του αλγορίθμου δειγματίστηκαν σε 11 χρήστες οι οποίοι σε ποσοστό 80% τα προτίμησαν σε σχέση με το απλό cross fade για δυσαρμονικούς ήχους. Ο αλγόριθμος παρουσιάζει ενδιαφέροντα αποτελέσματα και για αρμονικούς ήχους (προτίμηση χρηστών 60%) αλλά και για όχι τυπικά μουσικά σήματα όπως η ανθρώπινη φωνή (προτίμηση χρηστών 61%). Σαν παράπλευρο κέρδος της χρήσης του ημιτονοειδούς μοντέλου και των μεθόδων που αναπτύσσονται για τον αλγόριθμο Athena δημιουργείται μία μικρή βιβλιοθήκη με μερικά ακόμα εφέ ήχου: Το εφέ Nobeating που επιτρέπει σε δύο ήχους να συνηχήσουν χωρίς να προκαλούν δυσαρμονία, το εφέ SinusoidalOctaver που προσθέτει μία νότα σε διάστημα οκτάβας στην είσοδο του αλγορίθμου, το εφέ Pitchshifter που υλοποιεί ένα τυπικό εφέ τονικής μετατόπισης και τέλος το εφέ FMvibrato που δημιουργεί βιμπράτο με τη χρήση διαμόρφωσης συχνότητας. Όλα τα εφέ έχουν υλοποιηθεί σε περιβάλλον Matlab.

ABSTRACT

The goal of this dissertation thesis is to develop an algorithm for morphing sound and more specifically musical signal. The algorithm is named Athena and it analyzes two sounds, namely A and B, using a sinusoidal model in a user defined period of time. It smooths out the dissonance between the two sounds creating the impression that sound A morphs into sound B. It has been designed for a problem set of dysharmonic sounds. Dysharmonicity is defined using the Equivalent Rectangular Bandwidth approach which models the human ear as a sum of superimposed acoustic filters. The sounds are modeled using the MQ sinusoidal model on whose parameters Athena operates to create the desired effects.

The tracks of which the two sounds consist are extracted and the harmonic partial is subtracted from the original signal, thus yielding the noise. The tracks that create beating are matched to create new tracks of intermediate values. The rest of the channels fade in or out to create the effect of transition. The result of this process is added to the processed noise of the two samples creating the final outcome.

The results of the algorithm have been tested with 11 users of which 80% preferred them to a simple cross fade (for dysharmonic sounds). 60% preferred Athena for harmonic sounds and 61% for non-typical musical signals such as human speech. Using the power of the sinusoidal model and the methods developed for Athena a small library with some more sound effects is created: the Nobeating effect allows two dysharmonic sounds to sound simultaneously without beating, the SinusoidalOctaver adds an octave note to the algorithm input, the PitchShifter effect shifts the note to another pitch while the FMvibrato creates a vibrato using Frequency Modulation. All effects have been developed in the MATLAB environment

Ευχαριστίες

Ευχαριστώ τον κο. Ποταμιάνο για τη μακρά καθοδήγησή του σε μία μακρά εργασία και για τις φορές που με βοήθησε να σκεφτώ απλά. Τον Dan Ellis που μου προσέφερε μία πολύ συνεκτική βιβλιοθήκη μοντελοποίησης ήχου. Τον Ηλία Ιωσήφ για τις μικρές του συμβουλές πάνω στο πώς πρέπει να δουλεύεται μία διπλωματική εργασία. Τον Πάνο Πετρόπουλο για τις ιδέες του και για τις νότες διαύγειας που μου προσέφερε. Τον Γιάννη Κουτσούκο που υπέμεινε για χρόνια γυμνά ημίτονα στις μικρές μεταμεσονύχτιες ώρες. Τέλος τα αδέρφια Ζαφειρίου και τον Σταύρο Μαρκόνη που μου έδωσαν το ερέθισμα για μία τέτοια εργασία.

ΠΕΡΙΕΧΟΜΕΝΑ

1.	Εισαγωγή	
1.1	Τα κίνητρα της εργασίας.....	11
1.2	Ορισμός του προβλήματος.....	12
1.3	Προτεινόμενη λύση.....	12
2.	Ήχος, άνθρωπος και δυσαρμονία.....	14
2.1	Ο μηχανισμός πρόσληψης του ήχου από το ανθρώπινο αυτί.....	14
2.2	Η μαθηματική αναπαράσταση του ήχου και ο Μετασχηματισμός Fourier.....	17
2.3	Δυσαρμονικοί ήχοι (beats) και κακοφωνία	23
3.	Ανάλυση, Μοντελοποίηση και Σύνθεση ήχου.....	25
3.1	Η ιστορία της σύνθεσης ήχου.....	25
3.2	Τα είδη της σύνθεσης ήχου.....	26
3.3	Το ημιτονοειδές μοντέλο.....	27
4.	Η μεταμόρφωση ήχου.....	37
4.1	Τι είναι και τι δεν είναι η μεταμόρφωση ήχου.....	37
4.2	Η δομή του αλγορίθμου μεταμόρφωσης ήχου Athena.....	38
4.3	Οι λεπτομέρειες της υλοποίησης.....	39
4.3.1	Ο STFT και η συγχώνευση καναλιών.....	40
4.3.2	Το ταίριασμα των δυσαρμονικών καναλιών.....	42
4.3.3	Η σύνθεση της μεταμόρφωσης για τα δυσαρμονικά κανάλια.....	42
4.3.4	Το πρόβλημα της φάσης.....	43
4.3.5	Η σύνθεση της μεταμόρφωσης για τα κανάλια που δεν προκαλούν δυσαρμονία.....	44
4.3.6	Η σύνθεση των τριών μερών του νέου ήχου.....	44
4.3.7	Η διαχείριση του θορύβου.....	46
5.	Αξιολόγηση του αλγορίθμου Athena.....	50
5.1	Ο σκοπός της αξιολόγησης.....	50

5.2	Το τεστ της αξιολόγησης του αλγορίθμου.....	50
5.3	Τα αποτελέσματα της αξιολόγησης.....	52
5.3.1	Η πιο ομαλή μετάβαση.....	52
5.3.2	Αισθητική ικανοποίηση και ποιότητα του ήχου.....	53
6.	Η Βιβλιοθήκη εφέ του Athena.....	55
6.1	Ο αλγόριθμος Nobeating.....	55
6.2	Ο αλγόριθμος PitchShifter.....	57
6.3	Ο αλγόριθμος SinusoidalOctaver.....	58
6.4	Ο αλγόριθμος Fmvibrato	59
7.	Συμπεράσματα.....	63
7.1	Μελλοντική δουλειά.....	64
Παράρτημα: Οι κώδικες Matlab		65
Βιβλιογραφία		87

1. ΕΙΣΑΓΩΓΗ

1.1 Τα κίνητρα της εργασίας

Η ανάπτυξη των ηλεκτρονικών υπολογιστών, της ηλεκτρονικής και της ψηφιακής επεξεργασίας σήματος έχει διευρύνει αποφασιστικά την τέχνη της μουσικής ώστε οι όροι ηλεκτρονική μουσική, ηλεκτρονικά όργανα και μουσική από υπολογιστή να αποτελούν πλέον καθημερινότητα για τους δημιουργούς και το κοινό της. Θα λέγαμε μάλιστα πως η θέση του ηλεκτρονικού υπολογιστή είναι οργανική για πολύ ευρείες πρακτικές της τέχνης αυτής, από τα όργανα μέχρι την ηχογράφηση και από τη σύνθεση μέχρι τη διάδοση και την αναπαραγωγή.

Ένας τομέας που δεν έχει μείνει ανεπηρέαστος από τις εξελίξεις αυτές είναι ο χώρος των ψηφιακών μουσικών οργάνων όπου η διάδοση ισχυρών υπολογιστών στο ευρύ κοινό έχει ανοίξει τη δυνατότητα σε εκατομμύρια μουσικούς να χρησιμοποιούν με πολύ χαμηλό κόστος την τελευταία λέξη της τεχνολογίας. Χιλιάδες εκδοχές ψηφιακών οργάνων που αντικαθιστούν τα παραδοσιακά ή που ανοίγουν νέους ορίζοντες σε ηχοχρώματα και εφέ κάνουν την εμφάνισή τους και εμπλουτίζουν την ακουστική παλέτα κάθε μουσικού είδους.

Μία σημαντική κατηγορία τέτοιων ψηφιακών οργάνων είναι τα όργανα που χρησιμοποιούν ηχογραφημένα δείγματα (samples) φυσικών οργάνων τα οποία δίνονται στο χρήστη σαν βιβλιοθήκες ήχων έτοιμες να ζωντανέψουν από το χτύπημα ενός πλήκτρου ή από το ρυθμικό χτύπημα ενός drum pad. Η πιο ενδιαφέρουσα πλευρά αυτών των οργάνων είναι η δυνατότητά τους να χρησιμοποιούν τους φυσικούς ήχους όχι σαν νεκρά αντίγραφα του φυσικού κόσμου αλλά σαν ζωντανές οντότητες έτοιμες να μετατραπούν στο πιο μακρινό και αλλοπρόσαλλο ηχητικό ερέθισμα. Έχει ήδη διαφανεί (και προσεγγιστεί από διαφορετικές οδούς) η δυνατότητα για ένα συνεχές ηχοχρωμάτων όπου οι φυσικοί ήχοι είναι μόνο στιγμές ενός άπειρου φάσματος που διανύει όλες τις ενδιαμέσες καταστάσεις.

Τα θεμέλια αυτών των δυνατοτήτων βρίσκονται στην ψηφιακή επεξεργασία σήματος και η εργασία αυτή επιχειρεί να διεισδύσει στις πιο βασικές αρχές και διαδικασίες που απαιτούνται για τη μεταμόρφωση των ήχων. Η μεταμόρφωση του ήχου είναι η διαδικασία που κατά την πραγματοποίησή της φανερώνει τις ενδιάμεσες καταστάσεις ανάμεσα στα ηχοχρώματα δύο ήχων

1.2 Ορισμός του προβλήματος

Δεδομένων δύο ήχων A και B μεταμόρφωση ήχου (sound morphing) ονομάζουμε την σταδιακή αλλαγή κατά τη διάρκεια του χρόνου του ηχοχρώματος του ήχου A με τρόπο τέτοιο ώστε να φτάσει στο ηχοχρώμα του B [1].

Το πρόβλημα αυτό έχει αντιμετωπιστεί με διαφορετικούς τρόπους στο παρελθόν όμως δεν έχει μελετηθεί επαρκώς η διάσταση της δυσαρμονίας κατά τη διάρκεια της μεταμόρφωσης. Ο πιο στενός ορισμός αυτού του προβλήματος αφορά τις περιπτώσεις που οι δύο ήχοι προς μεταμόρφωση είναι μουσικοί και είναι δυσαρμονικοί μεταξύ τους. Επεκτείνεται όμως σε κάθε πιθανό ήχο αφού κατά κανόνα θα αποτελούνται από αρμονικά στοιχεία που είναι ασύμβατα μεταξύ τους. Σκοπός μας είναι να δημιουργήσουμε έναν πρωτότυπο αλγόριθμο μεταμόρφωσης ήχου δίνοντας όμως έμφαση στα χαρακτηριστικά του ήχου που προκαλούν δυσφορία στον ακροατή.

1.3 Προτεινόμενη λύση

Για να λύσουμε αυτή την υποκατηγορία των προβλημάτων της μεταμόρφωσης προτείνουμε έναν πρωτότυπο αλγόριθμο που ονομάζουμε Athena.

Δημιουργούμε ένα σετ ήχων που βρίσκονται σε τέτοια διαστήματα μεταξύ τους που από την κλασική μουσική αρμονία γνωρίζουμε ότι είναι δυσαρμονικοί. Επιλέγουμε το ημιτονοειδές μοντέλο για να αναλύσουμε τους ήχους μας και να εξάγουμε τα χαρακτηριστικά των ήχων τα οποία θα

επεξεργαστούμε για να δημιουργήσουμε το απαραίτητο εφέ. Συγκεκριμένα επιλέγουμε το μοντέλο MQ και μία βιβλιοθήκη υλοποίησής του στο περιβάλλον Matlab όπως παρουσιάζουμε στο κεφάλαιο 3. Ορίζουμε τα όρια των συχνοτικών φίλτρων με τα οποία προσεγγίζεται το ανθρώπινο αυτί. Θεωρούμε πως οι αρμονικές των ήχων που συνηχούν μέσα σε αυτά τα όρια που ορίζονται από το μοντέλο ERB προκαλούν δυσαρμονία όπως παρουσιάζεται στο κεφάλαιο 2.

Χωρίζουμε τις αρμονικές των ήχων σε δύο κατηγορίες. Αυτές που θα προκαλέσουν δυσαρμονία περνούν από ειδική επεξεργασία όπου βρίσκονται ενδιάμεσες τιμές συχνότητας, πλάτους και φάσης. Οι υπόλοιπες σβήνουν και εισάγονται ομαλά στο χρόνο. Στο αρμονικό μέρος του σήματος προστίθεται και ο ελαφρά επεξεργασμένος θόρυβος των δύο σημάτων και το τελικό σήμα προσφέρεται για ακρόαση στο χρήστη. Μία πλήρης περιγραφή του αλγορίθμου παρουσιάζεται στο κεφάλαιο 4.

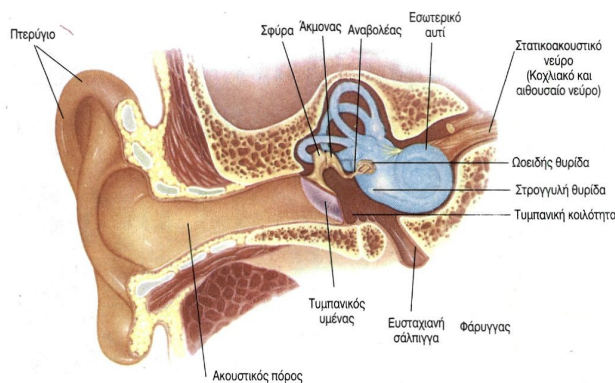
Στα πλαίσια της εργασίας παρουσιάζουμε την ανταπόκριση των ακροατών στον πρωτότυπο αλγόριθμό μας αλλά προτείνουμε και μερικούς ακόμα αλγόριθμους που προέκυψαν από τις γνώσεις που αποκομίσαμε στο χώρο της μοντελοποίησης ήχου και της δυσαρμονίας. Μάλιστα ο αλγόριθμος NoBeating προσφέρει τη δυνατότητα στον χρήστη να εξομαλύνει ακριβώς τις αρμονικές που προκαλούν δυσαρμονία και μεταμορφώνονται στον αλγόριθμο Athena. Ο NoBeating και η υπόλοιπη βιβλιοθήκη εφέ παρουσιάζονται στο κεφάλαιο 6.

2. ΗΧΟΣ, ΑΝΘΡΩΠΟΣ ΚΑΙ ΔΥΣΑΡΜΟΝΙΑ

Στο κεφάλαιο αυτό εξετάζουμε τα βασικά χαρακτηριστικά του ήχου και τον τρόπο με τον οποίο αυτός γίνεται αντιληπτός από το ανθρώπινο αυτί. Εισάγουμε τα μαθηματικά εργαλεία που θα χρησιμοποιηθούν στην εργασία με ιδιαίτερη έμφαση στο διακριτό μετασχηματισμό Fourier. Τέλος μελετάμε με μεγαλύτερη ακρίβεια τις συνθήκες κάτω από τις οποίες το ανθρώπινο αυτί ενοχλείται από τη δυσαρμονία.

2.2 Ο μηχανισμός πρόσληψης του ήχου από το ανθρώπινο αυτί

Ο ήχος αποτελείται από δονήσεις στο αέριο που ονομάζουμε αέρα. Όταν κάποιο σώμα δονείται, οι αυξομειώσεις της πίεσης που δημιουργεί στον αέρα μεταδίδονται μέσω αυτού

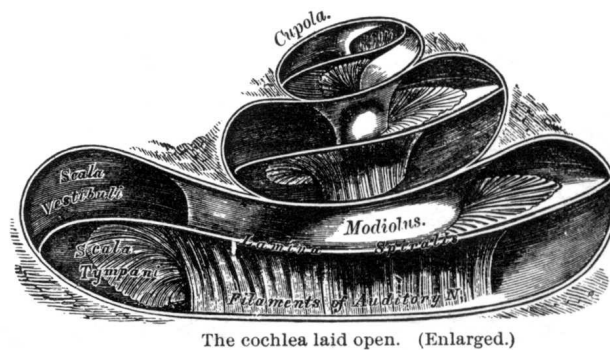


σχήμα 2.1: Τα βασικά συστατικά του ανθρώπινου αφτιού

περίπου όπως τα κύματα που δημιουργεί μία πέτρα στην επιφάνεια μίας λίμνης. Τα κύματα του ήχου ονομάζονται διαμήκη γιατί η διεύθυνση διάδοσής τους είναι παράλληλη στη διεύθυνση ταλάντωσης των σωματιδίων του μέσου που διεγείρουν, δηλαδή του αέρα. Τα χαρακτηριστικά αυτών των κυμάτων είναι και τα χαρακτηριστικά αυτού που ονομάζουμε ήχος. Μπορούμε μάλιστα να κάνουμε την εξής χαλαρή συσχέτιση: τα πλάτη τους σχετίζονται με την ένταση, η

συχνότητά τους με την τονικότητα, οι μικροδιαφορές στη συχνότητά τους με το ηχόχρωμα ενώ η διάρκειά τους στο χρόνο με τη διάρκεια του ήχου [2].

Για να αντιληφθεί ο άνθρωπος αυτά τα κύματα του ήχου έχει ανάγκη από ένα διαμεσολαβητή ανάμεσα στον αέρα και τον εγκέφαλό του. Τον ρόλο αυτό αναλαμβάνει το ανθρώπινο αυτί. Το αυτί χωρίζεται σε τρία μέρη, το εξωτερικό αυτί, το μέσο αυτί και το εσωτερικό αυτί. Το εξωτερικό αυτί συλλέγει το ηχητικό σήμα το οποίο φτάνει στο τύμπανο μέσω του ακουστικού πόρου. Ο τυμπανικός υμένας δονείται σύμφωνα με τη συχνότητα του ήχου, δόνηση που ενισχύεται και μεταφέρεται στο εσωτερικό αυτί μέσω ενός λεπτεπίλεπτου συστήματος μοχλών που αποτελείται από τα πολύ μικρά κόκκαλα σφύρα, άκμονα και αναβολέα. Ο δέκτης της κίνησης αυτής ονομάζεται ωοειδής θυρίδα και σαν αποτέλεσμα του πολύ μικρού του μεγέθους σε σχέση με το τύμπανο (και των μοχλών) δέχεται τις



Σχήμα 2.2: ο κοχλίας

δονήσεις με περίπου 18 φορές μεγαλύτερη ένταση.

Αυτή είναι και η πόρτα του εσωτερικού αυτιού μέσα στο οποίο βρίσκεται ο κοχλίας που περιέχει ένα πολύπλοκο σύστημα καναλιών, τον υμενώδη και τον οστέινο λαβύρινθο. Αν μπορούσαμε να ξετυλίξουμε τον κοχλία θα είχε μήκος περίπου 30mm. Θα βλέπαμε ότι το τούνελ στο εσωτερικό του χωρίζεται στα δύο από την βασική μεμβράνη, η οποία όμως αφήνει μία οπή στην άπω πλευρά του κοχλία ώστε τα δύο μέρη να επικοινωνούν. Το σύστημα περιέχει υγρό, την ενδόλεμφο. Η κυκλοφορία αυτού του υγρού ανάμεσα στα δύο μέρη του κοχλία διεγείρει τη βασική μεμβράνη μέχρι ενός σημείου. Ένα σύστημα από τριχίδια συνδεδεμένα με εγκεφαλικά νεύρα

εντοπίζουν το σημείο αυτό της διέγερσης κάτι που μας επιτρέπει να ξεχωρίζουμε τις διαφορετικές συχνότητες της δόνησης. Ένα σύστημα ανάδρασης του εγκεφάλου ενισχύει τις διεγέρσεις αυτές για να διακρίνει λεπτομέρεια ή τις συμπιέζει για να εξομαλύνει πολύ μεγάλες εντάσεις. [3]

Πρέπει όμως να επισημάνουμε πως η διακριτική ικανότητα του αυτιού δεν είναι άπειρη. Ένας μέσος άνθρωπος μπορεί να ακούσει συχνότητες από τα 20Hz μέχρι τα 20000Hz (αν και υπάρχουν ενδείξεις για “αίσθηση” ήχου και έξω από αυτά τα όρια) διότι μικρότερες ή μεγαλύτερες συχνότητες δεν διεγείρουν τη βασική μεμβράνη. Το γνωστό σε όλους σύστημα αναφοράς των decibels ήχου αφορά άλλον έναν περιορισμό του ανθρώπινου αυτιού. Τα 0dB εκπροσωπούν την ένταση των $10^{-12} \text{ watts/m}^2$ που είναι η πιο μικρή ένταση ήχου που μπορεί να γίνει διακριτή από το ανθρώπινο αυτί.

Ένας σημαντικός περιορισμός όμως που θα μας απασχολήσει και στη συνέχεια είναι η αδυναμία διάκρισης πολύ μικρών διαφορών στη συχνότητα και την ένταση. Ο παρακάτω πίνακας δίνει τις μέγιστες τιμές διαφοράς τις οποίες μπορεί να διακρίνει το αυτί.

Συχνότητα Hz	Ένταση dB										
	5	10	15	20	30	40	50	60	70	80	90
31	220	150	120	97	76	70					
62	120	120	94	85	80	74	61	60			
125	100	73	57	52	46	43	48	47			
250	61	37	27	22	19	18	17	17	17	17	
550	28	19	14	12	10	9	7	6	1		
1000	16	11	8	7	6	6	6	6	5	5	4
2000	14	6	5	4	3	3	3	3	3	3	
4000	10	8	7	5	5	4	4	4	4		
8000	11	9	8	7	6	5	4	4			
11700	12	10	7	6	6	6	5				

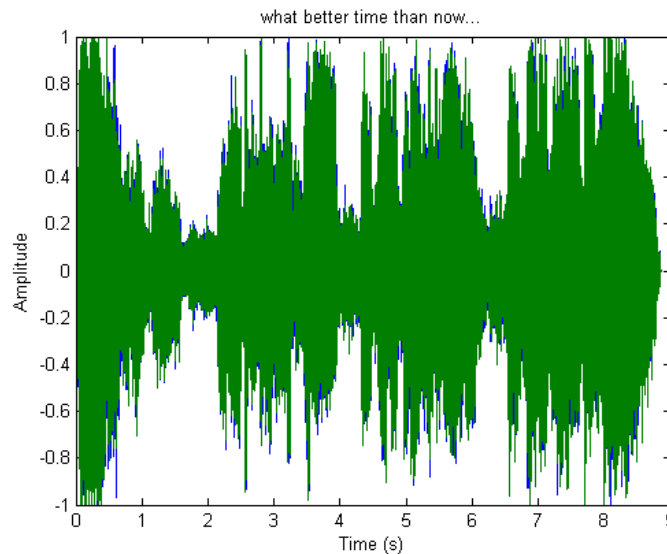
σχήμα 2.3

Όπως είναι φανερό είμαστε πιο ευαίσθητοι στις μικροαλλαγές έντασης και συχνότητας που αφορούν μεγαλύτερες συχνότητες. Ας σημειώσουμε προς το παρόν πως

οι παραπάνω τιμές για αυτή την “αδυναμία” του αυτιού αφορούν ήχους που ακούγονται ασύγχρονα και δεν συνηθούν.

2.3 Η μαθηματική αναπαράσταση του ήχου και ο μετασχηματισμός Fourier

Περνώντας σε έναν πιο τυπικό ορισμό του ήχου μας είναι χρήσιμο να τον περιγράψουμε σαν ένα ηχητικό σήμα, σαν μία συνάρτηση εξαρτημένη από την ελεύθερη μεταβλητή του χρόνου της οποίας κάθε τιμή αντιστοιχεί σε κάποια τιμή έντασης [4]. Μάλιστα μπορούμε να την περιορίσουμε στο χρόνο θέτοντας τα όρια της ίσα με την στιγμή που ο ήχος αρχίζει και παύει να γίνεται αντιληπτός από το ανθρώπινο αυτί. Το παράδειγμα ενός τέτοιου σήματος $x(t)$ μπορούμε να δούμε στο σχήμα 2.4.



σχήμα 2.4: ένα σήμα φωνής 9 δευτερολέπτων

Από το παραπάνω σήμα φωνής μπορούμε να βγάλουμε σημαντικά συμπεράσματα για τη σχέση έντασης και χρόνου σε ένα δοσμένο σήμα. Ένα έμπειρο μάτι (ή αλγόριθμος) μπορεί να διακρίνει την ύπαρξη διαφορετικών λέξεων καθώς και να εντοπίσει τα κενά. Όπως όμως είδαμε και στην προηγούμενη παράγραφο η σημαντική πληροφορία για το ηχόχρωμα βρίσκεται στη συχνότητα των δονήσεων.

Η θεωρία του Γάλλου φυσικομαθηματικού Jean Baptiste

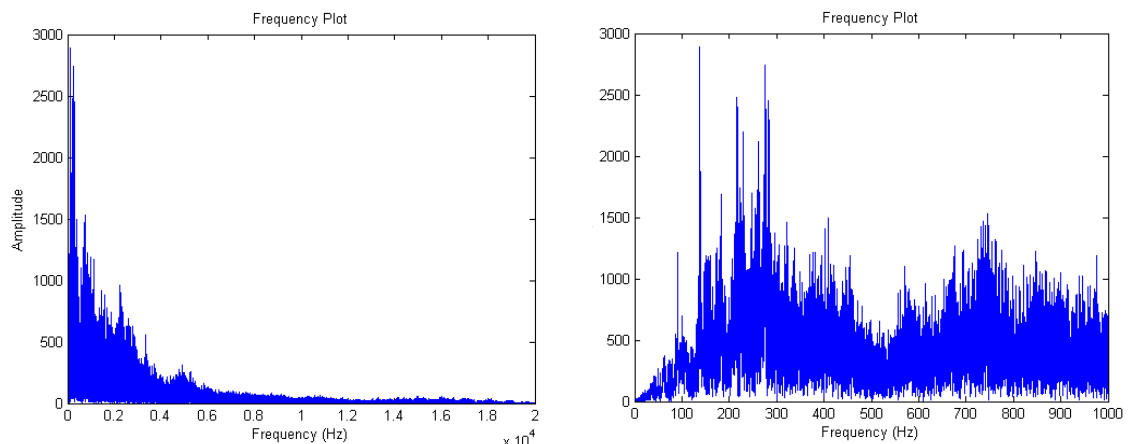
Fourier μας επιτρέπει να αναλύσουμε μία συνάρτηση σε άθροισμα συναρτήσεων απλών συχνοτήτων. Σύμφωνα λοιπόν με το μετασχηματισμό Fourier:

$$X(f) \equiv \int_{-\infty}^{+\infty} x(t) e^{-j\pi f t} dt \quad (2.1)$$

με την προϋπόθεση ότι το σήμα $x(t)$ ικανοποιεί τις 3 συνθήκες Dirichlet:

- $\int_{-\infty}^{\infty} |x(t)| dt < \infty$
- Ο αριθμός των μεγίστων και των ελαχίστων του $x(t)$ σε κάθε πεπερασμένο διάστημα των πραγματικών αριθμών να είναι πεπερασμένος
- Ο αριθμός των ασυνεχειών του $x(t)$ σε κάθε πεπερασμένο διάστημα των πραγματικών αριθμών να είναι πεπερασμένος.

Το μέτρο της $|X(f)|$ εκφράζει το πλάτος με το οποίο συνεισφέρει η κάθε συχνότητα. Εφαρμόζοντας τον μετασχηματισμό Fourier στο παραπάνω σήμα παίρνουμε μία περιγραφή του σήματος στο πεδίο των συχνοτήτων.



σχήμα 2.5: τα πλάτη για $f > 0$

Παρατηρούμε πως τα μεγαλύτερα πλάτη παρατηρούνται στην περιοχή των 150-400Hz, εκεί που εντοπίζονται οι πιο σημαντικές συχνότητες της ανθρώπινης ομιλίας. Γίνεται φανερό πως υπάρχει πληροφορία σε όλες τις συχνότητες. Μάλιστα αν εξετάσουμε και τις τιμές πάνω από τα 20000Hz θα βρούμε μηδαμινή μεν, υπαρκτή δε προσφορά στο σήμα μας. Από τον

αντίστροφο μετασχηματισμό

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad (2.2)$$

διαφαίνεται πως από την πληροφορία των συχνοτήτων μπορούμε να ανακτήσουμε το αρχικό σήμα $x(t)$. Ο

μετασχηματισμός Fourier είναι μέλος μίας μεγαλύτερης ομάδας μαθηματικών εργαλείων που ονομάζονται ανάλυση Fourier κάποια από τα οποία φωτίζουν βασικές ιδιότητες του ήχου και κάποια άλλα που χρειάστηκαν στην ανάπτυξη των βασικών αλγορίθμων αυτής της εργασίας. Ιδιαίτερο ενδιαφέρον έχει η σειρά Fourier σύμφωνα με την οποία κάθε **περιοδικό** σήμα μπορεί να εκφραστεί με το εξής άθροισμα [5]

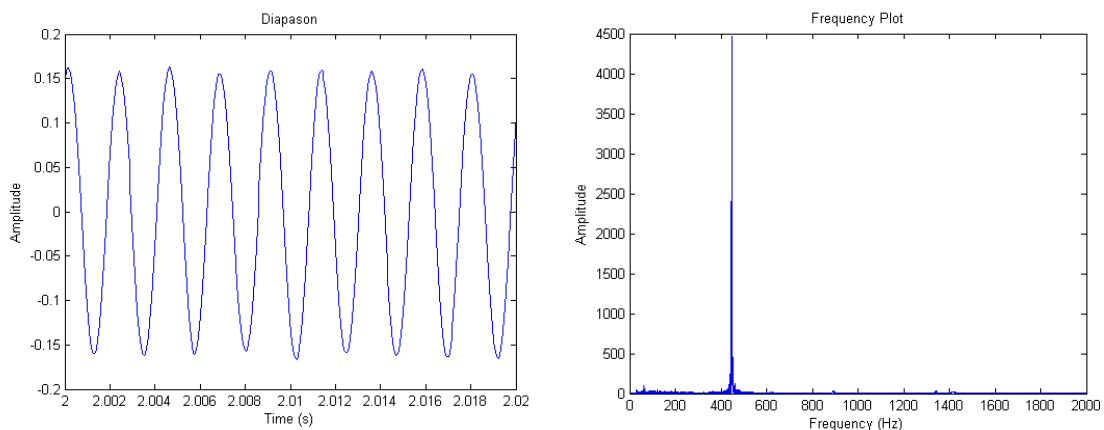
$$x(t) = \sum_{n=-\infty}^{\infty} x_n e^{j2\pi f_0 t} = \frac{a_0}{2} + 2 \sum [a_n \cos(2\pi n f_0 t) + b_n \sin(2\pi n f_0 t)] \quad (2.3)$$

όπου

$$a_n = \frac{2}{T_0} \int_a^{a+T_0} x(t) \cos(2\pi f_0 t) dt$$

$$b_n = \frac{2}{T_0} \int_a^{a+T_0} x(t) \sin(2\pi f_0 t) dt$$

Αυτή η σχέση καθίσταται σημαντική αν λάβουμε υπ' όψιν



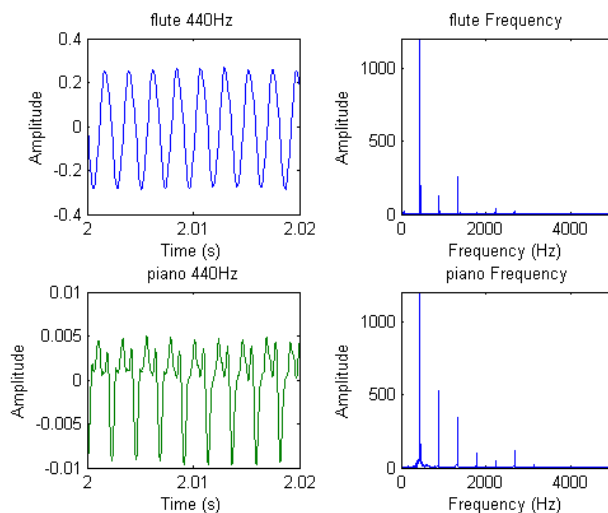
σχήμα 2.6: διαπασών

πως από το άπειρο σύνολο των πιθανών ήχων, μία ιδιαίτερα σημαντική ομάδα για τον άνθρωπο είναι οι ήχοι που

παρουσιάζουν τοπική περιοδικότητα. Οι ήχοι αυτοί έχουν το επιπλέον χαρακτηριστικό του τόνου [6], ο οποίος μάλιστα καθορίζεται από την περίοδο(συχνότητα) της ταλάντωσης. Το απλούστερο μη συνθετικό παράδειγμα τέτοιου ήχου είναι το διαπασών, ένα μουσικό όργανο το οποίο πάλλεται στη συχνότητα των 440Hz και του οποίου το σήμα και ο εξαγόμενος μετασχηματισμός Fourier είναι σχεδόν όμοιος με αυτόν του απλού ημιτόνου $x(t)=\cos(2\pi 440t)$.

Αυτή η συχνότητα των 440Hz μάλιστα έχει καθιερωθεί στη σύγχρονη μουσική ως η νότα Λα(A). Τοπικά περιοδικοί όμως δεν είναι μόνο οι ήχοι μουσικών οργάνων αλλά και η ανθρώπινη ομιλία όπως και πολλά άλλα σήματα. Παρακολουθώντας τη σειρά Fourier διαπιστώνουμε πως οποιοσδήποτε περιοδικός ήχος μπορεί να ανασυντεθεί με τη χρήση απλών βασικών ημιτονοειδών συναρτήσεων. Εξετάζοντας μάλιστα την περιοδική περιοχή σημάτων μουσικών οργάνων βλέπουμε ότι οι σημαντικές διαφορές που παρατηρούνται στο ηχόχρωμά τους οφείλονται (μεταξύ άλλων) στη διαφορετική συνεισφορά των πρώτων πολλαπλασίων της βασικής συχνότητας f_0 στη σχέση (2.3). Η συχνότητα f_0 ονομάζεται θεμέλιος συχνότητα και τα πολλαπλάσια της αρμονικές [7].

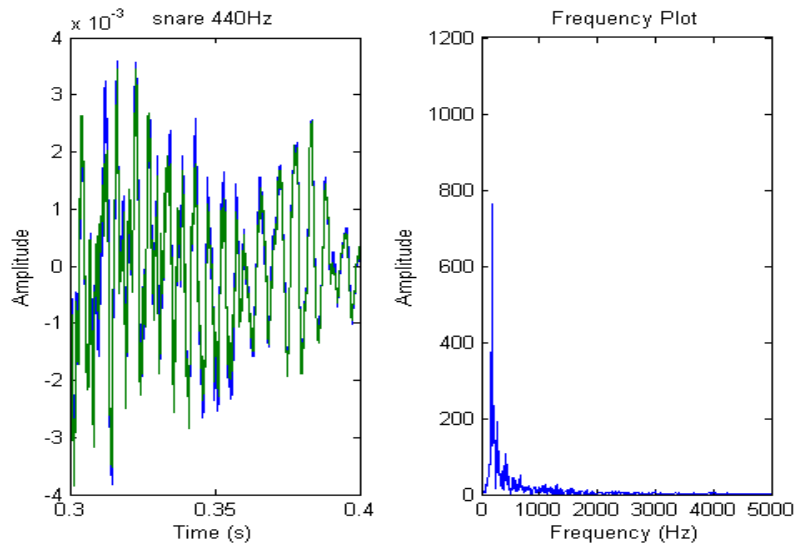
Όπως βέβαια φάνηκε και παραπάνω δεν έχουν όλοι οι ήχοι περιοδικότητα και μάλιστα αυτοί που έχουν είναι πλούσιοι



σχήμα 2.7: Λα από πιάνο και φλάουτο

σε πολλές άλλες συχνότητες. Ήχοι όπως αυτοί που παράγονται

από τα κρουστά (βλ. σχήμα 2.8) χαρακτηρίζονται από μία μεγάλη διασπορά στις συχνότητες όπου όπως και στην ανθρώπινη ομιλία ή τα τονικά όργανα το ηχόχρωμα δίνεται από το σύνολο των άπειρων συχνοτήτων.



σχήμα 2.8: ταμπούρο

Η ψηφιακή επεξεργασία σήματος και η ανάλυση Fourier όμως μας δίνει τη δυνατότητα να εισχωρήσουμε ακόμα βαθύτερα στα χαρακτηριστικά των ήχων και να τους μελετήσουμε συγχρόνως και με λεπτομέρεια στο χρόνο και τη συχνότητα.

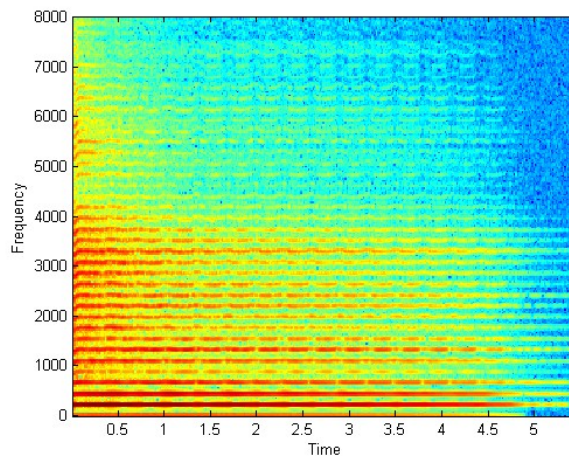
Εν συντομία αναφέρουμε πως σήματα συνεχούς χρόνου(αναλογικά) όπως τα σήματα ήχου είναι δυνατόν να μετατραπούν σε σήματα διακριτού χρόνου (ψηφιακά) ώστε να τα επεξεργαστεί κάποιο ψηφιακό σύστημα και να μετατραπούν ξανά σε αναλογικά ώστε να οδηγήσουν κάποιο ηχείο και να γίνουν ξανά διαθέσιμα προς ακρόαση στο ανθρώπινο αυτί. Η διαδικασία αυτή περιλαμβάνει τη δειγματοληψία, την κβάντιση και την κωδικοποίηση του αρχικού σήματος το οποίο εκφράζεται πλέον από δείγματα (samples). Στην καθημερινότητά μας συναντάμε πολλές μορφές τέτοιας ψηφιοποίησης του ήχου οι οποίες διαφέρουν:

- στη συχνότητα δειγματοληψίας.
- στη λεπτομέρεια της κβάντισης.
- στον τρόπο κωδικοποίησης.

Ενδεικτικά μόνο αναφέρουμε πως η κωδικοποίηση .wav που χρησιμοποιούμε σε αυτή την εργασία συνήθως χρησιμοποιείται με 16bits/sample(24bits/sample σε επαγγελματικές εφαρμογές) και 44,1kHz συχνότητα δειγματοληψίας(συχνά μικρότερη ανάλογα με την εφαρμογή) ενώ δεν συμπιέζει με κανένα τρόπο τα δεδομένα.

Ο Διακριτός Μετασχηματισμός Fourier ή Discrete Fourier Transform(DFT) [8] μας επιτρέπει να επεξεργαστούμε πεπερασμένο αριθμό από δείγματα και να εξάγουμε μία πεπερασμένη ακολουθία μιγαδικών. Οι συχνότητες των εξαγόμενων στοιχείων είναι ακέραια πολλαπλάσια μίας βασικής συχνότητας, της συχνότητας δειγματοληψίας. Η λογική του είναι πως το σετ των δειγμάτων προς επεξεργασία είναι μία περίοδος ενός περιοδικού σήματος

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \quad \text{όπου } N \text{ ο αριθμός των δειγμάτων.}$$



σχήμα 2.9: Λα από σαξόφωνο

Στους μιγαδικούς που μας επιστρέφει ο DFT βρίσκουμε πληροφορία για το πλάτος και τη φάση για ζώνες (bins) συχνοτήτων οι οποίες αντιπροσωπεύουν συχνότητες με εύρος $F_s/2N$. Ο DFT είναι ένας πολύ σημαντικός αλγόριθμος για την Ψηφιακή Επεξεργασία Σήματος και πολλές παραλλαγές του υπό την ονομασία Fast Fourier Transform(FFT) βρίσκονται υλοποιημένες σε χιλιάδες εφαρμογές γύρω μας. Οι αλγόριθμοι

FFT είναι πιο γρήγορες υλοποιήσεις του DFT που όμως επιστρέφουν τα ίδια ακριβώς αποτελέσματα.

Συνδυάζοντας τη δύναμη του FFT με βασικές αρχές παραθύρωσης της επεξεργασίας σήματος μας δίνεται ένα πολύ σημαντικό εργαλείο για την ανάλυση του ήχου που ονομάζεται φασματογράφημα. Το σήμα προς μελέτη χωρίζεται σε επικαλυπτόμενα πακέτα δειγμάτων τα οποία περνάνε μέσα από ένα κατάλληλο παράθυρο (συνήθως Hamming για ήχο) και υφίστανται τον διακριτό μετασχηματισμό Fourier. Έτσι έχουμε τη συχνοτική ανάλυση για κάθε καρέ (frame) ήχου κάτι που μας επιτρέπει να εξετάσουμε τα συχνοτικά χαρακτηριστικά όπως αυτά εξελίσσονται κατά τη διάρκεια του χρόνου.

Παρατηρώντας στο σχήμα 2.9 μία νότα Λα από σαξόφωνο βλέπουμε βασικά σημεία που συζητήσαμε στην αρχή της παραγράφου. Στην αρχή του χρόνου ο ήχος είναι πλούσιος στο σύνολο των συχνοτήτων και έχει για πολύ λίγο έναν κρουστό χαρακτήρα που ονομάζεται ατάκα. Στη συνέχεια ο περιβάλλον “θόρυβος” μειώνεται δραματικά και οι ακέραιες αρμονικές αφήνονται να δώσουν τον τόνο σε όλη την υπόλοιπη διάρκεια της νότας. Μάλιστα οι μικρές κοιλάδες που παρατηρούνται προδίδουν το φυσικό “βιμπράτο” στο φύσημα του οργανοπαίκτη.

2.4 Δυσαρμονικοί ήχοι (beats) και καχοφωνία

Για δύο υπερτιθέμενους ήχους συντρέχουν λόγοι για τους οποίους ο εγκέφαλος του ακροατή δημιουργεί μία άσχημη αίσθηση καχοφωνίας ή δυσαρμονίας. [9]

Οι λόγοι αυτοί έχουν να κάνουν με την ίδια την ταλάντωση του ήχου. Θεωρώντας δύο απλά ημίτονα $\sin(f_1 t)$ και $\sin(f_2 t)$ όπου η f_1 βρίσκεται κοντά στην f_2 και λαμβάνοντας υπ' όψιν την τριγωνομετρική εξίσωση:

$$\sin a + \sin b = 2 \sin \frac{1}{2}(a+b) \cos \frac{1}{2}(a-b)$$

παίρνουμε

$$\sin(f_1 t) + \sin(f_2 t) = 2 \sin\left[\frac{1}{2}(f_1 + f_2)t\right] \cos\left[\frac{1}{2}(f_1 - f_2)t\right]$$

Ο πρώτος όρος είναι ένα ημίτονο στο μέσο της συχνότητας των δύο συνιστωσών του οποίου όμως το πλάτος ταλαντώνεται πολύ αργά με τη συχνότητα της μισής διαφοράς των πολύ κοντινών τιμών. Το φαινόμενο αυτό ονομάζεται διακρότημα και έχει άμεση συσχέτιση με τη δυσαρμονία στη μουσική [10].

Η παραπάνω φυσική διαδικασία αντανακλάται και από τα χαρακτηριστικά του ανθρώπινου αυτιού. [11] Το σχήμα και η οργάνωσή της βασικής μεμβράνης καταδεικνύει ότι διαφορετικές συχνότητες αντηχούν με διαφορετική ένταση σε διαφορετικά σημεία της. Σύμφωνα με την επικρατούσα θεωρία η βασική μεμβράνη που συζητήθηκε παραπάνω μπορεί να προσεγγιστεί από απλά επικαλυπτόμενα συχνοτικά φίλτρα πεπερασμένου μεγέθους. Τα φίλτρα αυτά είναι μη γραμμικά καθώς εξαρτώνται από την ένταση του ήχου και γίνονται όλο και πιο ευρέα για πιο ψηλές συχνότητες.

Αν δύο ηχητικά στοιχεία συμπέσουν μέσα σε ένα υποτιθέμενο φίλτρο το αυτί “δυσκολεύεται” να τους ξεχωρίσει και δημιουργεί αυτό το αίσθημα δυσφορίας. Σύμφωνα με τους Moore και Glasberg που διεξήγαγαν ακουστικά πειράματα με τη μέθοδο του εγχοπτόμενου ήχου (notched noise method) το εύρος ζώνης αυτών των ιδεατών φίλτρων που ονομάζουν ισοδύναμο τετραγωνικό εύρος ζώνης (Equivalent Rectangular Bandwidth, ERB) [12] διαφέρει ανάλογα με την κεντρική υπό μελέτη συχνότητα και μπορεί να προσεγγιστεί από τη συνάρτηση: $ERB(f)=0.108f+24.7$ [13]. Σύμφωνα μάλιστα με αυτή την προσέγγιση υποστηρίζεται πως η δυνατότητα διάκρισης ήχων μέσα σε αυτά τα όρια είναι ανεξάρτητη από το διακρότημα.

3. ΑΝΑΛΥΣΗ, ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΣΥΝΘΕΣΗ ΗΧΟΥ

Όταν ένα ψηφιακό σύστημα προσπαθεί να μιμηθεί κάποιον ήχο τότε το ζητούμενο είναι η παραγωγή ενός σήματος που θα προσεγγίζει σε ένα βαθμό την κυματομορφή του αρχικού προς μίμηση ήχου. Συνήθως χρησιμοποιείται κάποιο μαθηματικό μοντέλο ή μία συλλογή από αλγορίθμους και συναρτήσεις που αναλύουν τον ήχο στα βασικά του συστατικά και προσπαθούν να τα αναπαράγουν με κάποια πιστότητα. Κοινός παρονομαστής για όλα τα μοντέλα είναι πως κατά τη διαδικασία της ανάλυσης και της ανασύνθεσης υπάρχει απώλεια πληροφορίας η οποία εξαρτάται από τα συγκεκριμένα χαρακτηριστικά, τη συμβατότητα του μοντέλου με τον υπό μοντελοποίηση ήχο αλλά και τις αναπόφευκτες συχνοτικές απώλειες που εισάγουν αθροιστικά τα εργαλεία που χρησιμοποιούνται για την επεξεργασία του ήχου. Στο κεφάλαιο αυτό περιηγούμεστε σύντομα στην ιστορία και τις τεχνικές σύνθεσης ήχου και εισάγουμε το ημιτονοειδές μοντέλο που χρησιμοποιούμε σε επόμενα κεφάλαια για την επεξεργασία των ήχων.

3.1 Η ιστορία της σύνθεσης ήχου

Ίσως το πρώτο παράδειγμα σύνθεσης ήχου εντοπίζεται στα 1897 από τον Thaddeus Cahill ο οποίος δημιούργησε το πρώτο σύστημα που παρήγαγε ήχο αποκλειστικά με ηλεκτρονικό τρόπο εφαρμόζοντας μία πρώτη εκδοχή της προσθετικής σύνθεσης (additive synthesis) που συνοψίζεται στην υπέρθεση βασικών ημιτονοειδών ηλεκτρικών σημάτων για την προσέγγιση ήχων από φυσικά όργανα. Τα πρώτα πειράματα στην ψηφιακή σύνθεση γίνονται στα εργαστήρια της Bell στα τέλη του 1950 όπου εμφανίζονται υβρίδια της wavetable σύνθεσης [14]. Στην ίδια δεκαετία ο Ιάnnης Ξενάκης δίνει τους πρώτους ορισμούς για την Granular σύνθεση και συνθέτει μουσικά κομμάτια χρησιμοποιώντας την. Στον αναλογικό χώρο, στα τέλη του 1960 ο Robert Moog καταφέρνει να κατασκευάσει

synthesizers προσιτά στο ευρύ κοινό. Την ίδια περίοδο ο John Chowning περιγράφει την Frequency Modulation σύνθεση η οποία διαδίδεται ευρέως μετά τη χρήση της από τη Yamaha.

Τη δεκαετία του 1980 οι ηλεκτρονικοί υπολογιστές έχουν πλέον εξαπλωθεί τόσο πολύ που η ψηφιακή σύνθεση ήχου γίνεται υπόθεση πλειάδας επιστημόνων και ερασιτεχνών. Οι υπολογιστές είναι πλέον αρκετά δυνατοί ώστε να υποστηρίξουν τεχνικές όπως η Spectral Modelling σύνθεση του Serra. Δεκάδες τεχνικές ορισμένες ανάλογα με τις δυνατότητες του υλικού ή τις αισθητικές προδιαγραφές των δημιουργών τους κάνουν την εμφάνισή τους. Τα όρια μεταξύ αναλογικής και ψηφιακής σύνθεσης γίνονται δυσδιάκριτα. Πλέον εμφανίζονται ψηφιακά εργαλεία που μοντελοποιούν αναλογικά εργαλεία που μοντελοποιούν φυσικούς ήχους ή ακόμα και ψηφιακά εργαλεία που μοντελοποιούν ψηφιακά εργαλεία.

3.2 Τα είδη της σύνθεσης ήχου

Ακολουθεί μία σύντομη περιγραφή των αρχών λειτουργίας κάποιων βασικών τεχνικών σύνθεσης ήχου.

Στην additive (προσθετική) σύνθεση [15] τίθενται σε εφαρμογή τα βασικά συμπεράσματα του μετασχηματισμού Fourier και στην πιο απλή της μορφή παράγει το σήμα

$$y(t) = \sum_{k=1}^K r_k \cos(2\pi k f_0 t + \varphi_k)$$

για να περιγράψει τις συνιστώσες του υπό κατασκευή ήχου. K είναι ο αριθμός των διαθέσιμων ταλαντωτών.

Στην wavetable σύνθεση οι ταλαντωτές αντικαθίστανται από προαποθηκευμένες κυματομορφές που αντιπροσωπεύουν τη βασική περίοδο ταλάντωσης μίας νότας ενός φυσικού οργάνου.

Στην LA σύνθεση χρησιμοποιούνται μικρά δείγματα για την ατάκα της νότας ενώ η συνέχεια περιγράφεται με subtractive ή με wavetable σύνθεση.

Στην subtractive σύνθεση ένας πλούσιος σε αρμονικές ήχος (συχνά από έναν τετραγωνικό ταλαντωτή) περνάει μέσα από συχνοτικά φίλτρα για να μείνει σαν υπόλοιπο το επιθυμητό αποτέλεσμα.

Στη granular σύνθεση ένας ήχος δειγματίζεται και κόβεται σε πολύ σύντομα κομμάτια, όχι μεγαλύτερα των 50

ms. Αυτά τα κομμάτια που ονομάζονται grains (κόκκοι) υπερτίθενται αφού έχουν υποστεί αλλοιώσεις στην ταχύτητα, τη φάση ή τη συχνότητα.

Στην frequency modulation σύνθεση η συχνότητα κάποιου ταλαντωτή ελέγχεται από έναν δεύτερο. Για να επιτευχθούν αρμονικοί ήχοι ο ελεγκτής ρυθμίζεται σε συχνότητες πολλαπλάσιες αυτής του πρώτου ενώ για μη αρμονικούς σε άλλες. Ο αριθμός και η διάταξη των ταλαντωτών εξαρτάται από την εκάστοτε υλοποίηση.

Ένα πλήθος άλλων υβριδικών τεχνικών έχουν εφευρεθεί χρησιμοποιώντας τις παραπάνω γενικές αρχές. Ιδιαίτερη μνεία πρέπει να γίνει στη σύνθεση με φυσικά μοντέλα όπου κάθε υλοποίηση βασίζεται σε μία μαθηματική ανάλυση και μοντελοποίηση του προς μίμηση φυσικού συστήματος.

3.3 Το ημιτονοειδές μοντέλο

Το μοντέλο ήχου που χρησιμοποιούμε είναι μία υλοποίηση του μοντέλου των McAuley και Quatieri με όνομα MQ [16]. Το μοντέλο MQ ανήκει στην οικογένεια της μοντελοποίησης ήχου που μοντελοποιεί όχι την φυσική λειτουργία της πηγής του ήχου (physical modeling) αλλά το φάσμα του ίδιου του παραγόμενου ήχου (spectral modeling). Είναι λιγότερο ή περισσότερο κατάλληλο για να μοντελοποιήσει κάθε είδους ήχο αλλά αποδίδει πολύ καλύτερα σε μουσικά σήματα με ξεκάθαρο αρμονικό περιεχόμενο. Το μοντέλο μας εκφράζει τους ήχους με το εξής άθροισμα [17]:

$$s(t) = \sum_{r=1}^R A_r \cos[\theta_r(t)] + n(t)$$

όπου A_r και θ_r είναι το στιγμιαίο πλάτος και η φάση του r ημιτόνου και $n(t)$ είναι η συνιστώσα του θορύβου τη στιγμή t .

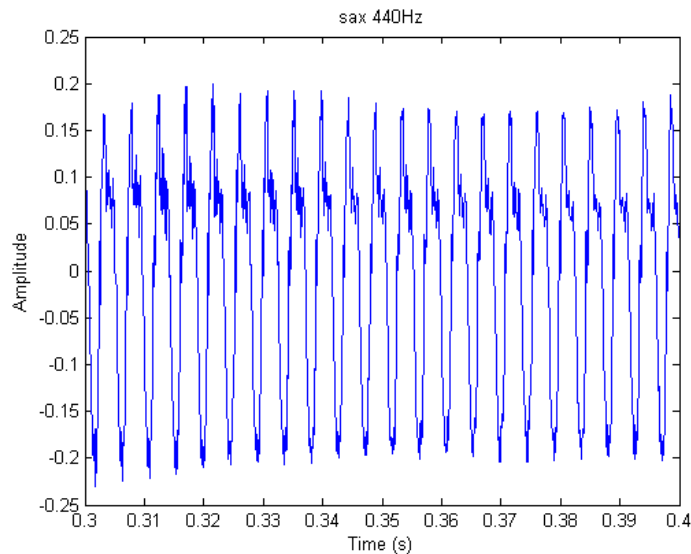
Το μοντέλο θεωρεί ότι τα ημίτονα είναι σταθερές συνιστώσες του σήματος και ότι καθένα έχει ένα αργά μεταβαλλόμενο πλάτος και συχνότητα. Η στιγμιαία φάση θ_r εξάγεται από τη στιγμιαία συχνότητα [18] ως:

$$\theta_r = \int_0^t \omega_r(\tau) d\tau$$

Είναι ένα μοντέλο που δέχεται πάρα πολλές παραμέτρους για την καλύτερη μοντελοποίηση του ήχου όμως θα το εξετάσουμε με τη χρήση συγκεκριμένων σταθερών τιμών και προδιαγραφών μιας και ο τελικός μας στόχος δεν είναι η τελειοποίηση της μοντελοποίησης παρά η χρήση της σαν εργαλείο για την παρέμβαση πάνω στους υπό επεξεργασία ήχους.

Το σήμα ήχου αντιμετωπίζεται σαν άθροισμα ημιτόνων. Τα πλάτη, οι φάσεις και οι συχνότητες των ημιτόνων εξάγονται από έναν μετασχηματισμό Fourier μικρού χρόνου αλλά υψηλής ανάλυσης με τη μέθοδο της εύρεσης κορυφών (peak extraction) από τη σχετική συνάρτηση πλάτους.

Οι ηχογραφήσεις που χρησιμοποιούμε στην εργασία έχουν κωδικοποιηθεί σύμφωνα με το πρωτόκολλο .wav, έχουν συχνότητα δειγματοληψίας 16000Hz και βάθος ανάλυσης 16bits. Για την παρουσίαση του ημιτονοειδούς μοντέλου χρησιμοποιούμε μία νότα Λα (440Hz) από ένα σαξόφωνο. Την επιλέγουμε λόγω του σαφούς αρμονικού της περιεχομένου, του ξεκάθολου μη αρμονικού της “φυσήματος” και του “βιμπράτο” που έχει ηχογραφηθεί και θα κρίνει τη δυνατότητα του συστήματός μας να παρακολουθήσει τις ανεπαίσθητες συχνοτικές αλλαγές.



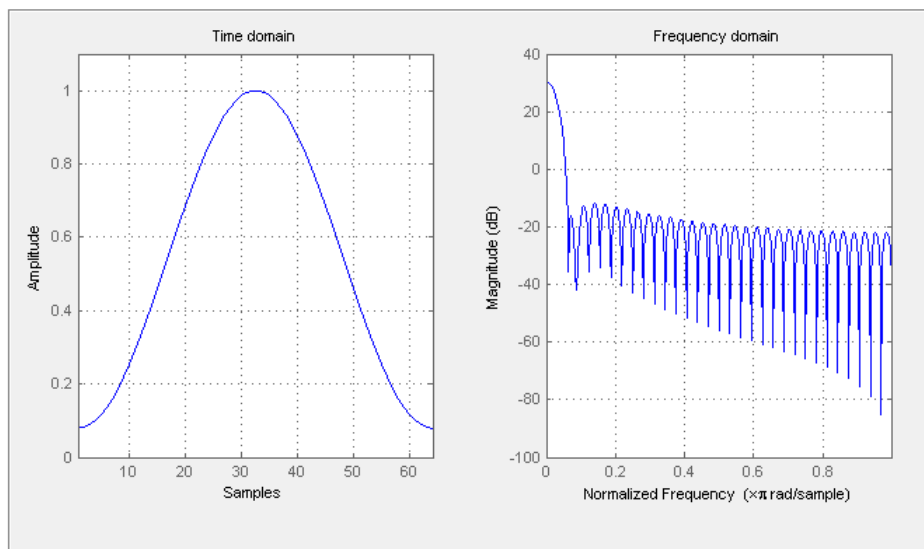
σχήμα 3.1: Λα από σαξόφωνο

Αρχικά θα δημιουργήσουμε το φασματογράφημα του σήματός μας. Επιλέγουμε να χρησιμοποιήσουμε παράθυρα των

256 bits ή αλλιώς 16ms τα οποία δειγματοληπτούν κάθε 128 bits ή αλλιώς 8ms. Οι τιμές αυτές είναι πολύ συνηθισμένες για “τυπική” λειτουργία συστημάτων ήχου αφού “χωρούν” άνετα τις περιόδους των μεσαίων συχνοτήτων χωρίς να είναι ιδιαίτερα μεγάλες και να προκαλούν ασάφεια στη συχνοτική ανάλυση των παραθύρων.

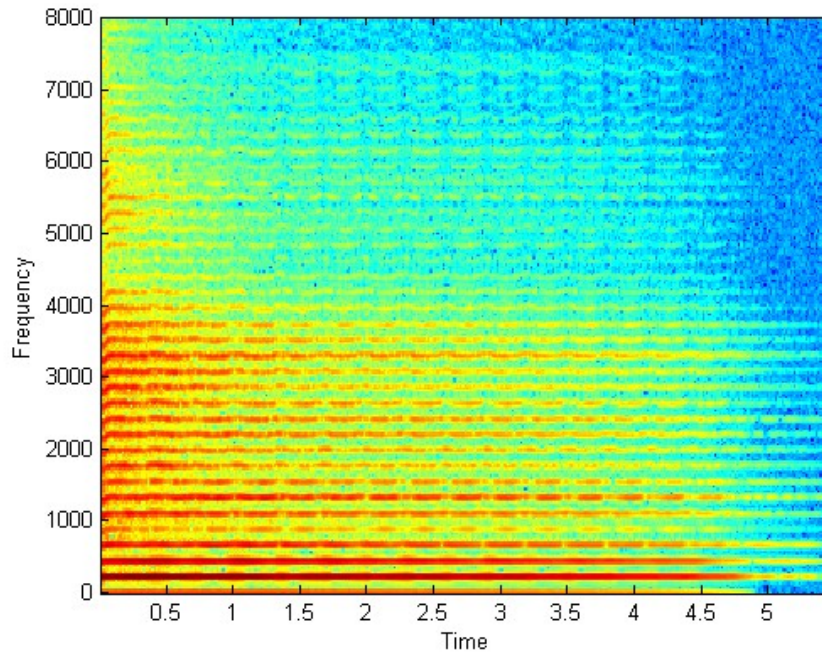
Στη συνέχεια περνούν μέσα από ένα παράθυρο Hamming. Το παράθυρο Hamming περιγράφεται ως

$$w(n) = a - b \cos\left(\frac{2\pi n}{N-1}\right) \text{ όπου } a=0.54, \quad b=1-a=0.46$$



σχήμα 3.2: το παράθυρο Hamming

Ξανά επιλέγεται ως ένα “τυπικό” παράθυρο που έχει ένα σχετικά μικρό εύρος στον κυρίως λοβό αλλά ιδιαίτερα χαμηλές τιμές από τον δεύτερο κιάλας λοβό (leakage factor:0.03%, relative sidelobe attenuation: -42.5 dB, Mainlobe width:0.039063). Πρέπει να σημειωθεί εδώ ότι όταν ο στόχος μας είναι η καλύτερη δυνατή μοντελοποίηση του ήχου το παράθυρο πρέπει να επιλεχθεί με γνώμονα την περίοδο της θεμελίου συχνότητας του υπό επεξεργασία σήματος [19].

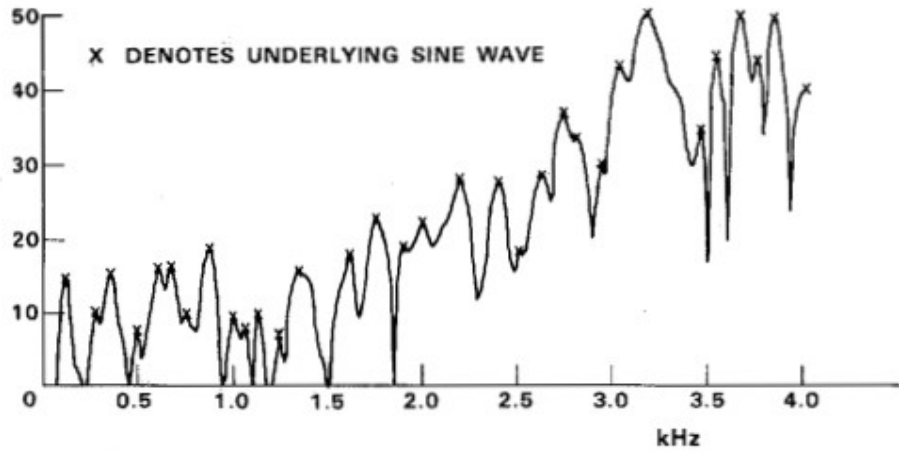


σχήμα 3.3: το φασματογράφημα

Στη συνέχεια υπολογίζεται ο FFT του κάθε καρέ (frame) και έχουμε το φασματογράφημα του υπό επεξεργασία ήχου. Παράλληλα μία δεύτερη διαδικασία λαμβάνει χώρα σε κάθε καρέ. Υπολογίζεται η στιγμιαία συχνότητα [20] (στο μέσον του καρέ) για κάθε ζώνη συχνοτήτων σαν η παράγωγος του χρόνου της φάσης. $f(t) = \frac{1}{2\pi} \varphi'(t)$

Τελικά το σήμα φωτογραφίζεται ανά 128 samples και η συχνοτική ανάλυση μετρά 129 ζώνες συχνοτήτων -μαζί με τη μηδενική- όπου κάθε ζώνη έχει εύρος περίπου 125 Hz.

Στη συνέχεια εντοπίζονται για κάθε καρέ ήχου τα τοπικά μέγιστα στην ένταση. Σκοπός μας είναι να εξάγουμε κανάλια που ακολουθούν την κάθε αρμονική σε όλες τις μικροαλλαγές της στη συχνότητα. Απαιτούμε για τη ύπαρξη ενός καναλιού να ξεπερνά το κατώφλι των 0.1dB και δεν επιτρέπουμε κίνηση μεγαλύτερη των 3 ζωνών συχνοτήτων (ποσότητα που ονομάζουμε Δ). Αν τα τοπικά μέγιστα ήταν ίδια σε αριθμό σε κάθε καρέ δεν θα υπήρχε κανένα πρόβλημα αφού θα τα ταιριάζαμε ένα προς ένα. Επειδή όμως κάτι τέτοιο δεν μπορεί να συμβαίνει εισάγεται η έννοια της γέννησης και του θανάτου ενός καναλιού.

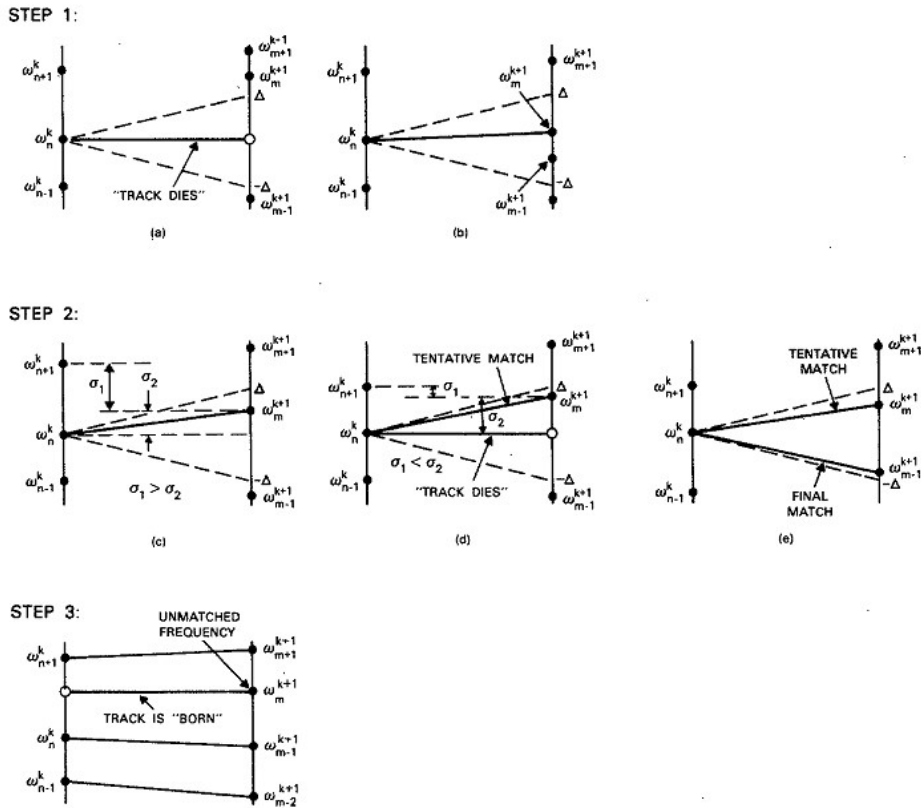


σχήμα 3.3: τα τοπικά μέγιστα στις συχνότητες ενός καρέ

Σκοπός είναι το ταίριασμα κάθε συχνότητας ω_n^k στο καρέ k σε κάποια συχνότητα ω_n^{k+1} στο καρέ $k+1$. Αυτό επιτυγχάνεται σε 3 βήματα:

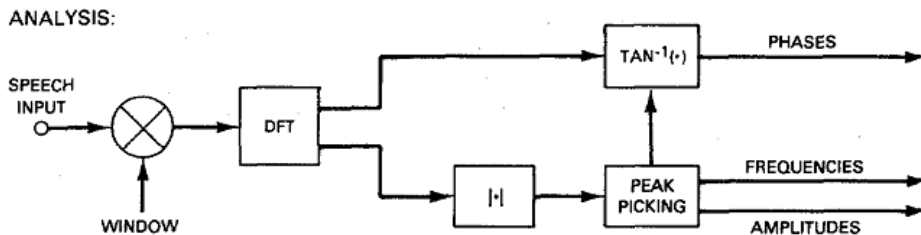
Βήμα 1ο: έστω ότι έχει βρεθεί ταίρι για τις συχνότητες $\omega_0^k, \omega_1^k, \dots, \omega_{n-1}^k$ και ψάχνουμε κάποιο για την ω_n^k . Αν όλες οι συχνότητες του $k+1$ βρίσκονται έξω από το διάστημα $|\omega_n^k - \omega_m^{k+1}| \geq \Delta$ το κανάλι θεωρείται νεκρό. Αν αντίθετα υπάρχει συχνότητα ω_{n+1}^{k+1} στο $k+1$ και είναι η κοντινότερη ώστε $|\omega_n^k - \omega_m^{k+1}| < |\omega_n^k - \omega_i^{k+1}| < \Delta$ τότε η ω_m^{k+1} θεωρείται υποψήφιο ταίρι. Το αν θα γίνει μόνιμο φαίνεται στο βήμα 2.

Βήμα 2ο: Αν το ω_n^k έχει προσωρινά ταιριάζει με το ω_m^{k+1} και ισχύει $|\omega_m^{k+1} - \omega_n^k| < |\omega_m^{k+1} - \omega_{i+1}^k|$ για $i > n$ τότε είναι το τελικό του ταίρι και η διαδικασία συνεχίζεται στο βήμα 1 για ω_{n+1}^k . Αν κάτι τέτοιο δεν ισχύει τότε η ω_m^{k+1} στο καρέ $k+1$ ταιριάζει καλύτερα στην ω_{n+1}^k του k απ' ό τι στην ω_n^k . Τότε δύο περιπτώσεις πρέπει να εξεταστούν. Στην πρώτη η συχνότητα που απομένει ω_{m-1}^{k+1} (αν υπάρχει) βρίσκεται έξω από τα όρια ταιριασματος και το κανάλι που σχετίζεται με την ω_n^k πρέπει να θεωρηθεί νεκρό. Στην άλλη περίπτωση η ω_{m-1}^k είναι μέσα στα επιθυμητά όρια και έχουμε τελικό ταίριασμα. Και στις δύο περιπτώσεις επιστρέφουμε στο βήμα 1 με τη συχνότητα ω_{n+1}^k



σχήμα 3.4: οι διαδικασίες γέννησης και θανάτου των καναλιών

Βήμα 3ο: Όταν όλες οι συχνότητες του καρέ k έχουν ελεγχθεί είναι πιθανόν να περισσεύουν συχνότητες στο $k+1$. Τότε για μία υποτιθέμενη τέτοια συχνότητα ω_m^{k+1} θεωρούμε πως γεννιέται ένα νέο κανάλι.

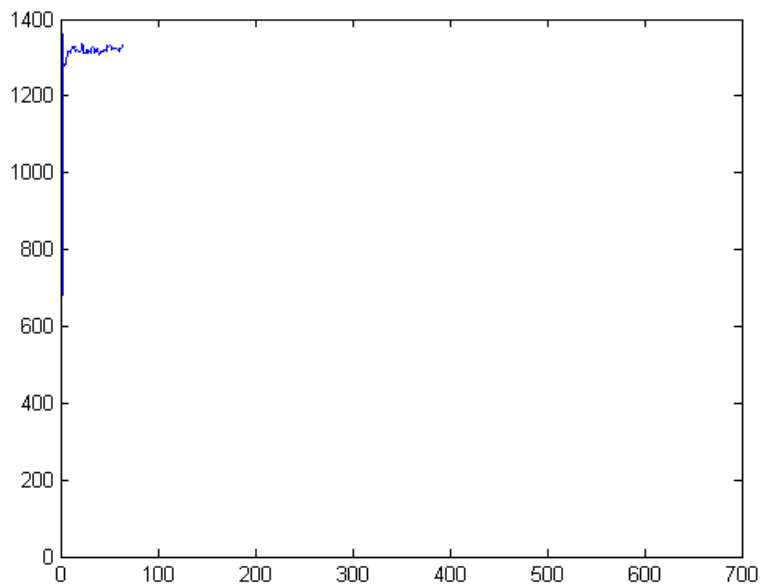


σχήμα 3.5

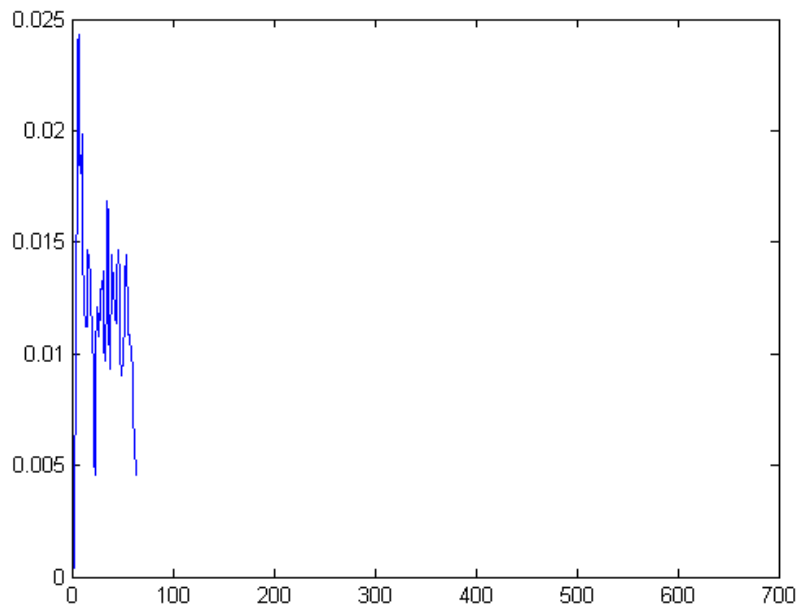
Η έξοδος αυτής της διαδικασίας είναι τιμές συχνότητας, πλάτους και φάσης για όσα κανάλια εντοπίστηκαν. Ένα

διάγραμμα της παρουσιάζεται στο σχήμα 3.5

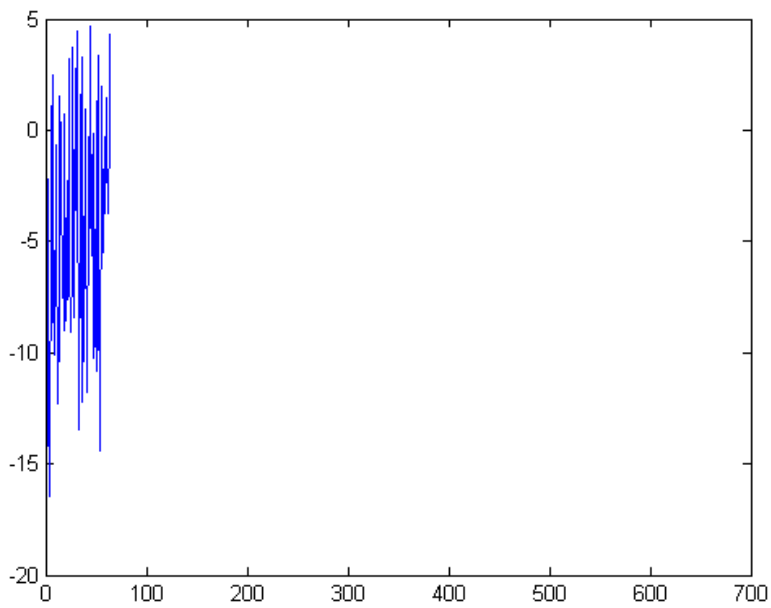
Αυτά τα δεδομένα χρησιμοποιούνται για την ανασύνθεση του αρμονικού μέρους του αρχικού σήματος. Ένα παράδειγμα ενός τέτοιου καναλιού για την περίπτωση του ήχου που μελετάμε φαίνεται στα σχήματα 3.6-8. Το κανάλι αυτό ακολουθεί την αρμονική των 1320Hz η οποία είναι έντονη στα πρώτα μόνο samples του ήχου γι' αυτό και το κανάλι πεθαίνει στη συνέχεια. Είναι σημαντικό να παρατηρήσουμε πως αντίθετα με αυτό που θα περιμέναμε από τη θεωρία οι πραγματικές τιμές της εν λόγω αρμονικής δεν είναι σταθερές πάνω στους 1320 κύκλους καθώς επηρεάζονται από πολλές μικροαλλοιώσεις.



σχήμα 3.6: διάγραμμα δειγμάτων-συχνότητας



σχήμα 3.7: διάγραμμα δειγμάτων-πλάτους



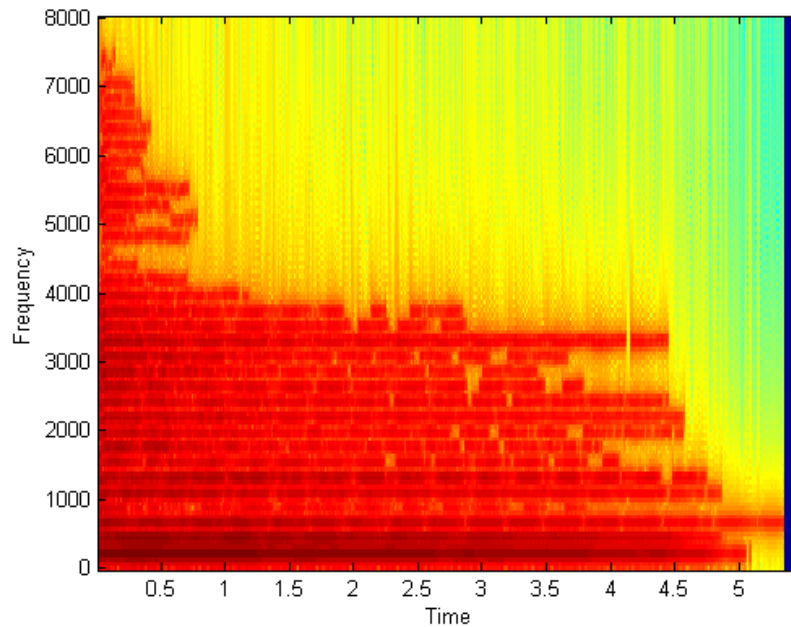
Σχήμα 3.8: διάγραμμα δειγμάτων-φάσης

Παρά τη διαίσθηση που θα επέβαλλε την ανασύνθεση του k -στού καρέ σύμφωνα με την εξίσωση

$$s(n) = \sum_{l=1}^{L(k)} A_l^k \cos[n\omega_l^k + \theta_l^k]$$

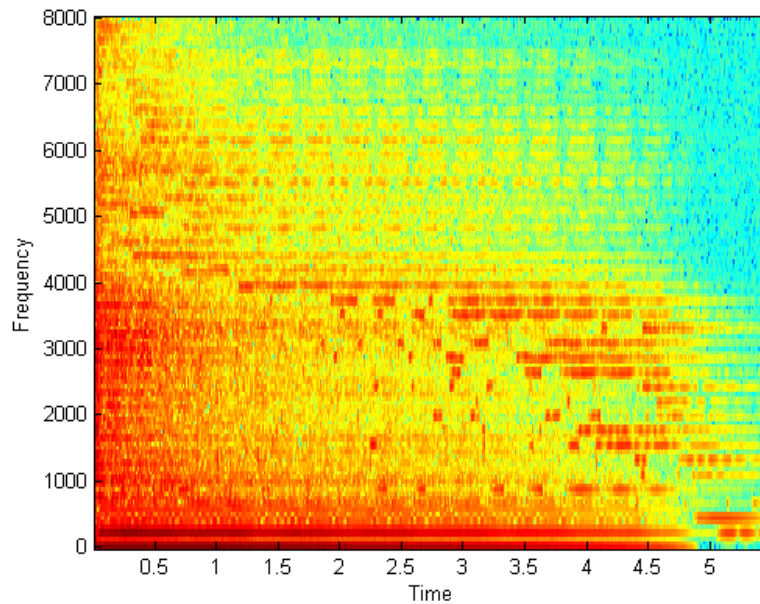
η χρονικά μεταβαλλόμενη φύση των παραμέτρων οδηγεί σε ανακολουθίες στα όρια των καρέ. Η λύση σε αυτό το πρόβλημα είναι η υπέρθεση ανά κανάλι των διαφορετικών καρέ με την αίρεση όμως ότι κάθε ένα από αυτά έχει περάσει μέσα από ένα τριγωνικό παράθυρο που αφήνει αναλλοίωτη την κεντρική του τιμή αλλά συνεισφέρει ανάλογα με τη θέση του μαζί με τα γειτονικά παράθυρα για τις ενδιάμεσες τιμές. Τα τριγωνικά αυτά παράθυρα έχουν μήκος διπλό από αυτό του υπό επεξεργασία καρέ.

Τελικά οι επεξεργασμένες αρμονικές συντίθενται μία προς μία και στη συνέχεια υπερτίθενται για να μας δώσουν την τελική προσέγγιση του αρχικού σήματος όπως φαίνεται στο σχήμα 3.9.



σχήμα 3.9: το ανασυντεθειμένο σήμα

Στη συνέχεια αφαιρούμε το συντεθειμένο σήμα από το αρχικό για να απομονώσουμε το μη αρμονικό, θορυβώδες κομμάτι του σήματος. [21]



σχήμα 3.10: ο εναπομείνας θόρυβος

$n(t) = x(t) - s(t)$ το οποίο φαίνεται στο σχήμα 3.10.

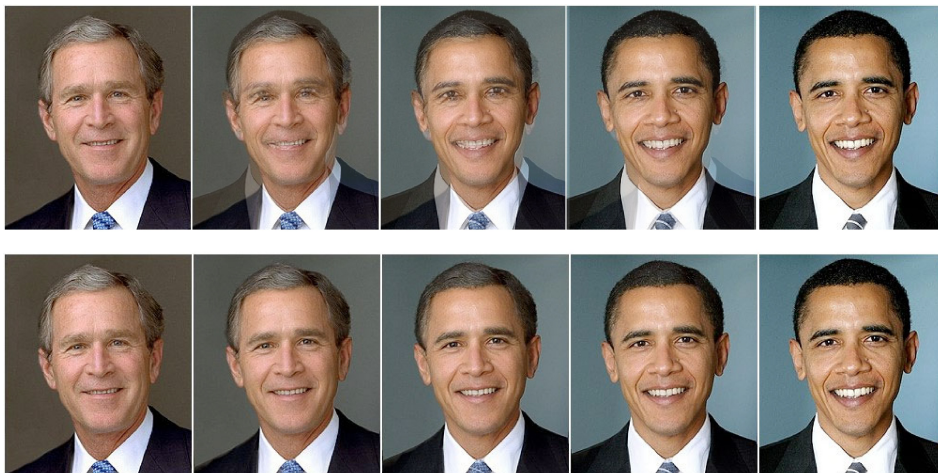
4. Η ΜΕΤΑΜΟΡΦΩΣΗ ΗΧΟΥ

Στο κεφάλαιο αυτό εξηγούμε τι είναι η μεταμόρφωση ήχου, παρουσιάζουμε τον αλγόριθμο Athena και το σετ προβλημάτων που προσπαθεί να επιλύσει και εξηγούμε τις λεπτομέρειες της υλοποίησης και τα προβλήματα που αντιμετωπίσαμε κατά την ανάπτυξή του.

4.1 Τι είναι και τι δεν είναι η μεταμόρφωση ήχου

Δεδομένων δύο ήχων A και B μεταμόρφωση ήχου (sound morphing) ονομάζουμε την σταδιακή αλλαγή κατά τη διάρκεια του χρόνου του ηχοχρώματος του ήχου A με τρόπο τέτοιο ώστε να φτάσει στο ηχοχρώμα του B [22]. Δεν πρέπει να συγχέεται με το χιαστί σβήσιμο (cross fade) κατά το οποίο ο ήχος A χαμηλώνει σταδιακά σε ένταση ενώ ο ήχος B δυναμώνει από μηδενική ένταση για να φτάσει την πλήρη του ένταση μόλις σβήσει εντελώς ο ήχος A.

Στην εικόνα 4.1 βλέπουμε τα οπτικά ανάλογα ενός cross fade και μίας μεταμόρφωσης ήχου για δύο εικόνες. Αυτό που “ενοχλεί” στην παραπάνω εικόνα είναι το γεγονός πως τα στοιχεία που σαν γενικεύσεις είναι κοινά ανάμεσα στις δύο



σχήμα 4.1: τα οπτικά ανάλογα ενός cross fade (πάνω) και ενός morph (κάτω)

εικόνες (π.χ. η γραμμή των ώμων ή το χαμόγελο) δεν μπορούν

να συνυπάρξουν στην ίδια εικόνα και επικαλύπτονται (στην περίπτωση του ήχου θα λέγαμε πως είναι δυσαρμονικά). Αντίθετα η περίπτωση της μεταμόρφωσης επιτρέπει ενδιάμεσες καταστάσεις που συνυπολογίζουν τιμές και από τις δύο εικόνες. [23]

Η λογική του cross fading στον ήχο μπορεί να συμβολιστεί από μία συνάρτηση.

$$C(t) = a(t)A(t) + [1 - a(t)]B(t)$$

4.2 Η δομή του αλγορίθμου μεταμόρφωσης ήχου Athena

Ο αλγόριθμος Athena έχει σαν αντικείμενο την επίλυση μιας συγκεκριμένης υποκατηγορίας προβλημάτων στη μεταμόρφωση ήχου αν και παρουσιάζει ενδιαφέροντα αποτελέσματα και σε άλλες περιπτώσεις. Μας αφορούν κυρίως μουσικά σήματα προερχόμενα από φυσικά ή ψηφιακά όργανα που έχουν σαφή θεμέλιο συχνότητα ή συγχορδίες περισσότερων της μίας νοτών. Μάλιστα ο αρχικός στόχος ήταν να μελετηθούν προβλήματα όπου οι ήχοι A και B έχουν ασύμβατο ή λίγο συμβατό αρμονικό περιεχόμενο και να προταθεί μία λύση που θα εξομαλύνει αυτές τις διαφωνίες. Η υλοποίησή του βασίζεται στην MQ ημιτονοειδή ανάλυση/σύνθεση που παρουσιάστηκε στην παράγραφο 3.3 και ειδικά στην υλοποίηση του Dan Ellis (η οποία μάλιστα χρησιμοποιήθηκε σε βιβλιοθήκη μοντελοποίησης ήχου) και έχει γίνει σε περιβάλλον MATLAB.

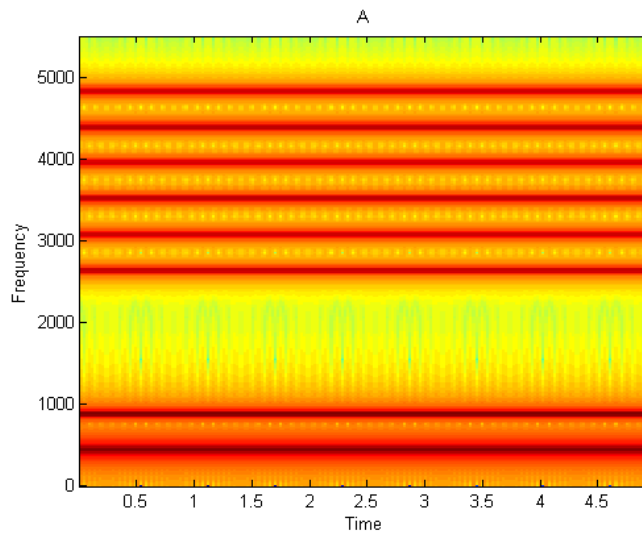
Εν συντομία η δομή του είναι η εξής (θα αναλυθεί στην επόμενη παράγραφο μαζί με οπτικά παραδείγματα): Οι ήχοι A και B υφίστανται ξεχωριστά την ανάλυση MQ ώστε να εξαχθούν οι τιμές που μοντελοποιούν την ημιτονοειδή συνιστώσα και ο θόρυβος. Στη συνέχεια οι αρμονικές* τους ταιριάζουν με τέτοιο τρόπο ώστε αυτές που συνηχώντας θα προκαλέσουν δυσαρμονία (beating) να υποστούν περαιτέρω επεξεργασία. Οι δυσαρμονίες εξομαλύνονται και οι υπόλοιπες αρμονικές αφήνονται να συνηχήσουν ελεύθερα. Στα κομμάτια των ήχων A και B που δεν βρίσκονται μέσα στο διάστημα μεταμόρφωσης παρεμβάλλεται το συνθετικό περιεχόμενο που αντιστοιχεί στο διάστημα αυτό. Τέλος στο ανασυντεθειμένο

* Με την αγγλική έννοια του overtone όπου $2f_0$ είναι η πρώτη αρμονική, $3f_0$ είναι η δεύτερη αρμονική κοκ.

σήμα προστίθεται ο ελαφρά επεξεργασμένος θόρυβος ώστε να τελειοποιηθεί το ηχητικό αποτέλεσμα και ο τελικός ήχος C επιστρέφεται στο χρήστη.

4.3 Οι λεπτομέρειες της υλοποίησης

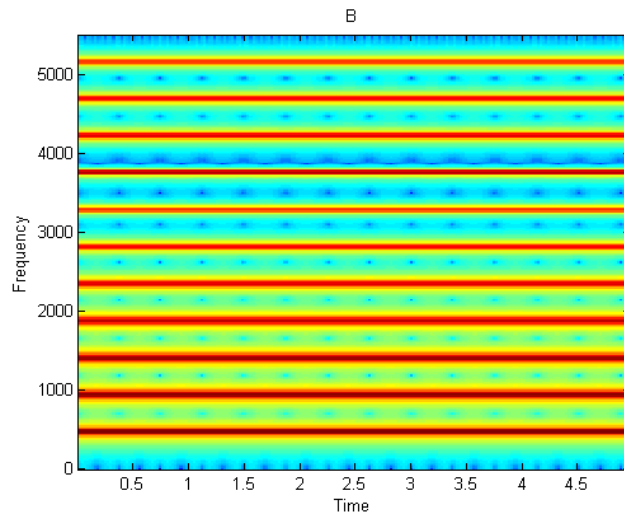
Για τις ανάγκες της παρουσίασης του αλγορίθμου κρίνεται απαραίτητη η χρήση οπτικών παραδειγμάτων ώστε να φανούν οι λεπτομέρειες και τα επιμέρους προβλήματα που χρειάστηκε να αντιμετωπίσουμε κατά την υλοποίηση. Χρησιμοποιούμε δύο συνθετικούς ήχους A και B με θεμέλιο συχνότητα στα 440 και 470 Hz αντίστοιχα και με έντονο περιεχόμενο στις πρώτες 10 αρμονικές. Σημειώνεται ότι ο ήχος A δεν έχει αρμονικές στις θέσεις 3, 4 και 5 ενώ ο ήχος B είναι πλήρης μέχρι την 10η αρμονική και περιέχει μία εμβόλιμη αρμονική στα 3520Hz που αντιστοιχεί στην 8η αρμονική του ήχου A. Θα παρουσιάσουμε μία μεταμόρφωση που λαμβάνει χώρα από το 1sec μέχρι τα 4sec.



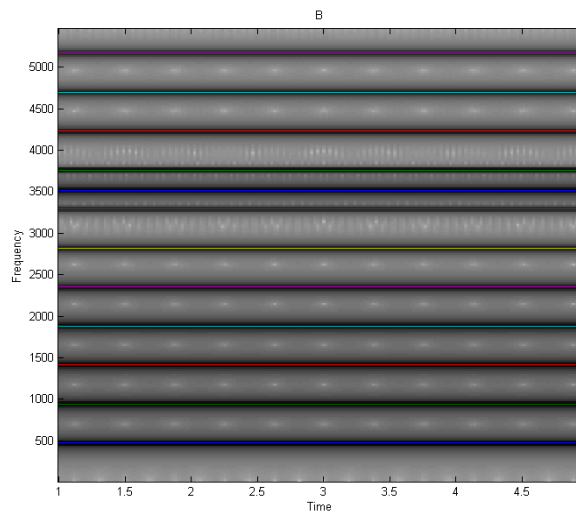
σχήμα 4.2: ο συνθετικός ήχος A

4.3.1 Ο STFT και η συγχώνευση καναλιών

Οι δύο αυτοί ήχοι περνάνε ξεχωριστά τη διαδικασία του Short Time Fourier Transform. Στη συνέχεια υπολογίζονται τα κανάλια που αντιπροσωπεύουν τις ξεχωριστές αρμονικές αλλά υπολογίζεται συγχρόνως και η γωνιακή συχνότητα κάθε



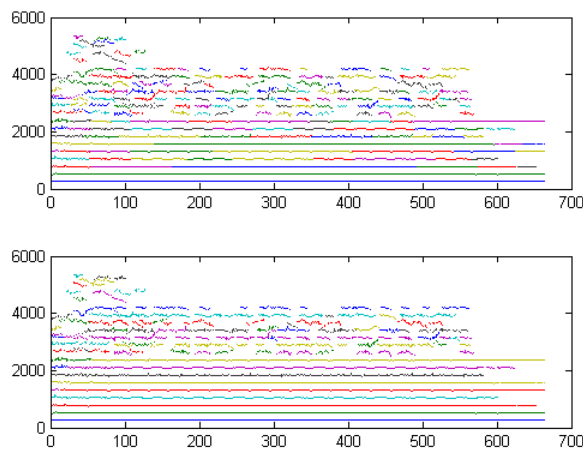
σχήμα 4.3: ο συνθετικός ήχος B



σχήμα 4.4: τα κανάλια του ήχου B

καναλιού με συχνότητα ίση με την συχνότητα επιβολής κάδρων (frames) του STFT. Στο σχήμα 4.4 φαίνονται τα κανάλια που εντοπίστηκαν στον ήχο Β.

Εδώ υπεισέρχεται και το πρώτο πρόβλημα της υλοποίησης. Όπως αναφέρθηκε και στην παράγραφο 2.3 το μοντέλο MQ θεωρεί κανάλια τα οποία γεννιούνται και πεθαίνουν ανάλογα με τα πλάτη τους. Αυτό όμως σημαίνει πως σε φυσικά σήματα μία αρμονική που χάνει την ενέργειά της στη διάρκεια του χρόνου μόνο για να επανέλθει λίγα frames αργότερα θα απαιτήσει τη δημιουργία δύο ή και περισσότερων ξεχωριστών καναλιών όπως φαίνεται και σε ένα δείγμα από φλάουτο στο σχήμα 4.5. Επιλέξαμε λοιπόν να συγχωνεύσουμε



σχήμα 4.5: κανάλια από φλάουτο πριν και μετά τη συγχώνευση

τα κανάλια με τρόπο τέτοιο ώστε μία αρμονική να καλύπτεται από ένα κανάλι. Για να γίνει αυτό παίρνουμε τον μέσο όρο των συχνοτήτων κάθε καναλιού και τα εξετάζουμε εξαντλητικά από το πρώτο μέχρι το τελευταίο ώστε να βρούμε τα κανάλια των οποίων η μέση συχνότητα βρίσκεται μέσα σε μία στενή ζώνη γύρω από την υπό εξέταση μέση συχνότητα. Σε τέτοια περίπτωση η πληροφορία για τη συχνότητα, το πλάτος και τις φάσεις του δεύτερου καναλιού αντιγράφεται στο πρώτο και εν συνεχεία διαγράφεται από τη λίστα των καναλιών. Στο σχήμα 4.5 παρουσιάζεται και η συγχωνευμένη εκδοχή όπου τα κανάλια ξεχωρίζουν από το κοινό τους χρώμα.

4.3.2 Το ταίριασμα των δυσαρμονικών καναλιών

Μετά τη συγχώνευση των καναλιών με βάση τη μέση συχνότητά τους σειρά έχει μία εκ νέου σύγκριση των καναλιών αυτή τη φορά όμως μεταξύ των ήχων A και B. Η λογική μας εδώ είναι πως φάχνουμε για τα κανάλια τα οποία αν συνηχήσουν θα παράγουν δυσαρμονία. Για να γίνει αυτό υπολογίζεται ξανά ο μέσος όρος της συχνότητας κάθε καναλιού, αυτή τη φορά όμως όχι για όλη τη διάρκεια του χρόνου αλλά για το διάστημα για το οποίο τα A και B θα πρέπει να μεταμορφωθούν το ένα στο άλλο ή πιο απλά να συνηχήσουν. Εδώ για κάθε υπό εξέταση κανάλι του ήχου A ορίζουμε μία διαφορετική ζώνη συχνοτήτων μέσα στην οποία θα “απαγορέψουμε” σε κάποιο κανάλι του ήχου B να ηχήσει. Αυτή η ζώνη ορίζεται από το μοντέλο ERB που περιγράψαμε στην παράγραφο 1.3 και είναι ίση με $ERB(f)=0.108 f+24.7$. Βρίσκοντας ένα κανάλι του ήχου B σε μικρότερη απόσταση από το 1/2 του ERB την ορίζουμε σαν νέα ελάχιστη απόσταση και φάχνουμε τα επόμενα κανάλια για κάποια ακόμη μικρότερη. Όταν έχουμε βρει την πιο μικρή τα δύο κανάλια (ένα από τον ήχο A και ένα από τον ήχο B) συνταιριάζονται σε ένα ζευγάρι το οποίο θα πρέπει να υποστεί περαιτέρω επεξεργασία για να αποφευχθεί η δυσαρμονία. Όταν όλα τα “προβληματικά” κανάλια έχουν ταιριάξει σημειώνουμε για τα υπόλοιπα πως θα πρέπει να αντιμετωπιστούν διαφορετικά.

4.3.3 Η σύνθεση της μεταμόρφωσης για τα δυσαρμονικά κανάλια

Το επόμενο και βασικότερο βήμα της επεξεργασίας αφορά αποκλειστικά το διάστημα της συνήχησης και μεταμόρφωσης των δύο ήχων. Χωρίζεται σε δύο μέρη με βάση το προηγούμενο. Το πρώτο μέρος αφορά τα ζεύγη καναλιών που θα δημιουργούσαν δυσαρμονία. Μία βηματική συνάρτηση εξετάζει μία μία τις τρεις διαστάσεις των καναλιών ξεκινώντας από τις συχνότητες. Ορίζουμε μία μεταβλητή

$$k_i=1-(i \frac{1}{N}) \text{ με } k=1,2,\dots,N \text{ και } N=t_{morph}/framerate$$

και θέτουμε τις συχνότητες του νέου καναλιού

$$f_{morph}(i) = f_1(i)k(i) + f_2(1 - k(i))$$

ώστε στην αρχή του χρόνου μεταμόρφωσης να συνεισφέρει περισσότερο η f_1 και στο τέλος η f_2 . Στην περίπτωση που ένα από τα κανάλια δεν έχει πληροφορία για το συγκεκριμένο frame (κάτι που σημαίνει πως το κανάλι έχει ίσως προσωρινά σκοτωθεί) αντιγράφεται η πληροφορία του άλλου απ' ευθείας στο νέο κανάλι. Αποτέλεσμα της παραπάνω μεθόδου είναι πως τα δυσαρμονικά κανάλια τελικά μεταμορφώνονται το ένα στο άλλο.

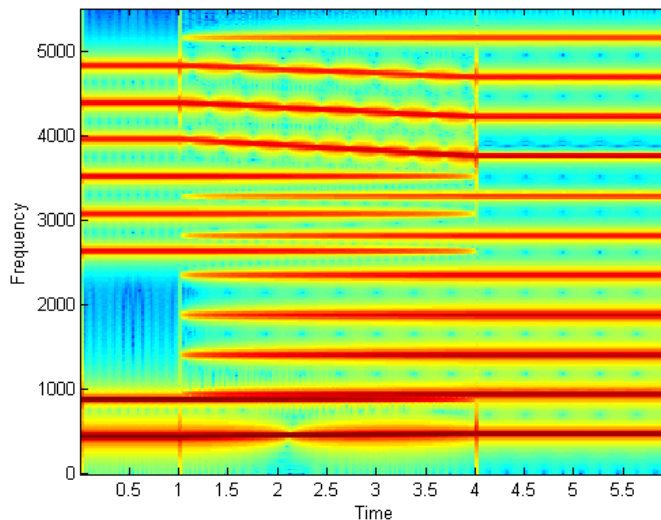
Η δεύτερη υπό επεξεργασία συνιστώσα του κάθε καναλιού είναι το πλάτος ή καλύτερα η περιβάλλουσα πλάτους. Εδώ η αντιμετώπιση είναι πανομοιότυπη με τις συχνότητες. Με τον ίδιο ορισμό του k οι τιμές πλάτους για το νέο κανάλι είναι

$$m_{morph}(i) = m_1(i)k(i) + m_2(1 - k(i))$$

. Ξανά όταν λείπει πληροφορία για το ένα από τα δύο κανάλια των πηγών (αυτό φυσικά θα είναι το ίδιο με της συχνότητας) αντιγράφεται στο τελικό σήμα αυτή που είναι παρούσα.

4.3.4 Το πρόβλημα της φάσης

Τέλος, η τρίτη και πιο περίπλοκη υπό επεξεργασία συνιστώσα είναι αυτή της φάσης. Οι τιμές της φάσης είναι



σχήμα 4.6: μία "κακή" επεξεργασία της φάσης δημιουργεί ασυνέχειες στο σήμα

ιδιαίτερα ευαίσθητες στην επεξεργασία, μάλιστα όπως θα φανεί

και παρακάτω αν ένα συνθετικό σήμα βρεθεί εκτός φάσης σε σχέση με το αρχικό καθίσταται αδύνατη η καλή προσέγγιση του μοντελοποιημένου σήματος. Η αντιμετώπιση που προτείνεται εδώ είναι η εξής. Οι τιμές της φάσης p διατηρούνται ανέπαφες μέχρι τη στιγμή που ξεκινάει η μεταμόρφωση. Επιπλέον το πρώτο frame της μεταμόρφωσης τίθεται ίσο με τη φάση του καναλιού που προέρχεται από το σήμα A όπως άλλωστε και για τη συχνότητα αφού $f_{morphstart} = f_1 * 1 + f_2 * 0$.

Ανατρέχοντας στον ορισμό της φάσης ενός διακριτού περιοδικού σήματος βλέπουμε πως $\varphi(nT) = \varphi((n-1)T) + 2\pi f(nT)$. Ορίζουμε λοιπόν πως από το δεύτερο frame και έπειτα τη φάση p αντιστοίχως με αποτέλεσμα να συνθέτουμε τη φάση όλου του διαστήματος μεταμόρφωσης σύμφωνα με την f_{morph} .

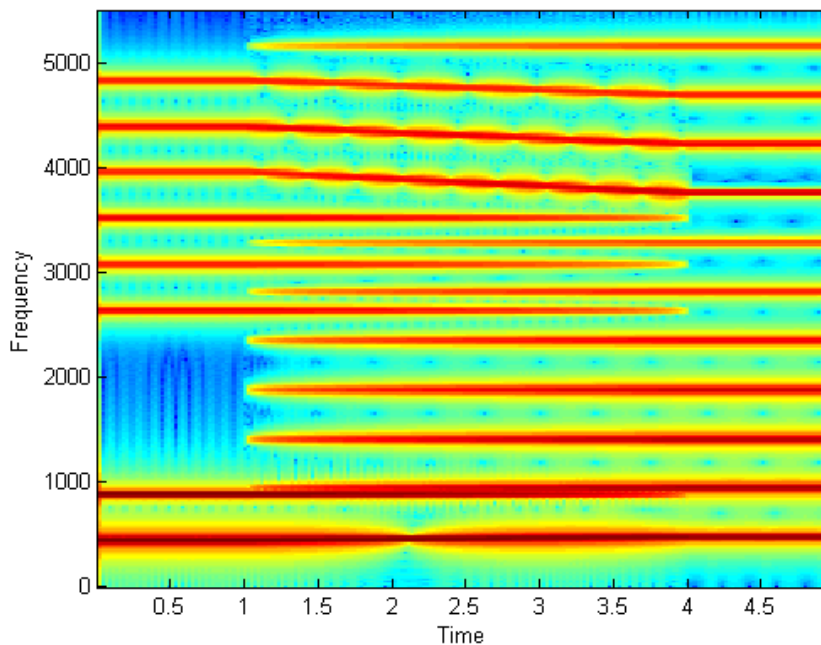
Ένα νέο πρόβλημα προκύπτει στο τέλος του διαστήματος μεταμόρφωσης. Η συνθετική πλέον φάση έχει πάψει να παρακολουθεί τις φυσικά μετρημένες τιμές των αυθεντικών σημάτων με αποτέλεσμα να δημιουργείται ασυνέχεια στο σημείο όπου το σήμα B πρέπει να συνεχίσει να ηχεί μόνο του. Ένα τέτοιο παράδειγμα μπορούμε να δούμε στο σχήμα 4.6. Η λύση εδώ είναι να εξετάσουμε την τελευταία τιμή της φάσης πριν το τέλος της μεταμόρφωσης και να προσθέσουμε στην επόμενη τιμή της (που αντιστοιχεί στη φάση του σήματος B) τη διαφορά. Η λύση αυτή στο πρόβλημα των φάσεων έχει ένα προφανές κόστος ότι το τρίτο μέρος του συνθετικού σήματος που αντιστοιχεί στο καθαρό σήμα B είναι εκτός φάσης σε σχέση με το αρχικό σήμα αφού κληρονομεί τη φάση του δεύτερου μέρους του συνθετικού σήματος (δηλαδή της μεταμόρφωσης). Αυτό όμως δύσκολα γίνεται αντιληπτό από τον ακροατή αφού όπως υποστηρίζουν οι Lipshitz et al [24] οι διαφορές στη φάση δεν είναι εν γένει αντιληπτές από το ανθρώπινο αυτί σε μουσικά σήματα (κάτι που δεν ισχύει για σήματα φωνής) ειδικά μάλιστα στο περιοδικό μέρος μίας νότας δηλαδή τις περιπτώσεις που αφορούν το σήμα B.

4.3.5 Η σύνθεση της μεταμόρφωσης για τα κανάλια που δεν προκαλούν δυσαρμονία

Εδώ μας έχουν απομείνει τα κανάλια που αν συνηχίσουν δεν προκαλούν δυσαρμονία. Ουσιαστικά αφού έχουμε επεξεργαστεί το “δύσκολο” μέρος του σήματος ο στόχος μας είναι τα κανάλια που προέρχονται από τον ήχο A να σβήσουν

σταδιακά κατά τη διάρκεια της μεταμόρφωσης ενώ τα κανάλια που θα πρέπει να ηχήσουν όταν θα έχει τελειώσει το διάστημα της μεταμόρφωσης πρέπει να εισαχθούν ομαλά. Ο τρόπος για να το πετύχουμε αυτό είναι ο εξής: Κατά τη διάρκεια της μεταμόρφωσης αφήνουμε όλες τις τιμές των καναλιών αυτών πλην μίας ανέπαφες. Στις τιμές του πλάτους όμως γίνεται μία απλή επεξεργασία.

Για τα κανάλια του ήχου Α έχουμε: $m_{morph}(i) = m_1 k(i)$



σχήμα 4.7: το μεταμορφωμένο σήμα

ενώ για τα κανάλια του ήχου Β έχουμε: $m_{morph}(i) = m_2(1 - k(i))$

Στο σημείο αυτό για κάποιες περιπτώσεις σημάτων προέκυψε το αισθητικό πρόβλημα ότι τα εν λόγω κανάλια “έκρουσαν” από τον ακροατή την μεταμόρφωση των δυσαρμονικών ήχων και θόλωναν το αποτέλεσμα. Καταλήξαμε στην προσθήκη στον αλγόριθμο μίας επιπλέον παραμέτρου στην οποία μπορεί να υψωθεί το k ώστε το σβήσιμο και η εισαγωγή των μη δυσαρμονικών καναλιών να γίνεται πιο δραματικά (πιο σύντομα στο χρόνο) και να αφήνει τα δυσαρμονικά κανάλια να

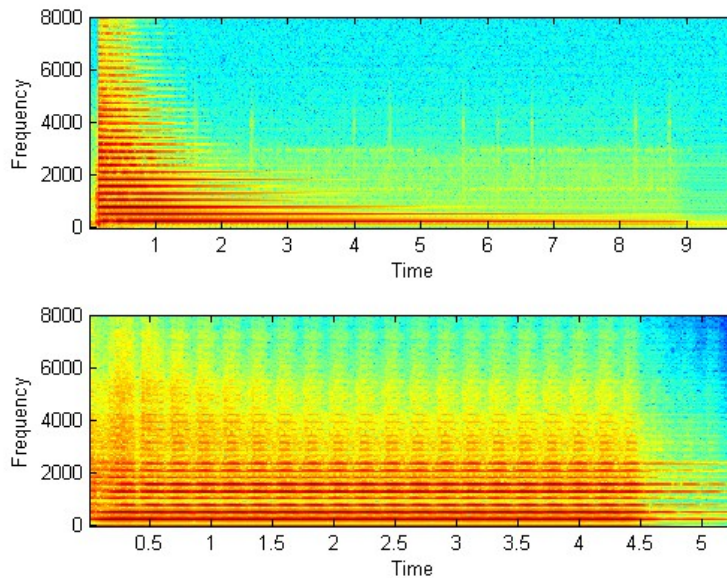
δίνουν τον τόνο στη μεταμόρφωση. Για τα κανάλια του ήχου A έχουμε:

$$m_{morph}(i) = m_1 k(i)^{power} \quad \text{ενώ για τα κανάλια του ήχου B έχουμε:}$$

$$m_{morph}(i) = m_2 (1 - k(i))^{power}$$

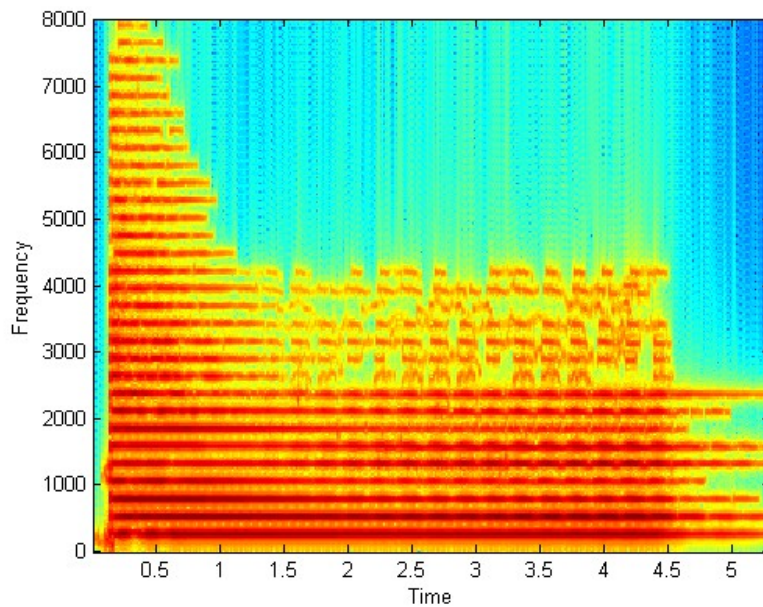
4.3.6 Η σύνθεση των τριών μερών του νέου ήχου

Τελειώνοντας τις παραπάνω διαδικασίες έχουμε στα χέρια μας τις τιμές φάσης, συχνότητας και πλάτους για τα τρία μέρη του νέου ήχου. Αυτά είναι ο ήχος A μέχρι την αρχή του διαστήματος μεταμόρφωσης, το ίδιο το διάστημα της μεταμόρφωσης και το μέρος του ήχου B που ξεκινά από το τέλος του διαστήματος μεταμόρφωσης και φτάνει μέχρι το τέλος του ήχου B. Τα στοιχεία αυτά ενώνονται και τροφοδοτούνται στον αλγόριθμο της σύνθεσης ήχου. Το αποτέλεσμα αυτής της διαδικασίας είναι ένας νέος ήχος που περιλαμβάνει το αρμονικό μέρος του τελικού ήχου που κατασκευάζουμε. Στην εικόνα 4.7 βλέπουμε πώς οι αρμονικές που θα προκαλούσαν δυσαρμονία έχουν μεταμορφωθεί η μία στην άλλη αλλά και πώς αυτές που δεν την προκαλούν αφήνονται να συνηχήσουν, όμως εισάγονται και σβήνουν σταδιακά στα όρια του χρόνου μεταμόρφωσης.



σχήμα 4.8: Ντο από ηλεκτρική κιθάρα και φλάουτο

Αφού έχουμε εξηγήσει τη λειτουργία του αλγορίθμου για συνθετικά σήματα πρέπει να εισάγουμε κάποια φυσικά σήματα ώστε να παρουσιάσουμε τη διαχείριση που έχει γίνει στο θέμα του θορύβου στην επόμενη παράγραφο. Στο σχήμα 4.8 βλέπουμε μία νότα Ντο από ηλεκτρική κιθάρα και φλάουτο. Ο αλγόριθμος μας βρίσκει 16 κανάλια που θα προκαλέσουν δυσαρμονία και καταλήγει στο συνθετικό σήμα που βλέπουμε



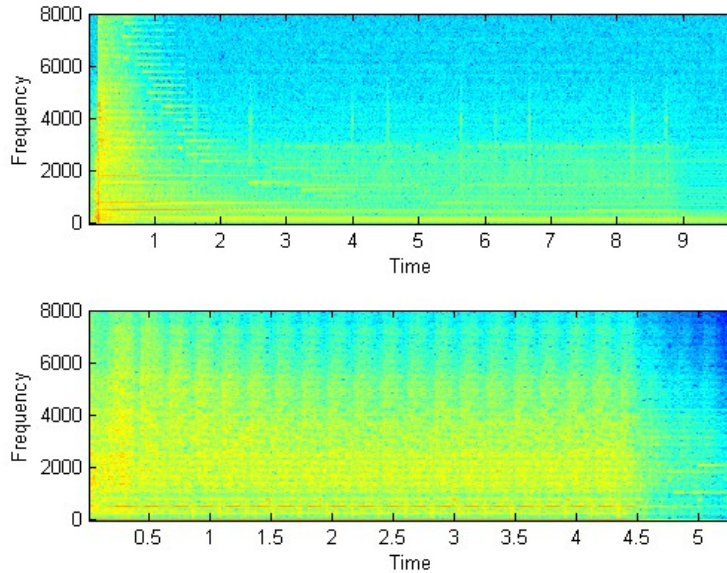
σχήμα 4.9: μεταμόρφωση κιθάρας σε φλάουτο

στο σχήμα 4.9

4.3.7 Η διαχείριση του θορύβου

Αποφύγαμε προς χάριν απλότητας να αναφέρουμε πως πριν γίνει οποιαδήποτε επεξεργασία στις τιμές των παραμέτρων των ήχων A και B γίνεται απ' ευθείας μία ανασύνθεσή τους. Το αποτέλεσμα αυτής της ανασύνθεσης αντιπροσωπεύει το αρμονικό κομμάτι των ήχων αυτών και αν αφαιρεθεί από το αρχικό σήμα αφήνει σαν υπόλοιπο την πληροφορία που ονομάζουμε θόρυβο και με την οποία περιγράφουμε όλες τις μη αρμονικές συνιστώσες του ήχου που μπορεί να έχουν να κάνουν με τον ίδιο το χαρακτήρα του οργάνου ή με τον περιβάλλοντα χώρο μέσα στον οποίο αντηχεί.

Μπορούμε να συνοψίσουμε τα παραπάνω με τις εξής σχέσεις:



Σχήμα 4.10: ο θόρυβος ηλεκτρικής κιθάρας και φλάουτου

$$Noise_A(t) = A(t) - A_{synthetic}(t)$$

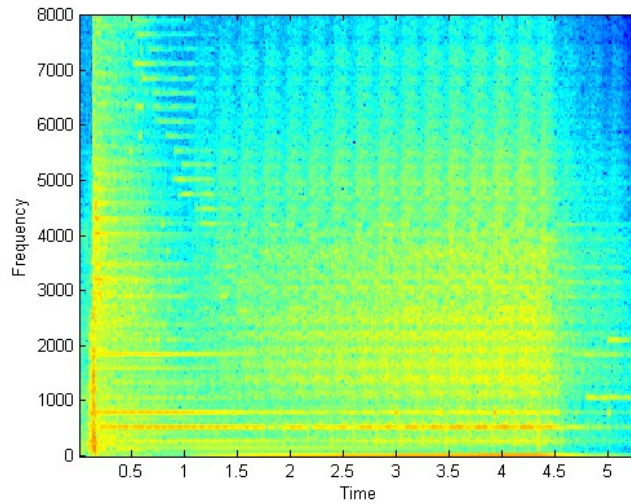
και

$$Noise_B(t) = B - B_{synthetic}(t)$$

Το αποτέλεσμα αυτής της διεργασίας για τους ήχους αυτούς μπορούμε να τους δούμε στην εικόνα 4.10.

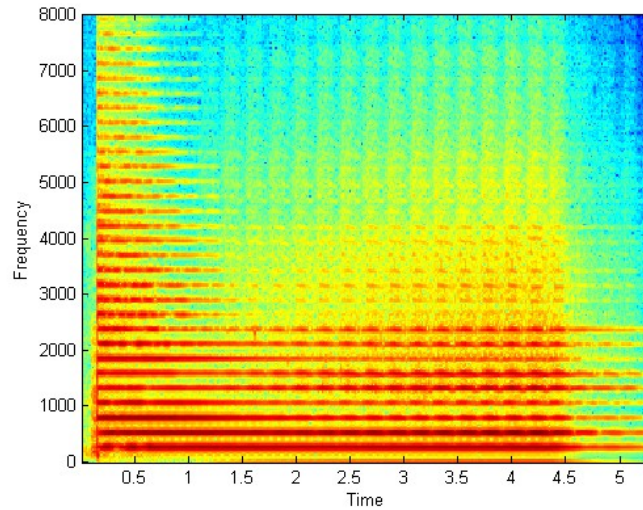
Στη συνέχεια κάνουμε ένα απλό crossfade ανάμεσα στα δύο αυτά σήματα και δημιουργούμε τη συνιστώσα του θορύβου για το τελικό σήμα όπως φαίνεται στην εικόνα 4.11 και για το διάστημα μεταμόρφωσης περιγράφεται από τον τύπο:

$$noise(t) = k(t)noise_A(t) + [1 - k(t)]noise_B(t)$$



σχήμα 4.11: ο θόρυβος της μεταμόρφωσης

Προσθέτοντας τον θόρυβο στο συνθετικό μεταμορφωμένο σήμα όπως το περιγράψαμε στην προηγούμενη παράγραφο λαμβάνουμε και το τελικό μεταμορφωμένο μας σήμα όπως το βλέπουμε και στο σχήμα 4.12

$$morph(t) = morph_{harmonic}(t) + noise(t)$$


σχήμα 4.12: μεταμόρφωση ηλ.κιθάρας σε φλάουτο

5. ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ATHENA

Στο κεφάλαιο αυτό εξηγούμε τις παραμέτρους που λάβαμε υπ' όψιν κατά την αξιολόγηση του αλγορίθμου Athena και παρουσιάζουμε τα αποτελέσματα της αναλύοντας τις σχετικές επιτυχίες ή αποτυχίες της υλοποίησής μας.

5.1 Ο σκοπός της αξιολόγησης

Από την αρχή της εργασίας έχει τεθεί σαν στόχος η δημιουργία ενός αλγορίθμου κατάλληλου για τη μεταμόρφωση μη αρμονικών σημάτων. Πρέπει να γίνει λοιπόν ένας έλεγχος των αποτελεσμάτων μας πλάι στον αλγόριθμο που χρησιμοποιείται στην πλειοψηφία των εμπορικών εφαρμογών, δηλαδή το κοινό crossfade. Ο έλεγχος πρέπει να περιλαμβάνει και ποιοτικά κριτήρια για την πιστότητα του ήχου σε σχέση με τα αρχικά σήματα μιας και ο εύκολος και απλός αλγόριθμος του crossfade παρά τα (όπως αναμένουμε) χειρότερα αποτελέσματα στην ίδια τη μεταμόρφωση, εντούτοις έχει το χαρακτηριστικό ότι κρατάει τα αρχικά σήματα ανέπαφα. Επίσης κρίνεται απαραίτητη μία αισθητική αξιολόγηση του παραγόμενου αποτελέσματος αλλά και μία αξιολόγηση της ιδιαίτερης μεταχείρισης που επιφυλάξαμε για τα πλάτη των καναλιών που δεν προκαλούν δυσαρμονία. Τέλος, θα πρέπει η αξιολόγηση να γίνει σε ένα ευρύ φάσμα ήχων με διαφορετικά χαρακτηριστικά που θα μπορέσουν να αποκαλύψουν τα δυνατά σημεία και τις αδυναμίες του αλγορίθμου μας. Στην επόμενη παράγραφο περιγράφεται η λογική του τεστ που δημιουργήσαμε για να καλύψουμε τις παραπάνω ανάγκες

5.2 Το τεστ αξιολόγησης του αλγορίθμου

Για την κάλυψη των παραπάνω στόχων χρησιμοποιήσαμε 13 εφαρμογές του αλγορίθμου σε 13 σετ των δύο ήχων. Πήραμε τρεις γενικές κατηγορίες ήχων.

Στην πρώτη κατηγορία θεωρήσαμε ήχους A και B που είναι “αρμονικοί” μεταξύ τους. Πρόκειται για ήχους που τα διαστήματα που ορίζονται από την κλασική αρμονία ως

αρμονικά και πιο συγκεκριμένα διαστήματα 3ης μικρής, 4ης καθαρής και ταυτοφωνία. Επίσης δοκιμάσαμε μεταμορφώσεις ανάμεσα σε αναστροφές μίας συγχορδίας.

Στη δεύτερη κατηγορία στην οποία επικεντρώνεται κυρίως το ενδιαφέρον μας διαλέξαμε διάφωνα διαστήματα όπως δεύτερη μικρή ή μεγάλη 7η. Δοκιμάσαμε επίσης και μεταμορφώσεις ανάμεσα σε συγχορδίες που ανήκουν σε διαφορετικές κλίμακες αλλά και μεταμορφώσεις συγχορδιών σε διάφωνες μοναδικές νότες.

Στην τρίτη κατηγορία δοκιμάσαμε μεταμορφώσεις ανάμεσα σε σήματα μουσικής και φωνής, μία κατηγορία που σαφώς δεν εμπίπτει στον ορισμό του προβλήματός μας όμως παρουσίασε ενδιαφέροντα αποτελέσματα.

Για κάθε σετ ήχων Α και Β ζητήσαμε από τους ακροατές να ακούσουν τρεις διαφορετικές εκδοχές τις μεταμόρφωσης. Η μία ήταν ένα απλό crossfade η δεύτερη μία μεταμόρφωση με τον αλγόριθμο Athena και η τρίτη μία μεταμόρφωση με τον αλγόριθμο Αθηνά αλλά με τα πλάτη των μη δυσαρμονικών σημάτων ζυγισμένα από όρους υψωμένους σε κάποια δύναμη (όπως έχει περιγραφεί στην παράγραφο 4.5), αριθμού που προέκυψε από πειραματισμό με τα υπό εξέταση σήματα.

Οι ακροατές κρίθηκαν να επιλέξουν ποια από τις τρεις εκδοχές πληρούσε καλύτερα κάθε ένα από τα εξής τρία κριτήρια.

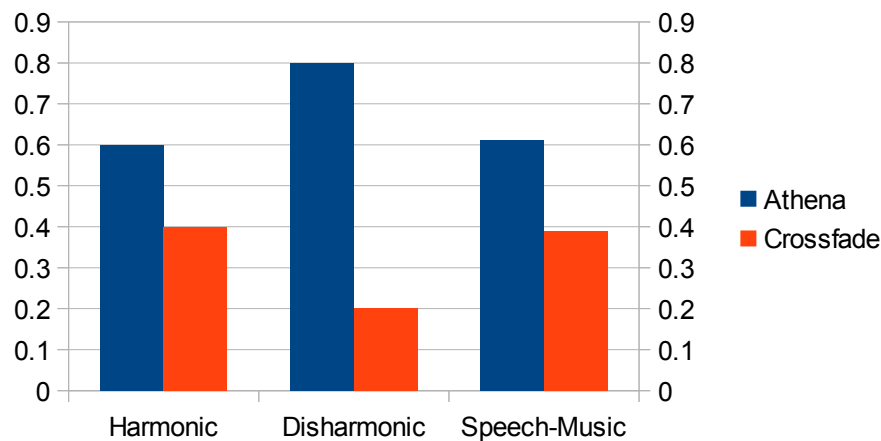
- Ποιος ήχος ακολουθεί πιστότερα και με καλύτερη ποιότητα τα αρχικά δείγματα;
- Σε ποιον ήχο είναι πιο ομαλή και ευχάριστη η μεταμόρφωση;
- Ποιος ήχος σας ικανοποιεί περισσότερο αισθητικά;

Τα δείγματα και οι διαφορετικές κατηγορίες παίχτηκαν στους ακροατές με τυχαία σειρά. Το δείγμα μας ήταν 11 ακροατές.

5.3 Τα αποτελέσματα της αξιολόγησης.

5.3.1 Η πιο ομαλή μετάβαση

Όπως έχει διαφανεί μέχρι εδώ το σημαντικότερο αποτέλεσμα που αναμένουμε και έχει μεγαλύτερη σημασία για τον αλγόριθμο είναι για τους δυσαρμονικούς ήχους η ερώτηση “Σε ποιον ήχο είναι πιο ομαλή και ευχάριστη η μεταμόρφωση;”.



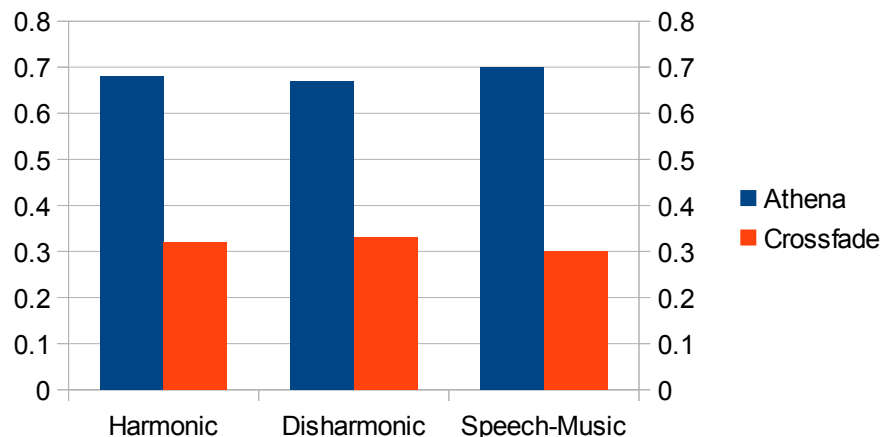
Σχήμα 5.1: πιο ομαλή και ποιοτική μεταμόρφωση

Όπως φαίνεται στο σχήμα 5.1 το 80% των ακροατών επέλεξε τα αποτελέσματα του αλγορίθμου Athena για το ερώτημα αυτό. Για τους αρμονικούς ήχους και για τα δείγματα φωνής έχουμε ένα παράπλευρο κέρδος. Με ποσοστά 60% και 61% αντίστοιχα οι ακροατές προτιμούν τον Athena σε σχέση με το απλό crossfade. Και στις δύο περιπτώσεις αυτή η οριακή διαφορά είναι εξηγήσιμη αν και ανεπαίσθητη. Στην περίπτωση των αρμονικών ήχων οι αρμονικές που επιλέγονται από τον αλγόριθμο για τη μεταμόρφωση είναι οριακά ίσες μεταξύ τους, δεν είναι δηλαδή δυσαρμονικές αλλά απόλυτα συμβατές. Όμως ξανά οι μικροδιαφορές ανάμεσα στα χαρακτηριστικά των οργάνων εξομαλύνονται σε σχέση με το απλό crossfade, εξ ου και η καλύτερη επίδοση του αλγορίθμου μας. Οι περιπτώσεις φωνής βρίσκονται ακριβώς στον αντίποδα της περίπτωσης των αρμονικών ήχων. Ο χαρακτήρας δίνεται από αυτό που έχουμε δεχθεί ως θόρυβο αλλά η μικρή εξομάλυνση που εφαρμόζει ο

αλγόριθμός μας φαίνεται να αναγνωρίζεται από τους χρήστες.

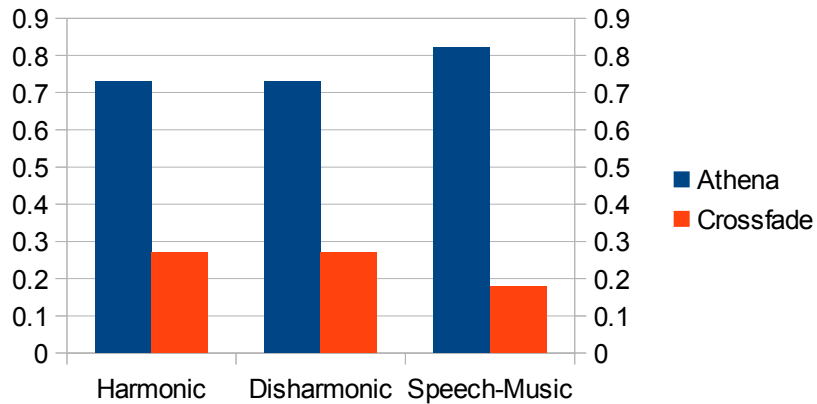
Ανάμεσα στις δύο εκδοχές των αλγορίθμων μας τα αποτελέσματα ποικίλλουν ανά περίπτωση. Στα δείγματα μεταμόρφωσης φλάουτου (Σι) σε κιθάρα (Ντο), δύο σήματα που θεωρούμε δυσαρμονικά, ο αλγόριθμος Athena με τα ζυγισμένα βάρη συγκεντρώνει το 73% των προτιμήσεων σε σχέση με τις άλλες δύο εκδοχές. Αντίθετα στη μεταμόρφωση δίφωνης συγχορδίας πιάνου (Ντο-Μι) σε διάφωνη συγχορδία πιάνου (Ντο#-Φα) οι χρήστες δείχνουν να προτιμούν με ποσοστό 64% τη μη ζυγισμένη εκδοχή του Athena ενώ μοιράζουν ίσα το υπόλοιπο ανάμεσα στη ζυγισμένη εκδοχή και το crossfade.

5.3.2 Αισθητική ικανοποίηση και ποιότητα ήχου



σχήμα 5.2: ο πιο αισθητικά ικανοποιητικός αλγόριθμος

Το ερώτημα για τη μεγαλύτερη αισθητική ικανοποίηση ήταν κατά βάση υποκειμενικό και τα αποτελέσματά του επηρεάζονται από παράγοντες που σε κάποιες περιπτώσεις δεν αφορούν άμεσα τους στόχους που έχουν τεθεί για την εργασία. Παρ' όλα αυτά η συνάρτηση των απαντήσεων με το είδος των υπό εξέταση ήχων μπορεί να φανεί χρήσιμη σε πιθανούς τρίτους χρήστες του αλγορίθμου. Όπως φαίνεται και στο σχήμα 5.2 πάνω από 2/3 (69%) χρήστες προτιμούν αισθητικά τα αποτελέσματα του αλγορίθμου μας.



σχήμα 5.3: καλύτερη πιστότητα και ποιότητα ήχου

Στο ερώτημα για την καλύτερη ποιότητα και πιστότητα του ήχου μας περίμενε μία έκπληξη. Δεδομένου ότι για τη μεταμόρφωση χρησιμοποιούμε μοντέλα των ήχων και όχι τους ήχους αυτούς καθαυτούς θα περιμέναμε ότι το crossfade που δεν επεμβαίνει στους αρχικούς ήχους θα είχε μεγαλύτερη απήχηση στους ακροατές. Παρ' όλα αυτά είναι εξηγήσιμο αποτέλεσμα αν λάβουμε υπ' όψιν πως η δυσαρμονία παράγει διακρότημα που είναι μία από τις πιο σημαντικές αιτίες της δυσφορίας στο ανθρώπινο αυτί. Συνολικά οι ακροατές επέλεξαν τα αποτελέσματα του αλγορίθμου μας στο 76% των περιπτώσεων.

6. Η ΒΙΒΛΙΟΘΗΚΗ ΕΦΕ ΤΟΥ ΑΘΗΝΑ

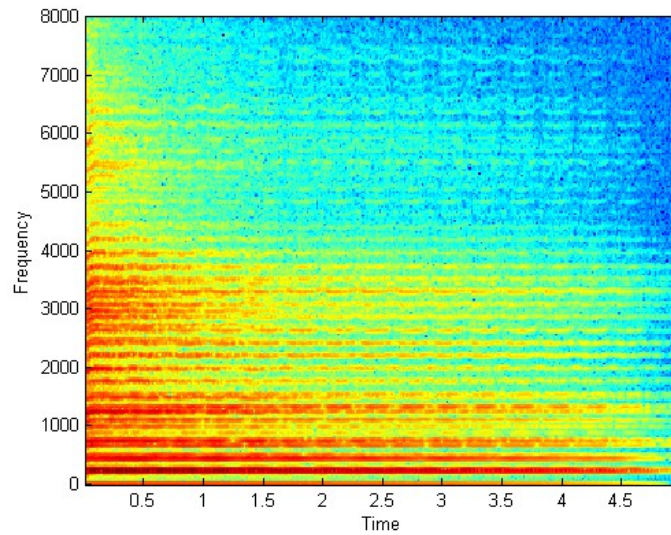
Σε αυτό το κεφάλαιο παρουσιάζονται με συντομία τα υπόλοιπα εφέ ήχου[25] που δημιουργήθηκαν με τη χρήση των συμπερασμάτων που προέκυψαν από την υλοποίηση του αλγορίθμου Athena αλλά και της δύναμης επεξεργασίας ήχου που μας προσφέρει το ημιτονοειδές μοντέλο.

6.1 Ο αλγόριθμος Nobeating

Όπως έχουμε αναφέρει και στην παράγραφο 2.3 όταν δύο ήχοι συνηχούν τότε συγκεκριμένες από τις αρμονικές τους που βρίσκονται μέσα σε κάποιο εξαρτώμενο από τις συχνότητες αυτές εύρος φάσματος δημιουργούν δυσαρμονία. Στόχος του αλγορίθμου Nobeating είναι να εξομαλύνει τις αρμονικές αυτές για δυσαρμονικούς ήχους και να τους επιτρέψει να ηχήσουν συγχρόνως εξαφανίζοντας όμως την αίσθηση ενόχλησης στο ανθρώπινο αυτί. Είναι μία παραλλαγή του αλγορίθμου Athena με τις εξής διαφοροποιήσεις:

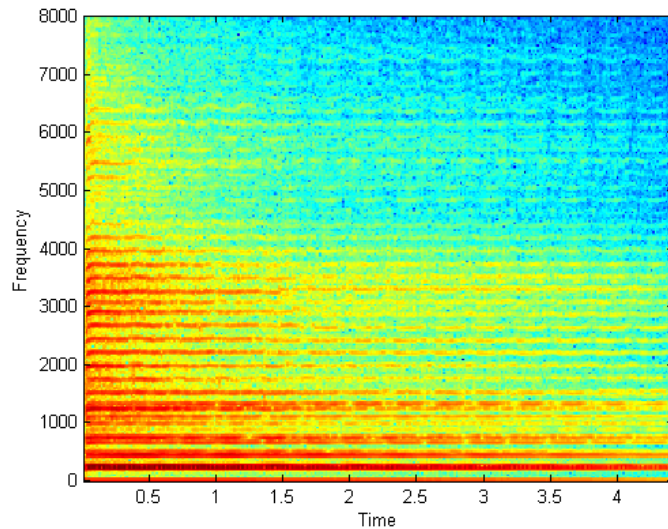
- Ο παράγοντας k που στον αλγόριθμο Athena επέτρεπε τη σταδιακή μεταμόρφωση του ήχου A στον ήχο B πλέον είναι μία σταθερά ίση με 0.5 ώστε να δημιουργεί σε όλη τη διάρκεια του χρόνου ένα μέσο όρο των τιμών των δύο σημάτων(φυσικά μόνο για τα κανάλια που δημιουργούν δυσαρμονία)
- Τα κανάλια που δεν δημιουργούν δυσαρμονία ουσιαστικά αφήνονται ανέπαφα

Στα σχήματα 6.1 και 6.2 μπορούμε να δούμε πώς η συνήχηση δύο διάφωνων νοτών από σαξόφωνο (Λ α και Σ ι) εξομαλύνεται με τη χρήση του αλγορίθμου Nobeating.



σχήμα 6.1: μία απλή μίξη των δύο νοτών

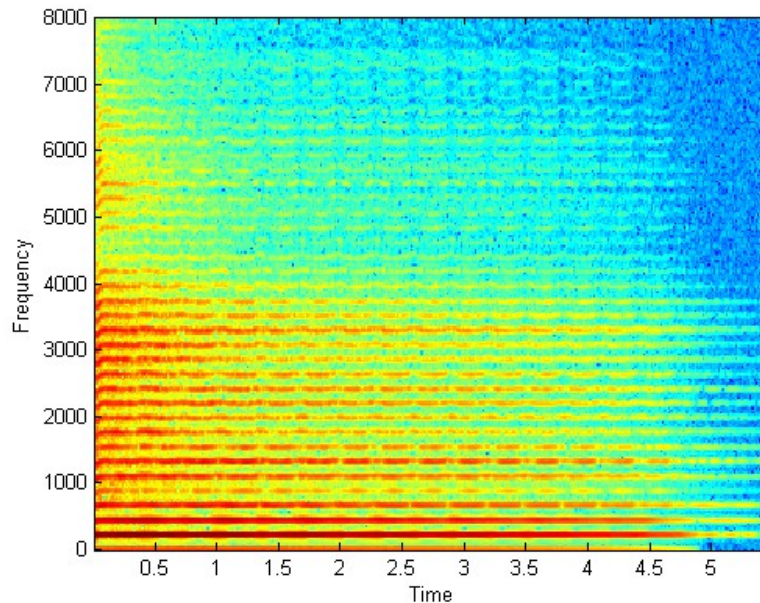
Πρέπει να σημειωθεί και μία ακόμα διαφοροποίηση. Εδώ οι ήχοι μοντελοποιούνται ξεχωριστά και εξάγεται η πληροφορία θορύβου. Τα δύο σήματα θορύβου που προκύπτουν ως αποτέλεσμα προστίθενται ξεχωριστά στο εξομαλυμένο σήμα της διαδικασίας του No beating. Ο αλγόριθμος έχει μία επιπλέον παράμετρο που καθορίζει τη διάρκεια που θα ηχήσει το τελικό σήμα.



σχήμα 6.2: οι δυσαρμονίες έχουν εξομαλυνθεί

6.2 Ο αλγόριθμος Pitchshifter

Σκοπός των pitchshifting αλγορίθμων είναι να μετακινήσουν ένα σήμα σε πιο ψηλές ή χαμηλές μουσικές νότες.



σχήμα 6.3: Λα από σαξόφωνο

Εδώ έχουμε να κάνουμε με ένα μοναδικό σήμα. Για την παρουσίαση του Pitchshifter αλλά και των υπόλοιπων αλγορίθμων θα χρησιμοποιήσουμε το σήμα Λα σαξόφωνου της εικόνας 6.3.

Η λογική που ακολουθείται στον αλγόριθμο Pitchshifter είναι πως παρεμβαίνουμε στο αρμονικό περιεχόμενο των καναλιών που προκύπτουν από την εφαρμογή του ημιτονοειδούς μοντέλου. Πριν οποιαδήποτε παρέμβαση όμως φροντίζουμε να εξάγουμε το αρμονικό περιεχόμενο του σήματος και να το αφαιρέσουμε από το αρχικό σήμα ώστε να εξάγουμε την πληροφορία για το θόρυβο. Στη συνέχεια επεξεργαζόμαστε τις συχνότητες των επιμέρους καναλιών με τον εξής τύπο.

$$f' = f * 2^{\left(\frac{\text{semitoneoffset}}{12}\right)}$$

Η τιμή semitoneoffset δίνεται σαν παράμετρος από τον

χρήστη στον αλγόριθμο και ορίζει τον αριθμό των ημιτονίων που πρέπει να μετακινηθεί το σήμα μας. Μπορεί να δεχτεί και αρνητικές τιμές ώστε να κινήσει το αρχικό σήμα προς χαμηλότερες νότες.

Σκοπός μας είναι να πολλαπλασιάσουμε κάθε κανάλι με το κατάλληλο μέγεθος και να το μετατοπίσουμε με τέτοιο τρόπο ώστε να αντιστοιχεί στα διαστήματα του συγκεκριμένου μουσικού συστήματος.

Στη συνέχεια ανασυνθέτουμε το σήμα με τις επεξεργασμένες συχνότητες και προσθέτουμε τον αρχικό θόρυβο που υπολογίσαμε στο προηγούμενο βήμα. Το αποτέλεσμα είναι η αρχική μας νότα μετακινημένη κατά τα ορισμένα ημιτόνια.

6.3 Ο αλγόριθμος SinusoidalOctaver

Σκοπός των αλγορίθμων Octaver είναι η σύνθεση μίας νότας με μία οκτάβα (ψηλότερα ή χαμηλότερα) απόσταση από την αρχική και η συνήχηση των δύο. Ο αλγόριθμος SinusoidalOctaver είναι μία υλοποίηση της παραπάνω αρχής. Εδώ το σήμα αναλύεται σε κανάλια χωρίς όμως να εξάγουμε τον θόρυβο αφού θα χρησιμοποιήσουμε και το αυθεντικό σήμα. Στη συνέχεια ελέγχεται η παράμετρος `upordown` που καθορίζεται από την επιλογή του χρήστη για την ψηλότερη ή χαμηλότερη οκτάβα.

Για την ψηλότερη οκτάβα κάθε κανάλι υφίσταται επεξεργασία σύμφωνα με τη σχέση

$$f' = f * 2$$

και για τη χαμηλότερη με την

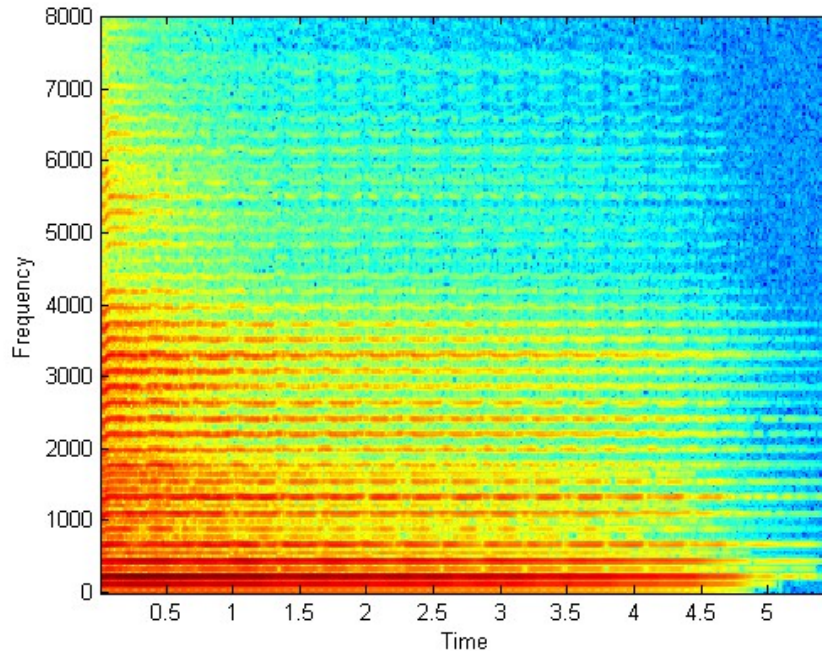
$$f' = f / 2$$

Στη συνέχεια γίνεται ανασύνθεση του σήματος με τη χρήση των επεξεργασμένων τιμών. Η μίξη του αρχικού με το συνθετικό μετατοπισμένο σήμα γίνεται με βάση τη σχέση

$$x_{out}(t) = x(t) * (1 - mix) + x_{octaver}(t) * mix$$

όπου το `mix` είναι μία καθορισμένη από το χρήστη τιμή που παίρνει τιμές από το 0 μέχρι το 1. Για `mix=0` ακούμε μόνο το αρχικό σήμα ενώ για `mix=1` ακούμε μόνο το

μετατοπισμένο κατά μία οκτάβα σήμα. Στο σχήμα 6.4 βλέπουμε την αρχική μας νότα να συνηχεί με μία κατά μία οκτάβα χαμηλότερη $\Lambda\alpha$ στα 220Hz, με την τιμή του mix ορισμένη στο 0.6



σχήμα 6.4: το αποτέλεσμα του *SinusoidalOctaver* για την χαμηλή οκτάβα της $\Lambda\alpha$

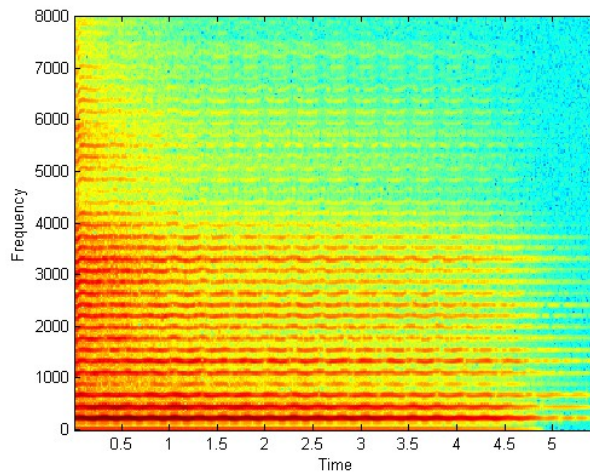
6.4 Ο αλγόριθμος FMvibrato

Ο αλγόριθμος FMvibrato είναι ένας πρωτότυπος αλγόριθμος που επεμβαίνει ιδιαίτερα δραστικά στο αρχικό σήμα με σκοπό να δημιουργήσει την αίσθηση του τονικού vibrato επεξεργαζόμενος όμως άμεσα το σύνολο των αρμονικών του σήματος. Δέχεται σαν είσοδο ένα σήμα αλλά και δύο παραμέτρους που καθορίζονται από το χρήστη. Την vibrato frequency που καθορίζει τη συχνότητα του βιμπράτο και την vibrato width που καθορίζει το πλάτος της αρμονικής ταλάντωσης των επιμέρους καναλιών.

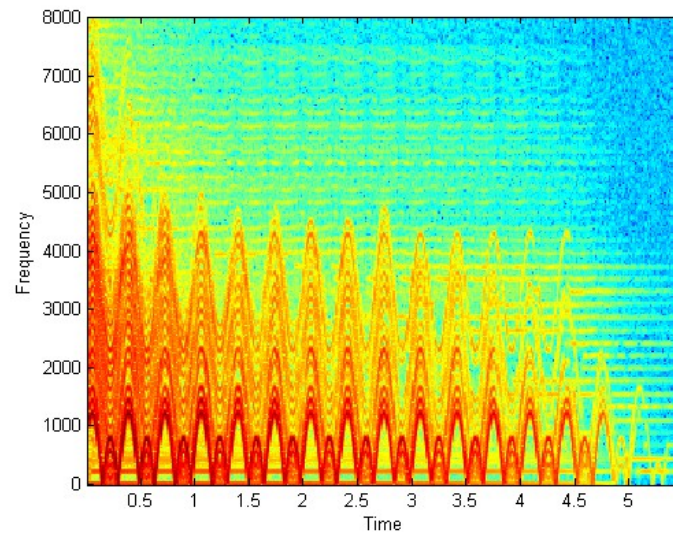
Το πρώτο βήμα του αλγορίθμου είναι η εξαγωγή των καναλιών που συναποτελούν το αρμονικό μέρος του αρχικού ήχου και ο υπολογισμός του θορύβου. Στη συνέχεια η συχνότητα του κάθε καναλιού διαμορφώνεται από ένα ημίτονο με τον εξής τρόπο:

$$f' = f + \text{vibratowidth} * \sin(2\pi * \text{vibratofrequency} * t)$$

Το σήμα ανασυντίθεται με βάση τις επεξεργασμένες τιμές και προστίθεται στον ήδη υπολογισμένο θόρυβο.



Σχήμα 6.5: βιμπράτο 4Hz σε νότα σαζόφωνου. Οι αρμονικές ταλαντώνονται κατά πλάτος 10Hz



σχήμα 6.6: μία ακραία τιμή ταλάντωσης στα 100Hz έχει ακραία αποτελέσματα...

7. ΣΥΜΠΕΡΑΣΜΑΤΑ

Συγκρίνοντας τα αποτελέσματα της εργασίας με τους στόχους που τέθηκαν αρχικά μπορούμε να πούμε ότι έχουν καλυφθεί σε μεγάλο βαθμό αφού για το συγκεκριμένο σετ προβλημάτων που ετέθη τα αποτελέσματα του αλγορίθμου είχαν σημαντική απήχηση στους χρήστες στους οποίους δειγματίστηκαν όπως δείξαμε στο κεφάλαιο 4. Η απήχηση δεν αφορούσε μόνο την ομαλότητα της μεταμόρφωσης που ήταν και ο κυρίως στόχος αλλά και την ποιότητα του ήχου και την υποκειμενική αισθητική ικανοποίηση.

Την ίδια στιγμή όμως τα αποτελέσματα των πειραμάτων στις επιμέρους περιπτώσεις δείχνουν πως η εφαρμοζόμενη μέθοδος εξαρτάται σε μεγάλο βαθμό από τους ίδιους τους ήχους και πως δεν μπορεί να οριστεί ένας γενικός κανόνας μεταμόρφωσης ήχου. Ο χώρος στον οποίο έχουν κινηθεί άλλες εργασίες [26] με διαφορετικό σετ προβλημάτων είναι η παραμετροποίηση και η παρέμβαση σε ψυχοακουστικούς παράγοντες και ο αυτοματοποιημένος ορισμός του διαστήματος της μεταμόρφωσης [27]. Άλλες εργασίες έχουν κινηθεί στο πεδίο του χρόνου με ανάστροφη λογική, δηλαδή την κλιμάκωση ή αποκλιμάκωση στο χρόνο των αρχικών δειγμάτων ώστε να συμπέσουν με τα οριζόμενα από το χρήστη όρια της μεταμόρφωσης αλλά και τα ιδιαίτερα χρονικά χαρακτηριστικά που επιβάλλουν τα δείγματα. Η προσθήκη τέτοιων παραμέτρων θα εμπλούτιζε τον αλγόριθμο Athena.

Ένας δεύτερος προβληματισμός που προέκυψε είναι πως η χρήση των εργαλείων μοντελοποίησης με σταθερές παραμέτρους (όπως άλλωστε θα απαιτούσε και ένα χρηστικό εφέ) κάποιες φορές απέτυχε να αναπαράγει με πιστότητα τους αρχικούς ήχους και μάλιστα υπάρχουν και περιπτώσεις ήχων που ήταν αδύνατο να μοντελοποιηθούν πιστά.

Η μεταμόρφωση ήχου είναι μία ιδέα που αριθμεί περίπου 20 χρόνια ζωής όμως ελάχιστες υλοποιήσεις είναι διαθέσιμες στον ενδιαφερόμενο χρήστη. Ελπίζουμε πως ο αλγόριθμος Αθηνά εκτός από μία ιδιαίτερη εκδοχή των βασικών ιδεών της μεταμόρφωσης θα προσφέρει και ένα απλό και

εύχρηστο εργαλείο για μία πιο ενδεδειγμένη ανάλυση των στοιχείων συμφωνίας ή διαφωνίας ανάμεσα σε δύο ήχους. Παρ' ότι η υλοποίηση έχει μία συγκεκριμένη αντιμετώπιση στα θέματα που ανακύπτουν ο τρόπος ανάπτυξης του αλγορίθμου είναι τέτοιος που πολύ δραστικές αλλαγές μπορούν να γίνουν με μηδαμινές αλλαγές στον κώδικα. Ένα πρώτο τέτοιο παράδειγμα είναι πως με την εξάλειψη του ελέγχου σύμπτωσης δύο καναλιών μέσα στο διάστημα του ERB ο Athena μετατρέπεται σε έναν αλγόριθμο μεταμόρφωσης όλου του φάσματος όπου κάθε κανάλι του ήχου A ταιριάζει με το πιο κοντινό κανάλι του B και έχουμε στα χέρια μας ένα πολύ δραστικό εφέ.

Πρέπει να αναφερθεί πως όπως φαίνεται και στο κεφάλαιο 5 η δύναμη που κρύβει το ημιτονοειδές μοντέλο μπορεί με απλό και σύντομο κώδικα να πραγματοποιήσει ψηφιακά εφέ με απεριόριστες δυνατότητες στην επεξεργασία του ήχου και μάλιστα χωρίς να χάνεται η ποιότητα και το χαρακτηριστικό ηχόχρωμα των υπό επεξεργασία ήχων.

7.1 Μελλοντική δουλειά

Όπως αβίαστα προκύπτει και από τα συμπεράσματα υπάρχει χώρος για διάφορες επεκτάσεις του αλγορίθμου Athena. Πριν τις αναφέρουμε όμως είναι σημαντικό να τονίσουμε πως η προσθήκη της λογικής του αλγορίθμου σε άλλες υλοποιήσεις μεταμόρφωσης θα είχε εξίσου ενδιαφέροντα αποτελέσματα. Η παράμετρος της δυσαρμονίας δεν αναφέρεται στη βιβλιογραφία και δεν έχει ληφθεί υπ' όψιν σε άλλες υλοποιήσεις. Το παράδειγμα του Loris που είναι η πιο συγγενής υλοποίηση αφού χρησιμοποιεί το μοντέλο MQ παντρεύει όλα ανεξαιρέτως τα κανάλια χωρίς διαχωρισμό τους.

Στις πιθανές προσθήκες πιο σημαντική όπως αναφέρθηκε παραπάνω θα ήταν η αξιοποίηση ψυχοακουστικών παραμέτρων. Όπως φάνηκε και από τα διαφορετικά πειράματα και την επικοινωνία με τους ακροατές υπάρχουν σημαντικά μέρη αυτού που ονομάζουμε ηχόχρωμα που δεν μπορούν να καθοριστούν μονοδιάστατα με τιμές συχνότητας ή πλάτους.

Ένας ενδιαφέρων χώρος επέκτασης με σημαντικό όμως κόστος σε χρόνο επεξεργασίας είναι η αυτοματοποιημένη

προσαρμογή του μοντέλου στον υπό επεξεργασία ήχο. Όπως έχει ήδη αναφερθεί υπήρξαν πολλές περιπτώσεις που μία καλύτερη μοντελοποίηση προσφέρει καλύτερα αποτελέσματα και στη μεταμόρφωση ενώ η άριστη ποιότητα του ήχου αποτελεί απαραίτητη απαίτηση για μουσικές εφαρμογές.

Αυτή είναι και η κύρια κατεύθυνση επέκτασης που υπαγορεύει η υλοποίηση και το σετ προβλημάτων που επιλέξαμε. Απαραίτητο βήμα είναι η προσαρμογή του αλγορίθμου για μεταμορφώσεις πραγματικού χρόνου (real time), προσαρμογή που θα τον καθιστούσε κατάλληλο για ενσωμάτωση σε ψηφιακά μουσικά όργανα. Για να επιτευχθεί κάτι τέτοιο πρέπει να αναπτυχθεί ένας αλγόριθμος εύρεσης των καναλιών με τη δυνατότητα να προβλέπει την εξέλιξη ενός καναλιού. Η μέχρι ώρας αδυναμία πρόβλεψης των καναλιών είναι το μόνο στοιχείο του αθροίσματος αλγορίθμων που συναποτελούν τον Athena που δεν μπορεί άμεσα να μεταφερθεί σε υλοποίηση πραγματικού χρόνου. Ένα πρόβλημα του οποίου η λύση είναι απλούστερη είναι ο υπολογισμός των μέσων όρων συχνότητας για κάθε κανάλι, κάτι που όμως μπορεί να αντικατασταθεί από μία υλοποίηση που ανιχνεύει την κίνηση των καναλιών -και άρα τη δυσαρμονία- κατά την εξέλιξή της και όχι εκ των υστέρων. Μία τέτοια μετατροπή θα μπορούσε να επικοινωνήσει και με τον απώτερο στόχο αυτής της μελέτης που είναι η δημιουργία ενός “συνεχούς” ανάμεσα σε ήχους διαφορετικών οργάνων και τόνων που θα έδινε ενδιαφέροντα αποτελέσματα σε μία υλοποίηση σε ψηφιακό μουσικό όργανο.

Μία άλλη ιδέα είναι η συνάρτηση των περιβαλλουσών πλάτους με ειδικά βάρη ανάλογα με τη δυσαρμονία που πρόκειται να προκαλέσουν. Βέβαια σημαντικές τροποποιήσεις μπορούν να γίνουν επιδιώκοντας μη συμβατικά αποτελέσματα όπως για παράδειγμα το να τονιστούν οι δυσαρμονίες.

ΠΑΡΑΡΤΗΜΑ: ΟΙ ΚΩΔΙΚΕΣ MATLAB

Ο αλγόριθμος Athena.

```
function [morph]=
athena(d1,d2,sr,morph_start,morph_length,amp_power)
%      d1 and d2 are the two sounds to be morphed
%      sr is the sample rate of the initial samples(it has
to be the same)
%      morph_start is the time the morphing has to start
in seconds
%      morph_length is the time the morphing has to last
in seconds
%      amp_power is a factor by which the amplitudes of
the non-beating
%      channels are compressed
%      this ccode has been written for the dissertation
thesis of Michael
%      Kotsirakis at the Technical University of Crete

per=128; %windowing period
win=256; %window size

%STFT and instantaneous frequency
[I1,S1]=ifgram(d1,win,win,per,sr);
[I2,S2]=ifgram(d2,win,win,per,sr);

%well... extract tracks
[R1,M1]=extractrax(abs(S1));
[R2,M2]=extractrax(abs(S2));

F1 = colinterpvals(R1,I1);
F2 = colinterpvals(R2,I2);
% Interpolate the (columnwise unwrapped) STFT phases to get
exact peak
% phases for every sample point (synthesis phase is
negative of analysis)
P1 = -colinterpvals(R1,unwrap(angle(S1)));
P2 = -colinterpvals(R2,unwrap(angle(S2)));

%just a matter of names
AngleS1=P1;
AngleS2=P2;

%preparing the seperate sounds for synthesis
F1noise = [0*F1(:,1),F1,0*F1(:,end)];
M1noise = [0*M1(:,1),M1,0*M1(:,end)];
P1noise = [0*P1(:,1),P1,0*P1(:,end)];

F2noise = [0*F2(:,1),F2,0*F2(:,end)];
```

```

M2noise = [0*M2(:,1),M2,0*M2(:,end)];
P2noise = [0*P2(:,1),P2,0*P2(:,end)];

synth1 = synthphtrax(F1noise,M1noise,P1noise,sr,win,per);
synth2 = synthphtrax(F2noise,M2noise,P2noise,sr,win,per);

%we extract the noise of each sound
noise1=d1-synth1(1:length(d1))';
noise2=d2-synth2(1:length(d2))';

%here we calculate the morphing times in samples and in
windows
morph_startNOISE=morph_start*sr;
morph_lengthNOISE=morph_length*sr;
morph_endNOISE=morph_startNOISE+morph_lengthNOISE;

morph_start=round(morph_start*sr/per);
morph_length=round(morph_length*sr/per);
morph_end=morph_start+morph_length;

%here we will calculate a mean for the frequencies of each
channel
%taking into account the whole signal, we will use it to
merge channels
%around the same frequency
F1mean=zeros(size(F1,1),1);

for i=1:size(F1,1)
    petitsum=0; %excuse our french :)
    for j=1:size(F1,2)
        if ~isnan(F1(i,j))
            F1mean(i)=F1mean(i)+F1(i,j);
            petitsum=petitsum+1;
        end
    end
    F1mean(i)=F1mean(i)/petitsum;
end

F2mean=zeros(size(F2,1),1);
for i=1:size(F2,1)
    petitsum=0;
    for j=1:size(F2,2)
        if ~isnan(F2(i,j))
            F2mean(i)=F2mean(i)+F2(i,j);
            petitsum=petitsum+1;
        end
    end
    if ~petitsum==0
        F2mean(i)=F2mean(i)/petitsum;
    end
end

```

```

end
end

%merging tracks around the same frequency for d1
newpos=1;
lef=size(F1,1);

for i=1:lef
    for j=i+1:lef
        if abs(F1mean(i)-F1mean(j))<20
            for n=1:size(F1,2)
                if isnan(F1(i,n)) && isnan(F1(j,n))
                    F1(i,n)=nan;
                    M1(i,n)=nan;
                    AngleS1(i,n)=nan;

                    elseif isnan(F1(i,n)) && ~isnan(F1(j,n))
                        F1(i,n)= F1(j,n);
                        M1(i,n)= M1(j,n);
                        AngleS1(i,n)= AngleS1(j,n);

                    elseif ~isnan(F1(i,n)) && ~isnan(F1(j,n))
                        F1(i,n)= F1(i,n);
                        M1(i,n)= M1(i,n);
                        AngleS1(i,n)= AngleS1(i,n);
                    end

                    F1(j,n)=nan;

                end

            end

        end

    end

end

end
end

%clear up unused channels
i=1;
while i<lef
    if sum(isnan(F1(i,:)))== size(F1,2)
        F1(i,:)=[];
        M1(i,:)=[];
        AngleS1(i,:)=[];
        lef=lef-1;
        i=i-1;
    end
    i=i+1;
end

%merging tracks around the same frequency for d2

lef=size(F2,1);
fin=lef;

```

```

for i=1:lef
    for j=i+1:lef
        if abs(F2mean(i)-F2mean(j))<20
            for n=1:size(F2,2)
                if isnan(F2(i,n)) && isnan(F2(j,n))
                    F2(i,n)=nan;
                    M2(i,n)=nan;
                    AngleS2(i,n)=nan;

                    elseif isnan(F2(i,n)) && ~isnan(F2(j,n))
%&& abs(F2(j,n)-F2mean(i))
                    F2(i,n)= F2(j,n);
                    M2(i,n)= M2(j,n);
                    AngleS2(i,n)= AngleS2(j,n);

                    elseif ~isnan(F2(i,n)) && ~isnan(F2(j,n))
                    F2(i,n)= F2(i,n);
                    M2(i,n)= M2(i,n);
                    AngleS2(i,n)= AngleS2(i,n);

                end

                F2(j,n)=nan;

            end

        end

    end

end

%clear up unused channels
i=1;
while i<lef
    if sum(isnan(F2(i,:)))== size(F2,2)
        F2(i,:)=[];
        M2(i,:)=[];
        AngleS2(i,:)=[];
        lef=lef-1;
        i=i-1;
    end
    i=i+1;
end

%we calculate the mean frequency for each channel this time
taking into
%account only the morphing time
F1mean=zeros(size(F1,1),1);

for i=1:size(F1,1)
    petitsum=0;
    for j=morph_start:morph_end

```

```

        if ~isnan(F1(i,j))
            Flmean(i)=Flmean(i)+F1(i,j);
            petitsum=petitsum+1;
        end
    end
    Flmean(i)=Flmean(i)/petitsum;
end

F2mean=zeros(size(F2,1),1);
for i=1:size(F2,1)
    petitsum=0;
    for j=morph_start:morph_end

        if ~isnan(F2(i,j))
            F2mean(i)=F2mean(i)+F2(i,j);
            petitsum=petitsum+1;
        end
    end
    if ~petitsum==0
        F2mean(i)=F2mean(i)/petitsum;
    end
end

for i=1:size(F1,1)
    %channels whose difference is below this threshold are
    bound to create
    %beating according to the ERB model.
    mindist=24.7*(4.37*(Flmean(i)/1000) + 1)/2;
    flag=0;

    for j=1:size(F2,1)

        dista=abs(Flmean(i)-F2mean(j));

        if dista<mindist %we still want the closest match
        for each channel
            flag=1;%a match has been found, this channel
            will be morphed
            mindist=dista;
            closest=j;

        end
    end

    if flag==1 && Flmean(i)

        %we prepare for the morphing placing these channels
        face to face
        F1n(newpos,:)=F1(i,:);
        M1n(newpos,:)=M1(i,:);
        AngleS1n(newpos,:)=AngleS1(i,:);
        F2n(newpos,:)=F2(closest,:);
        M2n(newpos,:)=M2(closest,:);
        AngleS2n(newpos,:)=AngleS2(closest,:);
    end
end

```

```
        newpos=newpos+1;
        F1mean(i)=inf;
        F2mean(closest)=inf;

    end

end

nof_continuous=newpos-1 %this tells us how many morphing
channels were ultimately found

%we find which source sound has the most channels...
if size(F1,1)>size(F2,1)
    big_size=size(F1,1);
    small_size=size(F2,1);

else
    big_size=size(F2,1);
    small_size=size(F1,1);

end

%fill up the rest of the new tables with the old values
for i=1:small_size
    if F1mean(i)~=inf
        F1n(newpos,:)=F1(i,:);
        M1n(newpos,:)=M1(i,:);
        AngleS1n(newpos,:)=AngleS1(i,:);
        newpos=newpos+1;

    end

    if F2mean(i)~=inf
        F2n(newpos,:)=F2(i,:);
        M2n(newpos,:)=M2(i,:);
        AngleS2n(newpos,:)=AngleS2(i,:);
        newpos=newpos+1;

    end

end

for i=small_size+1:big_size
    if size(F1,1)>size(F2,1)
        if F1mean(i)~=inf
            F1n(newpos,:)=F1(i,:);
            M1n(newpos,:)=M1(i,:);
            AngleS1n(newpos,:)=AngleS1(i,:);
            newpos=newpos+1;
        end
    else
        if F2mean(i)~=inf
            F2n(newpos,:)=F2(i,:);
            M2n(newpos,:)=M2(i,:);
```

```

        AngleS2n(newpos, :)=AngleS2(i, :);
        newpos=newpos+1;
    end
end
end

%
if size(F1n,1)<size(F2n,1)
    F1n=[F1n; zeros(size(F2n,1)-size(F1n,1),size(F1n,2))];
    M1n=[M1n; zeros(size(M2n,1)-size(M1n,1),size(M1n,2))];
    AngleS1n=[AngleS1n; nan(size(AngleS2n,1)-
size(AngleS1n,1),size(AngleS1n,2))];
else
    F2n=[F2n; zeros(size(F1n,1)-size(F2n,1),size(F2n,2))];
    M2n=[M2n; zeros(size(M1n,1)-size(M2n,1),size(M2n,2))];
    AngleS2n=[AngleS2n; nan(size(AngleS1n,1)-
size(AngleS2n,1),size(AngleS2n,2))];
end

%we prepare our morph tables
morph_tableF=nan(size(F1n,1),morph_length);
morph_tableM=nan(size(F1n,1),morph_length);
%morph_tableS=nan(size(S1,1),morph_length);
morph_tableAngleS=nan(size(AngleS1n,1),morph_length);

%from 1 to zero along the morphing time
k=1:-1/morph_length:0;

%this iteration takes care of the actual morphing
for i=1:nof_continuous

    for j=1:morph_length
        %first the frequencies
        left=F1n(i,morph_start+j);
        right=F2n(i,morph_start+j);
        if isnan(left)
            left=right;
        end

        if isnan(right)
            right =left;
        end

        morph_tableF(i,j)=(left*k(j)+right*(1-k(j)));

        %then amplitude
        left=M1n(i,morph_start+j);
        right=M2n(i,morph_start+j);
        if isnan(left)
            left=0;

```

```

end
if isnan(right)
    right = 0;
end

morph_tableM(i,j)=left*k(j)+right*(1-k(j));

%...and finally phase
left=AngleS1n(i,morph_start+j);
right=AngleS2n(i,morph_start+j);
if isnan(left)
    left=right;
end

if isnan(right)
    right =left;
end

morph_tableAngleS(i,j)=left;

    if isnan(AngleS1n(i,morph_start+j)
+AngleS2n(i,morph_start+j))~=1
        if j<=2

morph_tableAngleS(i,j)=AngleS1n(i,morph_start+j);
end
        if j>2%lookup the instantaneous phase theory

morph_tableAngleS(i,j)=2*pi*morph_tableF(i,j)/sr*per+morph_
tableAngleS(i,j-1);
end

end

end
end

%here we take care of all the other tracks, not to be
morphed
for i=nof_continuous+1:size(F1n,1)
    for j=1:morph_length
        left=F1n(i,morph_start+j);
        right=F2n(i,morph_start+j);
        if isnan(left)
            left=0;
        end

        if isnan(right)
            right =0;
        end
    end
end

```

```

        if left==0 && right==0
            left=NaN;
            right=NaN;
        end

        if F1n(i, :)==0
            morph_tableF(i, j)=right;
        end
    morph_tableAngleS(i, j)=AngleS2n(i, morph_start+j);
    end

    if F2n(i, :)==0

    morph_tableAngleS(i, j)=AngleS1n(i, morph_start+j);
        morph_tableF(i, j)=left;
    end

    left=M1n(i, morph_start+j);
    if isnan(left)
        left=0;
    end

    right=M2n(i, morph_start+j);

    if isnan(right)
        right =0;
    end

    if left==0 && right==0
        left=NaN;
    end

    %those channels have to fade in or out, choose
    amp_power
    %according to the desired intensity of the
    effect
    morph_tableM(i, j)=left*(power(morph_length-
    j, amp_power)/power(morph_length, amp_power))
    +right*(power(j, amp_power)/power(morph_length, amp_power));

    end
end

%final tables are made out of d1, morphing time and d2
F=[F1n(:, 1:morph_start) morph_tableF
F2n(:, morph_end+1:end)];
M=[M1n(:, 1:morph_start) morph_tableM
M2n(:, morph_end+1:end)];
%this is quite complex. the phase of each channel for d2
has to be given a little boost
%to make up for the cycles that the morphing phases have
been accumulating
A=morph_tableAngleS(:, morph_length)
+2*pi*F(:, morph_end+1)/sr*per-AngleS2n(:, morph_end+1);

```

```

P=[AngleS1n(:,1:morph_start) morph_tableAngleS repmat(A,
1,size(F2n(1,morph_end+1:end),2)
+AngleS2n(:,morph_end+1:end)];

% Pad each control matrix with an extra column at each end,
% to account for the N-H = H points lost fitting the first
& last windows
P = [0*P(:,1),P,0*P(:,end)];
F = [0*F(:,1),F,0*F(:,end)];
M = [0*M(:,1),M,0*M(:,end)];

%eliminate zero tracks
F = F(any(P,2),:);
M = M(any(P,2),:);
P = P(any(P,2),:);

% Now, the phase preserving resynthesis:

dr2 = synthphtrax(F,M,P,sr,win,per);

NOISEcross=nan(1,morph_lengthNOISE);

k=1:-1/morph_lengthNOISE:0;
%crossfade between the two noise signals...
for i=1:morph_lengthNOISE
    NOISEcross(i)=noise1(morph_startNOISE+i)*k(i)
+noise2(morph_startNOISE+i)*(1-k(i));
end

noise=[noise1(1:morph_startNOISE)' NOISEcross
noise2(morph_endNOISE+1:end)'];

%...and everything adds up
morph=dr2(1:length(noise))+noise;

% Thanks to Dan Ellis for the sine model library he
unknowingly provided
%
% @misc{Ellis03-sws,
%   Author = {Daniel P. W. Ellis},
%   Year = {2003},
%   Title = {Sinewave and Sinusoid+Noise
Analysis/Synthesis in {M}atlab},
%   Url =
{http://www.ee.columbia.edu/~dpwe/resources/matlab/sinemode
l/},
%   Note = {online web resource}}
% D. P. W. Ellis (2003) "Sinewave and Sinusoid+Noise
Analysis/Synthesis in Matlab",
% web resource, available:
http://www.ee.columbia.edu/~dpwe/resources/matlab/sinemodel

```

Ο αλγόριθμος Nobeating

```
function [morph] = nobeating(d1,d2,sr,lengthofsig)
%       d1 and d2 are the two sounds that we want to smooth
out
%       sr is the sample rate of the initial samples(it has
to be the same)
%       this code relies heavily on the Athena code, hence
the morphing
%       themed comments and variables. The difference here
is that k is
%       a constant, since we want to calculate the mean of
the beating
%       channels...
%       lengthofsig is the playing time of the new signal
in seconds
%       this code has been written for the dissertation
thesis of Michael
%       Kotsirakis at the Technical University of Crete

morph_length=lengthofsig;

per=128; %windowing period
win=256; %window size

%STFT and instantaneous frequency
[I1,S1]=ifgram(d1,win,win,per,sr);
[I2,S2]=ifgram(d2,win,win,per,sr);

%well... extract tracks
[R1,M1]=extractrax(abs(S1));
[R2,M2]=extractrax(abs(S2));

F1 = colinterpvals(R1,I1);
F2 = colinterpvals(R2,I2);
% Interpolate the (columnwise unwrapped) STFT phases to get
exact peak
% phases for every sample point (synthesis phase is
negative of analysis)
P1 = -colinterpvals(R1,unwrap(angle(S1)));
P2 = -colinterpvals(R2,unwrap(angle(S2)));

%just a matter of names
AngleS1=P1;
AngleS2=P2;

%preparing the seperate sounds for synthesis
F1noise = [0*F1(:,1),F1,0*F1(:,end)];
M1noise = [0*M1(:,1),M1,0*M1(:,end)];
P1noise = [0*P1(:,1),P1,0*P1(:,end)];
```

```

F2noise = [0*F2(:,1),F2,0*F2(:,end)];
M2noise = [0*M2(:,1),M2,0*M2(:,end)];
P2noise = [0*P2(:,1),P2,0*P2(:,end)];

synth1 = synthphtrax(F1noise,M1noise,P1noise,sr,win,per);
synth2 = synthphtrax(F2noise,M2noise,P2noise,sr,win,per);

%we extract the noise of each sound
noise1=d1-synth1(1:length(d1))';
noise2=d2-synth2(1:length(d2))';

%here we calculate the morphing times in samples and in
windows
morph_startNOISE=0.01*sr;
morph_lengthNOISE=morph_length*sr;
morph_endNOISE=morph_startNOISE+morph_lengthNOISE;

morph_start=round(0.01*sr/per);
morph_length=round(morph_length*sr/per);
morph_end=morph_start+morph_length;

%here we will calculate a mean for the frequencies of each
channel
%taking into account the whole signal, we will use it to
merge channels
%around the same frequency
F1mean=zeros(size(F1,1),1);

for i=1:size(F1,1)
    petitsum=0; %excuse our french :)
    for j=1:size(F1,2)

        if ~isnan(F1(i,j))
            F1mean(i)=F1mean(i)+F1(i,j);
            petitsum=petitsum+1;
        end
    end
    F1mean(i)=F1mean(i)/petitsum;
end

F2mean=zeros(size(F2,1),1);
for i=1:size(F2,1)
    petitsum=0;
    for j=1:size(F2,2)

        if ~isnan(F2(i,j))
            F2mean(i)=F2mean(i)+F2(i,j);
            petitsum=petitsum+1;
        end
    end
end
if ~petitsum==0

```

```

        F2mean(i)=F2mean(i)/petitsum;
    end
end

%merging tracks around the same frequency for d1
newpos=1;
lef=size(F1,1);

for i=1:lef
    for j=i+1:lef
        if abs(F1mean(i)-F1mean(j))<20
            for n=1:size(F1,2)
                if isnan(F1(i,n)) && isnan(F1(j,n))
                    F1(i,n)=nan;
                    M1(i,n)=nan;
                    AngleS1(i,n)=nan;

                    elseif isnan(F1(i,n)) && ~isnan(F1(j,n))
                        F1(i,n)= F1(j,n);
                        M1(i,n)= M1(j,n);
                        AngleS1(i,n)= AngleS1(j,n);

                    elseif ~isnan(F1(i,n)) && ~isnan(F1(j,n))
                        F1(i,n)= F1(i,n);
                        M1(i,n)= M1(i,n);
                        AngleS1(i,n)= AngleS1(i,n);
                    end

                    F1(j,n)=nan;

                end
            end
        end
    end
end

%clear up unused channels
i=1;
while i<lef
    if sum(isnan(F1(i,:)))== size(F1,2)
        F1(i,:)=[];
        M1(i,:)=[];
        AngleS1(i,:)=[];
        lef=lef-1;
        i=i-1;
    end
    i=i+1;
end

%merging tracks around the same frequency for d2

lef=size(F2,1);

```

```

fin=lef;
for i=1:lef
    for j=i+1:lef
        if abs(F2mean(i)-F2mean(j))<20
            for n=1:size(F2,2)
                if isnan(F2(i,n)) && isnan(F2(j,n))
                    F2(i,n)=nan;
                    M2(i,n)=nan;
                    AngleS2(i,n)=nan;

                    elseif isnan(F2(i,n)) && ~isnan(F2(j,n))
%&& abs(F2(j,n)-F2mean(i))
                    F2(i,n)= F2(j,n);
                    M2(i,n)= M2(j,n);
                    AngleS2(i,n)= AngleS2(j,n);

                    elseif ~isnan(F2(i,n)) && ~isnan(F2(j,n))
                    F2(i,n)= F2(i,n);
                    M2(i,n)= M2(i,n);
                    AngleS2(i,n)= AngleS2(i,n);

                end

                F2(j,n)=nan;

            end

        end
    end
end

%clear up unused channels
i=1;
while i<lef
    if sum(isnan(F2(i,:)))== size(F2,2)
        F2(i,:)=[];
        M2(i,:)=[];
        AngleS2(i,:)=[];
        lef=lef-1;
        i=i-1;
    end
    i=i+1;
end

%we calculate the mean frequency for each channel this time
taking into
%account only the morphing time
F1mean=zeros(size(F1,1),1);

for i=1:size(F1,1)
    petitsum=0;
    for j=morph_start:morph_end

```

```

        if ~isnan(F1(i,j))
            Flmean(i)=Flmean(i)+F1(i,j);
            petitsum=petitsum+1;
        end
    end
    Flmean(i)=Flmean(i)/petitsum;
end

F2mean=zeros(size(F2,1),1);
for i=1:size(F2,1)
    petitsum=0;
    for j=morph_start:morph_end

        if ~isnan(F2(i,j))
            F2mean(i)=F2mean(i)+F2(i,j);
            petitsum=petitsum+1;
        end
    end
    if ~petitsum==0
        F2mean(i)=F2mean(i)/petitsum;
    end
end

for i=1:size(F1,1)
    %channels whose difference is below this threshold are
    bound to create
    %beating according to the ERB model.
    mindist=24.7*(4.37*(F1mean(i)/1000) + 1)/2;
    flag=0;

    for j=1:size(F2,1)

        dista=abs(F1mean(i)-F2mean(j));

        if dista<mindist %we still want the closest match
    for each channel
            flag=1;%a match has been found, this channel
    will be morphed
            mindist=dista;
            closest=j;

        end
    end

    if flag==1 && Flmean(i)

        %we prepare for the morphing placing these channels
    face to face
        F1n(newpos,:)=F1(i,:);
        M1n(newpos,:)=M1(i,:);
        AngleS1n(newpos,:)=AngleS1(i,:);
        F2n(newpos,:)=F2(closest,:);
        M2n(newpos,:)=M2(closest,:);
    end
end

```

```

        AngleS2n(newpos, :)=AngleS2(closest, :);
        newpos=newpos+1;
        F1mean(i)=inf;
        F2mean(closest)=inf;

    end

end

nof_continuous=newpos-1 %this tells us how many morphing
channels were ultimately found

%we find which source sound has the most channels...
if size(F1,1)>size(F2,1)
    big_size=size(F1,1);
    small_size=size(F2,1);

else
    big_size=size(F2,1);
    small_size=size(F1,1);

end

%fill up the rest of the new tables with the old values
for i=1:small_size
    if F1mean(i)~=inf
        F1n(newpos, :)=F1(i, :);
        M1n(newpos, :)=M1(i, :);
        AngleS1n(newpos, :)=AngleS1(i, :);
        newpos=newpos+1;

    end

    if F2mean(i)~=inf
        F2n(newpos, :)=F2(i, :);
        M2n(newpos, :)=M2(i, :);
        AngleS2n(newpos, :)=AngleS2(i, :);
        newpos=newpos+1;

    end

end

for i=small_size+1:big_size
    if size(F1,1)>size(F2,1)
        if F1mean(i)~=inf
            F1n(newpos, :)=F1(i, :);
            M1n(newpos, :)=M1(i, :);
            AngleS1n(newpos, :)=AngleS1(i, :);
            newpos=newpos+1;

        end
    else
        if F2mean(i)~=inf
            F2n(newpos, :)=F2(i, :);

```

```

        M2n(newpos, :)=M2(i, :);
        AngleS2n(newpos, :)=AngleS2(i, :);
        newpos=newpos+1;
    end
end
end

%
if size(F1n,1)<size(F2n,1)
    F1n=[F1n; zeros(size(F2n,1)-size(F1n,1),size(F1n,2))];
    M1n=[M1n; zeros(size(M2n,1)-size(M1n,1),size(M1n,2))];
    AngleS1n=[AngleS1n; nan(size(AngleS2n,1)-
size(AngleS1n,1),size(AngleS1n,2))];
else
    F2n=[F2n; zeros(size(F1n,1)-size(F2n,1),size(F2n,2))];
    M2n=[M2n; zeros(size(M1n,1)-size(M2n,1),size(M2n,2))];
    AngleS2n=[AngleS2n; nan(size(AngleS1n,1)-
size(AngleS2n,1),size(AngleS2n,2))];
end

%we prepare our morph tables
morph_tableF=nan(size(F1n,1),morph_length);
morph_tableM=nan(size(F1n,1),morph_length);
%morph_tableS=nan(size(S1,1),morph_length);
morph_tableAngleS=nan(size(AngleS1n,1),morph_length);

%from 1 to zero along the morphing time
k=0.5*ones(morph_length,1);

%this iteration takes care of the actual morphing
for i=1:nof_continuous

    for j=1:morph_length
        %first the frequencies
        left=F1n(i,morph_start+j);
        right=F2n(i,morph_start+j);
        if isnan(left)
            left=right;
        end

        if isnan(right)
            right =left;
        end

        morph_tableF(i,j)=(left*k(j)+right*(1-k(j)));

        %then amplitude
        left=M1n(i,morph_start+j);
        right=M2n(i,morph_start+j);
        if isnan(left)

```

```

        left=0;
    end
    if isnan(right)
        right = 0;
    end

    morph_tableM(i,j)=left*k(j)+right*(1-k(j));

    %...and finally phase
    left=AngleS1n(i,morph_start+j);
    right=AngleS2n(i,morph_start+j);
    if isnan(left)
        left=right;
    end

    if isnan(right)
        right =left;
    end

    morph_tableAngleS(i,j)=left;

    if isnan(AngleS1n(i,morph_start+j)
+AngleS2n(i,morph_start+j))~=1
        if j<=2

    morph_tableAngleS(i,j)=AngleS1n(i,morph_start+j);
        end
        if j>2%lookup the instantaneous phase theory

    morph_tableAngleS(i,j)=2*pi*morph_tableF(i,j)/sr*per+morph_
tableAngleS(i,j-1);
        end

    end

    end
end

%here we take care of all the other tracks, not to be
morphed
for i=nof_continuous+1:size(F1n,1)
    for j=1:morph_length
        left=F1n(i,morph_start+j);
        right=F2n(i,morph_start+j);
        if isnan(left)
            left=0;
        end

        if isnan(right)
            right =0;
        end
    end
end

```

```

        if left==0 && right==0
            left=NaN;
            right=NaN;
        end

        if F1n(i, :)==0
            morph_tableF(i, j)=right;
        end
    morph_tableAngleS(i, j)=AngleS2n(i, morph_start+j);
    end

    if F2n(i, :)==0

    morph_tableAngleS(i, j)=AngleS1n(i, morph_start+j);
        morph_tableF(i, j)=left;
    end

    left=M1n(i, morph_start+j);
    if isnan(left)
        left=0;
    end

    right=M2n(i, morph_start+j);

    if isnan(right)
        right =0;
    end

    if left==0 && right==0
        left=NaN;
    end

    morph_tableM(i, j)=left*k(j)+right*(1-k(j));

    end
end

%final tables are made out of d1, morphing time and d2
F=[F1n(:, 1:morph_start) morph_tableF];
M=[M1n(:, 1:morph_start) morph_tableM];

P=[AngleS1n(:, 1:morph_start) morph_tableAngleS];

% Pad each control matrix with an extra column at each end,
% to account for the N-H = H points lost fitting the first
& last windows
P = [0*P(:, 1), P, 0*P(:, end)];
F = [0*F(:, 1), F, 0*F(:, end)];
M = [0*M(:, 1), M, 0*M(:, end)];

%eliminate zero tracks
F = F(any(P, 2), :);

```

```
M = M(any(P,2),:);
P = P(any(P,2),:);

% Now, the phase preserving resynthesis:

dr2 = synthphtrax(F,M,P,sr,win,per);
noise1=noise1';
noise2=noise2';
%...and everything adds up
morph=dr2+noise1(1:length(dr2))+noise2(1:length(dr2));
```

Ο αλγόριθμος Pitchshifter

```
function [dr1] = pitchshifter(d1,sr,steps)
%      d1 is the sound that will pass through the octaver
%      sr is the sample rate of the initial sample
%      steps is the number of semitones the original
signal will be
%      shifted, it can take negative values.
%
%      this code has been written for the dissertation
thesis of Michael
%      Kotsirakis at the Technical University of Crete

per=128; %windowing period
win=256; %window size

%STFT and instantaneous frequency
[I1,S1]=ifgram(d1,win,win,per,sr);

%well... extract tracks
[R1,M1]=extractrax(abs(S1));

F1 = colinterpvals(R1,I1);

% Interpolate the (columnwise unwrapped) STFT phases to get
exact peak
% phases for every sample point (synthesis phase is
negative of analysis)
P1 = -colinterpvals(R1,unwrap(angle(S1)));

%just a matter of names
AngleS1=P1;

%preparing the seperate sounds for synthesis
F1noise = [0*F1(:,1),F1,0*F1(:,end)];
M1noise = [0*M1(:,1),M1,0*M1(:,end)];
P1noise = [0*P1(:,1),P1,0*P1(:,end)];
```

```
synth1 = synthphtrax(F1noise,M1noise,P1noise,sr,win,per);  
noise1=d1-synth1(1:length(d1))';
```

```
F1=F1*power(2,steps/12);
```

```
dr1 = synthtrax(F1,M1,sr,256,128); % Divide M by 64 to  
factor out window, FFT weighting  
noise1=noise1';  
dr1=dr1+noise1(1:length(dr1));
```

Ο αλγόριθμος SinusoidalOctaver

```
function [dr1] = sinusoidaloctaver(d1,sr,mix,upordown)  
% d1 is the sound that will pass through the octaver  
% sr is the sample rate of the initial sample  
% mix defines the mix level of the original and the  
wet signal  
% 0 yields the original sound while 1 only the wet  
sound  
% upordown indicates whether the upper(any number) or  
the lower(0) octave  
% should be produced  
% this code has been written for the dissertation  
thesis of Michael  
% Kotsirakis at the Technical University of Crete
```

```
S = specgram(d1,256);  
[R,M]=extractrax(abs(S)); % find peaks in STFT  
*magnitude*  
F = R*sr/256; % Convert R from bins to Hz
```

```
if upordown==0  
    F2=F/2;  
else  
    F2=F*2;  
end
```

```
dr2 = synthtrax(F2,M/64/4,sr,256,128);
```

```
dr1=(1-mix)*d1(1:length(dr2))+mix*dr2';
```

Ο αλγόριθμος FMvibrato

```
function [drl] = FMvibrato(d1,sr,vibratowidth,vibratofreq)
%     d1 is the sound that will pass through the vibrato
%     sr is the sample rate of the initial sample
%     vibratowidth is the amplitude of the frequency
modulating sine
%     vibratofreq is the frequency of the frequency
modulating sine
%     this code has been written for the dissertation
thesis of Michael
%     Kotsirakis at the Technical University of Crete

vibratotime=1/vibratofreq;
vibratotime=round(vibratotime*sr/128);

per=128; %windowing period
win=256; %window size

%STFT and instantaneous frequency
[I1,S1]=ifgram(d1,win,win,per,sr);

%well... extract tracks
[R1,M1]=extractrax(abs(S1));

F1 = colinterpvals(R1,I1);

% Interpolate the (columnwise unwrapped) STFT phases to get
exact peak
% phases for every sample point (synthesis phase is
negative of analysis)
P1 = -colinterpvals(R1,unwrap(angle(S1)));

%preparing the separate sounds for synthesis
F1noise = [0*F1(:,1),F1,0*F1(:,end)];
M1noise = [0*M1(:,1),M1,0*M1(:,end)];
P1noise = [0*P1(:,1),P1,0*P1(:,end)];

synth1 = synthphtrax(F1noise,M1noise,P1noise,sr,win,per);
noisel=d1-synth1(1:length(d1))';

t=0:size(F1,2)-1;
sine=sin(2*pi*t*1/vibratotime+pi*rand());
for i=1:size(F1,1)
    F1(i,:)=F1(i,:)+vibratowidth*sine;
end

drl = synthtrax(F1,M1,sr,256,128);
noisel=noisel';
drl=drl+noisel(1:length(drl));
```

BIBΛΙΟΓΡΑΦΙΑ

- [1] Marcelo Caetano, Xavier Rodet *Automatic Timbral Morphing of Musical Instrument Sounds by High-Level Descriptors* Ann Arbor, MI: MPublishing, University of Michigan Library 2010
- [2] Dave Benson *Music: A Mathematical Offering* Department of Mathematics, University of Aberdeen Web Publication 2008
- [3] Werner Kahle, Michael Frotscher *Color Atlas of Human Anatomy, vol3: Nervous System and Sensory Organs* Thieme 2002
- [4] Phil Burk, Larry Polansky, Douglas Repetto, Mary Roberts, Dan Rockmore *Music and Computers: A Theoretical and Historical Approach* Key College Publishing 2008
- [5] John G. Proakis, Masoud Salehi *Communications Systems Engineering Second Edition* Pearson Education 2002
- [6] Eli Maor *Trigonometric Delights* Princeton University Press 1998
- [7] Jeremy F. Alm, James S. Walker *Time-Frequency Analysis of Musical Instruments* SIAM REVIEW Vol. 44, No. 3, pp. 457–476 2002
- [8] J. W. Cooley, A. A. W. Lewis, P. D. Welch *The Finite Fourier Transform* IEEE Transactions On Audio And Electroacoustics June 1969
- [9] R. Plomp and H. J. M. Steeneken *Interference between two simple tones* J. Acoust. Soc. Amer. 43 (4) 1968
- [10] Wikipedia Contributors *Beat (acoustics)* Wikipedia, The Free Encyclopedia May 2013
- [11] D. D. Greenwood *Critical bandwidth and the frequency coordinates of the basilar membrane* J. Acoust. Soc. Amer. 33 (10) 1961
- [12] B. C. J. Moore and B. R. Glasberg *A revision of Zwicker's loudness model* Acta Acustica, vol. 82, pp. 335-345, 1996
- [13] B. C. J. Moore *Cochlear hearing loss*. London: Whurr Publishers Ltd. 1998
- [14] Chris Chafe *A Short History of Digital Sound Synthesis by Composers in the U.S.A.* Center for Computer Research in Music and Acoustics (CCRMA) Department of Music, Stanford

BIBΛΙΟΓΡΑΦΙΑ

University, 1999

- [15] Johannes Luig Amir Rahimzadeh *Sinusoidal Modelling and Synthesis* Seminar Work for “Algorithmen in Akustik und Computermusik II, SE” June 2008
- [16] Robert J. McAulay Thomas F. Quatieri Speech Analysis/Synthesis Based on a Sinusoidal Representation IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-34, NO. 4, AUGUST 1986
- [17] Xavier Serra *Musical Sound Modeling with Sinusoids plus Noise* “Musical Signal Processing”. Swets & Zeitlinger Publishers 1997
- [18] Boualem Boashesh *Estimating and Interpreting The Instantaneous Frequency of a Signal-Part 1: Fundamentals* PROCEEDINGS OF THE IEEE, VOL. 80, NO. 4, APRIL 1992
- [19] Frederic J. Harris *On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform* Proc. IEEE, vol.66, pp 51-83, Jan.1978
- [20] Boualem Boashesh *Estimating and Interpreting the Instantaneous Frequency of a Signal-Part 2: Algorithms and Applications* PROCEEDINGS OF THE IEEE, VOL. 80, NO. 4, APRIL 1992
- [21] Yannis Stylianou *Applying the Harmonic Plus Noise Model in Concatenative Speech Synthesis* IEEE Transactions On Speech And Audio Processing, Vol. 9, No. 1, January 2001
- [22] Marcelo Caetano, Xavier Rodet *Automatic Timbral Morphing of Musical Instrument Sounds by High-Level Descriptors* Ann Arbor, MI: MPublishing, University of Michigan Library 2010
- [23] Marcelo Caetano, Xavier Rodet *Sound Morphing by Feature Interpolation* ICASSP 2011
- [24] Stanley P. Lipshitz, Mark Pocock John Vanderkooy *On the Audibility of Midrange Phase Distortion in Audio Systems* JAES Volume 30 Issue 9 pp. 580-595; September 1982
- [25] Daniel Arfib et al. *DAFX Digital Audio Effects* John Wiley & Sons, 2002

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [26] William A. Sethares, Andrew J. Milne, Stefan Tiedje, Anthony Prechtl, and James Plamondon *Spectral Tools For Dynamic Tonality and Audio Morphing* Computer Music Journal, 33:2, pp. 71–84, Summer 2009

- [27] Malcolm Slaney, Michele Covell and Bud Lassiter *Automatic Audio Morphing* International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, May 7-10, 1996