ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΠΑΡΑΡΤΗΜΑ ΡΕΘΥΜΝΟΥ
ΤΜΗΜΑ ΜΟΥΣΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ & ΑΚΟΥΣΤΙΚΗΣ

Πτυχιακή Εργασία

Υλοποίηση εικονικού μουσικού οργάνου ελεγχόμενου από μικρόφωνο

Σπουδαστής: Αλέξανδρος Καραγιαννόπουλος (Α.Μ 673)

Επίβλεψη: Χρυσούλα Αλεξανδράκη

Ρέθυμνο 2012

Περίληψη

Η παρούσα πτυχιακή εργασία αφορά στην ανάπτυξη αυτόνομου λογισμικού, το οποίο στοχεύει στη σε πραγματικό χρόνο μουσική ανάλυση του ηχητικού σήματος που λαμβάνεται από το μικρόφωνο του υπολογιστή, και στην επανασύνθεσή του από σήμα MIDI με ηχόχρωμα που επιλέγεται από το χρήστη.

Για την υλοποίηση του λογισμικού χρησιμοποιήθηκε η γλώσσα προγραμματισμού C++ και αξιοποιήθηκαν διάφορες προγραμματιστικές βιβλιοθήκες, που βοήθησαν στην υλοποίηση επιμέρους διεργασιών. Συνολικά, στο τελικό λογισμικό περιλαμβάνονται οι ακόλουθες διεργασίες: α) λήψη ήχου από το μικρόφωνο, β) ανάλυση ήχου για την εξαγωγή μουσικής πληροφορίας (τονικό ύψος, διάρκεια, ένταση), γ) μετατροπή της εξαγόμενης πληροφορίας σε MIDI και δ) αποστολή των δεδομένων MIDI σε συνθετητή MIDI (που μπορεί να είναι είτε εξωτερική συσκευή είτε άλλο λογισμικό) για την τελική ανασύνθεση με εναλλακτικό ηχόχρωμα. Τέλος, όλο το λογισμικό ελέγχεται από μία εύχρηστη γραφική διεπαφή χρήστη.

Το λογισμικό αυτό εξυπηρετεί μουσικούς που θέλουν να πειραματιστούν με διάφορα ηχοχρώματα και ενορχηστρώσεις, χρησιμοποιώντας μονάχα ένα μικρόφωνο που συλλαμβάνει τον ήχο οποιουδήποτε μονοφωνικού μουσικού οργάνου, ή ακόμα και της φωνή τους.

Λέξεις κλειδιά: *Ανάκτηση Μουσικής Πληροφορίας, Χρονική Κατάτμηση, Αναγνώριση Τονικού Ύψους, Αναγνώριση Αρχής Νότας*

Abstract

The current thesis is concerned with the development of a software application, which in real-time receives monophonic musical signal and processes it in order to extract data about the duration, amplitude and tone of the sounding notes. The result is transcribed to MIDI data and sent to a MIDI compliant synthesizer, producing the musical outcome based on the extracted data and the musical timbre chosen by the user.

The software application has been entirely implemented in the C++ programming language, using a number of open source programming libraries that provided the functionalities of audio capturing, audio signal processing, midi I/O handling and graphical user interface development.

The final application aims to aid musicians who want to experiment with various timbral choices and instrumentations, using a single microphone to capture the sound of any monophonic musical instrument or their own voice.

Key words: *Music Information Retrieval, Temporal Segmentation, Pitch Detection, Onset Detection*

Πίνακας Περιεχομένων:

1 Εισαγωγή

1.1 Βασική ιδέα

Στόχος της εργασίας είναι η ανάπτυξη μιας εφαρμογής λογισμικού που υλοποιεί ένα εικονικό μουσικό όργανο. Ο χρήστης μπορεί να ελέγχει το συγκεκριμένο όργανο αναπαράγωντας οποιοδήποτε μουσικό μονοφωνικό σήμα και μέσω της μετατροπής του σε MIDI σήμα, να του προσδίδει το ηχοχρώμα και την τονικότητα της επιλογής του. Για το σκοπό αυτό παρέχεται εξειδικευμένη γραφική διεπαφή χρήστη.

Το σήμα που καταφθάνει στην είσοδο της κάρτας ήχου του υπολογιστή υφίσταται κατάλληλη επεξεργασία, προκειμένου να ανακτηθεί πληροφορία που αφορά το σημασιολογικό του περιεχόμενο. Ειδικότερα, χρησιμοποιούνται αλγόριθμοι ανάκτησης μουσικής πληροφορίας μέσω των οποίων γίνεται τμηματοποίηση (segmentation) του ήχου σε ηχητικά συμβάντα (νότες) ώστε να αναγνωριστεί το τονικό ύψος, η ένταση και η διάρκειά τους. Στη συνέχεια, οι πληροφορίες αυτές μορφοποιούνται σε ρεύμα δεδομένων MIDI και αποστέλλονται σε κάποια γεννήτρια ήχου (synthesizer) προκειμένου να μετατραπούν εκ νέου σε μουσική πληροφορία.

Το λογισμικό αποτελείται από δομοστοιχεία (modules) τα οποία είναι υπεύθυνα για τις διάφορες λειτουργίες του. Στις λειτουργίες αυτές περιλαμβάνονται:

- Λήψη του Ήχου (Audio Capturing).
- Ανάκτηση Μουσικής Πληροφορίας (Music Information Retrieval).
- Μετατροπή Μουσικής Πληροφορίας σε MIDI.
- Επεξεργασία και Παρουσίαση δεδομένων μέσω γραφικής διεπαφής χρήστη.
- Σύνθεση και Αποστολή MIDI μηνυμάτων στην έξοδο της κάρτας ήχου.

1.2 Ερευνητικοί τομείς

Η εργασία στηρίζεται στον τομέα της Ψηφιακής Επεξεργασίας Σήματος (Digital Signal Processing - DSP) και στον τομέα Ανάκτησης Μουσικής Πληροφορίας (Music Information Retrieval - MIR).

1.2.1 Ψηφιακή Επεξεργασία Σήματος

Η ψηφιακή επεξεργασία σήματος ασχολείται με την ψηφιακή αναπαράσταση των σημάτων και την ανάλυση, τροποποίηση και εξαγωγή πληροφοριών από αυτά, με την βοήθεια ψηφιακών επεξεργαστών. Η επεξεργασία σήματος είναι ένα διεπιστημονικό γνωστικό πεδίο, ορισμένο με αυστηρά δικές του μεθοδολογίες και ορολογία. Περιπτώσεις κατά τις οποίες θέλουμε να αφαιρέσουμε το θόρυβο από ένα σήμα, ή να βρούμε τον μετασχηματισμό Fourier κάποιων δεδομένων, ή να μετατρέψουμε ένα σήμα σε μια μορφή πιο κατάλληλη για επεξεργασία και ανάλυση της πληροφορίας που εμπεριέχει, αποτελούν παραδείγματα των εφαρμογών της ψηφιακής επεξεργασίας σήματος.

Η εντυπωσιακή ανάπτυξη της μικροηλεκτρονικής και των υπολογιστών είχε καθοριστική επίδραση στην ψηφιακή επεξεργασία σημάτων και εικόνων. Οι τεχνικές ψηφιακής επεξεργασίας σημάτων χρησιμοποιούνται σήμερα σε πολλές περιοχές της επιστήμης και της τεχνολογίας, και βρίσκουν εφαρμογή στις επικοινωνίες, την αεροναυτική, τη σεισμολογία, τη βιοϊατρική τεχνολογία, την επεξεργασία εικόνας, βίντεο και ήχου, τη συμπίεση δεδομένων, τον αυτοματισμό κ.λ.π.

1.2.2 Ανάκτηση Μουσικής Πληροφορίας

Το πέρασμα της μουσικής στην ψηφιακή μορφή αποθήκευσης, σε συνδυασμό με την ευκολία πρόσβασης που προσφέρει το διαδίκτυο, οδήγησαν στην ταχύτατη αύξηση του μεγέθους των μουσικών αρχείων, προσωπικών συλλογών και μουσικών βιβλιοθηκών. Ο στόχος της Ανάκτησης Μουσικής Πληροφορίας (Music Information Retrieval – MIR) είναι η ανάπτυξη στρατηγικών που να επιτρέπουν την πρόσβαση σε αυτές τις εκτεταμένου μεγέθους μουσικές συλλογές, εξασφαλίζοντας ικανοποιητικά επίπεδα λειτουργικότητας στην αναζήτηση και φυλλομέτρηση (search & browse).

Επί του παρόντος, τα περισσότερα συστήματα πρόσβασης σε μουσικές συλλογές χρησιμοποιούν αποκλειστικά μεταδεδομένα κειμένου (textual metadata) όπως τον τίτλο ενός κομματιού, το όνομα του καλλιτέχνη, το όνομα του συνθέτη, το όνομα του αρχείου, το έτος παραγωγής, το είδος της μουσικής κ.ά. Μία από τις πιο συνήθεις εφαρμογές της ανάκτησης μουσικής πληροφορίας, είναι η έρευνα μεταδεδομένων στο διαδίκτυο για τον

κάθε μουσικό ψηφιακό δίσκο που μεταφέρει ένας χρήστης στον υπολογιστή του, έτσι ώστε να του παρέχονται αυτοματοποιημένα πληροφορίες για το δίσκο, όπως ο τίτλος του, οι τίτλοι των κομματιών, η διάρκεια των κομματιών κ.λ.π.

Τα μεταδεδομένα χωρίζονται σε δύο κατηγορίες, πραγματικά μεταδεδομένα (factual metadata), δηλαδή αντικειμενικές πληροφορίες και πολιτιστικά μεταδεδομένα (cultural metadata), που περιέχουν πληροφορίες υποκειμενικής φύσεως. Μερικά παραδείγματα υποκειμενικών εννοιών που περιλαμβάνονται στα πολιτιστικά δεδομένα είναι το ύφος (mood), συναίσθημα (emotion), είδος (genre) κ.ά. Για να δουλέψει ένα σύστημα που χρησιμοποιεί μεταδεδομένα πρέπει η περιγραφή της μουσικής να είναι ακριβής και το λεξιλογικό νόημα των μεταδεδομένων ευρέως κατανοητό. Προβλήματα όπως λάθη και ανακρίβειες που αφορούν πραγματικές πληροφορίες, μπορούν να περιορίσουν σοβαρά τη χρηστικότητα των συστημάτων που χρησιμοποιούν μεταδεδομένα και επομένως η εξασφάλιση συνοχής της ορθογραφίας, της κεφαλαιοποίησης, της σειράς των ονομάτων κ.τ.λ. είναι απολύτως απαραίτητη ώστε ένα σύστημα μεταδεδομένων να είναι λειτουργικό. Τα περισσότερα συστήματα μεταδεδομένων μουσικών υπηρεσιών χρησιμοποιούν συνδυαστικά πραγματικά και πολιτιστικά μεταδεδομένα. Κάποια από αυτά χρησιμοποιούν εμπειρογνώμονες για να ελέγχουν το περιεχόμενο τους, άλλα εκμεταλλεύονται την αλληλεπίδραση των χρηστών με μουσικά δεδομένα (user feedback), ενώ μερικά συνδυάζουν και τις δύο αυτές μεθόδους [A. Freed,2006].

Παρ' όλη τη χρησιμότητα τους και την ευρεία χρήση τους, τα μεταδεδομένα δεν είναι παρά ένα κομμάτι της έρευνας στον τομέα της ανάκτησης μουσικής πληροφορίας. Το αυξημένο κόστος σε συνδυασμό με το πόσο απαγορευτικά χρονοβόρα είναι τα συστήματα μεταδεδομένων που απασχολούν εμπειρογνώμονες, οι ανακρίβειες που μπορούν να προκύψουν σε συστήματα που βασίζονται στην αλληλεπίδραση με τους χρήστες, και η αδυναμία τους να παρέχουν στους χρήστες εργαλεία αναζήτησης καινούριας, άγνωστης μουσικής, συμπληρώνονται από μεθόδους ανάκτησης μουσικής πληροφορίας βασισμένες στο περιεχόμενο (content-based MIR methods).

Ως περιεχόμενο ορίζονται από τη μια τα μουσικά στοιχεία υψηλού επιπέδου (high level music content description) όπως: ο ρυθμός, η κλίμακα, η μελωδία, το ηχόχρωμα κ.ά. Από την άλλη τα χαμηλού επιπέδου χαρακτηριστικά του ηχητικού σήματος (Low-Level Audio Features) όπως: το φάσμα μέτρου βραχέως χρόνου (short-time magnitude

spectrum), το φάσμα MEL, το χρωμάγραμμα (chromagram), η ανίχνευση έναρξης μουσικών γεγονότων (onset detection), η αναγνώριση Tempo κ.ά. [Michael A. Casey, 2008].

Use Case	Specificity	Description
Music Identification	H	Identify a compact disk, provide metadata about an unknown track, mobile music information retrieval
Plagiarism detection	H	Identify mis-attribution of musical performances, mis-appropriation of music intellectual property
Copyright monitoring	H/M	Monitor music broadcast for copyright infringement or royalty collection
Melody	H/M	Find works containing a melodic fragment
Identical Work/TItle	M	Retrieve performances of same opus number or song title
Performer	M	Find music by a specific artist
Sounds like	M	Finds music that sounds like a given recording
Performance alignment	M	Mapping the performance onto another independent of tempo and repetition structure
Composer	M	Find work by one composer
Recommendation	M/L	Find music that matches the user's personal profile
Mood	L	Find music using emotional concepts
Style/Genre	L	Find music that belongs to a generic category
Instrument(s)	L	Find works with same instrumentation

Πίνακας 1.1: Παραδείγματα εφαρμογών MIR και του βαθμού ειδικότητας τους.

Οι στόχοι των συστημάτων MIR με βάση το περιεχόμενο διαιρούνται σε κατηγορίες ανάλογα με το βαθμό ειδικότητάς (specificity) τους, όπως φαίνεται στον Πίνακα 1.1. Τα συστήματα υψηλής ειδικότητας στοχεύουν στο να ταιριάξουν τμήματα ηχητικού σήματος. Παραδείγματα τέτοιων στόχων είναι η ταυτοποίηση ενός CD, η εύρεση μεταδεδομένων ενός άγνωστου κομματιού, η παρακολούθηση των πνευματικών

δικαιωμάτων κ.ά. Τα συστήματα μέσης ειδικότητας ταιριάζουν μουσικά στοιχεία υψηλού επιπέδου, αλλά όχι ηχητικό σήμα. Τέτοια συστήματα στοχεύουν στην αναζήτηση κομματιών με βάση τη μελωδία, στην αναγνώριση του συνθέτη, στην εύρεση διασκευών κ.λ.π. Τέλος, τα συστήματα χαμηλής ειδικότητας ταιριάζουν μόνο συνολικές (στατιστικές) ιδιότητες μιας συλλογής: αναγνώριση μουσικού είδους, στυλ ύφους, οργάνου κ.λ.π.

MIREX 2007 Task	Evaluation Metric	Best Result
High Level Tasks		
Mood Recognition	Accuracy	61.5%
Classical Composer Recognition	Accuracy	53.72%
Cover Song (Work) Recognition	Accuracy	52%
Artist Recognition	Accuracy	48.14%
Low Level Tasks		
Polyphonic Pitch Tracking	F-Measure	0.614
Onset	F-Measure	0.81
Similarity Tasks		
Query By Humming	Mean Reciprocal Rank	0.92
Audio (Track) Similarity	Avg. Fine Score (0-1)	0.56
Melody Similarity	Avg. Fine Score (0-1)	0.59

Πίνακας 1.2: Αποτελέσματα των συστημάτων με τις καλύτερες επιδόσεις κατάταξης και αναγνώρισης (Mirex 2007).

Ένα τυπικό πρόβλημα που αντιμετωπίζουν αυτά τα συστήματα MIR είναι το λεγόμενο σημασιολογικό κενό μεταξύ ηχητικού σήματος και μουσικής (audio / music semantic gap), δηλαδή το πώς να περάσουμε από τα μετρήσιμα και υπολογίσιμα χαρακτηριστικά του ηχητικού σήματος σε περιγραφές μέσω μουσικών στοιχείων υψηλού επιπέδου, που είναι σαφώς πιο εύκολα διαχειρίσιμα από την ανθρώπινη αντίληψη [L. Lu, 2006]. Στον Πίνακα 1.2 παρατίθενται τα αποτελέσματα των συστημάτων με τις καλύτερες επιδόσεις από το συνέδριο Mirex (2007).

1.3 Προκλήσεις

Στον τομέα της Ανάκτησης Μουσικής Πληροφορίας έχει γίνει αρκετή πρόοδος τα τελευταία χρόνια, ωστόσο οι αλγόριθμοι που υπάρχουν για την αναγνώριση των μουσικών χαρακτηριστικών (Onset, Offset, Pitch κ.τ.λ.) είναι πολλοί, εξειδικευμένοι, και έχουν ένα καθόλου ευκαταφρόνητο ποσοστό λάθους. Πρόκληση αποτελεί η σωστή επιλογή και διαχείριση των αλγορίθμων καθώς και ο σχεδιασμός των επεξεργαστικών βημάτων, ώστε να περιοριστεί ο αριθμός των λαθών στο ελάχιστο δυνατό.

Κατά κανόνα, τα μεγαλύτερα ποσοστά επιτυχίας στην κατάτμηση ηχητικών δεδομένων μέσω της αναγνώρισης νοτών, επιτυγχάνονται όταν η κατάτμηση επιτελείται σε ηχητικά αρχεία. Στην παρούσα εργασία η κατάτμηση επιτελείται σε ζωντανές ηχητικές ροές, γεγονός που επιβαρύνει σημαντικά το ποσοστό επιτυχίας των αλγορίθμων. Ειδικότερα, στην κατάτμηση ηχητικών δεδομένων, οι αλγόριθμοι που χρησιμοποιούνται χωρίζονται σε online και offline αλγόριθμους.

Τα αποτελέσματα των offline αλγορίθμων είναι ορθότερα επειδή εφαρμόζονται σε ηχητικά αρχεία, έχοντας έτσι πρόσβαση σε όλο το εύρος του μουσικού αρχείου και τη δυνατότητα να χρησιμοποιούν αυτές τις πληροφορίες με παράλληλες διεργασίες βελτίωσης της απόδοσης τους, όπως normalisation, Dc-Removal, και threshodling. Αντίθετα, οι online αλγόριθμοι, ανάλογα με το μέγεθος της προσωρινής μνήμης (audio buffer) στην οποία αποθηκεύονται τμήματα του ήχου, έχουν πρόσβαση στα αντίστοιχα δείγματα πριν και μετά το τρέχον σημείο επεξεργασίας. Για να αντισταθμιστεί η διαφορά της απόδοσής τους, έχουν προταθεί και εφαρμόζονται εναλλακτικές τεχνικές όπως dynamic thresholding, median filtering κ.ά.

Επιπρόσθετα, στις εφαρμογές πραγματικού χρόνου, απαιτείται η ελαχιστοποίηση των όποιων καθυστερήσεων υπεισέρχονται κατά την ανάλυση και ανασύνθεση του σήματος, με σημαντικότερη την καθυστέρηση που οφείλεται στην αποθήκευση τμήματος του ήχου στην προσωρινή μνήμη (audio buffer) πριν την αποστολή του στον επεξεργαστή. Η καθυστέρηση αυτή είναι γνωστή ως blocking delay και θεωρητικά υπολογίζεται από τη σχέση 4·(buffersize/(2·samplerate), όπου τέσσερα είναι ο αριθμός των δειγμάτων που απαιτούνται για την ανίχνευση ενός onset. Με συχνότητα δειγματοληψίας 44100 Hz και

μέγεθος του audio buffer ορισμένο στα 512 δείγματα, η καθυστέρηση που προκύπτει είναι ίση με 23.2 ms.

Τέλος, προσωπική πρόκληση αποτελεί η εκμάθηση της προγραμματιστικής γλώσσας C++ και η εξοικείωση με τις απαραίτητες προγραμματιστικές βιβλιοθήκες για την υλοποίηση μιας εφαρμογής εκτεταμένου περιεχομένου όπως αυτή.

2 Σχετιζόμενη Έρευνα

2.1 Κατάτμηση ηχητικών δεδομένων

Η χρονική κατάτμηση ενός ακουστικού κύματος σε μικρότερα στοιχεία είναι θεμελιώδες βήμα για τη μετατροπή των ήχων σε σημασιολογικά αντικείμενα. Τις τελευταίες δύο δεκαετίες, έχει αφιερωθεί σημαντική έρευνα σε αυτό το αντικείμενο και έχουν αναπτυχθεί διάφοροι αλγόριθμοι για τον αυτόματο διαχωρισμό μουσικών σημάτων στα όρια των αντικειμένων του ήχου: αρχή (onset) και τέλος (offset) νότας [Moelants and Rampazzo, 1997][Klapuri, 1999b]. Συστήματα ικανά να εντοπίζουν τα onset τη στιγμή που συμβαίνουν, προσδίδουν νέες προοπτικές στην αλληλεπίδραση μεταξύ ακουστικών και εικονικών μουσικών οργάνων [Puckette et al., 1998].

Η εξαγωγή της χρονικής πληροφορίας των onset είναι χρήσιμη στις εφαρμογές επεξεργασίας ήχου για την ακριβή μοντελοποίηση της ατάκας των ήχων [Masri, 1996] [Jaillet and Rodet, 2001], βοηθά τα συστήματα μεταγραφής στον εντοπισμό της αρχής των νοτών [Bello, 2003][Klapuri, 2004], και μπορεί να χρησιμοποιηθεί σε προγράμματα σύνταξης ήχων (sound editors) για το διαχωρισμό ηχητικών αρχείων στα λογικά τους μέρη [Smith, 1996]. Οι μέθοδοι ανίχνευσης των onset έχουν χρησιμοποιηθεί στην ταξινόμηση μουσικής [Gouyon and Dixon, 2004], στο χαρακτηρισμό ρυθμικών μοτίβων [Dixon et al., 2004], καθώς και σε συστήματα αναγνώρισης ρυθμού (tempo) για να εντοπίσουν τη θέση των παλμών (beats) [Scheirer, 1998b][Davies and Plumbley, 2004].

Υπάρχουν διάφορες προσεγγίσεις για τον εντοπισμό των onset σε μουσικούς ήχους. Οι προσεγγίσεις αυτές κατά κανόνα έχουν δύο στόχους: την κατασκευή συναρτήσεων ανίχνευσης αλλαγών στο σήμα (onset detection functions) και την επιλογή των κορυφών της συνάρτησης (peak picking), ώστε να εξαχθούν οι χρόνοι των onset [Bello et al., 2005].

΄Ενα πρώτο βήμα για την ανάκτηση διακριτών χρόνων onset είναι η αξιολόγηση του ποσοστού μεταβολής του σήματος. Για ένα δεδομένο χρονικό διάστημα υπολογίζεται ένα μέτρο βασισμένο στα χαρακτηριστικά του σήματος και με τη συγκέντρωση συνεχών παρατηρήσεων σχηματίζεται η συνάρτηση εντοπισμού των onset [Bello et al., 2005]

[Klapuri, 1999b]. Ο στόχος των συναρτήσεων αυτών είναι να παράσχουν μία μεσαίου επιπέδου εκπροσώπηση του σήματος, χρησιμοποιώντας μικρότερη δειγματοληψία από το αρχικό ηχητικό. Αυτό έχει ως αποτέλεσμα το παράγωγο τους να παρουσιάζει απότομες κορυφές τη στιγμή που εντοπίζεται ένα onset και να μη παρουσιάζει κορυφές κατά τη διάρκεια της εκτέλεσης μιας συνεχούς νότας, ή από το θόρυβο περιβάλλοντος. Σε δεύτερο στάδιο βρίσκεται η επιλογή των κορυφών από τις οποίες θα ανακτηθεί ο ακριβής χρόνος εμφάνισης των σχετικών onset. Γενικά υπάρχουν τρεις μέθοδοι για την κατασκευή αυτών των συναρτήσεων εντοπισμού:

- Αναγνώριση στο πεδίο του χρόνου κατευθείαν πάνω στην κυματομορφή.
- Αναγνώριση στο πεδίο της συχνότητας χρησιμοποιώντας διάφορες ζώνες συχνοτήτων ή ένα phase vocoder.
- Αναγνώριση χρησιμοποιώντας τεχνικές μηχανικής μάθησης (machine learning techniques) για διάφορα χαρακτηριστικά του σήματος.

Σε πολλές περιπτώσεις, πριν την κατασκευή μιας συνάρτησης εντοπισμού είναι απαραίτητη κάποια προετοιμασία για να τονίσει κάποια χαρακτηριστικά του σήματος και να εξασθενήσει κάποια άλλα. Ανάλογα με τις απαιτήσεις του συστήματος, τα βήματα της απαιτούμενης προεπεξεργασίας (pre-processing) μπορεί να περιλαμβάνουν ομαλοποίηση (normalization) της ενέργειας, ώστε να ελαχιστοποιηθούν οι αλλαγές έντασης στο σήμα, καθώς και αλγόριθμους αφαίρεσης κλικ (click) και θορύβου [Brossier, 2006].

2.1.1 Συναρτήσεις ανίχνευσης onset

Energy

Οι κρουστικοί ήχοι παρουσιάζουν έντονες εξάρσεις ενέργειας κατά την έναρξή τους. Για να εντοπιστεί η αρχή ενός κρουστικού ήχου μετριέται η ενέργεια του σήματος ώστε να ανιχνευθούν αυτές οι εξάρσεις. Ο Andrew W. Schloss χρησιμοποίησε την περιβάλλουσα έντασης του σήματος (amplitude envelope) για να εντοπίσει τις ατάκες των κρουστικών ήχων, όπως φαίνεται στην παρακάτω συνάρτηση:

$$D_H[n] = \sum_{m=-N/2}^{N/2} w[m]x[n+m]^2 \qquad (2.1)$$

όπου w[m] είναι ένα παράθυρο εξομάλυνσης που αξιολογεί το μέσο ενέργειας του παραθύρου με πλάτος N [Schloss, 1985]. Αυτή η προσέγγιση είναι επιτυχής στην ανίχνευση κρουστικών ήχων με οξείες ατάκες που παρουσιάζουν απότομες διακυμάνσεις ενέργειας, αλλά αποτυγχάνει στην ανίχνευση onset που διακρίνονται μέσω αλλαγών της συχνότητας και της χροιάς.

High Frequency Content

Ο Paul Masri πρότεινε τον εντοπισμό ενεργειακών εξάρσεων στο πεδίο της συχνότητας χρησιμοποιώντας ευρείες ζώνες συχνοτήτων [Masri, 1996], δίνοντας έτσι έμφαση στις αλλαγές των συστατικών του φάσματος με υψηλό συχνοτικό περιεχόμενο:

$$D_H[n] = \sum_{k=1}^{N} k \left| X_k[n] e^{j\phi_k[n]} \right|^2 \qquad (2.2)$$

όπου $X_k[n]$ είναι το φασματικό εύρος του σήματος και $\Phi_k[n]$ η φάση του, σε χρόνο n. Αυτή η μέθοδος, επειδή δίνει έμφαση στις συχνοτικές αλλαγές στο υψηλό μέρος του φάσματος και ιδιαίτερα στις εξάρσεις ευρυζωνικού θορύβου, έχει καλά αποτελέσματα στην αναγνώριση κρουστικών onset. Ωστόσο, είναι λιγότερο επιτυχής στη αναγνώριση onset, όταν η πηγή του ήχου δεν προκαλεί ευρείες εξάρσεις ενέργειας, όπως συμβαίνει στα έγχορδα με δοξάρι, στα πνευστά σαν το φλάουτο κ.ο.κ.

Spectral difference

Αλλαγές στο αρμονικό περιεχόμενο και στη θεμελιώδη συχνότητα που προκαλούνται ομαλά, σα να ολισθαίνουν από τη μία στην άλλη, δεν εντοπίζονται επιτυχώς από τις μεθόδους Energy και HFC. Μία από τις μεθόδους που μετράνε τις αλλαγές στο αρμονικό περιεχόμενο είναι γνωστή ως Φασική Διαφορά (Spectral Difference) [Foote and Uchihashi, 2001]. Αυτή η μέθοδος υπολογίζει το μέγεθος της διαφοράς του φασματικού περιεχομένου δύο διαδοχικών δειγμάτων που προκύπτουν από μετασχηματισμό Fourier μικρής διάρκειας (Short Time Fourier Transform). Παρακάτω φαίνεται η συγκεκριμένη συνάρτηση:

$$D_s[n] = \sum_{k=0}^{N} \left| |X_k[n]|^2 - |X_k[n-1]^2| \right| \quad (2.3)$$

Αυτή η συνάρτηση επιχειρεί να προσδιορίσει το ποσοστό της μεταβολής από το ένα δείγμα στο άλλο, σε αντίθεση με τις συναρτήσεις Energy και HFC, όπου οι παρατηρήσεις γίνονται μεμονωμένα σε κάθε δείγμα ξεχωριστά.

Phase deviation

Μια εναλλακτική προσέγγιση παρουσίασε ο Juan-Pablo Bello με τη δημιουργία μιας συνάρτησης που μετρά τη χρονική αστάθεια της φάσης. Έτσι, τα τονικά onset αναγνωρίζονται εντοπίζοντας σημαντικές διακυμάνσεις της φάσης [Bello et al., 2003]. Η φάση ενός σήματος σε σταθερή κατάσταση αναμένεται να γυρίζει σταθερά γύρω από τον τριγωνομετρικό κύκλο. Η φασική καθυστέρηση και η γωνιακή ταχύτητά του, λοιπόν, μπορούν να θεωρηθούν σταθερές και η επιτάχυνσή του μηδενική, οπότε για να εντοπιστούν αλλαγές σε ένα μη σταθερό σήμα αρκεί να παρατηρήσουμε τη φασική επιτάχυνση. Η συνάρτηση αυτή κατασκευάστηκε από τον ποσοτικό προσδιορισμό της απόκλισης της φάσης:

$$\hat{\phi}_\kappa[n] = princ\arg\left(\frac{\vartheta^2 \varphi_k[n]}{\vartheta n^2} \right) \quad (2.4)$$

όπου princarg (Principal Argument Function) είναι μια συνάρτηση που δίνει το ακτινικό μέτρο του ορίσματος ενός μιγαδικού αριθμού στο εύρος [-π,π]. Έτσι προκύπτει η συνάρτηση:

$$D_\varphi[n] = \sum_{k=0}^{N} \left| \hat{\varphi}_k[n] \right| \quad (2.5)$$

Ένα μειονέκτημα αυτής της προσέγγισης είναι ότι σημαντικές αλλαγές της φάσης μπορεί να συμβούν χωρίς να σχετίζονται με κάποια μουσική αλλαγή. Για παράδειγμα, τα θορυβώδη σημεία του σήματος παρουσιάζουν συνήθως ασταθή φάση. Παρόλο ότι αυτό δεν μπορεί να επηρεάσει τονικά γεγονότα με έντονο αρμονικό περιεχόμενο, σε κρουστικούς ήχους και όταν το σήμα είναι θορυβώδες, μπορούν να παρουσιαστούν μεγάλες αποκλίσεις.

Complex-domain distance

Για να εντοπιστούν τόσο τα τονικά, όσο και τα κρουστικά onset, χρησιμοποιούνται συνδυαστικά οι προσεγγίσεις spectral difference και phase deviation [Duxbury et al., 2003] για να παραχθεί μια πρόβλεψη για το τρέχον δείγμα:

$$\hat{X}_k[n] = \left|X_k[n]\right| e^{j\hat{\phi}_k[n]}$$

όπου Φκ δίνεται από τη συνάρτηση (2.4). Μετρώντας την απόσταση μεταξύ του προβλεπόμενου δείγματος και του τρέχοντος δείγματος, που προκύπτει από τον STFT (Short Time Fourier Transform), έχουμε:

$$D_c[n] = \sum_{k=0}^{N} \left|\hat{X}_k[n] - X_k[n]\right|^2 \qquad (2.6)$$

Αυτό το μέτρο υπολογίζει την απόσταση ανάμεσα στο τρέχον δείγμα και το δείγμα που προβλέφθηκε προηγουμένως, θεωρώντας ότι το πλάτος και η μετατόπιση της φάσης είναι σταθερά μέτρα.

Kullback-Liebler distance

Μπορούν να παρθούν εναλλακτικά μέτρα για να υπολογιστεί η απόσταση μεταξύ δύο συνεχόμενων δειγμάτων. Στοχεύοντας στον τονισμό των ενεργειακών αυξήσεων και αγνοώντας τις μειώσεις, μπορεί να χρησιμοποιηθεί η απόσταση Kullback-Liebler:

$$D_{kl}[n] = \sum_{k=0}^{N} \left|X_k[n]\right| \log \frac{\left|X_k[n]\right|}{\left|X_k[n-1]\right|} \qquad (2.7)$$

Αυτή η συνάρτηση αναδεικνύει τις θετικές αλλαγές του πλάτους στο σήμα, παρουσιάζοντας μεγάλες κορυφές καθώς από την σιωπή περνάμε σε κάποιο ηχητικό γεγονός. Μια παραλλαγή αυτής της προσέγγισης παρουσιάζεται από τους [Hainsworth and Macleod, 2003], αφαιρώντας το $\left|X_k[n]\right|$ και εντείνοντας έτσι τις διακυμάνσεις του πλάτους:

$$D_{mkl}[n] = \sum_{k=0}^{N} \log \frac{\left|X_k[n]\right|}{\left|X_k[n-1]\right|} \tag{2.8}$$

Ο Paul Brossier στοχεύοντας να αποτρέψει τη συνάρτηση από το να παίρνει αρνητικές τιμές, κάτι το οποίο θα αύξανε την πολυπλοκότητα της επιλογής των κορυφών στο επόμενο στάδιο επεξεργασίας, διαμόρφωσε περαιτέρω την παραπάνω συνάρτηση ως εξής:

$$D'_{kl}[n] = \sum_{k=0}^{N} \log \left(1 + \frac{\left|X_k[n]\right|}{\left|X_k[n-1] + \in\right|}\right) \tag{2.9}$$

όπου $\in$ είναι μια σταθερά με τιμή $\in = 10^{-6}$, σχεδιασμένη για να αποφεύγονται μεγάλες διακυμάνσεις όταν το σήμα έχει πολύ χαμηλά επίπεδα ενέργειας, αποτρέποντας έτσι την παρουσία μεγάλων κορυφών τις χρονικές στιγμές που υπάρχουν offset [Brossier, 2006].

2.1.2 Επιλογή Κορυφών

Η τελική επιλογή των onset γίνεται εντοπίζοντας τα τοπικά μέγιστα των συναρτήσεων ανίχνευσης, όταν αυτά αντιστοιχούν σε αντιληπτά onset, τα οποία υπερβαίνουν ένα όριο (threshold). Μία εναλλακτική παρουσιάζεται στο άρθρο των [Puckette et al., 1998], σύμφωνα με το οποίο ο χρόνος των onset εξάγεται όταν παρουσιάζονται έντονες αλλαγές στο πλάτος της συνάρτησης ανίχνευσης και όχι με βάση τα τοπικά μέγιστα. Αυτή η μέθοδος έχει χρησιμοποιηθεί στο αντικείμενο "bonk~" του προγράμματος PureData και φαίνεται να παρουσιάζει καλά αποτελέσματα όσον αφορά τα κρουστικά onset. Μία ακόμη πρόταση βασίζεται στις τεχνικές μηχανικής μάθησης (machine learning techniques), όπου γίνεται η αναγνώριση κάποιων χαρακτηριστικών σχηματικών μοτίβων στη συνάρτηση ανίχνευσης onset [Abdallah and Plumbley, 2003]. Ωστόσο, τέτοιες προσεγγίσεις καθίστανται σχεδόν απαγορευτικές σε εφαρμογές πραγματικού χρόνου, λόγω της πολυπλοκότητάς τους καθώς και του υψηλού υπολογιστικού τους κόστους.

Post-processing

Εάν ληφθούν κάποια μέτρα πριν προβούμε στη διαδικασία αναζήτησης τον τοπικών μέγιστων, περιορίζεται ο αριθμός των ψευδών κορυφών που μπορούν να παρουσιαστούν. Μερικά τυπικά μέτρα του σταδίου της μετεπεξεργασίας που εφαρμόζονται στις συναρτήσεις ανίχνευσης onset είναι τα εξής: φιλτράρισμα χαμηλών συχνοτήτων (Low pass filtering), αφαίρεση Dc θορύβου (bias Dc-removal) και ομαλοποίηση (normalization) [Bello et al., 2005]. Το φιλτράρισμα μεσαίων και υψηλών συχνοτήτων στοχεύει στην μείωση του θορύβου του σήματος, ώστε να ελαχιστοποιηθούν οι λανθασμένες ανιχνεύσεις. Αυτό μπορεί να υλοποιηθεί αποτελεσματικά χρησιμοποιώντας ένα FIR φίλτρο.

$$\tilde{D}[n] = D[n] + \sum_{m=1}^{M} a_m D[n-m] \tag{2.10}$$

Αυτή η διαδικασία μειώνει αποτελεσματικά τις ψευδείς κορυφές, και παρ' ότι επιφέρει στο σύστημα ένα μικρό ποσοστό επιπλέον υπολογιστικού κόστους, μπορεί να θεωρηθεί κατάλληλη για εφαρμογές πραγματικού χρόνου. Για να περιοριστεί η καθυστέρηση που επιφέρει το φίλτρο, το παράθυρο της συνάρτησης γύρω από το τρέχον δείγμα φιλτράρεται και από τις δύο κατευθύνσεις, προσομοιώνοντας μηδενική καθυστέρηση φάσης.

Οι διεργασίες Dc-Removal και normalization προσδίδουν σταθερό εύρος στη συνάρτηση (συνήθως μεταξύ των τιμών 0 και 1) και έτσι, δίνοντας στη συνάρτηση ένα συγκεκριμένο προφίλ, ανεξάρτητα από το πλάτος και τη φύση του ήχου, εξασφαλίζεται η βελτίωση των αποτελεσμάτων της διαδικασίας επιλογής κορυφών (peak-picking). Σε εφαρμογές πεπερασμένου χρόνου αυτές οι διεργασίες παρουσιάζουν καλά αποτελέσματα, λόγω της δυνατότητας να χρησιμοποιούν πληροφορίες από μεγάλο χρονικό τμήμα του μουσικού σήματος τόσο πριν, όσο και μετά το τρέχον δείγμα. Στις εφαρμογές πραγματικού χρόνου, αυτό προσεγγίζεται χρησιμοποιώντας ένα μεγάλο παράθυρο επεξεργασίας, το οποίο όμως αυξάνει κατά πολύ την καθυστέρηση του συστήματος, καθιστώντας την διεργασία ακατάλληλη για τέτοιου τύπου εφαρμογές.

Dynamic thresholding

Για να εντοπιστούν τα onset πρέπει να εντοπιστούν οι κορυφές της μετεπεξεργασμένης συνάρτησης ανίχνευσης που αντιστοιχούν σε πραγματικούς χρόνους onset, και να απορριφθούν οι υπόλοιπες κορυφές που οδηγούν σε ψευδή onset.

Ανάλογα με το περιεχόμενο του σήματος και κυρίως την ένταση, μπορούν να παρατηρηθούν σημαντικές διακυμάνσεις στο πλάτος των συναρτήσεων ανίχνευσης. Για να αντιμετωπιστούν αυτές οι έντονες μεταβολές χρησιμοποιείται δυναμική κατωφλίωση (dynamic thresholding): με βάση κάθε παρατήρηση στη συνάρτηση ανίχνευσης, υπολογίζεται ένα κατώφλι βασιζόμενο σε ένα μικρό αριθμό μελλοντικών και παρελθοντικών παρατηρήσεων, το οποίο συγκρίνεται με το πλάτος της τρέχουσας παρατήρησης. Μία μέθοδος κατασκευής δυναμικού κατωφλίου είναι η Frame Histogramming [Hainsworth and Macleod, 2003], όπου το πιο κατάλληλο όριο της συνάρτησης ανίχνευσης καθορίζεται από τη μελέτη του πλήθους των παρατηρήσεων γύρω από την τρέχουσα χρονική στιγμή. Η μέθοδος του κινητού μέσου (moving median) έχει αποδειχτεί επιτυχής στη μείωση του θορύβου, στη μείωση του αριθμού ψευδών κορυφών [Rabiner et al., 1995] και έχει εφαρμοστεί επιτυχημένα σε συναρτήσεις ανίχνευσης onset [Bello et al. 2005]. Αυτή η μέθοδος αποδεικνύεται βιώσιμη όσον αφορά στο υπολογιστικό κόστος, επειδή ο μέσος υπολογίζεται απλά μέσω της ταξινόμησης ενός πίνακα τιμών. Το δυναμικό κατώφλι υπολογίζεται από την τιμή του μέσου και ένα μικρό αριθμό δειγμάτων γύρω από το τρέχον δείγμα:

$$\delta_t[n] = \lambda \cdot median(D[n-a],...D[n],...D[n+b]+\delta) \qquad (2.11)$$

όπου ο τομέας D[n − a], · · · D[n], · · · D[n + b] περιέχει α δείγματα πριν το n, και β δείγματα μετά το n. Ο συντελεστής διαβάθμισης λ και ο συντελεστής εξομάλυνσης δ είναι προκαθορισμένες παράμετροι. Τέλος, τα onset επιλέγονται με βάση τα τοπικά μέγιστα στο εύρος D[n] − δt[n].

Real-time peak-picking

Για να επιτευχθεί μια εύρωστη συλλογή σχετικών μέγιστων, χωρίς ωστόσο να δημιουργείται μεγάλη χρονική καθυστέρηση, προτάθηκε μια τροποποίηση της προηγούμενης προσέγγισης [Brossier, 2006]. Το δυναμικό κατώφλι υπολογίζεται βάσει ενός μικρού παραθύρου γύρω από την τρέχουσα θέση, συνυπολογίζοντας τη μέση τιμή:

$$\begin{aligned}\tilde{\delta}_t[n] &= \lambda \cdot median(D[n-a],...D[n],...D[n+b]+\delta) \\ &+ \alpha \cdot mean(D[n-a],...D[n],...D[n+b]+\delta) \\ &+ \delta,\end{aligned} \tag{2.12}$$

όπου α είναι ένας θετικός συντελεστής στάθμισης. Η εισαγωγή της μέσης τιμής στην εξίσωση γίνεται με σκοπό να προσομοιωθούν τα αποτελέσματα που θα είχαν οι μέθοδοι DC-removal και normilazation σε εφαρμογές πεπερασμένου χρόνου. Για να γίνει αυτό χρησιμοποιείται μικρό παράθυρο επεξεργασίας, επιτρέποντας έτσι στη διαδικασία Peak-picking να αντιμετωπίζει δυναμικές αλλαγές που συναντώνται σε μουσικά ηχητικά σήματα. Πειραματικά αποτελέσματα [Brossier et al., 2004b] έχουν δείξει ότι για μικρές τιμές των a και b, αυτή η μέθοδος είναι ανθεκτική σε αυτές τις αλλαγές.

Μετά τη μετεπεξεργασία της συνάρτησης και τον υπολογισμό του δυναμικού κατωφλίου, η διαδικασία Peak-Picking συνίσταται στην επιλογή των τοπικών μεγίστων που βρίσκονται πάνω από αυτό. Η ανίχνευση των τοπικών μεγίστων συνεπάγεται τη σύγκριση τουλάχιστον τριών διαδοχικών παρατηρήσεων, κάτι που απαιτεί γνώση τουλάχιστον μιας παρατήρησης μετά την κορυφή. Οι χρόνοι των onset τελικά ορίζονται ως κάθε τοπικό μέγιστο της συνάρτησης Peak-Picking:

$$\hat{D}[n] = D[n] = \tilde{\delta}_t[n] \tag{2.13}$$

όπου D[n] είναι οποιαδήποτε από τις συναρτήσεις onset που παρουσιάστηκαν στην ενότητα 2.1.1 και δ[n] η σχέση (2.12).

Silence Gate

Ακουστικές δοκιμές έχουν δείξει ότι στους δίσκους βινυλίου ο αριθμός των ψευδών onset είναι μεγαλύτερος σε σχέση με τους ψηφιακούς δίσκους, όπου ο θόρυβος περιβάλλοντος είναι ασθενέστερος [Brossier et al., 2004b]. Πολλές φορές οι διακυμάνσεις του πλάτους σε περιοχές με χαμηλή ενέργεια εκλαμβάνονται λανθασμένα ως κορυφές από τις συναρτήσεις ανίχνευσης. Για αυτό το λόγο εφαρμόζεται μια πύλη σιωπής (silence gate), ελαττώνοντας έτσι τα λάθη που προκαλούνται από τον θόρυβο περιβάλλοντος και το θόρυβο κβαντισμού. Όταν το σήμα πέσει κάτω από ένα ορισμένο επίπεδο τα onset απορρίπτονται, επιτυγχάνοντας με αυτό τον τρόπο την ελαχιστοποίηση των onset που εντοπίζονται εσφαλμένα κατά τη διάρκεια είτε παύσεων, είτε ησυχίας.

2.2 Ανάλυση τονικού ύψους

Ο στόχος ενός συστήματος ανίχνευσης τονικού ύψους (pitch detection system) είναι να αναγνωρίσει τους ήχους που διαμορφώνουν την αίσθηση της τονικότητας και να εκτιμήσει τη συχνότητα που αντιστοιχεί στο αντιλαμβανόμενο τονικό ύψος. Πολλά από τα μοντέλα αναγνώρισης τονικού ύψους προέρχονται από τεχνικές επεξεργασίας λόγου [Rabiner et al., 1976] [Wise et al., 1976] και χρησιμοποιούνται σε διάφορα συστήματα όπως: στην αυτόματη δημιουργία παρτιτούρας (automated music transcription), στην παρακολούθηση παρτιτούρας (score following), στην αναγνώριση και ταξινόμηση μουσικής (music recongition and classification), στην τροποποίηση μελωδίας (melody modification), στη χρονική ευθυγράμμιση (time stretching) καθώς και σε άλλα ηχητικά εφέ.

Υπάρχει ένας μεγάλος αριθμός μεθόδων για την εκτίμηση της τονικότητας σημάτων ομιλίας [Rabiner, 1989] [Gómez et al., 2003b] και μουσικής [Roads, 1996] [Klapuri, 2000] [Cheveigné, 2004], τα οποία κατά κύριο λόγο λειτουργούν εκτιμώντας τη θεμέλιο συχνότητα κάθε μουσικού συμβάντος (νότας). Η θεμέλιος συχνότητα f0 ενός περιοδικού σήματος είναι η αντίστροφος της περιόδου του. Η περίοδος εν προκειμένω μπορεί να οριστεί ως "το μικρότερο μέλος ενός συνόλου άπειρων χρονικών μετατοπίσεων που αφήνουν το σήμα αμετάβλητο" [Cheveigné and Kawahara, 2002]. Στη μουσική, ωστόσο, το σήμα δεν είναι απόλυτα περιοδικό και ο ορισμός αυτός εφαρμόζεται σε ένα συγκεκριμένο χρονικό τμήμα γύρω από το τρέχον σημείο της ανάλυσης.

Στις περισσότερες περιπτώσεις η θεμέλιος συχνότητα μίας νότας αντιστοιχεί στην αντιλαμβανόμενη τονικότητα, χωρίς αυτό όμως να αποτελεί ανεξαίρετο κανόνα [Pressnitzer et al., 2001], καθώς το αντιλαμβανόμενο τονικό ύψος των μουσικών οργάνων εξαρτάται και από τις αρμονικές συχνότητες που παράγουν [Yost, 1996]. Τα μουσικά όργανα έχουν διαφορετικές αρμονικές δομές και το πλάτος των αρμονικών τους μεταβάλλεται με το χρόνο, προσδίδοντας στο κάθε όργανο ξεχωριστή χροιά. Η συχνότητα των αρμονικών σε ένα μη περιοδικό σήμα υπολογίζεται από τη σχέση:

$$f_n = (n+1) f_0 \sqrt{1 + Bn^2} \tag{2.14}$$

όπου n είναι ο αριθμός της τάξης της αρμονικής συχνότητας και B είναι ένας παράγοντας που λέγεται παράγοντας μη αρμονικότητας (inharmonicty factor) και η τιμή του κυμαίνεται ανάλογα με τις φυσικές ιδιότητες του κάθε οργάνου [Fletcher and Rossing, 1998].

Οι διάφοροι διαθέσιμοι αλγόριθμοι για την εκτίμηση της θεμελίου συχνότητας κατηγοριοποιούνται γενικά σε δύο κατηγορίες: α) στις μεθόδους που εκτιμούν την περιοδικότητα της κυματομορφής του σήματος (μέθοδοι στο πεδίο του χρόνου - time domain methods) και β) στις μεθόδους που αναζητούν αρμονικά μοτίβα στο φάσμα (μέθοδοι στο πεδίο της συχνότητας - frequency or spectral domain methods). Οι φασματικές προσεγγίσεις τείνουν να έχουν καλά αποτελέσματα στο ψηλότερο μέρος του φάσματος και υστερούν στο χαμηλό, ενώ αντίθετα οι χρονικές παρουσιάζουν περισσότερα λάθη στις υψηλές συχνότητες, κυρίως όσο πλησιάζουν στη συχνότητα δειγματοληψίας. Μερικά συστήματα χρησιμοποιούν συνδυαστικά μεθόδους βασισμένες στο πεδίο του χρόνου και της συχνότητας, στοχεύοντας να εκμεταλλευτούν τα δυνατά σημεία του καθενός ώστε να έχουν καλύτερα αποτελέσματα στο συνολικό εύρος του φάσματος του ήχου [Lyon and Dyer, 1986].

Η συνολική διαδικασία εκτίμησης της θεμελίου συχνότητας χωρίζεται σε τρία βασικά στάδια [Hess, 1984]:

1. Προεπεξεργασία του ηχητικού σήματος (Preprocessing).
2. Εξαγωγή της συχνότητας.
    - Στο πεδίο του χρόνου.
    - Στο πεδίο της συχνότητας.
3. Μετεπεξεργασία αποτελεσμάτων (Post - processing).

2.2.1 Προεπεξεργασία

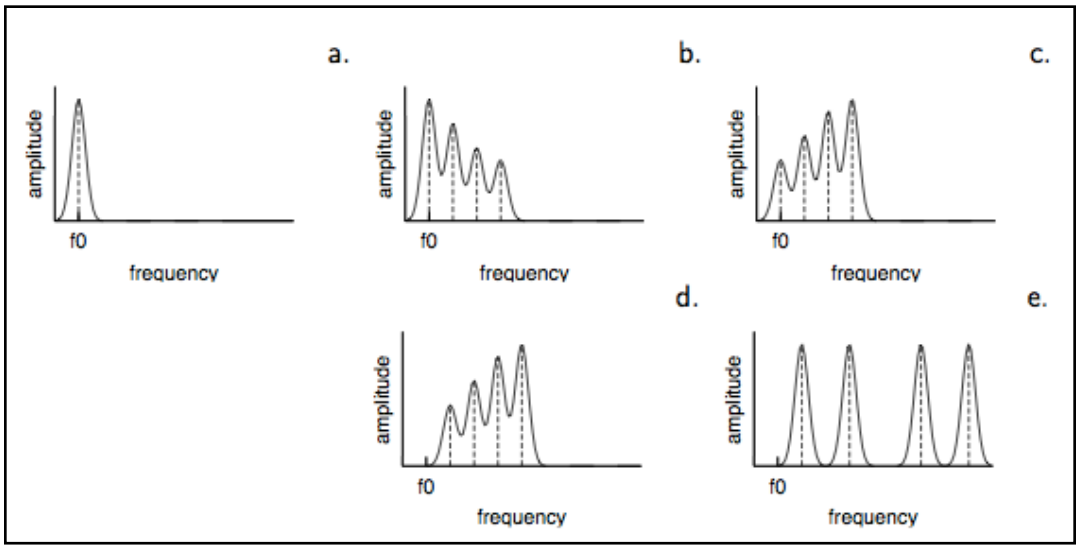
Για να μεγιστοποιηθεί η ακρίβεια ενός αλγόριθμου αναγνώρισης τονικού ύψους, μπορεί να γίνει κάποια προεπεξεργασία στο σήμα. Μία περίπτωση είναι η ενίσχυση των μεσαίων συχνοτήτων του σήματος, ώστε να προσομοιώνεται η αντίληψη της ακουστότητας της ανθρώπινης ακοής και να μεγιστοποιείται η ενέργεια του σήματος στην περιοχή όπου είναι πιο πιθανό να βρίσκονται οι υποψήφιες συχνότητες. Τα φίλτρα στάθμισης-A (A-weighting) και τα φίλτρα στάθμισης-C (C-weighting) είναι ιδανικά για την περίπτωση αυτή αφού ενισχύουν τις συχνότητες στο εύρος 1 kHz εώς 5 kHz και εξασθενούν τις χαμηλές και υψηλές συχνότητες του φάσματος.

Στο πεδίο του χρόνου αυτά τα φίλτρα είναι αποτελεσματικά για τις εφαρμογές πραγματικού χρόνου, επειδή προκαλούν μικρές καθυστερήσεις και έχουν γραμμικό υπολογιστικό κόστος. Ωστόσο, αυτά τα φίλτρα είναι αρκετά πολύπλοκα στο σχεδιασμό και απαιτούν τη διαδοχική σύνδεση βραχέων φίλτρων για να μειωθεί η συσσώρευση των σφαλμάτων [Schlichth¨arle, 2000]. Στο φασματικό τομέα μπορεί να γίνει ακριβής ισοστάθμιση σε όλο το πεδίο του φάσματος, μοντελοποιώντας την απόκριση συχνότητας του έξω και μέσου αυτιού. Αυτά τα βήματα είναι προαιρετικά, αλλά τείνουν να αυξάνουν την ακρίβεια του συστήματος και την αντοχή του στην παρουσία θορύβου με αντίτιμο το αυξημένο υπολογιστικό τους κόστος.

Άλλη μια περίπτωση που παρουσιάζει ενδιαφέρον είναι η αφαίρεση ή μείωση των μη σταθερών συνιστωσών του σήματος, ώστε να κρατηθούν μόνο τα ημιτονοειδή στοιχεία του [Cano, 1998] [Duxbury et al., 2001], βελτιώνοντας έτσι την εκτίμηση της θεμέλιου συχνότητας.

2.2.2 Αναγνώριση τονικού ύψους στο πεδίο της συχνότητας.

Οι διάφορες μουσικές χροιές μπορεί να έχουν εντελώς διαφορετικό φασματικό περιεχόμενο αλλά να προσδίδουν την ίδια τονική αίσθηση, με αποτέλεσμα να περιπλέκουν τη διαδικασία της αναγνώρισης της συχνότητας. Διάφορα παραδείγματα διαφορετικών φασματικών μοτίβων που τα αντιλαμβανόμαστε σαν τονικά όμοια φαίνονται στην Εικόνα 2.1 [de Cheveigné, 2004].



Εικόνα 2.1: Διάφορα φασματικά μοτίβα που παράγουν την ίδια τονική αίσθηση. a) καθαρός τόνος, b) αρμονικός τόνος, c) αρμονικός τόνος με ισχυρότερους αρμονικούς από τη θεμέλιο, d) τόνος μόνο από αρμονικούς, e) μη αρμονικός τόνος.

Στο πεδίο της συχνότητας διακρίνονται γενικά δύο τύποι μεθόδων για την αναγνώριση του τονικού ύψους: α) οι μέθοδοι θέσης (spectral place methods), οι οποίες βασίζονται στον εντοπισμό της θεμελιώδους συχνότητας επιλέγοντας φασματικές συνιστώσες ανάλογα με τη θέση τους στο φάσμα και β) οι μέθοδοι διαστημάτων (spectral interval methods), οι οποίες βασίζονται στον υπολογισμό των διαστημάτων μεταξύ των αρμονικών συχνοτήτων [Klapuri [2000].

Fast comb spectral model

Αυτός ο αλγόριθμος προέρχεται από έναν απλό αλγόριθμο αντιστοίχισης μοτίβων [Lang, 2003]. Οι N φασματικές συνιστώσες με την περισσότερη ενέργεια αποθηκεύονται σε έναν πίνακα τιμών και κατόπιν επιλέγεται αυτή με το μεγαλύτερο πλάτος, ώστε να συγκριθεί με τις υπόλοιπες N-1 συνιστώσες. Εάν μία από αυτές είναι υποαρμονική της κυρίαρχης συνιστώσας, με ανοχή στη σύγκριση που καθορίζεται από έναν παράγοντα μη αρμονικότητας, και αν το πλάτος της είναι μεγαλύτερο από το μισό της κυρίαρχης συνιστώσας, τότε επιλέγεται αυτή ως θεμέλιος συχνότητα. Η αναλογία του μισού για τη σύγκριση του πλάτους των συνιστωσών, καθώς και ο παράγοντας μη αρμονικότητας που γράφεται από τη σχέση: $n - 0.2 < fn2 / fn1 < n + 0.2$, έχουν τεθεί εμπειρικά [Brossier, 2006]. Αυτή η διαδικασία συνεχίζεται έως ότου έχουν συγκριθεί όλες οι συνιστώσες μεταξύ τους. Αυτή η μέθοδος λειτουργεί αποτελεσματικά για τα μοτίβα (a) έως (c) της Εικόνας 2.1.

Multi comb spectral filtering

Υπάρχουν πολλές εφαρμογές που λειτουργούν ταιριάζοντας φασματικά μοτίβα χρησιμοποιώντας συχνοτικά ιστογράμματα (frequency histogramms) [de Cheveign´e, 2004]. Ο ακόλουθος αλγόριθμος βασίζει τη λειτουργία του σε ένα phase vocoder παρόμοιο με αυτό που χρησιμοποιείται στις συναρτήσεις ανίχνευσης onset στην παράγραφο 2.1.1. Αρχικά το σήμα εισόδου φιλτράρεται περνώντας από ένα A-weighting IIR φίλτρο και στη συνέχεια γίνεται normalized. ΄Ετσι τονίζεται το μεσαίο κομμάτι του φάσματος, εξασθενούν οι χαμηλές και υψηλές συχνότητες, ενώ εξομαλύνονται οι κορυφές με σχετικά μικρότερη ενέργεια από τις υπόλοιπες. Στη συνέχεια επιλέγονται οι εναπομείναντες κορυφές και περνάνε από ένα κτενοειδές φίλτρο (comb filter). Από κάθε κορυφή παράγεται ένα σύνολο τονικών υποθέσεων βασιζόμενο στους πρώτους Z υποαρμονικούς:

$$f_z^0 = \frac{f}{z} \; with \; \{ \; 1 \le z \le Z, \;\; z \in N \qquad (2.15)$$

όπου f είναι η συχνότητα της κορυφής η οποία υπολογίζεται μέσω μιας μεθόδου τετραγωνικής παρεμβολής (quadratic interpolation). Για κάθε υπόθεση κατασκευάζεται ένα αρμονικό πλέγμα:

$$C_z(k) = \begin{cases} 1 \; if \; \exists \; m \; s.t. \left| \frac{1}{m} \frac{1}{f_z^0} - \frac{1}{k} \frac{N}{f_s} \right| < \frac{\omega_b}{k} \\ 0 \; otherwise \end{cases} \quad (2.16)$$

όπου f_s είναι η συχνότητα δειγματοληψίας, m είναι ένας ακέραιος που ορίζει το μέγιστο αριθμό των αρμονικών συνιστωσών που λαμβάνονται υπόψη, και ω_b είναι ένας αριθμός που καθορίζει την ανεκτικότητα στη σύγκριση του αρμονικού περιεχομένου. Τα κριτήρια για κάθε υποψήφιο ταίριασμα είναι διαφορετικά, τα πιο σημαντικά εκ των οποίων είναι η επαλήθευση του αριθμού των αρμονικών συνιστωσών και της συνολικής ενέργειας τους [Lepain, 1999].

2.2.3 Αναγνώριση τονικού ύψους στο πεδίο του χρόνου.

Μία μέθοδος για τον εντοπισμό της θεμελιώδους συχνότητας στο πεδίο του χρόνου συνίσταται στην παρατήρηση μοτίβων περιοδικότητας του σήματος μέσω της κυματομορφής του. Ένας από τους πιο γρήγορους τρόπους για να υπολογιστεί η τονικότητα είναι να μετρηθούν οι διελεύσεις από το μηδέν (zero-crossings) σε ένα συγκεκριμένο χρονικό διάστημα, κάτι το οποίο απαιτεί εξονυχιστική έρευνα σε κάθε παράθυρο επεξεργασίας για να εντοπιστούν οι αλλαγές των προσήμων.

Αυτή η μέθοδος είναι επιτυχής όταν το σήμα αποτελείται από απλούς ημιτονοειδείς τόνους, αλλά αποτυγχάνει όταν στοχεύει σε πιο πολύπλοκους τύπους σημάτων. Για παράδειγμα, ο αριθμός των διελεύσεων από το μηδέν ενός αρμονικού ήχου συχνά δεν έχει σχέση με το μήκος κύματος του, καθώς το πρόσημο της κυματομορφής μπορεί να αλλάξει πάνω από μία φορά σε μία περίοδο. Επιπλέον, η παρουσία θορύβου στο σήμα μπορεί να δυσκολέψει ακόμη περισσότερο τη σωστή καταμέτρηση των διελεύσεων από το μηδέν, είτε αυξάνοντας είτε μειώνοντας τον αριθμό των αλλαγών προσήμου εντός του παραθύρου επεξεργασίας. Υπάρχει μια παραλλαγή αυτής τη μεθόδου, στην οποία μετριούνται οι κορυφές ενός ορισμένου χρονικού διαστήματος, αλλά και σε αυτήν αντιμετωπίζονται παρόμοιες δυσκολίες. Σε γενικές γραμμές, η επιλογή ενός αξιόπιστου σημείου αναφοράς ώστε να υπολογιστεί η περίοδος είναι δύσκολη [de Cheveigńe, 2004].

Schmitt trigger

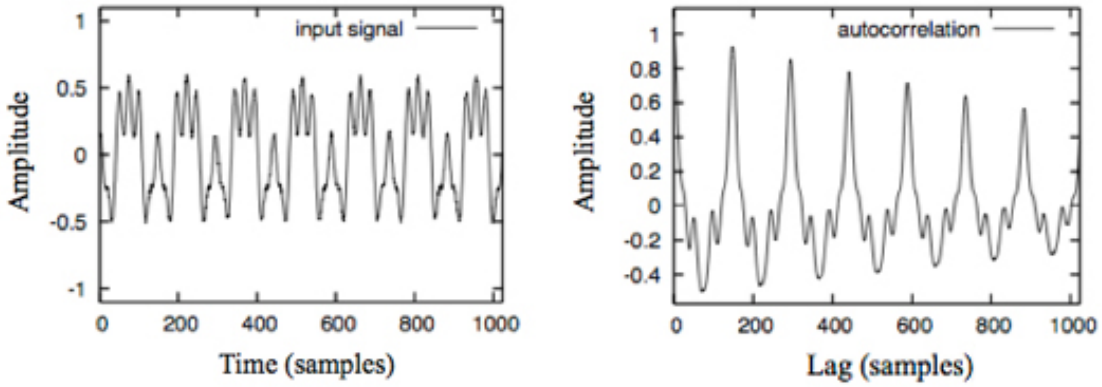
Μία πιο επιτυχημένη προσέγγιση έγκειται στην υλοποίηση ενός κυκλώματος Schmitt trigger [Simpson, 1987]. Πρόκειται για ένα συγκριτικό κύκλωμα που χρησιμοποιεί διπλή κατωφλίωση. Όταν η τάση εισόδου γίνεται μεγαλύτερη από το «ανώτερο» κατώφλι, η τάση εξόδου είναι υψηλή, ενώ όταν η τάση εισόδου είναι μικρότερη από το κατώτερο κατώφλι, η τάση εξόδου είναι χαμηλή. Το κύκλωμα λειτουργεί ως μνήμη, η οποία περιγράφει έναν κύκλο υστέρησης και συνιστά έναν ανιχνευτή περιόδου. Στην περίπτωση του μουσικού ήχου η τάση εισόδου είναι το ηχητικό σήμα και για να αντεπεξέλθει το σύστημα στις συνεχόμενες αλλαγές πλάτους, τα κατώφλια διαμορφώνονται ανάλογα με το χαμηλότερο και υψηλότερο δείγμα του τρέχοντος παραθύρου. Η θεμελιώδης συχνότητα βρίσκεται άμεσα ως το αντίστροφο του ρυθμού εναλλαγής από την υψηλή τάση στη χαμηλή. Η απλότητα αυτού του μοντέλου είναι και το μειονέκτημά του, καθώς για να αντιμετωπίσει την πολυπλοκότητα της μουσικής χροιάς χρειάζεται περαιτέρω βελτιώσεις [Lang, 2003].

Autocorrelation

Οι μέθοδοι αυτοσυσχέτισης λειτουργούν συγκρίνοντας, σε επίπεδο δειγμάτων, τις ομοιότητες μεταξύ τμημάτων του σήματος με τμήματα του ίδιου σήματος που έχουν μετατεθεί χρονικά [Klapuri, 2000]. Η συνάρτηση αυτοσυσχέτισης (ACF - Autocorrelation Function) ενός διακριτού σήματος x(k) με μήκος ακολουθίας K ορίζεται ως εξής:

$$r(n)=\frac{1}{K}\sum_{k=0}^{K-n-1}x(k)x(k+n) \qquad (2.16)$$

Όταν το σήμα εισόδου είναι περιοδικό αυτή η συνάρτηση παράγει κορυφές στα ακέραια πολλαπλάσια της περιόδου όπως φαίνεται στην Εικόνα 2.2.



Εικόνα 2.2: Κυματομορφή περιοδικού σήματος εισόδου και της συνάρτησης αυτοσυσχέτισης.

Οι μέθοδοι αυτοσυσχέτισης είναι αποτελεσματικοί στην ανίχνευση μεσαίων και χαμηλών συχνοτήτων και συνήθως χρησιμοποιούνται για επεξεργασία λόγου, όπου το φάσμα των συχνοτήτων είναι περιορισμένο. Ωστόσο, όταν χρησιμοποιούνται σε μουσικά σήματα το υπολογιστικό κόστος αυξάνεται σημαντικά.

Για να μειωθεί το υπολογιστικό κόστος η μέθοδος της αυτοσυσχέτισης μπορεί να μεταφερθεί στο πεδίο της συχνότητας ως εξής:

$$r(n)=\frac{1}{K}\sum_{k=0}^{K-1}\left|\mathrm{X}(k)\right|^2\cos\left(\frac{2\pi nk}{K}\right) \qquad (2.17)$$

Εκφραζόμενη με αυτόν τον τρόπο η συνάρτηση αυτοσυσχέτισης αποτελεί πλέον μία φασματική προσέγγιση, η οποία επιλέγει την θεμελιώδη συχνότητα σταθμίζοντας τα φασματικά στοιχεία σύμφωνα με τη θέση τους. Σε γενικές γραμμές, οι μέθοδοι αυτοσυσχέτισης δείχνουν να είναι αρκετά ανθεκτικές στο θόρυβο αλλά παρουσιάζουν μειονεκτήματα στην αντιμετώπιση των φασματικών ιδιαιτεροτήτων, τόσο των σημάτων μουσικής όσο και των σημάτων λόγου.

YIN

Ο αλγόριθμος YIN [de Cheveign´e and Kawahara, 2002] είναι ένα ακόμη μοντέλο αναγνώρισης συχνότητας που έγκειται στο πεδίο του χρόνου, ο οποίος παρέχει έναν αποτελεσματικό τρόπο αναγνώρισης και εξαγωγής υποψήφιων συχνοτήτων από μεγάλο φασματικό εύρος. Με την παραδοχή ότι το άθροισμα χτ - χt+τ παίρνει την ελάχιστη τιμή

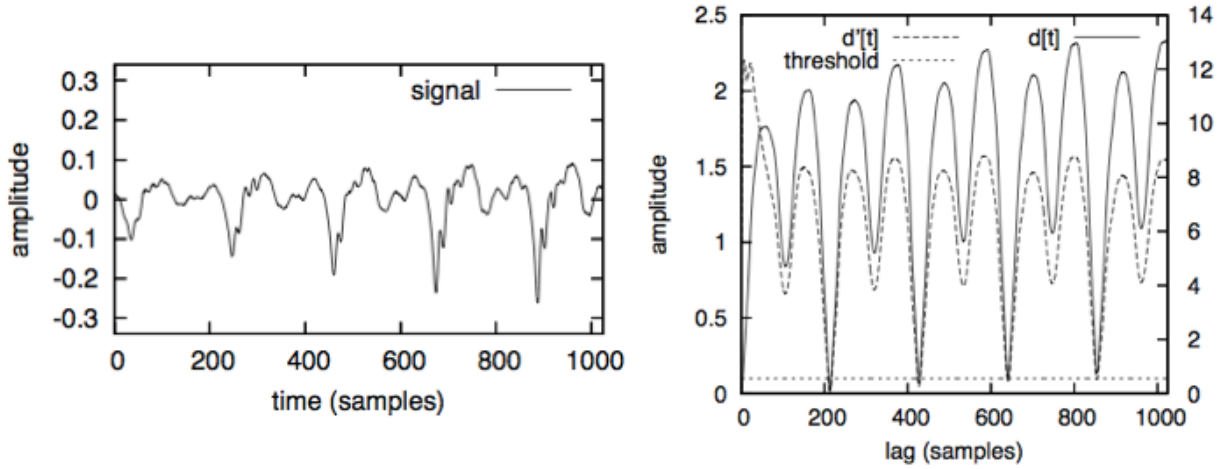
όταν η περίοδος του σήματος είναι τ, η μαθηματική σχέση που διέπει τη συνάρτηση διαφοράς τετραγώνου dt(τ) είναι η εξής:

$$d_t(\tau)=\sum_{j=t+1}^{t+W}(x_j-x_{j+\tau})^2 \qquad (2.18)$$

όπου W είναι το μέγεθος του παραθύρου ανάλυσης, t ο χρόνος, και τ η χρονική καθυστέρηση. Η συνάρτηση YIN είναι η ομαλοποιημένη τετραγωνική διαφορά που προκύπτει διαιρώντας την τετραγωνική διαφορά οποιασδήποτε δεδομένης χρονικής καθυστέρησης, με τη μέση διαφορά των τιμών με μικρότερη χρονική καθυστέρηση από αυτή:

$$d'_t(\tau)=\begin{cases}1, & if\ \tau=0\\ d_t(\tau)\Big/\left[\frac{1}{\tau}\sum_{j=1}^{\tau}d_t(j)\right], & otherwise.\end{cases} \qquad (2.19)$$

Στη συνέχεια επιλέγεται η ελάχιστη τιμή κάτω από ένα ορισμένο όριο, η οποία αντιστοιχεί στην περίοδο της νότας που εκτελείται όπως φαίνεται στην Εικόνα 2.3. Αν δε βρεθεί καμία τιμή κάτω από το όριο, το τμήμα που εξετάζεται θεωρείται "άφωνο". Μεγαλώνοντας την τιμή του ορίου σε αρκετές περιπτώσεις επιτυγχάνεται η αναγνώριση κάποιων συχνοτήτων που διαφορετικά θα είχαν απορριφθεί εσφαλμένα, αλλά από την άλλη μεριά η χρήση μεγάλων τιμών οδηγεί συχνά σε λάθη αναγνώρισης των συχνοτήτων με διαφορά οκτάβας. Μια εναλλακτική τεχνική για τις περιπτώσεις που καμία ελάχιστη τιμή δεν είναι κάτω από το ορισμένο όριο, είναι να χρησιμοποιείται η ελάχιστη τιμή του dt για να εκτιμηθεί η περίοδος. Ωστόσο, αυτή η τεχνική ενώ αυξάνει ένα μικρό ποσοστό των τμημάτων που εσφαλμένα αναγνωρίζονται ως "άφωνα" (voiced), αυξάνει κατά πολύ περισσότερο το ποσοστό των "άφωνων" τμημάτων που εσφαλμένα αναγνωρίζονται ως "έμφωνα" (unvoiced).



Εικόνα 2.3: Κυματομορφή σήματος εισόδου, της συνάρτησης διαφοράς τετραγώνου d[t] και της συνάρτησης YIN d'[t].

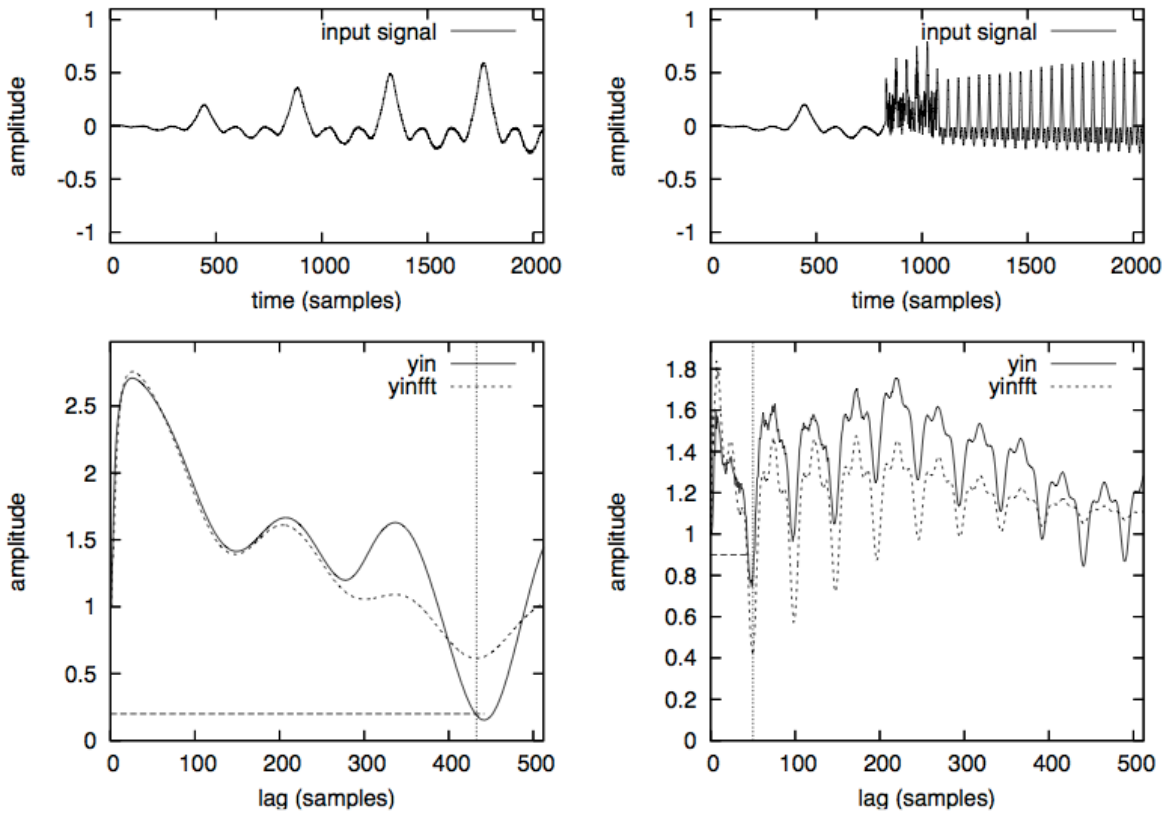
Το γεγονός ότι η συνάρτηση YIN χρειάζεται μόνο την ελάχιστη τιμή κάτω από το ορισμένο όριο για να υπολογίσει την περίοδο του σήματος, την καθιστά πιο γρήγορη και πιο αποδοτική, από υπολογιστική σκοπιά, σε σχέση με τη συνάρτηση αυτοσυσχέτισης. Ωστόσο, η συνολική αποδοτικότητα της όταν πρόκειται για εφαρμογές πραγματικού χρόνου εξαρτάται άμεσα από τη θεμελιώδη συχνότητα του τμήματος προς ανάλυση, κάτι που γίνεται αισθητό στις χαμηλές συχνότητες και στις περιοχές σιωπής όπου το υπολογιστικό κόστος αυξάνεται.

Με βάση το μοντέλο του YIN έχει σχεδιαστεί μια νέα μέθοδος αναγνώρισης τονικού ύψους γνωστή ως Spectral domain YIN. Πρόκειται για μια μέθοδο η οποία υπολογίζεται στο πεδίο της συχνότητας και σχεδιάστηκε με στόχο να έχει μικρότερο υπολογιστικό κόστος από την μέθοδο YIN [Brossier, 2006]. Η συνάρτηση Spectral domain YIN είναι η τετραγωνική διαφορά μεταξύ του συνολικού φασματικού μέτρου του τρέχοντος παραθύρου επεξεργασίας και μίας φασικά μετατοπισμένης έκδοσής του:

$$\hat{d}_t(\tau)=\frac{2}{N}\sum_{k=0}^{N/2+1}\left|(1-e^{2\pi kj\tau/\mathrm{N}})\,\mathrm{X}_t[k]\right|^2 \qquad (2.20)$$

Εκτός από το μειωμένο υπολογιστικό κόστος που έχει η φασματική έκδοση της μεθόδου λόγω του υπολογισμού της μέσω δύο FFT (Fast Fourier Transformation), επιπλέον δε χρειάζεται την εφαρμογή ορίου για την επιλογή της περιόδου, περιορίζοντας

την έρευνά της στην εύρεση της ελάχιστης τιμής του d’t. Η χρονική έκδοση της συνάρτησης YIN παρουσιάζει διάφορες κοιλίες κοντινού πλάτους στα ακέραια πολλαπλάσια τις περιόδου, ενώ η φασματική έκδοση παρουσιάζει τη χαμηλότερη κοιλία στην περίοδο του σήματος, με τις υπόλοιπες να έχουν εμφανώς διαφορετικό πλάτος Εικόνα 2.4.



Εικόνα 2.4: Κυματομορφή σήματος εισόδου της συνάρτησης YIN και της συνάρτησης YINFFT. Η κάθετη γραμμή τέμνει την κυματομορφή yinfft στην ελάχιστη τιμή της και η οριζόντια αναπαριστά το όριο επιλογής της μεθόδου YIN.

2.2.4 Post-processing

Οι φασματικές ιδιομορφίες του σήματος και στοιχεία όπως κάποια διαμόρφωση πλάτους ή συχνότητας, είναι πιθανό να προκαλέσουν λανθασμένες αλλαγές του εκτιμώμενου τονικού ύψους, που δεν αντιστοιχούν σε αντιληπτές αλλαγές της τονικότητας από τον ακροατή. Στόχος του βήματος της μετεπεξεργασίας είναι η μείωση των λανθασμένων συχνοτικών εκτιμήσεων του συστήματος, με τρόπο ώστε να μην επιφέρει επιπλέον χρονική καθυστέρηση στο σύστημα και να μην παρεμβαίνει στην ομαλή εναλλαγή μεταξύ νοτών που αντιστοιχούν σε αληθείς τονικές αλλαγές.

Μια κοινή προσέγγιση για την εξομάλυνση της εξόδου του συστήματος είναι η συνέλιξή του με την κρουστική απόκριση ενός φίλτρου διέλευσης χαμηλών συχνοτήτων. Το φιλτράρισμα αυτό είναι επιτυχές στην εξάλειψη της παραμόρφωσης "jitter"' και του θορύβου [Hess, 1984], αλλά δεν αποτρέπει τις λανθασμένες εκτιμήσεις τονικού ύψους. Για να μειωθούν οι λανθασμένες εκτιμήσεις, το φίλτρο αυτό μπορεί να χρησιμοποιηθεί σε συνδιασμό με την τεχνική μέσης τιμής (median based approach), αποθηκεύοντας μερικές τονικές εκτιμήσεις σε ένα μικρό χρονικό διάστημα μετά από τα onset:

$$P_{note} = median\,(P_q, P_{q+1}, ..., P_{q+\delta}) \tag{2.21}$$

όπου δ, είναι ο αριθμός των μελλοντικών παρατηρήσεων και έχει άμεσο αντίκτυπο στην καθυστέρηση που προστίθεται στο σύστημα. Κατά αυτόν τον τρόπο μειώνονται οι πιθανότητες λαθών όταν αναλύεται η ατάκα μιας νότας με transient στοιχεία (μικρής διάρκειας μεταβατικός ήχος που υπάρχει συχνά στην αρχή μιας νότας, ο οποίος αποτελείται από τυχαίες μη αρμονικές συνιστώσες), καθώς και όταν εναλλάσσονται οι εντάσεις των αρμονικών συνιστωσών κατά τη χρονική εξέλιξη μιας νότας. Επιπροσθέτως, χρησιμοποιώντας αυτήν την τεχνική απορρίπτονται λανθασμένα onset, που μπορεί να προκαλούνται από θόρυβο, αναπνοές κ.τ.λ. σε περιοχές σιωπής ή παύσης.

3 Υλοποίηση

3.1 Γλώσσα Προγραμματισμού

Η υλοποίηση της εφαρμογής έγινε με τη γλώσσα προγραμματισμού C++. Η C++ είναι γλώσσα υψηλού επιπέδου, μεταγλωττιζόμενη, και μπορεί να υλοποιήσει διάφορα προγραμματιστικά παραδείγματα, όπως αυτά του διαδικασιοστρεφούς και αντικειμενοστραφούς προγραμματισμού.

Η επιλογή της γλώσσας έγινε δεδομένης της απαίτησης που παρουσιάζουν οι εφαρμογές πραγματικού χρόνου για όσο το δυνατόν μικρότερες χρονικές καθυστερήσεις. Στη συγκεκριμένη περίπτωση η C++, όντας μία από τις πιο γρήγορες γλώσσες προγραμματισμού, επιλέχτηκε για να καλύψει αυτήν ακριβώς την ανάγκη. Επιπλέον, η δημοτικότητα της γλώσσας προσφέρει φορητότητα (με μικρές ή καθόλου αλλαγές στον κώδικα), παρότι είναι μεταγλωττιζόμενη, μέσω του εύρους των μεταγλωττιστών που υπάρχουν και απευθύνονται σε όλες τις πλατφόρμες λογισμικών συστημάτων. Ένα ακόμη πλεονέκτημα που παρουσιάζει η επιλογή της C++, σύμφωνα με τις ανάγκες της προκείμενης εφαρμογής, είναι η ύπαρξη μεγάλου φάσματος προγραμματιστικών βιβλιοθηκών ανοιχτού κώδικα (open source) με αντικείμενο την επεξεργασία ήχου.

3.2 Προγραμματιστικά εργαλεία (Βιβλιοθήκες - APIs)

Για την υλοποίηση της εφαρμογής αξιοποιήθηκαν διάφορες προγραμματιστικές βιβλιοθήκες (APIs - Application Programming Interfaces). Μία προγραμματιστική βιβλιοθήκη είναι ένα αυτόνομο κομμάτι κώδικα, υλοποιημένο για να προσφέρει έτοιμες λειτουργίες σε έναν προγραμματιστή, επεκτείνοντας τις δυνατότητες μια γλώσσας προγραμματισμού. Οι βιβλιοθήκες αυτές μπορεί να αποτελούνται από ρουτίνες, δομές δεδομένων, κλάσεις και μεταβλητές, και έχουν συνήθως συγκεκριμένο αντικείμενο.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν τέσσερις προγραμματιστικές βιβλιοθήκες ανοιχτού κώδικα: η PortAudio, η Aubio, η wxWidgets και η RtMidi.

3.2.1 PortAudio API

Η PortAudio είναι μια δωρεάν, ανοιχτού κώδικα, προγραμματιστική βιβλιοθήκη, υλοποιημένη στη γλώσσα προγραμματισμού C. Η βιβλιοθήκη παρέχει έλεγχο σε πραγματικό χρόνο επί των μονάδων εισόδου/εξόδου ήχου (audio I/O) του υπολογιστή. Μέσω της PortAudio μπορούν να γραφτούν προγράμματα στην C ή C++ για την αναπαραγωγή και ηχογράφηση ήχου. Η PortAudio υποστηρίζει Windows, Unix/Linux kai Mac OS X.

Η δημιουργία της PortAudio προτάθηκε από τον Ross Bencina στα πλαίσια του music-dsp mailing list, με στόχο την ανάπτυξη μιας δωρεάν προγραμματιστικής βιβλιοθήκης διαχείρισης ήχου, που θα επέτρεπε την ανταλλαγή προγραμμάτων μεταξύ προγραμματιστών που χρησιμοποιούν διαφορετικά λειτουργικά συστήματα. Η PortAudio αναπτύχθηκε και συντηρείται από τον Ross Bencina και τον Phill Burk, με την παράλληλη υποστήριξη πολλών ακόμα προγραμματιστών. Περισσότερες πληροφορίες υπάρχουν στην ιστοσελίδα www.portaudio.com.

Η επιλογή της PortAudio έγινε με κριτήρια την φορητότητα της εφαρμογής και την κάλυψη των αναγκών σύλληψης του ήχου.

3.2.2 Aubio API

Η Aubio είναι μια δωρεάν, ανοιχτού κώδικα, προγραμματιστική βιβλιοθήκη, υλοποιημένη στη γλώσσα προγραμματισμού C. Η βιβλιοθήκη παρέχει τη δυνατότητα ανάκτησης δεδομένων από μουσικό ηχητικό σήμα σε πραγματικό χρόνο. Πιο συγκεκριμένα η Aubio προσφέρει εργαλεία κατάτμησης του σήματος σε ηχητικά συμβάντα, αναγνώρισης τονικού ύψους, αναγνώρισης ρυθμού, μετατροπής audio σε midi, διάφορα ψηφιακά φίλτρα και phase vocoder. Η Aubio υποστηρίζει Windows, Unix/Linux kai Mac OS X.

Η Aubio αναπτύχθηκε από τον Paul M. Brossier και παρουσιάστηκε το 2006 σαν κομμάτι της διδακτορικής του διατριβής. Στόχος του ήταν η συγκέντρωση και η προσαρμογή υφιστάμενων αλγορίθμων ανάκτησης μουσικών δεδομένων σε μία προγραμματιστική βιβλιοθήκη, έτσι ώστε να υπάρχει η δυνατότητα χρήσης τους σε

εφαρμογές πραγματικού χρόνου. Περισσότερες πληροφορίες υπάρχουν στην ιστοσελίδα www.aubio.org.

Η επιλογή της Aubio έγινε λόγω των ποικίλων αλγορίθμων που προσφέρει, οι οποίοι βρίσκουν άμεση εφαρμογή στην ανάπτυξη του λογισμικού της παρούσας εργασίας.

3.2.3 RtMidi API

Η RtMidi είναι μια δωρεάν, ανοιχτού κώδικα, προγραμματιστική βιβλιοθήκη, υλοποιημένη στη γλώσσα προγραμματισμού C++, η οποία παρέχει έλεγχο σε πραγματικό χρόνο επί των μονάδων εισόδου/εξόδου MIDI (I/O) του υπολογιστή. Η RtMidi είναι μια βιβλιοθήκη που σχεδιάστηκε να λειτουργεί σε όλα τα δημοφιλή λειτουργικά συστήματα υποστηρίζοντας Windows, Unix/Linux kai Mac OS X.

Η RtMidi αναπτύχθηκε από τον Gary P. Scavone. Αποτελείται από δύο κλάσεις RtMidiIn και RtMidiOut και στοχεύει στην απλούστευση της επικοινωνίας του υπολογιστή με συσκευές και λογισμικά Midi. Περισσότερες πληροφορίες υπάρχουν στην ιστοσελίδα www.music.mcgill.ca/~gary/rtmidi.

Η επιλογή της RtMidi έγινε για να καλύψει τις ανάγκες της εφαρμογής στη σύνθεση και διαχείριση Midi μηνυμάτων, καθώς και για την δυνατότητα υποστήριξης πολλαπλών λειτουργικών συστημάτων.

3.2.4 wxWidgets API

Η wxWidgets (παλιότερα γνωστά ως wxWindows) είναι μια δωρεάν, ανοιχτού κώδικα, προγραμματιστική βιβλιοθήκη, υλοποιημένη στη γλώσσα προγραμματισμού C++. Παρέχει ένα πλήρες πακέτο προγραμματιστικών εργαλείων για την υλοποίηση οποιασδήποτε γραφικής διεπαφής χρήστη (GUI - Graphical User Interface). Η wxWidgets υποστηρίζει Windows, Linux και Mac OS X και ξεχωρίζει από τις υπόλοιπες βιβλιοθήκες δημιουργίας GUI, κάνοντας τις εφαρμογές της να είναι εμφανισιακά εφάμιλλες με τις εφαρμογές του εκάστοτε λειτουργικού συστήματος στο οποίο εκτελούνται. Ακόμη, υποστηρίζει λειτουργικά συστήματα κινητών συσκευών συμπεριλαμβανομένων των Windows Mobile, iPhone SDK και embedded GTK+. Μέσω

των πολλών γλωσσικών συνδέσεων “language bindings” που έχουν δημιουργηθεί για την wxWidgets, προσφέρεται η δυνατότητα υλοποίησης εφαρμογών GUI σε διάφορες γλώσσες προγραμματισμού όπως Python, Java, Perl, Ruby κ.α.

Η wxWidgets δημιουργήθηκε το 1992 από τον Julian Smart στο Πανεπιστήμιο του Εδιμβούργου, με στόχο τη δημιουργία φορητών εφαρμογών μεταξύ των λειτουργικών συστημάτων Unix και Windows. Έκτοτε η βιβλιοθήκη αναπτύσσεται συνεχώς με την υποστήριξη μεγάλου αριθμού προγραμματιστών. Περισσότερες πληροφορίες υπάρχουν στην ιστοσελίδα www.wxwidgets.org.

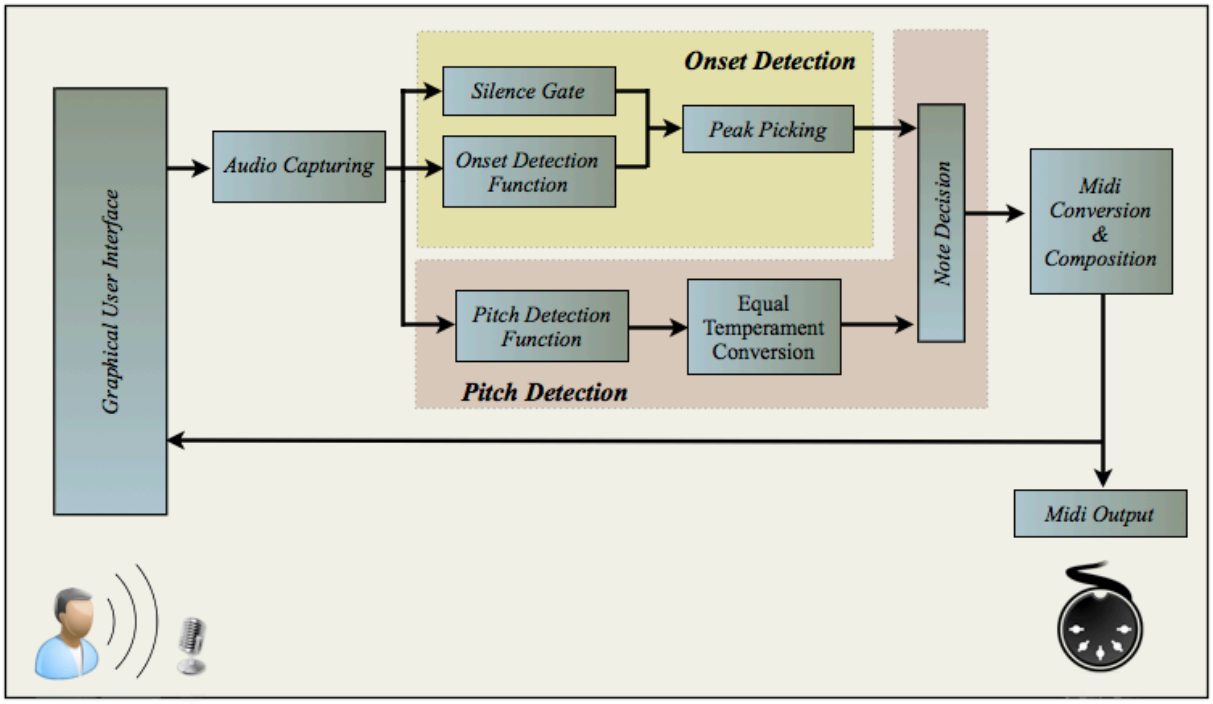
Η επιλογή της βιβλιοθήκης έγινε βάσει της φορητότητας των εφαρμογών που παράγονται μέσω αυτής, της ομοιότητας του συντακτικού περιεχομένου της με τη C++, καθώς και την ομοιότητα της εμφάνισης των παραγόμενων εφαρμογών, με τη γνήσια εμφάνιση των εφαρμογών του εκάστοτε λογισμικού στο οποίο εκτελούνται.

3.3 Επισκόπηση λειτουργιών προγράμματος

Το λογισμικό αποτελείται από τις εξής λειτουργίες:

- Λήψη του Ήχου (Audio Capturing).
- Ανάκτηση Μουσικής Πληροφορίας (Music Information Retrieval).
- Μετατροπή Μουσικής Πληροφορίας σε MIDI.
- Επεξεργασία και Παρουσίαση δεδομένων μέσω γραφικής διεπαφής χρήστη.
- Σύνθεση και Αποστολή MIDI μηνυμάτων στην έξοδο της κάρτας ήχου.

Οι διάφορες μονάδες που επιτελούν τις λειτουργίες του λογισμικού επικοινωνούν μεταξύ τους, διαμορφώνοντας το συνολικό σύστημα που παρουσιάζεται στην εικόνα Εικόνα 3.1.



Εικόνα 3.1: Διάγραμμα ροής.

3.4 Ιεραρχία τάξεων

Για την υλοποίηση της εφαρμογής δημιουργήθηκαν οι πέντε παρακάτω τάξεις:

- τάξη AudioCapturer
- τάξη PitchDetector
- τάξη MidiInterface
- τάξη BasicDrawPanel
- τάξη GUI

Στις επόμενες υποενότητες αναλύονται όλες οι τάξεις της εφαρμογής και ορισμένες λειτουργίες τους. Για περισσότερες πληροφορίες αναφορικά με την υλοποίηση των τάξεων, ο αναγνώστης μπορεί να μελετήσει τα αρχεία του πηγαίου κώδικα της εφαρμογής ή το documentation που δημιουργήθηκε μέσω Doxygen. Εκτός από το φάκελο διανομής της εργασίας, το Documentation έχει αναρτηθεί και στην ιστοσελίδα https://dl.dropbox.com/u/49553054/Documentation.zip, από όπου ο αναγνώστης μπορεί να το κατεβάσει.

3.4.1 AudioCapturer

Η τάξη AudioCapturer είναι υπεύθυνη για τη σύλληψη του ηχητικού σήματος από την κάρτα ήχου του υπολογιστή και συμπεριλαμβάνει τις εξής μεθόδους:

- Initialize_PA() - μέθοδος που αναλαμβάνει την αρχικοποίηση απαραίτητων εσωτερικών δομών δεδομένων και μεταβλητών της PortAudio.

- OpenStream_PA(PaStreamCallback *audioCallback) - μέθοδος που ανοίγει την μονάδα ηχητικής ροής. Η υλοποίησή της παρουσιάζεται στην Εικόνα 3.2. Η μεταβλητή audioCallback αναφέρεται σε μια μέθοδο Callback, η οποία παρουσιάζεται στην Εικόνα 3.3. Αυτή η μέθοδος καλείται αυτόματα από την κάρτα ήχου του υπολογιστή κάθε φορά που είναι έτοιμη να επεξεργαστεί τα επόμενα στοιχεία του σήματος εισόδου.

```
void AudioCapturer::OpenStream_PA(PaStreamCallback
                                        *audioCallback )
{
      PaStreamParameters micStreamParams;
      micStreamParams.device = device_index;
      micStreamParams.channelCount = CHANNEL_COUNT;
      micStreamParams.sampleFormat = SAMPLE_FORMAT;
      micStreamParams.suggestedLatency = SUGGESTED_LATENCY;
      micStreamParams.hostApiSpecificStreamInfo = NULL;
      paUserData mydata;
      PaError err = Pa_OpenStream(&micStream,
                                        &micStreamParams,
                                        NULL,
                                        sample_rate,
                                        buffer_size,
                                        paNoFlag,
                                        audioCallback,
                                        &mydata);
}
```

Εικόνα 3.2: Υλοποίηση μεθόδου OpenStream_PA().

- StartStream_PA() - μέθοδος που ξεκινάει τη ροή ήχου.

- StopStream_PA() - μέθοδος που σταματάει τη ροή ήχου.

- CloseStream_PA() - μέθοδος που κλείνει τη μονάδα ηχητικής ροής.

- Terminate_PA() - μέθοδος που καλείται να καταστρέψει τα εσωτερικά στοιχεία της PortAudio από τη μνήμη.

```
int paAubioCallback(  const void *inputBuffer,
                      void *outputBuffer,
                      unsigned long framesPerBuffer,
                      const PaStreamCallbackTimeInfo* timeInfo,
                      PaStreamCallbackFlags statusFlags,
                      void *userData)
{
    (void) outputBuffer;
    const float *aubioIn = (const float*)inputBuffer;
    aubio.Process_AubioPitch(aubioIn);
    return 0;
}
```

Εικόνα 3.3: Υλοποίηση μεθόδου paAubioCallBack().

3.4.2 PitchDetector

Η τάξη PitchDetector είναι υπεύθυνη για την επεξεργασία του ηχητικού σήματος, ώστε να ανακτηθούν οι απαραίτητες μουσικές πληροφορίες για τη σύνθεση των MIDI μηνυμάτων. Σε αυτή την τάξη συμπεριλαμβάνονται οι εξής μέθοδοι:

- Initialize_AubioPitch() - μέθοδος αρχικοποίησης απαραίτητων εσωτερικών δομών δεδομένων και μεταβλητών της Aubio.

- Process_AubioPitch(const float *input) - μέθοδος που καλείται από τη μέθοδο Callback για να επεξεργαστεί το σήμα εισόδου, ώστε να στείλει πληροφορίες στην τάξη σύνθεσης MIDI μηνυμάτων και στη γραφική διεπαφή. Η μεταβλητή input αναφέρεται σε έναν πίνακα τιμών που περιέχει δείγματα του σήματος εισόδου.

- Pass_Gui_Pointer(Gui *guiP) - μέθοδος που δίνει πρόσβαση στα στοιχεία της τάξης GUI από την τάξη AudioPitch.

- Terminate_AubioPitch() - μέθοδος που διαγράφει στοιχεία που έχουν δημιουργηθεί και αποθηκευτεί στη μνήμη.

Στις εικόνες παρακάτω φαίνονται τα βήματα που ακολουθούνται στη μέθοδο Process_AubioPitch(). Στην Εικόνα 3.4(a) παρουσιάζεται το πρώτο τμήμα του κώδικα στο οποίο μετά την αρχικοποίηση κάποιων μεταβλητών, υπάρχει ένας βρόγχος for(). Αρχικά μέσα στο βρόγχο υπολογίζεται το άθροισμα των τετραγωνισμένων τιμών του πλάτους των δειγμάτων εισόδου επί την τιμή της μεταβλητής gain, η οποία εξαρτάται από τη θέση του ρυθμιστικού έντασης του σήματος εισόδου (input volume slider) της διεπαφής χρήστη.

Στη συνέχεια μέσω της if() και της μεταβλητής counter, η οποία αυξάνεται κατά μία μονάδα σε κάθε επανάληψη του βρόγχου της for(), ελέγχεται η συχνότητα δειγματοληψίας που χρησιμοποιείται για να υλοποιηθεί η κυματομορφή του σήματος.

Τέλος, στην τελευταία γραμμή κώδικα μέσω της συνάρτησης fvec_write_sample(), οι τιμές των δειγμάτων του σήματος εισόδου καταγράφονται σε έναν καινούργιο πίνακα τιμών (buffer) που καλείται ibuf.

```
void PitchDetector::Process_AubioPitch(const float *input)
{
    unsigned int j;
    unsigned int pos = 0;
    int level = 0;

    for (j=0;j<paudio.buffer_size;j++) {
           counter++;
           counter2++;
           counter3++;
           sum = sum +pow((input[j]*gui->gain),2);
           if (counter== round(paudio.sample_Rate/1000)){
                 gui->AddWaveSample((int)(input[j]*gui->gain));
                 counter = 0;
           }
           fvec_write_sample(ibuf, input[j]*(gui->gain/100.0), 0, pos);

                                  ...
```

Εικόνα 3.4(a): Υλοποίηση μεθόδου Process_AubioPitch().

Στην Εικόνα 3.4(b) η πρώτη if() ελέγχει την τιμή της μεταβλητής pos, η οποία αυξάνει κατά μία μονάδα σε κάθε βρόχο της for(). Με αυτόν τον τρόπο η μεταβλητή pos μας επιτρέπει να ξέρουμε ποιο δείγμα του buffer διαβάζει το πρόγραμμα. Εφόσον πρόκειται για το μεσαίο δείγμα, το πρόγραμμα εκτελεί τις γραμμές κώδικα μέσα στην if() συμπεριλαμβανομένων των ακόλουθων συναρτήσεων επεξεργασίας της Aubio:

- aubio_proc_do() - Η συνάρτηση αυτή δέχεται τα νέα δείγματα εισόδου και μέσω ενός μετασχηματισμού Fourier υπολογίζει και επιστρέφει στην έξοδό της το πλάτος και τη φάση του σήματος.

- aubio_onsetdetection() - Πρόκειται για τη βασική συνάρτηση εντοπισμού onset. Η συνάρτηση αυτή δέχεται στην είσοδο τον αλγόριθμο αναγνώρισης onset που έχει επιλεχτεί από το χρήστη και τις πληροφορίες από την έξοδο της συνάρτησης aubio_proc_do(). Στην έξοδό της στέλνει πληροφορίες που αφορούν στη διαδικασία επιλογής των κορυφών μέσω της μεταβλητής onset.

```
                                        ...

(2*)    if (pos == hopSize-1) {
            aubio_pvoc_do (pv,ibuf, fftgrain);
            aubio_onsetdetection(odFunction,fftgrain, onset);
            if (usedoubled) {
                  aubio_onsetdetection(odFunction2,fftgrain, onset2);
                  onset->data[0][0] *= onset2->data[0][0];
            }
            isonset = aubio_peakpick_pimrt(onset,peakPicker);
            pitch = aubio_pitchdetection(pitchdet,ibuf);
            gui->freq=(int)pitch;
            midiPitch = round(aubio_freqtomidi(pitch)) + gui->pitch_offset;
            if (median){
                  uint_t i = 0;
                  for (i = 0; i < note_buffer->length - 1; i++) {
                        note_buffer->data[0][i] = note_buffer->data[0][i+1];
                  }
                  note_buffer->data[0][note_buffer->length - 1] = midiPitch;
            }
            curlevel = aubio_level_detection(ibuf, silence);

                                        ...
                                                                (2*)  2xTabsIn
```

Εικόνα 3.4(b): Υλοποίηση μεθόδου Process_AubioPitch().

- aubio_peakpick_pimrt() - Συνάρτηση που επιστρέφει στην έξοδό της πληροφορία για την ύπαρξη ή απουσία εγκεκριμένης κορυφής η οποία αποθηκεύεται στη μεταβλητή isonset.

- aubio_pitchdetection() - Είναι η βασική συνάρτηση που αναλαμβάνει την αναγνώριση τονικού ύψους. Στην είσοδό της δέχεται έναν πίνακα με τις τιμές πλάτους των δειγμάτων του σήματος εισόδου, πληροφορίες για τη δειγματοληψία, το μέγεθος του παραθύρου επεξεργασίας, τον αριθμό των καναλιών και τον επιλεγμένο αλγόριθμο αναγνώρισης. Στην έξοδό της επιστρέφει το εκτιμώμενο τονικό ύψος.

- aubio_level_detection() - Η συνάρτηση αυτή δέχεται έναν πίνακα με τις τιμές πλάτους των δειγμάτων του σήματος εισόδου και τη στάθμη έντασης του ήχου που αντιστοιχεί σε συνθήκες ησυχίας. Στην έξοδό της επιστρέφει τη στάθμη (rms) έντασης του σήματος ή την τιμή 1 όταν ανιχνεύεται ησυχία.

Παρακάτω, μέσω της δεύτερης if() ελέγχεται η επιλογή του χρήστη για την εφαρμογή ενός επιπλέον αλγορίθμου εντοπισμού onset και μέσω της τρίτης if() ελέγχεται εάν ο χρήστης έχει επιλέξει να χρησιμοποιεί την τεχνική μέσης τιμής, όπως περιγράφεται στην Παράγραφο 2.2.4.

Στην Εικόνα 3.4(c) παρουσιάζεται το κομμάτι του κώδικα που είναι υπεύθυνο για την τελική επιλογή της νότας που θα σταλεί στην MIDI έξοδο του συστήματος. Η πρώτη if() ελέγχει το περιεχόμενο της μεταβλητής isonset και στην περίπτωση που το αποτέλεσμα του ελέγχου είναι αληθές, τότε διακρίνονται δύο επιμέρους περιπτώσεις.

Στην πρώτη περίπτωση ελέγχεται μέσω της δεύτερης if() εάν η στάθμη έντασης είναι κατώτερη από το επίπεδο ησυχίας. Τότε το περιεχόμενο της μεταβλητής isonset αντιστρέφεται και μέσω μίας τρίτης if() ελέγχεται αν ο χρήστης έχει επιλέξει να χρησιμοποιεί την τεχνική μέσης τιμής. Εφόσον και αυτό είναι αληθές, η τιμή της μεταβλητής isready μηδενίζεται, ενώ ταυτόχρονα στέλνεται ένα μήνυμα note off για τη νότα που είναι αποθηκευμένη στη μεταβλητή προσωρινής αποθήκευσης tempNote.

```
                                        ...
(3*)  if (isonset) {
            if (curlevel == 1.) {
                  isonset = 0;
                  if (median) isready = 0;
                        midi.ConvertToMidi("NOTE_OFF",tempNote,curlevel);
            }else {
                  if (median) {
                        isready = 1;
                  }else {
                        midi.ConvertToMidi("NOTE_OFF",tempNote,curlevel);
                        midi.ConvertToMidi("NOTE_ON",midiPitch,curlevel);
                        midi.ConvertToVisNote(midiPitch-gui->pitch_offset);
                        wxString mystring(midi.musicNote);
                        wxCommandEvent eventDisp(wxEVT_C,DISPLAY_INFO_ID);
                        eventPost_D.SetString(mystring);
                        gui->GetEventHandler()->AddPendingEvent(eventDisp);
                        tempNote = midiPitch;
                  }
            }
      }else {
            if (median) {
                  if (isready > 0)
                        isready++;
                  if (isready == median){
                        midi.ConvertToMidi("NOTE_OFF",tempNote,curlevel);
                        tempNote = vec_median(note_buffer);
                        if (tempNote>45 && tempNote<108){
                              midi.ConvertToMidi("NOTE_ON",tempNote,curlevel);
                              midi.ConvertToVisNote(tempNote-gui->pitch_offset);
                              wxString mystring(midi.musicNote);
                              wxCommandEvent eventDisp(wxEVT_C,DISPLAY_INFO_ID);
                              eventPost_D.SetString(mystring);
                              gui->GetEventHandler()->AddPendingEvent(eventDisp);
                        }
                  }
            }
       }
                                        ...
                                                                    (3*)   3xTabsIn
```

Εικόνα 3.4(c): Υλοποίηση μεθόδου Process_AubioPitch().

Στη δεύτερη περίπτωση, εφόσον η στάθμη του σήματος δεν αντιστοιχεί στο επίπεδο ησυχίας, ελέγχεται μέσω της τέταρτης if() αν χρησιμοποιείται η τεχνική μέσης τιμής. Εφόσον αυτό αληθεύει, τότε στη μεταβλητή isready αποθηκεύεται η τιμή 1. Στην αντίθετη περίπτωση, στέλνεται ένα μήνυμα note off για τη νότα που είναι αποθηκευμένη στην μεταβλητή tempNote και ένα μήνυμα note on για τη νότα που είναι αποθηκευμένη στη μεταβλητή midiPitch. Επιπλέον, δημιουργείται και στέλνεται στη τάξη GUI ένα event το οποίο είναι υπεύθυνο για την ανανέωση των γραφικών δεδομένων κειμένου της

διεπαφής. Τέλος, στη μεταβλητή tempNote αποθηκεύεται η τρέχουσα τιμή της μεταβλητής midiPitch.

Επιστρέφοντας στην περίπτωση που ο έλεγχος της μεταβλητής isonset έχει ψευδές αποτέλεσμα, τα βήματα ελέγχου που ακολουθούν αφορούν στην περίπτωση που χρησιμοποιείται η τεχνική μέσης τιμής. Αρχικά, ελέγχεται η τιμή της μεταβλητής isready και αν είναι μεγαλύτερη του μηδενός, η τιμή της αυξάνεται κατά μία μονάδα. Στο δεύτερο βήμα ελέγχεται το αν η τιμή της στις επόμενες επαναλήψεις του βρόγχου θα φτάσει να γίνει ίση με το πλήθος των υποψήφιων τιμών που χρησιμοποιούνται στην τεχνική μέσης τιμής. Σε αυτήν την περίπτωση στέλνεται ένα μήνυμα note off για τη νότα που είναι αποθηκευμένη στη μεταβλητή tempNote και αμέσως μετά στέλνεται ένα μήνυμα note on για τη νότα που προκύπτει από την τεχνική μέσης τιμής. Τέλος, ακολουθούνται ξανά τα βήματα ανανέωσης της γραφικής διεπαφής σύμφωνα με τα πιο πρόσφατα δεδομένα.

Στην Εικόνα 3.4(d) φαίνεται το τελευταίο κομμάτι του κώδικα, στο οποίο ελέγχεται η συχνότητα με την οποία στέλνονται στη διεπαφή χρήστη κάποιες πληροφορίες. Στην πρώτη if(), μέσω της μεταβλητής counter2, ελέγχεται η συχνότητα τροφοδοσίας τιμών μέσης τετραγωνικής ρίζας του πλάτους (rms) των δειγμάτων εισόδου, προς το αντικείμενο απεικόνισης της στάθμης εισόδου (input level). Στη δεύτερη if() η μεταβλητή counter3 ελέγχει τη συχνότητα με την οποία στέλνεται στη γραφική διεπαφή, ένα event κίνησης του ποντικιού.

```
                                        ...
                    pos = -1;
                }
                pos++;
        }
        if (counter2== round(paudio.sample_rate/20)){
                level=sqrt(sum/1024*2);
                gui->AddGaugeSample(level);
                sum=0;
                counter2=0;
        }
        if (counter3 == round(paudio.sample_rate/15)){
                wxMouseEvent   eventPost(wxEVT_MOTION);
                gui->GetEventHandler()->AddPendingEvent(eventPost);
                counter3=0;
        }
}
```

Εικόνα 3.4(d): Υλοποίηση μεθόδου Process_AubioPitch().

Τέτοιου είδους events, σε κανονικές συνθήκες, στέλνονται αυτόματα όταν ο χρήστης κουνήσει το ποντίκι, αλλά στην προκειμένη περίπτωση τα παράγουμε μέσω κώδικα χωρίς στην πραγματικότητα να κουνιέται το ποντίκι. Αυτό γίνεται για να επιλυθεί ένα σφάλμα (bug), που προέκυψε με το ρυθμό ανανέωσης της οθόνης μεταξύ Mac OS X και wxWidgets. Περισσότερες πληροφορίες σχετικά με αυτό το πρόβλημα υπάρχουν στην ιστοσελίδα: http://forums.wxwidgets.org/viewtopic.php?f=23&t=9004&hilit=refresh+Refresh+Mac.

3.4.3 MidiInterface

Η τάξη MidiInterface αναλαμβάνει τη σύνθεση MIDI μηνυμάτων βάσει των εξαγόμενων μουσικών πληροφοριών, καθώς και την αποστολή τους στη MIDI έξοδο του υπολογιστή. Στην τάξη αυτή συμπεριλαμβάνονται οι εξής μέθοδοι:

- Initalize_RtMidi() - μέθοδος που δημιουργεί ένα αντικείμενο της κλάσης RtMidiOut, το οποίο επιτρέπει την πρόσβαση στις MIDI εξόδους του υπολογιστή.

- Convert_To_Midi(string msgType, double midiNote, double velocity) - Μέθοδος που μετατρέπει σε πραγματικό χρόνο τις εξαγόμενες μουσικές πληροφορίες σε μορφή MIDI, ώστε να συνθέσει και να στείλει στην έξοδο ένα MIDI Note On ή Note off μήνυμα. Η μεταβλητή midiNote περιέχει τη MIDI νότα, η velocity τη MIDI ένταση και η msgType καθορίζει τον τύπο του μηνύματος που θα σταλεί στην έξοδο (NoteOn ή NoteOff). Στην Εικόνα 3.5 παρατίθεται ο κώδικας υλοποίησης της μεθόδου.

- Program_Change(int midiInst) - μέθοδος που στέλνει ένα MIDI program change μήνυμα, ώστε να αλλάξει το μουσικό όργανο που αναπαράγει τις MIDI νότες. Η μεταβλητή MidiInst περιέχει πληροφορίες για το επιλεγμένο μουσικό όργανο.

- Convert_To_MusicNote(int midiNote) - Μέθοδος που μετατρέπει τη MIDI νότα σε γραφική πληροφορία (π.χ. A4, C3, D2), για να παρουσιαστεί στη γραφική διεπαφή.

- Terminate_RtMidi() - Μέθοδος που καλείται να διαγράψει το αντικείμενο RtMidiOut όταν δεν είναι πλέον απαραίτητο.

```
void MidiInterface::Convert_To_Midi(string msgType,double
                                    midiNote, double velocity)
{
     note = round(midiNote);
     amp  = 127+(int)floor(velocity);

     if (msgType=="NOTE_ON"){
             message.push_back(144);
             message.push_back(note);
             message.push_back(amp);
             midiout->sendMessage(&message);
     }

     if (msgType=="NOTE_OFF"){
             message.push_back(144);
             message.push_back(note);
             message.push_back(0);
             midiout->sendMessage(&message);
     }
     message.clear();
}
```

Εικόνα 3.5: Υλοποίηση μεθόδου Convert_To_Midi().

3.4.4 BasicDrawPanel

Η τάξη αυτή αναλαμβάνει τη γραφική αναπαράσταση της κυματομορφής του σήματος εισόδου στο κεντρικό παράθυρο της εφαρμογής και αποτελεί κομμάτι της γραφικής διεπαφής. Στην τάξη BasicDrawPanel χρησιμοποιούνται οι τάξεις wxPaintDc και wxPaintEvent, οι οποίες κληρονομούνται από τα wxWidgets, ώστε να διαχειρίζεται paint events (ένα paint event στέλνεται όταν τα περιεχόμενα ενός παραθύρου χρειάζεται να ξανά-ζωγραφιστούν). Επιπλέον, στην τάξη αυτή συμπεριλαμβάνονται οι μέθοδοι render() και addPoint(), των οποίων η υλοποίηση φαίνεται στην Εικόνα 3.6.

Η addPoint() καλείται μέσω της τάξης PitchDetector, για να αποθηκεύσει τα δείγματα εισόδου στη μνήμη, και η render() καλείται όταν διαχειρίζονται paint events, ώστε να ενώσει τα σημεία που καθορίζονται από την addPoint() και να ζωγραφίσει την κυματομορφή.

```
void DrawPanel::render(wxDC&  dc)
{
     if (firstrun){

       for (int i=0; i<dc.GetSize().GetWidth(); i++)
          mydeque.push_front(0);
          firstrun = false;
     }
     dc.SetPen( wxPen( wxColor(0,255,0), 1.5,wxSOLID ) );
     zeroPoint = dc.GetSize().GetHeight() / 2;
     for (unsigned int i=0; i<(mydeque.size()-1); i++){
     dc.DrawLine(i,zeroPoint-mydeque[i],i+1,zeroPoint-mydeque[i+1]);}
}


void DrawPanel::addPoint(int point)
{
     mydeque.push_front(point);
     mydeque.pop_back();
     Refresh();
}
```

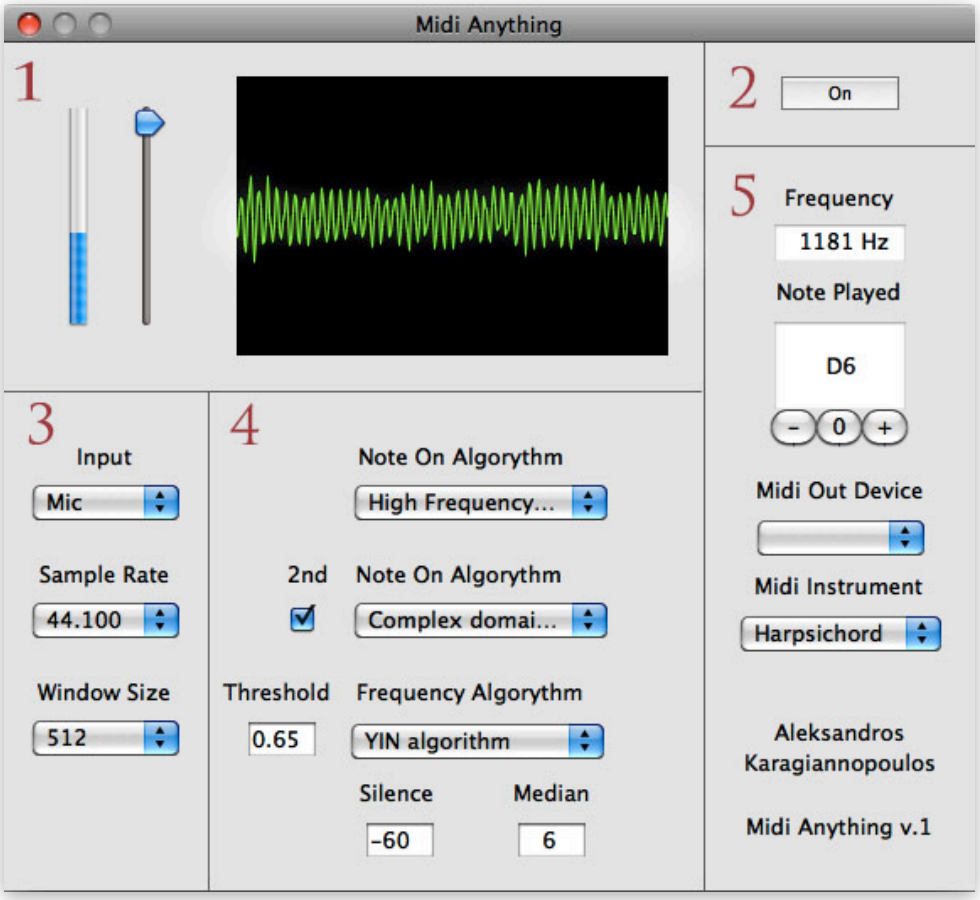
Εικόνα 3.6: Υλοποίηση των μεθόδων render και addPoint.

3.4.5 GUI

Η τάξη GUI αναλαμβάνει όλη τη δημιουργία της κεντρικής γραφικής διεπαφής χρήστη. Μέσω της βιβλιοθήκης wxWidgets παρέχονται όλες οι πληροφορίες και τα εργαλεία που χρειάζονται για τη δημιουργία των μενού, των κουμπιών και γενικότερα όλων των λειτουργιών της εφαρμογής. Η τάξη GUI αποτελείται από 1000 περίπου γραμμές κώδικα και είναι εκτός του αντικειμένου μελέτης της εργασίας αυτής, οπότε κρίθηκε σκόπιμο να μην αναλυθεί περαιτέρω. Ωστόσο, όπως και με τις υπόλοιπες τάξεις, ο αναγνώστης μπορεί να μελετήσει τον πηγαίο κώδικα από την ιστοσελίδα που υποδείχθηκε στην Παράγραφο 3.4.

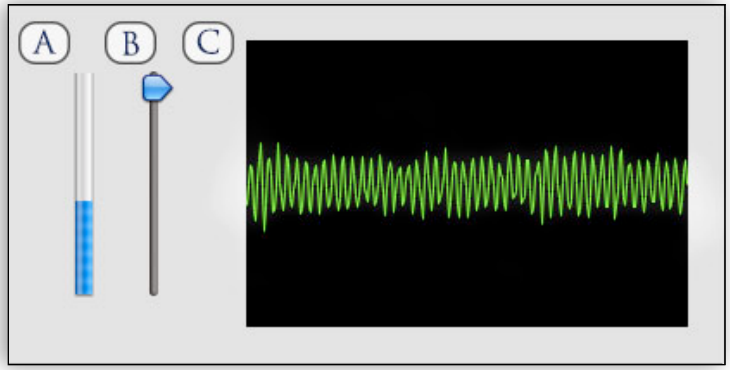
4 Περιγραφή εφαρμογής

Η εφαρμογή αποτελείται από ένα κεντρικό παράθυρο, το οποίο παρουσιάζεται στην Εικόνα 4.1. Τα επιμέρους τμήματα του παραθύρου έχουν αριθμηθεί, από το 1 έως το 5, για τη διευκόλυνση του χρήστη.



Εικόνα 4.1: Το κεντρικό παράθυρο της εφαρμογής.

- Το πρώτο τμήμα περιλαμβάνει τρία αντικείμενα για την απεικόνιση και τη ρύθμιση στοιχείων του σήματος εισόδου, όπως φαίνεται στην Εικόνα 4.2. Το πρώτο αντικείμενο από αριστερά είναι ένας μετρητής της στάθμης εισόδου (input gauge), το δεύτερο είναι ένα ρυθμιστικό της έντασης εισόδου (input volume slider) και το τρίτο είναι ένα πλαίσιο στο οποίο παρουσιάζεται η κυματομορφή του σήματος εισόδου.

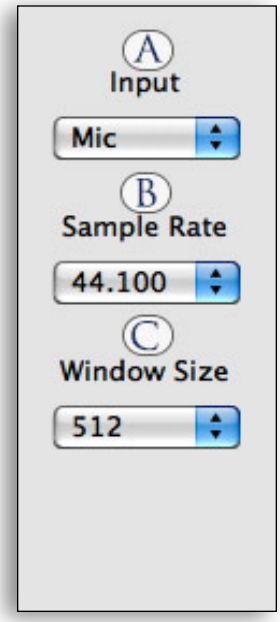


Εικόνα 4.2: Το πρώτο τμήμα της γραφικής διεπαφής. a) Input gauge b) Input Slider c) Waveform.

• Το δεύρο τμήμα αποτελείται εξ ολοκλήρου από ένα κουμπί δύο καταστάσεων (toggle button), με τις επιλογές ενεργό/ανενεργό (on/off) και αφορά στην κατάσταση λειτουργίας της εφαρμογής.

• Το τρίτο τμήμα, όπως φαίνεται στην Εικόνα 4.3, έχει τρία κουμπιά αναδυόμενης λίστας (combo box), μέσω των οποίων ο χρήστης μπορεί να διαλέξει τη συσκευή εισόδου, τη συχνότητα δειγματοληψίας και το μέγεθος του παραθύρου επεξεργασίας.

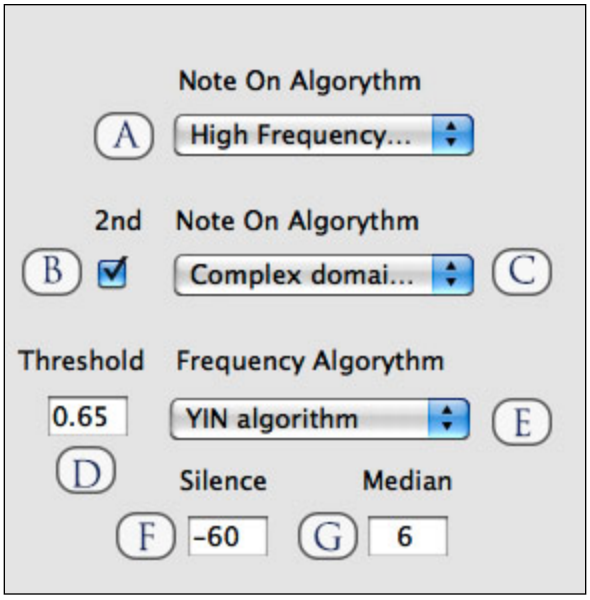
a. Input (Line-In - Built-In Mic).
b. Sample Rate (22.050, 44.100, 96.000 Samples/Sec).
c. Window Size (256, 512, 1024 Samples).



Εικόνα 4.3: **Το τρίτο τμήμα της γραφικής διεπαφής.**
a) Input b) Sample Rate c) Window Size.

• Το τέταρτο τμήμα απαρτίζεται από τρία κουμπιά αναδυόμενης λίστας, τρία κουτιά εισαγωγής κειμένου (Text Control Box) και ένα κουμπί δύο καταστάσεων, των οποίων οι λειτουργίες αναλύονται παρακάτω. Το τέταρτο τμήμα παρουσιάζεται παρακάτω στην Εικόνα 4.4.

a. Note On Algorithm (Επιλογές αναδυόμενης λίστας: *Energy, Spectral Difference, High Frequency Content, Complex domain distance, Phase deviation, Kullback-Liebler distance, Modified Kullback-Liebler*). Το κουμπί αυτό δίνει τη δυνατότητα στο χρήστη να επιλέξει ποιον αλγόριθμο εντοπισμού onset θα χρησιμοποιήσει.
b. Κουμπί ενεργοποίησης/απενεργοποίησης δεύτερου αλγορίθμου note on. Αυτό το κουμπί δίνει τη δυνατότητα στο χρήστη να χρησιμοποιήσει έναν επιπλέον αλγόριθμο εντοπισμού onset.
c. 2nd Note On Algorithm (Επιλογές αναδυόμενης λίστας: *Energy, Spectral Difference, High Frequency Content, Complex domain distance, Phase deviation, Kullback-Liebler distance, Modified Kulback/Liebler*). Το κουμπί αυτό δίνει τη δυνατότητα στο χρήστη να επιλέξει ποιος θα είναι ο επιπλέον αλγόριθμος εντοπισμού onset που θα χρησιμοποιηθεί.
d. Κουτί εισαγωγής κειμένου, μέσω του οποίου ο χρήστης ρυθμίζει το Threshold (Εύρος ορίου: 0,10 έως 1,20).
e. Frequency Algorithm (Επιλογές αναδυόμενης λίστας: YIN algorithm, Multi comb filter, Schmitt trigger, Fast comb filter, Spectral YIN). Το κουμπί αυτό δίνει τη δυνατότητα στο χρήστη να επιλέξει τον αλγόριθμο αναγνώρισης τονικού ύψους που θα χρησιμοποιήσει.
f. Κουτί εισαγωγής κειμένου, μέσω του οποίου ορίζεται η στάθμη ησυχίας του χώρου. (Εύρος στάθμης: -100 έως 0).
g. Κουτί εισαγωγής κειμένου που καθορίζει πόσα δείγματα καθυστέρησης χρησιμοποιούνται στην τεχνική της μέσης τιμής, όπως υποδείχθηκε στην Παράγραφο 2.2.4 (Εύρος τιμών: 0 έως 20).

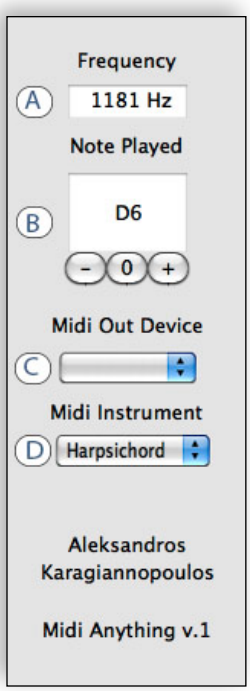


Εικόνα 4.4: Το τέταρτο τμήμα της γραφικής διεπαφής.

• Το πέμπτο τμήμα, που φαίνεται στην Εικόνα 4.5, περιλαμβάνει τέσσερα επιμέρους αντικείμενα που επιτελούν τις εξής λειτουργίες.

a. Κουτί παρουσίασης κειμένου στο οποίο αναγράφεται η τελευταία συχνότητα που αναλύθηκε σε Hertz.
b. Κουτί παρουσίασης κειμένου στο οποίο αναγράφεται η νότα (στο Αγγλικό σύστημα) που αντιστοιχεί στη συχνότητα που αναλύθηκε. Τα τρία κουμπιά από κάτω επιτρέπουν στο χρήστη να επέμβει στηννότα που θα παράγει η γεννήτρια MIDI, αυξάνοντας ή μειώνοντας την τονικότητα με το διάστημα της επιλογής του. Τα κουμπιά με τα πρόσημα + και - διαμορφώνουν την τονικότητα κατά ένα ημιτόνιο, ενώ το κουμπί με τον αριθμό 0, επαναφέρει την τονικότητα στην αρχική της τιμή.
c. Κουμπί αναδυόμενης λίστας το οποίο επιτρέπει στο χρήστη να επιλέξει τη Midi έξοδο που θέλει να χρησιμοποιήσει.
d. Κουμπί αναδυόμενης λίστας το οποίο επιτρέπει στο χρήστη να επιλέξει το MIDI μουσικό όργανο που θα αναπαράγει τις εξαγόμενες νότες (Επιλογές αναδυόμενης λίστας: *Acoustic Grand Piano, Bright Acoustic Piano, Electric Grand Piano, Harpsichord, Clavinet, Celesta, Vibraphone, Xylophone, Marimba, Dulcimer, Drawbar Organ, Percussive Organ, Accordion, Church Organ, Harmonica, Acoustic Guitar (nylon), Acoustic Guitar (steel), Electric Guitar (clean), Electric Guitar (jazz),*

Guitar Harmonics, Violin, Viola, Cello, Contrabass, Orchestral Harp, Alto Sax, Baritone Sax, Oboe, English Horn, Clarinet, Piccolo, Flute, Blown Bottle, Whistle, Ocarina, Sitar, Banjo, Acoustic Bass, Steel Drums, Applause).



Εικόνα 4.5: Το πέμπτο τμήμα της γραφικής διεπαφής.

5 Πειραματική αξιολόγηση

5.1 Σύνοψη Πειραμάτων

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

- Τρομπέτα
- Πιάνο
- Ανθρώπινη φωνή
- Βιολί
- Κιθάρα

Η επιλογή των μουσικών οργάνων έγινε για να αξιολογηθεί η εφαρμογή ως προς την αξιοπιστία των αποτελεσμάτων της, χρησιμοποιώντας ηχητικές πηγές με διαφορετική χροιά αλλά και περιβάλλουσα έντασης.

Όλα τα πειράματα έγιναν με αναπαραγωγή ενός αρχείου ήχου μονοφωνικής μελωδίας, όπου με καλώδιο συνδέθηκε η έξοδος ήχου (audio out) του υπολογιστή με την είσοδο μικροφώνου (line - in) του ίδιου υπολογιστή. Για τα πειράματα προστέθηκε μία επιπλέον γραμμή κώδικα, με την οποία οι εξαγόμενες νότες εκτυπώθηκαν σε ένα αρχείο κειμένου.

Τα αρχεία ήχου του πιάνου και του βιολιού δημιουργήθηκαν στο Finale, μέσω των MIDI οργάνων που διαθέτει. Τα υπόλοιπα αρχεία ήχου προέρχονται από ηχογραφήσεις φυσικών οργάνων, που πραγματοποιήθηκαν σε studio. Όλες οι παρτιτούρες των αρχείων που προβάλλονται παρακάτω έχουν δημιουργηθεί στο Finale.

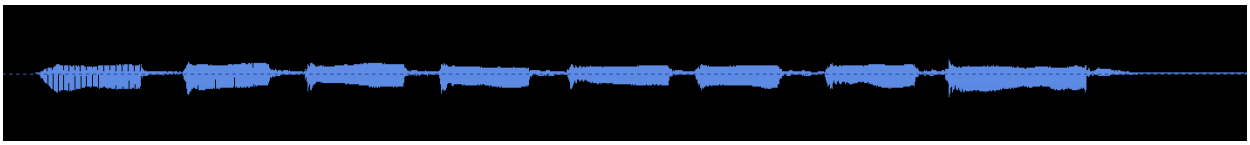
Η μέθοδος διεξαγωγής των πειραμάτων που ακολουθήθηκε είναι η εξής: Σε πρώτο στάδιο ρυθμίστηκαν οι παράμετροι που αφορούν τα onset, δηλαδή οι αλγόριθμοι Onset Detection και το threshold. Στη συνέχεια, αφού βελτιστοποιήθηκαν οι ανιχνεύσεις των onset, ρυθμίστηκαν οι παράμετροι που αφορούν στην αναγνώριση της συχνότητας, όπως ο αλγόριθμος Pitch Detection ή ο αριθμός των δειγμάτων καθυστέρησης (median), και στις περιπτώσεις όπου αυτό δεν ήταν αρκετό, δοκιμάστηκαν εναλλακτικές τιμές δειγματοληψίας και μεγέθους του παραθύρου επεξεργασίας.

5.2 Πείραμα 1 - Τρομπέτα

Στο παράδειγμα 1 δόθηκε βάρος στα αποτελέσματα που μπορεί να έχουν οι διαφορετικές τιμές των παραμέτρων και στην διαδικασία επιλογής των καταλληλότερων, ώστε να έχουμε τα καλύτερα δυνατά αποτελέσματα.

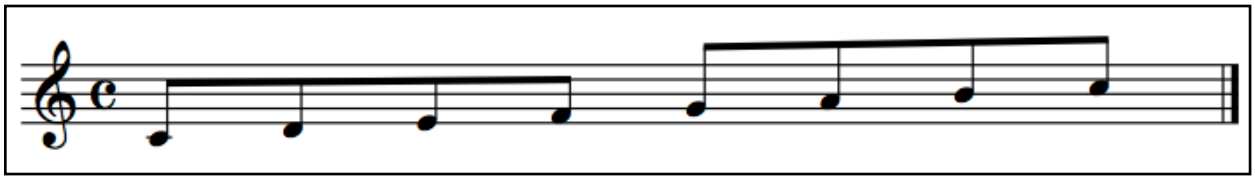
Για την τρομπέτα εξετάζονται δύο ηχογραφημένες μελωδίες. Η πρώτη είναι η φυσική κλίμακα του Ντο, και η δεύτερη είναι τα πρώτα μέτρα από το κομμάτι Playing Love (Trumpet Version) - The Legend of 1900 soundtrack, όπως εκτελέστηκαν από ένα μαθητή τρομπέτας.

Το πρώτο αρχείο ήχου που χρησιμοποιήθηκε είναι το Trumpet_c.wav και η κυματομορφή του είναι:



Εικόνα 5.2.1: Κυματομορφή αρχείου ήχου Trumpet_c.wav.

Η παρτιτούρα του αρχείου είναι:



Εικόνα 5.2.2 Παρτιτούρα αναφοράς του αρχείου ήχου Trumpet_c.wav.

Στον Πίνακα 5.2.1 στην πρώτη σειρά, παρατίθενται οι αριθμοί των MIDI νοτών που αντιστοιχούν στην παραπάνω παρτιτούρα, με τη σειρά που αυτές εκτελούνται. Στη δεύτερη σειρά καταγράφονται οι αντίστοιχες νότες στο Αγγλικό σύστημα, όπως ακριβώς τις παρουσιάζει και η εφαρμογή (στην αναπαράσταση τους από την εφαρμογή χρησιμοποιείται το σύμβολο # είτε πρόκειται για υφέσεις είτε για διέσεις, π.χ. η νότα Σι ύφεση θα αναπαρασταθεί γραφικά σαν Λα δίεση, οπότε για την αποφυγή συγχύσεων κρίθηκε σκόπιμο να αναφέρονται και στους πίνακες των πειραμάτων με τον ίδιο τρόπο).

Νότες αναφοράς								
MIDI	60	62	64	65	67	69	71	72
English	C4	D4	E4	F4	G4	A4	B4	C5

Πίνακας 5.2.1: Νότες αναφοράς του αρχείου ήχου Trumpet_c.wav.

Οι τιμές των παραμέτρων ορίστηκαν ως εξής:

Sample Rate	44,100
Window Size	512
Threshold	0.10
Silence	-60
Median	5
Note On Algorithm	Energy
2nd Note On Algorithm	-
Pitch Detection Algorithm	Yin

Τα αποτελέσματα του πειράματος παρατίθενται στον Πίνακα 5.2.2, όπου με πράσινο χρώμα επισημαίνονται οι νότες που αναγνωρίστηκαν σωστά, με κίτρινο οι νότες που εμφανίστηκαν λανθασμένα πολλαπλές φορές ή καθόλου, και με κόκκινο οι νότες που αναγνωρίστηκαν με λανθασμένη τονικότητα.

Εξαγόμενες νότες								
MIDI	60 60 60	62 62	64 64	65 65	67	69 69 69	71 71 71 71 71	85 85
English	C4 C4 C4	D4 D4 D4	E4 E4	F4 F4	G4	A4 A4 A4	B4 B4 B4 B4 B4	C6 # C6 #

Πίνακας 5.2.2: Εξαγόμενες νότες από το αρχείο Trumpet_c.wav.

Από τη σύγκριση των νοτών αναφοράς με τις εξαγόμενες νότες, παρατηρούμε ότι στις περισσότερες των περιπτώσεων η συχνότητα αναγνωρίστηκε σωστά. Ωστόσο, προέκυψαν πολλά λάθη με τα onset. Πιο συγκεκριμένα οι αποκλίσεις είναι:

- Η πρώτη νότα της μελωδίας (Ντο) εμφανίστηκε λανθασμένα τρεις φορές.
- Η δεύτερη νότα της μελωδίας (Ρε) εμφανίστηκε λανθασμένα τρεις φορές.
- Η τρίτη νότα της μελωδίας (Μι) εμφανίστηκε λανθασμένα δύο φορές.
- Η τέταρτη νότα της μελωδίας (Φα) εμφανίστηκε λανθασμένα δύο φορές.
- Η έκτη νότα της μελωδίας (Λα) εμφανίστηκε λανθασμένα τρεις φορές.
- Η έβδομη νότα της μελωδίας (Σι) εμφανίστηκε λανθασμένα πέντε φορές.
- Η όγδοη νότα της μελωδίας (Ντο) δεν αναγνωρίστηκε επιτυχώς.

Συνολικά, από τις 8 νότες, μόνο 1 αναγνωρίστηκε με επιτυχία. Στη συνέχεια αλλάζοντας τις παραμέτρους σε:

Παράμετροι	**Τιμές Παραμέτρων**
Sample Rate	44,100
Window Size	512
Threshold	0.10
Silence	-60
Median	5
Note On Algorithm	Spectral Difference
2nd Note On Algorithm	-
Pitch Detection Algorithm	Yin

παίρνουμε τα αποτελέσματα που παρατίθενται στον Πίνακα 5.2.3.

Εξαγόμενες νότες								
MIDI	60 60	62 62	64	65 65	67	69 69	71 71 71 71	85
English	C4 C4	D4 D4	E4	F4 F4	G4	A4 A4	B4 B4 B4 B4	C6 #

Πίνακας 5.2.3: Εξαγόμενες νότες από το αρχείο Trumpet_c.wav.

Από τη σύγκριση των αποτελεσμάτων παρατηρούμε ότι υπάρχει μείωση των ψευδών onset. Πιο συγκεκριμένα οι αποκλίσεις αυτή τη φορά είναι:

- Η πρώτη νότα της μελωδίας (Ντο) εμφανίστηκε λανθασμένα δύο φορές.
- Η δεύτερη νότα της μελωδίας (Ρε) εμφανίστηκε λανθασμένα τρεις φορές.
- Η τέταρτη νότα της μελωδίας (Φα) εμφανίστηκε λανθασμένα δύο φορές.
- Η έκτη νότα της μελωδίας (Λα) εμφανίστηκε λανθασμένα δύο φορές.
- Η έβδομη νότα της μελωδίας (Σι) εμφανίστηκε λανθασμένα τέσσερις φορές.
- Η όγδοη νότα της μελωδίας (Ντο) δεν αναγνωρίστηκε επιτυχώς.

Συνολικά, από τις 8 νότες, μόνο 2 αναγνωρίστηκαν με επιτυχία. Συνεχίζοντας κατά αυτόν τον τρόπο παρατηρούμε ότι ο αλγόριθμος εντοπισμού onset, που παρουσιάζει τα λιγότερα λάθη χρησιμοποιώντας σταθερά το ίδιο threshold, είναι ο Complex Domain. Ωστόσο, με το Threshold ρυθμισμένο στην ελάχιστη τιμή (0.10), τα αποτελέσματα περιλαμβάνουν ακόμα πολλαπλά ψευδή onset. Οπότε το επόμενο βήμα είναι η σταδιακή αύξηση της τιμής του threshold, έως ότου αυτά εξαλειφθούν. Ρυθμίζοντας λοιπόν τις παραμέτρους σε:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	512
Threshold	0.20
Silence	-60
Median	5
Note On Algorithm	Complex domain Distance
2nd Note On Algorithm	-
Pitch Detection Algorithm	Yin

παίρνουμε τα αποτελέσματα που παρατίθενται στον Πίνακα 5.2.4.

Εξαγόμενες νότες								
MIDI	59	62	64	65	67	69	71 71	85
English	B3	D4	E4	G4	F4	A4	B4 B4	C6 #

Πίνακας 5.2.4: Εξαγόμενες νότες από το αρχείο Trumpet_c.wav.

Από τη σύγκριση των αποτελεσμάτων βλέπουμε ότι τα πολλαπλά onset σχεδόν εξαλείφθηκαν. Πιο συγκεκριμένα οι αποκλίσεις αυτή τη φορά είναι:

- Η πρώτη νότα της μελωδίας (Ντο) δεν αναγνωρίστηκε επιτυχώς.
- Η έβδομη νότα της μελωδίας (Σι) εμφανίστηκε δύο φορές.
- Η όγδοη νότα της μελωδίας (Ντο) δεν αναγνωρίστηκε επιτυχώς.

Συνολικά, από τις 8 νότες, οι 5 αναγνωρίστηκαν με επιτυχία. Αυξάνοντας το threshold σταδιακά, παρατηρούμε ότι για την τιμή 0.30 δεν εμφανίζονται πλέον διπλά onset. Συνεχίζουμε λοιπόν αλλάζοντας τον αλγόριθμο αναγνώρισης της συχνότητας. Οπότε ρυθμίζοντας τις παραμέτρους σε:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	512
Threshold	0.30
Silence	-60
Median	5
Note On Algorithm	Complex domain Distance
2nd Note On Algorithm	-
Pitch Detection Algorithm	Multi Comb Filter

παίρνουμε τα αποτελέσματα που παρατίθενται στον Πίνακα 5.2.5.

Εξαγόμενες νότες								
MIDI	59	62	64	65	67	69	71	71
English	B3	D4	E4	G4	F4	A4	B4	B4

Πίνακας 5.2.5: Εξαγόμενες νότες από το αρχείο Trumpet_c.wav.

Από τη σύγκριση των αποτελεσμάτων παρατηρούμε ότι τα τονικά αποτελέσματα δε βελτιώθηκαν ιδιαίτερα. Πιο συγκεκριμένα οι αποκλίσεις αυτή τη φορά είναι:

- Η πρώτη νότα της μελωδίας (Ντο) δεν αναγνωρίστηκε επιτυχώς έχοντας την ίδια απόκλιση με προηγουμένως.
- Η όγδοη νότα της μελωδίας (Ντο) δεν αναγνωρίστηκε επιτυχώς, αλλά έχει αρκετά μικρότερη απόκλιση σε σύγκριση με προηγουμένως.

Συνολικά, από τις 8 νότες, οι 6 αναγνωρίστηκαν με επιτυχία. Συνεχίζοντας τις δοκιμές με διαφορετικούς αλγόριθμους αναγνώρισης συχνότητας, παρατηρούμε ότι ο Spectral Yin παρουσιάζει τα καλύτερα αποτελέσματα, παρουσιάζοντας μία μόνο απόκλιση στην

τελευταία νότα (Ντο). Επόμενο βήμα είναι να αλλάξουμε την τιμή της παραμέτρου Median, δηλαδή τα δείγματα καθυστέρησης. Οπότε ρυθμίζοντας τις παραμέτρους σε:

Παράμετροι	**Τιμές Παραμέτρων**
Sample Rate	44,100
Window Size	512
Threshold	0.30
Silence	-60
Median	9
Note On Algorithm	Complex domain Distance
2nd Note On Algorithm	-
Pitch Detection Algorithm	Spectral Yin

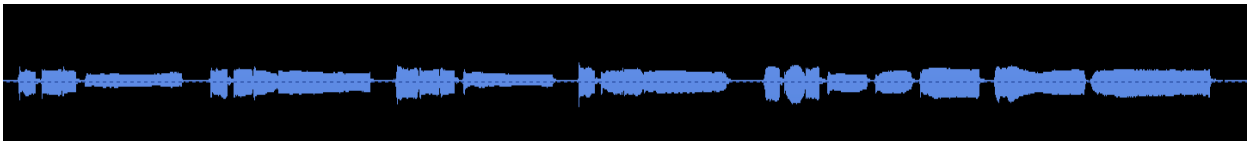
παίρνουμε τα αποτελέσματα που παρατίθενται στον Πίνακα 5.2.6.

Εξαγόμενες νότες								
MIDI	60	62	64	65	67	69	71	72
English	C4	D4	E4	G4	F4	A4	B4	C5

Πίνακας 5.2.6: Εξαγόμενες νότες από το αρχείο Trumpet_c.wav.

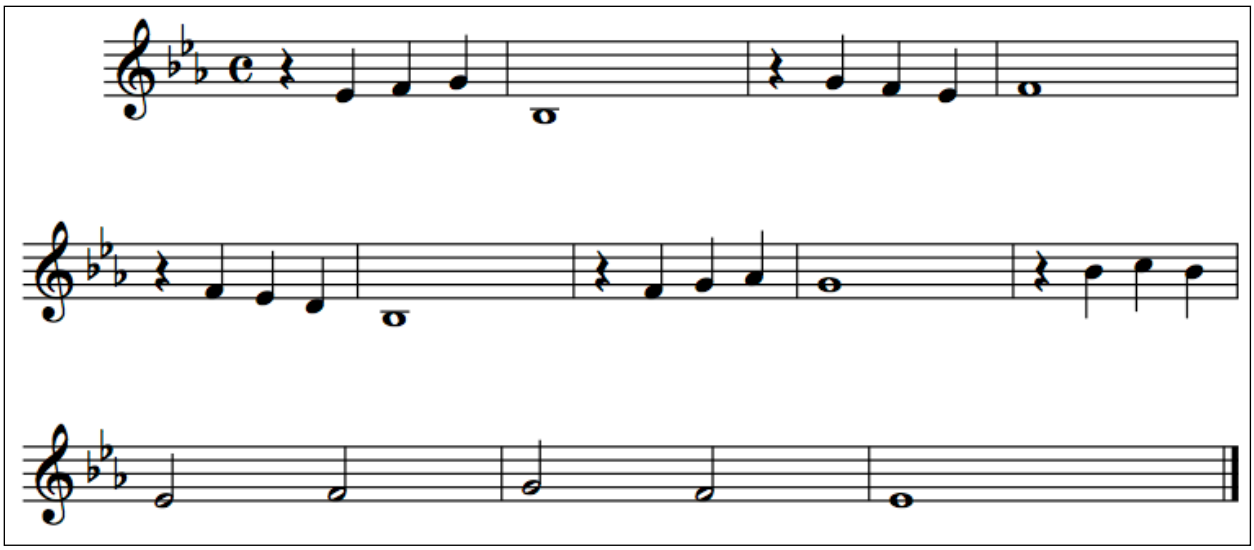
Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών βλέπουμε ότι, με αυτές τις παραμέτρους, όλες οι νότες αναγνωρίστηκαν με επιτυχία. Αν λάβουμε υπόψη τα χαρακτηριστικά της τρομπέτας, όπως απότομες ατάκες, πλούσιες αρμονικές και τονικότητα που βασίζεται στον εκτελεστή (ο οποίος στην προκειμένη περίπτωση ήταν αρχάριος), μπορούμε να πούμε ότι τα αποτελέσματα είναι απολύτως ικανοποιητικά.

Το δεύτερο αρχείο ήχου που χρησιμοποιήθηκε είναι το Trumpet.wav και η κυματομορφή του είναι:



Εικόνα 5.2.3: Κυματομορφή αρχείου ήχου Trumpet.wav.

Η παρτιτούρα του αρχείου είναι:



Εικόνα 5.2.3 Παρτιτούρα αναφοράς του αρχείου ήχου Trumpet.wav.

Στον Πίνακα 5.2.7 παρακάτω παρατίθενται οι νότες αναφοράς.

Νότες Αναφοράς												
MIDI (12 πρώτες νότες)	63	65	67	58	67	65	63	65	65	63	62	58
English (12 πρώτες νότες)	D4#	F4	G4	A3#	G4	F4	D4#	F4	F4	D4#	D4	A3#
MIDI (Υπόλοιπες νότες)	65	67	68	67	70	72	70	63	65	67	65	63
English (Υπόλοιπες νότες)	F4	G4	G4#	G4	A4#	C5	A4#	D4#	F4	G4	F4	D4#

Πίνακας 5.2.7: Νότες αναφοράς του αρχείου ήχου Trumpet.wav.

Κατά τη διεξαγωγή του πειράματος χρησιμοποιήθηκαν οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα στην προηγούμενη μελωδία. Οπότε ρυθμίζοντας τις παραμέτρους σε:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	512
Threshold	0.30
Silence	-60
Median	9
Note On Algorithm	Complex domain Distance
2nd Note On Algorithm	-
Pitch Detection Algorithm	Spectral Yin

παίρνουμε τα αποτελέσματα που παρατίθενται παρακάτω στον Πίνακα 5.2.8.

Εξαγόμενες νότες												
MIDI (12 πρώτες νότες)	63	65	67	58	67	65	63	65	65	63	62	58
English (12 πρώτες νότες)	D4#	F4	G4	A3#	G4	F4	D4#	F4	F4	D4#	D4	A3#
MIDI (Υπόλοιπες νότες)	65	67	68	67	70	72	70	63	65	67	65	63
English (Υπόλοιπες νότες)	F4	G4	G4#	G4	A4#	C5	A4#	D4#	F4	G4	F4	D4#

Πίνακας 5.2.8: Εξαγόμενες νότες από το αρχείο Trumpet.wav.

Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών, βλέπουμε ότι με αυτές τις παραμέτρους όλες οι νότες αναγνωρίστηκαν με επιτυχία. Τα αποτελέσματα και για αυτή τη μελωδία είναι άριστα.

Στα επόμενα πειράματα ακολουθήθηκε η ίδια μεθοδολογία για να βρεθούν οι τιμές των παραμέτρων που παρουσιάζουν τα καλύτερα δυνατά αποτελέσματα.

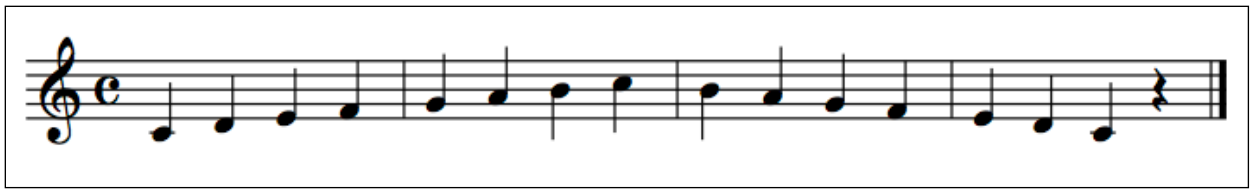
5.3 Πείραμα 2 - Πιάνο

Στο πείραμα 2 εξετάζονται δύο μελωδίες που συντέθηκαν και έγιναν export από το Finale. Η πρώτη είναι η φυσική κλίμακα του Ντο και η δεύτερη είναι μία μελωδία που συντέθηκε για το πείραμα.

Το αρχείο ήχου που χρησιμοποιήθηκε είναι το Piano_c.aif και η κυματομορφή του είναι:

Εικόνα 5.3.1: Κυματομορφή αρχείου ήχου Piano_c.aif.

Η παρτιτούρα του αρχείου είναι:



Εικόνα 5.3.2 Παρτιτούρα αναφοράς του αρχείου ήχου Piano_c.aif.

Στον Πίνακα 5.3.1 παρατίθενται οι νότες αναφοράς.

Νότες αναφοράς															
MIDI	60	62	64	65	67	69	71	72	71	69	67	65	64	62	60
English	C4	D4	E4	F4	G4	A4	B4	C3	B4	A4	G4	F4	E4	D4	C4

Πίνακας 5.3.1: Νότες αναφοράς του αρχείου ήχου Piano_c.aif.

Κατά τη διεξαγωγή του πειράματος οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα για το συγκεκριμένο αρχείο ήχου είναι οι εξής:

Παράμετροι	**Τιμές Παραμέτρων**
Sample Rate	44,100
Window Size	512
Threshold	0.15
Silence	-60
Median	5
Note On Algorithm	Energy
2nd Note On Algorithm	-
Pitch Detection Algorithm	Yin

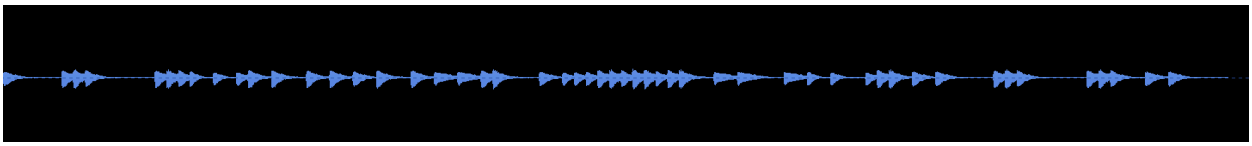
Τα αποτελέσματα του πειράματος παρατίθενται παρακάτω στον Πίνακα 5.3.2.

Εξαγόμενες νότες															
MIDI	60	62	64	65	67	69	71	72	71	69	67	65	64	62	60
English	C4	D4	E4	F4	G4	A4	B4	C3	B4	A4	G4	F4	E4	D4	C4

Πίνακας 5.3.2: Εξαγόμενες νότες από το αρχείο Piano_c.aif.

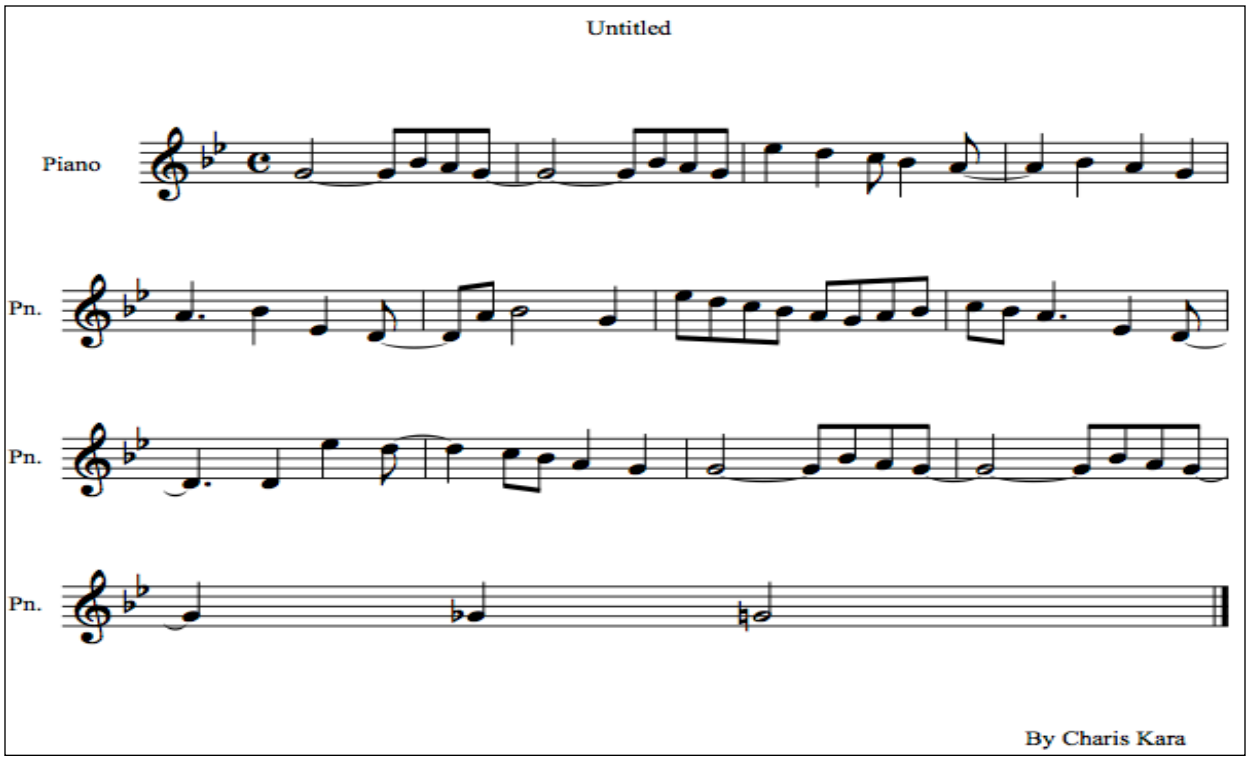
Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών παρατηρούμε ότι το πείραμα, με τις συγκεκριμένες παραμέτρους, ήταν επιτυχές και δεν προέκυψαν καθόλου αποκλίσεις.

Για τη δεύτερη μελωδία χρησιμοποιήθηκε το αρχείο ήχου Piano.aif και η κυματομορφή του είναι:



Εικόνα 5.3.3: Κυματομορφή αρχείου ήχου Piano.aif

Η παρτιτούρα του αρχείου είναι:



Εικόνα 5.3.4: Παρτιτούρα αναφοράς του αρχείου ήχου Piano.aif

Στον Πίνακα 5.3.3 παρατίθενται οι νότες αναφοράς ανάλογα με τη σειρά που εμφανίστηκαν στο πεντάγραμμο.

Νότες αναφοράς																			
MIDI 1η σειρά	67	70	69	67	70	69	67	75	74	72	70	69	70	69	67				
Eng. 1η σειρά	G4	A4#	A4	G4	A4#	A4	G4	D5#	D5	C5	A4#	A4	A4#	A4	G4				
MIDI 2η σειρά	69	70	63	62	69	70	67	75	74	72	70	69	67	69	70	72	70	69	63	62
Eng. 2η σειρά	A4	A4#	D4#	D4	A4	A4#	G4	D5#	D5	C5	A4#	A4	G4	A4	A4#	C5	A4#	A4	D4#	D4
MIDI 3η/4η σειρά	62	75	74	72	70	69	67	67	70	69	67	70	69	67	66	67				
Eng. 3η/4η σειρά	D4	D5#	D5	C5	A4#	A4	G4	G4	A4#	A4	G4	A4#	A4#	G4	F4#	G4				

Πίνακας 5.3.3: Νότες αναφοράς του αρχείου ήχου Piano.aif.

Κατά τη διεξαγωγή του πειράματος οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα για το συγκεκριμένο αρχείο ήχου είναι οι εξής:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	512
Threshold	0.15
Silence	-60
Median	5
Note On Algorithm	Energy
2nd Note On Algorithm	-
Pitch Detection Algorithm	Yin

Τα αποτελέσματα του πειράματος παρατίθενται στον Πίνακα 5.3.4.

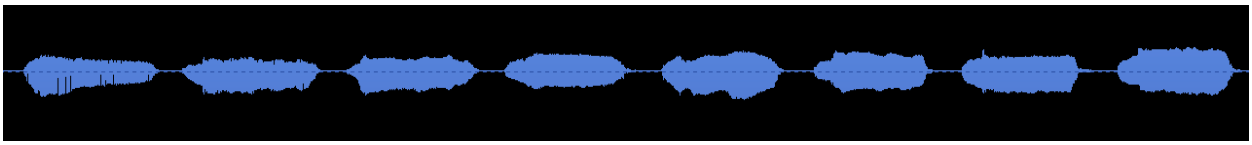
Εξαγόμενες νότες																			
MIDI 1η σειρά	67	70	69	67	70	69	67	75	74	72	70	69	70	69	67				
Eng. 1η σειρά	G4	A4#	A4	G4	A4#	A4	G4	D5#	D5	C5	A4#	A4	A4#	A4	G4				
MIDI 2η σειρά	69	70	63	62	69	70	67	75	74	72	70	69	67	69	70	72	70	69	63	62
Eng. 2η σειρά	A4	A4#	D4#	D4	A4	A4#	G4	D5#	D5	C5	A4#	A4	G4	A4	A4#	C5	A4#	A4	D4#	D4
MIDI 3η/4η σειρά	62	75	74	72	70	69	67	67	70	69	67	70	69	67	66	67				
Eng. 3η/4η σειρά	D4	D5#	D5	C5	A4#	A4	G4	G4	A4#	A4	G4	A4#	A4#	G4	F4#	G4				

Πίνακας 5.3.4: Εξαγόμενες νότες από το αρχείο Piano.aif.

Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών παρατηρούμε ότι το πείραμα, με τις συγκεκριμένες παραμέτρους, ήταν απολύτως επιτυχές και δεν προέκυψαν καθόλου αποκλίσεις.

5.4 Πείραμα 3 - Φωνή

Στο πείραμα 3 εξετάζονται οι ίδιες μελωδίες που χρησιμοποιήθηκαν στο πείραμα της τρομπέτας, εκτελεσμένες μια οκτάβα πιο χαμηλά από δύο διαφορετικές φωνές. Το αρχείο ήχου της πρώτης μελωδίας είναι το Voice_c.wav και η κυματομορφή του είναι:



Εικόνα 5.4.1: Κυματομορφή αρχείου ήχου Voice_c.wav.

Στον Πίνακα 5.4.1 παρατίθενται οι νότες αναφοράς του αρχείου.

Νότες αναφοράς								
MIDI	48	50	52	53	55	57	59	60
English	C3	D3	E3	F3	G3	A3	B3	C4

Πίνακας 5.4.1: Νότες αναφοράς του αρχείου ήχου Voice_c.wav.

Κατά τη διεξαγωγή του πειράματος, οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα για το συγκεκριμένο αρχείο ήχου είναι οι εξής:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	1,024
Threshold	0.60
Silence	-60
Median	20
Note On Algorithm	High Frequency Content
2nd Note On Algorithm	Complex Domain Distance
Pitch Detection Algorithm	Yin

Τα αποτελέσματα του πειράματος παρατίθενται στον Πίνακα 5.4.2.

Εξαγόμενες νότες								
MIDI	60	50	52	66	55	57	59	72
English	C4	D3	E3	F4 #	G3	A3	B3	C5

Πίνακας 5.4.2: Εξαγόμενες νότες από το αρχείο Voice_c.wav.

Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών βλέπουμε ότι συνολικά από τις 8 νότες, οι 5 αναγνωρίστηκαν με επιτυχία. Οι αποκλίσεις είναι οι εξής:

- Η πρώτη και η όγδοη νότα της μελωδίας (Ντο) δεν αναγνωρίστηκαν επιτυχώς, αλλά μια οκτάβα πιο πάνω.
- Η τέταρτη νότα τη μελωδίας (Φα) δεν αναγνωρίστηκε επιτυχώς, αλλά ένα ημιτόνιο πιο πάνω.

Σε αυτή τη μελωδία παρουσιάστηκαν πολλά προβλήματα και με τον εντοπισμό των onset και με την αναγνώριση της συχνότητας. Εντύπωση προκαλεί το γεγονός ότι δε μπόρεσε να βρεθεί τιμή για το threshold που να αποτρέψει το διπλό onset στην νότα Σι, χωρίς ταυτόχρονα να προκαλέσει, λανθασμένα, την απόκρυψη κάποιας άλλης νότας. Ακόμα και σε συνδιασμό με 20 δείγματα καθυστέρησης (median), δεν αποφεύχθηκε το

διπλό onset. Αυτό πιθανώς οφείλεται σε αστάθεια έντασης κατά την εκτέλεση της νότας από τον ερμηνευτή. Όσον αφορά τα συχνοτικά προβλήματα, δεδομένης της αστάθειας που έχει μια ερασιτεχνική ανθρώπινη φωνή, ήταν αναμενόμενο να υπάρχουν αποκλίσεις.

Το αρχείο ήχου της δεύτερης μελωδίας είναι το Voice.wav και η κυματομορφή του είναι:

Εικόνα 5.4.2: Κυματομορφή αρχείου ήχου Voice.wav.

Στον Πίνακα 5.4.3 παρατίθενται οι νότες αναφοράς του αρχείου.

Νότες Αναφοράς												
MIDI (12 πρώτες νότες)	51	53	55	46	55	53	51	53	53	51	50	46
English (12 πρώτες νότες)	D3 #	F3	G3	A2 #	G3	F3	D3 #	F3	F3	D3 #	D3	A2 #
MIDI (Υπόλοιπες νότες)	53	55	56	55	58	60	58	51	53	55	53	51
English (Υπόλοιπες νότες)	F3	G3	G3 #	G3	A3 #	C4	A3 #	D3 #	F3	G3	F3	D3 #

Πίνακας 5.4.3: Νότες αναφοράς του αρχείου ήχου Voice.wav.

Κατά τη διεξαγωγή του πειράματος, οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα για το συγκεκριμένο αρχείο ήχου είναι οι εξής:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	512
Threshold	0.25
Silence	-60
Median	20
Note On Algorithm	Phase Deviation
2nd Note On Algorithm	-
Pitch Detection Algorithm	Yin

Τα αποτελέσματα του πειράματος παρατίθενται παρακάτω στον Πίνακα 5.4.4.

Εξαγόμενες Νότες												
MIDI (12 πρώτες νότες)	51	53	55	46	55	53	51	53	53	51	50 50	46
English (12 πρώτες νότες)	D3 #	F3	G3	A2 #	G3	F3	D3 #	F3	F3	D3 #	D3 D3	A2 #
MIDI (Υπόλοιπες νότες)	53	55	56	55	58	60	58	51	53	54	53	51
English (Υπόλοιπες νότες)	F3	G3	G3 #	G3	A3 #	C4	A3 #	E3	F3	F3 #	F3	D3 #

Πίνακας 5.4.4: Εξαγόμενες νότες από το αρχείο Voice.wav.

Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών βλέπουμε ότι συνολικά από τις 24 νότες, οι 20 αναγνωρίστηκαν με επιτυχία. Οι αποκλίσεις είναι οι εξής:

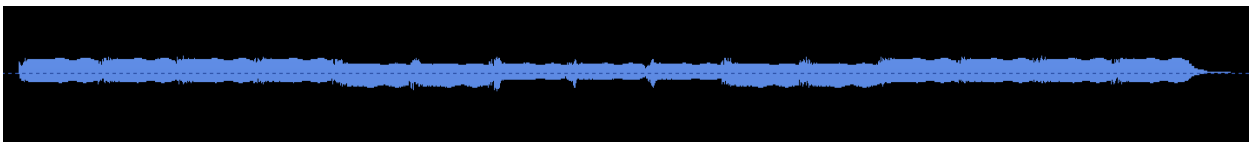
• Η έβδομη νότα της μελωδίας (Ρε#) δεν καταγράφηκε.

- Η ενδέκατη νότα της μελωδίας (Ρε) εμφανίστηκε λανθασμένα δύο φορές.
- Η δέκατη όγδοη νότα της μελωδίας (Μι ύφεση) αναγνωρίστηκε λανθασμένα ένα ημιτόνιο πιο ψηλά.
- Η εικοστή νότα της μελωδίας (Σολ) αναγνωρίστηκε λανθασμένα ένα ημιτόνιο πιο χαμηλά.

Σε αυτή τη μελωδία, οι παράμετροι που ορίστηκαν προηγουμένως δεν επέφεραν καλά αποτελέσματα, με αποτέλεσμα να δοκιμαστούν και να ρυθμιστούν εκ νέου. Αυτό λογικά συνέβη διότι την κάθε μελωδία την τραγουδάει διαφορετικός άνθρωπος. Τελικά πάντως στη δεύτερη μελωδία τα αποτελέσματα είναι καλύτερα.

5.5 Πείραμα 4 - Βιολί

Στο πείραμα 4 εξετάζονται οι ίδιες μελωδίες που χρησιμοποιήθηκαν στο πείραμα με το πιάνο. Το αρχείο ήχου της πρώτης μελωδίας που έγινε export από το Finale είναι το Violin_c.aif και η κυματομορφή του είναι:



Εικόνα 5.5.1: Κυματομορφή αρχείου ήχου Violin_c.aif.

Στον Πίνακα 5.5.1 παρατίθενται οι νότες αναφοράς.

Νότες αναφοράς															
MIDI	60	62	64	65	67	69	71	72	71	69	67	65	64	62	60
English	C4	D4	E4	F4	G4	A4	B4	C3	B4	A4	G4	F4	E4	D4	C4

Πίνακας 5.5.1: Νότες αναφοράς του αρχείου ήχου Violin_c.aif.

Κατά τη διεξαγωγή του πειράματος οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα για το συγκεκριμένο αρχείο ήχου είναι οι εξής:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	512
Threshold	0.15
Silence	-60
Median	20
Note On Algorithm	High Frequency Content
2nd Note On Algorithm	Complex Domain Distance
Pitch Detection Algorithm	Yin

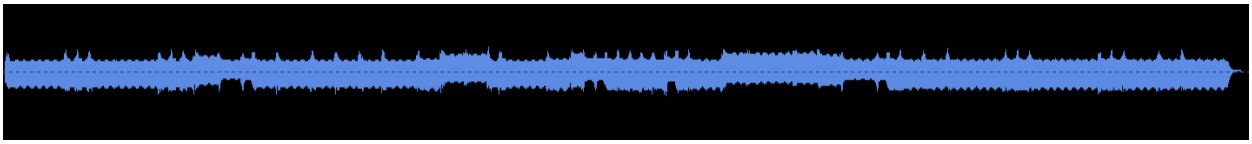
Τα αποτελέσματα του πειράματος παρατίθενται στον Πίνακα 5.5.2.

Εξαγόμενες νότες															
MIDI	60	62	64	65	67	69	71	72	71	69	67	65	64	62	60
English	C4	D4	E4	F4	G4	A4	B4	C3	B4	A4	G4	F4	E4	D4	C4

Πίνακας 5.5.2: Εξαγόμενες νότες από το αρχείο Violin_c.aif.

Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών βλέπουμε ότι όλες οι νότες αναγνωρίστηκαν με επιτυχία. Αξίζει, ωστόσο, να σημειωθεί ότι σε αυτό το πείραμα με το MIDI βιολί, λόγω του vibrato που εκτελούν οι MIDI γεννήτριες στον ήχο του βιολιού, παρουσιάστηκαν σε όλες τις νότες πολλαπλά onset. Αυτό, στη συγκεκριμένη περίπτωση, αντιμετωπίστηκε αυξάνοντας την τιμή της παραμέτρου median μέχρι τα 20 δείγματα καθυστέρησης, γεγονός, όμως, που αυξάνει αισθητά τη χρονική απόκριση του συστήματος.

Το αρχείο ήχου της δεύτερης μελωδίας είναι το Violin.aif και η κυματομορφή του είναι:



Εικόνα 5.5.2: Κυματομορφή αρχείου ήχου Violin.aif

Στον Πίνακα 5.5.3 παρατίθενται οι νότες αναφοράς ανάλογα με τη σειρά που εμφανίστηκαν στο πεντάγραμμο.

Νότες αναφοράς																			
MIDI 1η σειρά	67	70	69	67	70	69	67	75	74	72	70	69	70	69	67				
Eng. 1η σειρά	G4	A4#	A4	G4	A4#	A4	G4	D5#	D5	C5	A4#	A4	A4#	A4	G4				
MIDI 2η σειρά	69	70	63	62	69	70	67	75	74	72	70	69	67	69	70	72	70	69	63	62
Eng. 2η σειρά	A4	A4#	D4#	D4	A4	A4#	G4	D5#	D5	C5	A4#	A4	G4	A4	A4#	C5	A4#	A4	D4#	D4
MIDI 3η/4η σειρά	62	75	74	72	70	69	67	67	70	69	67	70	69	67	66	67				
Eng. 3η/4η σειρά	D4	D5#	D5	C5	A4#	A4	G4	G4	A4#	A4	G4	A4#	A4#	G4	F4#	G4				

Πίνακας 5.5.3: Νότες αναφοράς του αρχείου ήχου Violin.aif.

Κατά τη διεξαγωγή του πειράματος οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα για το συγκεκριμένο αρχείο ήχου είναι οι εξής:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	1,024
Threshold	0.10
Silence	-60
Median	20
Note On Algorithm	High Frequency Content
2nd Note On Algorithm	Complex Domain Distance
Pitch Detection Algorithm	Yin

Τα αποτελέσματα του πειράματος παρατίθενται στον Πίνακα 5.5.4.

Εξαγόμενες νότες																			
MIDI 1η σειρά	67	70	69	67	70	69	67	87	74	72	70	69	70	69	67				
Eng. 1η σειρά	G4	A4#	A4	G4	A4#	A4	G4	D6#	D5	C5	A4#	A4	A4#	A4	G4				
MIDI 2η σειρά	69	70	63	62	69	70	67	87	74	72	70	69	67	69	70	72	70	69	63	62
Eng. 2η σειρά	A4	A4#	D4#	D4	A4	A4#	G4	D6#	D5	C5	A4#	A4	G4	A4	A4#	C5	A4#	A4	D4#	D4
MIDI 3η/4η σειρά	62	87	74	72	70	69	67	67	70	69	67	70	69	67	66	67				
Eng. 3η/4η σειρά	D4	D6#	D5	C5	A4#	A4	G4	G4	A4#	A4	G4	A4#	A4#	G4	F4#	G4				

Πίνακας 5.5.4: Εξαγόμενες νότες από το αρχείο Violin.aif.

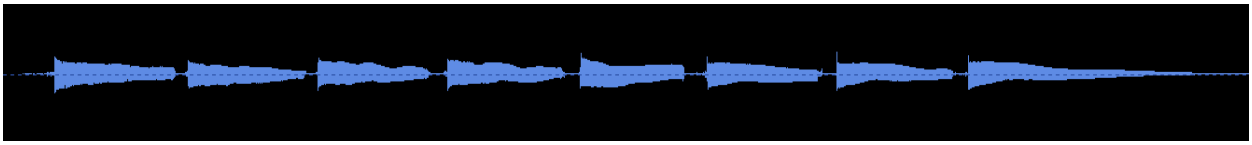
Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών βλέπουμε ότι από τις 52 νότες οι 47 αναγνωρίστηκαν με επιτυχία. Πιο συγκεκριμένα οι αποκλίσεις είναι:

- Η πρώτη νότα της μελωδίας (Σολ) δεν καταγράφηκε.
- Η νότα Μι ύφεση (παρουσιάζεται ως D5#) δεν αναγνωρίστηκε επιτυχώς σε ολόκληρη τη μελωδία, αλλά μια οκτάβα ψηλότερα.
- Η εικοστή πρώτη νότα της μελωδίας δεν καταγράφηκε.

Στο δεύτερο πείραμα με το βιολί, παρουσιάστηκε το ίδιο πρόβλημα με τα πολλαπλά onset που συναντήθηκε και στο πρώτο. Ωστόσο, με την ίδια αντιμετώπιση, τελικά τα αποτελέσματα κυμάνθηκαν σε ικανοποιητικά επίπεδα.

5.6 Πείραμα 5 - Κιθάρα

Στο πείραμα 5 εξετάζονται οι ίδιες μελωδίες που χρησιμοποιήθηκαν στο πείραμα με την τρομπέτα και τη φωνή. Το αρχείο ήχου της πρώτης μελωδίας είναι το Guitar_c.wav και η κυματομορφή του είναι:



Εικόνα 5.6.1: Κυματομορφή αρχείου ήχου Guitar_c.wav.

Στον Πίνακα 5.6.1 παρατίθενται οι νότες αναφοράς του αρχείου.

Νότες αναφοράς								
MIDI	60	62	64	65	67	69	71	72
English	C4	D4	E4	F4	G4	A4	B4	C3

Πίνακας 5.6.1: Νότες αναφοράς του αρχείου ήχου Guitar_c.wav.

Κατά τη διεξαγωγή του πειράματος, οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα για το συγκεκριμένο αρχείο ήχου είναι οι εξής:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	512
Threshold	0.30
Silence	-60
Median	5
Note On Algorithm	Energy
2nd Note On Algorithm	-
Pitch Detection Algorithm	Yin

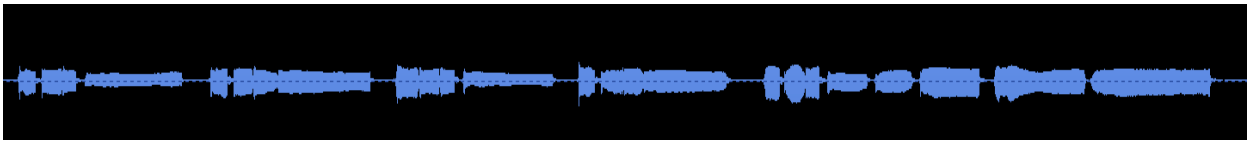
Τα αποτελέσματα του πειράματος παρατίθενται στον Πίνακα 5.6.2.

Εξαγόμενες νότες								
MIDI	60	62	64	65	67	69	71	72
English	C4	D4	E4	F4	G4	A4	B4	C3

Πίνακας 5.6.2: Εξαγόμενες νότες από το αρχείο Guitar_c.wav.

Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών, βλέπουμε ότι όλες οι νότες αναγνωρίστηκαν με επιτυχία.

Το δεύτερο αρχείο ήχου που χρησιμοποιήθηκε είναι το Guitar.wav και η κυματομορφή του είναι:



Εικόνα 5.6.2: Κυματομορφή αρχείου ήχου Guitar.wav.

Στον Πίνακα 5.6.3 παρατίθενται οι νότες αναφοράς του αρχείου.

Νότες Αναφοράς												
MIDI (12 πρώτες νότες)	63	65	67	58	67	65	63	65	65	63	62	58
English (12 πρώτες νότες)	D4#	F4	G4	A3#	G4	F4	D4#	F4	F4	D4#	D4	A3#
MIDI (Υπόλοιπες νότες)	65	67	68	67	70	72	70	63	65	67	65	63
English (Υπόλοιπες νότες)	F4	G4	G4#	G4	A4#	C5	A4#	D4#	F4	G4	F4	D4#

Πίνακας 5.6.3: Νότες αναφοράς του αρχείου ήχου Guitar.wav.

Κατά τη διεξαγωγή του πειράματος, οι παράμετροι που απέφεραν τα καλύτερα αποτελέσματα για το συγκεκριμένο αρχείο ήχου είναι ίδιες με προηγουμένως. Έτσι ρυθμίζοντας τις παραμέτρους ως εξής:

Παράμετροι	Τιμές Παραμέτρων
Sample Rate	44,100
Window Size	512
Threshold	0.30
Silence	-60
Median	5
Note On Algorithm	Energy
2nd Note On Algorithm	-
Pitch Detection Algorithm	Spectral Yin

παίρνουμε τα αποτελέσματα που παρατίθενται παρακάτω στον Πίνακα 5.6.4.

Εξαγόμενες νότες												
MIDI (12 πρώτες νότες)	63	65	67	58	67	65	63	65	65	63	62	58
English (12 πρώτες νότες)	D4#	F4	G4	A3#	G4	F4	D4#	F4	F4	D4#	D4	A3#
MIDI (Υπόλοιπες νότες)	65	67	68	67	70	72	70	63	65	67	65	63
English (Υπόλοιπες νότες)	F4	G4	G4#	G4	A4#	C5	A4#	D4#	F4	G4	F4	D4#

Πίνακας 5.6.4: Εξαγόμενες νότες από το αρχείο Guitar.wav.

Από τη σύγκριση των νοτών αναφοράς και των εξαγόμενων νοτών βλέπουμε ότι, με αυτές τις παραμέτρους, όλες οι νότες αναγνωρίστηκαν με επιτυχία. Τα αποτελέσματα για αυτή τη μελωδία είναι άριστα.

5.7 Σύγκριση Αποτελεσμάτων

Σε γενικές γραμμές κατά τη διεξαγωγή των πειραμάτων διαπιστώσαμε ότι η εφαρμογή λειτουργεί σε ικανοποιητικά επίπεδα. Τα λάθη που προέκυψαν χωρίζονται σε λάθη που αφορούν στην εκτίμηση της τονικότητας και λάθη που αφορούν στον εντοπισμό των onset.

Η ακρίβεια της εφαρμογής ως προς τον εντοπισμό των onset υπολογίζεται σύμφωνα με τη διαδικασία αξιολόγησης που παρουσιάζεται στο ***Mirex 2012 Onset Detection***, με βάση τα πειράματα που περιγράφονται στις παραπάνω παραγράφους. Συνοψίζοντας, αυτό περιλαμβάνει τη χρήση ενός audio editor, ώστε να συγκριθούν οι χρόνοι εντοπισμού των εξαγόμενων onset με τους χρόνους onset των νοτών αναφοράς. Για να θεωρηθεί σωστός ο εντοπισμός ενός onset πρέπει ο χρόνος εξαγωγής του να συμπέσει με το χρόνο onset της νότας αναφοράς, με ανεκτές αποκλίσεις της τάξης των +- 50 ms. Εάν αυτό δε συμβαίνει τότε υπάρχει εσφαλμένο αρνητικό onset, ενώ αν ο χρόνος είναι έξω από τα όρια ανοχής θεωρείται εσφαλμένα θετικό onset. Υποκατηγορίες αυτών των δύο

είναι τα συγχωνευόμενα onset και τα διπλά onset αντίστοιχα. Παρακάτω στον Πίνακα 5.7.1 παρουσιάζονται τα αποτελέσματα ακρίβειας των onset ανά όργανο όπου:

- **GtO** αριθμός νοτών αναφοράς (number of ground truth onsets)
- **Ocd** αριθμός σωστά αναγνωρισμένων onset (number of correctly detected onsets)
- **Ofn** αριθμός μη αναγνωρισμένων, υπαρκτών onset (number of false negative onsets)
- **Ofp** αριθμός αναγνωρισμένων, μη υπαρκτών, onset (number of false positive onsets)
- **Precision** P = Ocd / (Ocd +Ofp)
- **Recall** R = Ocd / (Ocd + Ofn)
- **F-measure** F = 2*P*R/(P+R)

Onset Detection	**Τρομπέτα**	**Πιάνο**	**Φωνή**	**Βιολί**	**Κιθάρα**
GtO	32	66	32	66	32
Ocd	32	66	30	64	32
Ofn	0	0	1	2	0
Ofp	0	0	1	0	0
Precision	1.000	1.000	0.967	1.000	1.000
Recall	1.000	1.000	0.967	0.969	1.000
F-measure	1.000	1.000	0.967	0.984	1.000

Πίνακας 5.7.1: Πίνακας ακρίβειας εφαρμογής ως προς τον εντοπισμό των onset ανά μουσικό όργανο.

Η ακρίβεια της εφαρμογής ως προς το σύνολο των διεργασιών που επιτελεί (κατάτμηση ηχητικών δεδομένων και αναγνώριση τονικού ύψους) υπολογίζεται σύμφωνα με τη διαδικασία αξιολόγησης που παρουσιάζεται στο ***Mirex 2012 Frequency Estimation & Tracking***. Όπου Precision ορίζεται ως ο λόγος των σωστά αναγνωρισμένων νοτών προς τις νότες αναφοράς, και Recall ορίζεται ως ο λόγος των σωστά αναγνωρισμένων νοτών προς το σύνολο των αναγνωρισμένων νοτών. Μία νότα αναφοράς θεωρείται ότι έχει αναγνωρισθεί σωστά, όταν το σύστημα έχει επιστρέψει μια νότα με την ίδια τονική αξία, με χρόνο onset που συμπέφτει με το δικό της (με όριο ανοχής +-50 ms) και offset που βρίσκεται εντός του 20% του εύρους της. Στον Πίνακα 5.8.2 παρουσιάζεται η ακρίβεια της εφαρμογής ανά μουσικό όργανο, όπου:

- **GtN** αριθμός νοτών αναφοράς (number of ground truth notes)
- **CtN** αριθμός σωστά αναγνωρισμένων νοτών (number of correctly transcribed notes)
- **tN** αριθμός αναγνωρισμένων νοτών (number of transcribed notes)

Note Tracking	**Τρομπέτα**	**Πιάνο**	**Φωνή**	**Βιολί**	**Κιθάρα**
GtN	32	66	32	66	32
CtN	32	66	25	61	32
tN	32	66	32	64	32
Precision	1.000	1.000	0.781	0.924	1.000
Recall	1.000	1.000	0.781	0.953	1.000
F-measure	1.000	1.000	0.781	0.938	1.000

Πίνακας 5.8.2: Πίνακας ακρίβειας συνολικής λειτουργίας της εφαρμογής ανά μουσικό όργανο.

Το MIDI βιολί που χρησιμοποιήθηκε στο πείραμα 4 προσομοιώνει το φυσικό vibrato του βιολιού, που είναι άρρηκτα συνδεδεμένο με κάθε μουσική εκτέλεση βιολιού, με αποτέλεσμα να παρουσιάζονται προβλήματα τόσο στην αναγνώριση της τονικότητας όσο και στον εντοπισμό ψευδών νοτών κατά τη διάρκεια της ίδιας νότας. Ωστόσο, με τις κατάλληλες ρυθμίσεις τα αποτελέσματα είναι ικανοποιητικά με την τιμή του F-measure να είναι 0.938.

Οι δύο αντρικές φωνές που χρησιμοποιήθηκαν στο πείραμα 3 ήταν ερασιτεχνικές, και ακούγοντας τις εκτελέσεις τους από τα αρχεία ήχου, μπορεί να εντοπίσει κανείς τις αστάθειες στην ένταση της φωνής, οι οποίες είναι η πιο πιθανή αιτία των διπλών onset. Όσον αφορά το τονικό κομμάτι, ωστόσο, δεν παρατηρούνται κάποια φάλτσα κατά την εκτέλεση των μελωδιών. Τα λάθη που παρατηρήθηκαν είναι ως επί το πλείστον λάθη οκτάβας ή ημιτονίου.

Στην πείραμα της τρομπέτας προκαλεί εντύπωση το γεγονός ότι δεν παρουσιάστηκαν καθόλου αποκλίσεις. Αφενός επειδή, όπως και με το πείραμα της φωνής, ο μουσικός ήταν ερασιτέχνης, και αφετέρου επειδή οι πρώτες αρμονικές της τρομπέτας έχουν υψηλή

ένταση. Ωστόσο, με τις κατάλληλες ρυθμίσεις τα αποτελέσματα του πειράματος είναι εξαιρετικά.

Το πιάνο και η κιθάρα είναι και τα δύο έγχορδα, στα οποία η χορδή κρούεται για να παράγει ήχο. Όπως ήταν αναμενόμενο, ο εντοπισμός της αρχής των νοτών, με βάση τις μεταβολές της ενέργειας, ήταν απόλυτα επιτυχής. Η αναγνώριση της τονικότητας των νοτών, χρησιμοποιώντας μερικά δείγματα καθυστέρησης, ώστε να αποφευχθεί το κομμάτι της ατάκας με τα transient στοιχεία, στέφθηκε με απόλυτη επιτυχία. Συνολικά τα αποτελέσματα των πειραμάτων και στις δύο περιπτώσεις είναι άψογα.

6 Συμπεράσματα

6.1 Σύνοψη

Ο αρχικός στόχος της παρούσας πτυχιακής εργασίας ήταν η υλοποίηση ενός εικονικού μουσικού οργάνου, μέσω του οποίου ο χρήστης, χρησιμοποιώντας μονοφωνικές ηχητικές ροές, θα μπορούσε σε πραγματικό χρόνο να αλλάζει το ηχόχρωμα και την τονικότητα της μουσικής εκτέλεσής του.

Το εγχείρημα αυτό αποδείχτηκε αρκετά πολύπλοκο και αντιμετωπίστηκαν αρκετές δυσκολίες. Για να πραγματοποιηθεί κάτι τέτοιο επιτυχώς, πρέπει να επιλυθούν κάποια υπο-προβλήματα:

- Ανίχνευση αρχής και τέλους νότας.
- Αναγνώριση τονικού ύψους.
- Διαχωρισμός μουσικής/θορύβου (όπου θόρυβος νοείται οτιδήποτε δεν είναι μουσική), ούτως ώστε αφενός να αποφεύγεται η λανθασμένη αναγνώριση ψευδών νοτών και αφετέρου να μη δαπανώνται υπολογιστικοί πόροι.
- Χρηστικότητα της εφαρμογής.

Τελικά ο διαχωρισμός μουσικής/θορύβου παραλείφθηκε σε αυτήν την εφαρμογή, λόγω πολυπλοκότητας του εγχειρήματος της εκπαίδευσης "έξυπνων" συστημάτων και της ακαταλληλότητας των εναλλακτικών λύσεων για εφαρμογές πραγματικού χρόνου, εφόσον αυξάνουν σημαντικά το υπολογιστικό κόστος και τη χρονική απόκριση του συστήματος.

Επίσης, η καθεαυτή διαχείριση των αλγορίθμων ανίχνευσης και αναγνώρισης των νοτών, ήταν ένα εγχείρημα με πολλαπλές δυσκολίες. Εκτός από την πολυπλοκότητα της σχεδίασης του συστήματος αποφάσεων για τις τελικές νότες, παρουσιάστηκε πρόβλημα στο ποσοστό ελευθερίας επιλογών που θα πρέπει να έχει ο χρήστης. Τελικά, επιλέχθηκε να δοθεί αρκετή ελευθερία στο χρήστη, ώστε να μπορεί να αξιοποιηθεί η εφαρμογή στο μέγιστο δυνατό, θυσιάζοντας, ωστόσο, τη φιλική προς το χρήστη εικόνα που πρέπει να έχει μια τέτοια εφαρμογή, καθώς και κάποιους επιθυμητούς αυτοματισμούς.

Τα διάφορα λάθη που προκύπτουν κατά τη χρήση του εικονικού οργάνου αφορούν εν μέρη το σχεδιασμό του συστήματος, αλλά και τους ίδιους τους αλγόριθμους αναγνώρισης που χρησιμοποιήθηκαν για την υλοποίησή του. Εξάλλου, οι αλγόριθμοι αυτοί δεν έχουν σταματήσει στις μέρες μας να είναι αντικείμενο μελέτης και βελτίωσης από τον ερευνητικό κλάδο της Ανάκτησης Μουσικής Πληροφορίας (Music Information Retrieval - MIR).

Παρ' όλες τις δυσκολίες που αντιμετωπίστηκαν, η εφαρμογή λειτουργεί σε ένα μεγάλο βαθμό με επιτυχία, ειδικά όταν επιλεχθούν οι κατάλληλοι αλγόριθμοι σε συνδυασμό με τα σωστά εύρη τιμών για τις διάφορες παραμέτρους.

Μελετώντας τη συναφή βιβλιογραφία του ερευνητικού χώρου Ανάκτησης Μουσικής Πληροφορίας (Music Information Retrieval - MIR), διαπιστώθηκε ότι για τη δημιουργία ενός χρήσιμου, χρηστικού και αυτοματισμένου συστήματος, όπως το παρόν, απαιτούνται συνδυαστικές γνώσεις και οι εφαρμογές τους από πολλούς τομείς μελέτης του MIR.

6.2 Μελλοντικές επεκτάσεις

Οι μελλοντικές επεκτάσεις της εφαρμογής μπορούν να αφορούν στη βελτίωση του συστήματος έκβασης αποφάσεων της τελικής νότας, ώστε να είναι πιο αξιόπιστο και αυτόνομο. Δηλαδή, να μη χρειάζεται ο χρήστης να ορίζει τιμές για τις διάφορες παραμέτρους, αλλά να ορίζονται αυτόματα, ή έστω, μέσω κάποιων προεπιλογών (presets), ανάλογα με το μουσικό μέσο που χρησιμοποιεί.

Επίσης, ως προς τη βελτίωση της απόδοσης του εικονικού οργάνου, μπορεί να υλοποιηθεί ένα σύστημα διαχωρισμού μεταξύ μουσικής και θορύβου, ώστε να αποφεύγονται οι ψευδείς αναγνωρίσεις νοτών σε τμήματα που δεν περιέχουν μουσικό περιεχόμενο. Ακόμη, μπορούν να τεθούν προς εκτενέστερη μελέτη και βελτίωση οι αλγόριθμοι που χρησιμοποιήθηκαν.

Η εφαρμογή υλοποιήθηκε και χρησιμοποιήθηκε στο λειτουργικό σύστημα Mac OS X. Ωστόσο, οι προγραμματιστικές βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίησή

της, υποστηρίζουν και τα λειτουργικά συστήματα Linux και Windows. Επομένως η εφαρμογή, με κατάλληλη τροποποίηση, μπορεί να επεκταθεί, ώστε να λειτουργεί και σε αυτά τα λειτουργικά συστήματα.

Μία άλλη μελλοντική επέκταση μπορεί να είναι η υλοποίηση ενός virtual synthesizer, ή η δυνατότητα σύνδεσης με κάποιο υπάρχον VSTi ή Audio Unit instrument. ΄Ετσι, ο χρήστης θα έχει πρόσβαση σε περισσότερες και πιο φυσικές μουσικές χροιές, αποκλειστικά μέσω λογισμικού, χωρίς να χρειάζεται επιπλέον συσκευές (hardware). Μια ακόμη λειτουργία που μπορεί να προστεθεί στην εφαρμογή, είναι η αποθήκευση των νοτών σε μορφή παρτιτούρας σε ένα ξεχωριστό αρχείο μορφής MusicXML, PDF κ.λ.π.

Τέλος, μπορεί να επεκταθεί ολόκληρο το αντικείμενο της εφαρμογής, για να περιλαμβάνει αναγνώριση τονικότητας σε ζωντανές ηχητικές ροές πολυφωνικής μουσικής, καθώς επίσης και η επιλογή Offline λειτουργίας, με την οποία ο χρήστης θα μπορεί να ανοίγει και να επεξεργάζεται κατά παρόμοιο τρόπο ηχητικά αρχεία.

7 Παραπομπές

A. Freed. Music metadata quality: A multiyear case study using the music of Skip James. In Proc. AES 121st Convention., San Francisco, CA, 2006.

Michael A. Casey, Member IEEE, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: Current directions and future challenges. In Proceedings of the IEEE, volume 96, no. 4, pages 670-672, 2008.

Lu, D. Liu, and H. J. Zhang. Automatic mood detection and tracking of music audio signals. In IEEE Trans. Audio Speech Language Process., volume. 14, no. 1, pages 5–18, 2006.

Dirk Moelants and Christian Rampazzo. 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.

Anssi Klapuri. 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, 1999b.

Miller S. Puckette, Theodore Apel, and David D. Zicarelli. Real-time analysis tools for PD and MSP. In Proceedings of the International Computer Music Conference (ICMC), Ann Arbor, University of Michigan, USA, 1998.

Paul Masri. Computer modeling of Sound for Transformation and Synthesis of Musical Signal. PhD dissertation, University of Bristol, UK, 1996.

Juan-Pablo Bello. Towards the Automated Analysis of Simple Polyphonic Music. PhD thesis, Centre for Digital Music, Queen Mary University of London, London, UK, 2003.

Anssi Klapuri. Signal Processing Methods for the Automatic Transcription of Music. PhD thesis, Tampere University of Technology, Tampere, Finland, 2004.

Florent Jaillet and Xavier Rodet. Improved modelling of attack transients in music analysis synthesis. In Proceedings of the International Computer Music Conference (ICMC), pages 30–33, Havana, Cuba, 2001.

Leslie S. Smith. Using an onset-based representation for sound segmentation. In Proceedings of the International Conference on Neural networks and their Applications (NEURAP), pages 274–281, Marseilles, France, March 20-22 1996.

Fabien Gouyon and Simon Dixon. 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.

Simon Dixon, Fabien Gouyon, and Gerhard Widmer. 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.

Eric D. Scheirer. Tempo and beat analysis of acoustic musical signals. Journal of the Acoustical Society of America, 103(1):588–601, 1998b.

Juan-Pablo Bello, Laurent Daudet, Samer Abdallah, Christopher Duxbury, Mike P. Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. IEEE Transactions in Speech and Audio Processing, 13(5):1035–1047, September 2005.

Paul M. Brossier. Automatic Annotation of Musical Audio for Interactive Applications. Centre for Digital Music Queen Mary University of London, pages, August 2006.

Andrew W. Schloss. On the Automatic Transcription of Percussive Music - From Acoustic Signal to High-Level Analysis. PhD thesis, Department of Hearing and Speech, Stanford University, California, USA, 1985.

Jonhatan Foote and Shingo Uchihashi. The beat spectrum: a new approach to rhythm analysis. In Proceedings of the IEEE International Conference on Multi- media and Expo (ICME 2001), pages 881–884, Tokyo, Japan, August 2001.

Juan-Pablo Bello, Mike P. Davies, and Mark B. Sandler. Phase-based note onset detection for music signals. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 441–444, Hong- Kong, 2003.

Christopher Duxbury, Mike E. Davies, and Mark B. Sandler. Complex domain onset detection for musical signals. In Proceedings of the International Conference on Digital Audio Effects (DAFx-03), pages 90–93, London, UK, 2003.

Samer Abdallah and Mark D. Plumbley. Unsupervised onset detection: a probabilistic approach using ICA and a hidden Markov classifier. In Cambridge Music Processing Colloquium, Cambridge, UK, 2003.

Stephen Hainsworth and Malcom Macleod. Onset detection in music audio signals. In Proceedings of the International Computer Music Conference (ICMC), pages 163–166, Singapore, 2003.

Lawrence R. Rabiner, Marvin R. Sambur, and Carolyn E. Schmidt. Applications of a nonlinear smoothing algorithm to speech processing. IEEE Transactions on Acoustics, Speech, and Signal Processing, 23(6):552–557, December 1975.

Paul M. Brossier, Juan-Pablo Bello, and Mark D. Plumbley. Real-time temporal segmentation of note objects in music signals. In Proceedings of the International Computer Music Conference (ICMC), pages 458–461, Miami, Florida, USA, November 2004b.

Lawrence R. Rabiner. A tutorial on HMM and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, 1989.

Emilia G´omez, Georges Peterschmitt, Xavier Amatriain, and Perfecto Herrera. Content-based melodic transformations of audio for a music processing appli- cation. In Proceedings of the International Conference on Digital Audio Effects (DAFx-03), pages 333–338, London, UK, 2003b.

Curtis Roads. The Computer Music Tutorial. MIT Press, Cambridge, Massachusetts, 1996, ISBN: 0-262-68082-3.

Anssi Klapuri. Qualitative and quantitative aspects in the design of periodicity estimation algorithms. In Proceedings of the European Signal Processing Conference (EUSIPCO), 2000.

Alain de Cheveign´e. Pitch: Neural Coding and Perception, chapter Pitch perception models. Springer Verlag, New York, 2004. ISBN 0-387-23472-1. Edited by Christopher J. Plack, Arther N. Popper, Richard R. Fay and Andrew J. Oxenham.

Alain de Cheveign´e and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. Journal of the Acoustical Society of America, 111(4): 1917–1930, 2002.

Daniel Pressnitzer, Roy D. Patterson, and Katrin Krumbholz. The lower limit of melodic pitch. Journal of the Acoustical Society of America, 109(5):2074–2084, 2001.

William A. Yost. Pitch strength of iterated rippled noise. Journal of the Acoustical Society of America, 100(5):3329–3335, 1996.

Neville H. Fletcher and Thomas D. Rossing. The Physics of Musical Instruments. Springer-Verlag, New York, 2nd edition, 1998. ISBN 0-387-98374-0.

Lawrence R. Rabiner, Michael J. Cheng, Aaron E. Rosenberg, and Carol A. Mc- Gonegal. A comparative performance study of several pitch detection algorithms. IEEE Transactions on Acoustics, Speech, and Signal Processing, 24(5):399–418, October 1976.

James D. Wise, James R. Caprio, and Thomas W. Parks. Maximum likelihood pitch estimation. IEEE Transactions on Acoustic, Speech and Signal Processing, 24 (5):418–423, October 1976.

Wolfgang Hess. Pitch determination of speech signals. algorithms and devices. Journal of the Acoustical Society of America, 76(4):1277–1278, October 1984.

Richard F. Lyon and Lounette Dyer. Experiments with a computational model of the cochlea. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 1975–1978, Tokyo, Japan, 1986.

Dietrich Schlichth ¨arle. Digital Filters : Basics and Design. Springer, Berlin, 2000. ISBN 3-540-66841-1.

Pedro Cano. Fundamental frequency estimation in the SMS analysis. In Proceedings of the International Conference on Digital Audio Effects (DAFx-98), pages 99– 102, Barcelona, Spain, 1998.

Mario Lang. TuneIt, a simple command-line instrument tuner for Linux. http://delysid.org/tuneit.html, 2003.

Philippe Lepain. Polyphonic pitch extraction from music signals. Journal of New Music Research, 28(4):296–309, 1999.

Robert E. Simpson. Introductory Electronics for Scientists and Engineers. Allyn and Bacon, Boston, Massachusetts, USA, 2nd edition, 1987.

Matthew E. P. Davies and Mark D. Plumbley. Causal tempo tracking of audio. In Proceedings of the International Symposium on Music Information Retrieval (ISMIR), pages 164–169, Barcelona, Spain, October 2004.