

ΑΤΕΙ ΚΡΗΤΗΣ – ΠΑΡΑΡΤΗΜΑ ΡΕΘΥΜΝΟΥ
ΤΜΗΜΑ ΜΟΥΣΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΑΚΟΥΣΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:

**Δημιουργία και επεξεργασία μουσικής παρτιτούρας
από ζωντανές ηχητικές ροές με αξιοποίηση
αλγορίθμων αναγνώρισης τονικού ύψους**

Γραμματικάκη Μαρία Γεωργία
Α.Μ. 273

Επίβλεψη: Αλεξανδράκη Χρυσούλα

Περίληψη

Η παρούσα πτυχιακή εργασία αφορά στην ανάπτυξη μίας εφαρμογής η οποία σε πραγματικό χρόνο κάνει αναγνώριση μονοφωνικής μελωδίας από το σήμα ήχου που προέρχεται από το μικρόφωνο ή την γραμμή εισόδου line in και αυτόματα καταγράφει τα εξαγόμενα αποτελέσματα σε συμβολική μορφή (μουσική πάρτη).

Για τη δημιουργία της μουσικής πάρτης αξιοποιούνται αλγόριθμοι αναγνώρισης τονικού ύψους οι οποίοι έχουν τη δυνατότητα να εφαρμόζονται σε ζωντανές ηχητικές ροές (live audio streams). Η παραγόμενη παρτιτούρα μπορεί να εκφραστεί σε μορφή MIDI και παρέχεται δυνατότητα επεξεργασίας, αποθήκευσης και αναπαραγωγής της.

Στα πλαίσια της εργασίας γίνεται εκτενής μελέτη συναφούς βιβλιογραφίας και υπάρχοντος αντίστοιχου λογισμικού που διατίθεται στο εμπόριο.

Η χρήση της τελικής εφαρμογής ενδείκνυται για μελωδική υπαγόρευση (dictée), για την εκμάθηση κάποιου μουσικού οργάνου καθώς και για εφαρμογές που σχετίζονται με ζωντανό μουσικό αυτοσχεδιασμό.

Λέξεις Κλειδιά: Αυτόματη Δημιουργία Παρτιτούρας, Ανάκτηση Μουσικής Πληροφορίας, Αναγνώριση Τονικού Ύψους, Μουσική Σημειογραφία

Abstract

The current thesis is concerned with the development of a software application performing real-time music transcription of monophonic melody originating from an audio signal (i.e. from the microphone or line in jack). Furthermore, the application provides utilities for playback and editing the generated score.

Music score transcription uses pitch detection algorithms operating on live audio streams. The generated score can be exported in MIDI format. MIDI files may be loaded on the application and further editing may be applied. The thesis includes a comprehensive study on pitch detection algorithms and similar score generation software.

The use of the final application is intended for melodic dictation (dictée), music instrument learning, as well as live music improvisation.

Key Words: Automatic Music Transcription, Music Information Retrieval, Pitch Detection/Tracking, Musical Notation

Ευχαριστίες

Ευχαριστώ θερμά την κ. Χρυσούλα Αλεξανδράκη για την εμπιστοσύνη που μου έδειξε και την ευκαιρία που μου έδωσε να ασχοληθώ με την πτυχιακή μου εργασία. Καθώς και για την υπομονή της και την πολύτιμη βοήθεια της που είχα καθ' όλη τη διάρκεια εκπόνησης της πτυχιακής.

Επίσης, να ευχαριστήσω τους φίλους και την οικογένειά μου για την υπομονή και τη στήριξη τους σε όλη τη διάρκεια της εργασίας.

Πίνακας Περιεχομένων

1	Εισαγωγή	7
1.1	Βασική Ιδέα	7
1.2	Αυτόματη Δημιουργία Παρτιτούρας.....	7
1.2.1	Σχετιζόμενη Έρευνα.....	8
1.2.2	Εμπορικές Εφαρμογές.....	11
1.3	Μουσική Σημειογραφία	14
2	Αλγόριθμοι που αξιοποιούνται στην αυτόματη καταγραφή.....	16
2.1	Αναγνώριση Μουσικής.....	16
2.2	Αναγνώριση Τονικού Ύψους	16
2.2.1	Αναγνώριση στο πεδίο του χρόνου.....	17
2.2.2	Αναγνώριση στο πεδίο της συχνότητας.....	19
2.2.3	Αναγνώριση βασισμένη σε μοντέλα του συστήματος ακοής.....	22
2.3	Διαχωρισμός μουσικής/θορύβου	22
2.4	Κατάτμηση ηχητικών δεδομένων	22
3	Περιγραφή Εφαρμογής	24
3.1	Γραφική Διεπαφή Χρήστη	24
3.1.1	Το μενού της εφαρμογής	24
3.1.2	Τα κουμπιά ελέγχου της εφαρμογής	28
3.1.3	Το πεδίο που εμφανίζεται η παρτιτούρα	29
3.2	Χρήση της εφαρμογής.....	29
3.2.1	Καταγραφή παρτιτούρας	29
3.2.2	Αναπαραγωγή παρτιτούρας.....	30
3.2.3	Επεξεργασία παρτιτούρας	30
3.2.4	Εισαγωγή/Εξαγωγή Αρχείου MIDI.....	31
4	Υλοποίηση της Εφαρμογής	33
4.1	Γλώσσα Προγραμματισμού - Εργαλεία Υλοποίησης	33
4.1.1	Το JSyn API.....	33
4.1.2	Το JMusic API.....	34
4.2	Ιεραρχία Τάξεων.....	36
4.2.1	M2SPitchDetector.java.....	36
4.2.2	PDParmFrame.java	40
4.2.3	NewNoteListener.java	40
4.2.4	M2SNote.java	40
4.2.5	M2SStave.java	41
4.2.6	M2SNotate.java.....	41
4.2.7	Metronome.java.....	41

5	Πειραματική Αξιολόγηση	43
5.1	Σύνοψη πειραμάτων	43
5.2	Πείραμα 1 - Πιάνο	43
5.3	Πείραμα 2 – Φλάουτο	50
5.4	Πείραμα 3 – Βιολί.....	52
5.5	Πείραμα 4 – Κιθάρα	54
5.6	Σύγκριση Αποτελεσμάτων.....	55
6	Συμπεράσματα	57
6.1	Σύνοψη	57
6.2	Μελλοντικές Επεκτάσεις	58
7	Παραπομπές.....	59

1 Εισαγωγή

1.1 Βασική Ιδέα

Η παρούσα πτυχιακή εργασία αφορά τη δημιουργία μίας εφαρμογής η οποία σε πραγματικό χρόνο κάνει αναγνώριση μονοφωνικής μελωδίας από το μικρόφωνο ή τη γραμμή εισόδου line in και αυτόματα καταγράφει τα εξαγόμενα αποτελέσματα σε συμβολική μορφή (μουσική πάρτη).

Για τη δημιουργία της μουσικής πάρτης γίνεται αξιοποίηση αλγορίθμων αναγνώρισης τονικού ύψους οι οποίοι έχουν τη δυνατότητα να εφαρμόζονται σε ζωντανές ηχητικές ροές (live audio streams). Η παραγόμενη παρτιτούρα μπορεί να εκφραστεί σε μορφή MIDI και παρέχεται δυνατότητα επεξεργασίας αποθήκευσης και αναπαραγωγής της.

Στα πλαίσια της εργασίας γίνεται εκτενής μελέτη συναφούς βιβλιογραφίας και υπάρχοντος αντίστοιχου λογισμικού που διατίθεται στο εμπόριο.

Η χρήση της τελικής εφαρμογής ενδείκνυται για μελωδική υπαγόρευση (dictée), για την εκμάθηση κάποιου μουσικού οργάνου καθώς και για εφαρμογές που σχετίζονται με ζωντανό μουσικό αυτοσχεδιασμό.

1.2 Αυτόματη Δημιουργία Παρτιτούρας

Η διαδικασία της Αυτόματης Δημιουργίας Παρτιτούρας αναφέρεται στην διεθνή ορολογία ως Automatic Score Creation ή Automatic Music Transcription και είναι ένα από τα αντικείμενα μελέτης του ερευνητικού χώρου της Ανάκτησης Μουσικής Πληροφορίας (MIR - Music Information Retrieval) [1].

Ως Αυτόματη Δημιουργία Παρτιτούρας ορίζεται η διαδικασία της ανάλυσης ενός μουσικού σήματος με στόχο την αναγνώριση νωτών και την καταγραφή τους σε παρτιτούρα. Προκειμένου να αναγνωρισθεί πλήρως μία νότα, θα πρέπει να εντοπιστούν τα ακόλουθα χαρακτηριστικά από το ηχητικό σήμα:

- Τονικό ύψος (pitch)
- Χρονική στιγμή έναρξης (onset time)
- Διάρκεια (duration)
- Ηχόχρωμα – χροιά (timbre) (σε πολυφωνική μουσική με διαφορετικά όργανα)

Η καταγραφή παίρνει τη μορφή της συμβατικής μουσικής σημειογραφίας ή οποιασδήποτε συμβολικής αναπαράστασης η οποία παρέχει τις απαραίτητες πληροφορίες για την εκτέλεση ενός μουσικού κομματιού. Η παρούσα εργασία αφορά την Δυτική ή Ευρωπαϊκή μουσική σημειογραφία στην οποία οι μουσικές νότες αποτελούν τα σύμβολα μέσω των οποίων καθορίζονται τα παραπάνω χαρακτηριστικά. Συνήθως, η παρτιτούρα ενός έργου είτε παρέχεται απευθείας από το συνθέτη/εκτελεστή είτε καταγράφεται στο πεντάγραμμο από εκπαιδευμένους μουσικούς οι οποίοι ακούνε το μουσικό κομμάτι και στη συνέχεια καταγράφουν οι ίδιοι τις κατάλληλες νότες ή συγχορδίες, κάτι που απαιτεί χρόνο και προχωρημένες μουσικές γνώσεις (σχετικό μάθημα μουσικής το οποίο εξετάζει την ικανότητα αυτή της καταγραφής αποτελεί το dictée).

Η αυτοματοποιημένη καταγραφή σε ψηφιακή μορφή επιτρέπει αρκετές λειτουργίες, όπως την αναπαραγωγή και εκτέλεση του μουσικού κομματιού, τη μουσικολογική ανάλυσή του,

την εξαγωγή μουσικής πληροφορίας από μεγάλες βάσεις δεδομένων μουσικής ή την επεξεργασία του ακουστικού σήματος αναφορικά με το μουσικό περιεχόμενο. Η διαδικασία της αυτόματης καταγραφής μπορεί να ενσωματωθεί σε διαδραστικά μουσικά συστήματα ή συστήματα απομακρυσμένης εκμάθησης (Distance Learning Environment), επιτρέποντας τη μεταξύ των χρηστών ανταλλαγή αρχείων παρτιτούρας τα οποία έχουν παραχθεί με αυτόματη καταγραφή. Με τη διαδικασία αυτή λοιπόν, οι καθηγητές μουσικής ή οι χρήστες λογισμικών μουσικής σημειογραφίας και άλλων σχετικών εφαρμογών θα μπορούν εύκολα και σε σαφώς λιγότερο χρόνο να εισάγουν κάποιο μουσικό κομμάτι και να προχωρούν σε περαιτέρω επεξεργασία (για παράδειγμα σε διορθώσεις ή συμπληρώσεις) και χρήση αυτού ανάλογα με τις δυνατότητες του εργαλείου το οποίο χρησιμοποιούν (για παράδειγμα θα μπορούν να αποστείλουν το αρχείο στον μαθητή προς μελέτη).

Συμπερασματικά, η αυτόματη δημιουργία της μουσικής σημειογραφίας για ένα ακουστικό σήμα θα διευκόλυνε σημαντικά το έργο της εκμάθησης μουσικής και θα διευρύνει τις δυνατότητες των χρηστών μουσικών εφαρμογών λογισμικού.

1.2.1 Σχετιζόμενη Έρευνα

Η αυτόματη καταγραφή της μουσικής έχει μακρά ιστορία ως ερευνητικό θέμα καθώς πρωτοεμφανίστηκε τη δεκαετία του '70. Έχει κάνει μια σταθερή πρόοδο διότι η δυσκολία της μουσικής ανάλυσης μεγαλώνει βαδίζοντας από τη μελέτη μονοφωνικών μελωδιών σε πολυφωνικούς ήχους και μίξεις ήχων διαφορετικών μουσικών οργάνων. Η εξέλιξη αυτή συνοδεύτηκε από μια στροφή προς πιο εξειδικευμένα θέματα έρευνας, όπως ο διαχωρισμός των από διαφορετική πηγή προερχόμενων ήχων (sound source separation) και η εκτίμηση της θεμελιώδους συχνότητας (F0), που κατά κανόνα αντιστοιχεί στο τονικό ύψος (pitch detection) [2].

Η αυτόματη καταγραφή μονοφωνικών ακουστικών σημάτων είναι πρακτικά ένα πρόβλημα το οποίο έχει επιλυθεί καθώς αρκετοί αλγόριθμοι έχουν προταθεί, οι οποίοι είναι αξιόπιστοι, εμπορικά εφαρμόσιμοι και λειτουργούν σε πραγματικό χρόνο, και σε αρκετές περιπτώσεις έχουν ενσωματωθεί στον εξοπλισμό στούντιο, περιλαμβάνοντας pitch-to-Midi μετατροπή επιτρέποντας στη συνέχεια προσαρμογή και διόρθωση. Η παρούσα εργασία βασίζεται στην προσέγγιση αυτή πραγματεύεται την κατάλληλη κωδικοποίηση των δεδομένων που ανακτώνται για τη δημιουργία ενός αρχείου κατανοητού από τους μουσικούς, την παρτιτούρα [3][4].

Αντίθετα, κάποιο εμπορικό σύστημα **αυτόματης καταγραφής πολυφωνικής μουσικής** γενικής εφαρμογής δεν έχει ακόμα εκδοθεί. Σε ένα άρθρο [5], ο Anssi Klauri, συζητά το πρόβλημα της καταγραφής μουσικής. Δηλώνει πως παρά τον σημαντικό αριθμό των προσπαθειών για την επίλυση του προβλήματος, πρακτικά ένα γενικού σκοπού εφαρμόσιμο σύστημα καταγραφής δεν έχει μέχρι σήμερα αναπτυχθεί. Ωστόσο, ερευνητές ανά τον κόσμο έχουν παρουσιάσει ποικίλες προσεγγίσεις για την καταγραφή συγκεκριμένων κατηγοριών ήχου της μουσικής.

Η αυτόματη καταγραφή της μουσικής στο σύνολό της δεν είναι καθόλου απλή και εμπεριέχει μια σειρά υποπροβλημάτων προς επίλυση:

- Εκτίμηση των θεμελιωδών συχνότητων μονοφωνικών και πολυφωνικών μουσικών ήχων
- Εκτίμηση της χρονικής διάρθρωσης των ήχων (του μουσικού ρυθμού)
- Αναγνώριση των μουσικών οργάνων
- Καταγραφή των κρουστών
- Αρμονική ανάλυση και καταγραφή των συγχορδιών

- Υψηλότερο επίπεδο μουσικολογικής δομής για την επίλυση διφορούμενων καταστάσεων.

Όπως προκύπτει από τη βιβλιογραφία, μεγαλύτερη βαρύτητα έχει δοθεί στην εκτίμηση των ταυτόχρονων θεμελιωδών συχνοτήτων πολυφωνικών ήχων.

Μεγάλος αριθμός ερευνητών οι οποίοι επιχειρήσαν την αυτόματη καταγραφή πολυφωνικής μουσικής έχουν παρουσιαστεί τα τελευταία τουλάχιστον 30 χρόνια, μετά την δουλειά του Mouzer τη δεκαετία του '70 [6]. Τα διάφορα συστήματα ανωτάτου επιπέδου τεχνικής παρέμειναν χωρίς αμφιβολία φτωχά σε σύγκριση με τις δυνατότητες των ίδιων των μουσικών ως προς την ακρίβεια και την ευελιξία. Τα τελευταία 15 χρόνια ωστόσο έχει παρατηρηθεί μια σημαντική πρόοδος.

Οι διαφορετικές προσεγγίσεις της αυτόματης καταγραφής της μουσικής περιλαμβάνουν, εκτός από το πολύ βασικό πεδίο έρευνας της επίλυσης μίξεων ήχων μουσικών οργάνων τα οποία παράγουν τόνο/pitch (σε αντιδιαστολή με τα μουσικά όργανα που παράγουν κρότο και ονομάζονται άτονα) (**polyphonic pitch recognition/detection**), το οποίο βασίζεται σε τεχνικές επεξεργασίας σήματος και προτείνει μεθόδους εκτίμησης των ταυτόχρονων θεμελιωδών συχνοτήτων (F0) ή διαχωρισμού των ήχων που συνυπάρχουν, μελέτες αναφορικά με την καταγραφή των κρουστών (percussion transcription) και την κατηγοριοποίηση των μουσικών οργάνων.

Μια ακόμη προσέγγιση αποτελεί η ανάλυση του **μουσικού τοπίου ή δομής (music scene)**. Οι προτεινόμενες μέθοδοι στην περίπτωση αυτή στηρίζονται στην κατανόηση του μουσικού ακουστικού σήματος στο επίπεδο των μουσικά ανειδίκευτων ακροατών. Αντίθετα δηλαδή με την προσέγγιση η οποία βασίζεται σε τεχνικές επεξεργασίας σήματος, και στοχεύει στην ανάλυση και κατανόηση του ακουστικού σήματος στο επίπεδο των καταρτισμένων μουσικών ακροατών, είτε αναγνωρίζοντας το σύνολο από νότες που αποτελούν ένα μουσικό κομμάτι είτε διαχωρίζοντας τα επιμέρους σήματα που το συνθέτουν, η περιγραφή του μουσικού τοπίου έγκειται στην αναγνώριση των διακριτικών γεγονότων που χαρακτηρίζουν μια μουσική εκτέλεση, όπως η μελωδική γραμμή (melodic line), τα μπάσα (bass line), ο ρυθμός (rhythm ή tempo tracking), το ρεφρέν και η επανάληψη φράσεων, καθώς και η ποιότητα του ήχου (ηχώχρωμα/timbre) των μουσικών οργάνων [7].

Για να ορίσουμε το βαθμό στον οποίο είναι εφικτοί οι στόχοι αναφορικά με την αυτόματη καταγραφή της μουσικής, είναι ουσιαστικό να ξεκινήσουμε από τη μελέτη του τί είναι ικανοί οι άνθρωποι ως ακροατές να αναγνωρίσουν. Ο μέσος ακροατής μπορεί να αντιληφθεί αρκετές ιδιότητες της μουσικής σε ένα σύνθετο ακουστικό σήμα. Μπορεί με ευκολία να παρακολουθήσει ακόμη και ασυνείδητα το ρυθμό, με απλούς χτύπους του χεριού ή ποδιού για παράδειγμα, να μουρμουρίσει τη βασική μελωδία (περισσότερο ή λιγότερο σωστά), να αναγνωρίσει τα διαφορετικά μουσικά όργανα ή κάποια από αυτά και να διαχωρίσει την ορχήστρα από τις στροφές ενός τραγουδιού. Οι αλλαγές της αρμονίας και οι ποικίλες μουσικές λεπτομέρειες απαιτούν συνειδητή προσπάθεια ώστε να γίνουν αντιληπτές. Όμοια με την φυσική μας γλώσσα, η ανάγνωση και η καταγραφή της μουσικής απαιτούν εκπαίδευση. Δεν αρκεί η εκμάθηση της μουσικής σημειογραφίας, καθώς η αναγνώριση των διαφόρων τονικών διαστημάτων και οι χρονικές σχέσεις που ενυπάρχουν στη μουσική θα πρέπει να κατανοηθούν σε βάθος. Θα πρέπει λοιπόν κανείς να έχει την ικανότητα να κωδικοποιήσει αυτές τις παραμέτρους με κάποιον συμβολικό τρόπο στο μυαλό του προτού προχωρήσει στην καταγραφή τους. Όσο πιο πλούσια σε φωνές είναι η μουσική υπό εξέταση, τόσο μεγαλύτερη είναι η απαίτηση για ένα εκπαιδευμένο μουσικά αυτί και για γνώσεις σχετικές με το μουσικό είδος το οποίο μελετάται και τις ποικίλες τεχνικές εκτέλεσης του εκάστοτε μουσικού οργάνου.

Σε αυτό το σημείο είναι χρήσιμο να αναφερθούμε στην **ιστορία της αυτόματης μουσικής καταγραφής** που ακολούθησε μετά την εμφάνιση του Moorer και του συστήματος καταγραφής για δύο φωνές που υλοποίησε [6], [8] όπως παρουσιάζεται συνολικά στο πρώτο κεφάλαιο (§1.Introduction) του βιβλίου “Signal Processing Methods for Music Transcription” των Anssi Klapuri και Manuel Davy (2006) [7].

“ Οι Chafe et al. [9], Piszcalski [10] και Maher [11] συνέχισαν τη δουλειά του Moorer κατά τη δεκαετία του '80. Στα πρώιμα αυτά συστήματα, ο αριθμός των ταυτόχρονων φωνών ήταν περιορισμένος στις δύο και οι σχέσεις των ταυτόχρονων pitch περιοριστικές με διαφόρους τρόπους. Στο κομμάτι της ανάλυσης του ρυθμού, ο πρώτος αλγόριθμος εκτίμησης του ρυθμού σε γενικού περιεχομένου ακουστικά σήματα προτάθηκε από τους Goto και Muraoka [12] την επόμενη δεκαετία, του '90, παρότι είχε προηγηθεί αξιοσημείωτος όγκος δουλειάς στην εκτίμηση του ρυθμού παραμετρικών δεδομένων από νότες και ο αλγόριθμος του Schloss [13] για την εκτίμηση του ρυθμού κρουστών οργάνων. Οι πρώτες απόπειρες καταγραφής των κρουστών οργάνων έγιναν στα μέσα του '80 από τον Schloss [13] και αργότερα από τον Bilmes [14]. Και οι δυο κατηγοριοποιούσαν διαφορετικού τύπου “χτύπους” σε συνεχείς ηχογραφήσεις. Η καταγραφή πολυφωνικών κρουστών ήχων πραγματοποιήθηκε αργότερα από τους Goto και Muraoka [15]. Μια αναλυτική παρουσίαση των πρώτων αυτών σταδίων της εξέλιξης της μουσικής καταγραφής έχει γίνει από τον Tanguiane [16].

Από την αρχή της δεκαετίας του '90, το ενδιαφέρον για την μουσική καταγραφή έχει μεγαλώσει ταχύτατα και δεν είναι εφικτό να γίνει ο συνολικός απολογισμός της δουλειάς στα πλαίσια αυτής της εργασίας. Για το λόγο αυτό, θα αναφερθούμε μόνο σε συγκεκριμένες τάσεις και επιτυχημένες προσεγγίσεις που έχουν σημειωθεί. Μια από αυτές είναι η αξιοποίηση **στατιστικών μεθόδων (statistical methods)**. Σημαντικά παραδείγματα χρήσης στατιστικών μεθόδων στην ανάλυση των πολλαπλών pitch της πολυφωνικής μουσικής αποτελούν οι μέθοδοι που προτάθηκαν από τους Kashimo et al [17], Goto [18], Davy και Godsill [19], και Ryyanen και Klapuri [20]. Στο πεδίο της εκτίμησης του ρυθμού, στατιστικές μέθοδοι εφαρμόστηκαν από τους Cemgil και Karpen [21], Hainsworth και MacLeod [22], και Klapuri et al. [23], ενώ στη μελέτη της καταγραφής κρουστών οργάνων από τους Gillet και Richard [24] και Paulus και Klapuri [25]. Η χρήση όμως στατιστικών μεθόδων αναγνώρισης προτύπων επικράτησε στο πεδίο της κατηγοριοποίησης των μουσικών οργάνων [26].

Μιά ακόμη τάση αποτέλεσε η χρησιμοποίηση **υπολογιστικών μοντέλων του ανθρώπινου ακουστικού συστήματος (computational models of the human auditory system)**. Τέτοιες τεχνικές για την μουσική καταγραφή παρουσιάστηκαν από τον Martin [27], και οδήγησαν στην πρόταση σχετικών μεθόδων για την ανάλυση των pitch πολυφωνικών ήχων από τους Karjalainen και Tolonen [28] και Klapuri [29] και μεθόδων για την εκτίμηση του ρυθμού από τον Scheiner [30], για παράδειγμα.

Μιά σπουδαία προσέγγιση ήταν αυτή της μοντελοποίησης του **μουσικού τοπίου (auditory scene analysis - asa)**, η οποία μελετά τα όσα γίνονται εύκολα αντιληπτά από κάθε άνθρωπο. Οι αρχές του asa εφαρμόστηκαν στην ανάλυση των συχνοτήτων πολυφωνικών μουσικών σημάτων από τους Mellinger [31] και Kashimo και Tanaka [32], και αργότερα από τους Godsmark και Brown [33] και τους Sterian et al. [34]. Πιο πρόσφατα, κάποιες μέθοδοι **μάθησης χωρίς επίβλεψη (unsupervised learning)** έχουν προταθεί κατά τις οποίες ένας ελάχιστος αριθμός από πρωταρχικές υποθέσεις λαμβάνονται για το υπό ανάλυση σήμα.

Μέθοδοι βασισμένες στην **ανάλυση ανεξάρτητων συνιστωσών (independent component analysis)** προτάθηκαν από τον Casey [35], [36] και άλλες διαφορετικές μέθοδοι παρουσιάστηκαν αργότερα από τους Lepain [37], Smaragdis [38], [39], Abdallah [40], [41], Virtanen (§9.Unsupervised Learning Methods for Source Separation in Monaural Music Signals), FitzGerald [42], [43] και Paulus και Virtanen [44].”

Εκτός από την αναφορά στις σημαντικότερες τάσεις και εργασίες που έχουν παρουσιαστεί κρίνεται εξίσου απαραίτητη η παρουσίαση της συγκριτικής αξιολόγησης αυτών όπως παρουσιάζεται στη συνέχεια του ίδιου κεφαλαίου του βιβλίου “Signal Processing Methods for Music Transcription” των Anssi Klauri και Manuel Davy (2006) [7].

“ Ένα πρακτικά γενικού σκοπού εφαρμόσιμο σύστημα καταγραφής δεν έχει ακόμη υπάρξει. Παρόλα αυτά έχει επιτευχθεί μιά κάποια επιτυχία στην καταγραφή πολυφωνικής μουσικής θέτοντας περιορισμούς στην πολυπλοκότητα. Στην καταγραφή οργάνων τα οποία παράγουν τόνους (pitch), τυπικό περιορισμό αποτελεί ο μέγιστος αριθμός των ταυτόχρονων ήχων [28], [19], η παρεμβολή ντραμς και άλλων κρουστών δεν επιτρέπεται [45], ή μόνο ένα συγκεκριμένο όργανο λαμβάνεται υπόψη [46]. Κάποια ελπιδοφόρα αποτελέσματα στην καταγραφή μουσικής του πραγματικού κόσμου από ηχογραφήσεις CD έχουν να επιδείξουν οι Goto [18] και οι Ryyanen και Klauri [20].

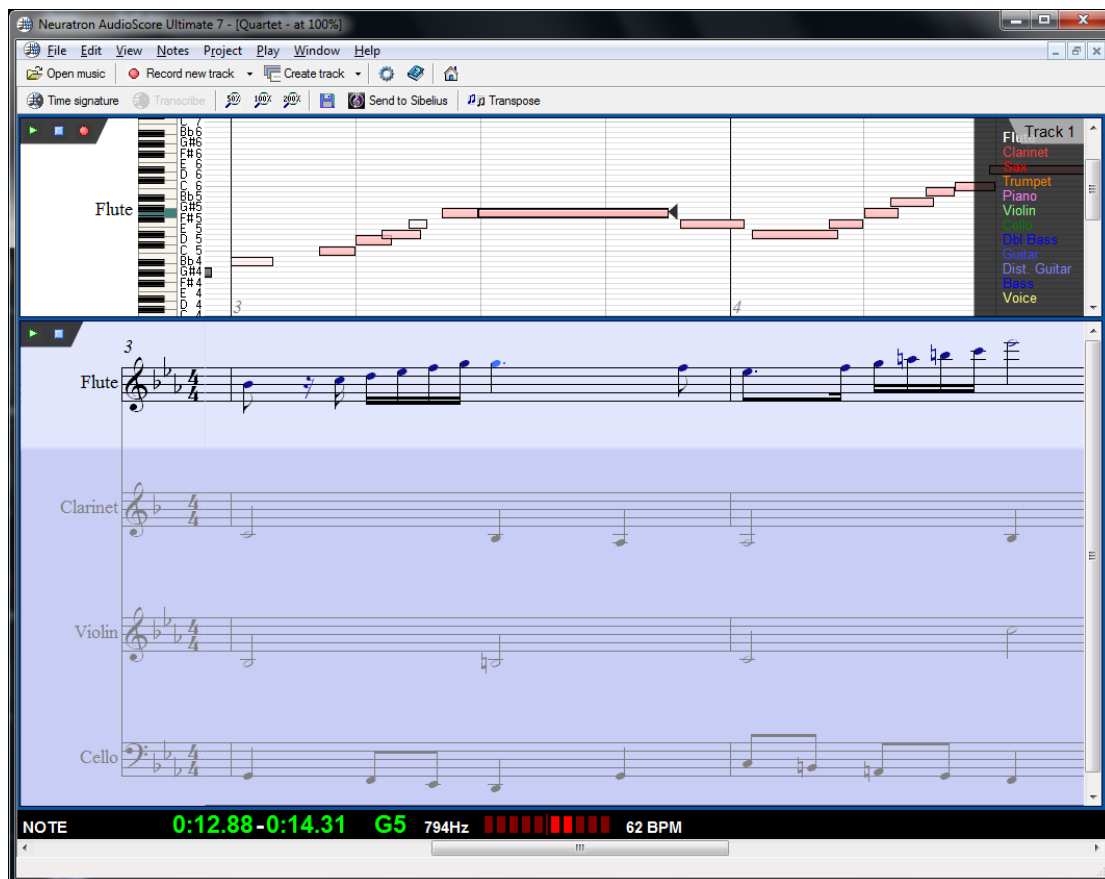
Στην καταγραφή των κρουστών έχει επιτευχθεί αρκετά καλή ακρίβεια η οποία περιλαμβάνει περιορισμένο αριθμό μουσικών οργάνων και άτονα μουσικά όργανα (μουσικά όργανα τα οποία δεν παράγουν τόνο/pitch αλλά κρότο) [24], [44]. Επίσης υποσχόμενα αποτελέσματα έχουν παρουσιαστεί στην καταγραφή των μπάσων και του τυμπάνου σε πραγματικού κόσμου ηχογραφήσεις, αλλά αυτό αποτελεί ένα ακόμη ανοιχτό πρόβλημα (Zils et al. [47], FitzGerald et al. [48], Yoshii et al. [49]).

Η εκτίμηση του ρυθμού πολύπλοκων ακουστικών σημάτων μπορεί να γίνει με αρκετή αξιοπιστία με τις δεδομένες ανωτάτου επιπέδου τεχνικές, αλλά οι δυσκολίες παραμένουν ιδιαίτερα κατά την ανάλυση της κλασικής μουσικής και ρυθμικά περίπλοκου υλικού. Μια συγκριτική αξιολόγηση των εργασιών εκτίμησης του ρυθμού παρουσιάζεται στα [22], [50] και [23]. Οι έρευνες αναφορικά με την κατηγοριοποίηση των μουσικών οργάνων έχουν επικεντρωθεί στο επίπεδο των μεμονωμένων ήχων, παρότι πιο πρόσφατα επιχειρούνται και σε πολυφωνικά ακουστικά σήματα [51], [52], [53], [54].”

Το βασικότερο ίσως συμπέρασμα που προκύπτει από τα όσα έχουν αναφερθεί είναι πως στην αναζήτηση της δημιουργίας ενός ολοκληρωμένου και αποτελεσματικού συστήματος προς όφελος της μουσικής επιστήμης και τέχνης, απαιτείται συνδυαστική γνώση από περισσότερα του ενός πεδία έρευνας - επεξεργασία σήματος, ακουστική, επιστήμη των υπολογιστών, μουσική, γλωσσολογία και πειραματική ψυχολογία [55].

1.2.2 Εμπορικές Εφαρμογές

Η πιο γνωστή εμπορική εφαρμογή αυτόματης μουσικής καταγραφής είναι το AudioScore Ultimate [56] της εταιρίας Neuratron (κυκλοφόρησε πρώτη φορά το 1999). Το πρόγραμμα AudioScore είναι συμβατό με μουσικά προγράμματα όπως το Sibelius, το Finale, το Cubase κ.α. Πλέον στις νεότερες εκδόσεις πέραν της δυνατότητας που έχει για αναγνώριση και καταγραφή μονοφωνικής μουσικής, μπορεί να γίνει και πολυφωνικής μουσικής από μουσικά cd ή αρχεία mp3.



ΕΙΚΟΝΑ 1-1: ΠΑΡΑΔΕΙΓΜΑ ΕΚΤΕΛΕΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ AUDIOSCORE ULTIMATE 7 ΤΗΣ NAURATRON. ΑΝΑΓΝΩΡΙΣΗ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΠΟΛΥΦΩΝΙΚΗΣ ΜΟΥΣΙΚΗΣ.

Συνοπτικά χαρακτηριστικά του προγράμματος AudioScore Ultimate 7:

- Αυτόματη δημιουργία παρτιτούρας από μουσικές ηχητικές ροές (μόνο για μονοφωνική μουσική), αναγνωρίζοντας νότες και ρυθμικά μοτίβα σε πραγματικό χρόνο
- Αυτόματη δημιουργία παρτιτούρας από αρχεία ήχου mp3, μουσικά cds και αρχεία ήχου wav (πολυφωνικής και μονοφωνικής μουσικής)
- Αυτόματη δημιουργία παρτιτούρας από αρχεία MIDI
- Δυνατότητα αποθήκευσης της παρτιτούρας σε τύπους αρχείων NIFF (.nif), MusicXML (.xml) και PhotoScore (.opt)
- Δυνατότητα επεξεργασίας της παρτιτούρας μέσω του πενταγράμμου ή του piano roll (εικονικού πιάνου)
- Καταγραφή νοτών και παύσεων μέχρι και αξία 32ου, τρίηχα, αναγνώριση κλειδιού και κλίμακας
- Άνοιγμα της εξαγόμενης παρτιτούρας απευθείας στο πρόγραμμα Sibelius
- Αναγνώριση μέχρι και 16 νότες/όργανα, που ηχούν ταυτόχρονα, από αρχεία ήχου mp3, wav ή audio cd
- Εύρος αναγνώρισης τονικού ύψους: F0 – C8 (22Hz – 4186Hz)

Ακολουθεί ο πίνακας 1-1 με αντίστοιχες εφαρμογές που υπάρχουν στο εμπόριο.

ΠΙΝΑΚΑΣ 1-1: ΕΦΑΡΜΟΓΕΣ ΑΥΤΟΜΑΤΗΣ ΜΟΥΣΙΚΗΣ ΚΑΤΑΓΡΑΦΗΣ ΠΟΛΥΦΩΝΙΚΗΣ Η ΜΟΝΟΦΩΝΙΚΗΣ ΜΟΥΣΙΚΗΣ.

Πρόγραμμα	Εκδότης	Real-time Mic To Score	Μονοφωνική /Πολυφωνική	Ιστοσελίδα
7Canaries	TallStick Software	Ναι	πολυφωνική	http://tallstick.com/
AKoff Music Composer	AKoff Sound Lab	Όχι	πολυφωνική	http://www.akoff.com
AmazingMIDI	Tetsuya Araki	Όχι	πολυφωνική	http://www.pluto.dti.ne.jp/~araki/amazingmidi
Audioscore	Neuratron	Ναι	πολυφωνική	http://www.neuratron.com
Capella	WHC	Όχι	μονοφωνική	http://www.whc.de
Digital Ear	Epinois Software	Ναι	μονοφωνική	http://www.digital-ear.com
Inst2midi	Nerd	Ναι	μονοφωνική	http://www.nerds.de/english/inst2midi.html
Intelliscore	Innovative Music Systems, Inc	Ναι	πολυφωνική	http://www.intelliscore.net
Melodyne	Celemony	Ναι	πολυφωνική	http://www.celemony.com
Solo Explorer	Recognisoft	Ναι	μονοφωνική	http://www.recognisoft.com
Sound2MIDI	EarMaster ApS	Ναι	μονοφωνική	http://www.earmaster.com/sound2midi/
WIDI	WidiSoft	Ναι	πολυφωνική	http://www.widisoft.com

Όπως παρουσιάστηκε στην παράγραφο 1.2.1 και από τη μελέτη των συγκεκριμένων προγραμμάτων παρατηρούμε ότι μια ολοκληρωμένη εφαρμογή της αυτόματης δημιουργίας παρτιτούρας από ηχητικές ροές δεν έχει υπάρξει ακόμα. Οι περιορισμοί είναι ακόμα μεγάλοι και οι αδυναμίες πολλές.

Στην ιστοσελίδα της εφαρμογής Intelliscore οι δημιουργοί της αναφέρουν προς τους χρήστες του ότι η εφαρμογή έχει περισσότερη ακρίβεια όταν το ηχητικό υλικό που θα μετατραπεί σε παρτιτούρα είναι απλό, με λίγα όργανα και αν είναι δυνατόν χωρίς κρουστά. Αναφέρουν ότι μπορεί να υπάρχουν κρουστά αλλά τα αποτελέσματα θα είναι σε μεγάλο βαθμό ανακριβή και η παρτιτούρα που προκύπτει πιθανότατα να χρειάζεται διορθώσεις. Παρόλα αυτά, οι χρήστες της εφαρμογής αναφέρουν ότι η χρήση της εφαρμογής, τους εξοικονομεί 35% του χρόνου τους από το να έγραφαν μόνοι τους την παρτιτούρα.

Επίσης, στην ιστοσελίδα μία άλλης εφαρμογής, του Solo Explorer, αναφέρουν τους περιορισμούς που έχει η εφαρμογή και παραθέτουν έναν πίνακα με το ποσοστό σφάλματος της εφαρμογής.

ΠΙΝΑΚΑΣ 1-2: ΠΟΣΟΣΤΑ ΣΦΑΛΜΑΤΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ SOLO EXPLORER.

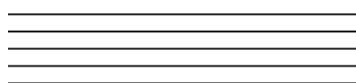
Σφάλμα αναγνώρισης συχνότητας, %	0.33
Λανθασμένα διάσπαση νότας, %	4.60
Χαμένες-μη αναγνωρισμένες νότες, %	18.50
Σφάλμα αναγνώρισης ρυθμού, %	25.70
Σφάλμα εκτίμησης νότας, %	6.60

1.3 Μουσική Σημειογραφία

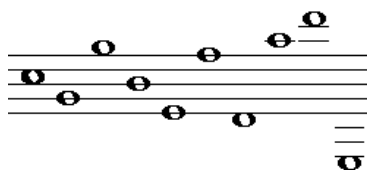
Η συμβολική αναπαράσταση της μουσικής (μουσικές παρτιτούρες) εφευρέθηκε με σκοπό τη δημιουργία εγγράφων μουσικών δημιουργιών τα οποία αντέχουν στο χρόνο και οι μουσικοί μπορούν να πάρουν τις πληροφορίες που χρειάζονται για την εκτέλεση της μουσικής.

Υπάρχουν πολλών ειδών μουσικά σύμβολα στην παγκόσμια μουσική σημειογραφία, που ποικίλουν ανάλογα το είδος της μουσικής ή τη γεωγραφική περιοχή (π.χ. αραβική, βυζαντινή παρασημαντική, αρχαία αγκιστροειδής κλπ). Διεθνώς όμως, σαν πρότυπο έχει επικρατήσει η Ευρωπαϊκή σημειογραφία που μας ενδιαφέρει εδώ άμεσα.

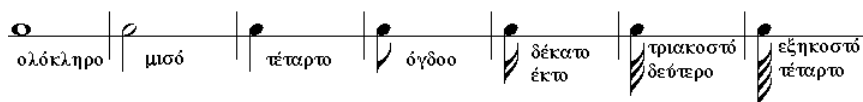
Η πλατφόρμα της Ευρωπαϊκής μουσικής σημειογραφίας ονομάζεται πεντάγραμμο και είναι πέντε (5) παράλληλες, οριζόντιες, ευθείες γραμμές, που ισαπέχουν μεταξύ τους και αποτελούν τη βάση που θα τοποθετηθούν τα μουσικά σύμβολα.



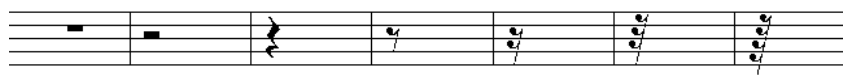
Οι ήχοι που αντιστοιχούν σε συχνότητες του ακουστικού αλλά και ωφέλιμου για τη μουσική φάσματος (frequency range) ονομάζονται νότες ή φθογγόσημα και τοποθετούνται στις 5 γραμμές ή στα 4 διαστήματα του πενταγράμμου, ακόμα και στις βοηθητικές γραμμές πάνω ή κάτω από το πεντάγραμμο.



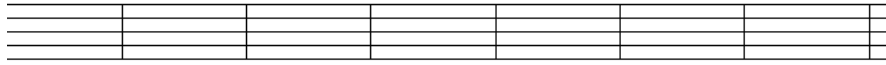
Η διάρκεια των φθογγόσημων στη μονάδα του χρόνου λέγεται αξία. Η αξία των φθογγόσημων ακολουθεί τη σειρά: ολόκληρα, μισά, τέταρτα, όγδοα, δέκατα έκτα, τριακοστά δεύτερα και εξηκοστά τέταρτα. Συμβολίζονται δε ως εξής:



Υπάρχουν φθογγόσημα που δεν αντιστοιχούν σε συχνότητα ακουστικού φάσματος, αλλά ισοδυναμούν με διαστήματα σιωπής. Αυτά λέγονται παύσεις. Οι παύσεις χρονικά αντιστοιχίζονται με τα φθογγόσημα που αντιπροσωπεύουν ακουστούς ήχους. Συμβολίζονται δε ως εξής (με την ίδια αντιστοιχία ως προς τις νότες):



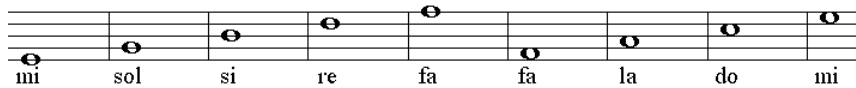
Το πεντάγραμμο χωρίζουμε με κάθετες διαστολές, που λέγονται γραμμές του μέτρου. Η απόσταση μεταξύ δύο διαστολών ονομάζεται μέτρο.



Τα φθογγόσημα που εμπεριέχονται σε κάθε μέτρο, πρέπει να ισοδυναμούν (σε αξία) χρονικά με τον αριθμό που βρίσκεται στο πρώτο (ή το πρώτο που συναντάται από αριστερά) μέτρο. Ο αριθμός αυτός ονομάζεται ρυθμός και καθορίζει το πλήθος των φθογγόσημων προς τη χρονική τους αξία. (Ακολουθεί παράδειγμα για ρυθμό τριών τετάρτων):



Για την ονοματολογία των φθογγόσημων, στην Ελλάδα χρησιμοποιείται στο Ιταλικό σύστημα, ενώ στις υπόλοιπες Ευρωπαϊκές χώρες το Σαξονικό και Γερμανικό. Έτσι, οι νότες για το Ιταλικό σύστημα είναι: Ντο, ρε, μι, φα, σολ, λα, σι. Ενώ για το Σαξονικό σύστημα είναι: C,D,E,F,G,A,B. Εδώ πρέπει να αναφέρουμε ότι οι νότες ονοματίζονται από κάτω προς τα πάνω. Έτσι, στις 5 γραμμές του πενταγράμμου βρίσκονται οι νότες μι, σολ, σι, ρε, φα (από τη χαμηλότερη γραμμή προς την υψηλότερη), ενώ στα 4 διαστήματα βρίσκονται οι νότες φα, λα, ντο, μι αντίστοιχα.



Πρέπει να προσθέσουμε ότι για λόγους ευκολίας, απόδοσης των διαφόρων μουσικών οργάνων με διαφορετική έκταση φωνής αλλά και ευκολίας γραφής, γίνεται συχνά η χρήση κλειδιών, όπως το κλειδί του σολ, του φα ή του ντο σε διαφορετικές κατά περίπτωση θέσεις.



κλειδιά του φα, ντο και σολ

Ακολουθεί η διαδοχική σειρά των φθογγόσημων από το χαμηλότερο ήχο στον υψηλότερο που καλείται σκάλα ή κλίμακα με τη χρήση (και το σύμβολο) των κλειδιών του Φα και του Σολ αντίστοιχα (στο παράδειγμα η σκάλα ξεκινά από ντο και καταλήγει σε ντο):



2 Αλγόριθμοι που αξιοποιούνται στην αυτόματη καταγραφή

2.1 Αναγνώριση Μουσικής

Στόχος του παρόντος κεφαλαίου είναι η περιγραφή ενός συστήματος αυτόματης αναγνώρισης μονοφωνικής μουσικής. Σε γενικές γραμμές το σήμα εισόδου ενός τέτοιου συστήματος μπορεί να είναι είτε ένα προ-ηχογραφημένο ηχητικό μονοφωνικό σήμα, οπότε και αναφέρεται ως offline pitch detection, είτε μια ζωντανή μονοφωνική ηχητική ροή (live audio stream) προερχόμενη από μικρόφωνο ή γραμμή line in. Στη δεύτερη περίπτωση εφαρμόζονται αλγόριθμοι για online pitch detection, με σημαντικότερες προκλήσεις και απαιτήσεις «πραγματικού χρόνου». Σε κάθε περίπτωση η έξοδος ενός τέτοιου συστήματος είναι μια συμβολική/περιγραφική μορφή της μουσικής η οποία μπορεί να αναπαρίσταται είτε ως παρτιτούρα είτε ως ροή δεδομένων MIDI.

Στην αναγνώριση μουσικής, περιλαμβάνεται ένα σύνολο από επιμέρους υπο-προβλήματα που πρέπει να αντιμετωπιστούν, τα οποία είναι:

- [1] Διαχωρισμός μουσικής/θορύβου (όπου θόρυβος νοείται οτιδήποτε δεν είναι μουσική)
- [2] Ανίχνευση σιωπής (που διαφέρει από το παραπάνω καθώς η σιωπή μπορεί να αποτελεί μουσικό συμβάν – π.χ. μουσική παύση)
- [3] Αναγνώριση τονικού ύψους
- [4] Ανίχνευση αρχής νότας (onset)
- [5] Εκτίμηση διάρκειας νότας

Παρακάτω παρουσιάζονται με περισσότερες λεπτομέρειες τα βασικά στάδια και περιγράφονται οι μέθοδοι για την επίλυσή τους όπως προκύπτουν από τη βιβλιογραφία.

2.2 Αναγνώριση Τονικού Ύψους

Το πρώτο και κύριο στάδιο για την καταγραφή μουσικής σε παρτιτούρα είναι η αναγνώριση του τονικού ύψους (pitch detection) ή όπως είναι διαφορετικά γνωστό σαν όρος εκτίμηση της θεμελίου συχνότητας F0 (fundamental frequency estimation), κάθε μουσικού συμβάντος (νότας). Σημαντική έρευνα έχει γίνει στο συγκεκριμένο αντικείμενο εδώ τα τελευταία 50 χρόνια με αποτέλεσμα να έχει δημιουργηθεί ένα πλήθος μεθόδων για την αναγνώριση τονικού ύψους, βασισμένων σε διάφορες μαθηματικές αρχές.[57] [58]

Υπάρχουν πέντε ειδών βασικές κατηγορίες που εμπίπτουν οι αλγόριθμοι αναγνώρισης τονικού ύψους:

- [1] Αναγνώριση στο πεδίο του χρόνου (Time Domain detection)
- [2] Αναγνώριση στο πεδίο της συχνότητας (Frequency Domain detection)
- [3] Αναγνώριση βασισμένη σε μοντέλα του συστήματος της ανθρώπινης ακοής (Perceptual detection)

2.2.1 Αναγνώριση στο πεδίο του χρόνου

Αυτή είναι η πιο απλή προσέγγιση, κατά την οποία γίνεται εξέταση του σήματος εισόδου για την παρακολούθηση των διακυμάνσεων πλάτους στο πεδίο του χρόνου, και εντοπίζονται επαναλαμβανόμενα μοτίβα στην κυματομορφή, που δίνουν ενδείξεις ως προς την περιοδικότητα της. Από την αντιστροφή της τιμής της περιόδου προκύπτει η τιμή της συχνότητας. Αυτού του είδους οι τεχνικές μπορούν να εφαρμοστούν μόνο σε μονοφωνική μουσική και έχουν καλά αποτελέσματα μόνο όταν υπάρχει μεγάλη περιοδικότητα στο σήμα. Το θετικό όμως είναι ότι μπορούν να υλοποιηθούν εύκολα και έχουν καλή απόδοση ως προς το χρόνο εξαγωγής των αποτελεσμάτων.

Μία προσέγγιση αναγνώρισης στο πεδίο του χρόνου είναι η **Zero Crossings (Διέλευση από το μηδέν)**, κατά την οποία μετράται η απόσταση μεταξύ δυο σημείων διέλευσης της κυματομορφής από το μηδέν. Ωστόσο, αυτό δεν λειτουργεί καλά με σύνθετες κυματομορφές που αποτελούνται από πολλαπλά κύματα ημιτόνου με διαφορετικές χρονικές περιόδους. Παρ' όλα αυτά, υπάρχουν περιπτώσεις στις οποίες η μέθοδος Zero Crossings μπορεί να είναι αποτελεσματική, για παράδειγμα σε ορισμένες εφαρμογές ομιλίας, όπου υπάρχει μια και μοναδική πηγή. Για σήματα με ισχυρή θεμέλιο η μέθοδος Zero Crossings λειτουργεί σωστά, αγνοώντας τυχών μικρές απότομες διελεύσεις από το μηδέν που υπάρχουν στο σήμα από της υψηλές αρμονικές. [59]

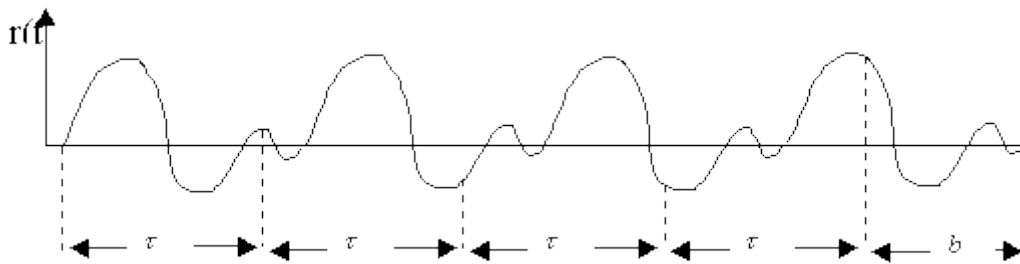
Σε πιο σύνθετες προσεγγίσεις, αυτές της **Autocorrelation (Αυτοσυσχέτισης)** συγκρίνονται τα τμήματα του σήματος με άλλα τμήματα του ίδιου σήματος, μετατοπισμένα στο χρόνο. Σκοπός αυτής της μεθόδου είναι να εντοπίζει ομοιότητες στα συγκρινόμενα σήματα, οι οποίες είναι ενδείξεις περιοδικότητας. Μέρος του σήματος εισόδου παραμένει σε ένα buffer και καθώς νέο μέρος του σήματος εισέρχεται στη μονάδα συγκρίνεται τα δύο μέρη του σήματος ελέγχοντας για επαναλαμβανόμενα μοτίβα. Εάν βρεθούν, αυτό σημαίνει ότι το σήμα είναι περιοδικό, μετράται ο χρόνος μεταξύ των δυο όμοιων μοτίβων και υπολογίζεται η θεμέλιος συχνότητα. Η βασική μαθηματική συνάρτηση της Αυτοσυσχέτισης μιας ακολουθίας $x(k)$ μήκους N είναι,

$$r_{xx}(n) = \frac{1}{N} \sum_{k=0}^{N-n-1} (x(k) * x(k+n)) \quad 2-1$$

όπου n είναι η μετατόπιση του σήματος, ή το μήκος της περιόδου και $x(n)$ το μετατοπισμένο στο χρόνο σήμα. Αυτή η τεχνική είναι αρκετά απλή, γρήγορη και αξιόπιστη ειδικά στη συχνοτική περιοχή των μεσαίων. Αν εφαρμοστή μόνη της χρειάζεται μεγάλη επεξεργαστική ισχύς αλλά αυτό το πρόβλημα λύνεται εύκολα όταν συνδυάζεται με άλλες τεχνικές. Όπως η Zero Crossing, με χρήση φίλτρων πριν το σήμα περάσει στη μονάδα Αυτοσυσχέτισης, ή όταν προηγούνται μετασχηματισμοί Fourier. [60]

Μια τεχνική που ανέπτυξε ο Noll [61] είναι η **Maximum Likelihood (Μέγιστη Πιθανότητα)** όπου προσπαθεί να βρει το βέλτιστο τρόπο αναγνώρισης τονικού ύψους στο πεδίο του χρόνου. Ο Noll προσπάθησε να αποδείξει ότι η μέθοδος του είναι πραγματικά η μέγιστη πιθανότητα. Για να περιγράψει τη μεθόδό του, θεωρεί ένα περιοδικό σήμα $r(t)$ μήκους T με άγνωστη περίοδο τ . Το σήμα χωρίζεται σε N τμήματα μήκους τ , ($1 < \tau \leq T$), για παράδειγμα

$$T = N\tau + b \quad 0 \leq b \leq \tau \quad 2-2$$



ΕΙΚΟΝΑ 2-1: ΤΟ ΣΗΜΑ ΧΩΡΙΖΕΤΑΙ ΣΕ Ν ΤΜΗΜΑΤΑ ΜΗΚΟΥΣ τ ΣΥΝ ΕΝΑ ΕΠΙΠΛΕΟΝ ΤΜΗΜΑ ΜΗΚΟΥΣ b .

Στη συνέχεια, τα τμήματα αυτά προστίθενται (διαδοχικά), σύμφωνα με:

$$r(t, \tau) = \begin{cases} \frac{1}{N+1} \sum_{n=0}^N r(t+n\tau) & 0 \leq t \leq b \\ \frac{1}{N} \sum_{n=0}^N r(t+n\tau) & b \leq t \leq \tau \end{cases} \quad \text{2-3}$$

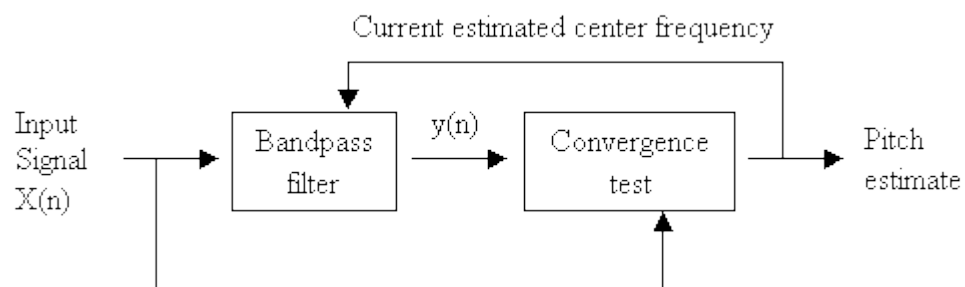
Αυτά τα τμήματα προστίθεται διαδοχικά, όταν $\tau = \tau_0$, έτσι έχουμε την παρακάτω αντικειμενική συνάρτηση:

$$J(\tau) = (N+1) \sum_{t=0}^b r^2(t, \tau) + N \sum_{t=b+1}^{\tau} r^2(t, \tau) \quad \text{2-4}$$

μεγιστοποιώντας το τ μέχρι να γίνει το και βρίσκοντας έτσι το μέγιστο και κατά συνέπεια την θεμέλιο συχνότητα.

Μία άλλη μέθοδος αναγνώρισης τονικού ύψους είναι αυτή των **Προσαρμοστικών Φίλτρων (AFPD – Adaptive Filter Pitch Detectors)**, όπου υπάρχουν πολλές προσεγγίσεις. Παρακάτω παρουσιάζονται δυο από αυτές.

- a. **Narrow band pass filter:** Το σήμα εισόδου περνά από ένα φίλτρο διέλευσης ζώνης. Στη συνέχεια τα δυο αυτά σήματα (αυτό που έχει περάσει από το φίλτρο αλλά και το καθαρό σήμα εισόδου) περνούν από διαφορετικά κυκλώματα αναγνώρισης τονικού ύψους. Στη συνέχεια, το εξαγόμενο σήμα οδηγείτε ξανά στο φίλτρο διέλευσης ζώνης ώστε να ελέγξει την κεντρική συχνότητα του φίλτρου, με αποτέλεσμα το φίλτρο να συγκλίνει με τη συχνότητα του σήματος εισόδου.



ΕΙΚΟΝΑ 2-2: ΑΝΑΓΝΩΡΙΣΗ ΤΟΝΙΚΟΥ ΥΨΟΥΣ ΜΕ ΤΗ ΧΡΗΣΗ ΦΙΛΤΡΟΥ ΔΙΕΛΕΥΣΗΣ ΖΩΝΗΣ.

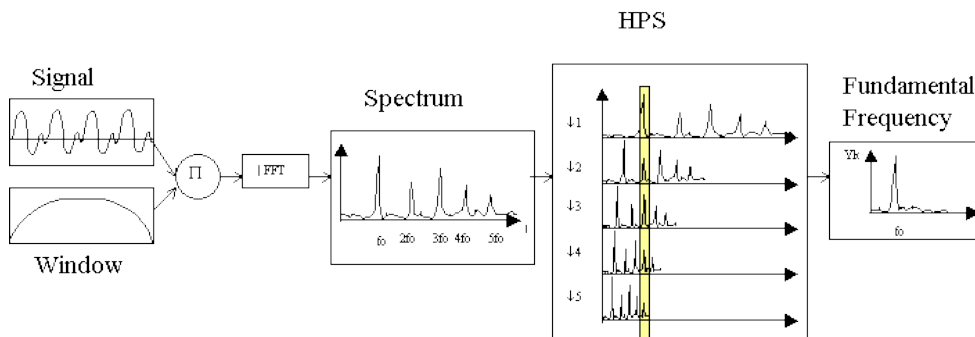
- b. **Optimum Comb Method:** Σε αυτή τη μέθοδο σκοπός είναι να βρεθεί το φίλτρο «χτένας» που θα ελαχιστοποιεί το σήμα εισόδου. Για να γίνει αυτό, οι εγκοπές του φίλτρου χτένας θα πρέπει να συντονιστούν στη συχνότητα του σήματος εισόδου. Έτσι βρίσκοντας το βέλτιστο φίλτρο χτένας, εντοπίζεται και η θεμέλιος συχνότητα.

2.2.2 Αναγνώριση στο πεδίο της συχνότητας

Στις μεθόδους αναγνώρισης στο πεδίο της συχνότητας το σήμα εισόδου χωρίζεται σε μικρά τμήματα από συχνότητες οι οποίες αποτελούν το συνολικό εύρος συχνοτήτων. Μέσα σε αυτό το φάσμα οι συχνότητες έχουν διαφορετικές εντάσεις. Στόχος είναι να εντοπίζεται η ισχυρότερη συχνότητα, που συνήθως είναι και η θεμέλιος.

Μια τυπική προσέγγιση αναγνώρισης στο πεδίο της συχνότητας είναι η **Short Time Fourier Transform (STFT) (Μετασχηματισμοί Fourier Μικρής Διάρκειας)**. Σκοπός αυτής της μεθόδου είναι να εντοπίζονται κορυφές στο συχνοτικό φάσμα, που αντιστοιχούν στις κυρίαρχες συχνότητες. Αφού εντοπιστούν οι κορυφές, η μονάδα αναγνώρισης θα πρέπει να αποφασίσει ποιες από αυτές τις κορυφές είναι θεμέλιες συχνότητες και ποιες είναι αρμονικές. Ένας γρήγορος αλγόριθμος ανίχνευσης θεμελίου συχνότητας απλά θα επιλέξει για τονικό ύψος τη συχνότητα με τη μεγαλύτερη ένταση. Ένας πιο εξελιγμένος αλγόριθμος θα ελέγξει για σχέσεις μεταξύ των αρμονικών και από αυτές θα αποφανθεί για την θεμέλιο. Ένα πρόβλημα αυτής της μεθόδου είναι ότι χωρίζει το φάσμα συχνοτήτων σε ίσα τμήματα, ενώ σε αντίθεση η ανθρώπινη ακοή είναι λογαριθμική, έτσι η αναγνώριση τονικού ύψους στις χαμηλές συχνότητες είναι πολύ συχνά λανθασμένη. Κι εδώ το πρόβλημα λύνεται με το συνδυασμό διαφορετικών μεθόδων.

Μια άλλη μέθοδος στο πεδίο της συχνότητας είναι η **Harmonic Product Spectrum (Φάσμα Γινομένου Αρμονικών)**, όπου ο προσδιορισμός της θεμελίου συχνότητας γίνεται μετρώντας τις συχνότητες των υψηλών αρμονικών και υπολογίζοντας το μέγιστο κοινό διαιρέτη αυτών. Για τον προσδιορισμό του μέγιστου κοινού διαιρέτη δημιουργείτε ένα ιστόγραμμα για κάθε αρμονικό και στους ακέραιους διαιρέτες του. Η συχνότητα με τη μέγιστη κορυφή στο ιστόγραμμα αντιπροσωπεύει και το μέγιστο κοινό διαιρέτη, κατ επέκταση και τη θεμέλιο συχνότητα του σήματος. Μερικά θετικά χαρακτηριστικά της μεθόδου αυτής είναι το χαμηλό κόστος, η ανοχή της στον θόρυβο και προσαρμοστικότητα στις μεταβολές του τύπου σήματος εισόδου (αλλαγές του αριθμού των αρμονικών προς εξέταση κ.α.). Παρακάτω παρουσιάζεται το διάγραμμα αυτής της μεθόδου.

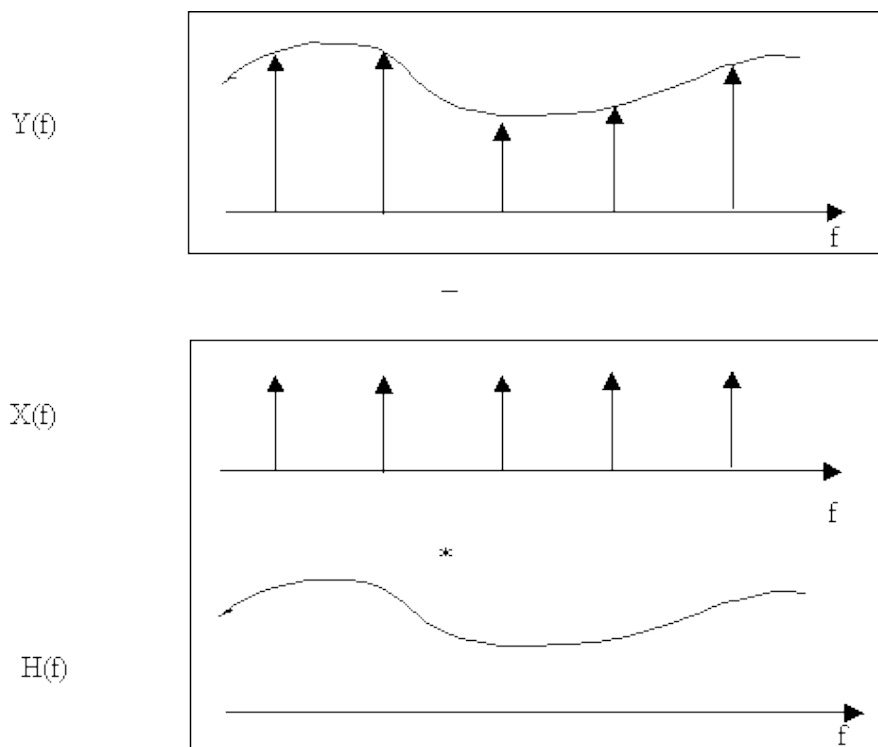


ΕΙΚΟΝΑ 2-3: ΥΛΟΠΟΙΗΣΗ ΜΕΘΟΔΟΥ HARMONIC PRODUCT SPECTRUM

Η μέθοδος **Ceptrum** είναι ακόμα μια προσέγγιση στο πεδίο της συχνότητας, η ονομασία της οποία προέρχεται από την αντιστροφή των τεσσάρων πρώτων γραμμάτων της λέξης

spectrum (φάσμα). Ένας απλός τρόπος για να περιγράψουμε τη μέθοδο Cepstrum είναι ότι διαχωρίζει το κομμάτι του σήματος με την πιο ισχυρή τονικότητα από το υπόλοιπο τμήμα του σήματος. Αυτό μπορεί να εφαρμοστεί με καλά αποτελέσματα σε φωνητικά ή σε ήχο από μουσικά όργανα που το φάσμα τους είναι το άθροισμα της αρχικής διέγερσης (οι αρχικοί παλμοί ταλάντωσης, που είναι και η συχνότητα του σήματος) και των συντονισμών (το φιλτραρισμένο μέρος του σήματος που δημιουργείται από το σώμα του μουσικού οργάνου).

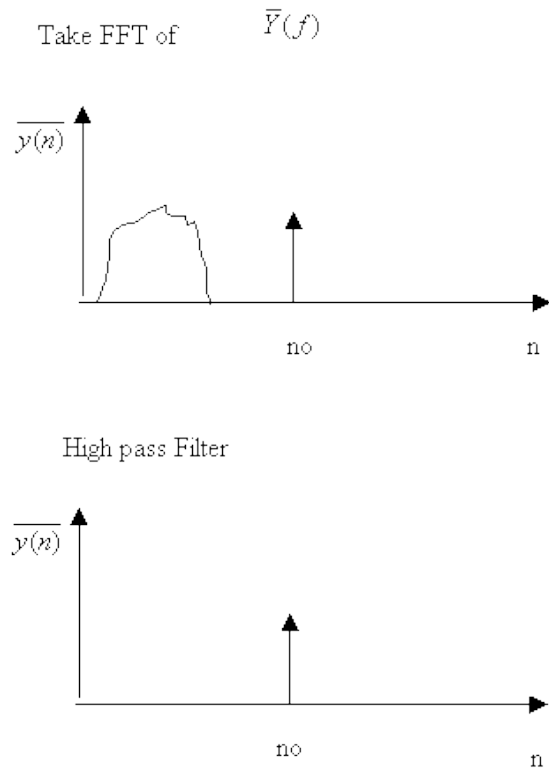
Τεχνικά, η μέθοδος **Cepstrum**, είναι η αντιστροφή του μετασχηματισμού Fourier ενός λογαριθμικού φάσματος Fourier. Αυτό είναι η απόλυτη τιμή του λογάριθμου (με βάση το 10) της εξόδου ενός διακριτού μετασχηματισμού Fourier. Το αποτέλεσμα των υπολογισμών Cepstrum είναι μία ακολουθία στο χρόνο, όπως και το σήμα εισόδου. Αν το σήμα εισόδου είναι αρμονικό, αυτό σημαίνει ότι θα είναι περιοδικό στην γραφική παράσταση προς τη συχνότητα. Παίρνοντας πάλι το μετασχηματισμό Fourier, θα έχουμε μία κορυφή η οποία θα αντιστοιχεί στη θεμέλιο συχνότητα, έτσι απομονώνουμε τη θεμελιώδη περίοδο. Αυτή η διαδικασία μπορεί να περιγραφεί και σαν μία διαδικασία αποσυνέλιξης (de-convolution). Αν το σήμα εισόδου παράγεται από μία σειρά παλμών σε συνέλιξη με ένα φίλτρο που θα πολλαπλασιαστεί στο πεδίο της συχνότητας και στη συνέχεια μετατρέποντας σε λογάριθμο θα έχουμε μετατροπή από πολλαπλασιασμό σε πρόσθεση. Εφαρμόζοντας και πάλι μετασχηματισμό Fourier θα έχουμε αποσυνέλιξη και παίρνουμε τη θεμέλιο συχνότητα. Παρακάτω παρουσιάζεται σε διαγράμματα αυτή η διαδικασία .



$$\text{Log}(Y(f)) = \text{Log}(X(f)) + \text{Log}(H(f))$$

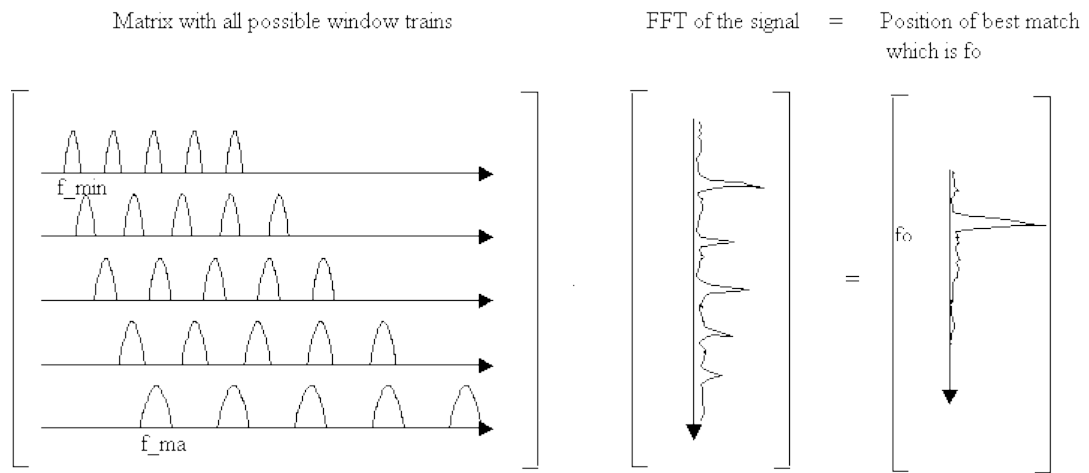
$$\bar{Y}(f) = \bar{X}(f) + \bar{H}(f)$$

ΕΙΚΟΝΑ 2-4: ΣΥΝΕΛΙΞΗ ΣΤΟ ΧΡΟΝΟ = ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ ΣΤΗ ΣΥΧΝΟΤΗΤΑ, ΠΟΥ ΜΕΤΑΤΡΕΠΕΤΑΙ ΣΕ ΠΡΟΣΘΕΣΗ ΜΕ ΤΗ ΧΡΗΣΗ ΛΟΓΑΡΙΘΜΟΥ.



ΕΙΚΟΝΑ 2-5: Η ΘΕΜΕΛΙΩΔΗΣ ΠΕΡΙΟΔΟΣ ΜΠΟΡΕΙ ΝΑ ΥΠΟΛΟΓΙΣΤΕΙ ΠΕΡΝΟΝΤΑΣ ΤΟ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟ FOURIER ΚΑΙ ΠΕΡΝΟΝΤΑΣ ΤΟ ΣΗΜΑ ΑΠΟ ΦΙΛΤΡΟ.

Η μέθοδος **Maximum Likelihood (Μέγιστη Πιθανότητα)** μπορεί να εφαρμοστεί και στο πεδίο της συχνότητας. Η ιδέα είναι να γίνει συσχέτιση της FFT ανάλυσης του φάσματος με όλες τις πιθανές σειρές κρουστικών (impulse trains), όπου η μέγιστη συσχέτιση θα γίνει στη θεμέλιο συχνότητα. Παρακάτω παρουσιάζεται γραφικά η διαδικασία αυτής της τεχνικής.

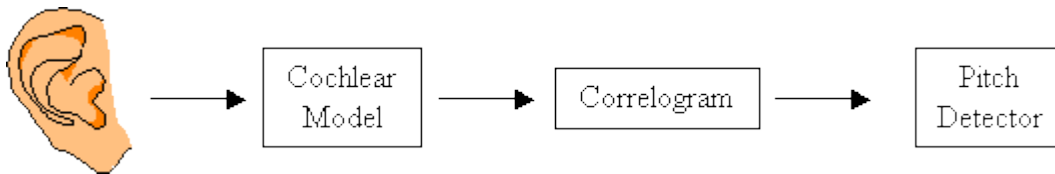


ΕΙΚΟΝΑ 2-6: ΥΛΟΠΟΙΗΣΗ ΤΕΧΝΙΚΗΣ MAXIMUM LIKELIHOOD ΣΤΟ ΠΕΔΙΟ ΤΗΣ ΣΥΧΝΟΤΗΤΑΣ

2.2.3 Αναγνώριση βασισμένη σε μοντέλα του συστήματος ακοής

Μετά από δεκαετίες συστηματικής μελέτης, η επιστήμη της ακοολογίας έχει καταφέρει να αναλύσει με μεγάλη λεπτομέρεια το σύστημα της ανθρώπινης ακοής. Έτσι δημιουργήθηκε ένας αλγόριθμος αναγνώρισης τονικού ύψους (**Perceptual Pitch Detector**), από τους Slaney M. και Lyon R.F [62] βασισμένος στη θεωρία “Duplex Theory” του Licklider, που αφορά την αντίληψη του ήχου από το σύστημα της ανθρώπινης ακοής. Ένα τέτοιο σύστημα αποτελείται από τρία μέρη: το εξωτερικό και μέσο αυτί, ο κοχλίας και το κεντρικό νευρικό σύστημα.

Το πρώτο στάδιο του αλγορίθμου είναι μια αρχική επεξεργασία του σήματος μέσα από φίλτρα βασισμένα στις αποκρίσεις του εξωτερικού και του μέσου αφτιού. Το επόμενο στάδιο είναι η μετατροπή του σήματος εισόδου σε αναπαράσταση στο πεδίο της συχνότητας μέσω της επεξεργασίας του από μία σειρά φίλτρων διέλευσης ζώνης και η ενέργεια στη βασική μεμβράνη (basilar membrane) του κοχλίου μετατρέπεται σε πυροδοτήσεις των νεύρων και ακολούθως σε μια σειρά από διεγέρσεις στο πεδίο του χρόνου. Στο επόμενο στάδιο επεξεργασίας δημιουργεί το correlogram, ένα γράφημα δύο διαστάσεων, στο οποίο κάθε σειρά είναι αυτοσυσχέτιση μικρού χρόνου για κάθε κοχλιακό κανάλι. Τέλος, στο στάδιο της αναγνώρισης του τονικού ύψους συνδυάζονται οι πληροφορίες από όλες τις σειρές του correlogram και εκτιμάται η θεμέλιος συχνότητα.



ΕΙΚΟΝΑ 2-7:ΤΑ ΣΤΑΔΙΑ ΑΝΑΓΝΩΡΙΣΗΣ ΤΟΝΙΚΟΥ ΥΨΟΥΣ ΤΗΣ ΜΕΘΟΔΟΥ PERCEPTUAL PITCH DETECTOR

2.3 Διαχωρισμός μουσικής/θορύβου

Πριν από το βασικό στάδιο της αναγνώρισης της μουσικής νότας, συνήθως προηγείται μια βαθμίδα αρμόδια για την ανίχνευση μουσικής κατά την οποία γίνεται διαχωρισμός μεταξύ μουσικής και θορύβου. Αυτό είναι απαραίτητο για την ανίχνευση και απόρριψη τμημάτων που δεν περιέχουν μουσική, ούτως ώστε αφενός να αποφεύγεται η λανθασμένη αναγνώριση ψευδών νοτών και αφετέρου να μην δαπανώνται υπολογιστικοί πόροι.

2.4 Κατάτμηση ηχητικών δεδομένων

Η χρονική κατάτμηση ενός ακουστικού κύματος (temporal segmentation) σε πιο σύντομα τμήματα είναι ένα θεμελιώδες βήμα στο μετασχηματισμό των ήχων σε σημασιολογικά αντικείμενα (semantic objects).

Στις τελευταίες δύο δεκαετίες σημαντική έρευνα έχει αφιερωθεί σε αυτό το αντικείμενο και διαφορετικοί αλγόριθμοι έχουν αναπτυχθεί για να χωρίσουν αυτόματα τα σήματα μουσικής στα όρια των αντικειμένων του ήχου: όπου η νότα αρχίζει - onset - και τελειώνει - offset [63],[64]. Η εξαγωγή των χρόνων των onsets είναι χρήσιμη στις εφαρμογές επεξεργασίας ήχου για την ακριβή διαμόρφωση των attacks των ήχων [65], βοηθά τα συστήματα καταγραφής στον εντοπισμό των onset των νοτών [55] και μπορεί να

χρησιμοποιηθεί σε προγράμματα σύνταξης ήχων (sound editors) για τον διαχωρισμό των ηχητικών αρχείων στα λογικά τους μέρη. Οι μέθοδοι ανίχνευσης των onsets έχουν χρησιμοποιηθεί για την ταξινόμηση της μουσικής [66] και τον χαρακτηρισμό ρυθμικών προτύπων [67].

Διάφορες προσεγγίσεις για την ανίχνευση των onset σε μουσικούς ήχους παρουσιάζονται στη διεθνή βιβλιογραφία από χρονικές και filter-bank τεχνικές έως και στατιστικές

μεθόδους. Αυτές οι προσεγγίσεις έχουν γενικά δύο στόχους: την κατασκευή μιας συνάρτησης ανίχνευσης των αλλαγών στο σήμα, και την επιλογή της κορυφής αυτής της συνάρτησης ώστε να εξάγουμε τους αντιλαμβανόμενους χρόνους των onset [68].

3 Περιγραφή Εφαρμογής

3.1 Γραφική Διεπαφή Χρήστη

Η εφαρμογή αποτελείται από ένα κεντρικό παράθυρο, στο οποίο υπάρχουν 3 επιμέρους τμήματα,

- το μενού
- τα κουμπιά ελέγχου της εφαρμογής
- το πεδίο που εμφανίζεται η παρτιτούρα

και έχει την παρακάτω μορφή,



ΕΙΚΟΝΑ 3-1: ΤΟ ΚΕΝΤΡΙΚΟ ΠΑΡΑΘΥΡΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή έχει όνομα **Real-Time Automated Music Transcription**, το οποίο και εμφανίζεται σαν τίτλος του κεντρικού παραθύρου.

3.1.1 Το μενού της εφαρμογής

Το μενού αποτελείται από 3 καρτέλες, με επιλογές χρήσης της εφαρμογής, στις οποίες εμφανίζονται και οι συντομεύσεις τους από το πληκτρολόγιο,

- File
- Tools
- Play

File

Το μενού File έχει την παρακάτω μορφή και επιλογές,

File	Tools	Play
New		Ctrl+N
Open MIDI file...		Ctrl+O
Open jMusic XML file...		
Open jm file..		
Close		Ctrl+W
Clear all notes		Ctrl+X
Save as a MIDI file...		Ctrl+S
Save as a jMusic XML file...		
Save as a jm file...		
Quit		Ctrl+Q

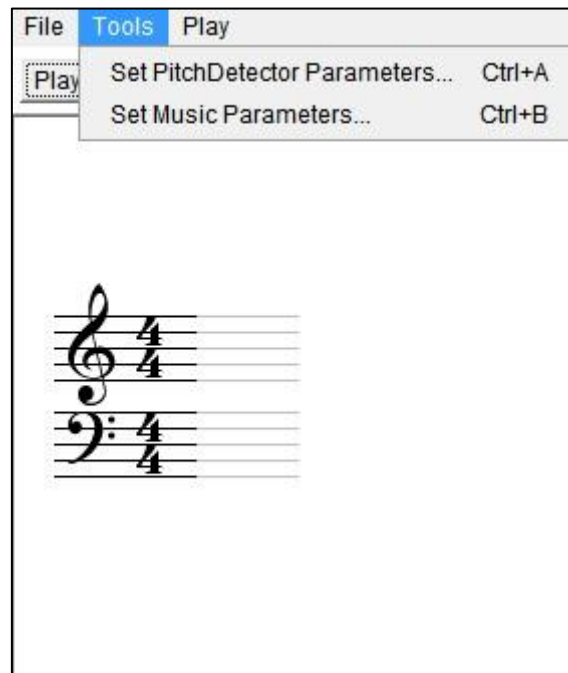
ΕΙΚΟΝΑ 3-2: ΤΟ ΜΕΝΟΥ FILE

Οι επιλογές του μενού File,

- **New** (συντόμευση από το πληκτρολόγιο Ctrl+N) – ανοίγει ένα νέο παράθυρο
- **Open MIDI file** (συντόμευση από το πληκτρολόγιο Ctrl+O) – ανοίγει ένα MIDI αρχείο, το οποίο εμφανίζεται με μορφή παρτιτούρας, την οποία ο χρήστης μπορεί αναπαράγει ηχητικά, να τροποποιήσει και να σώσει σε αρχείο MIDI ή σε κάποιο άλλο από τα διαθέσιμα είδη αρχείων
- **Open jMusic XML file**– ανοίγει ένα JMusic XML αρχείο (ουσιαστικά ένα XML αρχείο), το οποίο εμφανίζεται με μορφή παρτιτούρας, την οποία ο χρήστης μπορεί αναπαράγει ηχητικά, να τροποποιήσει και να σώσει σε αρχείο JMusic XML ή κάποιο άλλο από τα διαθέσιμα είδη αρχείων
- **Open jm file** - ανοίγει ένα jm αρχείο (αρχείο του JMusic), το οποίο εμφανίζεται με μορφή παρτιτούρας, την οποία ο χρήστης μπορεί αναπαράγει ηχητικά, να τροποποιήσει και να σώσει σε αρχείο jm ή κάποιο άλλο από τα διαθέσιμα είδη αρχείων
- **Close** (συντόμευση από το πληκτρολόγιο Ctrl+W) - κλείνει το παράθυρο της εφαρμογής που είναι σε πρώτο πλάνο (αν είναι πολλά παράθυρα ανοιχτά) ή γίνεται έξοδος της εφαρμογής (αν είναι μόνο ένα παράθυρο ανοιχτό)
- **Clear all notes** (συντόμευση από το πληκτρολόγιο Ctrl+C) – αδειάζει η παρτιτούρα από όλες τις νότες που υπάρχουν
- **Save as a MIDI file** (συντόμευση από το πληκτρολόγιο Ctrl+S) – εξάγεται η παρτιτούρα σε μορφή αρχείου MIDI
- **Save as a JMusic XML file** – εξάγεται η παρτιτούρα σε μορφή αρχείου XML
- **Save as a jm file** – εξάγεται η παρτιτούρα σε μορφή αρχείου jm (αρχείο της βιβλιοθήκης JMusic)
- **Quit** (συντόμευση από το πληκτρολόγιο Ctrl+Q) - έξοδος της εφαρμογής (κλείνουν όλα τα ανοιχτά παράθυρα της εφαρμογής)

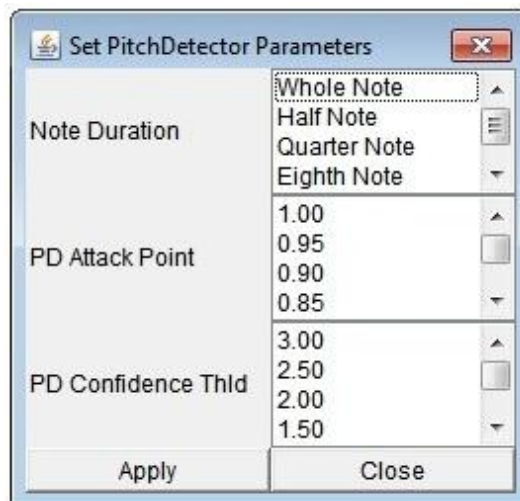
Tools

Το μενού Tools έχει την παρακάτω μορφή και δυο επιλογές,



ΕΙΚΟΝΑ 3-3: ΤΟ ΜΕΝΟΥ TOOLS

Επιλέγοντας ο χρήστης της εφαρμογής το **Set PitchDetector Parameters**, ανοίγει το παρακάτω παράθυρο,

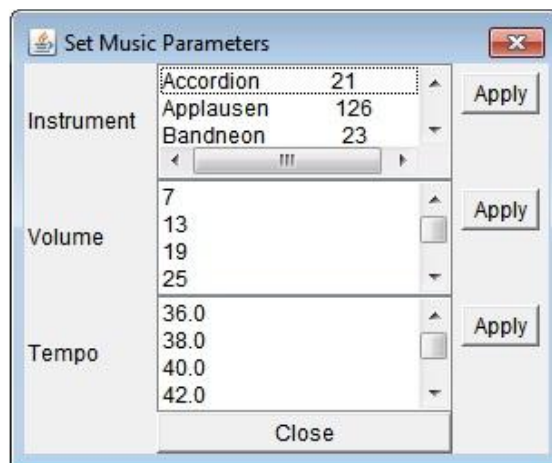


ΕΙΚΟΝΑ 3-4: ΤΟ ΠΑΡΑΘΥΡΟ SET PITCHDETECTOR PARAMETERS

Στο συγκεκριμένο παράθυρο ο χρήστης μπορεί να επιλέξει τις παρακάτω παραμέτρους για τη σωστή λειτουργία του Pitch Detector,

- **Note Duration** – η αξία κάθε νότας που αναγνωρίζεται (επειδή δεν έχει υλοποιηθεί αλγόριθμος που αναγνωρίζει τη διάρκεια κάθε νότας, όλες οι νότες παίρνουν την ίδια αξία, που ορίζει ο χρήστης)
Οι επιλογές είναι Whole Note(Ολόκληρο), Half Note(Μισό), Quarter Note(Τέταρτο), Eighth Note(Όγδοο), Sixteenth Note(Δέκατο έκτο).
- **Pitch Detector Attack Point** – αφορά το ποσοστό που θα πρέπει να έχει μειωθεί η ένταση του σήματος από την προηγούμενη τιμή της, ώστε να θεωρήσει ο αλγόριθμος ότι η νότα σταματάει.
Το εύρος τιμών είναι 0.50 - 1.00, με αύξηση 0.05.
- **Pitch Detector Confidence Threshold** – αφορά την ανώτερη τιμή για την οποία ο αλγόριθμος θεωρεί ότι είναι σωστή η αναγνώριση της συχνότητας.
Το εύρος τιμών είναι 0.50 - 1.00, με αύξηση 0.05.

Επιλέγοντας ο χρήστης της εφαρμογής το **Set Music Parameters**, ανοίγει το παρακάτω παράθυρο,



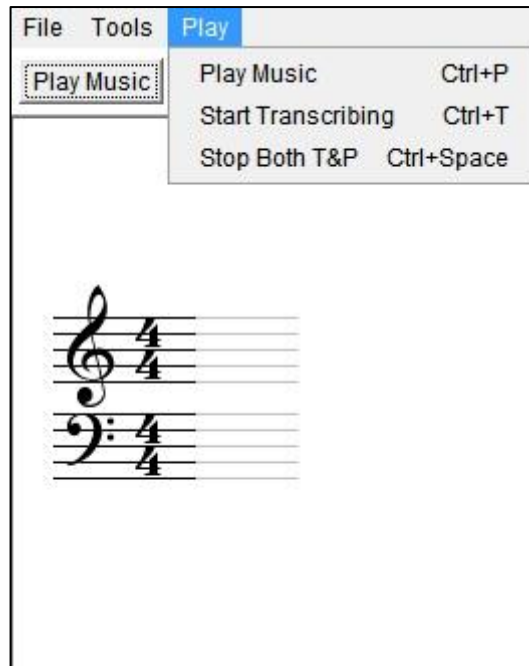
ΕΙΚΟΝΑ 3-5: ΤΟ ΠΑΡΑΘΥΡΟ SET MUSIC PARAMETERS

Στο συγκεκριμένο παράθυρο ο χρήστης μπορεί να επιλέξει,

- **Instrument:** το MIDI όργανο, με το οποίο θα γίνεται η αναπαραγωγή της μουσικής
- **Volume:** την ένταση αναπαραγωγής της μουσικής
- **Tempo:** το τέμπο αναπαραγωγής της μουσικής. Αλλάζοντας σε αυτό το πεδίο το τέμπο, αλλάζει και το τέμπο του μετρονόμου της εφαρμογής

Play

Το μενού Play έχει την παρακάτω μορφή και επιλογές,



ΕΙΚΟΝΑ 3-6: ΤΟ ΜΕΝΟΥ PLAY

Οι επιλογές του μενού Play,

- **Play Music** (συντόμευση από το πληκτρολόγιο Ctrl+P) – γίνεται αναπαραγωγή της παρτιτούρας σε MIDI
- **Start Transcribing** (συντόμευση από το πληκτρολόγιο Ctrl+T) – έναρξη της καταγραφής της ηχητικής ροής από το μικρόφωνο ή το line in της κάρτας ήχου και καταγραφή της στη παρτιτούρα σε πραγματικό χρόνο
- **Stop Both T&P** (συντόμευση από το πληκτρολόγιο Ctrl+Space) – σταματάει την αναπαραγωγή της παρτιτούρας ή την καταγραφή της μουσικής, καθώς και το μετρονόμο

3.1.2 Τα κουμπιά ελέγχου της εφαρμογής

Ο χρήστης της εφαρμογής έχει τη δυνατότητα να ελέγξει τα βασικά μέρη της εφαρμογής, εκτός από το μενού Play, και τις συντομεύσεις από το πληκτρολόγιο, και μέσω των κουμπιών ελέγχου. Τα κουμπιά έχουν την παρακάτω εμφάνιση,



ΕΙΚΟΝΑ 3-7: ΤΑ ΚΟΥΜΠΙΑ ΕΛΕΓΧΟΥ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Τα κουμπιά αυτά ουσιαστικά είναι οι επιλογές του μενού Play, δηλαδή

- **Play Music** – γίνεται αναπαραγωγή της παρτιτούρας σε MIDI
- **Start Transcribing** – έναρξη της καταγραφής της ηχητικής ροής από το μικρόφωνο ή το line in της κάρτας ήχου και καταγραφή της στη παρτιτούρα σε πραγματικό χρόνο
- **Stop Both Transcribing/Music Playback**– σταματάει την αναπαραγωγή της παρτιτούρας ή την καταγραφή της μουσικής, καθώς και το μετρονόμο

Επιπλέον, παράλληλα με τα παραπάνω κουμπιά υπάρχει και ένα toggle box με το οποίο ξεκινάει και σταματάει ο Μετρονόμος.

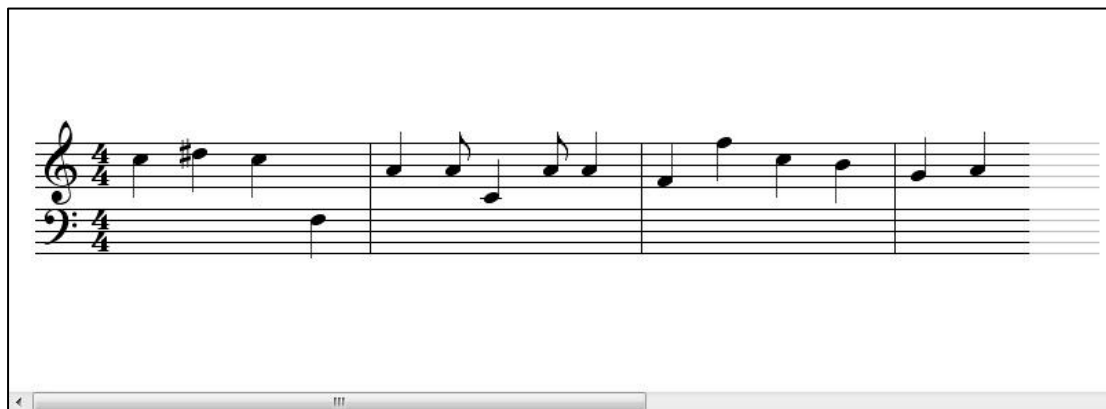
Ο Μετρονόμος υπάρχει για να βοηθάει τη χρήση της εφαρμογής να κρατάει ένα σταθερό τέμπο. Το τέμπο του Μετρονόμου αλλάζει από το παράθυρο Set Music Parameters, του μενού Tools.

Σε περαιτέρω ανάπτυξη της εφαρμογής, αφού υλοποιηθούν αλγόριθμοι εξαγωγής δεδομένων για τα onset των νοτών, ο Μετρονόμος θα χρησιμεύει στην σωστή αναγνώριση των νοτών (και παύσεων) και την κατάτμηση τους στο πεδίο του χρόνου.

3.1.3 Το πεδίο που εμφανίζεται η παρτιτούρα

Η εξαγόμενη μελωδία εμφανίζεται στο πεδίο της παρτιτούρας της εφαρμογής. Η παρτιτούρα αποτελείται από ένα πεντάγραμμο στο κλειδί του Σολ και από ένα στο κλειδί του Φα. Μπορεί να εμφανιστεί μόνο μονοφωνική μελωδία, μίας και η εφαρμογή είναι για μονοφωνική μουσική.

Η εμφάνιση του πεδίου της παρτιτούρας είναι η παρακάτω,



ΕΙΚΟΝΑ 3-8: ΤΟ ΠΕΔΙΟ ΤΗΣ ΠΑΡΤΙΤΟΥΡΑΣ

Στο πεδίο της παρτιτούρας υπάρχει ένα οριζόντιο scroll bar, για να μπορεί να δει ο χρήστης όλη την παρτιτούρα όπως καταγράφεται.

Ο χρήστης μέσω αυτού του πεδίου μπορεί να αλλάξει τα παρακάτω στοιχεία στην παρτιτούρα,

- το μέτρο της παρτιτούρας
- την κλίμακα
- να προσθέσει με το mouse νέες νότες
- να αλλάξει ή να διαγράψει νότες
- να αλλάξει την αξία των νοτών

3.2 Χρήση της εφαρμογής

3.2.1 Καταγραφή παρτιτούρας

Για την καταγραφή της παρτιτούρας, ο χρήστης αφού ανοίξει της εφαρμογή, μπορεί να επιλέξει τις παραμέτρους του PitchDetector, από το παράθυρο Set PitchDetector Parameters (πρόσβαση από το menu Tools → Set PitchDetector Parameters, που παρουσιάστηκε στην παράγραφο 3.1.1, εικόνα 3-4). Αν θέλει μπορεί να παραλείψει αυτό το βήμα, και η καταγραφή θα γίνει με τις default τιμές των παραμέτρων, οι οποίες είναι:

- **Note Duration:** Quarter Note
- **PD Attack Point:** 0.75
- **PD Confidence Threshold:** 0.80

Ο χρήστης ξεκινά την αυτόματη καταγραφή πατώντας το κουμπί Start Transcribing (εναλλακτικά από το menu Play → Start Transcribing ή τη συντόμευση από το πληκτρολόγιο Ctrl+T). Η καταγραφή σταματάει από το κουμπί Stop Transcribing/Music Playback (εναλλακτικά από το menu Play → Stop Both T & P ή τη συντόμευση από το πληκτρολόγιο Ctrl+space). Εάν ο χρήστης σταματήσει την καταγραφή και συνεχίσει ξανά τότε οι νέες νότες θα εμφανιστούν μετά την τελευταία νότα που καταγράφηκε πριν σταματήσει.

Αν κατά την καταγραφή δεν εμφανίζονται καθόλου νότες, σημαίνει ότι η ένταση στο mic in/line in είναι πολύ χαμηλή και θα πρέπει να δυναμώσει.

Ο χρήστης έχει τη δυνατότητα οποιαδήποτε στιγμή να προσθέσει νότες, να διαγράψει ή να αλλάξει νότες στην παρτιτούρα (περισσότερες λεπτομέρειες στην παράγραφο 3.2.3).

3.2.2 Αναπαραγωγή παρτιτούρας

Για την αναπαραγωγή της παρτιτούρας ο χρήστης μπορεί να επιλέξει όργανο MIDI, ένταση αναπαραγωγής και tempo στο παράθυρο Set Music Parameters (πρόσβαση από το menu Tools → Set Music Parameters, που παρουσιάστηκε στην παράγραφο 3.1.1, εικόνα 3-5). Κι εδώ αν θέλει, μπορεί να παραλείψει αυτό το βήμα και η αναπαραγωγή θα γίνει με τις default τιμές των παραμέτρων, οι οποίες είναι:

- **Instrument:** Piano
- **Volume:** 85
- **Tempo:** 60

Η αναπαραγωγή ξεκινά με το κουμπί Play Music (εναλλακτικά από το menu Play → Play Music ή τη συντόμευση από το πληκτρολόγιο Ctrl+P) και σταματά από το κουμπί Stop Transcribing/Music Playback (εναλλακτικά από το menu Play → Stop Both T & P ή τη συντόμευση από το πληκτρολόγιο Ctrl+space). Η αναπαραγωγή γίνεται πάντα από την αρχή της παρτιτούρας.

3.2.3 Επεξεργασία παρτιτούρας

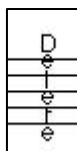
Ο χρήστης έχει τη δυνατότητα να επεξεργαστεί την παρτιτούρα που καταγράφεται ή να δημιουργήσει μια παρτιτούρα προσθέτοντας και αφαιρώντας νότες με το mouse. Επιτρέπεται η επεξεργασία μόνο μονοφωνικής παρτιτούρας.

Για να προσθέσει μια νέα νότα με το mouse πατά (με το αριστερό κουμπί του mouse) στο σημείο της παρτιτούρας που θέλει να την προσθέσει.

Για να αλλάξει το τονικό ύψος μιας νότας πρέπει να επιλέξει τη νότα που επιθυμεί και κρατώντας πατημένο το αριστερό κουμπί του mouse να σύρει τη νότα προς τα πάνω (αν θέλει να αυξήσει το τονικό ύψος), ή προς τα κάτω (αν θέλει να το ελαττώσει).

Για να αλλάξει την αξία μιας νότα πρέπει να επιλέξει τη νότα που επιθυμεί και κρατώντας πατημένο το αριστερό κουμπί του mouse να σύρει τη νότα προς τα αριστερά (αν θέλει να ελαττώσει την αξία της ή να την κάνει παύση) και προς τα δεξιά (αν θέλει να την αυξήσει).

Για να διαγράψει μια νότα πρέπει να επιλέξει τη νότα που επιθυμεί και κρατώντας πατημένο το αριστερό κουμπί του mouse, να σύρει τη νότα προς τα αριστερά ως ότου εμφανιστεί το σύμβολο Delete (βρίσκεται μετά την αξία 16^{ου} και πριν την παύση 16^{ου}).

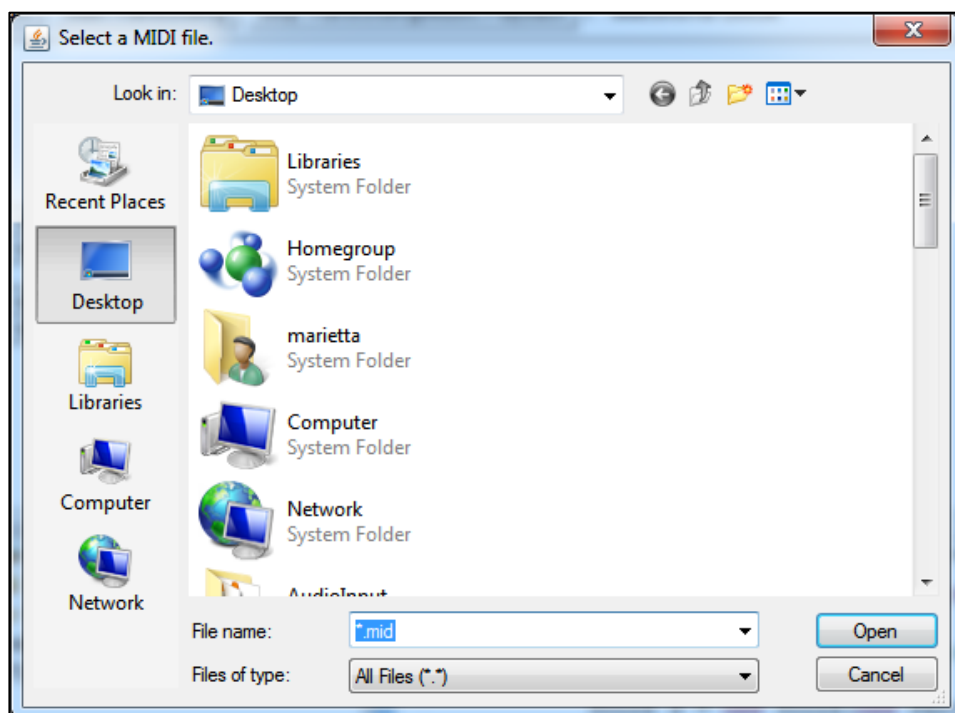


ΕΙΚΟΝΑ 3-9: ΣΥΜΒΟΛΟ DELETE - ΓΙΑ ΤΗ ΔΙΑΓΡΑΦΗ ΝΟΤΑΣ

3.2.4 Εισαγωγή/Εξαγωγή Αρχείου MIDI

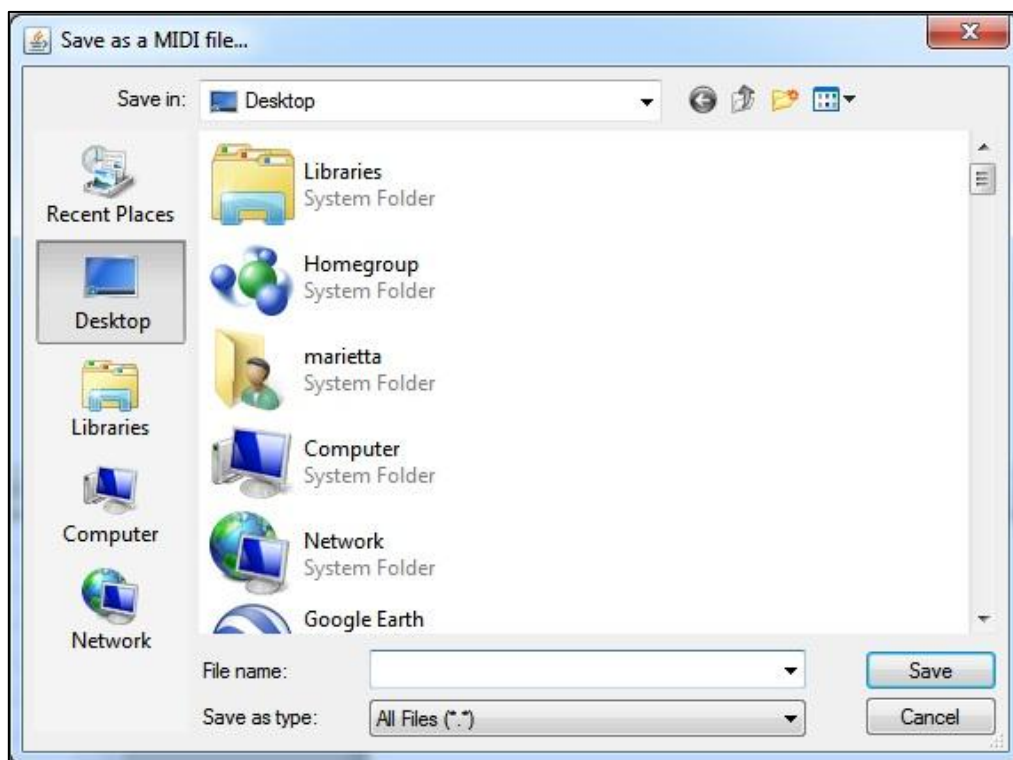
Ο χρήστης έχει τη δυνατότητα να εισάγει μια παρτιτούρα MIDI στην εφαρμογή. Αν εισάγει μια πολυφωνική παρτιτούρα, τότε στην εφαρμογή θα εμφανιστεί μόνο η μελωδία του πρώτου πενταγράμμου.

Για να εισάγει ένα αρχείο MIDI, επιλέγει στο Menu → File το Open MIDI file (εναλλακτικά με τη συντόμευση από το πληκτρολόγιο Ctrl+O). Θα εμφανιστεί το παρακάτω παράθυρο διαλόγου στο οποίο ο χρήστης μπορεί να διαλέξει φάκελο στον οποίο υπάρχει το αρχείο MIDI, και το ίδιο το αρχείο.



ΕΙΚΟΝΑ 3-10: ΠΑΡΑΘΥΡΟ ΔΙΑΛΟΓΟΥ ΓΙΑ ΕΠΙΛΟΓΗ ΑΝΟΙΓΜΑΤΟΣ ΕΝΟΣ MIDI ΑΡΧΕΙΟΥ.

Για την εξαγωγή της παρτιτούρας σε αρχείο MIDI, ο χρήστης επιλέγει στο Menu → File το Save as a MIDI file (εναλλακτικά με τη συντόμευση από το πληκτρολόγιο Ctrl+S). Θα εμφανιστεί και αυτή τη φορά ένα παράθυρο διαλόγου (εικόνα 3-11) στο οποίο μπορεί να ορίσει φάκελο στον οποίο θα αποθηκευτεί το εξαγόμενο αρχείο αλλά και το όνομα του αρχείου. Δεν χρειάζεται να γράψει την κατάληξη .mid, παρά μόνο το όνομα του αρχείου, μιας και την προσθέτει αυτόματα η εφαρμογή.



ΕΙΚΟΝΑ 3-11: ΠΑΡΑΘΥΡΟ ΔΙΑΛΟΓΟΥ ΓΙΑ ΤΗΝ ΕΞΑΓΩΓΗ ΤΗΣ ΠΑΡΤΙΤΟΥΡΑΣ ΣΕ ΑΡΧΕΙΟ MIDI

Ο χρήστης μπορεί να επιλέξει για το αρχείο MIDI που θα αποθηκεύσει τις παραμέτρους όργανο MIDI, ένταση αναπαραγωγής και tempo, στο παράθυρο Set Music Parameters (πρόσβαση από το menu Tools → Set Music Parameters, που παρουσιάστηκε στην παράγραφο 3.1.1, εικόνα 3-5). Μπορεί να παραλείψει αυτό το βήμα και η το εξαγόμενο αρχείο θα έχει τις default τιμές των παραμέτρων, οι οποίες είναι:

- **Instrument:** Piano
- **Volume:** 85
- **Tempo:** 60

4 Υλοποίηση της Εφαρμογής

4.1 Γλώσσα Προγραμματισμού - Εργαλεία Υλοποίησης

Η υλοποίηση της εφαρμογής έγινε με τη γλώσσα προγραμματισμού Java [69]. Επιλέχθηκε η Java γιατί είναι μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού στον κόσμο, μοντέρνα, απλή και σχετικά εύκολη στην εκμάθηση, με μεγάλες δυνατότητες, αντικειμενοστραφής, ανεξάρτητη λειτουργικού συστήματος και πλατφόρμας, και από τις πιο κατάλληλες για την ανάπτυξη εφαρμογών αντίστοιχων με αυτή της πτυχιακής εργασίας.

Η γλώσσα Java είναι μια υψηλού επιπέδου, αντικειμενοστραφής (object oriented) και interpreted γλώσσα προγραμματισμού. Η διερμηνεία του κώδικα σε γλώσσα μηχανής γίνεται από μια εφαρμογή διερμηνέα, η οποία λέγεται Java Virtual Machine (JVM). Η χρήση εικονικής μηχανής για τη διερμηνεία του κώδικα επιτρέπει την εκτέλεση του κώδικα σε οποιαδήποτε πλατφόρμα είναι εφοδιασμένη με ένα JVM, γεγονός που καθιστά τα προγράμματα σε γλώσσα Java ιδιαίτερα μεταφέρσιμα. Ένα επιπλέον χαρακτηριστικό της γλώσσας είναι ότι επιτρέπει την ενσωμάτωση Java εφαρμογών σε ιστοσελίδες. Οι ενσωματωμένες σε ιστοσελίδες εφαρμογές ονομάζονται Applets.

Επίσης, υπάρχουν πολλά δωρεάν πακέτα επέκτασης των δυνατοτήτων της Java σε μορφή βιβλιοθηκών ή κλάσεων (API – Application Programming Interface), όπως αυτά που χρησιμοποιήθηκαν, το JSyn API και το JMusic API. Σημαντικό ρόλο στην επιλογή της γλώσσας έπαιξαν οι γνώσεις που απέκτησα για τη Java, κατά την παρακολούθηση του μαθήματος «Ειδικά Θέματα Μουσικού Προγραμματισμού» του Τμήματος Μουσικής Τεχνολογίας και Ακουστικής του ΤΕΙ Κρήτης.

4.1.1 Το JSyn API

Η προγραμματιστική βιβλιοθήκη που χρησιμοποιήθηκε για την αναγνώριση τονικού ύψους είναι το JSyn API [70]. Το JSyn ανέπτυξε ο Phill Burk, και το παρουσίασε πρώτη φορά στο ICMC (International Computer Music Conference) το 1998. Από εκεί και μετά το JSyn συνεχίζει να αναπτύσσεται με ενημερωμένες εκδόσεις σχεδόν κάθε χρόνο. Η τελευταία έκδοση JSyn Beta 16.4.4 ανακοινώθηκε το Σεπτέμβριο του 2011. Το JSyn διατίθεται δωρεάν για χρήση μη εμπορικών εφαρμογών και ο πηγαίος κώδικας είναι κλειστός.

Προς το παρόν το JSyn υποστηρίζει δύο λειτουργικά συστήματα. Το Microsoft Windows και το Apple Mac OS. Η υποστήριξη για το λειτουργικό Linux θα είναι εφικτή στο άμεσο μέλλον. Οι πρώτες εκδόσεις συνδυάζαν Java με πηγαίο κώδικα σε γλώσσα C, για τη σύνθεση ήχου και τον έλεγχο των μονάδων εισόδου/εξόδου του ήχου, αλλά από το 2010 όλο το API είναι σε Java, με ονομασία Pure Java JSyn .

Το JSyn επιτρέπει την ανάπτυξη διαδραστικών μουσικών προγραμμάτων με την χρήση της γλώσσας προγραμματισμού Java. Μπορεί κάποιος να τρέξει αυτά τα προγράμματα ως ανεξάρτητες εφαρμογές ή σαν applets τα οποία τρέχουν μέσα σε κάποιο web browser. Οι εφαρμογές που μπορεί να γίνουν με το JSyn αφορούν σε ηχητικά εφέ, ηχητικά περιβάλλοντα και μουσική. Είναι βασισμένο στο παραδοσιακό μοντέλο ηχητικών μονάδων - γεννητριών οι οποίες συνδεδεμένες μεταξύ τους μπορούν να παράγουν πιο περίπλοκους ήχους.

Ο κύριος λόγος που επιλέχθηκε το JSyn για την ανάπτυξη της εφαρμογής είναι οι κλάσεις που περιλαμβάνονται στο πακέτο για τον έλεγχο μονάδων εισόδου/εξόδου ήχου και η ήδη υλοποιημένη κλάση που χρησιμοποιείτε στην παρούσα εφαρμογή για την αναγνώριση τονικού ύψους, **PitchDetector** αλλά και η κλάση **PeakFollower** που χρησιμοποιείται το διαχωρισμό μουσικής-θορύβου και την εξαγωγή της διάρκειας των νοτών και των παύσεων. Επίσης, για τον έλεγχο της μονάδα εισόδου γίνεται χρήση της βασικής κλάσης του JSyn, **SynthEngine**. Οι τρεις αυτές κλάσεις, αλλά και η χρήση τους, παρουσιάζονται αναλυτικότερα στην παράγραφο 4.2.

4.1.2 Το JMusic API

Η προγραμματιστική βιβλιοθήκη που χρησιμοποιείται για την απεικόνιση της παρτιτούρας είναι το JMusic API [71]. Το JMusic αναπτύχθηκε από τους Andrew Sorensen και Andrew Brown, και η πρώτη του έκδοση παρουσιάστηκε το 1998. Σχεδιάστηκε με σκοπό να προσφέρει στους μουσικούς και στους προγραμματιστές μια βιβλιοθήκη με εργαλεία μουσικής σύνθεσης και ηχητικής επεξεργασίας. Παρέχει ένα σταθερό πλαίσιο για τη σύνθεση μουσικής με χρήση υπολογιστή σε γλώσσα προγραμματισμού Java, και μπορεί να χρησιμοποιηθεί για τον σχεδιασμό οργάνων, διαδραστικών εφαρμογών και για μουσική ανάλυση.

Μπορεί να χρησιμοποιηθεί εύκολα από μουσικούς γιατί όλη η μουσική δομή των δεδομένων είναι βασισμένη σε ηχητικά δεδομένα νότας και παρτιτούρας, και παρέχει μεθόδους για οργάνωση, επεξεργασία, και ανάλυση αυτών των μουσικών δεδομένων. Η μουσική πληροφορία στο JMusic είναι δομημένη σε μια ιεραρχική δομή βασισμένη στο μοντέλο της μουσικής παρτιτούρας και είναι η παρακάτω,

Score (Παρτιτούρα) (Περιέχει έναν αριθμό από μουσικές Πάρτες)

|

+----**Part (Πάρτα)** (Περιέχει έναν αριθμό από μουσικές Φράσεις)

|

+----**Phrase (Φράση)** (Περιέχει έναν αριθμό από Νότες)

|

+----**Note(Νότα)** (Περιέχει τις πληροφορίες για ένα μοναδικό μουσικό γεγονός)

Note

Η κλάση **jm.music.data.Note** είναι η βασική δομή για τις νότες στο JMusic. Το αντικείμενο Note περιέχει τις παρακάτω πληροφορίες:

- Pitch – η συχνότητα ης νότας
- Dynamic – η ένταση της νότας
- RythmValue – Η διάρκεια της νότας (π.χ ολόκληρο)
- Pan – η θέση της νότας στο πεδίο του χώρου (stereo –LR)
- Duration – η διάρκεια της νότας σε κλάσμα του δευτερολέπτου
- Offset – μια απόκλιση από της “κανονική” έναρξη της νότας

Phrase

Η κλάση **jm.music.data.Phrase** είναι πιο πολύπλοκη από τα αντικείμενα Νότας αλλά μπορεί να περιγραφεί με απλά λόγια ως οι φωνές. Μια πάρτα για πιάνο για παράδειγμα έχει δύο πεντάγραμμα-φράσεις, ένα για κάθε χέρι. Ουσιαστικά ένα Phrase είναι μία λίστα με νότες.

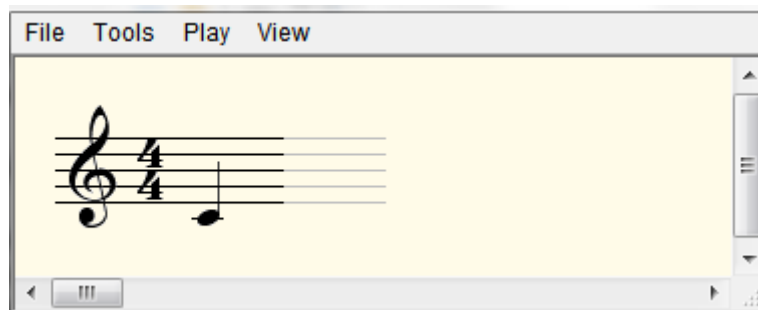
Part

Η κλάση **jm.music.data.Part** περιέχει τις νότες (σε φράσεις) την οποία αναπαράγει ένα όργανο. Μία Πάρτα περιέχει πολλές Μουσικές Φράσεις, έχει μια ονομασία (π.χ. Βιολί 1), ένα κανάλι, και ένα όργανο.

Score

Η κλάση **jm.music.data.Score** είναι η παρτιτούρα. Όπου περιέχει τις Πάρτες (Parts) αλλά έχει και δεδομένα για το μέτρο και την κλίμακα του μουσικού έργου.

Παρακάτω ακολουθεί ένα παράδειγμα κώδικα σε JMusic σε αντιστοιχία με το πρόγραμμα τυπώματος της φράσης "Hello world" ως το πρώτο πρόγραμμα που προτείνεται κατά την εκμάθηση μιας νέας γλώσσας προγραμματισμού. Το πιο απλό πρόγραμμα στην περίπτωση του προτύπου JMusic αποτελεί το 'τύπωμα' ενός μουσικού κομματιού ενός μόνο μέτρου με τη νότα C4 διάρκειας ενός τετάρτου σε ρυθμό 4/4.



ΕΙΚΟΝΑ 4-1: ΜΟΥΣΙΚΗ ΣΗΜΕΙΟΓΡΑΦΙΑ ΣΕ JMusic- ΝΟΤΑ C4, ΑΞΙΑΣ ΤΕΤΑΡΤΟΥ

```
import jm.JMC;
import jm.music.data.Note;
import jm.music.data.Phrase;
import jm.util.View;

public class Dots01 implements JMC {
    public static void main(String[] args) {
        Note n;
        n = new Note(C4, QUARTER_NOTE);
        Phrase phr = new Phrase();
        phr.addNote(n);

        View.notate(phr);
    }
}
```

Οι παρτιτούρες που παράγονται από το JMusic μπορούν να μετατραπούν σε MIDI ή σε αρχεία ήχου για αποθήκευση και περαιτέρω επεξεργασία ή αναπαραγωγή σε πραγματικό χρόνο. Έχει τη δυνατότητα να αναγνωρίζει αρχεία MIDI, αρχεία ήχου, XML, και .jm (αρχεία JMusic). Τέλος, μπορεί να υποστηρίξει JavaSound, QuickTime και MIDIShare σε πραγματικό χρόνο.

Το JMusic διατίθεται δωρεάν και ο κώδικας είναι ελεύθερος (open-source project). Είναι γραμμένο εξολοκλήρου σε Java, και υποστηρίζει λειτουργικά συστήματα και πλατφόρμες Microsoft Windows, Apple Mac OS, Linux, BSD, Solaris ή οποιαδήποτε πλατφόρμα υποστηρίζει Java.

Στην παρούσα εργασία το JMusic χρησιμοποιείται για την μετατροπή της αναγνωρισμένης συχνότητας σε νότα (στο συγκεκριμένο σύστημα), τη δημιουργία και επεξεργασία της παρτιτούρας, στην αναπαραγωγή των αρχείων και η ανάπτυξη του μεγαλύτερου μέρους της εφαρμογής. Περισσότερες λεπτομέρειες για τις κλάσεις που έχουν χρησιμοποιηθεί δίνονται στην παράγραφο 4.2

4.2 Ιεραρχία Τάξεων

Για την υλοποίηση της εφαρμογής δημιουργήθηκαν πέντε τάξεις και μια διεπαφή, τα οποία περιέχονται στο πακέτο `gr.teicrete.mta.m2s`. Οι τάξεις αυτές είναι,

- τάξη `M2SPitchDetector.java`
- τάξη `PDParmFrame.java`
- διεπαφή `NewNoteListener.java`
- τάξη `M2SNote.java`
- τάξη `M2SStave.java`
- τάξη `M2SNotate.java`
- τάξη `Metronome.java`

4.2.1 M2SPitchDetector.java

Η τάξη αυτή αναλαμβάνει τις εξής λειτουργίες:

- Εισαγωγή της ηχητικής ροής από την κάρτα ήχου
- Αναγνώριση τονικού ύψους
- Αναγνώριση διάρκειας νότας (δεν έχει υλοποιηθεί)
- Εισαγωγή του ύψους και της διάρκειας νότας μέσω του interface `NewNoteListener.java` στην τάξη `M2SNote.java`

Στην τάξη αυτή γίνεται χρήση του πακέτου `JSyn`. Παρακάτω ακολουθεί ο κατασκευαστής (constructor) της τάξης.

```
public M2SPitchDetector() {
    //Intialize JSyn, connects and start units.
    synth = new SynthesisEngine();
    synth.start(44100, -1, 2, -1,2);
    synth.add(lineIn = new LineIn());
    lineIn.start();
    synth.add(pitcher = new PitchDetector());
    synth.add(peakFollower = new PeakFollower());
    synth.add(filterLP = new FilterLowPass());
    filterLP.frequency.setValue(filterFreq);
    lineIn.output.connect(filterLP.input);
    filterLP.output.connect(pitcher.input);
    lineIn.output.connect(peakFollower.input);
}
```

```
        peakFollower.halfLife.set(0.01);  
    }
```

- **SynthesisEngine()**- η κεντρική μονάδα του JSyn, η οποία ελέγχει και την είσοδο της κάρτας ήχου,

```
synth.start(44100, -1, 2, -1, 2);
```

όπου παίρνει σαν στερεοφωνική είσοδο, την επιλεγμένη από το λειτουργικό σύστημα.

- **LineIn()** – μέσω αυτής της τάξης, η έξοδος του εξωτερικού σήματος εισόδου (της ηχητικής ροής) περνάει στις εσωτερικές μονάδες της τάξης (PeakFollower() και FilterLowPass()). Το JSyn επιτρέπει την πολλαπλή παράλληλη χρήση του lineIn.
- **PeakFollower()** – παρακολουθεί τις κορυφές του εισαγόμενου σήματος και χρησιμοποιείται για την παρακολούθηση της συνολικής έντασης του σήματος. Στην εφαρμογή χρησιμοποιείται από τον αλγόριθμο για την παρακολούθηση της έντασης του σήματος, για να αναγνωρίζεται πότε μία νότα σταματάει.
- **PitchDetector()** – είναι ο βασικός αλγόριθμος που αναλαμβάνει την αναγνώριση του τονικού ύψους.

Η τεχνική που χρησιμοποιείται για την αναγνώριση του τονικού ύψους είναι σε πειραματικό στάδιο. Σε πρώτο επίπεδο, υπολογίζονται τα **Zero Crossings** και οι τοπικές κορυφές της κυματομορφής. Στη συνέχεια, εφαρμόζεται η τεχνική της **Αυτοσυσχέτισης (Autocorrelation)** στα εξαγόμενα δεδομένα.

Σκοπός του Phil Burk, κατασκευαστή της συγκεκριμένης τάξης, είναι να αποφύγει την υπολογιστική πολυπλοκότητα της πλήρους Αυτοσυσχέτισης, σε επίπεδο δειγμάτων, εξαρχής στην κυματομορφή, ή την πολυπλοκότητα μιας τεχνικής βασισμένης σε FFT ανάλυση. Ένας κεντρικός στόχος του κατασκευαστή, ήταν η δημιουργία ενός αλγορίθμου με τεχνικές στο **πεδίο του χρόνου (Time Domain PD)**, ώστε να υποστηρίζεται αναγνώριση τονικού ύψους σε πραγματικό χρόνο, για μονοφωνικό σήμα, με μικρή καθυστέρηση (latency) για την κιθάρα¹.

Η τάξη M2SPitchDetector.java έχει την κεντρική μέθοδο detectPitch() στην οποία λαμβάνονται τιμές για τις κορυφές τις κυματομορφής αλλά και για το confidence του PitchDetector(). Η υλοποίηση της μεθόδου είναι η εξής,

¹ Οι πληροφορίες για τις τεχνικές που εφαρμόζονται στην τάξη PitchDetector() δόθηκαν από τον Phil Burk, μέσω επικοινωνίας με mail (Σεπτέμβρης 2011).

```

double detectPitch() {
    peakamp = peakFollower.output.getValue();
    conf = pitcher.confidencePort.getValue();

    if(conf>confThld) {
currentFreq = 44100.0 / pitcher.periodPort.getValue();

    }

    if (peakamp > peakFollower.halfLife.getValue()) {
        if(peakamp> attack){
            noteStarted = true;

        }
    }
    if (noteStarted && peakamp <attack*0.5) {
        noteStopped = true;
    }

    if(noteStarted && prevFreq != currentFreq) {
        freq = currentFreq;
        noteStarted = false;
        prevFreq = 0.0;
    }
    else if(noteStopped) {
        freq = currentFreq;
        noteStopped = false;
        noteStarted = false;
    }
    else{
        freq = 0.0;
    }

    return freq;
}

```

Η μέθοδος αυτή, επιστρέφει την συχνότητα(return freq) που έχει αναγνωριστεί από τον αλγόριθμο, αφού πρώτα επιβεβαιωθούν οι συνθήκες από τις μονάδες ελέγχου if().

Στο πρώτο if() λαμβάνεται υπόψη το confidence της τάξης PitchDetector, και έχουμε νέο εξαγόμενο τονικό ύψος μόνο όταν το confidence είναι πάνω από μία τιμή. Η τιμή για το confThld ορίζεται από τη χρήστη στο παράθυρο Set PitchDetector Parameters, δύνοντας στο χρήστη την επιλογή για την τιμή του confidence, πάνω από το οποίο θα επιστρέφει τιμή ο PitchDetector.

Στο δεύτερο if() ελέγχεται αν η ένταση έχει ανέβει από το σημείο σιωπής, και τότε θεωρείτε ότι η νότα έχει ξεκινήσει.

Στο τρίτο if() ελέγχεται πότε η νότα σταματάει. Αυτές οι δύο συνθήκες ουσιαστικά πότε το τονικό ύψος θα περάσει σαν έξοδος της μεθόδου και θα δημιουργηθεί μία νέα νότα.

Στο τέταρτο if(), ελέγχεται αν η παρούσα τιμή που έχει αναγνωρίσει ο PitchDetector είναι διαφορετική από την προηγούμενη αναγνωρισμένη τιμή. Αν είναι διαφορετική, τότε αυτή η τιμή εμφανίζεται το πεντάγραμμα.

Στη συνέχεια υπάρχει άλλη μία if() όπου ελέγχει αν μια νότα έχει σταματήσει, οπότε και πάλι η τιμή freq παίρνει το εξαγόμενο τονικό ύψος currentFreq.

Στην παρούσα τάξη υπάρχει και η μέθοδος run(), η οποία είναι:

```
public void run() {
    double newP = 0.0;
    double newD = 0.0;
    newD = dur;

    while(myLoop != null) {
        newP = detectPitch();

        //caListener.updateConfAmp();
        if(newP > 0) {
            prevFreq = newP;
            noteListener.addNewNote(new M2SNote(newP, newD));
            newP = 0.0;
        }
        try {
            Thread.sleep(sleepTime);
        } catch (InterruptedException ex) {
            ex.printStackTrace();
        }
    }
}
```

Αυτή η μέθοδος είναι ένας βρόγχος, στον οποίο τρέχει η μέθοδος detectPitch(). Ο βρόγχος έχει μία τιμή αναμονής sleepTime,

```
Thread.sleep(sleepTime);
```

Η τιμή για το sleepTime, εκφρασμένη σε millisecond, ελέγχει κάθε πότε ο PitchDetector θα ανανεώνεται. Είναι μία μεταβλητή η οποία εξαρτάται από την αξία νότας (Note Duration) που έχει επιλέξει ο χρήστης στο παράθυρο Set PD Parameters.

Τέλος, στην παρούσα τάξη υπάρχουν και οι μέθοδοι,

- start() – ξεκινά ο Pitch Detector
- stop() – σταματάει ο PitchDetector
- setNoteDuration()- ορίζεται η τιμή για την αξία κάθε νότας και αυτόματα την τιμή για το sleepTime
- setAttackPoint() – ορίζεται η τιμή για το attack point
- setConfThld() – ορίζεται τιμή για το confidence threshold

Οι μέθοδοι start() και stop() καλούνται από την τάξη M2SNotate.java, και οι μέθοδοι setNoteDuration(), setAttackPoint(), setConfThld(), καλούνται από την τάξη setPDParameters.java .

FilterLowPass() – ο αλγόριθμος της τάξης PitchDetector() εξάγει πιο σωστά αποτελέσματα για το τονικό ύψος, όταν προηγηθεί ένα φίλτρο διέλευσης χαμηλών (LowPass Filter), περιορίζοντας τους υψηλούς αρμονικούς. Γι αυτό το λόγο, στην εφαρμογή χρησιμοποιείται ένα τέτοιο φίλτρο με συχνότητα 1800Hz.

4.2.2 PDParmFrame.java

Η τάξη αυτή είναι ένα παράθυρο στο οποίο ο χρήστης, ορίζει τις παραμέτρους που χρειάζεται για τη σωστή λειτουργία του Pitch Detector. Οι παράμετροι αυτοί αναλύθηκαν στο κεφάλαιο 3.1.1 και είναι οι εξής,

- **Note Duration**
- **Pitch Detector Attack Point**
- **Pitch Detector Confidence Threshold**

Η υλοποίηση της τάξης κληρονομεί (extends) την κλάση java.awt.Dialog, το οποίο είναι ένα παράθυρο μέσα στο οποίο έχουν κατασκευαστεί τα απαραίτητα κουμπιά, οι τίτλοι και οι λίστες και οι λειτουργίες τους.

Κατά την ανάπτυξη του αλγορίθμου παρατηρήθηκε ότι η εξαγόμενη παρτιτούρα ήταν πιο σωστή όταν άλλαξε το sleepTime, ανάλογα με την αξία κάθε νότας. Με λίγα λόγια, είναι η τιμή που εκφράζει την ελάχιστη αξία που μπορεί να αναγνωρίσει ο PitchDetector.

4.2.3 NewNoteListener.java

Η NewNoteListener.java είναι μια διεπαφή όπου περιλαμβάνει την μέθοδο,

```
void addNewNote (M2SNote n);
```

Η μέθοδος αυτή υλοποιείται στην τάξη M2SNotate.java υλοποιείται στην τάξη M2SNotate.java και είναι η παρακάτω,

```
public void addNewNote (M2SNote n) {  
    phr.addNote (n);  
    score.addPart (new Part (phr));  
    stave.setPhrase (phr);  
    stave.repaint ();  
}
```

Η παραπάνω υλοποίηση της μεθόδου δηλαδή αναλαμβάνει τα εξής δυο πράγματα,

- να προσθέτει κάθε νέα νότα που καταγράφεται από την τάξη M2SPitchDetector.java στη λίστα με τις νότες (score),
- να εμφανίζεται ως γραφική αναπαράσταση η κάθε νότα στο πεντάγραμμο (stave)

4.2.4 M2SNote.java

Η τάξη αυτή είναι η μήτρα για την δημιουργία των αντικειμένων νότας. Δηλαδή, δέχεται από την τάξη M2SPitchDetector.java τις τιμές της εξαγόμενης συχνότητας και της διάρκειας κάθε νότας. Κληρονομεί (extends) τα στοιχεία και τις μεθόδους της τάξης jm.music.data.Note, του πακέτου JMusic.

Κάθε νότα δημιουργείται στην κλάση M2SPitchDetector.java, μέσω της διεπαφής NewNoteListener.java, όταν ο PitchDetector αναγνωρίζει μια νέα νότα. Αυτό γίνεται με την παρακάτω γραμμή κώδικα,

```
noteListener.addNewNote (new M2SNote (newP, newD));
```


Μέσα σε αυτή την τάξη γίνεται επίσης η μετατροπή της εξαγόμενης συχνότητας στο αντίστοιχο τονικό ύψος της νότας, σε MIDI. Αυτή η μετατροπή γίνεται με τη χρήση της μεθόδου,

```
Convert.getMidiPitchByFrequency((float)newP);
```

Η παραπάνω μέθοδος έχει τη δυνατότητα να βρίσκει το πλησιέστερο τονικό ύψος από την εξαγόμενη συχνότητα, μιας και η συχνότητα μπορεί να μην αντιστοιχεί πάντα σε κάποια νότα του συγκεκριμένου συστήματος.

Λόγω του ότι δεν έγινε υλοποίηση αλγορίθμων για την εξαγωγή της διάρκειας νότας, οι νότες που δημιουργούνται έχουν όλες την ίδια αξία, την οποία επιλέγει ο χρήστης, από το παράθυρο Set PitchDetector Parameters.

4.2.5 M2SStave.java

Η τάξη αυτή αναλαμβάνει τη γραφική αναπαράσταση της παρτιτούρας στο κεντρικό παράθυρο της εφαρμογής. Κληρονομεί (extends) τα στοιχεία και τις μεθόδους της τάξης `jm.gui.cpn.PianoStave`, του πακέτου `JMusic`. Περιέχει δηλαδή, όλες τις πληροφορίες που χρειάζεται η εφαρμογή για την δημιουργία της γραφικής αναπαράστασης της παρτιτούρας.

4.2.6 M2SNotate.java

Η τάξη αυτή αναλαμβάνει όλη τη δημιουργία της κεντρικής γραφικής διεπαφή χρήστη (GUI) της εφαρμογής. Κληρονομεί (extends) την τάξη `java.awt.Frame` και περιέχει όλες τις πληροφορίες για τη δημιουργία των μενού, των κουμπιών και των λειτουργιών της εφαρμογής.

Σε αυτή την κλάση γίνεται η υλοποίηση της μεθόδου `addNewNote()`, της διεπαφή `NewNoteListener.java`, όπως παρουσιάζεται στην παράγραφο 4.2.3.

Επίσης, υλοποιούνται οι μέθοδοι για το άνοιγμα και την αποθήκευση αρχείων καθώς και για την αναπαραγωγή της παρτιτούρας με MIDI. Πιο συγκεκριμένα υλοποιούνται οι μέθοδοι,

- `openMIDI()` – ανοίγει ένα αρχείο MIDI
- `openJM()` – ανοίγει ένα αρχείο `JMusic`
- `openJMXML()` – ανοίγει ένα αρχείο `JMusic XML`
- `saveMIDI()` – αποθηκεύει την παρτιτούρα σε αρχείο MIDI
- `saveJM()` - αποθηκεύει την παρτιτούρα σε αρχείο `JMusic`
- `saveJMXML()` - αποθηκεύει την παρτιτούρα σε αρχείο `JMusic XML`

4.2.7 Metronome.java

Η τάξη αυτή είναι ένας μετρονόμος, δημιουργημένος σε `JMusic`. Αποτελείται από 2 νότες ξυλόφωνου που επαναλαμβάνονται σε ένα συγκεκριμένο τέμπο, το οποίο μπορεί να αλλάξει με τη μέθοδο `changeMetroTempo()`. Ο χρήστης μπορεί να ορίσει το τέμπο από το παράθυρο Set Music Parameters.

Επίσης, υπάρχουν δυο μέθοδοι `start()` και `stop()`, οι οποίες καλούνται στην τάξη `M2SNotate.java` και αποτελούν τα ON/OFF του μετρονόμου μέσω ενός toggle box που

βρίσκεται στο πεδίο με τα κουμπιά ελέγχου της εφαρμογής (παρουσιάζεται στην παράγραφο 3.1.2).

Οι 2 νότες του μετρονόμου είναι η E7 (2637.02Hz), και επιλέχθηκε η συγκεκριμένη νότα γιατί αν ο χρήστης κατά τη διάρκεια την καταγραφής της μουσικής δεν χρησιμοποιήσει ακουστικά (η αν πάρει σαν είσοδο, την έξοδο του ίδιου υπολογιστή που τρέχει η εφαρμογή), θα εμφανίζονται οι νότες του μετρονόμου σαν μέρος της καταγραφόμενης μελωδίας.

5 Πειραματική Αξιολόγηση

5.1 Σύνοψη πειραμάτων

Για την πειραματική αξιολόγηση επιλέχθηκαν να χρησιμοποιηθούν ηχητικές ροές από τέσσερα διαφορετικά μουσικά όργανα, τα οποία είναι,

- Πιάνο
- Φλάουτο
- Βιολί
- Κιθάρα

Η επιλογή αυτή έγινε για να αξιολογηθεί ο αλγόριθμος ως προς την αξιοπιστία του σε κυματομορφές οργάνων με διαφορετική χροιά αλλά και περιβάλλουσα (envelope), και να βρεθούν οι κατάλληλες τιμές για τις παραμέτρους του αλγορίθμου που ορίζει ο χρήστης (Note Duration, Attack Point, PD Confidence Threshold).

Σε όλα πειράματα έγινε αναπαραγωγή ενός αρχείου ήχου μονοφωνικής μελωδίας, όπου με καλώδιο, συνδέθηκε η έξοδος audio out του υπολογιστή στην είσοδο mic in/line in (του ίδιου υπολογιστή).

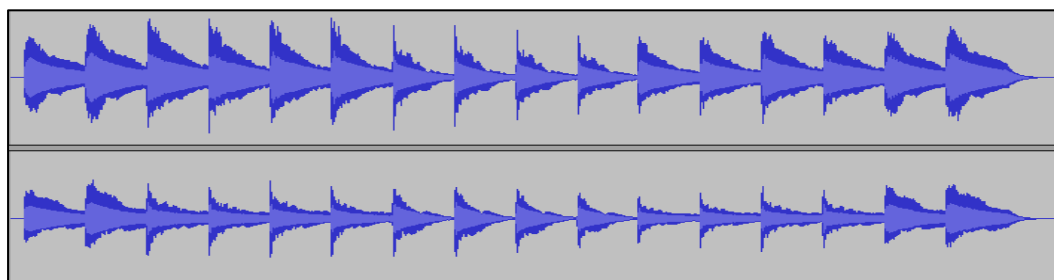
Η παρτιτούρα και το αρχείο ήχου για το πιάνο δημιουργήθηκαν στο Sibelius, και ο ήχος είναι MIDI. Οι παρτιτούρες και τα αρχεία ήχου για το φλάουτο και το βιολί είναι από ηχογραφήσεις που έγιναν με φυσικά όργανα στο studio του Τμήματος Μουσικής Τεχνολογίας και Ακουστικής (ΑΤΕΙ Κρήτης). Η παρτιτούρα και το αρχείο ήχου για την κιθάρα είναι από ηχογράφιση φυσικού οργάνου, η οποία είναι από το διαδίκτυο [72].

5.2 Πείραμα 1 - Πιάνο

Στο Πείραμα 1 βάρος δόθηκε στα διαφορετικά αποτελέσματα που μπορεί να έχουν οι διαφορετικές τιμές για τις παραμέτρους του Pitch Detector, και να εντοπιστούν ποιες είναι οι κατάλληλες για την πιο αξιόπιστη καταγραφή της μελωδίας.

Εξετάζονται δυο μελωδίες για πιάνο. Η πρώτη είναι η κλίμακα του Σολ, και η δεύτερη είναι η ίδια μελωδία που εξετάζεται στο Πείραμα 2 για φλάουτο.

Το αρχείο ήχου που χρησιμοποιήθηκε είναι το piano1.mp3 και η κυματομορφή του είναι,



ΕΙΚΟΝΑ 5-1: ΚΥΜΑΤΟΜΟΡΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ PIANO1.MP3

Η αρχική παρτιτούρα της μελωδίας είναι,

ΕΙΚΟΝΑ 5-2: ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ1.MP3

Καθώς όλες οι νότες είναι τέταρτα, ορίστηκε σαν σταθερή τιμή για το Note Duration το Quarter Note για όλες τις δοκιμές. Οι υπόλοιπες τιμές των παραμέτρων του PitchDetector που ορίστηκαν είναι,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
PD Attack Point	0.75
PD Confidence Threshold	0.80

Η παρτιτούρα που δημιουργήθηκε από την αυτόματη καταγραφή είναι,

ΕΙΚΟΝΑ 5-3: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ1.MP3. ΔΟΚΙΜΗ 1^η.

Από τη σύγκριση των παρτιτούρων, βλέπουμε ότι σε μεγαλύτερο ποσοστό η αναγνώριση έγινε σωστά. Υπάρχουν όμως κάποιες αποκλίσεις οι οποίες είναι,

- Η 2^η νότα της μελωδίας (ΡΕ) δεν αναγνωρίστηκε με επιτυχία
- Η 5^η νότα (ΣΟΛ) εμφανίστηκε δυο φορές
- Το ίδιο και η 10^η νότα (ΛΑ)

Συνολικά από τις 16 νότες που θα έπρεπε να καταγραφούν, τελικά μόνο οι 13 καταγράφηκαν με επιτυχία. Στη συνέχεια αλλάζοντας τις τιμές των παραμέτρων σε,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
PD Attack Point	0.75
PD Confidence Threshold	0.85

Η παρτιτούρα που δημιουργήθηκε είναι,

ΕΙΚΟΝΑ 5-4: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ1.MP3. ΔΟΚΙΜΗ 2^η.

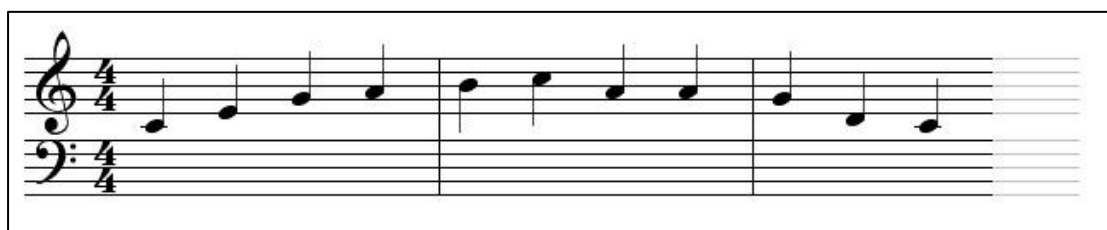
Αυτή τη φορά οι αποκλίσεις είναι,

- Η 8^η νότα (ΝΤΟ) δεν καταγράφηκε με επιτυχία
- Το ίδιο και η 15^η νότα (ΡΕ)

Αλλάζοντας τις τιμές των παραμέτρων σε,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
PD Attack Point	0.75
PD Confidence Threshold	0.90

Η παρτιτούρα που δημιουργήθηκε είναι,



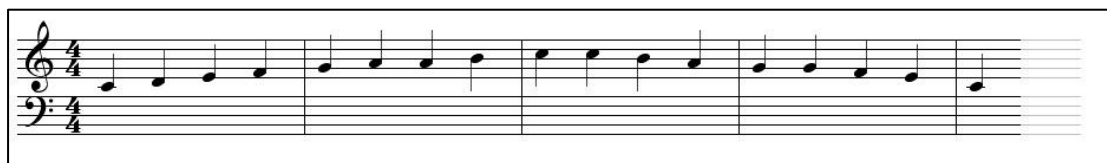
ΕΙΚΟΝΑ 5-5: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ1.MP3. ΔΟΚΙΜΗ 3^η.

Εδώ παρατηρούμε ότι η απόκλιση είναι πολύ μεγαλύτερη, και το αποτέλεσμα χαρακτηρίζεται ανεπιτυχές μιας και οι χαμένες νότες είναι πλέον 7.

Αλλάζοντας πάλι τις τιμές των παραμέτρων σε,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
PD Attack Point	0.80
PD Confidence Threshold	0.80

Η παρτιτούρα που δημιουργήθηκε είναι,



ΕΙΚΟΝΑ 5-6: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ1.MP3. ΔΟΚΙΜΗ 4^η.

Εδώ παρατηρούμε ότι έχουμε πάλι αντίστοιχα αποτελέσματα με τις δυο πρώτες δοκιμές.

Δηλαδή, στο σύνολο η παρτιτούρα που δημιουργήθηκε αυτόματα αντιστοιχεί στο μεγαλύτερο ποσοστό με την αυθεντική. Η αποκλίσεις αυτή τη φορά είναι,

- Η 6^η νότα (ΛΑ) εμφανίστηκε λανθασμένα δυο φορές
- Η 12^η νότα (ΣΟΛ) εμφανίστηκε λανθασμένα δυο φορές
- Η 15^η νότα (ΡΕ) δεν αναγνωρίστηκε

Συνολικά από αυτές τις δοκιμές στην πρώτη μελωδία, η τιμή για το PD Attack Point πρέπει να είναι 0.75 για να έχουμε τα καλύτερα αποτελέσματα. Αν παίρνει τιμές πέρα από το εύρος 0.70-0.80 η εξαγόμενη παρτιτούρα απέχει πάρα πολύ από την αρχική. Επίσης, η παράμετρος του PD Confidence Threshold δίνει καλύτερα αποτελέσματα όταν είναι 0.80 ή 0.85. Επίσης, κι εδώ όταν η τιμή ξεφεύγει από αυτό το εύρος η εξαγόμενη παρτιτούρα δεν είναι σωστή.

Για τη δεύτερη μελωδία για πιάνο, χρησιμοποιήθηκε η μελωδία για Φλάουτο του παρακάτω μουσικού έργου.

by Gabriel Negrin

Flute mf

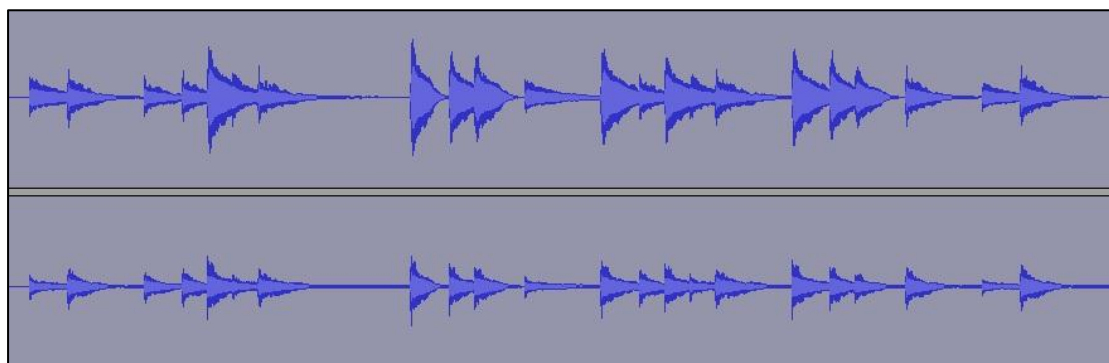
Violin

Fl.

Vln.

ΕΙΚΟΝΑ 5-7: Η ΠΑΡΤΙΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΜΕΡΟΣ ΤΟΥ ΦΛΑΟΥΤΟΥ(ΚΑΙ ΠΙΑΝΟΥ) ΚΑΙ ΤΟΥ ΒΙΟΛΙΟΥ.

Η κυματομορφή του αρχείου ήχου (piano2.mp3) είναι,



ΕΙΚΟΝΑ 5-8: Η ΚΥΜΑΤΟΜΟΡΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ2.MP3

Στις δοκιμές για τη δεύτερη μελωδία, η τιμή Note Duration παρέμεινε ίδια για όλες τις δοκιμές, αξία Ογδού. Οι τιμές των παραμέτρων του PitchDetector που ορίστηκαν είναι,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
PD Attack Point	0.75
PD Confidence Threshold	0.80

Η πρώτη παρτιτούρα είναι η αρχική και η δεύτερη είναι η εξαγόμενη.

ΕΙΚΟΝΑ 5-9: Η ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΑΡΧΕΙΟ ΗΧΟΥ ΡΙΑΝΟ2.MP3

ΕΙΚΟΝΑ 5-10: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ2.MP3. ΔΟΚΙΜΗ 1^η.

Η πρώτη παρατήρηση που μπορεί να γίνει είναι ότι όσο πιο πολύπλοκη η μελωδία (ως προς το τονικό ύψος αλλά και ως προς τις αξίες), τόσο πιο πολύ απέχει η εξαγόμενη παρτιτούρα από την αυθεντική.

Οι αποκλίσεις στις δυο παρτιτούρες εντοπίζονται στα εξής σημεία,

- Η 1^η νότα (ΛΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 2^η νότα (ΡΕ) αναγνωρίστηκε λάθος, καταγράφηκε ως (ΛΑ)
- Εμφανίζεται λανθασμένα η νότα (ΡΕ) (στο πρώτο τρίηχο), χωρίς να υπάρχει στην αρχική μελωδία
- Η 6^η νότα (ΛΑ) αναγνωρίστηκε ένα ημιτόνιο πάνω
- Η 10^η νότα (ΜΙ) εμφανίζεται λανθασμένα δυο φορές
- Η 13^η νότα (ΜΙ) εμφανίζεται λανθασμένα δυο φορές
- Η 14^η νότα (ΝΤΟ) εμφανίζεται λανθασμένα δυο φορές
- Η 17^η νότα (ΣΙ) αναγνωρίστηκε ένα ημιτόνιο κάτω
- Η 18^η νότα (ΡΕ) εμφανίζεται λανθασμένα δυο φορές
- Η 22^η νότα (ΡΕ) δεν καταγράφηκε
- Η 23^η νότα (ΜΙ ύφεση) αναγνωρίστηκε λάθος(ένα τόνο κάτω), καταγράφηκε ως ΦΑ
- Η 24^η νότα (ΛΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 25^η νότα (ΡΕ) θα έπρεπε να εμφανίζεται περισσότερες φορές μιας και είναι αξία Μισό

Μπορούμε να παρατηρήσουμε από τα συνολικά αποτελέσματα, ότι η αναγνώριση των συχνοτήτων έγινε σε μεγάλο βαθμό σωστά (με εξαίρεση 6 νότες), και οι περισσότερες αποκλίσεις εντοπίζονται στις λανθασμένα διπλές νότες.

Μία άλλη παρατήρηση που μπορεί να γίνει είναι ότι οι νότες με αξία μεγαλύτερη του ογδού θα έπρεπε να εμφανίζονται στην εξαγόμενη παρτιτούρα ως επαναλαμβανόμενα όγδοα, κάτι που δεν γίνεται. Αυτό πιθανός να οφείλεται στη χροιά του οργάνου. Δηλαδή, το

sustain των νοτών είναι μικρό, με απότομη πτώση της έντασης μετά το attack. Έτσι, ο Pitch Detector, δεν αναγνωρίζει σωστά τις μεγάλες σε διάρκεια νότες.

Αλλάζοντας τις τιμές των παραμέτρων σε,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
PD Attack Point	0.75
PD Confidence Threshold	0.85

Η πρώτη παρτιτούρα είναι η αρχική και η δεύτερη είναι η εξαγόμενη.

ΕΙΚΟΝΑ 5-11: Η ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΑΡΧΕΙΟ ΗΧΟΥ ΡΙΑΝΟ2.MP3

ΕΙΚΟΝΑ 5-12: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ2.MP3. ΔΟΚΙΜΗ 2^η.

Οι αποκλίσεις στις δυο παρτιτούρες εντοπίζονται στα εξής σημεία,

- Η 7^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 9^η νότα (ΦΑ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 11^η νότα (ΡΕ) δεν αναγνωρίστηκε
- Η 13^η νότα (ΜΙ) δεν αναγνωρίστηκε
- Η 14^η νότα (ΡΕ) δεν αναγνωρίστηκε
- Η 16^η νότα (ΡΕ) δεν αναγνωρίστηκε

Σε αυτή τη δοκιμή παρατηρούμε ότι δεν εμφανίστηκαν λανθασμένα διπλές νότες αλλά 4 δεν αναγνωρίστηκαν καθόλου.

Αλλάζοντας πάλι τις τιμές των παραμέτρων σε,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
PD Attack Point	0.75
PD Confidence Threshold	0.90

Η πρώτη παρτιτούρα είναι η αρχική και η δεύτερη η εξαγόμενη.

ΕΙΚΟΝΑ 5-13: Η ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΑΡΧΕΙΟ ΗΧΟΥ ΡΙΑΝΟ2.MP3

ΕΙΚΟΝΑ 5-14: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ2.MP3. ΔΟΚΙΜΗ 3^η.

Οι αποκλίσεις στις δυο παρτιτούρες εντοπίζονται στα εξής σημεία,

- Η 1^η νότα (ΛΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 5^η νότα (ΦΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 6^η νότα (ΛΑ) αναγνωρίστηκε λάθος (ένα ημιτόνιο πάνω)
- Η 7^η νότα (ΣΙ) δεν αναγνωρίστηκε
- Η 9^η νότα (ΦΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 11^η νότα (ΝΤΟ) δεν αναγνωρίστηκε
- Η 13^η νότα (ΡΕ) εμφανίζεται λανθασμένα δυο φορές
- Η 15^η νότα (ΦΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 16^η νότα (ΛΑ) δεν αναγνωρίστηκε
- Η 18^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 19^η νότα (ΡΕ) εμφανίζεται λανθασμένα δυο φορές
- Η 22^η νότα ()αναγνωρίστηκε λάθος (τρία ημιτόνια πάνω)
- Η 23^η νότα (ΡΕ ύφεση) αναγνωρίστηκε λάθος (ΝΤΟ δίεση πάνω)

Και σε αυτή τη δοκιμή παρατηρούμε ότι από τις συνολικά 24 νότες, 3 δεν αναγνωρίστηκαν καθόλου, 4 αναγνωρίστηκαν ως διαφορετικές νότες και 6 εμφανίστηκαν λανθασμένα 2 φορές. Το ότι κάποιες νότες δεν αναγνωρίστηκαν καθόλου οφείλεται μάλλον στο γεγονός ότι το PD Confidence Threshold, ορίστηκε πιο πάνω από όσο θα έπρεπε.

Αλλάζοντας τις τιμές των παραμέτρων σε,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
PD Attack Point	0.65
PD Confidence Threshold	0.85

Η πρώτη παρτιτούρα είναι η αρχική και η δεύτερη η εξαγόμενη.

Musical score for piano. The top staff is the original score, and the bottom staff is the extracted part. The tempo is marked as quarter note = 80. The key signature has one flat (B-flat). The time signature is 5/8. The score consists of 8 measures. The first four measures contain triplets of eighth notes. The fifth measure contains a quarter note and a quarter rest. The sixth measure contains a quarter note and a quarter rest. The seventh measure contains a quarter note and a quarter rest. The eighth measure contains a quarter note and a quarter rest.

ΕΙΚΟΝΑ 5-15: Η ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΑΡΧΕΙΟ ΗΧΟΥ ΡΙΑΝΟ2.MP3

Musical score for piano, showing the extracted part. The score consists of 8 measures. The first four measures contain triplets of eighth notes. The fifth measure contains a quarter note and a quarter rest. The sixth measure contains a quarter note and a quarter rest. The seventh measure contains a quarter note and a quarter rest. The eighth measure contains a quarter note and a quarter rest.

ΕΙΚΟΝΑ 5-16: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ ΡΙΑΝΟ2.MP3. ΔΟΚΙΜΗ 4^η.

Οι αποκλίσεις στις δυο παρτιτούρες εντοπίζονται στα εξής σημεία,

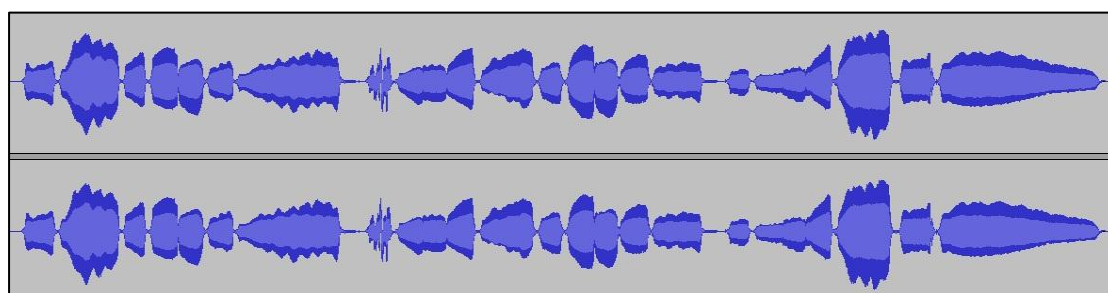
- Η 1^η νότα (ΛΑ) εμφανίζεται λανθασμένα δυο φορές
- Εμφανίζεται λανθασμένα η νότα (ΦΑ)
- Η 6^η νότα (ΛΑ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 7^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 10^η νότα (ΡΕ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 13^η νότα (ΡΕ) εμφανίζεται λανθασμένα δυο φορές
- Η 15^η νότα (ΦΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 17^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)

Σε αυτή τη δοκιμή παρατηρούμε ότι τα σφάλματα και οι χαμένες νότες είναι λιγότερες αλλά αναγνωρίζονται πολλές από τις νότες με ένα ημιτόνιο διαφορά. Αυτό πιθανώς να οφείλετε στο γεγονός ότι η τιμή PD Attack point είναι πολύ χαμηλή, άρα η τιμή που λαμβάνεται υπόψη για το τονικό ύψος είναι νωρίς στο attack της νότας.

5.3 Πείραμα 2 – Φλάουτο

Για το δεύτερο πείραμα χρησιμοποιήθηκε η ίδια μελωδία με αυτή του αρχείου ήχου riano2.mp3, αυτή τη φορά από φλάουτο (ηχογράφηση φυσικού οργάνου).

Η κυματομορφή του αρχείου ήχου (flute.mp3) είναι,



ΕΙΚΟΝΑ 5-17: ΚΥΜΑΤΟΜΟΡΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ FLUTE.MP3

Οι τιμές των παραμέτρων του PitchDetector που ορίστηκαν είναι,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
Note Duration	Eighth Note
PD Attack Point	0.75
PD Confidence Threshold	0.85

Η πρώτη παρτιτούρα είναι η αρχική και η δεύτερη (εικόνες 2 και 3) η εξαγόμενη.

ΕΙΚΟΝΑ 5-18: Η ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΑΡΧΕΙΟ ΗΧΟΥ FLUTE.MP3

ΕΙΚΟΝΑ 5-19: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ FLUTE.MP3. ΔΟΚΙΜΗ 1^η.

Η πρώτη παρατήρηση που μπορεί να γίνει, είναι ότι έχουν δημιουργηθεί πολύ περισσότερες νότες, σε σχέση με το πείραμα για την αντίστοιχη μελωδία με πιάνο. Αυτό οφείλεται στο γεγονός ότι το διάστημα του sustain κάθε νότας από το φλάουτο είναι πολύ μεγαλύτερο σε διάρκεια και σταθερό. Έτσι, ο Pitch Detector, αναγνωρίζει τη νότα σε όλη τη διάρκειά της.

Οι αποκλίσεις στις δυο παρτιτούρες εντοπίζονται στα εξής σημεία,

- Η 1^η νότα (ΛΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 2^η νότα (ΡΕ) εμφανίζεται 4 φορές αντί για 2 (2 όγδοα ισούνται με ένα τέταρτο, που είναι και η αξία της αρχικής νότας)
- Η 3^η νότα (ΝΤΟ) εμφανίζεται λανθασμένα δυο φορές
- Η 4^η νότα (ΡΕ) εμφανίζεται λανθασμένα δυο φορές
- Η 7^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 9^η νότα (ΜΙ) εμφανίζεται λανθασμένα δυο φορές
- Η 10^η νότα (ΡΕ) εμφανίζεται λανθασμένα δυο φορές
- Η 15^η νότα (ΦΑ) εμφανίζεται λανθασμένα δυο φορές
- Η 17^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 20^η νότα εμφανίζεται λανθασμένα δυο φορές
- Η 22^η νότα εμφανίζεται λανθασμένα δυο φορές

Κι σε αυτό το πείραμα, το κυρίως πρόβλημα είναι η διπλή λανθασμένη εμφάνιση των νोटών.

Στη συνέχεια άλλαξε η τιμή για το Note Duration και οι άλλες δυο παράμετροι έμειναν ίδιες,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
Note Duration	Quarter Note
PD Attack Point	0.75
PD Confidence Threshold	0.85

Η πρώτη παρτιτούρα είναι η αρχική και η δεύτερη, η εξαγόμενη.

The image shows a musical score for a piano piece. It is in 5/8 time and has a tempo of 80. The score consists of two staves: a treble clef staff and a bass clef staff. The treble staff contains a melody with several triplet markings (indicated by a '3' over a group of notes). The bass staff contains rests for the first four measures, followed by a few notes in the fifth measure. The word 'PIANO' is written to the left of the staves.

ΕΙΚΟΝΑ 5-20: Η ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΑΡΧΕΙΟ ΗΧΟΥ FLUTE.MP3

The image shows a musical score for a flute piece. It is in 5/4 time. The score consists of two staves: a treble clef staff and a bass clef staff. The treble staff contains a melody with various note values and accidentals. The bass staff contains rests for the first four measures, followed by a few notes in the fifth measure.

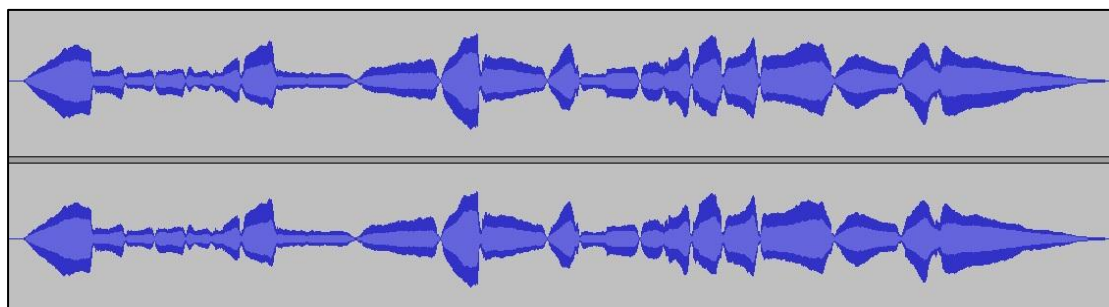
ΕΙΚΟΝΑ 5-21: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ FLUTE.MP3. ΔΟΚΙΜΗ 2^η.

Στην περίπτωση που ο χρήστης της εφαρμογής επιλέξει Note Duration, μεγαλύτερο από τις αξίες που υπάρχουν στην μεταγραφόμενη μελωδία, παρατηρούμε ότι πολλές νότες δεν αναγνωρίζονται καθόλου. Αυτό οφείλεται στο γεγονός ότι για διαφορετικές τιμές του Note Duration, η ο χρόνος που ο Pitch Detector παίρνει τιμές για να τις αναγνωρίσει είναι διαφορετικός. Θα μπορούσε ο χρόνος αυτός να είναι σταθερός και πολύ μικρός (πχ της τάξης των 100ms), αλλά τότε ο Pitch Detector ανανεώνεται τόσο συχνά που αναγνωρίζονται διπλές (και πολλαπλές) οι νότες, με αποτέλεσμα η παρτιτούρα που δημιουργείται να μην είναι σωστή.

5.4 Πείραμα 3 – Βιολί

Για το τρίτο πείραμα χρησιμοποιήθηκε μελωδία με βιολί (ηχογράφηση φυσικού οργάνου), η οποία είναι από το ίδιο έργο με τη μελωδία για φλάουτο.

Η κυματομορφή του αρχείου ήχου (violin.mp3) είναι,

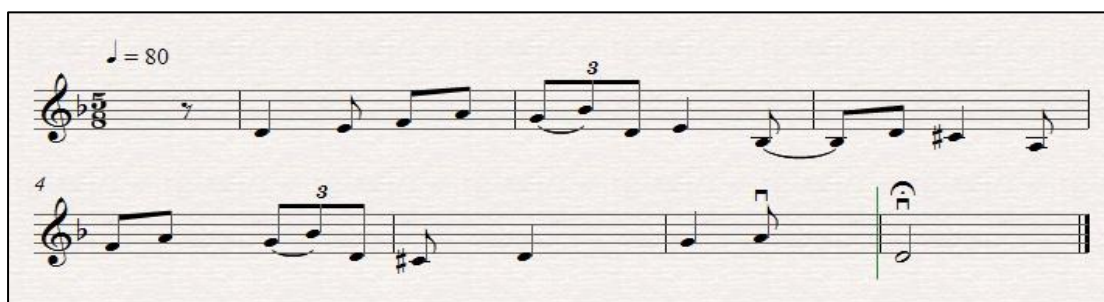


ΕΙΚΟΝΑ 5-22: ΚΥΜΑΤΟΜΟΡΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ VIOLIN.MP3

Οι τιμές των παραμέτρων του PitchDetector που ορίστηκαν είναι,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
Note Duration	Eighth Note
PD Attack Point	0.75
PD Confidence Threshold	0.85

Η πρώτη παρτιτούρα είναι η αρχική και η δεύτερη (εικόνας 2 και 3) η εξαγόμενη.



ΕΙΚΟΝΑ 5-23: Η ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΑΡΧΕΙΟ ΗΧΟΥ VIOLIN.MP3

ΕΙΚΟΝΑ 5-24: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ VIOLIN.MP3.

Και στην αναγνώριση του τονικού ύψους σε μελωδία από βιολί, βλέπουμε ότι το αποτέλεσμα είναι σε γενικές γραμμές ικανοποιητικό.

Μια παρατήρηση κοιτώντας αρχικά τις 2 παρτιτούρες είναι ότι έχουν αναγνωριστεί κάποιες ψηλές νότες, οι οποίες δεν υπάρχουν στην αρχική παρτιτούρα. Αυτό οφείλεται στο γεγονός

ότι καθώς παίζει ο μουσικός βιολί, μπορεί να παραχθούν κάποιες υψηλές συχνότητες που δεν είναι θεμέλιες.

Πιο αναλυτικά, οι αποκλίσεις στις δυο παρτιτούρες εντοπίζονται στα εξής σημεία,

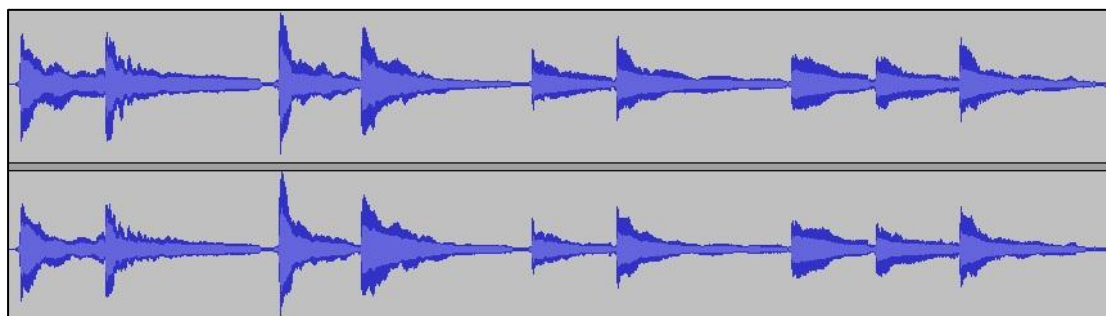
- Η 1^η νότα (ΡΕ) καταγράφεται ως 5 όγδοα αντί για 4
- Η 3^η νότα (ΦΑ) αναγνωρίστηκε ως ΛΑ
- Η 6^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 8^η νότα (ΜΙ) αναγνωρίστηκε λάθος (ένα τόνο κάτω)
- Η 9^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 10^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 12^η νότα (ΝΤΟ δίεση) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 14^η νότα (ΦΑ) δεν αναγνωρίστηκε
- Καταγράφηκε η νότα ΝΤΟ (πάνω από το πεντάγραμμα) χωρίς να υπάρχει στην αρχική παρτιτούρα
- Η 17^η νότα (ΣΙ) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Η 19^η νότα (ΝΤΟ δίεση) αναγνωρίστηκε λάθος (ένα ημιτόνιο κάτω)
- Καταγράφηκε η νότα ΡΕ χωρίς να υπάρχει στην αρχική παρτιτούρα
- Καταγράφηκε ξανά η νότα ΝΤΟ (πάνω από το πεντάγραμμα) χωρίς να υπάρχει στην αρχική παρτιτούρα

Πέρα από της υψηλές νότες που εμφανίστηκαν, έχουμε πάλι αντίστοιχες αποκλίσεις με τα προηγούμενα πειράματα. Δηλαδή, λάθος αναγνώριση του τονικού ύψους (κυρίως απόκλιση ενός ημιτονίου) και περισσότερες (διπλές) νότες.

5.5 Πείραμα 4 – Κιθάρα

Για το τέταρτο πείραμα χρησιμοποιήθηκε μελωδία με κιθάρα (ηχογράφηση φυσικού οργάνου).

Η κυματομορφή του αρχείου ήχου (guitar.mp3) είναι,



ΕΙΚΟΝΑ 5-25: ΚΥΜΑΤΟΜΟΡΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ GUITAR.MP3

Οι τιμές των παραμέτρων του PitchDetector που ορίστηκαν είναι,

Παράμετρος Pitch Detector	Τιμή Παραμέτρου
Note Duration	Half Note
PD Attack Point	0.75
PD Confidence Threshold	0.85

Το 8 κάτω από το κλειδί του Σολ, υποδεικνύει ότι οι νότες της παρτιτούρας ηχούν μία οκτάβα χαμηλότερα. Οπότε αναμένεται μετά την καταγραφή μέσω της εφαρμογής, όλες οι νότες θα εμφανίζονται στο κλειδί του Φα.

Η πρώτη παρτιτούρα είναι η αρχική και η δεύτερη, η εξαγόμενη.



ΕΙΚΟΝΑ 5-26: Η ΠΑΡΤΙΤΟΥΡΑ ΑΝΑΦΟΡΑΣ ΓΙΑ ΤΟ ΑΡΧΕΙΟ ΗΧΟΥ GUITAR.MP3



ΕΙΚΟΝΑ 5-27: Η ΕΞΑΓΟΜΕΝΗ ΠΑΡΤΙΤΟΥΡΑ ΑΠΟ ΤΗΝ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΑΡΧΕΙΟΥ ΗΧΟΥ GUITAR.MP3.

Όπως ήταν αναμενόμενο, οι νότες είναι μια οκτάβα κάτω. Παρατηρούμε, όμως, ότι όλες οι νότες είναι ένα τόνο ή ένα ημιτόνιο κάτω. Αναλυτικά,

- Η 1^η νότα (ΣΙ) αναγνωρίστηκε ως ΛΑ
- Η 2^η νότα (ΛΑ δίεση) αναγνωρίστηκε ως ΣΟΛ δίεση
- Η 3^η νότα (ΛΑ) αναγνωρίστηκε ως ΣΟΛ δίεση
- Η 4^η νότα (ΣΟΛ δίεση) αναγνωρίστηκε ως ΦΑ δίεση
- Η 5^η νότα (ΣΟΛ) αναγνωρίστηκε ως ΦΑ δίεση
- Η 6^η νότα (ΦΑ) αναγνωρίστηκε ως ΜΙ
- Η 7^η νότα (ΡΕ) αναγνωρίστηκε σωστά
- Η 8^η νότα (ΜΙ) αναγνωρίστηκε ως ΝΤΟ και εμφανίζεται λανθασμένα διπλή
- Η 9^η νότα (ΦΑ) αναγνωρίστηκε ως ΜΙ

5.6 Σύγκριση Αποτελεσμάτων

Σε γενικές γραμμές, βλέπουμε ότι η εφαρμογή λειτουργεί ικανοποιητικά και η αναγνώριση του τονικού ύψους γίνεται με επιτυχία.

Από τα αποτελέσματα των πειραμάτων διαπιστώνουμε ότι υπάρχουν τρεις διαφορετικές κατηγορίες προβλημάτων κατά την αυτόματη αναγνώριση τονικού ύψους και την δημιουργία παρτιτούρας, οι οποίες είναι οι εξής:

- Μη αναγνώριση του νότας
- Λανθασμένη αναγνώριση τονικού ύψους (κυρίως διαφορά ημιτονίου)
- Εμφάνιση διπλής νότας

Όσον αφορά την αναγνώριση του τονικού ύψους για διαφορετικά μουσικά όργανα διαπιστώσαμε ότι όσο πιο απλή είναι η χροιά ενός οργάνου, τόσο πιο καλά λειτουργεί η εφαρμογή. Η χροιά ενός οργάνου, μπορεί να περιγραφεί από την σχέση μεταξύ των φασματικών συστατικών του, τη σύνθεση των αρμονικών. Τόσο το πλήθος, όσο και η διαφορετικότητα των αρμονικών που προστίθενται στην θεμελιώδη συχνότητα, επηρεάζουν την ποιότητα του τόνου, τη φωτεινότητα και την έντασή του.

Βλέπουμε, για παράδειγμα, ότι το φλάουτο, έχει σχετικά ένα απλό και σταθερό φάσμα, και το sustain στην περιβάλλουσα διαρκεί πολύ και είναι σταθερό. Έτσι, το αποτέλεσμα της αναγνώρισης του τονικού ύψους είναι πιο κοντά στην αρχική παρτιτούρα. Αντίθετα, το φάσμα του βιολιού είναι πιο πλούσιο (με πολλές αρμονικές). Επίσης, είναι ένα όργανο χωρίς τάστα και κατά την μουσική εκτέλεση η ένταση δεν παραμένει σταθερή. Έτσι, η αναγνώριση του τονικού ύψους είναι πιο δύσκολη και εμφανίζει πολλά λάθη.

Συμπερασματικά, όσο πιο καθαρή, σταθερή και απλή φασματικά είναι μια μελωδία, τόσο πιο καλά αποτελέσματα έχουμε. Επίσης, είναι πιο εύκολο να αναγνωριστεί μια απλή μελωδία, με χαμηλό τέμπο και μεγάλες αξίες.

6 Συμπεράσματα

6.1 Σύνοψη

Ο αρχικός στόχος της παρούσας πτυχιακής εργασίας ήταν η δημιουργία μιας εφαρμογής με την οποία θα γίνεται αυτόματα δημιουργία παρτιτούρας, από μονοφωνικές ηχητικές ροές, σε πραγματικό χρόνο.

Κάτι τέτοιο όμως, αποδείχτηκε ότι είναι ένα πολύ πιο πολύπλοκο θέμα, με σημαντικές προκλήσεις και δυσκολίες, περισσότερες από ότι είχε αρχικά εκτιμηθεί.

Όπως αναλύθηκε στην παράγραφο 2.1, για να δημιουργηθεί αυτόματα μία μουσική παρτιτούρα από ηχητική ροή, πρέπει να επιλυθούν κάποια υπο-προβλήματα. Αυτά ενδεικτικά είναι:

- Διαχωρισμός μουσικής/θορύβου (όπου ως θόρυβος νοείται οτιδήποτε δεν είναι μουσική)
- Ανίχνευση παύσεων
- Αναγνώριση τονικού ύψους
- Ανίχνευση αρχής νότας
- Εκτίμηση διάρκειας νότας

Για την επίλυσή τους δεν επαρκεί η αξιοποίηση αλγορίθμων αναγνώρισης μόνο του τονικού ύψους, όπως ήταν ως στόχος της πτυχιακής. Για να γίνει εξαγωγή όλων των απαραίτητων δεδομένων πρέπει να αξιοποιηθούν και αλγόριθμοι αναγνώρισης της αρχής (onset) και της διάρκειας κάθε νότας (note segmentation).

Επίσης, η καθεαυτή αξιοποίηση και ανάπτυξη του αλγορίθμου για την αναγνώριση του τονικού ύψους αποδείχτηκε ένα εγχείρημα με πολλές δυσκολίες.

Παρόλες τις δυσκολίες, η εφαρμογή λειτουργεί σε ένα μεγάλο βαθμό με επιτυχία. Δηλαδή γίνεται αναγνώριση του τονικού ύψους, ειδικά όταν έχουν οριστεί οι παράμετροι του αλγορίθμου στα αντίστοιχα σωστά εύρη τιμών.

Η επιλογή των πακέτων JSyn και JMusic ήταν σωστή και αποτελεσματική για τη δημιουργία της εφαρμογής, καθώς και η χρήση της τάξης PitchDetector.java.

Το πιο σημαντικό κομμάτι, αυτό της αναγνώρισης του τονικού ύψους, θα μπορούσε να υλοποιηθεί καλύτερα, για να μην εμφανίζονται τα προβλήματα και η αποκλίσεις που παρουσιάστηκαν κατά την πειραματική αξιολόγηση.

Με τη μελέτη της συναφούς βιβλιογραφίας, συγκεκριμένα τους αλγόριθμους αναγνώρισης τονικού ύψους, αλλά και γενικά του ερευνητικού χώρου της Ανάκτησης Μουσικής Πληροφορίας (MIR - Music Information Retrieval), διαπιστώθηκε ότι για την δημιουργία ενός αυτόνομου και "έξυπνου" συστήματος καταγραφής μουσικής από ηχητικές ροές, σε πραγματικό χρόνο, απαιτούνται συνδυαστικές γνώσεις και οι εφαρμογές τους από διαφορετικά αντικείμενα μελέτης του MIR.

6.2 Μελλοντικές Επεκτάσεις

Πιθανές μελλοντικές επεκτάσεις αφορούν τη βελτίωση του αλγορίθμου αναγνώρισης του τονικού ύψους, ώστε να είναι πιο αξιόπιστος και αυτόνομος. Δηλαδή, να μην επηρεάζεται από την ένταση του σήματος σε τόσο μεγάλο βαθμό) και να μην χρειάζεται ο χρήστης να ορίζει τιμές για τις παραμέτρους του.

Επίσης, μπορούν να επεκταθούν οι υλοποιημένοι αλγόριθμοι ώστε να αναγνωρίζονται οι παύσεις, η έναρξη και η αξία κάθε νότας έτσι ώστε να έχουμε πλήρη και σωστή καταγραφή της μελωδίας και όχι μόνο του τονικού ύψους και να εφαρμοστούν αλγόριθμοι για την εξαγωγή αυτών των χαρακτηριστικών.

Μια άλλη λειτουργία που μπορεί να έχει η εφαρμογή ως προς την επεξεργασία της παρτιτούρας είναι η συγχώνευση πολλαπλών νοτών σε μία νότα, μία λειτουργία η οποία είναι διαθέσιμη σε άλλα εμπορικά προγράμματα.

Τέλος, μπορεί να παρέχεται η δυνατότητα εξαγωγής της παρτιτούρας σε μορφή αρχείου MusicXML, μιας και είναι πλέον το πιο διαδεδομένο πρότυπο ανταλλαγής μουσικής παρτιτούρας, καθώς και σε μορφή αρχείου PDF.

7 Παραπομπές

- [1] Music Information Retrieval – Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Music_information_retrieval (τελευταία πρόσβαση 3 Σεπτεμβρίου 2011)
- [2] Byrd, D. A. *Music Notation by Computer*. PhD thesis Department of Computer Science, Indiana University, August 1984. (διαθέσιμο @<http://www.informatics.indiana.edu/donbyrd/>)
- [3] Downie, J. S. *Music information retrieval* (Chapter 7). In Annual Review of Information Science and Technology 37, ed. Blaise Cronin, 295-340. Medford, NJ: Information Today, 2003. (διαθέσιμο @http://music-ir.org/downie_mir_arist37.pdf)
- [4] Klapuri, A. *Automatic Transcription of Music, Master of Science Thesis*. Department of Information Technology, Tampere University of Technology, April 1998
- [5] Klapuri, Ansi. *Automatic Music Transcription as we know it today*. Journal of New Music Research, Vol. 33 No. 3, pp. 269-282, Σεπτέμβρης 2004. (διαθέσιμο @http://www.cs.tut.fi/sgn/arg/klap/jnmr_klapuri.pdf)
- [6] Moorer, J.A. *On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer*. PhD theses, Department of Music, Stanford University, 1975. (διαθέσιμο @<https://ccrma.stanford.edu/files/papers/stanm3.pdf>)
- [7] Klapuri A, Davy M. *Signal processing methods for music transcription*. Springer, New York, 2006. (διαθέσιμο @<http://www.cs.tut.fi/~klap/amt.html>)
- [8] Moorer, J.A. *On the Transcription of Musical Sound by Computer*. Computer Music Journal, Vol. 1, No. 4 (November 1977), pp. 32-38 1977. (διαθέσιμο @<http://www.jstor.org/stable/40731298>)
- [9] Chafe C., Kashima J., Mont-Reynaud B., Smith J. *Techniques for Note Identification in Polyphonic Music*. Center for Computer Research in Music (CCRMA), Department of Music, Stanford University, Οκτώβρης 1985. (διαθέσιμο @<https://ccrma.stanford.edu/files/papers/stanm29.pdf>)
- [10] Piszczalski M. *A Computational Model of Music Transcription*. PhD theses, University of Michigan Ann Arbor, 1986.
- [11] Maher R.C. *An Approach for the Separation of Voices in Composite Music Signals*. PhD thesis, University of Illinois. Urbana, 1989. (διαθέσιμο @ http://www.coe.montana.edu/ee/rmaher/publications/maher_uiuc_dissertation_04_89.pdf)
- [12] Goto M., Muraoka Y. *A Beat Tracking System for Acoustic Signals of Music*. ACM International Conference in Multimedia, pp. 365-372, San Francisco, 1994. (διαθέσιμο @<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.9977>)
- [13] Schloss W.A. *On the Automatic Transcription of Percussive Music – from Acoustic Signal to High-Level Analysis*. PhD thesis, Center for Computer Research in Music and Acoustics, Stanford University, 1985. (διαθέσιμο @<https://ccrma.stanford.edu/files/papers/stanm27.pdf>)
- [14] Bilmes J.A. *Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm*. Master's thesis, Massachusetts Institute of Technology, 1993. (διαθέσιμο @ <http://ssli.ee.washington.edu/~bilmes/pgs/b2hd-bilmes1993-mit-thesis.html>)
- [15] Goto M., Muraoka Y. *A sound source separation system for percussion instruments*. Transactions of the Institute of Electronics, Information and Communication Engineers D-II, J77-D-II(5):901-911 (in Japanese), 1994.

- [16] Tanguiane A.S. *Artificial Perception and Music Recognition*. Springer, Berlin Heidelberg, 1993. (διαθέσιμο @ <http://www.springerlink.com/home/main.mpx>)
- [17] Kashino K., Nakadai K., Kinoshita T., Tanaka H. *Organisation of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism*. International Joint Conference on Artificial Intelligence, pp. 158-164, Montreal, Quebec, 1995. (διαθέσιμο @ <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.538>)
- [18] Goto M. *A predominant-F estimation method for real-world musical audio signals: Map estimation for incorporating prior knowledge about F0s and tone models*. Proc. Workshop on Consistent and Reliable Acoustic Cues for Sound Analysis, Aalborg, Denmark, 2001. (διαθέσιμο @ www.ee.columbia.edu/crac/papers/goto.pdf)
- [19] Davy M. and Godsill S. *Bayesian harmonic models for musical signal analysis*. Seventh Valencia International meeting Bayesian statistics 7, Tenerife, Spain, 2002.
- [20] Ryyanen, Klapuri A. *Polyphonic music transcription using note event modeling*. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA, 2005.
- [21] Cemgil A.T. and Kappen B. *Monte Carlo methods for tempo tracking and rhythm quantization*. Journal of Artificial Intelligence Research, 18:45-81, 2003.
- [22] Hainsworth S.W. and Macleod M.D. *Particle filtering applied to musical tempo tracking*. Journal of Applied Signal Processing, 15:2385-2395, 2004.
- [23] Klapuri A., Eronen A., and Astola J. *Analysis of the meter of acoustic musical signals*. IEEE Transactions on Speech and Audio Processing, 14(1), 2006.
- [24] Gillet O. and Richard G. *Automatic transcription of drum loops*. IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada, 2004.
- [25] Paulus J.K. and Klapuri A.P. *Conventional and periodic N-grams in the transcription of drum sequences*. IEEE International Conference on Multimedia and Expo, Volume 2, pp. 737-740, Baltimore, Maryland, USA, 2003.
- [26] Herrera P., Peeters G. and Dubnov S. *Automatic classification of musical instrument sounds*. Journal of New Music Research, 32:3-21, 2003.
- [27] Martin K.D. *Automatic transcription of simple polyphonic music: Robust front end processing*. Technical Report 399, MIT Media Laboratory Perceptual Computing Section, 1996.
- [28] Tolonen T. and Karjalainen M. *A computationally efficient multipitch analysis model*. IEEE Transactions on Speech and Audio Processing, 8(6):708-716, 2000.
- [29] Klapuri A.P. *A perceptually motivated multiple-F0 estimation method for polyphonic music signals*. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA, 2005.
- [30] Scheirer E.D. *Tempo and beat analysis of acoustical musical signals*. Journal of the Acoustical Society of America, 103(1):588-601, 1998.
- [31] Mellinger D.K. *Event Formation and Separation of Musical Sound*. PhD thesis Stanford University, Stanford, USA, 1991.
- [32] Kashino K. and Tanaka H. *A sound source separation system with the ability of automatic tone modeling*. International Computer Music Conference, pp. 248-255, Tokyo, Japan, 1993.
- [33] Godsmark D. and Brown G.J. *A blackboard architecture for computational auditory scene analysis*. Speech Communication, 27(3):351-366, 1999.
- [34] Sterian A., Simoni M.H. and Wakefield G.H. *Model-based musical transcription*. In International Computer Music Conference, Beijing, China, 1999.
- [35] Casey M.A. *Auditory Group Theory with Applications to Statistical Basis Methods for Structured Audio*. PhD thesis, Massachusetts Institute of Technology, 1998.

- [36] Casey M.A. and Westner A. *Separation of mixed audio sources by independent subspace analysis*. International Computer Music Conference, Berlin, Germany, 2000.
- [37] Lepain P. *Polyphonic pitch extraction from musical signals*. Journal of New Music Research, 28(4):296-309, 1999.
- [38] Smaragdis P. *Redundancy Reduction for Computational Audition, a Unifying Approach*. PhD thesis, Massachusetts Institute of Technology, 2001.
- [39] Smaragdis P. and Brown J.C. *Non-negative matrix factorization for polyphonic music transcription*. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA, 2003.
- [40] Abdallah S.A. *Towards Music Perception by Redundancy Reduction and Unsupervised Learning in Probabilistic Models*. PhD thesis, Department of Electronic Engineering, King's College London, 2002.
- [41] Abdallah S.A. and Plumbly M.D. *Polyphonic transcription by non-negative sparse coding of power spectra*. International Conference on Music Information Retrieval, pp. 318-325, Barcelona, Spain, 2004.
- [42] FitzGerald D. *Automatic Drum Transcription and Source Separation*. PhD thesis, Dublin Institute of Technology, 2004.
- [43] FitzGerald D., Coyle E. and Lawlor B. *Prior subspace analysis for drum transcription*. Audio Engineering Society 114th Convention, Amsterdam, Netherlands, 2003.
- [44] Paulus J. and Virtanen T. *Drum transcription with non-negative spectrogram factorisation*. European Signal Processing Conference, Antalya, Turkey, 2005.
- [45] Kameoka H., Nishimoto T. and Sagayama S. *Separation of harmonic structures based on tied Gaussian mixture model and information criterion for concurrent sounds*. In IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada, 2004.
- [46] Marolt M. *A connectionist approach to transcription of polyphonic piano music*. IEEE Transactions on Multimedia, 6(3):439-449, 2004.
- [47] Zils A., Pachet F., Delerue O. and Gouyon F. *Automatic extraction of drum tracks from polyphonic music signals*. International Conference on Web Delivering of Music, Darmstadt, Germany, 2002.
- [48] FitzGerald D., Lawlor R. and Coyle E. *Drum transcription in the presence of pitched instruments using prior subspace analysis*. Irish Signals & Systems Conference 2003, Limerick, Ireland, 2003.
- [49] Yoshii K., Goto M. and Okuno H.G. *Drum sound identification for polyphonic music using template adaptation and matching methods*. ISC A Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, Jeju, Korea, 2004.
- [50] Gouyon F., Klapuri A., Dixon S., Alonso M., Tzanetakis G., Uhle C. and Cano P. *An experimental comparison of audio tempo induction algorithms*. IEEE Transactions on Speech and Audio Processing, 2005.
- [51] Kashino K. and Murase H. *A sound source identification system for ensemble music based on template adaptation and music stream extraction*. SpeechCommunication, 27:337-349, 1999.
- [52] Berenzweig A.L. and Ellis D.P.W. *Locating singing voice segments within music signals*. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 119-122, New Paltz, USA, 2001.
- [53] Eggink J. and Brown G.J. *Instrument recognition in accompanied sonatas and concertos*. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 217-220, Montreal, Canada, 2004.
- [54] Vincent E. and Rodet X. *Instrument identification in solo and ensemble music using independent subspace analysis*. International Conference on Music Information Retrieval, Barcelona, Spain, 2004.

- [55] Klapuri A.P. *Signal Processing Methods for the Automatic Transcription of Music*. PhD thesis of Technology, Tampere University of Technology, Tampere, 2004. (διαθέσιμο @<http://www.cs.tut.fi/~klap/iiro>)
- [56] AudioScore Ultimate – Music Transcription Software, <http://www.neuratron.com/audioscore.htm> (τελευταία πρόσβαση 3 Σεπτεμβρίου 2011)
- [57] Gerhard, D. *Pitch Detection and Fundamental Frequency: History and Current Techniques*. Technical Report, Department of Computer Science, University of Regina, November 2003.
- [58] Patricio de la Cuadra. *Pitch Detection Methods Review*. Center for Computer Research in Music and Acoustics, Stanford University. (διαθέσιμο @<https://ccrma.stanford.edu/~pdelac/154/m154paper.htm>)
- [59] Roads, C. *The Computer Music Tutorial*, The MIT Press, 1998.
- [60] Bello J. P., Monti G. Sandler M.. *Techniques for Automatic Music Transcription*. Department of Electronic Engineering, King's College London, 2000.(διαθέσιμο @http://ciir.cs.umass.edu/music2000/papers/bello_abs.pdf)
- [61] Noll, A. M., *Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and maximum likelihood estimate*. Proceedings of the Symposium on Computer Processing in Communications, April, 1969.
- [62] Slaney M., Lyon R.F., *A Perceptual Pitch Detector*. Proceedings of the International Conference on Acoustics Speech and Signal Processing, p. 357-360 vol.1, 1990. (διαθέσιμο @<http://www.dicklyon.com/tech/Hearing/PerceptualPitch-SlaneyLyon.pdf>)
- [63] Moelants D. and Rampazzo C. *A computer system for the automatic detection of perceptual onsets in a musical signal*. In A. Camurri, editor, *KANSEI - The Technology of Emotion*, pages 141–146, Genova: AIMI-DIST, 1997.
- [64] Klapuri A. *Sound onset detection by applying psychoacoustic knowledge*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), volume 6, pages 3089–3092, 1999.
- [65] Jaillet F. and Rodet X. *Improved modelling of attack transients in music analysis synthesis*. In Proceedings of the International Computer Music Conference (ICMC), pages 30–33, Havana, Cuba, 2001.
- [66] Gouyon F. and Dixon S. *Dance music classification: a tempo based approach*. In Proceedings of the International Symposium on Music Information Retrieval (ISMIR), pages 501–504, Barcelona, Spain, October 2004.
- [67] Dixon S., Gouyon F., and Widmer G. *Towards characterisation of music via rhythmic patterns*. In Proceedings of the International Symposium on Music Information Retrieval (ISMIR), pages 509–516, Barcelona, Spain, October 2004.
- [68] Ravelli E., Sandler M. and Bello J.P. *Fast implementation for non-linear time scaling of stereo signals*. In Proceedings of the International Conference on Digital Audio Effects (DAFx-05), pages 182–185, Madrid, Spain, 2005.
- [69] Java – Γλώσσα Προγραμματισμού, <http://www.oracle.com/technetwork/java/index.html>
- [70] JSyn – Application Programming Interface, <http://www.softsynth.com/jsyn/>
- [71] JMusic - Application Programming Interface, <http://jmusic.ci.qut.edu.au/>
- [72] <http://mysite.verizon.net/vzevuty6/paulsfreeguitarscores/index.html> - ιστοσελίδα με διαθέσιμες δωρεάν παρτιτούρες κιθάρας μαζί με τα αρχεία ήχου