Περιεχόμενα

Περίληψη

Στην παρούσα εργασία μελετάται το πρόβλημα της αναγνώρισης μουσικής, δηλαδή η μετατροπή ενός ηχογραφημένου μουσικού σήματος σε συμβολική αναπαράσταση. Αυτή η διαδικασία απαιτεί την εξαγωγή διάφορων χαρακτηριστικών όπως του pitch, της διάρκειας και της έντασης κάθε νότας καθώς επίσης και τον καθορισμό των διαφορετικών οργάνων, του ρυθμού και άλλων στοιχείων "έκφρασης" του κομματιού. Η συγκεκριμένη εργασία επικεντρώνεται στην μελέτη και εφαρμογή μεθόδων εξαγωγής του pitch και της διάρκειας κάθε νότας. Συγκεκριμένα, γίνεται μια προσπάθεια κατηγοριοποίησης και σύντομης περιγραφής των διαφόρων μεθόδων που έχουν χρησιμοποιηθεί για την επεξεργασία μονοφωνικής και πολυφωνικής μουσικής.

Το μεγαλύτερο μέρος της εργασίας αυτής αφορά την υλοποίηση αλγορίθμων αναγνώρισης νοτών μέσα σε ένα μουσικό σήμα. Υλοποιήθηκαν δύο αλγόριθμοι επεξεργασίας μονοφωνικής μουσικής. Ο πρώτος επεξεργάζεται το μουσικό σήμα στο πεδίο του χρόνου χρησιμοποιώντας συντελεστές AMDF ενώ ο δεύτερος στο πεδίο της συχνότητας χρησιμοποιώντας Constant Q Transform. Επιπλέον υλοποιήθηκε ένας αλγόριθμος επεξεργασίας πολυφωνικής μουσικής. Στον αλγόριθμο αυτό, το φάσμα του σήματος σε ένα χρονικό διάστημα, αναλύεται ως γραμμικός συνδυασμός ανεξάρτητων φασματικών συνιστωσών κάθε μία από τις οποίες θεωρείται ότι προκύπτει από μία διαφορετική νότα. Για την ανάλυση αυτή χρησιμοποιήθηκαν δείγματα εκπαίδευσης τα οποία ηχογραφήθηκαν από ένα ηλεκτρονικό πιάνο. Η αναγνώριση νοτών σε ένα μουσικό σήμα βασίζεται στην προσέγγιση του σήματος ως γραμμικό συνδυασμό των δειγμάτων εκπαίδευσης. Για την αξιολόγηση των τριών αλγορίθμων ηχογραφήθηκαν από ηλεκτρονικό πιάνο σε αθόρυβο περιβάλλον αποσπάσματα από μουσικές συνθέσεις διαφόρων ειδών μουσικής και υπολογίστηκαν τα precision και recall για κάθε αλγόριθμο. Τα αποτελέσματα είναι πολύ καλά και για τους τρεις αλγορίθμους με ποσοστά επιτυχίας πάνω από 90%.

Abstract

This thesis concerns the problem of automatic music transcription, that is the conversion of a recorded music signal to a symbolic representation. This procedure requires the extraction of different features such as the pitch, the duration and the volume of each note as well as the determination of the instruments, the beat and other features which concern the expression of the musical composition. Emphasis of this thesis is laid on the consideration and development of techniques for the extraction of the pitch and the duration of each note. To be more precise, the different methods which have been developed both for the process of monophonic and polyphonic musical data are onganized into categories and described herein.

The most important part of this thesis refers to the development of algorithms which detects notes in a musical signal. Two algorithms have been developed for the process of monophonic music. The first processes the musical signal in the time domain by calculating the AMDF coefficients and the second processes it at the frequency domain by using a Constant Q Transform. Furthermore, an algorithm has been developed for the process of polyphonic music. According to this algorithm, the spectrum of the signal in each frame is decomposed in a linear combination of independent spectral components. Each of these components is regarded to have arisen from a different note. For this decomposition, training data have been used which have been recorded by an electric piano. The recognition of notes in a musical signal is based on the approximation of the signal as a linear combination of the training data. For the evaluation of those three algorithms, different kinds of music compositions have been recorded in noiseless environment by using an electric piano. Then, precision and recall are calculated for each algorithm. The results of the three algorithms are very good with success rate over than 90%.

Κεφάλαιο 1

Εισαγωγή

Ο όρος *Music Transcription* αναφέρεται στην επεξεργασία μουσικού σήματος για την μετατροπή του σε συμβολική αναπαράσταση. Η συμβολική αυτή αναπαράσταση μπορεί να είναι μια παρτιτούρα, ένα αρχείο midi ή οποιαδήποτε άλλη μορφή που επιτρέπει την εξαγωγή των βασικών ιδιοτήτων ενός μουσικού κομματιού όπως είναι τα διαφορετικά όργανα, οι νότες που παίζονται από κάθε όργανο, η διάρκεια κάθε νότας, ο ρυθμός του, καθώς και άλλα χαρακτηριστικά που αφορούν την "έκφραση" της μουσικής όπως όροι που καθορίζουν την ένταση του κομματιού σε κάποιο σημείο (pianisimo, piano, forte κτλ), ή όροι που καθορίζουν τον τρόπο που πρέπει να παιχτεί μια νότα (legato, staccato κτλ).

Η μετατροπή ενός μουσικού σήματος σε μια συμβολική μορφή προσφέρει πολλές καινούριες δυνατότητες στο χώρο της μουσικής. Καθώς το πρόβλημα της αναγνώρισης πολύπλοκης πολυφωνικής μουσικής είναι πολύ δύσκολο ακόμα και για έναν έμπειρο μουσικό, μια εφαρμογή που αυτοματοποιεί την παραπάνω διαδικασία δίνει τη δυνατότητα αναπαραγωγής ενός μουσικού κομματιού το οποίο μπορεί κάποιος να διαθέτει μόνο σε audio μορφή. Επιπλέον, δίνεται η δυνατότητα εκμάθησης ενός μουσικού οργάνου χωρίς δάσκαλο με την ανάπτυξη εφαρμογών που μπορούν να "άκούν" την εκτέλεση ενός κομματιού από τον μαθητή και να τον διορθώνουν.

Επίσης, η επεξεργασία μουσικού σήματος μπορεί να βρει εφαρμογή στην συμπίεση δεδομένων ήχου. Η συμβολική αναπαράσταση ενός μουσικού σήματος, για παράδειγμα σε μορφή midi, επιτρέπει σημαντική συμπίεση των δεδομένων ενώ ταυτόχρονα διατηρούνται τα βασικά χαρακτηριστικά του ήχου.

Τέλος, μπορούν να αναπτυχθούν εφαρμογές που θα επιτρέπουν στο χρήστη να ψάχνει σε μια βάση δεδομένων ένα μουσικό κομμάτι με βάση κάποια χαρακτηριστικά του, όπως τη μελωδία ή τον ρυθμό του.

Ωστόσο, το πρόβλημα του *Music Transcription* είναι ιδιαίτερα δύσκολο και δεν υπάρχουν μέχρι σήμερα εφαρμογές που να το λύνουν επαρκώς. Αν και το πρόβλημα της αναγνώρισης μονοφωνικής μουσικής οργάνων που παράγουν αρμονικούς ήχους (όχι κρουστά) έχει λυθεί, τα περισσότερα σύγχρονα συστήματα αναγνώρισης πολυφωνικού ήχου θέτουν αρκετούς περιορισμούς στη λειτουργία τους. Για παράδειγμα μπορεί να επιβάλλουν συγκεκριμένο εύρος για το pitch, δηλαδή αναγνωρίζουν νότες σε ένα συγκεκριμένο εύρος συχνοτήτων, ή να επιτρέπουν συγκεκριμένο βαθμό πολυφωνίας, για παράδειγμα μπορεί να επιτρέπεται μόνο δύο ή τρεις νότες να ακούγονται ταυτόχρονα. Άλλα συστήματα θέτουν κάποιους περιορισμούς ως προς την σχέση μεταξύ των νοτών που ακούγονται ταυτόχρονα. Για παράδειγμα κάποια συστήματα αναγνωρίζουν μόνο *ακόρντα*, συγκεκριμένους δηλαδή συνδυασμούς νοτών, συνήθως τριών ή τεσσάρων, που απέχουν συγκεκριμένες αποστάσεις μεταξύ τους. Επιπλέον συχνά θέτονται περιορισμοί ως προς το είδος και τον αριθμό των διαφορετικών οργάνων. Τα περισσότερα συστήματα επιτρέπουν ένα μόνο όργανο. Τέλος, άλλα συστήματα μπορεί να θέτουν λιγότερους περιορισμούς δείχνοντας ταυτόχρονα μεγαλύτερη ανοχή στα λάθη. Παρακάτω αναφέρονται ενδεικτικά κάποια από τα συστήματα που έχουν υλοποιηθεί μέχρι σήμερα.

Η πρώτη προσπάθεια αναγνώρισης πολυφωνικής μουσικής έγινε το 1975 από τον Moorer[1]. Το σύστημά του επέβαλε αρκετούς περιορισμούς ως προς το εύρος συχνοτήτων των νοτών και ως προς τις σχέσεις μεταξύ νοτών που ακούγονται ταυτόχρονα και επέτρεπε βαθμό πολυφωνίας μέχρι δύο. Το σύστημά του δοκιμάστηκε σε ήχους κιθάρας και βιολιού.

Το 1996 ο Martin[2] εφάρμοσε την τεχνική *Blackboard*, μια τεχνική που ανήκει στο χώρο της τεχνητής νοημοσύνης, για την αναγνώριση πολυφωνικής μουσικής πιάνου επιτρέποντας πολυφωνία μέχρι και τέσσερις φωνές. Το σύστημά του δοκιμάστηκε σε συγκεκριμένο είδος μουσικής, εκκλησιαστική μουσική του Bach, το οποίο συγκέντρωνε ορισμένα επιθυμητά για το συγκεκριμένο σύστημα χαρακτηριστικά όπως νότες που σχηματίζουν ακόρντα μεταξύ τους και αργό tempo. Πέρα από τους περιορισμούς αυτούς, βασικό πρόβλημα του συστήματός του ήταν τα λάθη οκτάβας.

Το 2000 ο Simon Dixon[3] εφάρμοσε το εξής σύστημα: αρχικά έκανε υποδειγματοληψία του σήματος για να μειώσει τα δεδομένα, στη συνέχεια εφάρμοσε STFT και υπολόγισε το Power Spectrum του σήματος, υπολόγισε τα ακρότατα του Power Spectrum εξάγοντας έτσι μια αρχική εικόνα για τις συχνότητες από τις οποίες αποτελείται το σήμα σε κάθε παράθυρο.Τέλος κάθε ένα από αυτά τα ακρότατα μελετάται στο πεδίο της συχνότητας για να προκύψουν τελικά οι νότες του μουσικού σήματος. Το σύστημα αυτό είχε επιτυχία 70-80%.

Το 2002 ο Christopher Raphael[4] δημιούργησε ένα σύστημα βασισμένο στα κρυφά μοντέλα Markov (HMM) για την αναγνώριση πολυφωνικής μουσικής πιάνου θέτοντας περιορισμούς στο εύρος των συχνοτήτων των νοτών καθώς και στο βαθμό πολυφωνίας (επιτρέπονται μέχρι και τέσσερις φωνές). Οι φανερές μεταβλητές στο σύστημά του είναι κάποια χαρακτηριστικά που εξάγονται από το σήμα ενώ οι κρυφές κάποιες “ ετικέτες ” βάση των οποίων προσπαθεί να μοντελοποιηθεί κάθε frame του σήματος. Οι “ ετικέτες ” αυτές περιγράφουν χαρακτηριστικά που αφορούν το pitch ή την κατάσταση της νότας μια δεδομένη χρονική στιγμή, δηλαδή το αν η νότα βρίσκεται στο *attack* τμήμα της (έχει δηλαδή μόλις πατηθεί), στο *sustain* τμήμα της (σταθερό μέρος) ή στο *rest* μέρος της (η νότα ακούγεται πιο εξασθενημένη). Το μοντέλο αυτό είχε ποσοστό αποτυχίας 39%.

Το 2004 η Matija Marolt[5] δημιούργησε ένα σύστημα χρησιμοποιώντας προσαρμοστικούς ταλαντωτές και νευρωνικά δίκτυα. Συγκεκριμένα, χρησιμοποιήθηκαν 88 δίκτυα προσαρμοστικών ταλαντωτών για τη διαδικασία του *partial tracking*, την εύρεση δηλαδή των συχνοτήτων που αποτελούν το μουσικό σήμα σε ένα συγκεκριμένο χρονικό διάστημα. Ένας προσαρμοστικός ταλαντωτής έχει την ικανότητα να προσαρμόζει τη συχνότητά του στη συχνότητα του σήματος εισόδου του. Χρησιμοποιούνται δίκτυα από προσαρμοστικούς ταλαντωτές (έως και δέκα ταλαντωτές σε κάθε δίκτυο) όπου η συχνότητα του n-οστού ταλαντωτή αρχικοποιείται σε συχνότητα n-απλάσια του αρχικού ταλαντωτή. Η χρησιμοποίηση δικτύων προσαρμοστικών ταλαντωτών εισάγει μεγαλύτερη αξιοπιστία στο σύστημα. Τέλος χρησιμοποιούνται 76 νευρωνικά δίκτυα, κάθε ένα από τα οποία είναι εκπαιδευμένο να αναγνωρίζει μία συγκεκριμένη νότα. Τα αποτελέσματα των προσαρμοστικών ταλαντωτών χρησιμοποιούνται από τα νευρωνικά δίκτυα για την για την αναγνώριση τελικά των νοτών. Το σύστημα περιορίστηκε στην αναγνώριση μουσικής πιάνου ενώ η πρώτη οκτάβα του πιάνου (νότες χαμηλών συχνοτήτων) παραλείπεται. Το ποσοστό επιτυχίας

του είναι 80-95%.

Σύντομη περιγραφή της παρούσας εργασίας

Στη συγκεκριμένη εργασία μελετάται η μετατροπή audio ήχου πιάνου σε συμβολική αναπαράσταση. Από το μουσικό σήμα εξάγονται χαρακτηριστικά που αφορούν το pitch και τη διάρκεια μιας νότας ενώ άλλα χαρακτηριστικά όπως ο ρυθμός και η " έκφραση " του κομματιού αγνοούνται. Υλοποιήθηκαν τρία συστήματα, τα δύο αφορούν την επεξεργασία μονοφωνικού μουσικού σήματος ενώ το τρίτο πολυφωνικού.

Το πρώτο σύστημα επεξεργάζεται μονοφωνικά μουσικά σήματα στο πεδίο του χρόνου χρησιμοποιώντας τη συνάρτηση AMDF (average magnitude difference function) για την εύρεση της περιοδικότητας του σήματος σε ένα συγκεκριμένο χρονικό τμήμα, και επομένως την εξαγωγή συμπερασμάτων για το pitch μιας νότας.

Το δεύτερο σύστημα επεξεργάζεται μονοφωνικά μουσικά σήματα στο πεδίο της συχνότητας. Χρησιμοποιείται ο *Constant Q Transform (CQT)* για τον μετασχηματισμό του σήματος στο πεδίο της συχνότητας, σε λογαριθμική κλίμακα συχνοτήτων. Διατηρούνται επίσης δεδομένα εκπαίδευσης για όλες τις νότες. Οι συντελεστές CQT που προκύπτουν για το μουσικό σήμα σε κάθε χρονικό τμήμα συγκρίνονται με τους συντελεστές CQT όλως των δεδομένων εκπαίδευσης με κριτήριο σύγκρισης το *μέσο τετραγωνικό σφάλμα MSE*. Η ελαχιστοποίηση του *MSE* αποτελεί κριτήριο για την εύρεση του pitch μιας νότας.

Το τρίτο σύστημα αποτελεί μια ανάπτυξη του δεύτερου έτσι ώστε να υποστηρίζεται η αναγνώριση πολυφωνικών μουσικών σημάτων. Συγκεκριμένα, θεωρώντας ότι το φασματικό περιεχόμενο του σήματος σε ένα χρονικό διάστημα είναι αποτέλεσμα του συνόλου των συχνοτήτων που παράγονται από κάθε μια από τις νότες που ακούγονται αυτό το χρονικό διάστημα, γίνεται προσπάθεια να βρεθεί γραμμικός συνδυασμός των δεδομένων εκπαίδευσης που να προσεγγίζει όσο το δυνατόν καλύτερα το σήμα.

Όλοι οι αλγόριθμοι οι οποίοι χρησιμοποιήθηκαν σε αυτά τα τρία συστήματα υλοποιήθηκαν στα πλαίσια αυτής της εργασίας εκτός από τον αλγόριθμο

υπολογισμού του Constant Q Transform για τον οποίο αναφέρεται η σχετική βιβλιογραφία.

Η διάρθρωση της εργασίας έχει ως εξής:

- Στο 2ο κεφάλαιο γίνεται μια εισαγωγή σε κάποιους βασικούς όρους για την περιγραφή ενός μουσικού σήματος και περιγράφονται συνοπτικά κάποιοι αλγόριθμοι επεξεργασίας μονοφωνικού σήματος. Οι αλγόριθμοι οργανώνονται σε δύο κατηγορίες στους time-domain και στους frequency-domain αλγορίθμους ανάλογα με το αν η επεξεργασία του σήματος γίνεται στο πεδίο του χρόνου ή στο πεδίο της συχνότητας. Γίνεται επίσης περιγραφή του *Constant Q Transform (CQT)* ενός μετασχηματισμού ιδιαίτερα χρήσιμου στην επεξεργασία δεδομένων μουσικής.

- Στο 3ο κεφάλαιο περιγράφεται το πρόβλημα της επεξεργασίας πολυφωνικού ήχου και γίνεται μια αναφορά σε διαφορετικές μεθόδους που έχουν χρησιμοποιηθεί μέχρι σήμερα.

- Στο 4ο κεφάλαιο περιγράφονται αναλυτικά οι δύο αλγόριθμοι επεξεργασίας μονοφωνικού ήχου πιάνου που υλοποιήθηκαν.

- Στο 5ο κεφάλαιο περιγράφεται αναλυτικά ο αλγόριθμος επεξεργασίας πολυφωνικού ήχου πιάνου που υλοποιήθηκε.

- Στο 6ο κεφάλαιο παρουσιάζονται συγκριτικά τα αποτελέσματα των δύο αλγορίθμων επεξεργασίας μονοφωνικού ήχου καθώς και τα αποτελέσματα του αλγορίθμου για πολυφωνικά μουσικά σήματα.

- Στο 7ο και τελευταίο κεφάλαιο παραθέτονται κάποια συμπεράσματα καθώς και προτάσεις για μελλοντική δουλειά.

Κεφάλαιο 2

Εύρεση του pitch σε μονοφωνικά σήματα

Η μετατροπή μονοφωνικών δεδομένων ήχου σε μια ακολουθίας από νότες είναι ένα σχετικά εύκολο πρόβλημα. Για τον εντοπισμό μιας νότας χρειάζεται να προσδιοριστούν τρία βασικά χαρακτηριστικά: *pitch*, *onset* και η διάρκεια της νότας.

Το *pitch* σχετίζεται με την *βασική συχνότητα* μιας νότας. Η *βασική συχνότητα* ορίζεται ως το αντίστροφο της περιόδου ενός περιοδικού σήματος ήχου. Τα σήματα ήχου που παράγονται από τα περισσότερα όργανα χαρακτηρίζονται από μία βασική συχνότητα και έναν αριθμό *αρμονικών*. Ως *αρμονική* ενός σήματος ορίζεται μια συνιστώσα στο πεδίο των συχνοτήτων με συχνότητα ακέραιο πολλαπλάσιο της βασικής συχνότητας του σήματος. Για παράδειγμα για ένα σήμα με βασική συχνότητα f0 η πρώτη αρμονική θα έχει συχνότητα 2f0 και η τρίτη 3f0. Το *pitch* χρησιμοποιείται για την ταυτοποίηση μιας συγκεκριμένης νότας ανεξάρτητα από το όργανο το οποίο την παράγει.

Κατά το μεγαλύτερο μέρος της διάρκειας μιας νότας τα βασικά χαρακτηριστικά της αντίστοιχης κυματομορφής παραμένουν σταθερά και έτσι η εξαγωγή συμπερασμάτων για το pitch είναι μια εύκολη διαδικασία. Ωστόσο, η διαδικασία αυτή είναι δυσκολότερη κατά το *attack* μέρος της νότας, δηλαδή το πρώτο τμήμα της νότας, το οποίο είναι συνήθως πιο "θορυβώδες".

Ο όρος *onset* χρησιμοποιείται για να περιγράψει χρονικά την *αρχή* μιας νότας, δηλαδή τη χρονική στιγμή που εμφανίζεται η συγκεκριμένη νότα στην ακολουθία

των νοτών.

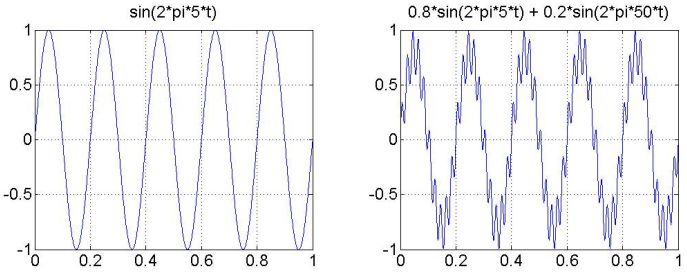
Στο κεφάλαιο αυτό περιγράφονται, χωρίς να αναφέρονται αλγοριθμικές λεπτομέρειες, κάποιες από τις τεχνικές επεξεργασίας μονοφωνικού μουσικού ήχου οι οποίες κατηγοριοποιούνται ανάλογα με το αν η επεξεργασία του σήματος γίνεται στο πεδίο του χρόνου ή της συχνότητας.

2.1 Time-Domain Αλγόριθμοι

Οι αλγόριθμοι αυτοί επεξεργάζονται το σήμα στο πεδίο του χρόνου. Στηριζόμενοι στο γεγονός ότι σε ένα περιοδικό σήμα κάποια χαρακτηριστικά επαναλαμβάνονται σε συγκεκριμένες χρονικές στιγμές προσπαθούν να εντοπίσουν τη βασική περίοδο του σήματος, το αντίστροφο δηλαδή της βασικής συχνότητας. Στην κατηγορία αυτή ανήκουν οι αλγόριθμοι zero crossing rate, autocorrelation και AMDF.

2.1.1 Zero Crossing Rate

Ο αλγόριθμος *Zero Crossing Rate (ZCR)* προσπαθεί να εξάγει συμπεράσματα για την συχνότητα ενός περιοδικού σήματος απλά εξετάζοντας πόσες φορές το σήμα διασχίζει τον άξονα του μηδενός. Η μέθοδος αυτή είναι απλή και αποτελεσματική για σήματα που το φασματικό τους περιεχόμενο αποτελείται από συχνότητες συγκεντρωμένες γύρω από τη βασική συχνότητα του σήματος αλλά για πιο πολύπλοκα σήματα δημιουργούνται προβλήματα όπως φαίνεται και από τα παρακάτω γραφήματα.



Σχήμα 2.1: Αριστερά ένα περιοδικό σήμα με συχνότητα f=5, δεξιά το άθροισμα δύο περιοδικών σημάτων με συχνότητες f=5, f=50.

Στο σχήμα που απεικονίζεται δεξιά είναι δύσκολο να εξαχθούν ασφαλή συμπεράσματα για τη βασική συχνότητα με τη μέθοδο ZCR καθώς η δεύτερη συνιστώσα στο πεδίο των συχνοτήτων (f=50) προκαλεί κάποιες επιπλέον μηδενικές τιμές στο σήμα.Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί είτε χρησιμοποιώντας το μέσο όρο και τη διακύμανση των μηδενικών σημείων του σήματος για να αποφασίσουμε ποια από αυτά δίνουν πληροφορίες για τη βασική συχνότητα, ή φιλτράροντας κατάλληλα το σήμα ώστε να κοπούν οι υψηλότερες συχνότητες.

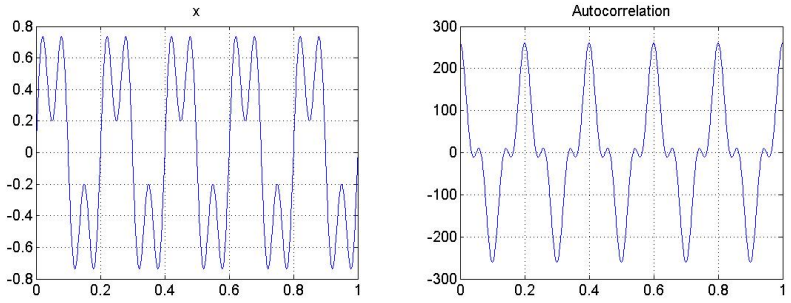
2.1.2 Συνάρτηση αυτοσυσχέτισης (Autocorrelation)

Η συνάρτηση αυτοσυσχέτισης χρησιμοποιείται για να συγκρίνει διαφορετικά στιγμιότυπα της ίδιας κυματομορφής. Παρέχει δηλαδή μια απεικόνιση της μεταβολής ενός σήματος με το χρόνο. Συγκεκριμένα, η συνάρτηση αυτοσυσχέτισης ενός σήματος x μεγέθους N δίνεται από τον τύπο:

$$R_x(m) = \sum_{n=0}^{N-1-m} x(n)x(n+m)$$

Περιοδικά σήματα εμφανίζουν περιοδικότητα στη συνάρτηση αυτοσυσχέτισής τους. Η συνάρτηση αυτοσυσχέτισης θα έχει τοπικά μέγιστα για $m = kT$ όπου T η περίοδος του σήματος x. Αυτό είναι λογικό καθώς τα χαρακτηριστικά του x επαναλαμβάνονται περιοδικά οπότε το σήμα $x(n)$ θα εμφανίζει μεγάλη ομοιότητα με το σήμα $x(n+kT)$, τον εαυτό του δηλαδή μετατοπισμένο κατά ακέραιο αριθμό περιόδων. Στο παρακάτω γράφημα φαίνεται η συνάρτηση αυτοσυσχέτισης της συνάρτησης $x = 0.5 \sin 2\pi 5t + 0.5 \sin 2\pi 15t$

Η περίοδος του σήματος μπορεί να υπολογιστεί εντοπίζοντας τη θέση του πρώτου από τα τοπικά μέγιστα της συνάρτησης αυτοσυσχέτισης, εξαιρώντας βέβαια αυτό που βρίσκεται στη θέση 0. Η μέθοδος αυτή είναι αποτελεσματική για τον εντοπισμό της περιόδου, ωστόσο όταν το φασματικό περιεχόμενο του σήματος αποτελείται και από άλλες ισχυρές συνιστώσες εκτός από τη βασική του συχνότητα το έργο αυτό γίνεται δυσκολότερο καθώς πρέπει να καθοριστεί πιο από τα τοπικά μέγιστα που προκύπτουν αντιστοιχεί στη περίοδο του σήματος.



(α′) x (β′) R_x

Σχήμα 2.2: Συνάρτηση αυτοσυσχέτισης της $x = 0.5 \sin 2\pi 5t + 0.5 \sin 2\pi 15t$

2.1.3 Συνάρτηση AMDF

Η συνάρτηση AMDF (average magnitude difference function) λειτουργεί παρόμοια με την συνάρτηση αυτοσυσχέτισης, δίνει δηλαδή μια εικόνα για την μεταβολή ενός σήματος στο χρόνο. Αυτό επιτυγχάνεται όχι πολλαπλασιάζοντας δύο διαφορετικά στιγμιότυπα του ίδιου σήματος όπως γινόταν με τη συνάρτηση αυτοσυσχέτισης αλλά παίρνοντας τη διαφορά τους. Συγκεκριμένα, η συνάρτηση AMDF ενός σήματος x μεγέθους N δίνεται από τον τύπο:

$$D_x(m) = \sum_{n=0}^{N-1-m} |x(n) - x(n+m)|$$

Ομοίως με τη συνάρτηση αυτοσυσχέτισης η AMDF ενός περιοδικού σήματος με περίοδο T είναι και αυτή περιοδική με περίοδο T. Συγκεκριμένα στα σημεία $m = kT$ η $D_x(m)$ εμφανίζει τοπικά ελάχιστα. Έτσι η περίοδος του σήματος υπολογίζεται εντοπίζοντας τη θέση του πρώτου τοπικού ελάχιστου της συνάρτησης AMDF. Η μέθοδος αυτή έχει μικρότερη χρονική πολυπλοκότητα από τη συνάρτηση αυτοσυσχέτισης καθώς δεν απαιτείται η πράξη του πολλαπλασιασμού.

2.2 Frequency-Domain Αλγόριθμοι

Η επεξεργασία ενός σήματος στο πεδίο της συχνότητας δίνει περισσότερες δυνατότητες για τον εντοπισμό του pitch, καθώς οι συνιστώσες του φάσματος

του σήματος διαχωρίζονται. Οι αλγόριθμοι αυτοί στηρίζονται στο γεγονός ότι οι συχνότητες ενός σήματος παραγόμενου από ένα μουσικό όργανο σχετίζονται αρμονικά μεταξύ τους. Στην ενότητα αυτή παραθέτονται κάποιες τεχνικές επεξεργασίας μουσικού σήματος στο πεδίο των συχνοτήτων. Παρουσιάζεται επίσης ο *Constant Q Transform*, ένας μετασχηματισμός στο πεδίο των συχνοτήτων καταλληλότερος από τον *Fourier Transform* για την επεξεργασία μουσικών δεδομένων.

2.2.1 Component Frequency Ratios

Η μέθοδος αυτή αναπτύχθηκε το 1979 από τον Martin Piszczalski[6]. Σύμφωνα με αυτήν τη μέθοδο, αρχικά το σήμα μετασχηματίζεται στο πεδίο των συχνοτήτων και εντοπίζονται τα *partials* του σήματος, οι συχνότητες δηλαδή του σήματος με *amplitude* μεγαλύτερο από ένα συγκεκριμένο κατώφλι. Στη συνέχεια τα *partials* που προέκυψαν συνδυάζονται ανά δύο. Για κάθε ζευγάρι που προκύπτει πρέπει να βρεθεί ο μικρότερος αριθμός αρμονικών που να εμπεριέχει αυτό το ζευγάρι συχνοτήτων. Για παράδειγμα αν στο ζευγάρι ανήκουν οι συχνότητες 122Hz και 204Hz ο μικρότερος αριθμός αρμονικών είναι 3 για την πρώτη συχνότητα και 5 για τη δεύτερη και η βασική συχνότητα που προκύπτει είναι 40Hz.

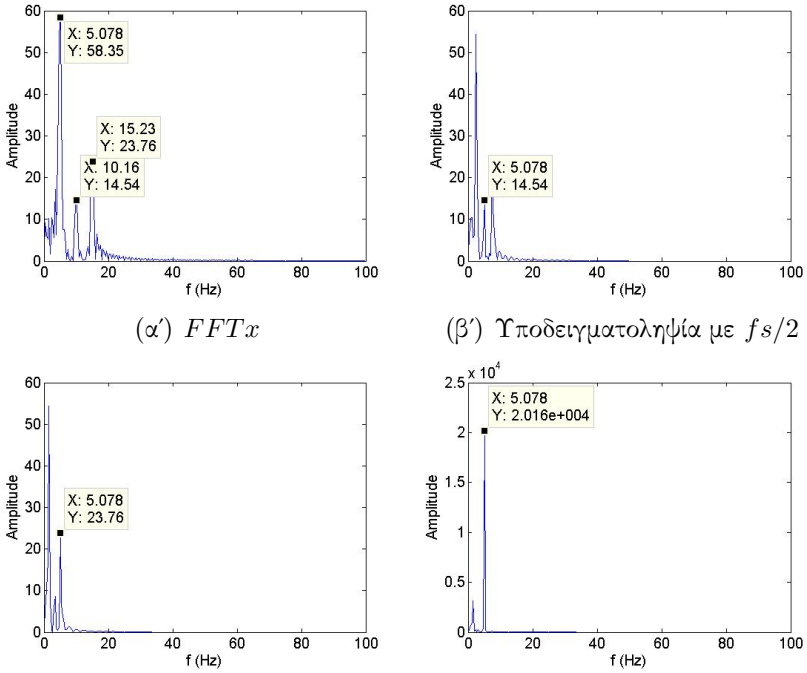
Υπολογίζονται οι πιθανές βασικές συχνότητες που προκύπτουν από όλους τους συνδυασμούς των *partials* και επιλέγεται τελικά ως βασική συχνότητα αυτή που προκύπτει από τα περισσότερα ζευγάρια. Η συμβολή κάθε ζευγαριού δεν είναι ισότιμη. Ζευγάρια συχνοτήτων με μεγαλύτερο *amplitude* επηρεάζουν περισσότερο στην τελική απόφαση.

Η μέθοδος αυτή είναι αρκετά αποτελεσματική, καθώς είναι δυνατός ο εντοπισμός της βασικής συχνότητας ενός σήματος ακόμα και αν αυτή απουσιάζει από το φασματικό περιεχόμενο του σήματος.

2.2.2 Harmonic Product Spectrum (HPS)

Η μέθοδος αυτή στηρίζεται στο γεγονός ότι φάσμα μιας νότας θα έχει τοπικά μέγιστα στη βασική συχνότητα της νότας καθώς και σε έναν αριθμό αρμονικών συχνοτήτων.

Σύμφωνα με τη μέθοδο αυτή αρχικά υπολογίζεται ο SFT του σήματος. Στη συνέχεια παράγονται N σήματα όπου N ο αριθμός των τοπικών μέγιστων του SFT. Το κάθε ένα από αυτά τα σήματα είναι αποτέλεσμα υποδειγματοληψίας του αρχικού, το πρώτο δηλαδή έχει συχνότητα δειγματοληψίας $fs/2$, όπου fs η συχνότητα δειγματοληψίας του αρχικού, το δεύτερο $fs/3$ και ομοίως τα υπόλοιπα. Τέλος τα σήματα αυτά πολλαπλασιάζονται μεταξύ τους. Η μέγιστη τιμή του σήματος που θα προκύψει από αυτό τον πολλαπλασιασμό αντιστοιχεί στη βασική συχνότητα του σήματος. Η διαδικασία αυτή είναι εμφανής στο παράδειγμα που ακολουθεί.



(α′) $FFTx$

(β′) Υποδειγματοληψία με $fs/2$

(γ′) Υποδειγματοληψία με $fs/3$

(δ′) Πολλαπλασιασμός των τριών σημάτων

Σχήμα 2.3: Στο γράφημα (α) φαίνεται ο μετασχηματισμός Fourier του σήματος $x = 0.6\sin(2\pi 5t) + 0.15\sin(2\pi 10t) + 0.25\sin(2\pi 15t)$, η βασική συχνότητα του σήματος είναι 5Hz και έχει δύο αρμονικές 10Hz και 15Hz. Στα γραφήματα (β) και (γ) γίνεται η υποδειγματοληψία. Στο γράφημα (δ) απεικονίζεται ο πολλαπλασιασμός των τριών σημάτων. Εμφανίζεται μέγιστη τιμή στη συχνότητα 5Hz

Η μέθοδος έχει χαμηλό υπολογιστικό κόστος και είναι ανθεκτική σε θόρυβο. Μειονέκτημα της μεθόδου είναι ότι είναι λιγότερο ακριβής για για χαμηλές συχνότητες απ'οτι για ψηλές. Επιπλέον, η ανάλυση εξαρτάται από τον αριθμό των δειγμάτων που χρησιμοποιούνται για τον μετασχηματισμό στο πεδίο των συχνοτήτων. Λιγότερα δείγματα συνεπάγονται μικρότερο υπολογιστικό κόστος αλλά χαμηλότερη ανάλυση.

2.2.3 Cepstrum

Το Cepstrum ορίζεται ως το τετράγωνο του μετασχηματισμού Fourier του λογαρίθμου του τετραγώνου του μετασχηματισμού Fourier ενός σήματος. Το Cepstrum δηλαδή ενός σήματος x δίνεται από την σχέση $|F\{\log(|F\{x\}|^2)\}|^2$.

Η χρησιμοποίηση του Cepstrum για την εύρεση του pitch στηρίζεται στο γεγονός ότι μια κυματομορφή που παράγεται από μια νότα θα περιέχει συχνότητες που σχετίζονται αρμονικά μεταξύ τους. Έτσι ο μετασχηματισμός Fourier του σήματος αυτού θα έχει κάποια μέγιστα στη βασική συχνότητα και στις αρμονικές της. Παίρνοντας το λογάριθμο του amplitude του DFT του σήματος τα πλάτη αυτών των μεγίστων μειώνονται και παίρνουμε ένα περιοδικό σήμα στο πεδίο των συχνοτήτων με περίοδο τη βασική συχνότητα του σήματος. Τέλος, παίρνοντας το μετασχηματισμό Fourier του νέου αυτού σήματος θα προκύψει ένα μόνο μέγιστο που αντιστοιχεί στη περίοδο του αρχικού σήματος, στο αντίστροφο της βασικής συχνότητας δηλαδή.

Η μέθοδος αυτή βέβαια είναι κατάλληλη μόνο για σήματα των οποίων οι συχνότητες σχετίζονται μεταξύ τους με τέτοιο τρόπο ώστε να προκύπτει περιοδικότητα παίρνοντας το λογάριθμο του φάσματος.

2.2.4 Στατιστικές μέθοδοι

Για την εύρεση του pitch έχουν αναπτυχθεί επίσης και κάποιες στατιστικές μέθοδοι, οι οποίες βασίζονται σε τεχνικές αναγνώρισης προτύπων και μέγιστης πιθανοφάνειας. Σε γενικές γραμμές, οι αλγόριθμοι αυτοί διατηρούν πληροφορίες για το φασματικό περιεχόμενο όλων των δυνατών νοτών. Για να βρεθεί η βασική συχνότητα μιας νέας παρατήρησης, πρέπει το φάσμα της να συγκριθεί με τα υπάρχοντα φάσματα των οποίων η πηγή και επομένως η βασική συχνότητα είναι γνωστά. Υπολογίζεται η πιθανότητα μια παρατήρηση να προέρχεται από

κάποια συγκεκριμένη πηγή και επιλέγεται ως πηγή της παρατήρησης εκείνη για την οποία μεγιστοποιείται η πιθανότητα αυτή.

2.2.5 Constant Q Transform

Για τον μετασχηματισμό ενός μουσικού σήματος στο πεδίο των συχνοτήτων συνήθως δεν προτιμάται ο *Discreet Fourier Transform (DFT)* αλλά ο *Constant Q Transform (CQT)*[7] καθώς ο DFT δημιουργεί κάποια προβλήματα.

Οι συχνότητες που χρησιμοποιούνται στη δυτική μουσική σχηματίζουν γεωμετρική κλίμακα, ενώ στον DFT οι συχνότητες απέχουν ίσες αποστάσεις μεταξύ τους. Αυτό σημαίνει ότι οι περισσότερες συνιστώσες ενός μετασχηματισμού Fourier δεν αντιστοιχούν σε συχνότητες ενός μουσικού σήματος. Ο Constant Q Transform είναι ένας μετασχηματισμός στο πεδίο των συχνοτήτων ο οποίος όμως χρησιμοποιεί μια συστοιχία από φίλτρα με γεωμετρικά τοποθετημένες συχνότητες. Οι συχνότητες αυτές δίνονται από τη σχέση $f_k = f_0 \cdot 2^{\frac{k}{b}}$ όπου f_0 μια αρχική συχνότητα με βάση την οποία υπολογίζονται οι υπόλοιπες, b ο αριθμός των φίλτρων ανά οκτάβα και k ακέραιος. Θέτοντας κατάλληλες τιμές στις μεταβλητές f_0, b και k μπορούμε να πάρουμε συχνότητες που αντιστοιχούν σε βασικές συχνότητες νοτών. Για παράδειγμα θέτοντας $f_0 = 27.5$, $b = 12$ και $k = 0...87$ προκύπτουν οι συχνότητες που αντιστοιχούν στις νότες του πιάνου. Με αυτόν τον τρόπο ο Constant Q Transform επιτυγχάνει να χρησιμοποιεί λιγότερες συνιστώσες συχνοτήτων απ'οτι ο DFT ενώ ταυτόχρονα καλύπτει αποτελεσματικά ένα μεγάλο εύρος συχνοτήτων.

Ένα επιπλέον πρόβλημα του μετασχηματισμού Fourier είναι ότι προσφέρει σταθερή *ανάλυση* για όλες τις συχνότητες. Ο όρος *ανάλυση* ορίζεται ως ο ρυθμός δειγματοληψίας διαιρεμένος με το μέγεθος του παραθύρου σε δείγματα, είναι δηλαδή ουσιαστικά η απόσταση μεταξύ δύο διαδοχικών σημείων στον άξονα των συχνοτήτων. Αυτό σημαίνει ότι ο βαθμός διαχωρισμού των συχνοτήτων στον μετασχηματισμό Fourier εξαρτάται από τη συχνότητα και είναι ίσος με $\frac{f_k}{Df} = \frac{Nf_k}{f_s}$ όπου f_k η τιμή της συχνότητας στο k-οστό σημείο στον άξονα συχνοτήτων, f_s η συχνότητα δειγματοληψίας, και N το μήκος του παραθύρου σε δείγματα. Δηλαδή για νότες με υψηλή βασική συχνότητα επιτυγχάνεται μεγάλος βαθμός διαχωρισμού ενώ για νότες με χαμηλή βασική συχνότητα μικρότερος. Αυτό πρακτικά σημαίνει ότι για πολύ ψηλές νότες χρησιμοποιούνται περισσότερα

δείγματα απ'οτι πραγματικά χρειάζονται ενώ για πολύ χαμηλές νότες τα δείγματα δεν επαρκούν.

Για να λυθεί το πρόβλημα αυτό πρέπει ο βαθμός διαχωρισμού των συχνοτήτων δηλαδή ο λόγος $\frac{f}{Df}$ να παραμένει σταθερός. Ο Constant Q Transform ανταποκρίνεται σε αυτήν την απαίτηση καθώς ισχύει:

$$\frac{f_k}{Df} = \frac{f_k}{f_{k+1} - f_k} = \frac{f_0 \cdot 2^{\frac{k}{b}}}{f_0 \cdot 2^{\frac{k+1}{b}} - f_0 \cdot 2^{\frac{k}{b}}} = \frac{1}{2^{\frac{1}{b}} - 1}$$

Έτσι ο Constant Q Transform προσφέρει σταθερό βαθμό διαχωρισμού $Q = (2^{\frac{1}{b}} - 1)^{-1}$ για όλες τις συχνότητες. Για να επιτευχθεί αυτό η εξίσωση μετασχηματισμού ενός σήματος στο πεδίο των συχνοτήτων χρησιμοποιώντας SFT που δίνεται από την εξίσωση:

$$X(f) = \sum_{n<N} x[n] \cdot e^{-j2\pi nf/N}$$

τροποποιείται χρησιμοποιώντας παράθυρα μεταβλητού μήκους N_k με $N_k = Q \cdot \frac{f_s}{f_k}$ με f_s τη συχνότητα δειγματοληψίας και k ακέραιος με $k < K$ όπου η τιμή του K εξαρτάται από τη μέγιστη συχνότητα f_{max}. Συνοψίζοντας ο Constant Q Transform ενός σήματος x δίνεται από τη σχέση:

$$X(f) = \sum_{n<N_k} x[n] \cdot e^{-j2\pi nQ/N_k} \tag{2.1}$$

όπου

$$Q = (2^{\frac{1}{b}} - 1)^{-1} \tag{2.2}$$

$$f_k = f_0 \cdot 2^{\frac{k}{b}} \tag{2.3}$$

$$N_k = \left\lceil Q \cdot \frac{f_s}{f_k} \right\rceil \tag{2.4}$$

$$K = \left\lceil b \log_2(\frac{f_{max}}{f_0}) \right\rceil \tag{2.5}$$

Ένα ακόμα πλεονέκτημα του μετασχηματισμού ενός σήματος στο πεδίο των συχνοτήτων χρησιμοποιώντας Constant Q Transform είναι ότι επιτυγχάνεται συγκεκριμένη απόσταση μεταξύ των αρμονικών συχνοτήτων του. Η πρώτη δηλαδή αρμονική ενός σήματος με βασική συχνότητα f θα απέχει από την f κατά log(2) η δεύτερη κατά log(3/2) η τρίτη κατά log(3) και ομοίως οι υπόλοιπες. Η απόσταση δηλαδή είναι ανεξάρτητη της βασικής συχνότητας f. Έτσι δημιουργείται ένα σταθερό πρότυπο ίδιο για κάθε διαφορετική νότα. Διαφορές υπάρχουν μόνο στο amplitude των συχνοτήτων. Η πληροφορία που προκύπτει από αυτές τις διαφορές μπορεί να χρησιμοποιηθεί για τον καθορισμό του *timbre*, του μουσικού οργάνου δηλαδή που αποτελεί πηγή του συγκεκριμένου μουσικού σήματος.

Κεφάλαιο 3

Εύρεση του pitch σε πολυφωνικά σήματα

Το πρόβλημα αναγνώρισης πολυφωνικού ήχου είναι πολύ πιο δύσκολο από αυτό της αναγνώρισης μονοφωνικού. Η δυσκολία του προβλήματος οφείλεται στο γεγονός ότι συχνότητες που παράγονται από διαφορετικές νότες μοιράζονται κοινό χώρο συχνοτήτων.

Όπως έχει αναφερθεί, μια νότα παράγει μια κυματομορφή με συνιστώσες στο πεδίο των συχνοτήτων τη βασική της συχνότητα f_0 και ένα αριθμό αρμονικών συχνοτήτων $\{h_k\}$. Όταν δύο νότες *σχετίζονται αρμονικά* μεταξύ τους, δηλαδή αν f_0^1 και f_0^2 οι βασικές τους συχνότητες, ισχύει η σχέση $f_0^1 = \frac{m}{n} f_0^2$, με m, n μικροί ακέραιοι, τότε είναι αναμενόμενο ότι κάποιες από τις αρμονικές των δύο νοτών θα επικαλύπτονται. Συγκεκριμένα, έστω η αρμονική h_k^1 της νότας με βασική συχνότητα f_0^1, ισχύει $h_k^1 = k f_0^1 = k \cdot \frac{m}{n} \cdot f_0^2$. Άρα, όλες οι αρμονικές h_k^1 με k ακέραιο πολλαπλάσιο του n συμπίπτουν με αρμονικές της δεύτερης νότας με βασική συχνότητα f_0^2. Δηλαδή ισχύει $h_n^1 = h_m^2, h_{2n}^1 = h_{2m}^2$ και ομοίως οι υπόλοιπες.

Στη δυτική μουσική είναι πολύ συχνό φαινόμενο οι νότες που ακούγονται ταυτόχρονα να σχετίζονται αρμονικά μεταξύ τους. Έτσι, κάποιες φορές είναι δύσκολο να διαχωριστούν οι διάφορες συνιστώσες στο πεδίο των συχνοτήτων και να ομαδοποιηθούν ανάλογα με την πηγή (νότα) που τις δημιούργησε. Το πρόβλημα αυτό γίνεται ακόμα πιο δύσκολο όταν οι βασικές συχνότητες δύο νοτών που ακούγονται ταυτόχρονα αποτελούν ακέραιο πολλαπλάσιο η μία της άλλης, συνδέονται δηλαδή με τη σχέση $f_0^1 = m f_0^2$, με m μικρό ακέραιο. Στην

περίπτωση αυτή όλες οι αρμονικές της δεύτερης νότας επικαλύπτονται από αυτές της πρώτης. Έτσι, είναι πιθανό η δεύτερη νότα να μην εντοπιστεί λόγω της παρουσίας της πρώτης.

Για τους λόγους αυτούς, οι απλές μέθοδοι που αναφέρθηκαν για την επεξεργασία μονοφωνικού σήματος είναι αναποτελεσματικές για το συγκεκριμένο πρόβλημα. Στο κεφάλαιο αυτό γίνεται μια αναφορά σε κάποιες τεχνικές που χρησιμοποιούνται για την επεξεργασία πολυφωνικού ήχου. Συχνά, μπορεί να χρησιμοποιούνται και συνδυασμοί αυτών των τεχνικών.

3.1 Προσομοίωση του ανθρώπινου ακουστικού μοντέλου

Ο Meddis ανέπτυξε μια τεχνική για την εύρεση του pitch ενός ήχου που προσομοιάζει τη διαδικασία που εκτελείται στο ανθρώπινο ακουστικό μοντέλο[8]. Το μοντέλο αυτό είναι γνωστό και με το όνομα unitary model. Συνοπτικά τα βήματα του αλγορίθμου είναι τα εξής:

1. Αρχικά το σήμα περνάει από μια συστοιχία ζωνοπερατών φίλτρων με λογαριθμικά κατανεμημένες ζώνες συχνοτήτων. Συνήθως χρησιμοποιούνται 40-128 φίλτρα με ζώνες που επικαλύπτονται μερικώς.

2. Στη συνέχεια τα σήματα που προκύπτουν στην έξοδο των φίλτρων συμπιέζονται, περνάνε από έναν ημιανορθωτή όπου μηδενίζονται οι αρνητικές τιμές του σήματος, και τέλος από ένα χαμηλοπερατό φίλτρο.

3. Κάθε σήμα που προκύπτει από το προηγούμενο βήμα χωρίζεται σε μικρά τμήματα (frames). Για κάθε frame υπολογίζεται η συνάρτηση αυτοσυσχέτισης.

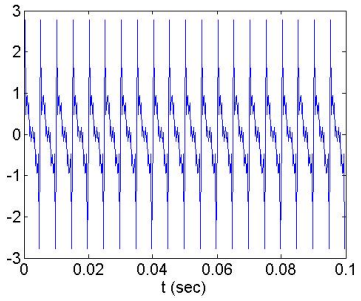
4. Οι συναρτήσεις αυτοσυσχέτισης που προκύπτουν σε κάθε κανάλι αθροίζονται δημιουργώντας μια *αθροιστική συνάρτηση αυτοσυσχέτισης (SACF, summary autocorrelation function)*. Συγκεκριμένα η συνάρτηση SACF ορίζεται ως εξής:

$$sr_\tau(t) = \sum_b r_{\tau,b}(t)$$

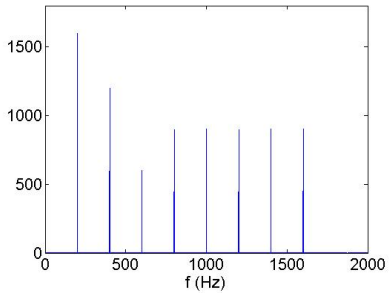
όπου $r_{\tau,b}(t)$ η συνάρτηση αυτοσυσχέτισης στην έξοδο του καναλιού b για το frame τ. Η περίοδος του σήματος τη χρονική στιγμή τ αντιστοιχεί στην τιμή t για την οποία η συνάρτηση $sr_\tau(t)$ μεγιστοποιείται.

Όπως φαίνεται και από τα παραπάνω, ο τρόπος εύρεσης της περιόδου με αυτήν τη μέθοδο διαφοροποιείται από τις τεχνικές που έχουν εξεταστεί μέχρι τώρα οι οποίες εκτιμούν την περιοδικότητα είτε εξετάζοντας το σήμα στο πεδίο του χρόνου είτε εξετάζοντας το φάσμα του σήματος. Σε αυτήν την μέθοδο συμπεράσματα για το pitch εξάγονται εξετάζοντας την περιοδικότητα της περιβάλλουσας του σήματος στο πεδίο του χρόνου.

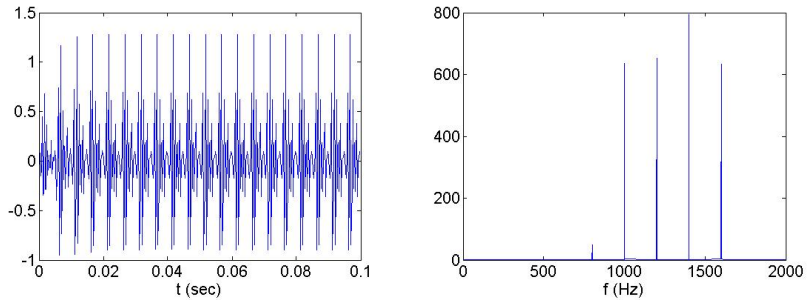
Η βασική ιδέα πίσω από αυτήν την μέθοδο είναι ότι ένα σήμα που αποτελείται από περισσότερες από μία συνιστώσες στο πεδίο των συχνοτήτων εμφανίζει περιοδικότητα στην περιβάλλουσά του στο πεδίο του χρόνου. Αυτό οφείλεται στο γεγονός ότι οι διάφορες συνιστώσες είτε αλληλοαναιρούνται είτε συμβάλλουν περιοδικά. Ο ρυθμός με τον οποίο οι συνιστώσες συμβάλουν και αναιρούνται εξαρτάται από τις αποστάσεις μεταξύ των συχνοτήτων. Όταν οι συχνότητες σχετίζονται μεταξύ τους αρμονικά, η απόσταση μεταξύ συνιστωσών η οποία κυριαρχεί είναι αυτή της βασικής συχνότητας. Έτσι είναι δυνατή η εξαγωγή της περιοδικότητας από το τελικό γράφημα που προκύπτει. Η παραπάνω διαδικασία είναι εμφανής στο γράφημα 3.1. Με την ημιανόρθωση του σήματος δημιουργούνται οι συχνότητες του φάσματος της περιβάλλουσας του σήματος, έτσι περνώντας το σήμα από ένα χαμηλοπερατό φίλτρο κατασκευάζεται η περιβάλλουσά του.



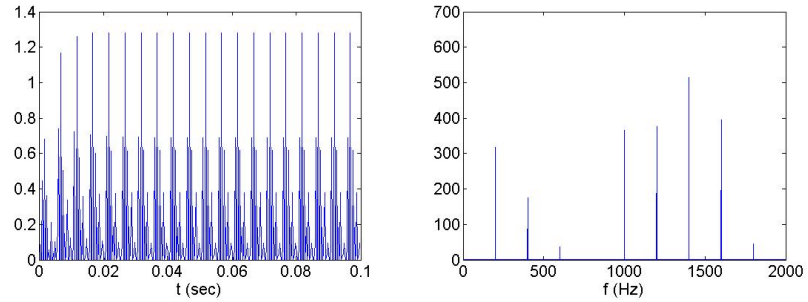
(α′)



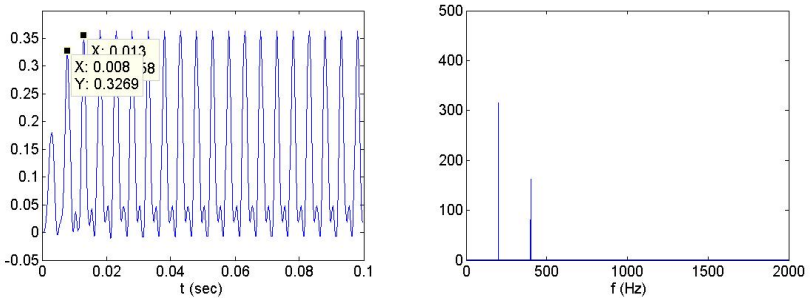
(β′)



(γ′) (δ′)



(ε′) (ϛ′)



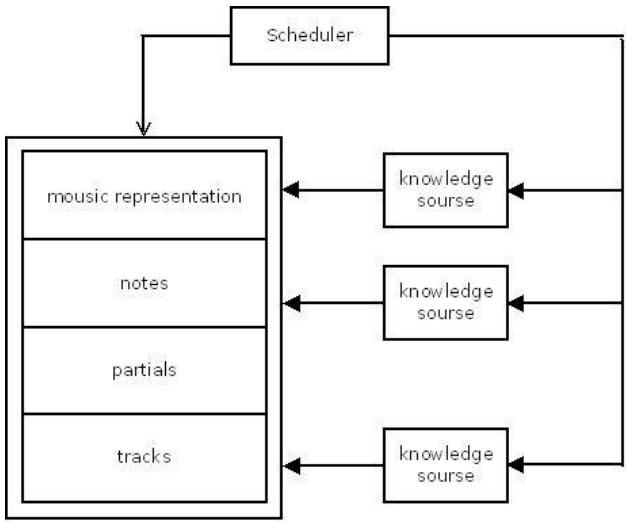
(ζ′) (η′)

Σχήμα 3.1: Στο γράφημα (α) φαίνεται το αρχικό σήμα x με βασική συχνότητα $200Hz$ επτά αρμονικές,ενώ στο (β) ο μετασχηματισμός φουριερ του. Στη συνέχεια το σήμα περνάει από ζωνοπερατό φίλτρο με εύρος συχνοτήτων 1000-$1600Hz$. Στα γραφήματα (γ) και (δ) φαίνεται το σήμα στην έξοδο του φίλτρου στο πεδίο του χρόνου και στο πεδίο της συχνότητας αντίστοιχα. Στη συνέχεια το σήμα περνάει από ημιανορθωτή. Τα αντίστοιχα γραφήματα είναι τα (ε) και (στ). Τέλος το σήμα περνάει από ένα χαμηλοπερατό φίλτρο. Τα αντίστοιχα γραφήματα είναι τα (ζ), (η). Είναι εμφανές ότι η περίοδος τους σήματος είναι 5 msec.

Η παραπάνω μέθοδος έχει εφαρμοστεί με επιτυχία και στον υπολογισμό του pitch πολυφωνικού ήχου. Οι αλγόριθμοι που στηρίζονται σε αυτήν την τεχνική λειτουργούν ως εξής: αρχικά εντοπίζουν τη βασική συχνότητα μιας νότας όπως περιγράφηκε παραπάνω, στη συνέχεια οι συχνότητες που εκτιμάται ότι αντιστοιχούν σε αυτή τη νότα διαγράφονται με κατάλληλο φιλτράρισμα και η διαδικασία επαναλαμβάνεται για τις υπόλοιπες συνιστώσες[9].

3.2 Συστήματα Blackboard

Τα συστήματα Blackboard είναι μια τεχνική από το χώρο της τεχνητής νοημοσύνης. Οι αλγόριθμοι αυτοί ξεκινούν από κάποια αρχικά δεδομένα για ένα πρόβλημα τα οποία συνεχώς ενημερώνονται με τη συμβολή κάποιων πηγών πληροφορίας μέχρι να επιτευχθεί η λύση του προβλήματος. Το όνομα Blackboard είναι μεταφορικό και αναφέρεται σε μια ομάδα επιστημόνων που σημειώνουν διαδοχικά κάποιες προτάσεις σε ένα μαυροπίνακα για την επίλυση ενός προβλήματος.



Σχήμα 3.2: Γραφική απεικόνιση του μοντέλου

Ένα σύστημα Blackboard αποτελείται από τρία τμήματα: τον *μαυροπίνακα* όπου αναπαρίστανται οι αρχικές υποθέσεις οι οποίες ενημερώνονται ανάλογα με τις καινούριες γνώσεις που προσθέτονται στο σύστημα, τις πηγές γνώ-

σεις οι οποίες επεξεργάζονται κατάλληλα τα δεδομένα του *μαυροπίνακα* όταν ικανοποιούνται κάποιες συνθήκες, και τον χρονοπρογραμματιστή (scheduler) ο οποίος καθορίζει τη σειρά με την οποία θα αλληλεπιδρούν οι πηγές γνώσεις στα δεδομένα του *μαυροπίνακα*. Ένα παράδειγμα εφαρμογής συστημάτων Blackboard στην αναγνώριση πολυφωνικού ήχου φαίνεται στο γράφημα 3.2.

Τα δεδομένα στο *μαυροπίνακα* οργανώνονται ιεραρχικά. Στο χαμηλότερο επίπεδο ο όρος *tracks* αναφέρεται σε δεδομένα που προκύπτουν από κάποια αρχική επεξεργασία του σήματος. Συνήθως αφορούν πληροφορία σχετική με τις συχνότητες και τα αντίστοιχα amplitude ενός σήματος στα σημεία εκείνα όπου η ενέργειά του είναι σχετικά μεγάλη. Αυτή η προσέγγιση χρησιμοποιήθηκε από τους Martin[10] και Bello[11]. Τα δεδομένα αυτά εισάγονται ως αρχική πληροφορία στο σύστημα. Στο δεύτερο στάδιο ο όρος *partials* αναφέρεται σε μια ενδιάμεση μορφή πληροφορίας μεταξύ των *tracks* και των *notes*. Στο τρίτο στάδιο της ιεραρχίας τα *partials* ομαδοποιούνται για το σχηματισμό νοτών. Στο τελευταίο στάδιο εξάγονται κάποιες επιπλέον πληροφορίες για το μουσικό κομμάτι. Για παράδειγμα οι νότες μπορεί να συνδυάζονται για το σχηματισμό ακόρντων. Τα τρία αυτά τελευταία στάδια προκύπτουν από την επεξεργασία των *tracks* από τις *πηγές γνώσης*. Κάθε *πηγή γνώσης* ενεργοποιείται από τον (scheduler) και εξετάζει αν ικανοποιούνται κάποιες συγκεκριμένες συνθήκες. Ανάλογα με το ποιες από αυτές τις συνθήκες ικανοποιούνται εκτελούνται και οι αντίστοιχες ενέργειες.

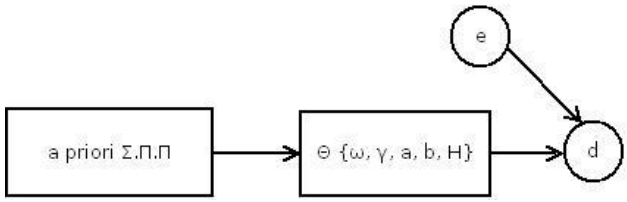
3.3 Πιθανοτικά μοντέλα

Οι αλγόριθμοι αυτοί στηρίζονται σε κάποιες αρχικές γνώσεις που διαθέτουν για τη μορφή ενός μουσικού σήματος και δημιουργούν ένα μοντέλο με κάποιες άγνωστες παραμέτρους οι οποίες πρέπει να εκτιμηθούν. Για παράδειγμα ένα σήμα μπορεί να μοντελοποιηθεί σύμφωνα με την παρακάτω συνάρτηση:

$$d = \sum_{q=1}^{Q} \gamma_q \sum_{h=1}^{H} a_{q,h} \sin h\omega_q t + b_{q,h} \cos h\omega_q t + e$$

όπου Q ο μέγιστος αριθμός νοτών που μπορούν να ακούγονται ταυτόχρονα, H ο αριθμός αρμονικών της νότας q, ω_q η βασική συχνότητα της q, $a_{q,h}$ και $b_{q,h}$ συντελεστές που προσδιορίζουν τα amplitude και phase της αρμονικής h της

νότας q και γ_q δυαδικός συντελεστής που προσδιορίζει αν υπάρχει ή όχι η q. Η μεταβλητή e χρησιμοποιείται για τη μοντελοποίηση του θορύβου



Σχήμα 3.3: Γραφική απεικόνιση του μοντέλου

Το σύνολο των μεταβλητών του σήματος $\Theta_q\{\gamma_q, H, a_{q,h}, b_{q,h}, \omega_q\}$ πρέπει να εκτιμηθεί με κριτήριο τη μεγιστοποίηση της πιθανότητας $P(d|\Theta, e)$. Για την εκτίμηση των παραμέτρων συνήθως χρησιμοποιείται ένας εκτιμητής κατά Bays ο οποίος επιτρέπει την ενσωμάτωση εκ των προτέρων γνώσης για τις συναρτήσεις πυκνότητας πιθανότητας των παραμέτρων. Η εκτίμηση των παραμέτρων γίνεται χρησιμοποιώντας Marcov Chain Monte Carlo μεθόδους. Στο σχήμα 3.3 απεικονίζεται γραφικά η σχέση μεταξύ των μεταβλητών του μοντέλου.

Η προσέγγιση αυτή χρησιμοποιήθηκε από τον Davy[12] καθώς και από τους Walmsley, Godsill και Rayner[13].

Μια διαφορετική μοντελοποίηση του προβλήματος στηρίζεται στη χρησιμοποίηση κρυφών μοντέλων Marcov, (HMM)[4]. Σε αυτήν την περίπτωση, οι φανερές μεταβλητές του συστήματος είναι κάποια χαρακτηριστικά που μπορούν να εξαχθούν από το μουσικό σήμα ενώ οι κρυφές μεταβλητές αντιστοιχούν σε κάποιες ετικέτες βάση των οποίων πρέπει να μοντελοποιηθεί κάθε frame του σήματος. Οι ετικέτες αυτές αναφέρονται σε πληροφορία σχετική με τη βασική συχνότητα, την ενέργεια και το τμήμα της νότας που αντιπροσωπεύει το συγκεκριμένο frame (attack, sustain, rest). Κάθε κατάσταση του HMM χαρακτηρίζεται από μια συγκεκριμένη ακολουθία ετικετών. Συμβολίζοντας το διάνυσμα των παρατηρήσεων με το γράμμα $\mathbf{O}$ και τις καταστάσεις με το γράμμα X, κάθε frame του σήματος πρέπει να αντιστοιχηθεί με μία κατάσταση έτσι ώστε να ικανοποιείται η σχέση:

$$\widehat{x}^N = \arg\max_{\widehat{x}^N} P(X^N = x^N |\ O^N = o^N)$$

με $\widehat{x}^N = \{\widehat{x}^1, \widehat{x}^2, ...\widehat{x}^N\}$.

Η διαδικασία αυτή προϋποθέτει την εκπαίδευση του μοντέλου έτσι ώστε να υπολογιστούν οι πιθανότητες $P(X^N = x^N |\, O^N = o^N)$. Για την εύρεση των ζητούμενων πιθανοτήτων συνήθως εφαρμόζεται ο αλγόριθμος forward-backward σε κάποια γνωστά μουσικά κομμάτια, δηλαδή σήματα των οποίων κάθε frame έχει αντιστοιχηθεί με κάποια από τις καταστάσεις του HMM.

3.4 Μάθηση μοντέλου από τα δεδομένα

Οι τεχνικές αυτές, σε αντίθεση με τα πιθανοτικά μοντέλα που περιγράφηκαν παραπάνω, δεν προσπαθούν να δημιουργήσουν κάποιο συγκεκριμένο παραμετρικό μοντέλο χρησιμοποιώντας κάποια γνώση για το μουσικό σήμα. Αντίθετα, το σήμα πρέπει να προσδιοριστεί από τα δεδομένα. Συγκεκριμένα, οι αλγόριθμοι αυτοί προσπαθούν να αναλύσουν το φάσμα του σήματος σε κάποιες γνωστές συνιστώσες. Αυτό επιτυγχάνεται χρησιμοποιώντας τεχνικές *Non-Negative Matrix Factorazation (NMF)* και *sparse coding*[14],[15].

Οι αλγόριθμοι NMF προσπαθούν να προσεγγίσουν έναν πίνακα $X \in R^{\geq 0, M \times N}$ διαστάσεων $N \times M$ ως το γινόμενο δύο μη αρνητικών πινάκων $W \in R^{\geq 0, M \times R}$ και $H \in R^{\geq 0, R \times N}$ με κριτήριο την ελαχιστοποίηση κάποιας συνάρτησης κόστους η οποία αποτελεί μέτρο της απόκλισης του πίνακα X από το γινόμενο WH. Συνήθως η συνάρτηση κόστους έχει την παρακάτω μορφή:

$$C = ||X - W \cdot H||_F$$

όπου ο τελεστής $|| \cdot ||_F$ αντιστοιχεί στην Frobenius Norm η οποία ορίζεται ως εξής:

$$||A||_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} a_{i,j}}$$

Ο όρος *sparse coding* αναφέρεται σε ένα είδος νευρωνικών κωδίκων όπου η παρουσία ενός αντικειμένου προσδιορίζεται από την ενεργοποίηση ενός μικρού αριθμού νευρώνων.

Η περιγραφή ενός μουσικού σήματος σύμφωνα με τη μέθοδο NMF μπορεί να γίνει ως εξής:

$$X^t(f) = \sum_{m}^{M} H_m^t W_m(f) + R^t(f)$$

όπου $X^t(f)$ είναι το φάσμα του σήματος τη χρονική στιγμή t, $W_n(f)$ το φάσμα του αντικειμένου n, H_n^t μια μεταβλητή που προσδιορίζει το πόσο δυνατή είναι η παρουσία του αντικειμένου n τη χρονική στιγμή t, ενώ ο όρος $R^t(f)$ χρησιμοποιείται για τη μοντελοποίηση των συνιστωσών του πίνακα X που δεν μπορούν να εκφραστούν ως γινόμενο των W και H.

Με την προσέγγιση του προβλήματος χρησιμοποιώντας ένα *sparse coding* αλγόριθμο, θεωρείται ότι το σήμα $X \in R^{\geq 0,N}$ προσδιορίζεται από το διάνυσμα $H_t \in R^{\geq 0,M}$ με τον τρόπου που περιγράφηκε παραπάνω. Η καινούρια πληροφορία που εισάγεται είναι ότι ένα μεγάλο ποσοστό των στοιχείων του διανύσματος H_t αναμένεται να έχει μηδενική τιμή.

Οι αλγόριθμοι αυτοί προσπαθούν να μάθουν έναν κατάλληλο πίνακα $W \in R^{\geq 0,M \times N}$ χρησιμοποιώντας κάποια δεδομένα εκπαίδευσης χωρίς να υποθέτουν καμία γνώση για τη μορφή του σήματος. Αυτό έχει σαν αποτέλεσμα την κατασκευή μοντέλων απλούστερων από αυτά που βασίζονται σε εκ των προτέρων γνώση για το μουσικό σήμα όπως τα συστήματα Blackboard και τα πιθανοτικά μοντέλα. Επιπλέον δίνουν τη δυνατότητα υποστήριξης διαφορετικών οργάνων ενώ συνήθως τα συστήματα που βασίζονται στη *γνώση* υποστηρίζουν κάποιο συγκεκριμένο όργανο.

Κεφάλαιο 4

Υλοποίηση αλγορίθμων μετατροπής μονοφωνικής μουσικής σε συμβολική αναπαράσταση

Στο κεφάλαιο αυτό περιγράφονται οι δύο αλγόριθμοι που υλοποιήθηκαν σε Matlab για την επεξεργασία μονοφωνικού ήχου. Ο πρώτος επεξεργάζεται το σήμα στο πεδίο του χρόνου προσπαθώντας να εντοπίσει περιοδικότητα υπολογίζοντας τη συνάρτηση AMDF. Ο δεύτερος υπολογίζει τον Constant Q Transform του σήματος και χρησιμοποιεί αναγνώριση προτύπων για την εύρεση της βασικής συχνότητας. Υλοποιήθηκε επίσης και ένας Onset Detector για να καθοριστεί η διάρκεια της κάθε νότας.

4.1 Υλοποίηση time-domain αλγορίθμου για τον υπολογισμό του pitch

Ο αλγόριθμος που υλοποιήθηκε χρησιμοποιεί τη συνάρτηση AMDF που περιγράφηκε στην ενότητα 2.1.3 για τον εντοπισμό περιοδικότητας σε ένα μονοφωνικό σήμα ήχου.

Το σήμα χωρίζεται σε μικρά τμήματα (frames) χρησιμοποιώντας παράθυρα μήκους 64msec και ολίσθηση παραθύρου κάθε 32msec. Υπάρχει δηλαδή 50% επικάλυψη μεταξύ των παραθύρων. Γίνεται επίσης zero-padding στο σήμα, δηλαδή προσθέτονται μηδενικά στο τέλος του έτσι ώστε να χωράει ακέραιος αριθμός παραθύρων. Συγκεκριμένα, συμβολίζοντας με N τον αριθμό των δειγμάτων του σήματος, W τον αριθμό των δειγμάτων σε κάθε παράθυρο και ολίσθηση

παραθύρου κατά S δείγματα, τότε πρέπει να προστεθούν $S - mod(N, W)$ μηδενικά.

Για κάθε frame του σήματος υπολογίζεται η ενέργεια. Η ενέργεια ενός frame ορίζεται ως το άθροισμα των τετραγώνων των τιμών του amplitude των δειγμάτων του σήματος και δίνεται από τον τύπο:

$$E_n = \sum_{m=1}^{W} |x_n(m)|^2 \tag{4.1}$$

όπου x_n το σήμα στο frame n και W ο αριθμός των δειγμάτων σε ένα frame. Αν η ενέργεια E_n είναι μικρότερη ενός threshold T τότε το frame n αγνοείται στον υπολογισμό του pitch καθώς θεωρείται ότι δεν αντιστοιχεί σε κάποια νότα αλλά σε παύση. Ορίζεται $T = 10^{-9}$

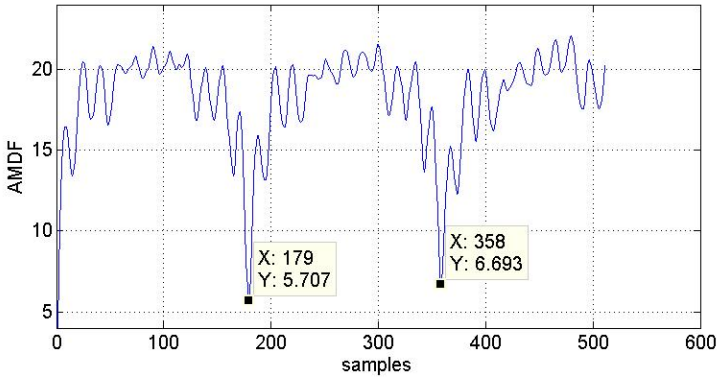
Για κάθε frame n με $E_n > 10^{-9}$ υπολογίζεται η συνάρτηση AMDF. Η συνάρτηση AMDF υπολογίζεται ως εξής:

$$D_n(m) = \sum_{k=(n-1)W+1}^{nW} x(k) - x(k+m) \tag{4.2}$$

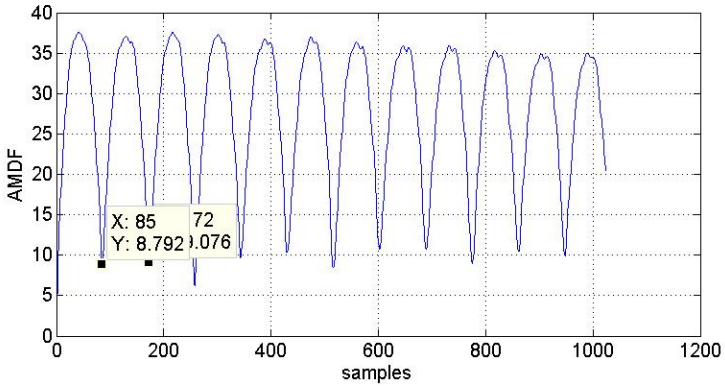
με $m = 1, 2, ... \lceil \frac{W}{2} \rceil$. Η συνάρτηση αυτή θα εμφανίζει τοπικά ελάχιστα στα σημεία $\frac{k}{m} T_0$ με $k = 1, 2...$, $T_0 = \frac{1}{F_0}$ και F_0 η βασική συχνότητα του σήματος. Το m είναι μικρός θετικός ακέραιος και εκφράζει την παρουσία της m αρμονικής στο σήμα. Δηλαδή κάποια από τα τοπικά ελάχιστα, συγκεκριμένα τα πιο έντονα, αντιστοιχούν στη βασική συχνότητα του σήματος και κάποια στις αρμονικές του.

Στο σχήμα 4.1(α′) απεικονίζεται η συνάρτηση AMDF για ένα frame κατά τη διάρκεια του οποίου ακούγεται η νότα F2 με συχνότητα 87.3Hz ενώ στο 4.1(β′) η νότα $F\#3$ με συχνότητα 184.99Hz. Η συχνότητα δειγματοληψίας και για τα δύο σήματα είναι 16000Hz. Όπως φαίνεται το σήμα στο (α′) εμφανίζει ελάχιστη τιμή κάθε 179 δείγματα δηλαδή είναι περιοδικό με περίοδο $T = \frac{179}{16000} = 0.0112sec$ και βασική συχνότητα $F_0 = \frac{1}{0.0112} = 89.28Hz$ ενώ αντίστοιχα για το (β′) υπολογίζεται $F_0 = 188.68Hz$.

Για να υπολογιστεί η περίοδος του σήματος για ένα frame πρέπει να βρεθούν τα τοπικά ελάχιστα τα οποία αντιστοιχούν στη περίοδο του σήματος και όχι σε κάποια αρμονική συχνότητα. Αυτό μπορεί να γίνει παίρνοντας τη μέση τιμή των



(α′)



(β′)

Σχήμα 4.1: Στο σχήμα (α′) η συνάρτηση AMDF για την νότα F2 και στο (β′) για τη νότα F#3.

τοπικών ελαχίστων και προσδιορίζοντας τη περίοδο με βάση το πρώτο ελάχιστο με τιμή μικρότερη της μέσης τιμής. Ωστόσο, κάποια σήματα, όπως αυτό του σχήματος 4.1.(α′), τα οποία έχουν πολλά τοπικά ελάχιστα που αντιστοιχούν σε αρμονικές θα έχουν μεγάλη μέση τιμή συγκριτικά με τις τιμές των τοπικών ελαχίστων που αντιστοιχούν στη βασική συχνότητα. Έτσι είναι πιθανόν να "επιβιώσει" κάποιο από τα ελάχιστα που αντιστοιχούν σε αρμονικές με αποτέλεσμα τον εσφαλμένο υπολογισμό του pitch. Για να λυθεί το πρόβλημα αυτό, συμπληρωματικά με το κριτήριο της μέσης τιμής, πρέπει να υπολογιστεί και η διακύμανση των τοπικών ελαχίστων. Ορίζοντας n ένα πίνακα $1 \times N$ με τις τιμές των δειγμάτων που αντιστοιχούν σε τοπικά ελάχιστα, το νέο κατώφλι T

υπολογίζεται ως εξής :

$$T = E\{D(n)\} - \sqrt{E\{(\mu_{D(n)} - D^2_{(n)})^2\}} \tag{4.3}$$

Η περίοδος του σήματος εκφράζεται από το δείγμα $n(i)$ για το οποίο ισχύει $D(n(i)) < T$ και $i < j \; \forall j \neq i$ με $D(n(j)) < T$. Ωστόσο, ούτε αυτό το κριτήριο είναι επαρκές. Για ψηλότερες νότες όπως η νότα $F\#3$ της οποίας η συνάρτηση AMDF φαίνεται στο σχήμα 4.1(β′) η τιμή του T μπορεί να είναι μικρότερη από την τιμή του $D(n(i))$ με $n(i)$ το δείγμα που αντιστοιχεί στη σωστή περίοδο. Συγκεκριμένα για το σήμα που απεικονίζεται στο 4.2(β′), ισχύει T = 8.0583 και $D(n(i)) = 8.792$. Δηλαδή το σωστό $n(i)$ θα κοπεί από το κατώφλι.

Τα προβλήματα αυτά είναι συνηθέστερα στο attack τμήμα της νότας. Για να λυθούν αυτά τα προβλήματα υλοποιήθηκε ο εξής αλγόριθμος που περιγράφεται παρακάτω.

Αλγόριθμος 4.1. *Υπολογισμός περιοδικότητας στη συνάρτηση AMDF για ένα frame του σήματος. Η περίοδος σε δείγματα αποθηκεύεται στη μεταβλητή t0.*

Βήμα 1. *Υπολογίζεται η ενέργεια E_n όπως περιγράφεται στη σχέση 4.1. Αν ισχύει $E_n < 10^{-9}$ ο αλγόριθμος τερματίζει, διαφορετικά προχωράει στο 2ο βήμα.*

Βήμα 2. *Υπολογίζεται η συνάρτηση AMDF D σύμφωνα με τη σχέση 4.2.*

Βήμα 3. *Υπολογίζονται τα δείγματα στα οποία εμφανίζονται τα τοπικά ελάχιστα της D και αποθηκεύονται στον πίνακα n. Ο n δηλαδή αποτελεί πίνακα με δείκτες στα τοπικά ελάχιστα της D.*

Βήμα 4. *Εφαρμόζεται threshold $T1 = 0.4(\max(D) - \min(D))$ στον πίνακα n. Οι τιμές $i \in n$ για τις οποίες ισχύει $D(i) > T1$ διαγράφονται. Οι τιμές αυτές είναι πολύ υψηλές για να θεωρηθούν υποψήφια t0 και πιθανότατα αντιστοιχούν σε αρμονικές του σήματος.*

Βήμα 5. *Γίνεται προσπάθεια εντοπισμού κάποιας περιοδικότητας από τα στοιχεία του πίνακα n. Συγκεκριμένα για κάθε δείγμα $i \in n$ πρέπει να βρεθούν όλα τα δείγματα $j \in n$ για τα οποία ισχύει $j \in [ki - \delta, ki + \delta]$ με k θετικό ακέραιο και δ μικρό θετικό ακέραιο. Το k εκφράζει τον αριθμό των περιόδων που χωρίζουν τα δείγματα i και*

j. Αναζητούνται δηλαδή μέσα στον πίνακα n ακέραια πολλαπλάσια του i. Η μεταβλητή δ δηλώνει ότι δεν είναι ανάγκη οι τιμές ki και j να ταυτίζονται απόλυτα. Για παράδειγμα για n = (80 120 158 200 241), i = 80 και δ = 3 τα δείγματα j που ικανοποιούν την παραπάνω απαίτηση είναι τα j =158, 241.

Βήμα 6. *Το δείγμα i για το οποίο βρέθηκε ο μεγαλύτερος αριθμός M δειγμάτων j που ικανοποιούν τη σχέση του προηγούμενου βήματος επιλέγεται ως t0. Ο αριθμός M πρέπει να είναι μεγαλύτερος του δύο, δηλαδή πρέπει να υπάρχουν τουλάχιστον τρεις περίοδοι στο frame.*

Βήμα 7. *Αν για κανένα i δε βρεθεί αριθμός M > 2 τότε το παράθυρο μεγαλώνει κατά W/2 και επαναλαμβάνονται τα βήματα 1-3.*

Βήμα 8. *Αν πάλι δε βρεθεί δείγμα i που να ικανοποιεί τις παραπάνω σχέσεις υπολογίζεται το κατώφλι T όπως ορίστηκε στη σχέση 4.3 και επιλέγεται ως t0 το πρώτο i που ικανοποιεί τη σχέση D(i) < T.*

Βήμα 9. *Αν δε βρεθεί καμία τιμή που να ικανοποιεί το βήμα 8 τότε το t0 παραμένει κενό και ο αλγόριθμος τερματίζει.*

Χρησιμοποιώντας την τιμή $t0$ υπολογίζεται η βασική συχνότητα f_0. Αν η ενέργεια E_n που υπολογίστηκε είναι μικρότερη του 10^{-9} τότε δεν μπορεί να υπολογιστεί η βασική συχνότητα. Διαφορετικά το f_0 υπολογίζεται από τη σχέση $f_0 = Fs/t0$ με Fs τη συχνότητα δειγματοληψίας. Αν δεν έχει βρεθεί τιμή για το $t0$ για το frame n από τον αλγόριθμο 1 η βασική συχνότητα του n θεωρείται ίση με αυτή του frame $n-1$, θέτω δηλαδή $f_0^n = f_0^{n-1}$.

Τέλος, η βασική συχνότητα που υπολογίζεται για κάθε frame αντιστοιχίζεται με μία νότα σύμφωνα με τη σχέση 4.4:

$$k = round(12 \log_2 \frac{f_0}{440} + 49) \tag{4.4}$$

όπου η συνάρτηση round υλοποιεί την πράξη της στρογγυλοποίησης. Το k παίρνει μια τιμή 1 έως 88. Η τιμή αυτή αντιστοιχεί σε μία από τις 88 νότες του πιάνου.

4.2 Υλοποίηση frequency-domain αλγορίθμου για τον υπολογισμό του pitch

Στην ενότητα αυτή περιγράφεται ένας αλγόριθμος για την αναγνώριση μονοφωνικής μουσικής πιάνου χρησιμοποιώντας τεχνικές αναγνώρισης προτύπων. Ειδικότερα, συλλέγονται κάποια δείγματα εκπαίδευσης και υπολογίζεται ο Constant Q Transform τους. Για κάθε νέο άγνωστο δείγμα, υπολογίζεται ο CQT και συγκρίνεται με τα δεδομένα εκπαίδευσης. Με κριτήριο την ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος, επιλέγεται κάποιο από τα δείγματα εκπαίδευσης βάση του οποίου αποδίδεται ταυτότητα στο άγνωστο δείγμα.

4.2.1 Υπολογισμός του Constant Q Transform

Επειδή ο υπολογισμός του Constant Q Transform όπως περιγράφεται στην ενότητα 2.2.5 έχει μεγάλη χρονική πολυπλοκότητα χρησιμοποιήθηκε ο Fast Fourier Transform για τον υπολογισμό του όπως περιγράφεται από τους Judith C. Brown και Miller S. Puckette στο "*An efficient algorithm for the calculation of a constant Q transform*"[16]. Σύμφωνα με αυτόν τον αλγόριθμο ο CQT εκφράζεται ως το γινόμενο δύο πινάκων: $x \cdot T^*$ όπου x πίνακας μεγέθους $1 \times N$ με $N \geq N_k \forall k < K$ με τα K και N_k όπως ορίστηκαν στις σχέσεις 2.5 και 2.4 αντίστοιχα, και T^* πίνακας μεγέθους $N \times K$ όπου η τιμή $T_{n,k}$ ενός στοιχείου στην n γραμμή της στήλης k δίνεται από τη σχέση 4.5:

$$T_{n,k} = \begin{cases} \frac{1}{N_k} w_{N_k}[n] e^{j2\pi nQ/N_k} & \text{if } n < N_k \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Οι υπολογισμοί μπορούν να επιταχυνθούν αν υπολογιστούν και διατηρηθούν αποθηκευμένες οι τιμές του πίνακα T^*. Με αυτόν τον τρόπο ο T^* υπολογίζεται μόνο μία φορά. Ωστόσο, ο πολλαπλασιασμός των x και T^* εξακολουθεί να έχει μεγάλο υπολογιστικό κόστος. Η ιδέα των Brown και Puckette ήταν να υλοποιήσουν τους υπολογισμούς στο πεδίο της συχνότητας. Ορίζεται πίνακας S μεγέθους $N \times K$ και S_k η k στήλη του S με $S_k = DFT(T_{:,k})$. Η στήλη k δηλαδή του S αποτελεί το διακριτό μετασχηματισμό Fourier της στήλης k του πίνακα T. Οι περισσότερες τιμές του S θα είναι πολύ χαμηλές. Έτσι εφαρμόζοντας κάποιο threshold στον S μηδενίζοντας τις τιμές μικρότερες του threshold ο πίνακας που προκύπτει θα έχει πολλά μηδενικά στοιχεία επιταχύνοντας έτσι τους υπολογισμούς. Ο CQT δίνεται από τη σχέση 4.6:

$$CQT\{x\} = \frac{1}{N} x^{ft} \cdot S^* \tag{4.6}$$

όπου x^{ft} ο μη κανονικοποιημένος μετασχηματισμός Fourier του πίνακα x μεγέθους $1 \times N$ και $N = \max(N_k)$.

Υλοποιήθηκε η συνάρτηση sparsekernel σε Matlab για την κατασκευή του πίνακα $\frac{S^*}{N}$. Η συνάρτηση παίρνει σαν ορίσματα την βασική συχνότητα της χαμηλότερης νότας minFreq = 65.4064, τη βασική συχνότητα της υψηλότερης νότας maxFreq = 4186.01, τον αριθμό των νοτών ανά οκτάβα bins=12, τη συχνότητα δειγματοληψίας fs, και το threshold που ορίζεται thresh = 0.005. Ο ψευδοκώδικας της συνάρτησης φαίνεται παρακάτω.

Ψευδοκώδικας 1 Συνάρτηση sparsekernel για τον υπολογισμό του πίνακα S

function sparKernel = sparseKerrnel(minFreq,maxFreq,bins,fs,thresh)

% Αρχικοποιήσεις
$Q = (2^{\frac{1}{bins}} - 1)^{-1}$;, $K = \left\lceil bins \cdot \log_2 \frac{maxFreq}{minFreq} \right\rceil$;, $fftLen = \left\lceil Q \cdot \frac{fs}{minFreq} \right\rceil$;
tempKernel = zeros(1,fftLen);, sparKernel = NULL;
for k = K down to 1
  $len = \left\lceil Q \cdot \frac{fs}{minFreq \cdot 2^{(k-1)/bins}} \right\rceil$;
  SET vector a = [0 1 2... len-1]^T ;
  tempKernel(1 to len) = $\frac{hamming(len)}{len \cdot \exp^{2 \cdot \pi \cdot j \cdot Q \cdot a/len}}$;
  specKernel = FFT(tempKernel); % Μετασχηματιστμός Fourier
  SET index = FIND ($|specKernel| \leq thresh$);% Εφαρμογή threshold
  SET specKernel(index) = 0;% Μηδενίζονται οι τιμές μικρότερες του threshold
  $sparKernel = sparse([specKernel; sparKernel])$; %Προστίθεται η νέα στήλη στον πίνακα.
end FOR sparKernel = $sparKernel^*/fftLen$; %O sparKernel είναι ο $\frac{S^*}{N}$

Τέλος υλοποιήθηκε η συνάρτηση cqt(x,sparKernel) όπου γίνεται ο πολλαπλασιασμός του μετασχηματισμού Fourier του x και του πίνακα sparKernel σύμφωνα με την εξίσωση 4.6

4.2.2 Συλλογή των δειγμάτων εκπαίδευσης

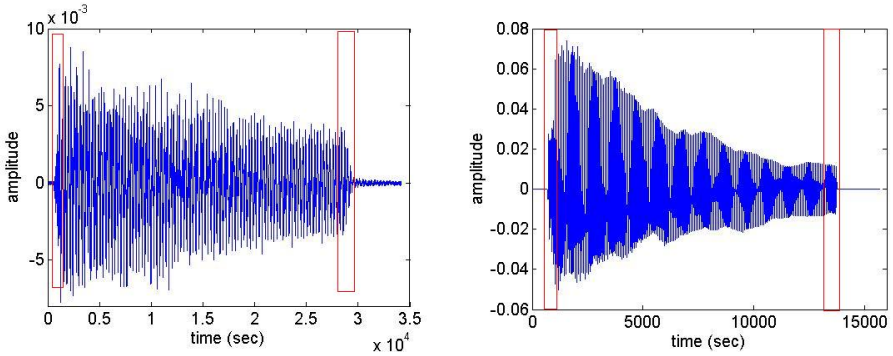
Τα δείγματα εκπαίδευσης ηχογραφήθηκαν χρησιμοποιώντας ηλεκτρικό πιάνο σε αθόρυβο περιβάλλον με ρυθμό δειγματοληψίας 32000Hz. Ηχογραφήθηκαν όλες οι νότες του πιάνου εκτός τις 15 πρώτες νότες, δηλαδή τις τρεις πρώτες νότες συν μία οκτάβα. Οι νότες αυτές δημιουργούν λάθη στην αναγνώριση και επιπλέον δεν χρησιμοποιούνται πολύ συχνά σε κομμάτια πιάνου. Έτσι θεωρήθηκε καλύτερο να ηχογραφηθούν οι 73 από τις 88 νότες του πιάνου.

Μια νότα μπορεί να παιχτεί σε διαφορετικές εντάσεις: *pianissimo* (πολύ απαλά), *piano* (απαλά), *mezzo-piano* (μέτρια απαλά), *mezzo-forte* (μέτρια δυνατά) *forte* (δυνατά) και *fortissimo* (πολύ δυνατά). Τα φασματικά χαρακτηριστικά της νότας διαφοροποιούνται ανάλογα με την ένταση της. Φυσικά αλλάζουν μόνο τα amplitude των αρμονικών και όχι οι θέσεις στις οποίες εντοπίζονται. Έτσι επιλέχθηκε κάθε νότα να ηχογραφηθεί σε τρεις διαφορετικές εντάσεις: απαλή (*pianissimo* ή *piano*), μέτρια (*mezzo-piano* ή *mezzo-forte*) και δυνατή (*forte* ή *fortissimo*). Δημιουργούνται δηλαδή συνολικά $73 \cdot 3 = 219$ αρχεία .wav.

Για κάθε ένα από αυτά τα 219 αρχεία πρέπει να υπολογιστεί ο Constant Q Transform. Φυσικά ο CQT δε θα υπολογιστεί για ολόκληρο το αρχείο αλλά για κάποια επιλεγμένα frames του αρχείου. Επειδή τα φασματικά χαρακτηριστικά μιας νότας διαφοροποιούνται κατά το attack, sustain, και rest τμήμα της, για κάθε αρχείο .wav υπολογίζεται ο CQT για ένα frame του αρχικού τμήματος της νότας, για ένα του μεσαίου και για ένα του τελευταίου τμήματός της.

Κάθε αρχείο .wav διαβάζεται από τη Matlab με τη συνάρτηση wavread. Το σήμα χωρίζεται σε frames μήκους 0.14sec με επικάλυψη 50%, ολίσθηση παραθύρου δηλαδή κάθε 0.07sec. Από αυτά τα frames πρέπει να επιλεχθούν μόνο τρία ως δεδομένα εκπαίδευσης. Για να γίνει αυτό, πρέπει να βρεθούν τα attack, sustain, και rest τμήματα της νότας. Στο σχήμα 4.2 φαίνεται ένα παράδειγμα δύο σημάτων όπου σημειώνονται τα attack και rest τμήματά τους.

Για να βρεθεί η αρχή και το τέλος της νότας μέσα σε ένα .wav αρχείο χρησιμοποιείται η ενέργεια. Η ενέργεια υπολογίζεται σε πολύ μικρά τμήματα του σήματος μήκους 3.5msec και ορίζεται όπως στην εξίσωση 4.1. Όταν βρεθεί το πρώτο χρονικό διάστημα (t1, t2) στο οποίο η ενέργεια είναι μεγαλύτερη ενός threshold, τότε θεωρείται ως αρχή της νότας η χρονική στιγμή t1 η οποία



(α′) C2 pianissimo (β′) A4 forte

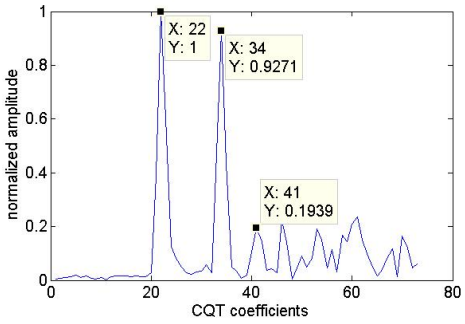
Σχήμα 4.2: Απεικόνιση του σήματος στο πεδίο του χρόνου για δύο διαφορετικές νότες. Τα attack και rest τμήματά τους σημειώνονται σε κόκκινα πλέσια.

αποθηκεύεται στη μεταβλητή *start*. Επειδή νότες διαφορετικής έντασης έχουν διαφορετική ενέργεια, πρέπει να οριστούν και διαφορετικά threshold για κάθε μια από τις τρεις εντάσεις. Συγκεκριμένα για τις δυνατές νότες ορίζεται threshold $T_1 = 2 \cdot 10^{-5}$, για τις μεσαίας έντασης $T_2 = 2 \cdot 10^{-6}$ ενώ για τις απαλές $T_3 = 5 \cdot 10^{-8}$.

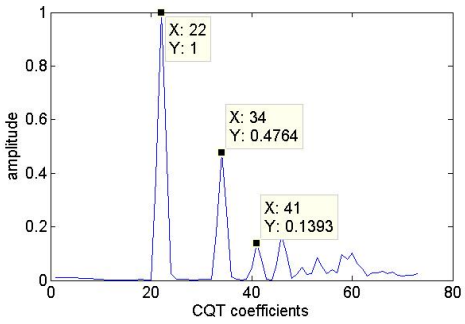
Για να βρεθεί το τέλος της νότας, η αναζήτηση ξεκινάει από το τέλος του αρχείου και υπολογίζεται η ενέργεια κάθε 3.5msec προχωρώντας προς τα πίσω. Όταν βρεθεί το πρώτο χρονικό διάστημα (t1, t2) όπου η ενέργεια είναι μεγαλύτερη του 10^{-10} αποθηκεύεται στη μεταβλητή $end\_$ η τιμή $t1 - 2 \cdot wlen$ όπου $wlen$ το μήκος του frame σε δευτερόλεπτα. Εντοπίζεται δηλαδή ένα χρονικό διάστημα στο οποίο η ενέργεια του σήματος είναι πολύ χαμηλή και πηγαίνουμε λίγο πιο πίσω όπου το σήμα είναι λίγο δυνατότερο. Η επιλογή της χρονικής στιγμής med όπου αρχίζει το μεσαίο frame δίνεται από την εξίσωση $med = \frac{2}{5}(end\_ - start)$.

Έτσι, η συνάρτηση cqt(x,sparKernel) θα εκτελεστεί τρεις φορές για ένα αρχείο .wav και σε κάθε εκτέλεση αλλάζει μόνο το όρισμα x. Αν s το σήμα που προκύπτει από το αρχείο .wav, στην πρώτη εκτέλεση της συνάρτησης το όρισμα x θα είναι το $s(start : start + wlen)$, στη δεύτερη το $s(med : med + wlen)$ και στην τρίτη το $s(end\_ : end\_ + wlen)$. Η συνάρτηση cqt επιστρέφει ένα διάνυσμα μεγέθους 1×73. Κρατάμε την απόλυτη τιμή του διανύσματος. Για μια νότα με συχνότητα f_k θα υπάρχει ένα τοπικό μέγιστο που αντιστοιχεί στη βασική

συχνότητα στη θέση $k_0 = 12 \cdot \log_2(\frac{f_k}{f_{min}})$. Για παράδειγμα για την νότα $B3$, την 39η νότα του πιάνου, με συχνότητα $f_k = 246.942$ θα υπάρχει τοπικό μέγιστο στη θέση $k_0 = 24$. Γενικότερα, για μια νότα με κλειδί key θα υπάρχει ένα τοπικό μέγιστο στο διάνυσμα των συντελεστών CQT στη θέση $k_0 = key - 15$. Θα υπάρχουν επίσης τοπικά μέγιστα που αντιστοιχούν στις αρμονικές συχνότητες της νότας. Το τοπικό μέγιστο για την πρώτη αρμονική της νότας με συχνότητα f_k θα βρίσκεται στη θέση $k_1 = 12 \cdot \log_2(\frac{2f_k}{f_{min}}) = 12 \cdot \log_2(\frac{f_k}{f_{min}}) + 12 \cdot \log_2 2 = k_0 + 12$, για τη δεύτερη αρμονική στη θέση $k_2 = 12 \cdot \log_2(\frac{3f_k}{f_{min}}) = k_0 + 19$ και ομοίως οι υπόλοιπες.



(α′) attack μέρος της A3



(β′) sustain μέρος της A3

Σχήμα 4.3: Απεικόνιση των συντελεστών CQT του attack και sustain τμήματος την νότας A3 (κλειδί 37)

Οι συντελεστές του διανύσματος που επιστρέφεται από τη συνάρτηση cqt(x,sparKernel) διαιρούνται με τη μέγιστη τιμή του διανύσματος έτσι ώστε όλα τα διανύσματα να είναι κανονικοποιημένα ως προς τη μονάδα. Με αυτόν τον τρόπο το amplitude δεν επηρεάζει την αναγνώριση μιας νότας. Στο σχήμα 4.3 φαίνεται ένα παράδειγμα συντελεστών CQT για το attack και το sustain μέρος μιας νότας. Οι συντελεστές CQT για όλα τα δεδομένα εκπαίδευσης αποθηκεύονται σε ένα αρχείο δεδομένων της Matlab. Το μέγεθος του πίνακα των δεδομένων εκπαίδευσης θα είναι δηλαδή 657×73 (219 αρχεία .wav $\times$ 3 frames για κάθε αρχείο $= 657$).

4.2.3 Αναγνώριση άγνωστης νότας

Στην ενότητα αυτή περιγράφεται ο αλγόριθμος που υλοποιήθηκε για την αναγνώριση μονοφωνικής μουσικής πιάνου. Η επεξεργασία του σήματος s γίνεται χρησιμοποιώντας παράθυρα μήκους wlen = 80msec και ολίσθηση παραθύρου κάθε wlag = 40msec. Γίνεται *zero-padding* στο σήμα ώστε να χωράει ακέραιο αριθμό παραθύρων. Υπολογίζεται ο πίνακας sparkernel με τη συνάρτηση sparsekernel. Για κάθε frame k υπολογίζεται η απόλυτη τιμή του Constant Q Transform με τη συνάρτηση cqt που περιγράφηκε παραπάνω. Ορίσματα της cqt είναι το $s_k = s((k-1) \cdot wlen + 1/F_s : k \cdot wlen)$ και ο πίνακας sparkernel. Οι συντελεστές που υπολογίζονται κανονικοποιούνται ως προς τη μονάδα και αποθηκεύονται στο διάνυσμα NC μεγέθους $1 \times N$ με N=73. Στη συνέχεια, πρέπει να βρεθεί μια καταχώρηση στον πίνακα των δεδομένων εκπαίδευσης A μεγέθους $M \times N$ με M = 657 που να ταιριάζει καλύτερα με το διάνυσμα NC. Επιλέγεται η γραμμή m του πίνακα A που ελαχιστοποιεί τη σχέση 4.7, το μέσο τετραγωνικό σφάλμα δηλαδή:

$$d = \sqrt{\sum_{n=1}^{N} |NC_n - A_{m,n}|^2} \tag{4.7}$$

Στον *Ψευδοκώδικα 2* φαίνεται αναλυτικότερα ο παραπάνω αλγόριθμος.

Ο πίνακας key0, όπως προκύπτει από τον *Ψευδοκώδικα 2*, στη θέση k περιέχει τον αριθμό της γραμμής του πίνακα A στην οποία βρίσκεται η καταχώρηση που περιγράφει καλύτερα το frame κ. Οι δείκτες αυτοί προς τον πίνακα A πρέπει να αντικατασταθούν με αριθμούς που ανταποκρίνονται στα κλειδιά του πιάνου. Δεδομένου ότι είναι γνωστό σε ποια νότα αντιστοιχεί κάθε γραμμή

του A, υλοποιήθηκε μια συνάρτηση που απλά αντικαθιστά τους δείκτες προς τις γραμμές του A με το κλειδί της αντίστοιχης νότας.

Ψευδοκώδικας 2 Ο αλγόριθμος αυτός αντιστοιχίζει σε κάθε frame του σήματος x μια καταχώρηση του A και επιστρέφει το αποτέλεσμα στον key0

Define $A \leftarrow$ TrainData ;
[x,F_s]= wavread(filename);
$wlen = 0.08 \cdot F_s$;
$wlag = 0.04 \cdot F_s$;
ZeroPadding(x);
sparKernel = sparseKerrnel(65.4064,4186.01,12,F_s,0.005);
SET K←(maximum number of frames fit in x);
SET key0← vector $1 \times K$;
for k=1 to K
  SET $x_k = s((k-1) \cdot wlen : k \cdot wlen)$;
  $C = |cqt(x_k, sparKernel)|$;
  $m = max(C)$;
  $NC = C/m$;
  SET min = ∞;
  for m=1 to (number of training data)
    d = $\sqrt{\sum_{n=1}^{N} |NC(n) - A_m(n)|^2}$;
    if $d < min$
      min=d;
      PosibleKeyPos = m;
    end if;
  end for;
  $key0_k$ = PosibleKeyPos;
end for;
return key0;

4.3 Onset Detector

Για τον καθορισμό της διάρκειας μιας νότας υλοποιήθηκε ένας Onset Detector. Ο Onset Detector εντοπίζει τη χρονική στιγμή στην οποία αρχίζει μια

νότα. Για να γίνει αυτό, το σήμα χωρίζεται σε επτά ζώνες συχνοτήτων, σε κάθε ζώνη υπολογίζεται η περιβάλλουσα του σήματος και εξάγονται τα τοπικά μέγιστα. Τα αποτελέσματα από τις επτά ζώνες συχνοτήτων συνδυάζονται ώστε να προκύψει το τελικό αποτέλεσμα. Αναλυτικότερα ο αλγόριθμος έχει ως εξής:

Βήμα 1. Δημιουργήθηκαν επτά ελλειπτικά φίλτρα έκτης τάξης με τη συνάρτηση ellip της Matlab. Οι ζώνες συχνοτήτων των φίλτρων τοποθετούνται σε λογαριθμική κλίμακα. Το πρώτο φίλτρο είναι χαμηλοπερατό με συχνότητα αποκοπής 127Hz, τα πέντε επόμενα ζωνοπερατά με όρια [127 254], [254 508], [508 1016], [1016 2032], [2032 4064] αντίστοιχα και το τελευταίο υψηπερατό με κάτω όριο 4064Hz.

Βήμα 2. Το αρχικό σήμα x φιλτράρεται με κάθε ένα από αυτά τα επτά φίλτρα και στην έξοδο τους προκύπτουν τα σήματα x_1, x_2... x_7.

Βήμα 3. Κάθε σήμα στην έξοδο ενός φίλτρου περνάει από έναν ανορθωτή. Θέτεται δηλαδή $x_i = |x_i|$, $\forall i = 1, 2...7$.

Βήμα 4. Κάθε σήμα x_i περνάει από ένα χαμηλοπερατό φίλτρο. Χρησιμοποιείται φίλτρο raised cosine μεγέθους 87.5 msec. Έτσι προκύπτουν οι περιβάλλουσες των x_i, e_i.

Βήμα 5. Στη συνέχεια τα τοπικά μέγιστα των e_i μπορούν να βρεθούν υπολογίζοντας την πρώτη παράγωγο. Τα τοπικά μέγιστα της παραγώγου αντιστοιχούν στα σημεία όπου τα e_i εμφανίζουν μέγιστη κλίση. Τα σημεία αυτά μπορούν να θεωρηθούν ότι σηματοδοτούν την αρχή μιας νότας. Ωστόσο, η περιβάλλουσα του αρχικού τμήματος μιας νότας δεν αυξάνεται μονότονα αλλά εμφανίζει αρκετά τοπικά μέγιστα σε σημεία γειτονικά του σημείου που αντιστοιχεί στην αρχή της νότας. Επιπλέον, σε νότες χαμηλής έντασης η χρονική στιγμή στην οποία αρχίζουν και η χρονική στιγμή στην οποία εμφανίζεται μέγιστη τιμή στην περιβάλλουσά τους μπορεί να απέχουν σημαντικά μεταξύ τους. Για αυτούς τους λόγους η συνάρτηση των διαφορών πιθανόν να αποτυγχάνει να εντοπίσει με ακρίβεια την αρχή μιας νότας. Καλύτερα αποτελέσματα μπορούν να επιτευχθούν αν χρησιμοποιηθεί μια συνάρτηση που εκφράζει τη διαφορά μεταξύ δύο διαδοχικών χρονικών στιγμών t_1, t_2 σε σχέση με την απόλυτη τιμή της περιβάλλουσας τη χρονική στιγμή t_2. Η προσέγγιση αυτή αναπτύχθηκε από τον Anssi Klapuri[17]. Έτσι υπολογίζεται η συνάρτηση $l_i = diff(e_i)/e_i$ η οποία ισοδύναμεί με την παράγωγο του λογαρίθμου της

e_i. Ισχύει δηλαδή $l_i = diff(\log e_i)$. Οι αρνητικές ή μηδενικές τιμές της e_i πρέπει να αντικατασταθούν κατάλληλα ώστε η l_i να έχει πραγματικές τιμές. Έτσι γίνεται η εξής αντικατάσταση πριν τον υπολογισμό των l_i: $e_i(find(e_i < 0.01)) = 0.01$.

Βήμα 6. Γίνεται υποδειγματοληψία των l_i με συχνότητα 65Hz ώστε να μειωθούν τα δεδομένα και να γίνουν εντονότερες οι διαφορές μεταξύ γειτονικών σημείων.

Βήμα 7. Υπολογίζεται η παράγωγος των l_i $d_i = diff(l_i)$.

Βήμα 8. Εφαρμόζεται threshold T στα d_i έτσι ώστε να μηδενιστούν οι πολύ χαμηλές τιμές οι οποίες δεν αποτελούν χρήσιμη πληροφορία για τον εντοπισμό της αρχής μιας νότας. Ορίζεται T $= 0.25$ και $d_i(find(d_i < T)) = 0$.

Βήμα 9. Γίνεται υπερδειγματοληψία των d_i με συχνότητα F_s, τη συχνότητα δειγματοληψίας του σήματος εισόδου x δηλαδή. Έτσι τα d_i έχουν τώρα μέγεθος ίσο με το μέγεθος του x. Τα d_i αθροίζονται και οι τιμές τους αποθηκεύονται στο διάνυσμα d.

Βήμα 10. Υπολογίζονται τα τοπικά μέγιστα του διανύσματος d και αποθηκεύονται με χρονική σειρά στο διάνυσμα *onsets*. Το διάνυσμα *onsets* δηλαδή είναι ένα διάνυσμα ίσου μεγέθους με το d. Αν T το σύνολο των χρονικών στιγμών στις οποίες εντοπίστηκε τοπικό μέγιστο στο διάνυσμα d το διάνυσμα *onsets* περιγράφεται από τη σχέη:

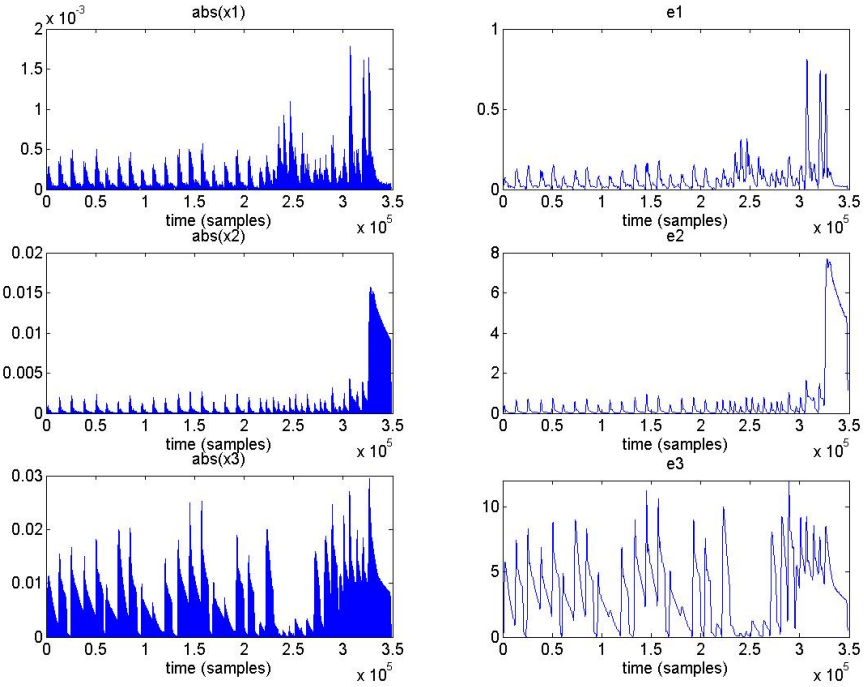
$$onsets(t) = \begin{cases} d(t) & \text{if } t \in T \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

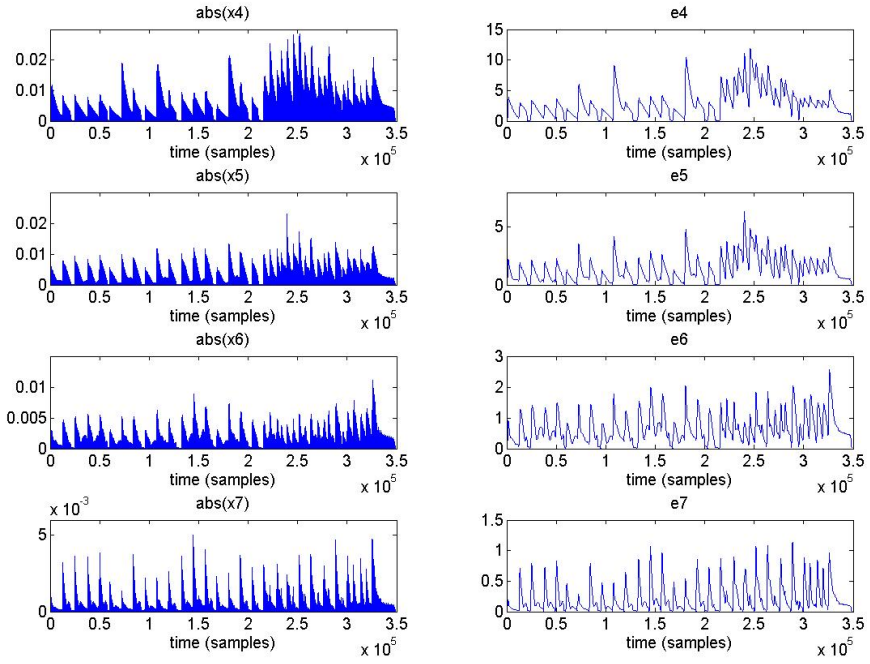
Βήμα 11. Στη συνέχεια εφαρμόζονται παράθυρα μήκους 80msec στο διάνυσμα *onsets*. Όλες οι τιμές του *onsets* εντός του παραθύρου αθροίζονται και το αποτέλεσμα της άθροισης τοποθετείται στη θέση k του *onsets* με $k = n + wlen/2$ όπου n η θέση του πρώτου δείγματος ενός παραθύρου μέσα στον *onsets* και $wlen$ το μήκος του παραθύρου σε δείγματα. Όλες οι άλλες τιμές του *onsets* εντός του παραθύρου μηδενίζονται. Θεωρείται δηλαδή ελάχιστη διάρκεια νότας 80msec, επομένως επιτρέπεται μόνο μία μη μηδενική τιμή στον *onsets* σε διάστημα 80msec.

Βήμα 12. Τέλος εφαρμόζεται threshold T στον *onsets* για να διαγραφούν οι μικρότερες τιμές. Ορίζεται T=1.5 και $onsets(find(onsets < T)) = 0$.

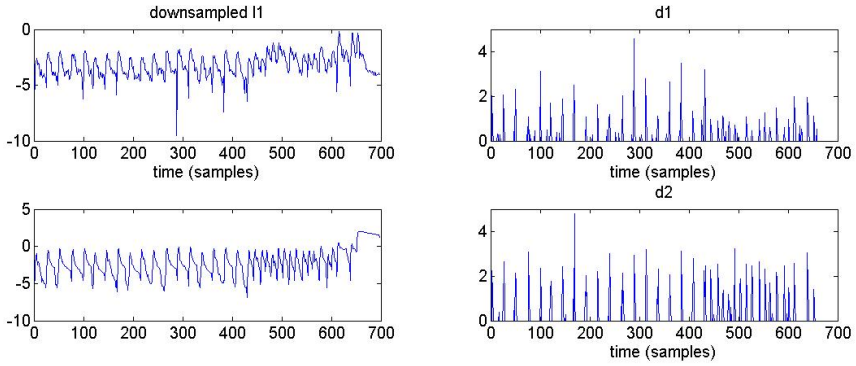
Κάθε μια από τις μη μηδενικές τιμές του *onsets* αντιστοιχεί στην εμφάνιση μιας νότας.

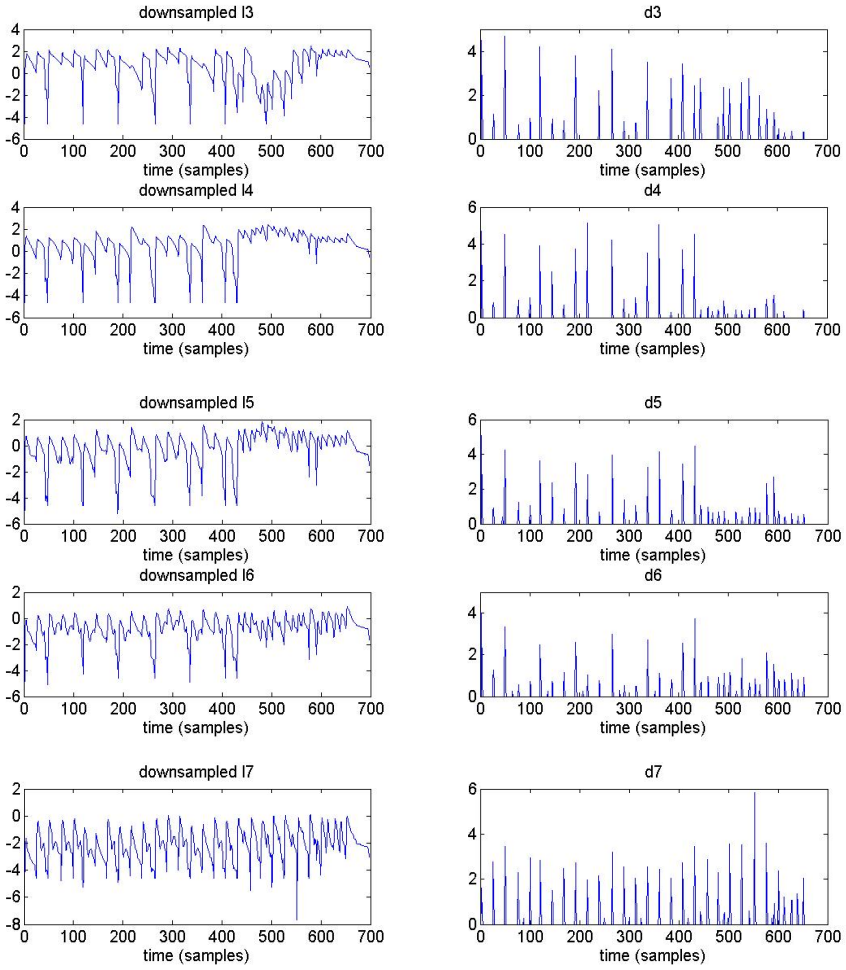
Στα παρακάτω σχήματα φαίνεται ένα παράδειγμα της εφαρμογής αυτού του αλγορίθμου σε ένα σήμα x. Στο σχήμα 4.4 φαίνεται η απόλυτη τιμή των σημάτων που προκύπτουν στις εξόδους των επτά φίλτρων και οι αντίστοιχες περιβάλλουσές τους. Στο σχήμα 4.5 φαίνονται τα υποδειγματοληπτημένα l_i όπως προκύπτουν από το βήμα 6 του αλγορίθμου καθώς και η αντίστοιχη παράγωγος για κάθε l_i όπως προκύπτει μετά την εφαρμογή του threshold όπως περιγράφεται στο βήμα 8. Τέλος στο σχήμα 4.6 φαίνεται το αρχικό σήμα x σε χρονική αντιστοιχία με τα *onsets* που προέκυψαν στο τέλος του αλγορίθμου.



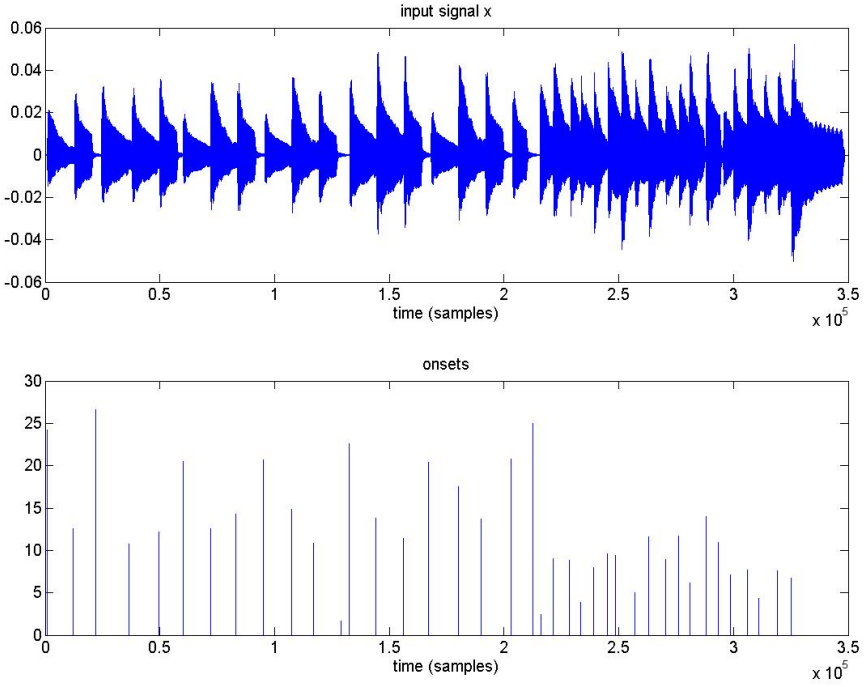


Σχήμα 4.4: Αριστερά φαίνονται τα σήματα x_i, δηλαδή η απόλυτη τιμή των σημάτων στις εξόδους των επτά φίλτρων και αριστερά οι αντίστοιχες περιβάλλουσες e_i





Σχήμα 4.5: Αριστερά τα υποδειγματοληπτημένα l_i όπως προκύπτουν από το βήμα 6 του αλγορίθμου και δεξιά τα αντίστοιχα τοπικά μέγιστα όπως προκύπτουν από το βήμα 8



Σχήμα 4.6: Στο πάνω γράφημα το αρχικό σήμα x και στο κάτω τα αντίστοιχα onsets που υπολογίστηκαν

Η αρχή μιας νότας και στους δύο αλγορίθμους για την αναγνώριση μονοφωνικού ήχου που αναπτύχθηκαν καθορίζεται από τον onset detector. Όταν σε ένα χρονικό διάστημα μεταξύ δύο μη μηδενικών σημείων στον πίνακα onset εμφανίζονται περισσότερες από μια νότες, θεωρείται σωστή μόνο αυτή με τη μεγαλύτερη διάρκεια. Αυτό συμβαίνει κάποιες φορές όταν το attack μέρος μιας νότας που είναι πιο θορυβώδες αναγνωρίζεται εσφαλμένα ως διαφορετική νότα. Σε αυτήν την περίπτωση, σε ένα χρονικό διάστημα όπου υπάρχει μόνο μια νότα σύμφωνα με τον onset detector, εμφανίζονται δύο νότες με την πρώτη να έχει πολύ μικρή διάρκεια. Αυτή η νότα θα διαγραφεί. Συγκεκριμένα, αν αντιστοιχούν στην πρώτη νότα ν_1 οι εγγραφές $k_1, k_2, ...k_m$ του πίνακα $key0$, ισχύει δηλαδή $key0(k_1) = \nu_1, key0(k_2) = \nu_1, ...key0(k_m) = \nu_1$, τότε θέτω $key0(k_1) = \nu_2, key0(k_2) = \nu_2, ...key0(k_m) = \nu_2$ με ν_2 τη δεύτερη νότα. Με αυτόν τον τρόπο διορθώνονται πιθανά λάθη που μπορεί να προκύψουν από το attack μέρος μιας νότας.

Επιπλέον, όταν κατά τη διάρκεια μίας νότας εμφανίζονται δύο ή περισσότερα onsets, τότε πιθανόν να πρόκειται για επαναλαμβανόμενες νότες και όχι για μία ενιαία. Αν κάθε μία από τις νότες που δημιουργούνται μετά το "σπάσιμο" της αρχικής σύμφωνα με τα onsets έχει χρονική διάρκεια μεγαλύτερη των 160msec τότε πρόκειται για επαναλαμβανόμενες νότες και η αρχική νότα διαχωρίζεται. Αντίθετα, αν για δύο ίδιες νότες ισχύει ότι η χρονική στιγμή στην οποία εντοπίζεται το τέλος της πρώτης απέχει από τη χρονική στιγμή στην οποία εντοπίζεται η αρχή της δεύτερης λιγότερο από 120msec και δεν έχει εντοπιστεί onset για τη δεύτερη, οι δύο νότες πρέπει να ενωθούν σε μία.

Όπως φαίνεται από τα παραπάνω, ο onset detector χρησιμοποιείται και για τη διόρθωση πιθανών λαθών των αλγορίθμων. Τέλος, ο onset detector χρησιμοποιείται για το "σπάσιμο" του αρχικού σήματος σε μικρότερα. Δηλαδή οι δύο αλγόριθμοι δεν παίρνουν σαν είσοδο ολόκληρο το αρχικό σήμα αλλά τμήματά του όπως αυτά ορίζονται από τα onsets που εντοπίστηκαν. Με αυτό τον τρόπο διασφαλίζεται ότι μία νότα δεν θα αλλάξει μέσα στο χρονικό διάστημα ενός frame.

Κεφάλαιο 5

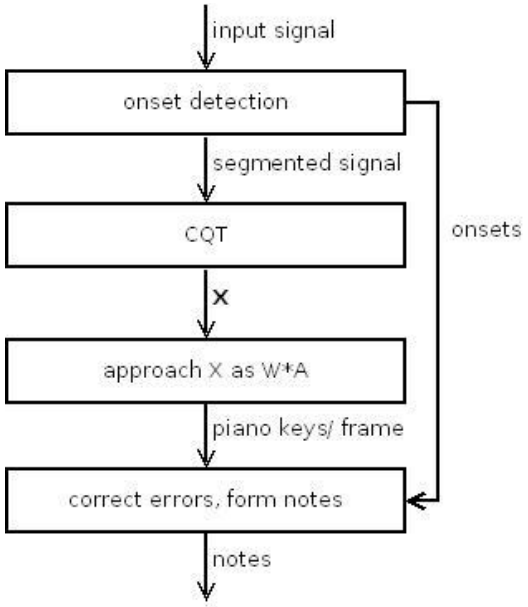
Υλοποίηση αλγορίθμου μετατροπής πολυφωνικής μουσικής πιάνου σε συμβολική αναπαράσταση

Στην ενότητα αυτή περιγράφεται ο αλγόριθμος που υλοποιήθηκε για την αναγνώριση πολυφωνικής μουσικής πιάνου. Ο αλγόριθμός αυτός στηρίζεται σε τεχνικές που αναφέρθηκαν στην ενότητα 3.4. Συγκεκριμένα, η απόλυτη τιμή του φάσματος ενός άγνωστου δείγματος X πρέπει να αναλυθεί σε κάποιες γνωστές συνιστώσες. Έτσι αν $\mathbf{X} \in R^{\geq 0,1\times N}$ και $\mathbf{A}$ ο πίνακας των δεδομένων εκπαίδευσης με $\mathbf{A} \in R^{\geq 0,M\times N}$, πρέπει να βρεθεί πίνακας $\mathbf{W} \in R^{\geq 0,1\times M}$ έτσι ώστε το γινόμενο των πινάκων $\mathbf{A}$, $\mathbf{W}$ να προσεγγίζει όσο το δυνατόν καλύτερα το $\mathbf{X}$.

Η δομή του αλγορίθμου φαίνεται στο σχήμα 5.1. Αρχικά χρησιμοποιείται ο onset detector που υλοποιήθηκε στην ενότητα 4.3 για να εντοπιστούν όλες οι χρονικές στιγμές στις οποίες εκτιμάται ότι αρχίζει μια νότα τα *onsets* δηλαδή. Τα σημεία αυτά θα χρησιμοποιηθούν για να χωριστεί το αρχικό σήμα εισόδου σε μικρότερα τμήματα. Η διάσπαση του αρχικού σήματος χρησιμοποιώντας τα *onsets* εξασφαλίζει ότι σε χρονικό διάστημα ενός frame δεν θα εμφανιστεί κάποια καινούρια νότα που δεν υπήρχε από την αρχή του frame.

Στο δεύτερο βήμα του αλγορίθμου, κάθε ένα από τα τμήματα του αρχικού σήματος χωρίζεται σε frames μήκους 80msec με ολίσθηση κάθε 40msec. Για κάθε frame υπολογίζεται ο Constant Q Transform. Ο CQT ενός frame αποτελεί το $\mathbf{X}$ μεγέθους 1×73 το οποίο πρέπει να προσεγγιστεί με τους πίνακες $\mathbf{A}$ και $\mathbf{W}$.

Στο τρίτο βήμα, εντοπίζονται οι νότες που ακούγονται στη διάρκεια ενός συγκεκριμένου frame. Ουσιαστικά δηλαδή υπολογίζονται κατάλληλες τιμές για τον **W** έτσι ώστε το γινόμενο **W**·**A** να προσεγγίζει βέλτιστα το **X**. Ο πίνακας **A** είναι ο ίδιος πίνακας δεδομένων εκπαίδευσης που υλοποιήθηκε για τον αλγόριθμο αναγνώρισης μονοφωνικής μουσικής που περιγράφεται στην ενότητα 4.2. Ο πίνακας αυτός έχει μέγεθος 657×73 όπως αναφέρεται στην ενότητα 4.2.2. Ο πίνακας **W** έχει μέγεθος 1×657 και αντιστοιχίζει ένα βάρος σε κάθε εγγραφή του πίνακα **A**. Ο πίνακας **W** ουσιαστικά εκφράζει αν μια νότα υπάρχει ή όχι στο συγκεκριμένο frame για το οποίο υπολογίστηκε ο **X** και αν υπάρχει ποιο πρέπει να είναι το βέλτιστο amplitude ώστε να προσεγγίζεται καλύτερα ο **X**. Έτσι, οι περισσότερες εγγραφές του **W** θα έχουν μηδενική τιμή.



Σχήμα 5.1: Σχηματική αναπαράσταση της δομής του αλγορίθμου.

Από το τρίτο βήμα προκύπτει μια εικόνα για το ποιες νότες ακούγονται σε κάθε frame. Στο 4ο βήμα η πληροφορία αυτή σε συνδυασμό με τα *onsets* που υπολογίστηκαν στο 1ο βήμα χρησιμοποιείται για να εξαχθούν νότες συγκεκριμένης χρονικής διάρκειας. Επιπλέον διορθώνονται τα αποτελέσματα του τρίτου βήματος όταν αυτό είναι απαραίτητο. Συγκεκριμένα κάποιες νότες μεγάλης χρονικής διάρκειας πιθανόν να εμφανίζονται με κάποια 'κενά' στην έξ-

οδο του 3ου βήματος. Δηλαδή μια νότα μπορεί να χάνεται σε κάποια frame και να εμφανίζεται σε κάποια επόμενα. Αυτό μπορεί να συμβεί σε νότες μεγάλης χρονικής διάρκειας καθώς όσο περνάει ο χρόνος η έντασή τους εξασθενεί και η εμφάνιση κάποιας άλλης νότας που ακούγεται ταυτόχρονα με αυτήν μπορεί να προκαλέσει την εξαφάνισή της σε κάποια frames. Αυτές οι διακεκομμένες νότες πρέπει να ενωθούν σε μία σε περίπτωση που δεν αποτελούν όντως ξεχωριστές νότες. Επιπλέον εφαρμόζεται ένας αλγόριθμος για τον εντοπισμό επαναλαμβανόμενων νοτών. Τέλος, νότες που η παρουσία τους δηλώνεται σε ένα μόνο frame δηλαδή νότες διάρκειας μικρότερης των 80msec διαγράφονται.

Στην ενότητα 5.1 περιγράφεται αναλυτικότερα η διαδικασία εύρεσης των διαφορετικών νοτών μέσα σε ένα frame ενώ στην ενότητα 5.2 η διαδικασία εξαγωγής νοτών με καθορισμένη χρονική διάρκεια.

5.1 Εντοπισμός νοτών σε κάθε frame

Αναπτύχθηκε ένας αλγόριθμος για τον υπολογισμό του πίνακα $\mathbf{W}$ από τους $\mathbf{A}$, $\mathbf{X}$. Ο αλγόριθμος αυτός εκτελεί το πολύ δέκα επαναλήψεις (δέκα είναι ο μέγιστος αριθμός νοτών που μπορούν να παίζονται ταυτόχρονα από ένα πιάνο) και σε κάθε επανάληψη εντοπίζεται η ύπαρξη μιας καινούριας νότας στον $\mathbf{X}$. Θεωρώντας ως εγγραφή i του πίνακα $\mathbf{A}$ την γραμμή i και συμβολίζοντας την ως $A_{i,:}$ και w_i την τιμή του πίνακα $\mathbf{W}$ στη θέση i, ορίζεται πίνακας P με:

$$P = \sum_{\forall i, w_i \neq 0} w_i \cdot A_{i,:} \tag{5.1}$$

Ο P αρχικά είναι μηδενικός και ενημερώνεται σε κάθε επανάληψη του αλγορίθμου σύμφωνα με την εξίσωση:

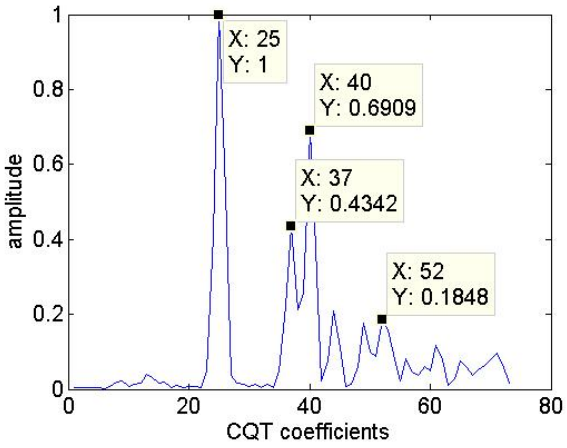
$$P^{(n)} = P^{(n-1)} + w_i \cdot A_{i,:} \tag{5.2}$$

όπου το n συμβολίζει τη n-οστή επανάληψη και τα w_i, $A_{i,:}$ επιλέγονται έτσι ώστε να ικανοποιούνται οι σχέσεις 5.1 και 5.2:

$$||X - (P^{(n-1)} + w_i \cdot A_{i,:})|| < ||X - (P^{(n-1)} + w_j \cdot A_{j,:})||, \forall j \neq i \tag{5.3}$$

$$||X - P^{(n-1)}|| - ||X - P^{(n)}|| > 0.06 \tag{5.4}$$

Η σχέση 5.3 εκφράζει το κατά πόσο η συγκεκριμένη επιλογή της εγγραφής i του πίνακα $\mathbf{A}$ είναι βέλτιστη για την n-οστή επανάληψη του αλγορίθμου, ενώ η σχέση 5.4 το κατά πόσο η προσθήκη της καινούριας αυτής νότας βελτιώνει το αποτέλεσμα όπως αυτό προκύπτει από τις προηγούμενες επαναλήψεις του αλγορίθμου. Περιορισμοί θέτονται επίσης και για την ελάχιστη μη μηδενική τιμή του πίνακα $\mathbf{W}$. Αν σε κάποια επανάληψη του αλγορίθμου δε βρεθεί κάποιο $A_{i,:}$ για το οποίο να μπορεί να υπολογιστεί w_i έτσι ώστε να ικανοποιούνται οι δύο παραπάνω σχέσεις ο αλγόριθμος τερματίζει πριν συμπληρωθούν δέκα επαναλήψεις.



Σχήμα 5.2: Constant Q Transform για ένα frame όπου ακούγονται οι νότες C4(40) και D#5(55)

Ουσιαστικά σε κάθε επανάληψη του αλγορίθμου πρέπει να γίνεται αναζήτηση σε ολόκληρο τον πίνακα $\mathbf{A}$ ώστε να ενημερωθεί κατάλληλα ο P. Για να αναγνωριστεί δηλαδή ένα μουσικό σήμα μεγέθους K frames χρειάζεται να γίνουν $O(10 \cdot K \cdot M)$ πράξεις με M τον αριθμό των γραμμών του $\mathbf{A}$. Επειδή η υπολογιστική πολυπλοκότητα του αλγορίθμου είναι πολύ μεγάλη, επιλέχθηκε να μη γίνεται αναζήτηση σε ολόκληρο τον πίνακα $\mathbf{A}$ αλλά μόνο σε συγκεκριμένες εγγραφές του που αντιστοιχούν σε νότες που πιθανόν να υπάρχουν μέσα στον $\mathbf{X}$. Συγκεκριμένα, υπολογίζονται όλα τα τοπικά μέγιστα του $\mathbf{X}$. Όπως έχει αναφερθεί, ένα τοπικό μέγιστο στον Constant Q Transform στη θέση k μπορεί να δηλώνει την ύπαρξης μιας νότας με κλειδί $k+15$ ή την παρουσία της πρώτης αρμονικής της νότας με κλειδί $k+15-12$ ή της δεύτερης αρμονικής της νότας με

κλειδί $k+15-12-7$ κτλ. Κάποια τοπικά μέγιστα δηλαδή αντιστοιχούν σε νότες και κάποια σε αρμονικές νοτών. Θεωρώντας ότι όλα τα τοπικά μέγιστα είναι πιθανόν να αντιστοιχούν σε νότες και αδύνατον να υπάρχει κάποια νότα στο $\mathbf{X}$ για την οποία δεν έχει βρεθεί τοπικό μέγιστο στην αντίστοιχη θέση μπορεί ο χώρος αναζήτησης στον πίνακα $\mathbf{A}$ να περιοριστεί μόνο στις εγγραφές που αντιστοιχούν σε νότες για τις οποίες έχουν βρεθεί τοπικά μέγιστα. Έτσι περιορίζεται σημαντικά η χρονική πολυπλοκότητα του αλγορίθμου. Για παράδειγμα, στο σχήμα 5.1 απεικονίζεται ο Constant Q Transform σε ένα frame όπου ακούγονται δύο νότες με κλειδιά 40 και 55 αντίστοιχα. Εμφανίζονται 11 τοπικά μέγιστα από τα οποία μόνο δύο, αυτά στις θέσεις 25 και 40, αντιστοιχούν σε νότες ενώ τα υπόλοιπα σε αρμονικές. Έτσι υπάρχουν μόνο 11 πιθανές νότες. Δεδομένου ότι υπάρχουν εννιά εγγραφές στον πίνακα $\mathbf{A}$ για μία νότα ο χώρος αναζήτησης περιορίζεται σε 99 εγγραφές αντί για 657 που ήταν αρχικά. Η διαδικασία αυτή περιγράφεται αναλυτικότερα στους *Αλγορίθμους 5.1* και *5.2*.

Αλγόριθμος 5.1. *Υπολογισμός του πίνακα X για ένα frame καθώς και των εγγραφών του πίνακα A που είναι πιθανόν να αντιστοιχούν σε νότες του X. Οι εγγραφές αυτές αποθηκεύονται στον πίνακα A΄.*

Βήμα 1-Υπολογισμος του X. *Υπολογίζεται ο Constant Q Transform χρησιμοποιώντας τη συνάρτηση cqt(x,sparKernel) Ο πίνακας X προκύπτει παίρνοντας την απόλυτη τιμή του αποτελέσματος της συνάρτησης cqt και κανονικοποιώντας έτσι ώστε η μέγιστη τιμή του X να είναι ίση με τη μονάδα.*

Βήμα 2-Εντοπισμών τοπικών μεγίστων στον X. *Εντοπίζονται όλα τα τοπικά μέγιστα του X μεγαλύτερα από ένα threshold T=0.09. Τα τοπικά μέγιστα αυτά θα χρησιμοποιηθούν για να καθοριστούν οι πιθανές νότες και επομένως ο χώρος αναζήτησης μέσα στον πίνακα A. Οι θέσεις στις οποίες εντοπίζονται τα τοπικά μέγιστα αποθηκεύονται στον πίνακα MaxInd.*

Βήμα 3-Δημιουργία του A΄. *Θέτοντας MaxInd = MaxInd+15, οι τιμές του MaxInd αντιστοιχούν σε κλειδιά νοτών που είναι πιθανόν να υπάρχουν στο συγκεκριμένο frame. Εντοπίζονται οι εγγραφές του πίνακα A για τις συγκεκριμένες νότες και αποθηκεύονται στον πίνακα A΄.*

Βήμα 4-Εκτέλεση του επαναληπτικού Αλγορίθμου 5.2. *Στη συνέχεια πρέπει να βρεθούν ποιες εγγραφές του A΄ αντιστοιχούν σε*

νότες του X. Αυτό γίνεται με τον επαναληπτικό αλγόριθμο 5.2 σε κάθε επανάληψη του οποίου εντοπίζεται και μια καινούρια νότα. Ο μέγιστος αριθμός επαναλήψεων του αλγορίθμου ορίζεται ως MaxPol = max(10,length(MaxInd)) όπου η τιμή length(MaxInd) αντιστοιχεί στον αριθμό των νοτών που θεωρείται πιθανόν να υπάρχουν σύμφωνα με το **βήμα 3**. *Ο αλγόριθμος 5.2 θα χρησιμοποιήσει όλα τα δεδομένα που υπολογίστηκαν σε αυτόν τον αλγόριθμο, δηλαδή τον πίνακα X, τον MaxInd και τον A′. Επιπλέον γίνεται αρχικοποίηση κάποιων μεταβλητών που θα χρησιμοποιηθούν στον 5.2. Οι μεταβλητές αυτές είναι οι $min = \infty$ και $PrevMin = \infty$. Η μεταβλητή min θα χρησιμοποιηθεί για την επαλήθευση ή όχι της σχέσης 5.3 και η PrevMin για τη σχέση 5.4.*

Αλγόριθμος 5.2. *Επαναληπτικός αλγόριθμος σε κάθε επανάληψη του οποίου εντοπίζεται και μια καινούρια νότα στον X. Τα κλειδιά των νοτών που εντοπίζονται αποθηκεύονται στον πίνακα keys μεγέθους $1 \times MaxPol$. Παρακάτω περιγράφεται η διαδικασία εντοπισμού μιας νότας κατά την n-οστή επανάληψη του αλγορίθμου.*

Βήμα 1-Αρχικοποιήσεις. *Στο βήμα αυτό γίνονται αρχικοποιήσεις μεταβλητών που θα χρησιμοποιηθούν στον αλγόριθμο. Δημιουργείται ο πίνακας P μεγέθους 1×73 ο οποίος αρχικά είναι μηδενικός και ενημερώνεται σε κάθε βήμα του αλγορίθμου σύμφωνα με τη σχέση 5.2. Ορίζονται επιπλέον και οι πίνακες StoreKeys μεγέθους $MaxPol \times 73$ και StoreW μεγέθους $MaxPol \times 1$. Οι δύο πίνακες αρχικά είναι μηδενικοί. Η γραμμή n του πίνακα StoreKeys θα περιέχει την εγγραφή του πίνακα A′ που αντιστοιχεί στη νότα που βρέθηκε στην n επανάληψη του αλγορίθμου και η γραμμή n του StoreW το αντίστοιχο βάρος που υπολογίστηκε.*

Βήμα 2-Εύρεση του βέλτιστου w_i για την εγγραφή $A'_{i,:}$. *Σε κάθε επανάληψη του αλγορίθμου πρέπει να υπολογιστεί ένα βάρος w για κάθε εγγραφή του A′. Για να γίνει αυτό, για την εγγραφή $A'_{i,:}$ υπολογίζεται ένα βάρος w_i χρησιμοποιώντας έναν αλγόριθμο LMS ο οποίος υπολογίζει την βέλτιστη τιμή του w_i για την οποία ελαχιστοποιείται η ποσότητα $e = ||X - (P^{(n-1)} + w_i \cdot A'_{i,:})||$. Η συνάρτηση LMS παίρνει τρία ορίσματα, το X, έναν πίνακα CopyStoreKeys μεγέθους $n \times 73$ και έναν πίνακα CopyStoreW μεγέθους $n \times 1$. Οι $n-1$ πρώτες γραμμές του CopyStoreKeys είναι οι $n-1$ πρώτες*

γραμμές του StoreKeys ενώ η τελευταία είναι η εγγραφή i του A′. Ο CopyStoreKeys δηλαδή έχει όλες τις εγγραφές του A για τις οποίες έχει εκτιμηθεί μέχρι την $n-1$ επανάληψη ότι αντιστοιχούν σε νότες και την υπό εξέταση εγγραφή $A'_{i,:}$. Ομοίως οι $n-1$ πρώτες γραμμές του CopyStoreW είναι οι $n-1$ πρώτες γραμμές του StoreW ενώ η γραμμή n είναι μια αρχική τιμή που δίνεται στο άγνωστο βάρος w_i. Η τιμή αυτή δεν δίνεται τυχαία αλλά υπολογίζεται από τη σχέση:

$$InitialW = X(k-15) - P^{(n-1)}(k-15)$$

όπου k είναι το κλειδί της νότας στην οποία αντιστοιχεί η εγγραφή $A'_{i,:}$. Με αυτόν τον τρόπο επιτυγχάνεται καλύτερη αρχικοποίηση των τιμών του πίνακα CopyStoreW έτσι ώστε ο LMS να συγκλίνει γρηγορότερα και σε καλύτερα αποτελέσματα. Σημειώνεται ότι στο βήμα αυτό δεν υπολογίζεται απλά μια τιμή w_i για την υπό εξέταση εγγραφή $A_{i,:}$ αλλά τροποποιούνται κατάλληλα και τα w που έχουν υπολογιστεί στις προηγούμενες επαναλήψεις του αλγορίθμου. Ο αλγόριθμος LMS περιγράφεται αναλυτικότερα στον **Ψευδοκώδικα 3**.

Βήμα 3 -Απόρριψη εγγραφής αν το υπολογιζόμενο *w* **είναι πολύ μικρό.** *Αν το βάρος w_i που επιστρέφεται στη θέση n του πίνακα CopyStoreW έχει τιμή μικρότερη από ένα threshold T=0.09 τότε η εγγραφή $A'_{i,:}$ απορρίπτεται και ο αλγόριθμος επιστρέφει στο* **Βήμα 2** *για να εξεταστεί η επόμενη εγγραφή του A' $A'_{i+1,:}$. Διαφορετικά ο αλγόριθμος συνεχίζει στο* **Βήμα 4** .

Βήμα 4 -Επιλογή της $A'_{i,:}$ ως εγγραφής που αντιστοιχεί σε πιθανή νότα ή απόρριψή της. *Υπολογίζεται ο πίνακας*

$$P^{(n)} = \sum_{m=1}^{n} CopyStoreW(m) \cdot CopyStoreKeys(m,:)$$

και η ευκλείδεια norm $d = ||X - P^{(n)}||$. Αν ικανοποιούνται οι δύο παρακάτω σχέσεις:

$$Min > d$$

$$PrevMin - d > 0.06$$

τότε η εγγραφή $A'_{i,:}$ είναι πιθανόν να αντιστοιχεί σε νότα που υπάρχει στο X. Σημειώνεται ότι οι δύο αυτές σχέσεις είναι ισοδύναμες με τις 5.3 και 5.4. Έτσι αποθηκεύεται σε μια μεταβλητή PosKey η εγγραφή $A'_{i,:}$ και ο αντίστοιχος πίνακας των βαρών CopyStoreW στην μεταβλητή PosW. Επίσης, ενημερώνεται και η μεταβλητή min θέτοντας $min = d$. Ο αλγόριθμος επιστρέφει στο **Βήμα 2** *για να εξεταστεί η επόμενη εγγραφή του A΄.*

Αν οι δύο σχέσεις δεν ικανοποιούνται τότε η $A'_{i,:}$ απλά απορρίπτεται και ο αλγόριθμος ομοίως συνεχίζει στο **Βήμα 2**.

Τα βήματα 2, 3 και 4 επαναλαμβάνονται μέχρι να εξεταστούν όλες οι εγγραφές του πίνακα A΄. Όταν γίνει αυτό ο αλγόριθμος συνεχίζει στο **Βήμα 5.**

Βήμα 5. -Προσθήκη της νέας νότας στον keys ή τερματισμός του αλγορίθμου. *Στο σημείο αυτό έχουν εξεταστεί όλες οι εγγραφές του πίνακα A΄ και οι τιμές για τις οποίες ικανοποιούνται οι σχέσεις 5.3 και 5.4 είναι αποθηκευμένες στις μεταβλητές PosKey και PosW. Σε περίπτωση που δεν έχει βρεθεί εγγραφή $A'_{i,:}$ που να ικανοποιεί τις δύο αυτές σχέσεις (η PosKey δηλαδή είναι κενή), σημαίνει πως δεν μπόρεσε να εντοπιστεί κάποια καινούρια νότα στην n-οστή επανάληψη και έτσι ο αλγόριθμος περνάει στο* **βήμα 6** *του τερματισμού έχοντας εντοπίσει συνολικά $n-1$ νότες στο X.*

Διαφορετικά, η νότα που αντιστοιχεί στην εγγραφή που έχει αποθηκευτεί στην PosKey θεωρείται ως μια ακόμα νότα του X και το κλειδί της σημειώνεται στη θέση n του πίνακα keys. Η εγγραφή του A΄ που βρίσκεται στον PosKey αποθηκεύεται επίσης και στην γραμμή n του πίνακα StoreKeys ενώ ενημερώνεται κατάλληλα και ο StoreW. Επιπλέον, διαγράφονται οι εγγραφές του A΄ που αντιστοιχούν στη συγκεκριμένη νότα ώστε η ίδια νότα να μην εντοπιστεί πάλι στην επόμενη επανάληψη του αλγορίθμου. Διαγράφονται δηλαδή 9 εγγραφές του A΄. Τέλος ενημερώνεται και η μεταβλητή PrevMin θέτοντας $PrevMin = min$. Με αυτόν τον τρόπο ο αλγόριθμος θυμάται την τιμή min που υπολογίστηκε στην n-οστή επανάληψη και θα επιλέξει να προσθέσει μια καινούρια νότα στην επανάληψη $n+1$ μόνο αν η αντίστοιχη τιμή min που υπολογίζεται είναι σημαντικά μικρότερη από την προηγούμενη PrevMin όπως ορίζεται και στη σχέση 5.4. Τα βήματα 1 έως 5 επαναλαμβάνονται μέχρι να συμπληρωθούν το πολύ MaxPol επαναλήψεις.

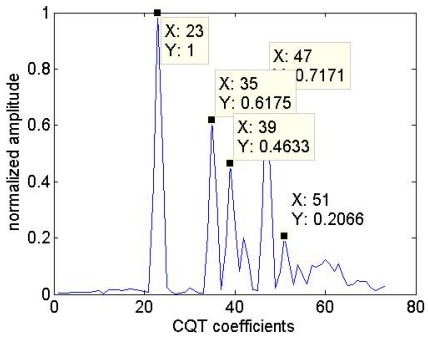
Βήμα 6. -Τερματισμός *Στο σημείο αυτό θεωρείται ότι έχουν εντοπιστεί όλες οι νότες του X και τα αντίστοιχα κλειδιά έχουν αποθηκευτεί στον πίνακα keys. Αν N ο αριθμός των επαναλήψεων που πραγματοποιήθηκαν τελικά, ο keys θα έχει N στοιχεία που αντιστοιχούν σε κλειδιά νοτών ενώ τα τελευταία $MaxPol - N$ στοιχεία θα είναι μηδενικά.*

Ψευδοκώδικας 3 Περιγραφή της συνάρτησης LMS. Πρέπει να βρεθούν τα βέλτιστα w ώστε η ποσότητα $w \cdot inp$ να προσεγγίζει όσο το δυνατόν καλύτερα το d. Κάθε φορά εξετάζεται ένα στοιχείο του διανύσματος d και με βάση αυτό τροποποιούνται τα w. Τα στοιχεία του διανύσματος d είναι σχετικά λίγα (το μέγεθος του d είναι 1×73) και δεν επαρκούν για τη σύγκλιση του αλγορίθμου. Έτσι γίνονται τρεις επαναλήψεις. Κάθε στοιχείο δηλαδή του d εξετάζεται τρεις φορές ώστε να υπάρχουν αρκετά δεδομένα για σύγκλιση του αλγορίθμου.

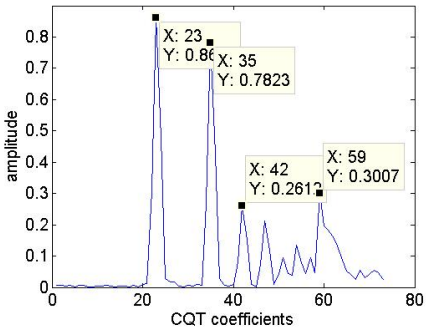
```
function w = LMS(d, inp,w)
DEFINE Max Number of Interations ← 3;
for i=1 to Max Number of Interations
   for n=1 to length of d
      u = inp(:,n);
      y= w' · u;
      e = d(n) - y ;
      if (i==1)% αρχικά χρησιμοποιείται μέγαλο mu για να
        mu=0.35; % επιταγχυνθεί η σύγκλιηση του αλγορίθμου
      elseif (i==2)
        mu=0.25;
      else % στην τελευταία επανάληψη χρησιμοποιείται μικρό mu
        mu=0.12; % ωστε να βρεθούν με μεγαλύτερη ακρίβεια τα σωστά βάρη
      end if;
      w = w + mu · u · e ;
   end for;
end for;
```

Στα παρακάτω σχήματα φαίνεται ένα παράδειγμα εφαρμογής των αλγορίθμων 5.1 και 5.2. Στο σχήμα 5.2 φαίνεται το X όπως προκύπτει από το **Βήμα 1**

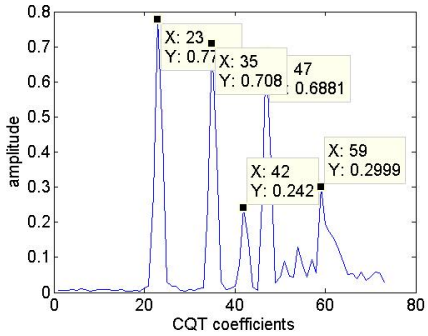
του *Αλγορίθμου 5.1*. Οι νότες που ακούγονται στο συγκεκριμένο frame είναι οι $A\#3$ με κλειδί 38, $D5$ με κλειδί 54 και $A\#5$ με κλειδί 62. Τα αντίστοιχα τοπικά μέγιστα βρίσκονται στις θέσεις 23, 39 και 47. Ο πίνακας MaxInd όπως προκύπτει στο **βήμα 3** του 5.1 έχει 10 στοιχεία τα (38 50 54 57 62 66 69 72 75 78). Τα δέκα αυτά στοιχεία είναι πιθανόν να αντιστοιχούν σε κλειδιά νοτών που υπάρχουν στο συγκεκριμένο frame. Ο πίνακας Α′ που δημιουργείται θα έχει συνολικά 90 εγγραφές (εννέα για κάθε πιθανή νότα).



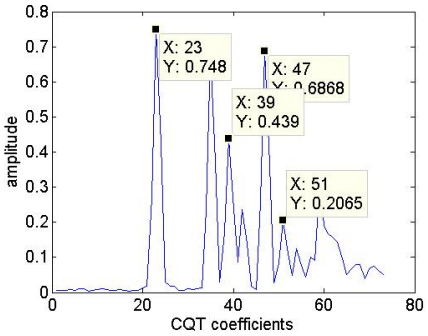
(α′) Το X όπως προκύπτει από το βήμα 1 του αλγορίθμου 5.1.



(β′) $P^{(1)}$



(γ′) $P^{(2)}$



(δ′) $P^{(3)}$

Σχήμα 5.3: Παράδειγμα εφαρμογής των αλγορίθμων 5.1 και 5.2. Στο σχήμα (α′) φαίνεται ο CQT ενός frame του σήματος το όποιο πρέπει να αναγνωριστεί. Στα επόμενα γραφήματα απεικονίζεται η σταδιακή προσέγγιση του X σε κάθε επανάληψη του αλγορίθμου 5.2 χρησιμοποιώντας τα δεδομένα εκπαίδευσης μέσω της συνάρτησης $P^{(n)} = \sum_{m=1}^{n} StoreW(m) \cdot StoreKeys(m,:)$ με n=1,2,3.

Στο τέλος της πρώτης επανάληψης του *Αλγορίθμου 5.2* (**Βήμα 5**), αφού εξετάστηκαν όλες οι εγγραφές του Α′, βρέθηκε ως καταλληλότερη μια από τις εγγραφές που αντιστοιχούν στη νότα με κλειδί 38. Ενημερώθηκαν οι κατάλληλες μεταβλητές και διαγράφονται εννέα εγγραφές του Α′. Στο σχήμα 5.2(β′) απεικονίζεται η προσέγγιση του X ως γινόμενο των πινάκων $StoreW$ και $StoreKeys$ όπως προκύπτει από την πρώτη επανάληψη του αλγορίθμου 5.2. Η προσέγγιση αυτή δίνεται από τη σχέση $P^{(1)} = StoreW(1) \cdot StoreKeys(1,:)$.

Η δεύτερη επανάληψη του 5.2 οδηγεί στην εύρεση της νότας με κλειδί το 62. Στο σχήμα 5.2(γ′) φαίνεται η προσέγγιση του X από τα δεδομένα εκπαίδευσης όπως προκύπτει από της δύο πρώτες επαναλήψεις. Απεικονίζεται δηλαδή ο πίνακας $P^{(2)} = \sum_{m=1}^{2} StoreW(m) \cdot StoreKeys(m,:)$

Με την τρίτη επανάληψη του 5.2 εντοπίζεται η νότα με κλειδί το 54. Στο σχήμα 5.2(δ′) απεικονίζεται ο πίνακας $P^{(3)} = \sum_{m=1}^{3} StoreW(m) \cdot StoreKeys(m,:)$. Στην τέταρτη επανάληψη καμία από τις 63 εγγραφές του Α′ που έχουν μείνει δεν ικανοποιεί τις σχέσεις 5.3 και 5.4 καθώς και τους περιορισμούς που έχουν τεθεί για το βάρος w έτσι ο αλγόριθμος τερματίζει έχοντας εκτελέσει τέσσερις επαναλήψεις στις οποίες εντοπίστηκαν με επιτυχία οι τρεις νότες που ακούγονται στο συγκεκριμένο frame.

5.2 Εξαγωγή νοτών συγκεκριμένης χρονικής διάρκειας

Με την εφαρμογή της διαδικασίας που περιγράφεται στην ενότητα 5.1 προκύπτει πληροφορία για το ποιες νότες ακούγονται σε κάθε frame. Η πληροφορία αυτή θα χρησιμοποιηθεί για την εξαγωγή νοτών με καθορισμένη χρονική διάρκεια και οργανώνεται στον πίνακα AllKeys μεγέθους N×73 όπου N ο αριθμός των frames που χρησιμοποιήθηκαν. Αν στο frame n έχει βρεθεί μόνο μια νότα με κλειδί k, στη γραμμή n της στήλη $k-15$ του πίνακα θα υπάρχει η τιμή 1 ενώ όλες οι άλλες στήλες στη συγκεκριμένη γραμμή θα έχουν την τιμή 0. Ο πίνακας AllKeys δηλαδή περιγράφεται από τη σχέση 5.5:

$$AllKeys_{n,k} = \begin{cases} 1 & \text{if } k+15 \ \exists \text{ in frame n} \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

Πριν την εύρεση της χρονικής διάρκειας για κάθε νότα πρέπει να διορθωθούν κάποια από τα δεδομένα του πίνακα AllKeys όπως αναφέρθηκε και στην αρχή του κεφαλαίου 5. Συγκεκριμένα η διαδικασία που ακολουθείται περιλαμβάνει τρία στάδια:

1. Νότες διάρκειας μικρότερης των 80msec δηλαδή νότες που η παρουσία τους δηλώνεται σε ένα μόνο frame διαγράφονται.

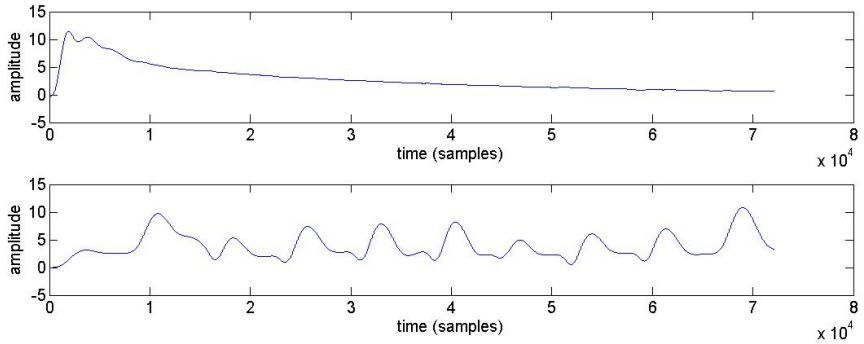
2. Διακεκομμένες νότες, δηλαδή νότες που εξαφανίζονται για σύντομα χρονικά διαστήματα και στη συνέχεια επανεμφανίζονται πρέπει να ελεγχθούν για το αν αποτελούν μία ενιαία νότα ή επαναλαμβανόμενες νότες. Αν αποτελούν μία νότα τα χωριστά τμήματα πρέπει να ενωθούν.

3. Τέλος, εφαρμόζεται αλγόριθμος για τον εντοπισμό επαναλαμβανόμενων νοτών οι οποίες πιθανόν να εμφανίζονται ως μία νότα.

Σε μονοφωνική μουσική είναι εύκολο να αντιστοιχηθεί ένα onset με μια νότα καθώς υπάρχει μόνο μία νότα σε κάθε χρονική στιγμή. Ωστόσο σε πολυφωνική μουσική η διαδικασία αυτή είναι πιο δύσκολη. Ένα onset δηλώνει ότι τη συγκεκριμένη χρονική στιγμή αρχίζει μια καινούρια νότα όμως δε δίνει πληροφορία για το ποια είναι αυτή η νότα. Έτσι όταν στον πίνακα AllKeys εμφανίζεται μια νότα σε έναν αριθμό από συνεχόμενα frames και στο αντίστοιχο χρονικό διάστημα εμφανίζονται παραπάνω από ένα onsets πρέπει να ελεγχθεί αν τα onsets αυτά προέρχονται από αυτή τη νότα η από κάποιες άλλες που ακούγονται ταυτόχρονα. Αν προέρχονται από αυτήν, πρόκειται για μια επαναλαμβανόμενη νότα. Αντίστροφα, αν μια νότα στον AllKeys εμφανίζεται διακεκομμένη και δεν έχουν βρεθεί onsets που να αντιστοιχούν σε αυτήν τότε η νότα αυτή πρέπει να ενωθεί.

Για να καθοριστεί αν κάποιο onset αντιστοιχεί σε μία συγκεκριμένη νότα ή σε κάποια άλλη η διαδικασία που ακολουθείται είναι σχετικά απλή. Για κάθε προβληματική νότα (διακεκομμένη νότα ή νότα για την οποία εντοπίζονται περισσότερα του ενός onsets) το σήμα εισόδου φιλτράρεται χρησιμοποιώντας ένα ζωνοπερατό ελλειπτικό φίλτρο με ζώνη συχνοτήτων τέτοια ώστε να κόβονται όλες οι συχνότητες που αντιστοιχούν σε βασικές συχνότητες κλειδιών πιάνου εκτός από τη βασική συχνότητα της υπό εξέτασης νότας. Στη συνέχεια στην περιβάλλουσα της κυματομορφής που προκύπτει μετά το φιλτράρισμα εντοπίζονται τα τοπικά μέγιστα τα οποία θεωρούνται τα onsets που αντιστοιχούν στην υπό

εξέταση νότα. Ένα παράδειγμα φαίνεται στο σχήμα 5.4 όπου απεικονίζονται οι περιβάλλουσες που προκύπτουν μετά το φιλτράρισμα του αρχικού σήματος για δύο νότες που ακούγονται σε ένα χρονικό διάστημα. Η πρώτη είναι η νότα D#5 η οποία εμφανίζεται διακεκομμένη στον πίνακα AllKeys ενώ η δεύτερη η $C4$. Η $C4$ εμφανίζεται ως μία ενιαία νότα στον AllKeys αλλά στην πραγματικότητα όπως φαίνεται στο σχήμα 5.4 πρόκειται για εννιά επαναλαμβανόμενες νότες. Αντίθετα η D#5 είναι μία ενιαία νότα σχετικά μεγάλης χρονικής διάρκειας της οποίας η εξαφάνιση σε κάποια frames οφείλεται στην παρεμβολή της $C4$.



Σχήμα 5.4: Στο πάνω γράφημα η περιβάλλουσα του σήματος που προκύπτει στην έξοδο του φίλτρου με ζώνη συχνοτήτων [604.8 640.75] που χρησιμοποιήθηκε για τη νότα D#5 και στο κάτω η η περιβάλλουσα του σήματος στην έξοδο του φίλτρου με ζώνη συχνοτήτων [254.28 269.4] για την $C4$. Οι δύο περιβάλλουσες απεικονίζονται σε χρονική αντιστοιχία.

Αναλυτικότερα, πρέπει να βρεθούν οι διακεκομμένες νότες και να ενωθούν αν αυτό είναι απαραίτητο. Η διαδικασία περιγράφεται στον *Ψευδοκώδικα 5*. Για να βρεθεί η αρχή και το τέλος κάθε νότας μέσα στον AllKeys χρησιμοποιείται η συνάρτηση διαφορών κάθε στήλης του AllKeys. Υπενθυμίζεται ότι η στήλη i του AllKeys θα έχει την τιμή 1 σε κάθε frame όπου εμφανίζεται η νότα με κλειδί $i + 15$ και 0 διαφορετικά. Έτσι η τιμή 1 στη θέση j της συνάρτησης διαφορών θα δηλώνει την αρχή της νότας $i + 15$ στο frame j ενώ η τιμή -1 στη θέση $j + k$ το τέλος της στο frame $j + k$. Όταν το τέλος μιας νότας απέχει από την αρχή μιας άλλης όμοιας νότας λιγότερο από 5 frames τότε πρέπει να ελεγχθεί η περίπτωση οι δύο νότες να αποτελούν μια ενιαία νότα. Ο έλεγχος αυτός γίνεται φιλτράροντας το σήμα κατάλληλα ώστε να επιβιώνουν μόνο συχνότητες

πολύ κοντά στη βασική συχνότητα της υπό εξέταση νότας. Συγκεκριμένα αν εξετάζεται η στήλη i του AllKeys τότε η υπό εξέταση νότα είναι αυτή με κλειδί το $i + 15$. Έστω πίνακας $F0$ με τις βασικές συχνότητες όλων των νοτών και $F0(k)$ η βασική συχνότητα της νότας με κλειδί k τότε το εύρος ζώνης του φίλτρου που χρησιμοποιείται για την νότα με κλειδί k είναι $[D(k)\ U(k)]$ με:

$$D(k) = \frac{F0(k) - F0(k-1)}{2} + F0(k-1)$$

και

$$U(k) = \frac{F0(k+1) - F0(k)}{2} + F0(k)$$

Ψευδοκώδικας 4 Η συνάρτηση ComputeDiffEnv φιλτράρει το σήμα εισόδου x με φίλτρο με ζώνη συχνοτήτων $[D(i)U(i)]$ όπως ορίστηκε παραπάνω και υπολογίζει την παράγωγο του λογαρίθμου της απόλυτης τιμής της περιβάλλουσας του σήματος που προκύπτει.

```
BEGIN function ComputeDiffEnv(x, D(i),U(i))
    SET f ← elliptic bandpass filter with passband [D(i) U(i)];
    s = filter(f,x(start:Ends(end)));
    s = |s|;
    SET env ← envelope of s;
    SET indx ← index to values of env less than 0.01;
    env(indx) = 0.01;
    env = log(env);
    env = downsample(env);
    DiffEnv = diff(env);
    SET indx ← index to values of DiffEnv less than 0.1;
    DiffEnv(indx) = 0;
    RETURN DiffEnv;
END function;
```

Στη συνέχεια υπολογίζεται η περιβάλλουσα του σήματος στην έξοδο του φίλτρου και εντοπίζονται τα τοπικά μέγιστα ακολουθώντας διαδικασία παρόμοια με αυτή που εφαρμόστηκε στον Onset Detector που περιγράφεται στην ενότητα

4.3. Δηλαδή χρησιμοποιείται η παράγωγος του λογαρίθμου της απόλυτης τιμής της περιβάλλουσας για τον εντοπισμό των τοπικών μεγίστων. Σημειώνεται ότι δεν υπολογίζεται η περιβάλλουσα ολόκληρου του σήματος αλλά μόνο του υπό εξέταση τμήματος. Αν για τις νότες ν_1, ν_2, ... ν_i ισχύει ότι το τέλος της μίας απέχει λιγότερο από πέντε frames από την αρχή της επόμενής της τότε ως αρχή του υπό εξέταση τμήματος ορίζεται η αρχή της ν_1 και ως τέλος το τέλος της ν_i. Τέλος, αν μεταξύ της αρχής μιας νότας και το τέλος της επόμενής της δε βρεθεί κάποιο onset, οι δύο νότες ενώνονται σε μία αντικαθιστώντας τα μηδενικά ανάμεσά τους στον πίνακα AllKeys με άσους.

Ψευδοκώδικας 5 Περιγραφή της διαδικασίας εντοπισμού διακεκομμένων νοτών και ένωσή τους όταν χρειάζεται.

```
SET x ← input signal;
for each column i of AllKeys
   DAllKeys = diff(AllKeys(:,i));
   SET st ← position of firt 1 in DAllKeys;
   SET en ← position of firt -1 in DAllKeys;
   while (isempty(st)==FALSE)
      SET new_st ← position of firt 1 after st in DAllKeys;
      while ((isempty(new_st)==FALSE)&&(new_st - en <5))
         SET en ← position of firt -1 after new_st in DAllKeys;
         ADD en to array Ends
         SET new_st ← position of firt 1 after en in DAllKeys;
      end while;
      if (isempty(Ends)==FALSE)
         DiffEnv = ComputeDiffEnv(x(start:Ends(end)), D(i),U(i)); %Ψευδοκώδικας 4
         peaks = findpeaks(DiffEnv);
         for j=1 to length(Ends)
            SET a ← all peaks between st and Ends(j);
            if (isempty(a)==FALSE)
               AllKeys(st:Ends(j),i) = 1;
            end if;
         end for;
      end if;
   end while;
end for;
```

Ψευδοκώδικας 6 Περιγραφή της διαδικασίας εντοπισμού επαναλαμβανόμενων νοτών.

```
SET x ← input signal;
for each line i of array Keys
   SET st ← the first onset after the temporal position keys(i,2)-0.08 · Fs ;
   SET en ← the first onset before the temporal position keys(i,3)-0.12 · Fs ;
   SET CurOns ← all onsets between st and en;
   SET PsblOns ← empty; %Στον πίνακα αυτό θα αποθηκευτούν μόνο τα onsets του
   CurOns τα οποία είναι πιαθανόν να αποτελούν onsets της υπό εξέτασης νότας. Πιθανά onsets
   θεωρούνται μόνο αυτά που χωρίζουν τη νότα σε νότες με διάρκεια μεγαλύτερη των 16msec.
   st = keys(i,2);
   if (length(CurOns)>1)
      for j=1 to length(CurOns)
         if ((CurOns(j)-st> 0.16 · Fs)&&(keys(i,3)-CurOns(j)> 0.16 · Fs))
            ADD CurOns(j) to PsblOns;
            st = CurOns(j);
         end if;
      end for;
      if (isempty(PsblOns)==FALSE)
         DiffEnv = ComputeDiffEnv(x(start:Ends(end)), D(i),U(i)); %Ψευδοκώδικας 4
         peaks = findpeaks(DiffEnv);
         for j=1 to length(PsblOns)
            indx = find all peaks between PsblOns(j)−0.1 · Fs and PsblOns(j)+0.12 · Fs;
            if (isempty(indx)==TRUE) % Διαγράφονται από τον PsblOns εγγραφές που δεν
            ταιριάζουν με τα υπολογισμένα από τον Onset Detector onsets
               delete PsblOns(j) from array PsblOns;
            end if;
         end for;
         if (isempty(PsblOns)==FALSE)
            st = keys(i,2);
            for j=1 to length(PsblOns)
               NewNote = [keys(i,1) st PsblOns(j)];
               ADD NewNote to Keys;
               st = PsblOns(j);
            end for;
         end if;
      end if;
end for;
```

Αφού έχουν αφαιρεθεί οι νότες πολύ μικρής διάρκειας και έχουν ενωθεί οι διακεκομμένες νότες, η πληροφορία που προέκυψε οργανώνεται στον πίνακα Keys μεγέθους N×3 όπου N ο αριθμός των νοτών που εντοπίστηκαν. Στην πρώτη στήλη του Keys σημειώνεται το κλειδί της νότας, στη δεύτερη η χρονική στιγμή στην οποία αρχίζει και στην τρίτη η χρονική στιγμή στην οποία τελειώνει. Οι νότες τοποθετούνται στον Keys με χρονική σειρά και φορά από την χαμηλότερη προς την υψηλότερη νότα. Τέλος, αυτό που μένει για να ολοκληρωθεί η διαδικασία εξαγωγής των νοτών είναι ο εντοπισμός των επαναλαμβανόμενων νοτών. Ο αλγόριθμος που αναπτύχθηκε για αυτό το σκοπό περιγράφεται με τον *Ψευδοκώδικα 6*.

Ο αλγόριθμος που περιγράφεται στον *Ψευδοκώδικα 6* εξετάζει κάθε μία από τις νότες που έχουν εντοπιστεί, δηλαδή τις γραμμές του Keys. Για κάθε νότα βρίσκει τα onsets που έχουν εντοπιστεί από τον Onset Detector σε διάστημα λίγο πριν την αρχή και λίγο πριν το τέλος της. Εξετάζεται αν η νότα μπορεί να χωριστεί σε περισσότερες νότες σύμφωνα με αυτά τα onsets που βρέθηκαν. Ένα onset θεωρείται πιθανόν να αντιστοιχεί σε επαναλαμβανόμενη νότα μόνο αν χωρίζοντας την αρχική νότα σύμφωνα με αυτό οι νότες που προκύπτουν έχουν χρονική διάρκεια μεγαλύτερη των 16msec. Τα onsets που ικανοποιούν αυτό το κριτήριο τοποθετούνται στον πίνακα PsblOns. Στη συνέχεια πρέπει να βρεθεί αν τα onsets του PsblOns αντιστοιχούν στην υπό εξέταση νότα ή σε κάποια άλλη που ακούγεται ταυτόχρονα. Για να γίνει αυτό το ακολουθείται διαδικασία όμοια με αυτή του αλγορίθμου για τον εντοπισμό των διακεκομμένων νοτών δηλαδή το σήμα φιλτράρεται κατάλληλα και υπολογίζονται τα τοπικά μέγιστα της περιβάλλουσας του σήματος που προκύπτει. Τα onsets του PsblOns που δε βρίσκονται σχετικά κοντά με κάποιο από τα τοπικά μέγιστα που υπολογίστηκαν διαγράφονται. Τέλος, αν παραμείνουν κάποιες εγγραφές στον PsblOns σημαίνει ότι εντοπίστηκε μια επαναλαμβανόμενη νότα. Η εγγραφή του Keys που εξετάζεται τροποποιείται αλλάζοντας την τρίτη στήλη που αφορά το τέλος της νότας και προστίθενται στον Keys οι καινούριες νότες με κλειδί όμοιο με αυτό της υπό εξέτασης νότας και χρονική διάρκεια που προκύπτει από τα onsets του PsblOns.

Η μέθοδοι που περιγράφηκαν για τον εντοπισμό διακεκομμένων και επαναλαμβανόμενων νοτών λειτουργούν επιτυχημένα στις περισσότερες περιπτώσεις. Όμως συχνά παρουσιάζονται προβλήματα όταν εξετάζονται δύο νότες οι οποίες σχετίζονται αρμονικά μεταξύ τους. Φιλτράροντας ένα σήμα ώστε να επιβιώνει

μόνο μία νότα είναι πιθανόν να προκύπτουν κάποια τοπικά μέγιστα τα οποία να μην οφείλονται σε αυτήν αλλά σε κάποια άλλη νότα μια ή περισσότερες οκτάβες χαμηλότερη από την υπό εξέταση νότα. Για παράδειγμα έστω ότι μια νότα ν1 που ακούγεται στο χρονικό διάστημα (t1,t2) εμφανίζεται διακεκομμένη. Επιπλέον την χρονική στιγμή t3 με $t1 < t3 < t2$ αρχίζει μια άλλη νότα ν2 μια οκτάβα χαμηλότερη από την ν1. Είναι πιθανόν, στην περιβάλλουσα του σήματος που προκύπτει μετά από κατάλληλο φιλτράρισμα ώστε να επιτρέπεται μόνο η διέλευση της ν1 να εμφανιστεί τοπικό μέγιστο τη χρονική στιγμή t3 και έτσι να θεωρηθεί ότι πρόκειται για δύο επαναλαμβανόμενες νότες και όχι για μία ενιαία όπως είναι πραγματικά.

Κεφάλαιο 6

Παρουσίαση αποτελεσμάτων

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα που προέκυψαν μετά την εφαρμογή των τριών αλγορίθμων που παρουσιάζονται στα κεφάλαια 4 και 5. Για την αξιολόγηση των αλγορίθμων έγιναν κάποιες ηχογραφήσεις σε αθόρυβο περιβάλλον με ρυθμό δειγματοληψίας 32000Hz χρησιμοποιώντας ηλεκτρικό πιάνο. Στην ενότητα 6.1 παρουσιάζονται συγκριτικά τα αποτελέσματα των δύο αλγορίθμων που υλοποιήθηκαν για επεξεργασία μονοφωνικής μουσικής και στην 6.2 τα αποτελέσματα του αλγορίθμου επεξεργασίας πολυφωνικής μουσικής.

6.1 Αποτελέσματα αλγορίθμων επεξεργασίας μονοφωνικής μουσικής

Για την αξιολόγηση των δύο αλγορίθμων που περιγράφονται στην ενότητα 4 ηχογραφήθηκαν οι μελωδίες από κάποια μουσικά κομμάτια. Συγκεκριμένα ηχογραφήθηκαν αποσπάσματα από φούγκες του Bach, ένα μέρος της σονάτας Alla Turka του Motzart, καθώς και αποσπάσματα από τα Decisive Battle και Battle with Gilgamesh του Nuebo Uematsu. Επειδή η μελωδία ενός κομματιού συνήθως περιλαμβάνει νότες μεσαίων και υψηλών συχνοτήτων κάθε κομμάτι παίζεται σε διαφορετικές οκτάβες έτσι ώστε να καλύπτεται μεγαλύτερο εύρος συχνοτήτων. Χρησιμοποιήθηκαν συνολικά 1072 νότες για την αξιολόγηση των αλγορίθμων.

Τα αποτελέσματα παρουσιάζονται σε πίνακες. Δημιουργείται ένας ξεχωριστός πίνακας για κάθε σύνθεση. Κάθε πίνακας αποτελείται από έξι στήλες. Στην 1η

στήλη σημειώνεται ο αλγόριθμος που χρησιμοποιήθηκε, ο αριθμός 1 αντιστοιχεί στον αλγόριθμο που επεξεργάζεται τα δεδομένα στο πεδίο του χρόνου και περιγράφεται στην ενότητα 4.1 ενώ ο αριθμός 2 στον αλγόριθμο που περιγράφεται στην ενότητα 4.2. Στη 2η στήλη αναγράφεται το εύρος νοτών του κομματιού, δηλαδή η χαμηλότερη και η υψηλότερη νότα του. Στη 3η στήλη σημειώνεται το ποσοστό των νοτών που εντοπίστηκαν σωστά καθώς και το ποσοστό των λανθασμένων νοτών. Μια νότα θεωρείται ότι εντοπίστηκε με επιτυχία όταν το κλειδί της είναι ίδιο με την πραγματική νότα και όταν η διάρκειά της διαφέρει με τη διάρκεια της πραγματικής νότας λιγότερο από 80msec. Το ποσοστό των σωστών νοτών υπολογίζεται ως το πηλίκο των νοτών που εντοπίστηκαν σωστά προς τον αριθμό των πραγματικών νοτών. Το ποσοστό των λάθος νοτών υπολογίζεται ως το πηλίκο των νοτών που εντοπίστηκαν λάθος προς το σύνολο των νοτών που εντοπίστηκαν. Στην 4η στήλη σημειώνονται οι νότες που δεν εντοπίστηκαν. Οι νότες αυτές χωρίζονται σε τρεις κατηγορίες ανάλογα με τον λόγο για τον οποίο δεν εντοπίστηκαν. Η πρώτη κατηγορία περιλαμβάνει τα σφάλματα οκτάβας, δηλαδή όταν μια νότα δεν εντοπίζεται και στη θέση της εντοπίζεται λανθασμένα μια νότα που διαφέρει από αυτήν κατά μια οκτάβα. Η δεύτερη κατηγορία αφορά επαναλαμβανόμενες νότες για τις οποίας δεν εντοπίζονται τα αντίστοιχα onsets έτσι περισσότερες από μία ίδιες νότες εμφανίζονται σαν μια ενιαία νότα. Η τρίτη κατηγορία αφορά λάθη που δεν ανήκουν σε καμία από τις δύο προηγούμενες κατηγορίες. Σε κάθε κατηγορία σημειώνεται το επί τις εκατό ποσοστό των νοτών που δεν εντοπίστηκαν και ανήκουν σε αυτή τη κατηγορία. Στην 5η στήλη σημειώνονται οι νότες που εντοπίστηκαν λανθασμένα. Ομοίως οι νότες αυτές κατηγοριοποιούνται ως λάθη οκτάβας, λάθη διάρκειας και άλλα λάθη. Σε κάθε στήλη αναγράφεται το ποσοστό των λανθασμένων που ανήκουν σε αυτήν την κατηγορία. Έτσι το άθροισμα των τριών κατηγοριών της στήλης αυτής θα δίνει την τιμή 100 ή 0 σε περίπτωση που δεν υπάρχει καμία λανθασμένη νότα. Αν για κάποια νότα εντοπιστεί σωστά το κλειδί της άλλα λάθος η διάρκειά της τότε η νότα δε θεωρείται ότι εντοπίστηκε σωστά. Το λάθος που προκύπτει ανήκει και στις λανθασμένες νότες αλλά και στις νότες που δεν εντοπίστηκαν. Όμως το λάθος αυτό σημειώνεται μόνο στην 4η στήλη στην κατηγορία των λαθών διάρκειας. Έτσι το άθροισμα των κατηγοριών της 3η στήλης θα έχει αποτέλεσμα διαφορετικό του 100 στην περίπτωση που υπάρχουν λάθη διάρκειας. Στην τελευταία στήλη αναγράφεται ο αριθμός των νοτών του συγκεκριμένου κομματιού.

Αλγό-ριθμος	Εύρος νοτών	Σωστ. νότες	Λάθ. νότες	Νότες που δεν εντοπίστηκαν			Λάθος νότες			αρ. νοτών
				οκτ.	επαν.	άλλο	οκτ.	διάρκ.	άλλο	
1	E4-C6	93.6	0	0	0	100	0	0	0	78
2	E4-C6	96.2	0	0	0	100	0	0	0	78
1	E3-C5	92.3	2.7	33.3	0	66.7	100	0	0	78
2	E3-C5	96.2	2.6	33.3	0	66.7	50	0	50	78
1	E2-C4	91	2.7	28.6	0	71.4	100	0	0	78
2	E2-C4	94.9	2.6	25	0	75	50	0	50	78

Πίνακας 6.1: Alla Turka σε τρεις διαφορετικές οκτάβες

Αλγό-ριθμος	Εύρος νοτών	Σωστ. νότες	Λάθ. νότες	Νότες που δεν εντοπίστηκαν			Λάθος νότες			αρ. νοτών
				οκτ.	επαν.	άλλο	οκτ.	διάρκ.	άλλο	
1	A3-D#6	92.4	1.8	0	0	100	100	0	0	119
2	A3-D#6	100	0	0	0	0	0	0	0	119
1	A2-D#5	90.8	4.42	36.4	0	63.6	100	0	0	119
2	A2-D#5	100	0	0	0	0	0	0	0	119

Πίνακας 6.2: Decisive Battle σε δύο διαφορετικές οκτάβες

Αλγό-ριθμος	Εύρος νοτών	Σωστ. νότες	Λάθ. νότες	Νότες που δεν εντοπίστηκαν			Λάθος νότες			αρ. νοτών
				οκτ.	επαν.	άλλο	οκτ.	διάρκ.	άλλο	
1	B3-B5	92	2	15.4	0	84.6	66.7	0	33.3	162
2	B3-B5	99.4	0	0	0	100	0	0	0	162
1	B2-B4	90.7	3.3	20	0	80	80	0	20	162
2	B2-B4	99.4	0	0	0	100	0	0	0	162

Πίνακας 6.3: Φούγκα (Bach) σε δύο διαφορετικές οκτάβες

Αλγό-ριθμος	**Εύρος νοτών**	**Σωστ. νότες**	**Λάθ. νότες**	**Νότες που δεν εντοπίστηκαν**			**Λάθος νότες**			**αρ. νοτών**
				οκτ.	επαν.	άλλο	οκτ.	διάρκ.	άλλο	
1	B3-D#6	99.3	1.4	0	0	0	50	50	0	138
2	B3-D#6	99.3	1.4	0	0	0	0	50	50	138
1	B2-D#5	98.6	2.9	0	0	0	25	50	25	138
2	B3-D#6	99.3	1.4	0	0	0	0	50	50	138

Πίνακας 6.4: Battle with Gilgamesh σε δύο διαφορετικές οκτάβες

Υπολογίζονται τα precision και recall για τους δύο αλγορίθμους τα οποία ορίζονται ως εξής:

$$\text{precision} = \frac{\{\Sigma\omega\sigma\tau\acute{\epsilon}\varsigma\ \nu\acute{o}\tau\epsilon\varsigma\} \cap \{\text{Νότες που εντοπίστηκαν}\}}{\{\text{Νότες που εντοπίστηκαν}\}} \tag{6.1}$$

$$\text{recall} = \frac{\{\Sigma\omega\sigma\tau\acute{\epsilon}\varsigma\ \nu\acute{o}\tau\epsilon\varsigma\} \cap \{\text{Νότες που εντοπίστηκαν}\}}{\{\Sigma\omega\sigma\tau\acute{\epsilon}\varsigma\ \nu\acute{o}\tau\epsilon\varsigma\}} \tag{6.2}$$

Έτσι για τον πρώτο αλγόριθμο υπολογίζονται precision και recall:

$$\text{precision}_1 = \frac{998}{1023} = 97.56\%$$

$$\text{recall}_1 = \frac{998}{1072} = 93.10\%$$

ενώ για τον δεύτερο:

$$\text{precision}_2 = \frac{1058}{1066} = 99.25\%$$

$$\text{recall}_2 = \frac{1058}{1072} = 98.69\%$$

Ο δεύτερος αλγόριθμος έχει αρκετά καλύτερα αποτελέσματα. Αυτό οφείλεται στο ότι είναι πιο ανεκτικός στο θορυβώδες attack μέρος των νοτών καθώς έχουν χρησιμοποιηθεί δεδομένα εκπαίδευσης που αφορούν το attack μέρος κάθε νότας. Αντίθετα ο πρώτος επηρεάζεται αρκετά από το τμήμα αυτό των νοτών. Έτσι πολλές φορές σε νότες με σύντομη χρονική διάρκεια δεν επιτυγχάνεται

ο εντοπισμός της σωστής περιοδικότητας με αποτέλεσμα είτε να εμφανίζεται κάποια λάθος νότα ή συνηθέστερα να εμφανίζονται κάποιες διαφορετικές νότες με διάρκεια μόλις ένα frame η κάθε μία, οι οποίες τελικά διαγράφονται.

Τα περισσότερα λάθη και των δύο αλγορίθμων εμφανίζονται σε νότες σύντομης χρονικής διάρκειας 100 έως 160msec. Λάθη διάρκειας συμβαίνουν για νότες μεγάλης χρονικής διάρκειας οι οποίες εξασθενούν στο τελευταίο μέρος τους με αποτέλεσμα είτε να εξαφανίζονται είτε να εντοπίζεται λανθασμένα κάποια άλλη νότα. Επιπλέον, για νότες χαμηλότερων συχνοτήτων τα αποτελέσματα δε φαίνεται να διαφοροποιούνται ιδιαίτερα. Τέλος, λάθη επαναλαμβανόμενων νοτών δεν εμφανίζονται καθώς ο εντοπισμός τους σε μονοφωνική μουσική είναι εύκολος.

6.2 Αποτελέσματα αλγορίθμου επεξεργασίας πολυφωνικής μουσικής

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα του αλγορίθμου που περιγράφεται στο κεφάλαιο 5. Για την αξιολόγηση του αλγορίθμου ηχογραφήθηκαν διαφορετικά ήδη μουσικής όπως αποσπάσματα από από τη σονάτα Alla Turka του Motzart, μία τρίφωνη φούγκα καθώς και ένα πρελούδιο του Bach, το Battle with Gilgamesh του Nuebo Uematsu, το Little thing called love των Queen, ένας αυτοσχεδιασμός σε στυλ Boogie Woogie Blues καθώς και ο 'Δρόμος ' του Μ. Λοϊζου. Τα αποτελέσματα παρουσιάζονται σε πίνακες παρόμοιους με αυτούς που χρησιμοποιήθηκαν για τους αλγόριθμους επεξεργασίας μονοφωνικής μουσικής. Προστίθεται μία επιπλέον κατηγορία στη στήλη με τις λάθος νότες η οποία αφορά νότες οι οποίες λανθασμένα θεωρούνται ως επαναλαμβανόμενες νότες. Επιπλέον προσθέτονται δύο στήλες στο τέλος όπου αναγράφεται ο μέσος βαθμός πολυφωνίας καθώς και ο μέγιστη πολυφωνία της σύνθεσης, δηλαδή ο μέγιστος αριθμός νοτών που ακούγονται ταυτόχρονα.

Εύρος νοτών	**Σωστες νοτ.**	**Λάθος νοτ.**	**Νότες που δεν εντοπίστηκαν**			**Λάθος νότες**				**αρ. νοτ.**	**μεση πολ.**	**μεγ. πολ.**
			οκτ.	επαν.	άλλο	οκτ.	επαν.	διάρκ.	άλλο			
A2-C6	96.8	4.7	25	0	12.5	33.3	0	41.7	25	253	2.33	4

Πίνακας 6.5: Alla Turka

Εύρος νοτών	Σωστες νοτ.	Λάθος νοτ.	Νότες που δεν εντοπίστηκαν			Λάθος νότες				αρ. νοτ.	μεση πολ.	μεγ. πολ.
			οκτ.	επαν.	άλλο	οκτ.	επαν.	διάρκ.	άλλο			
A2-C6	94.4	4.7	75	0	8.4	31.8	40.9	9.2	27.3	459	3.1	6

Πίνακας 6.6: Battle with Gilgamesh

Εύρος νοτών	Σωστες νοτ.	Λάθος νοτ.	Νότες που δεν εντοπίστηκαν			Λάθος νότες				αρ. νοτ.	μεση πολ.	μεγ. πολ.
			οκτ.	επαν.	άλλο	οκτ.	επαν.	διάρκ.	άλλο			
A2-C6	97.1	11.4	28.6	7.1	14.3	58.8	0	9.8	31.8	410	2.2	4

Πίνακας 6.7: Τρίφωνη φούγκα (Bach)

Εύρος νοτών	Σωστες νοτ.	Λάθος νοτ.	Νότες που δεν εντοπίστηκαν			Λάθος νότες				αρ. νοτ.	μεση πολ.	μεγ. πολ.
			οκτ.	επαν.	άλλο	οκτ.	επαν.	διάρκ.	άλλο			
D3-A5	98	11.1	0	0	0	40	40	16	8	204	2.5	4

Πίνακας 6.8: Πρελούδιο (Bach)

Εύρος νοτών	Σωστες νοτ.	Λάθος νοτ.	Νότες που δεν εντοπίστηκαν			Λάθος νότες				αρ. νοτ.	μεση πολ.	μεγ. πολ.
			οκτ.	επαν.	άλλο	οκτ.	επαν.	διάρκ.	άλλο			
D2-F#5	92.2	10.7	70.6	0	17.6	79.2	0	8.4	12.5	218	2.8	5

Πίνακας 6.9: Little thing called love (Queen)

Εύρος νοτών	Σωστες νοτ.	Λάθος νοτ.	Νότες που δεν εντοπίστηκαν			Λάθος νότες				αρ. νοτ.	μεση πολ.	μεγ. πολ.
			οκτ.	επαν.	άλλο	οκτ.	επαν.	διάρκ.	άλλο			
F2-G7	92.7	13.8	25	66.7	8.3	42.9	0	36.4	20.7	164	2.6	6

Πίνακας 6.10: Blues αυτοσχεδιασμός σε στυλ Boogie Woogie

Εύρος νοτών	Σωστες νοτ.	Λάθος νοτ.	Νότες που δεν εντοπίστηκαν			Λάθος νότες				αρ. νοτ.	μεση πολ.	μεγ. πολ.
			οκτ.	επαν.	άλλο	οκτ.	επαν.	διάρκ.	άλλο			
D2-G5	94.8	10.4	58.3	25	16.7	70.8	0	0	29.2	231	2.1	4

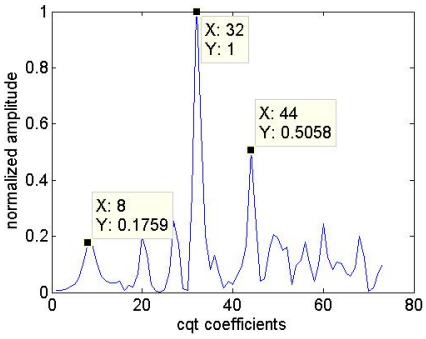
Πίνακας 6.11: Δρόμος (Λοϊζος)

Συνολικά, για την αξιολόγηση του αλγορίθμου χρησιμοποιήθηκαν 1939 νότες, εντοπίστηκαν 2099 νότες από τις οποίες οι 1860 ήταν σωστές. Έτσι τα precision και recall υπολογίζονται:

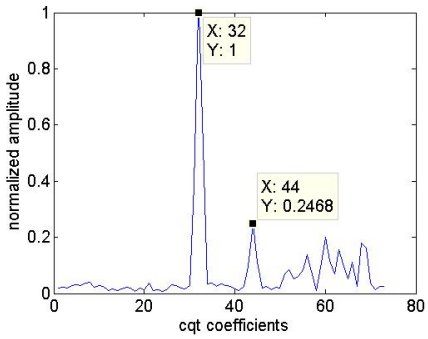
$$\text{precision} = \frac{1860}{1939} = 95.93\%$$

$$\text{recall} = \frac{1860}{2099} = 88.61\%$$

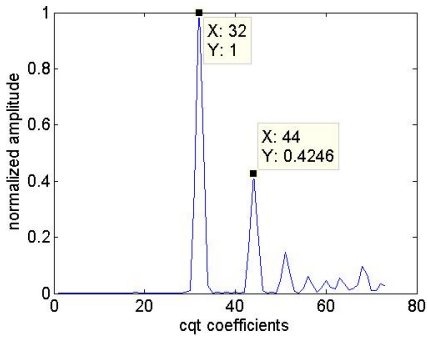
Συχνό πρόβλημα του αλγορίθμου αυτού είναι ότι εμφανίζονται αρκετές επιπλέον νότες. Οι λανθασμένες αυτές νότες, όταν δεν πρόκειται για λάθη διάρκειας ή για νότες που εσφαλμένα θεωρούνται ως επαναλαμβανόμενες νότες, συνήθως έχουν πολύ μικρή διάρκεια. Συγκεκριμένα, οι λανθασμένες νότες που ανήκουν στην πρώτη και τέταρτη κατηγορία όπως αυτές φαίνονται στην αντίστοιχη στήλη κάθε πίνακα, σε ποσοστό 81% έχουν διάρκεια περίπου 120msec (2 frames). Συχνότερα, όπως είναι αναμενόμενο, εμφανίζονται λάθη οκτάβας. Ένα μειωνέκτημα του αλγορίθμου το οποίο κάποιες φορές οδηγεί στην εμφάνιση τέτοιου ήδους λαθών είναι ότι σε κάθε επανάληψη του αλγορίθμου 5.2 επιλέγεται η εγγραφή του πίνακα $\mathbf{A}$ η οποία θεωρείται ως βέλτιστη εξετάζοντας βέβαια μόνο τις εγγραφές και τα αντίστοιχα βάρη που έχουν επιλεχθεί μέχρι εκείνη τη χρονική στιγμή. Επιλέγεται δηλαδή κάθε φορά η εγγραφή που είναι βέλτιστη αν δεν επιλεχθεί καμία άλλη νότα. Αυτό έχει σαν αποτέλεσμα οι πρώτες εγγραφές που επιλέγονται να είναι συνήθως πιο θορυβώδης γεγονός το οποίο μπορεί να επηρεάσει αρνητικά την επιλογή των επόμενων νοτών. Ένα παράδειγμα φαίνεται στο σχήμα 6.1. Στο σχήμα 6.1(α′) φαίνεται το διάνυσμα των συντελεστών CQT $\mathbf{X}$ που πρέπει να προσεγγιστεί με τα δεδομένα εκπαίδευσης. Οι νότες που ακούγονται στο δεδομένο χρονικό διάστημα είναι η G2 (τοπικό μέγιστο στο 8) και η F#5 (τοππικό μέγιστο στο 32). Στο σχήμα 6.1(β′) φαίνεται η πρώτη εγγραφή του πίνακα $\mathbf{A}$ που επιλέγεται. Όπως φαίνεται η συγκεκριμένη εγγραφή έχει χαμηλό amplitude στο συντελεστή 44 (πρώτη αρμονική της νότας F#5) συγκριτικά με την αντίστοιχη τιμή του $\mathbf{X}$. Έτσι σε επόμενη επανάληψη του αλγορίθμου 5.2 θα προστεθεί η νότα F#6) ώστε να αυξηθεί το amplitude του συγκεκριμένου συντελεστή. Αν είχε επιλεχθεί κάποια άλλη καταλληλότερη εγγραφή του $\mathbf{A}$ (υπενθυμίζεται ότι υπάρχουν 9 για κάθε νότα) όπως για παράδειγμα αυτή που φαίνεται στο σχήμα 6.1(γ′) θα είχε αποφευχθεί η προσθήκη αυτής της λανθασμένης νότας. Ωστόσο, το θορυβώδες τελείωμα της εγγραφής που φαίνεται στο 6.1(β′) την έκανε να φαίνεται καταλληλότερη.



(α′) X



(β′) Η πρώτη εγγραφή του A που επιλέγεται



(γ′) Μια καταλληλότερη εγγραφή του A

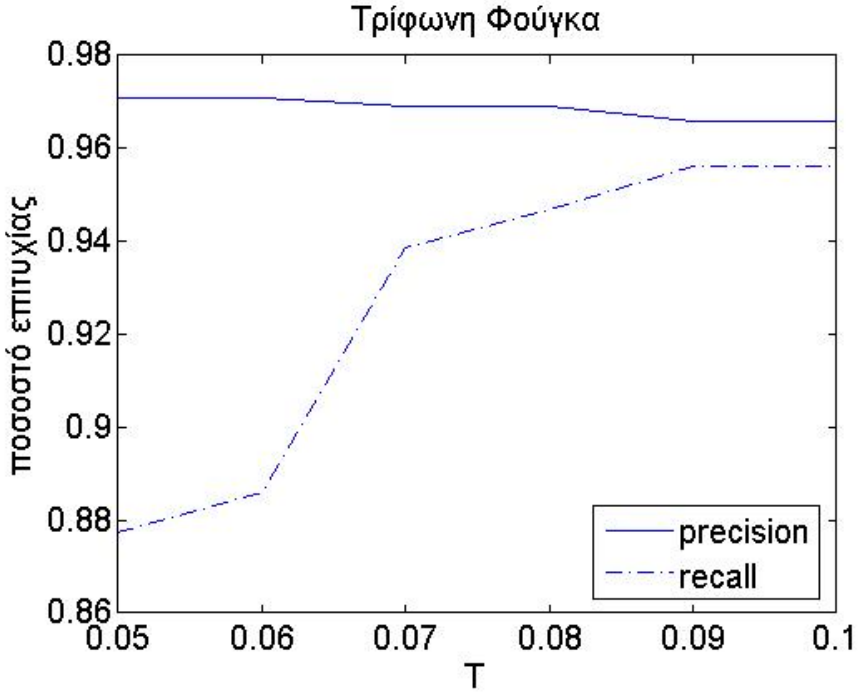
Σχήμα 6.1: Παράδειγμα εντοπισμού μιας λανθασμένης νότας.

Περισσότερα λάθη εμφανίζονται για νότες με μικρή βασική συχνότητα, χαμηλές νότες δηλαδή, οι οποίες είναι πιο θορυβώδεις. Εκεί οφείλονται και τα σχετικά υψηλά ποσοστά λανθασμένων νοτών που εμφανίζονται στα κομμάτια "Little thing called love" και "Δρόμος" τα οποία έχουν αρκετές χαμηλές νότες.

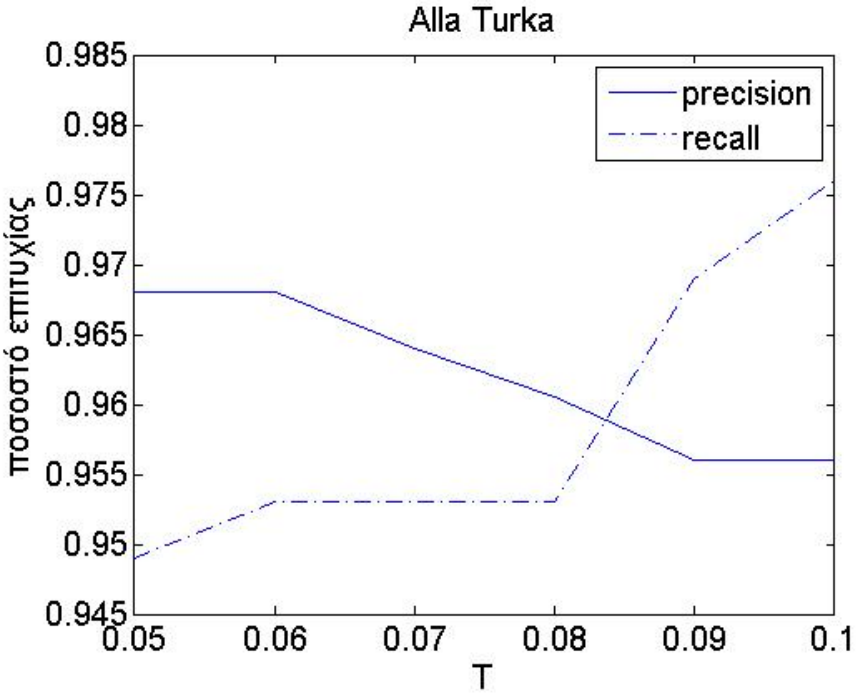
Λάθη επίσης εμφανίζονται συχνά και σε νότες με μεγάλη χρονική διάρκεια. Μια νότα με μεγάλη χρονική διάρκεια εξασθενεί στο τελευταίο μέρος της με αποτέλεσμα κάποιες φορές να εμφανίζεται ως διακεκομμένη. Όταν ο αλγόριθμος εντοπισμού διακεκομμένων νοτών που περιγράφεται στην ενότητα 5.2 αποτυγχάνει, η νότα εμφανίζεται ως επαναλαμβανόμενη. Εκεί οφείλεται το μεγάλο

ποσοστό των επαναλαμβανόμενων νοτών που εμφανίζονται στο πρελούδιο του Bach (40% των λανθασμένων νοτών) το οποίο περιέχει αρκετές νότες μεγάλης χρονικής διάρκειας. Επιπλέον, όταν μια νότα εξασθενεί, είναι ευκολότερο να γίνουν λάθη διάρκειας ή και άλλα λάθη. Όλες οι νότες της συγκεκριμένης σύνθεσης που δεν εντοπίστηκαν σωστά οφείλονται σε λάθη διάρκειας.

Τέλος, λάθη γίνονται και σε νότες πολύ μικρής διάρκειας όπως για παράδειγμα σε τρίλιες. Ο blues αυτοσχεδιασμός που παρουσιάζεται στον πίνακα 6.10 είχε αρκετές τρίλιες. Το μεγαλύτερο ποσοστό των νοτών του συγκεκριμένου κομματιού που δεν εντοπίστηκαν οφείλεται σε τρίλιες. Επιπλέον ο Onset Detector συνήθως αποτυγχάνει να εντοπίσει onsets για αυτές τις νότες με αποτέλεσμα να γίνονται και αρκετά λάθη διάρκειας ή λάθη επαναλαμβανόμενων νοτών.



(α′)



(β′)

Σχήμα 6.2: Παράδειγμα εντοπισμού μιας λανθασμένης νότας.

Το recall μπορεί να αυξηθεί σημαντικά σε κάποιες περιπτώσεις, προκαλώντας κάποια μείωση στο precision, αλλάζοντας το κατώφλι που έχει οριστεί στη σχέση 5.4:

$$||X - P^{(n-1)}|| - ||X - P^{(n)}|| > 0.06$$

Η σχέση αυτή καθορίζει το κατά πόσο η νέα νότα που προστίθεται βελτιώνει το αποτέλεσμα. Για να προστεθεί μια νέα νότα πρέπει το μέσο τετραγωνικό σφάλμα να μειώνεται τουλάχιστον κατά $T = 0.06$. Δοκιμάστηκαν διαφορετικές τιμές του T για τις συνθέσεις Alla Turka και για τη τρίφωνη φούγκα του Bach και υπολογίστηκαν τα precision και recall. Τα αποτελέσματα φαίνονται στο σχήμα 6.2. Το T μεταβάλλεται από 0.05 έως και 0.1 προκαλώντας καθώς αυξάνεται μείωση του precision έως και μία περίπου ποσοστιαία μονάδα και αύξηση του recall έως και δύο ποσοστιαίες μονάδας για τη σονάτα Alla Turka.

Αντίστοιχα για τη τρίφωνη φούγκα του Bach, το precision μειώνεται έως και μία ποσοστιαία μονάδα ενώ το recall αυξάνεται κατά επτά μονάδες.

Γενικότερα, η αύξηση του T επιτυγχάνει να μειώσει σημαντικά τα λάθη οκτάβας και άλλα λάθη που δεν οφείλονται σε εσφαλμένο υπολογισμό της διάρκειας ή σε λάθη επαναλαμβανόμενων νοτών. Οι λανθασμένες νότες οι οποίες αντιμετωπίζονται με την αύξηση του T είναι κυρίως νότες πολύ μικρής χρονικής διάρκειας, μικρότερης των 120mesc οι οποίες όπως αναφέρθηκε αποτελούν 81% των λαθών που δεν αφορούν λάθη διάρκειας ή επαναλαμβανόμενων νοτών. Έτσι η αύξηση του recall δεν αναμένεται να είναι τόσο έντονη σε συνθέσεις με υψηλό αριθμό λαθών που ανήκουν σε άλλες κατηγορίες, όπως για παράδειγμα στο πρελούδιο του Bach όπου εμφανίζεται μεγάλο ποσοστό λανθασμένων επαναλαμβανόμενων νοτών.

Κεφάλαιο 7

Συμπεράσματα

Στην εργασία αυτή μελετήθηκαν τεχνικές αναγνώρισης μονοφωνικής και πολυφωνικής μουσικής δίνοντας έμφαση στην ανάπτυξη αντίστοιχων αλγορίθμων. Υλοποιήθηκαν δύο αλγόριθμοι αναγνώρισης μονοφωνικής μουσικής των οποίων τα αποτελέσματα παρουσιάζονται συγκριτικά στο 6ο κεφάλαιο. Όπως φαίνεται από αυτά, η επεξεργασία του σήματος στο πεδίο του χρόνου και ειδικότερα η χρησιμοποίηση του Constant Q Transform όπως έγινε στον δεύτερο αλγόριθμο, δίνει αρκετά καλύτερα αποτελέσματα συγκριτικά με τον πρώτο αλγόριθμο όπου η επεξεργασία γίνεται στο πεδίο του χρόνου. Γενικότερα, η χρησιμοποίηση μιας αναπαράστασης του σήματος στο πεδίο των συχνοτήτων φαίνεται αποτελεσματικότερη καθώς επιτρέπει τον διαχωρισμό των συνιστωσών του. Επίσης, ο CQT πλεονεκτεί έναντι του μετασχηματισμού Fourier καθώς χρησιμοποιεί φίλτρα με γεωμετρικά τοποθετημένες συχνότητες οι οποίες ταυτίζονται με τις βασικές συχνότητες των νοτών της δυτικής μουσικής. Αυτό επιτρέπει την χρησιμοποίηση λιγότερων συντελεστών για την αναπαράσταση των δεδομένων. Επιπλέον, ο CQT παρέχει σταθερή ανάλυση σε όλες τις συχνότητες.

Ωστόσο, το πρόβλημα της αναγνώρισης μονοφωνικής μουσικής είναι ένα εύκολο πρόβλημα το οποίο αντιμετωπίζεται ικανοποιητικά και από τους δύο αλγορίθμους. Μεγαλύτερο τμήμα της εργασίας αυτής αποτελεί η ανάπτυξη ενός αλγορίθμου αναγνώρισης πολυφωνικής μουσικής. Ο αλγόριθμος αυτός αποτελεί μια επέκταση του 2ου αλγορίθμου αναγνώρισης μονοφωνικής μουσικής ώστε να υποστηρίζονται πολυφωνικά δεδομένα. Ο αλγόριθμος αυτός δε χρησιμοποιεί κάποιο πιθανοτικό μοντέλο αλλά δεδομένα εκπαίδευσης τα οποία χρησιμοποιούνται για την δημιουργία γραμμικού συνδυασμού που προσεγγίζει βέλτιστα το ζητούμενο σήμα. Συνήθως τέτοιου είδους αλγόριθμοι εφαρμό-

ζουν κάποια τεχνική μάθησης για τη δημιουργία του πίνακα των δεδομένων εκπαίδευσης. Ωστόσο, σε αυτή την εργασία τα δεδομένα εκπαίδευσης είναι εκ των προτέρων γνωστά. Δημιουργήθηκαν δεδομένα εκπαίδευσης για όλες τις νότες και σε διαφορετικές εντάσεις. Επιπλέον χρησιμοποιήθηκαν και δεδομένα εκπαίδευσης που αφορούν το attack μέρος μιας νότας. Με αυτόν τον τρόπο αντιμετωπίζεται σε μεγάλο βαθμό το πρόβλημα του θορυβώδους πρώτου μέρους των νοτών του πιάνου.

Τα αποτελέσματα που προέκυψαν από την εφαρμογή του αλγορίθμου σε ηχογραφήσεις από ηλεκτρονικό πιάνο ήταν ικανοποιητικά. Φυσικά, ο αλγόριθμος είναι κατάλληλος μόνο για αναγνώριση μουσικής πιάνου καθώς υπάρχουν δεδομένα εκπαίδευσης μόνο για πιάνο. Η εισαγωγή περισσότερων οργάνων απαιτεί τη δημιουργία δεδομένων εκπαίδευσης και για άλλα όργανα καθώς και ενός αλγορίθμου για την εύρεση του timbre δηλαδή την αναγνώριση του οργάνου. Επιπλέον, ίσως είναι απαραίτητες και κάποιες αλλαγές στον Onset Detector καθώς κάποια όργανα, όπως για παράδειγμα το βιολί, δεν έχουν τόσο έντονο attack μέρος όπως το πιάνο με αποτέλεσμα να είναι δυσκολότερος ο ακριβής εντοπισμός της χρονικής στιγμής στην οποία αρχίζει μια νότα.

Επίσης, οι παραπάνω αλγόριθμοι θα μπορούσαν να επεκταθούν ώστε να είναι δυνατή και εξαγωγή κι άλλων χαρακτηριστικών για μια νότα όπως είναι η έντασή της καθώς και άλλα χαρακτηριστικά που αφορούν την "έκφραση" της μουσικής (legato, staccato κτλ). Σημαντικό κομμάτι στην αναγνώριση μουσικής αποτελεί και η εξαγωγή του ρυθμού μιας σύνθεσης. Πληροφορίες για τη ρυθμική δομή μιας σύνθεσης μπορούν να προκύψουν από την περαιτέρω μελέτη και ανάλυση των αποτελεσμάτων του Onset Detector. Τέλος, η δημιουργία μιας πιο ολοκληρωμένης και φιλικής προς τον χρήστη εφαρμογής απαιτεί την υλοποίηση ενός γραφικού περιβάλλοντος.

Βιβλιογραφία

[1] Moorer, (1975). "On the Transcription of Musical Sound by Computer". Computer Music Journal, Nov. 1977.
[2] Martin, (1996). "A Blackboard System for Automatic Transcription of Simple Polyphonic Music". M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 385
[3] Dixon, Simon (2000) "On the computer recognition of solo piano music". in Proceedings of Australasian Computer Music Conference .
[4] Christopher Raphael, 2002. "Automatic Transcription of Piano Music". Proceedings of the International Conference on Music Information Retrieval.
[5] Matija Marolt, (2004). "A connectionist approach to automatic transcription of polyphonic piano music". IEEE transactions on multimedia, 2004.
[6] Martin Piszczalski and Bernard A. Galler, (1979). "Predicting musical pitch from component frequency ratios". The Journal of the Acoustical Society of America, 1979.
[7] Judith C. Brown, (1990). "Calculation of a constant Q spectral transform". Journal-Acoustical Society of America.
[8] R. Meddis and L. O"Mard, (1997). "A unitary model of pitch perception" The Journal of the Acoustical Society of America.
[9] Anssi Klapuri, (2004). "Signal Processing Methods for the Automatic Transcription of Music". Phd Thesis.
[10] Keith D. Martin, (1996). "A blackboard system for automatic transcription of simple polyphonic music". M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 385.
[11] Bello J.P and M.B. Sandler, (2000). "Blackboard Systems and Top-Down Processing for the Transcription of Simple Polyphonic Music". Proceeding of the COST G-6 Conference on Digital Audio Effects.

[12] M. Davy and S. J. Godsill, (2003) "Bayesian harmonic models for musical signal analysis". In Bayesian Statistics VII, J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith(Eds.), Oxford University Press.
[13]Paul J. Walmsley, Simon J. Godsill and Peter J. W. Rayner, (1999). "Polyphonic pitch tracking using joint Bayesian estimation of multiple frame parameters". Proc. 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York.
[14] Samer A. Abdallah and Mark D. Plumbley, (2004)."Polyphonic Music Transcription by Non-Negative Sparse Coding Of Power Spectra". Fifth International Conference on Music.
[15] Paris Smaragdis, Judith C. Brown, (2003). "Non-Negative Matrix Factorization for Polyphonic Music Transcription". IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.
[16] Judith C. Brown and Miller S. Puckette, (1992)."An efficient algorithm for the calculation of a constant Q transform". Journal-Acoustical Society of America.
[17] Anssi Klapuri,(1997). "Automatic Transcription of Music". Master Thesis, Tampere University of Technology.