



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αυτόματη Ανάλυση Ρυθμικής Συμπεριφοράς Μουσικών Εκτελέσεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ελευθέριος Σ. Σταμέλλος

Επιβλέπων : Γιώργος Καμπουράκης

Καθηγητής ΕΜΠ

Αθήνα, Ιούλιος 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αυτόματη Ανάλυση Ρυθμικής Συμπεριφοράς Μουσικών Εκτελέσεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ελευθέριος Σ. Σταμέλλος

Επιβλέπων : Γιώργος Καμπουράκης

Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24/07/2014.

.....
Γιώργος Καμπουράκης
Καθηγητής ΕΜΠ

.....
Βασίλειος Λούμος
Καθηγητής ΕΜΠ

.....
Ελευθέριος Καγιάφας
Καθηγητής ΕΜΠ

Αθήνα, Ιούλιος, 2014

.....
Ελευθέριος Σ. Σταμέλλος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός
Υπολογιστών Ε.Μ.Π.

Copyright © Ελευθέριος Σ. Σταμέλλος
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η παρούσα εργασία προτείνει μια μέθοδο αυτόματης ανάλυσης μουσικών εκτελέσεων, η οποία επιτυγχάνεται μέσω της αντιπαραβολής δύο αρχείων, μίας ψηφιακά επεξεργάσιμης παρτιτούρας και μίας ηχογράφησης κάποιας εκτέλεσης της παραπάνω παρτιτούρας. Συγκεκριμένα, ελέγχουμε αν ο εκτελεστής της ηχογράφησης τηρεί τις οδηγίες ρυθμικής αγωγής που του δίδονται πάνω στην παρτιτούρα. Σκοπός είναι να μπορούμε αυτόματα να εμπλουτίσουμε την παρτιτούρα εισόδου με καινούριες σημειώσεις που να υποδεικνύουν ρυθμικές διαφοροποιήσεις μεταξύ των δοθέντων αρχείων, αν κι όπου αυτές υπάρχουν.

Ξεκινάμε με την προεπεξεργασία των αρχείων η οποία έχει τους εξής βασικούς άξονες: τη σύνθεση της παρτιτούρας σε ήχο (δεδομένου ότι οι όποιες συγκρίσεις απαιτούν τα δύο αρχεία να παριστάνονται στην ίδια μορφή), την εξαγωγή των φασματικών ιδιοτήτων του προκύπτοντος αρχείου καθώς και της ηχογράφησης κάνοντας χρήση του Short Time Fourier Transform και, τελικά, την ανίχνευση βασικών χαρακτηριστικών της synthesized παρτιτούρας (λήψη των προαναφερθέντων ρυθμικών σημειώσεων και των χρονικών στιγμών έναρξης των θεμελιωδών ρυθμικών μονάδων της/beat).

Για να μπορούμε να αντιστοιχίσουμε κάθε χρονική στιγμή της ηχογράφησης με κάποιο σημείο εντός της παρτιτούρας, ευθυγραμμίζουμε χρονικά τις δύο προκύπτουσες φασματικές αναπαραστάσεις με χρήση της τεχνικής Dynamic Time Warping. Ακολουθεί η ποσοτικοποίηση της ρυθμικής συμπεριφοράς της ηχογράφησης μέσω του υπολογισμού της ταχύτητας δομικών μονάδων της (μέτρα) σε μονάδες Beats Per Minute, και η σύγκρισή της με την ταχύτητα εκτέλεσης που ζητείται, έμμεσα ή άμεσα, από τον εκτελεστή, μέσω των ρυθμικών σημειώσεων. Τελικά, καταλήγουμε στο βασικό ζητούμενο, στην παραγωγή μιας επαυξημένης παρτιτούρας, που θα είναι κατά βάση αυτή της εισόδου με την προσθήκη σημειώσεων / παρατηρήσεων για τα σημεία όπου υπάρχουν αποκλίσεις μεταξύ των εισόδων.

Όλα τα παραπάνω φτάνουν στο χρήστη μέσω ενός Graphic User Interface που έχει προγραμματιστεί στη Matlab κι εκμεταλλεύεται τον ψηφιακό επεξεργαστή μουσικής σημειογραφίας Musescore. Το GUI δέχεται δύο εισόδους (παρτιτούρα και ηχογράφηση), δίνει επιλογές για τρεις παραμέτρους που αφορούν τη Fourier ανάλυση και τελικά επιστρέφει σε μορφή .pdf την επαυξημένη παρτιτούρα.

Λέξεις - κλειδιά

Παρτιτούρα, σύνθεση ψηφιακής παρτιτούρας, scorewriter, Short Time Fourier Transform, μουσική σημειογραφία, MIDI, MusicXML, ευθυγράμμιση, Dynamic Time Warping, ρυθμός, tempo, Beats per Minute, γραφική διεπαφή χρήστη

Abstract

This paper proposes a method for automatic analysis of musical performances, achieved by comparing two files, a digitally editable music score sheet and a recording of a performance of the same score. More specifically, we verify if the performer of the recording follows the rhythmic instructions provided by the score. The aim is to be able to automatically enrich the input score with new notations indicating rhythmic differences between the two files, if and where they exist.

Firstly, we perform a pre-processing of the files which follows these basic axes: the synthesis of the score to sound (given that any comparison requires the two files to be represented in the same format), the export of spectral properties of the resulting file and the recording using Short Time Fourier Transform and ultimately the detection of key features of the synthesized score (taking the above rhythmic notations and onset of its fundamental rhythmic units/beats).

In order to match each recording point in time with a point in the score, we align the two resulting spectral representations using the Dynamic Time Warping technique. Next, we quantify the rhythmic behaviour of the recording by calculating the speed of its structural units (measures) in Beats Per Minute, and comparing it with the speed of execution requested by the performer, directly or indirectly, through rhythmic notations. Finally, we tackle our main question, the production of an augmented score, which corresponds to the input score with the addition of notations/comments on the points where there are discrepancies between the inputs.

This process is presented to the user via a Graphic User Interface programmed in Matlab which uses the digital scorewriter Musescore. The GUI accepts two inputs (score and recording), gives options for three parameters related to the Fourier analysis and finally returns the augmented score in .pdf format.

Keywords

Music score, score synthesis, scorewriter, Short Time Fourier Transform, musical notation, MIDI, MusicXML, alignment, Dynamic Time Warping, rhythm, tempo, Beats per Minute, graphical user interface

Περιεχόμενα

1. Εισαγωγή.....	9
1.1. Μουσική και μουσική σημειογραφία.....	9
1.2. Αντικείμενο διπλωματικής.....	17
1.3. Υπάρχουσα έρευνα.....	17
1.4. Οργάνωση κειμένου.....	19
2. Η παρτιτούρα.....	21
2.1. Εναλλακτικές Μορφές Scorewriter.....	21
2.2. Proprietary Software Scorewriter.....	22
2.3. FOSS Scorewriter.....	23
3. Η Ηχογράφηση.....	29
4. Preprocessing.....	31
4.1. Σύνθεση Παρτιτούρας σε Audio.....	31
4.1.1 MIDI.....	31
4.1.2 Από το MIDI σε Audio.....	37
4.1.3 Format Synthesized Παρτιτούρας.....	39
4.1.3.1 WAV /PCM.....	39
4.1.3.2 Vorbis OGG/Lossy Compression και Ψυχοακουστική.....	42
4.1.3.3 FLAC / Lossless Compression.....	52
4.2. Sample Rate Conversion	52
4.3. Downmix.....	53
4.4. Zero Padding.....	53
4.5. Normalization.....	53
4.6. Audio Feature Extraction.....	56
4.6.1 Σειρά Fourier – Μετασχηματισμός Fourier.....	56
4.6.2 DTFT – DFT.....	59
4.6.3 STFT.....	63
4.6.4 Spectrogram.....	65
4.6.5 Chromagram.....	68
4.7. Score Feature Extraction	71
4.7.1 MusicXML και DOM.....	71
4.7.2 Εξαγωγή Μουσικών Σημειώσεων/Οδηγιών.....	75
4.7.3 Beat onset extraction.....	80
4.7.3.1 Σταθεροποίηση διάρκειας beat synthesized παρτιτούρας.....	80
4.7.3.2 Beat onsets μέσω του MIDI.....	81
5. Alignment.....	82
5.1. Similarity Matrix.....	82
5.2. Lowest – Cost Path Calculation.....	84
5.3. Alignment Path.....	87
5.4. Warped Version Synthesis	88
6. Περιγραφή Εκτέλεσης.....	90
7. Αντιπαραβολή Εκτέλεσης - Παρτιτούρας.....	92
7.1. Κωδικοποίηση Σημειώσεων Παρτιτούρας.....	92
7.2. Just Noticeable Difference στο tempo.....	93

7.3. Σύγκριση μεταξύ bpm παρτιτούρας και ηχογράφησης - περιγραφή επαυξημένης παρτιτούρας.....	93
8. Δημιουργία επαυξημένης παρτιτούρας.....	98
9. Επίλογος: συμπεράσματα και προτάσεις για μελλοντικές επεκτάσεις	101
10. Βιβλιογραφία & Internet References.....	103
Παράρτημα 1.....	110
Παράρτημα 2.....	113
Παράρτημα 3.....	122

1. Εισαγωγή

A musical notation is a language which determines what you can say, what you want to say determines your language.

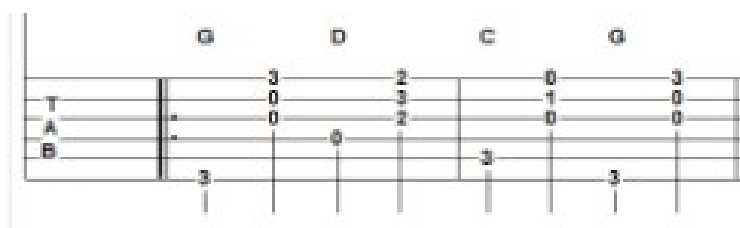
Cornelius Cardew -1961

1.1. Μουσική και μουσική σημειογραφία

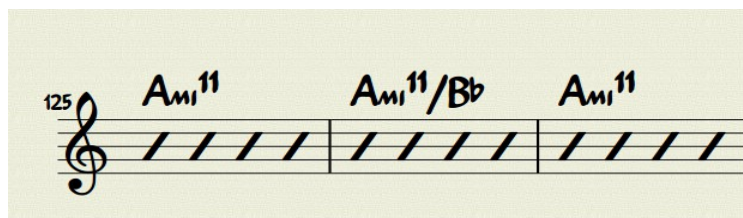
Μουσική και μουσική σημειογραφία πάνε μαζί, με έναν τρόπο που θυμίζει τη σχέση του προφορικού με το γραπτό λόγο. Όπως οι άνθρωποι μιλούσαν προτού δημιουργήσουν τις πρώτες συμβολικές γλώσσες, έτσι έπαιζαν και μουσική πριν αρχίσουν να τη γράφουν. Η ανάγκη να διατηρηθούν και να μεταδοθούν συνθέσεις και μουσικές παραδόσεις δρομολόγησε και τις πρώτες κωδικοποιημένες αναπαραστάσεις μελωδιών, με τα παλιότερα ευρήματα να χρονολογούνται από το 2000 π.Χ. [1],[2]. Από εκεί και ύστερα, συστήματα γραπτής απόδοσης της μουσικής συνέχισαν για χιλιάδες χρόνια να αναπτύσσονται σε όλο τον κόσμο, δίχως, κατά κανόνα να υπάρχει κάποια άμεση αλληλεπίδραση διαφορετικών συστημάτων.

Παράλληλα, η τέχνη της μουσικής συνέχισε να διαδίδεται και να αναπτύσσεται ακόμα κι όταν δε συνοδευόταν από τη γραπτή της εκδοχή. Προφορικές μουσικές παραδόσεις πολλών τόπων έχουν διατηρηθεί κι εξελιχθεί για χρόνια δημιουργώντας ένα ιστορικό (ενίοτε αδιάσπαστο, αν και μεταλλασσόμενο) συνεχές. Πολλοί μουσικοί μαθαίνουν και σήμερα να δουλεύουν την τέχνη τους χρησιμοποιώντας απλουστευτικές μεθόδους καταγραφής της (όπως η σημειογραφία με καθέτους, γνωστή ως slash notation ή οι ταμπλατούρες) ή και αποκλειστικά δια της ακοής.

a)



β)



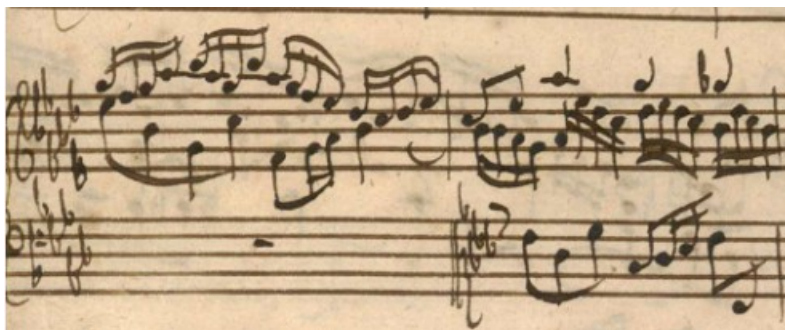
Εικόνα 1 Παραδείγματα αποσπασμάτων από κομμάτια αποτυπωμένα α) με ταμπλατούρα και β) slash notation.

Συχνά διατυπώνεται η πεποίθηση ότι όταν η συμβολική γραφή της μουσικής ακολουθεί συστήματα περιορισμένης εμβέλειας, ή, ακόμα περισσότερο, όταν απουσιάζει εντελώς, το ηχητικό αποτέλεσμα θα κινείται κι αυτό μέσα σε περιορισμένα περιθώρια, ακολουθώντας επαναλαμβανόμενους πεπατημένους δρόμους. Η διατύπωση αυτή, αν κι εκφράζει ένα σαφώς υποκειμενικό σκεπτικό, μπορεί κατά περίπτωση να απορριφθεί ή να επαληθευτεί. Αφενός, αγνοείται σε αυτή τη λογική ένας μεγάλος αριθμός από σπουδαιούς παίχτες που ακολουθούν σε διάφορες περιοχές την τοπική τους παραδοσιακή μουσική χωρίς να έχουν διδαχτεί κάποιο συμβολικό τρόπο απεικόνισής της. Πολλά τέτοια παραδείγματα συναντώνται και σε σύγχρονα μουσικά ρεύματα της λεγόμενης «εμπορικής» μουσικής. Ιδιαίτερα σε αυτοσχεδιαστικά αποσπάσματα τα οποία κρατάνε βασικό ρόλο στη jazz γλώσσα και μπορούν να αγγίξουν υψηλά μέτρα τεχνικών απαιτήσεων, η όποια σημειογραφία κατά κανόνα δεν είναι τη στιγμή της «δράσης» άμεσα χρήσιμη στους παίκτες κι ας είναι εξοικειωμένοι μαζί της [3]. Από την άλλη, είναι γεγονός πως όταν η σύνθεση, η ενορχήστρωση και η ίδια η εκτέλεση υπερβούν ορισμένα όρια, η παρουσία μίας γραπτής γλώσσας για τους μουσικούς γίνεται απολύτως απαραίτητη. Το κατεξοχήν παράδειγμα όπου συναντάται αυτή η απαίτηση είναι η έντεχνη δυτική μουσική ή όπως συνηθίζεται να ονομάζεται κλασική μουσική.¹

Δεν υπάρχει τρόπος να ορίσουμε την κλασική μουσική μονοσήμαντα. Οι παραδόσεις που έχουν να κάνουν με την ενορχήστρωση, τη φόρμα ή το ύφος των έργων που εμπίπτουν σε αυτή την κατηγορία έχουν με το χρόνο «απλωθεί» τόσο ώστε να περιλαμβάνουν ακραία διαφορετικές δημιουργίες.

Τελικά, το κεντρικότερο χαρακτηριστικό που μοιράζονται τα έργα που τοποθετούνται στην κλασική μουσική είναι η ίδια τους η σημειογραφία. Πρόκειται για τη γνωστή γραφή σε πεντάγραμμο, με νότες, αλλοιώσεις, κλειδιά κι άλλες πολλές σημειώσεις για τις επιμέρους λεπτομέρειες της εκτέλεσης.

α)

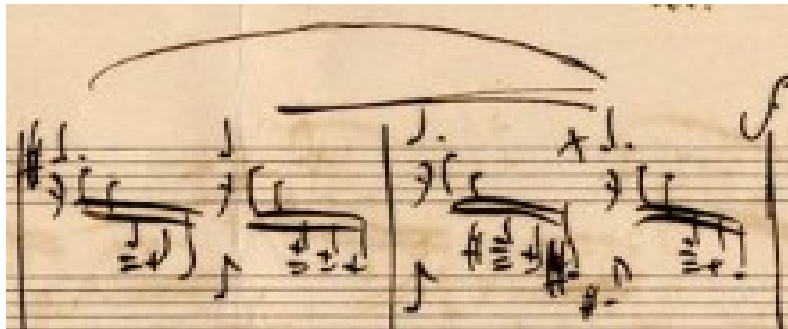


¹ Εδώ πρέπει να γίνει ο διαχωρισμός του ευρέος όρου “κλασική μουσική” και του πιο συγκεκριμένου “μουσική της κλασικής εποχής”, της εποχής δηλαδή που διήρκεσε λίγο – πολύ από το 1750 έως το 1820.

β)



γ)

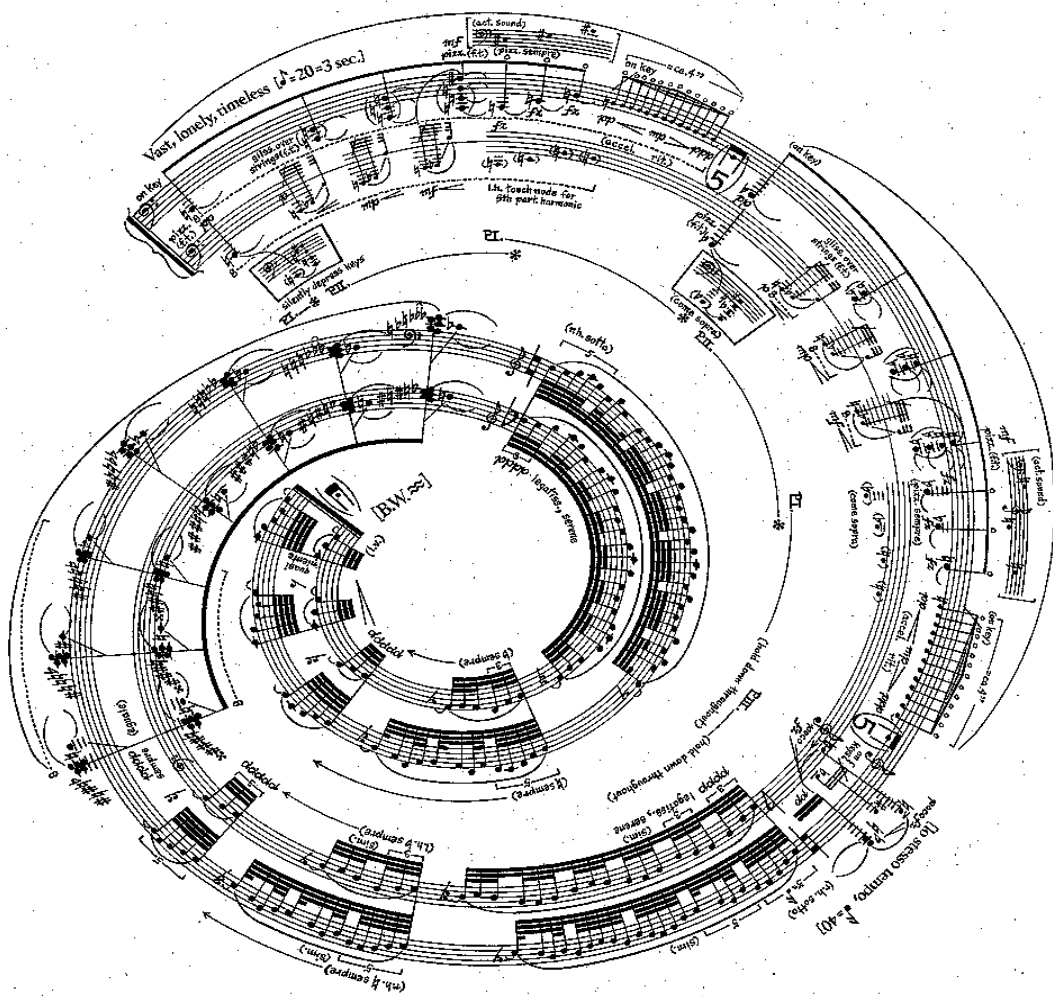


Εικόνα 2 Λεπτομέρειες τριών χειρόγραφων για έργα καθένα εκ των οποίων ανήκει σε διαφορετική περίοδο της κλασικής μουσικής: α) J.S.Bach, Φούγκα σε Λα ύφεση μείζονα, από το Καλοσυγκερασμένο Κλειδοκύμβαλο, 1744, Ύστερη Βαροκίε εποχή, β) Joseph Haydn, μέρος του Φαγκότο από τη συμφωνία αρ. 96 σε ρε μείζονα, Κλασική εποχή, γ) Johannes Brahms, Capriccio, Op. 76 No. 1, Ρομαντισμός

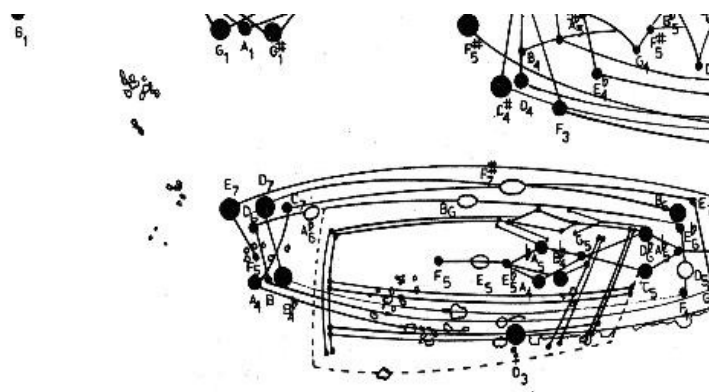
Η γραφή των νοτών πάνω και γύρω από γραμμές εισήχθη πρώτη φορά από τον μοναχό Guido d'Arezzo (990-1050 μ.Χ.) , η οριστική επικράτηση του πενταγράμμου επί των άλλων αναπαραστάσεων έρχεται περί τα 1600 [4]. Στην ίδια περίπου εποχή τοποθετείται και η έναρξη της Βαροκίε περιόδου που δίνει το έναυσμα σε μια ακολουθία τριών αιώνων κατά τη διάρκεια των οποίων γράφτηκαν τα σημαντικότερα έργα που ουσιαστικά καθορίζουν την κλασική μουσική.

Είναι απολύτως δίκαιο να πούμε πως η τυπική μουσική σημειογραφία πάνω στο πεντάγραμμο με τη διάδοσή της βοήθησε τη μουσική να περάσει σε νέα επίπεδα αρτιότητας. Είναι τόσο μεγάλη η σημασία της ως μέσο που, τελικά, έφτασε να θέτει σε ένα βαθμό και τα όρια στο αποτέλεσμα.

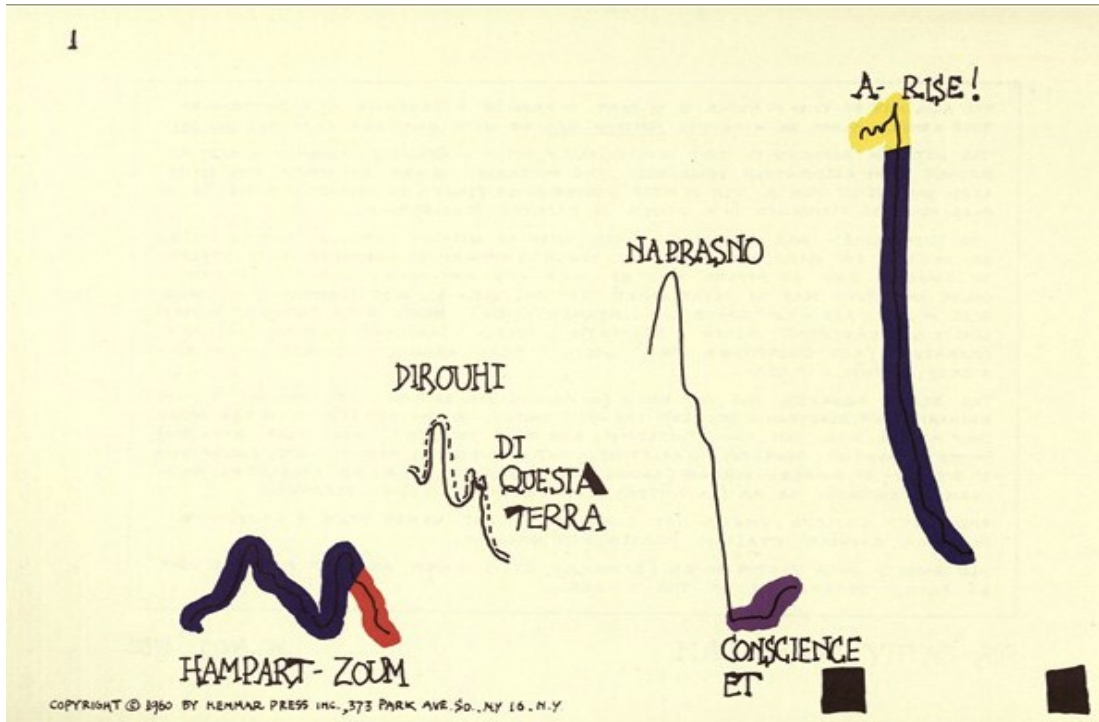
Είναι επόμενο, βέβαια, κάποιες στιγμές αυτό να δημιουργεί περισσότερους περιορισμούς από όσα προβλήματα λύνει. Τρανταχτή απόδειξη είναι το γεγονός πως, με το πέρασμα της μουσικής στη σύγχρονη εποχή (που οι απαρχές της ορίζονται γύρω στα 1900), εμφανίζονται καλλιτέχνες μέσα από τους κύκλους της έντεχνης παιδείας και δημιουργίας οι οποίοι παραγκωνίζουν βασικές συμβάσεις της κλασικής μουσικής και μαζί με αυτές την κλασική σημειογραφία.



Εικόνα 3 George Crumb, Spiral Galaxy , από το Makrokosmos Volume 1, 1972



Εικόνα 4 George Cacioppo, Λεπτομέρεια από την Cassiopeia, 1962



Εικόνα 5: John Cage. Λεπτομέρεια από την Aria. 1960

Γραφικές παρτιτούρες σαν τις παραπάνω σηματοδοτούν μία σαφή παρέκκλιση από την συνηθισμένη γραφή της μουσικής. Το μέσο εδώ είναι ταυτόχρονα το εργαλείο κι ο σκοπός: τέτοιου τύπου έργα έχουν μία εικαστική αξία που θέλει να εκδηλωθεί και με τη μουσική, την ίδια στιγμή, οι δημιουργοί τους, συνειδητοποιώντας την ανεπάρκεια των υπαρκτών μέσων, εισάγουν νέους τρόπους για την απόδοση των ολότελα πρωτοπόρων συνθέσεων τους.

Παρόλο που η εικόνα προκαλεί, δεν πρόκειται για κάποια ιδιαίτερη τομή στην ιστορία, ένα καπρίτσιο κάποιων φαντασμένων ή μία παροδική τάση πειραματισμού. Τα παραπάνω έργα (ανεξάρτητα από την υποκειμενική καλλιτεχνική τους αξία) εξυπηρετούν μία ανάγκη – κοινό τόπο στη δημιουργία καθ' όλη την ιστορία: την ώθηση των γνώριμων μέσων πέρα από τα εγκαθιδρυμένα όριά τους, προς υλοποίηση μίας ιδέας που, αν και θέλει να σταθεί κι έξω από το φαντασιακό του ενός, αδυνατεί να τα καταφέρει αντλώντας μόνο από την υπάρχουσα σοφία των πολλών.

Ο Erik Satie, το 1893, γύρω στο ξεκίνημα της εποχής που εκ των υστέρων θα οριζόταν ως σύγχρονη και μερικές δεκαετίες πριν την πρώτη εμφάνιση τη γραφικής παρτιτούρας, εξέδωσε τις πρώτες του Gnossiennes. Τα κομμάτια αυτά, γραμμένα για σόλο πιάνο, δεν είχαν μέτρα. Ο Satie αρνήθηκε αυτόν το θεμέλιο κανόνα της σύνθεσης και της μουσικής σημειογραφίας, προκειμένου να σχηματίσει ήχους άχρονους, ένα σύμπλεγμα νοτών που, αν κι έχει αρχή και τέλος στο χαρτί, φαντάζει για τον ακροατή άπειρο. Αυτή η συνθετική και σημειογραφική

«παράβαση» του ήταν απαραίτητη για να συνθέσει τις Gnossiennes, έργα απλούστατα και σύντομα που έχουν καθιερωθεί ως διαχρονικά δημιουργήματα μεγάλης αξίας.

№ 1



Εικόνα 6: Erik Satie, η πρώτη σειρά από τη Gnossienne no1, 1893

Η αφαιρετική διάθεση του Satie παρότρυνε την εκτελεστική ελευθεριότητα σε ό,τι αφορά τον χρόνο. Νωρίτερα, στις αρχές του 19ου αιώνα, η εφεύρεση του μετρονόμου ήρθε να ικανοποιήσει την απαίτηση για το ακριβώς αντίθετο αποτέλεσμα, την απόλυτη ρυθμική πειθαρχία. Οι συνθέτες είχαν έως τότε μόνο ένα μέσο να διατυπώσουν την επιθυμητή ταχύτητα, μία σειρά από ασαφείς έννοιες όπως 'allegro' (γρήγορα, κυριολεκτικά χαρούμενα ή και ανάλαφρα), 'andante' (αργά, κυριολεκτικά σαν αργό περπάτημα), 'adagio' (αργά, κυριολεκτικά με άνεση), 'presto' (γρήγορα), κ. ο. κ. Χάρis στο μετρονόμο ο ρυθμός μπορεί να καταγραφεί με απόλυτους όρους, κάνοντας χρήση της μονάδας bpm (beats per minute). Ο πρώτος που χρησιμοποίησε το μετρονόμο για να επιδείξει την ταχύτητα του έργου του ήταν ο (ιδιαίτερα ενθουσιασμένος με τη νέα εφεύρεση [5]) Beethoven στη σονάτα του για πιάνο no.29, Op.106, γνωστή κι ως Hammerklavier sonata. Πλέον, είναι απολύτως αναμενόμενο οι παρτιτούρες να συνοδεύονται από μετρονομικές ενδείξεις.



Εικόνα 7 Ludwig Van Beethoven, τα πρώτα μέτρα από την Hammerklavier sonata, 1818. Δίχως το μετρονόμο, πάνω αριστερά θα υπήρχε μόνο η ένδειξη Allegro

Το μόνο που παρέχει ο μετρονόμος είναι ένας σταθερός, μονότονος ήχος που υπενθυμίζει απαρέγκλιτα στον εκτελεστή την πάροδο ισόχρονων διαστημάτων. Δε ρυθμίζεται ώστε να προβλέπει αυξομειώσεις στο tempo, πόσο μάλλον έννοιες που σχετίζονται με το χρόνο αλλά συχνά εμπίπτουν

στη σφαίρα του αφηρημένου, όπως η παλμική αίσθηση ή ο αρμονικός ρυθμός, που συνήθως δεν καταγράφονται ρητά αλλά υπονοούνται. Εξάλλου, η ταχύτητα εκτέλεσης μπορεί να υποδηλώνεται μέσω κάποιο όρου που αναφέρεται ουσιαστικά στη διάθεση της εκτέλεσης (πχ *animato* που κυριολεκτικά σημαίνει ζωηρά κι έντονα, συνεπάγεται υψηλές ταχύτητες). Εν πάση περιπτώσει η επέκταση του λεξιλογίου για το tempo ήταν μια διαδικασία διαρκής αν και, υποχρεωτικά, για πρακτικούς λόγους, περιορισμένη. Αυτό όμως σε καμία περίπτωση δε συνεπάγεται πως οι μουσικοί του 18ου αιώνα, για παράδειγμα, δεν μεταχειρίζονταν το ρυθμό με την άνεση των ρομαντικών.

Ομοίως και με τις δυναμικές: οι όροι που αναφέρονται στην ένταση της εκτέλεσης, όπως πχ *piano* (απαλά), *forte* (δυνατά), *più piano* (πιο απαλά), χρησιμοποιούνταν σποραδικά μέχρι τα τέλη του 18ου αιώνα. Είναι βέβαιο, όμως, ότι οι μουσική της Baroque εποχής (περίπου 1600 έως 1750) απαιτούσε την αδιάκοπη εναλλαγή του δυνατού και του απαλού [6]. Επιπλέον, κανένας σύγχρονος εκτελεστής δε θα ερμήνευε «επίπεδα» ένα έργο του Bach στηριζόμενος στην απουσία των σημειώσεων τις οποίες έχει συνηθίσει να συναντά σε μεταγενέστερες παρτιτούρες. Πέραν όλων αυτών, οι όροι της σημειογραφίας γύρω από τις δυναμικές είναι σχετικοί και δεν υποδεικνύουν κάποιο απόλυτο μέγεθος έντασης. Το πόσο δυνατά θα αποδοθεί ένα πέρασμα σημειωμένο με *forte* ή πόσο απαλότερος θα γίνει ο ήχος όταν συναντάται ένα *diminuendo* εξαρτάται από τα γενικότερα επίπεδα έντασης του έργου, από την εποχή που γράφτηκε, από το όργανο στο οποίο εκτελείται, από την αίθουσα, από τον τρόπο που επηρεάζουν όλοι οι προηγούμενοι παράγοντες τους υπόλοιπους μουσικούς οι οποίοι παίζουν ταυτόχρονα εάν πρόκειται για μουσική γραμμένη για σύνολο κι άλλα πολλά ακόμα.

a)

The image shows a snippet of a musical score. At the top left, the tempo marking **molto** is written. The score consists of three staves. The top staff contains a melodic line with several notes, including a half note and a quarter note, with a **forte** dynamic marking. The middle staff is mostly empty, with a few notes. The bottom staff contains a bass line with a few notes. The score ends with a double bar line.

β)



Εικόνα 8 Ακραία παραδείγματα από τον ίδιο συνθέτη: α) rrrrrrr, από την 6η συμφωνία του Tchaikovsky, β) ffff από την Ουβερτούρα 1812 του ίδιου. Πηγή εικόνας: Petrucci Music Library

Σε τελευταία ανάλυση, αυτός που έχει και τον τελευταίο λόγο στον ακριβή τρόπο απόδοσης κάθε ενός έργου είναι ο ίδιος ο μουσικός που το παίζει. Το ίδιο έργο μπορεί να επιδέχεται πολλές ερμηνείες κι αυτό είναι κάτι που γνωρίζουν καλά οι συνθέτες, οι εκτελεστές και οι ακροατές. Είναι απολύτως αναμενόμενο το ενδεχόμενο, λοιπόν, τα ίδια σύμβολα της μουσικής σημειογραφίας να επιφέρουν κατά περίπτωση πολύ διαφορετικά ηχητικά αποτελέσματα.

Όλα τα παραπάνω οδηγούν σε ένα βασικό συμπέρασμα: **η κλασική μουσική σημειογραφία είναι (σχεδόν πάντα) απαραίτητη και (σχεδόν πάντα) ανεπαρκής**. Η επιθυμία για την επίτευξη περαιτέρω σύγκλισης μεταξύ της γραπτής μουσικής απόδοσης και των αστάθμητων χαρακτηριστικών της ανθρώπινης ερμηνείας είναι, λοιπόν, το κίνητρο για την παρούσα εργασία.

1.2. Αντικείμενο διπλωματικής

Η κεντρική ιδέα είναι η δημιουργία μίας προγραμματιστικής εφαρμογής που θα λαμβάνει ένα **μουσικό κείμενο (παρτιτούρα)** μαζί με μια **ηχογραφημένη εκτέλεσή του** και θα προχωρά στην **αντιπαραβολή** τους.

Η αντιπαραβολή συνίσταται στη **χρονική ευθυγράμμιση** των δύο αρχείων εισόδου, στον **αυτόματο εντοπισμό σημείων της παρτιτούρας με σημειώσεις** που αφορούν στο tempo και τελικά τη **σύγκριση** των σημείων αυτών με τα αντίστοιχα της ηχογράφησης.

Από την επεξεργασία αυτή εξάγουμε ένα νέο κείμενο, το οποίο στην ουσία θα είναι η **αρχική παρτιτούρα εμπλουτισμένη με νέες πληροφορίες** οι οποίες θα αποτυπώνουν το κατά πόσο ο εκτελεστής της ηχογράφησης ήταν «πιστός» στην παρτιτούρα.

1.3. Υπάρχουσα έρευνα

Η έρευνα γύρω από την αυτοματοποίηση της ανάλυσης μουσικών

ερμηνειών βρίσκεται σε σταθερή ανάπτυξη, έχοντας να επιδείξει σημαντικό όγκο βιβλιογραφίας κι εφαρμογών. Ένα σχετικό παράδειγμα εφαρμογής λογισμικού είναι η εργαλειοθήκη AMPACT (Automated Music Performance Analysis and Comparison Toolkit [7]) για το περιβάλλον της Matlab. Αντικείμενό της είναι η αυτόματη ανάλυση ηχογραφημένων ερμηνειών με εκμετάλλευση των δεδομένων που παρέχονται από τις αντίστοιχες παρτιτούρες.

Αναφέρουμε παρακάτω ενδεικτικά ορισμένες σημαντικές μελέτες. Επελέγησαν συγκεκριμένα αυτές πάνω τις οποίες στηρίχτηκε σε μεγάλο βαθμό η παρούσα εργασία.

Software-based extraction of objective parameters from music performances [8], διδακτορική διατριβή του **Alexander Lerch** για το Technische Universität Berlin. Η έρευνα αυτή παρουσιάζει μία προσέγγιση στην προγραμματιστική εξαγωγή χαρακτηριστικών μουσικών παραμέτρων (tempo, timing, ακουστότητα - βλ. παράγραφο 4.1.3.2 - ηχόχρωμα και ένταση) από αρχεία ήχου. Σκοπός είναι η δημιουργία ενός εργαλείου αυτοματοποίησης της διαδικασίας αυτής.

Extracting expressive performance information from recorded music [9], μεταπτυχιακή διπλωματική εργασία του **Eric David Scheirer** για το Massachusetts Institute of Technology. Στη μελέτη αυτή περιγράφεται μία διαδικασία (στην οποία δίδεται η ονομασία *expressive performance extraction, εξαγωγή εκφραστικής ερμηνείας*) η οποία αντιστοιχεί μία ψηφιακή παράσταση κάποιας μουσικής ερμηνείας σε MIDI αρχείο (βλ. παράγραφο 4.1.1) με οδηγό την παρτιτούρα του εκτελούμενου έργου.

Information Retrieval for Music and Motion [10] του **Meinard Müller**, εκδόσεις Springer. Από το συγκεκριμένο βιβλίο βοήθηθήκαμε κυρίως από το τέταρτο κεφάλαιο, αναφορικά με την τεχνική Dynamic Time Warping.

Επιπλέον, στον τομέα της ψυχοακουστικής (βλ. παράγραφο 4.1.3.2): *Music Compression Algorithms and Why You Should Care [11]*, εργασία του **Alexander Benjamin** για το Washington University in St. Louis, Department of Electrical and Systems Engineering

Audio Signal Processing and Coding [12] των **Andreas Spanias, Ted Painter, Venkatraman Atti**, εκδόσεις John Wiley & Sons.

Σημειώσεις του καθηγητή **Rod Nave [13]**, Georgia State University, Department of Physics and Astronomy

Για το ζήτημα του Just Noticeable Difference στο tempo: *Just Noticeable Difference and Tempo Change [14]*, δημοσίευση του **Kim Thomas**, University of Central Oklahoma, στο Journal of Scientific Psychology, Μάιος 2007.

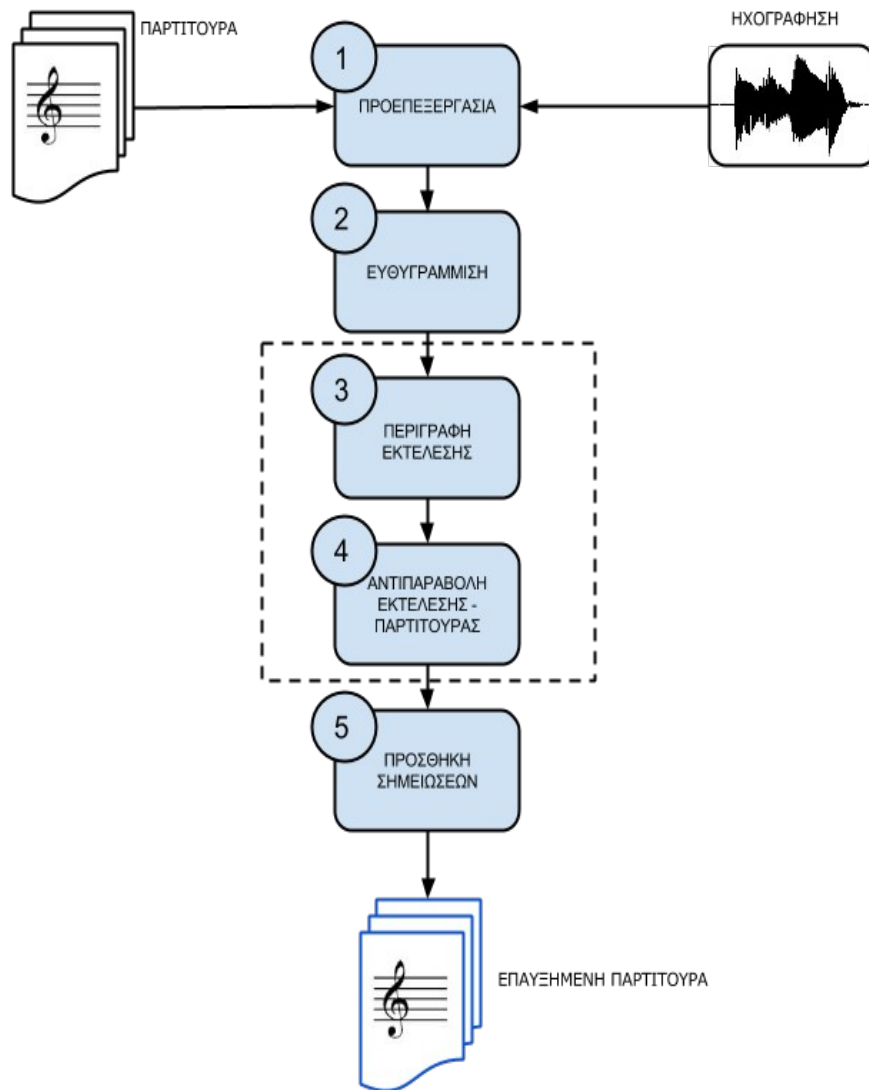
Music Perception [15], M.R. Jones et al. , εκδόσεις Springer, *Chapter 6, Tempo and Rhythm*, του **J. Devin McAuley**.

Ειδική μνεία πρέπει να γίνει στον **Alf Gabrielsson** [16] του οποίου το έργο βελτίωσε σημαντικά την κατανόηση των μουσικών φαινομένων με γνώμονα τις παρελκόμενες διεργασίες του νου καθώς και στον **Dan Ellis** [17] για την εκτεταμένη έρευνά του στο πεδίο της ψηφιακής επεξεργασίας σήματος στο κομμάτι που σχετίζεται με την επεξεργασία μουσικών ερμηνειών.

Εκτενής βιβλιογραφία μπορεί για τα παραπάνω ερευνητικά πεδία μπορεί να βρεθεί και στο link βιβλιογραφίας του AMPACT [18].

1.4. Οργάνωση κειμένου

Το κείμενο ακολουθεί λίγο – πολύ τη δομή του κώδικα της εφαρμογής η οποία το συνοδεύει. Στα κεφάλαια 2 και 3 ασχολούμαστε με τα δύο αρχεία εισόδου, την παρτιτούρα και την ηχογράφιση. Στο κεφάλαιο 4 επικεντρωνόμαστε στις εργασίες προεπεξεργασίας των δύο προαναφερθέντων αρχείων, σε όλα τα βήματα, δηλαδή, που πρέπει να προηγηθούν ώστε να εξασφαλίσουμε τα απαιτούμενα στοιχεία για την καθαυτή σύγκριση τους. Στο κεφάλαιο 5 περιγράφεται η διαδικασία της χρονικής αντιστοίχισης της ηχογραφημένης εκτέλεσης με τη (συντεθειμένη σε ήχο πλέον) παρτιτούρα. Στο 6 περνάμε στην ανάλυση της μεθόδου με την οποία παράγουμε μία μετρήσιμη, κατανοητή από τον υπολογιστή περιγραφή του ρυθμού της εκτέλεσης. Στο κεφάλαιο 7 καταγράφεται η διαδικασία και τα κριτήρια με τα οποία συγκρίνουμε τις δύο εισόδους. Το 8 αναφέρεται στον τρόπο με τον οποία δημιουργείται η τελική έξοδος / επαυξημένη παρτιτούρα του συστήματός μας. Το κεφάλαιο 9 περιέχει τα τελικά συμπεράσματα της εργασίας καθώς και προτάσεις για δυνάμει περαιτέρω επεκτάσεις. Τέλος, στο 10 περιέχονται διαδικτυακές αναφορές και βιβλιογραφία. Περιλαμβάνονται επιπλέον και τρία παραρτήματα: το πρώτο έχει screenshots του GUI της εφαρμογής μας, το δεύτερο ένα παράδειγμα χρήσης του και το τρίτο τον κώδικα της εφαρμογής.



Εικόνα 9 Διάγραμμα – περίληψη της εργασίας

2. Η παρτιτούρα

Ο όρος παρτιτούρα (εκ του ιταλικού *partitura*) αναφέρεται στην «τυπωμένη μουσική», σε αντιπαράθεση με την ηχογραφημένη ή την ζωντανή μουσική ερμηνεία. Η λέξη τυπωμένη χρησιμοποιείται εδώ καταχρηστικά, μπορεί η μουσική να αποδίδεται και στην οθόνη του υπολογιστή. Σε κάθε περίπτωση, όμως, θα πρέπει να μπορεί να γίνεται αντιληπτή από μουσικούς. Εφαρμογές που επιτρέπουν την δημιουργία ψηφιακά επεξεργάσιμων μουσικών κειμένων, οι ονομαζόμενοι **scorewriters**, έκαναν την εμφάνιση τους ήδη από την δεκαετία του 80 [19]. Έκτοτε, ο χώρος του musical notation software αναπτύσσεται σταθερά, όπως συμπεραίνεται από την ποσότητα και τις δυνατότητες των διαθέσιμων scorewriters.

2.1. Εναλλακτικές Μορφές Scorewriter

Οι διαρκείς τάσεις ριζοσπαστικοποίησης της μουσικής σύνθεσης τα τελευταία περίπου 100 χρόνια (και ειδικά με την ευρεία καθιέρωση της ηλεκτρονικής μουσικής) έχουν φέρει στο προσκήνιο μία σειρά από αναπτυσσόμενες καινοτομίες που βοηθούν προς αυτή την κατεύθυνση, αντικαθιστώντας ή συμπληρώνοντας τις συνηθισμένες παρτιτούρες, δίνοντας νέες διεξόδους σε συνθέτες, μουσικούς εκτελεστές αλλά και άλλων τύπων ερμηνευτές [20]. Μερικά τέτοια ενδεικτικά παραδείγματα είναι τα εξής:

- **Antescofo [21]**, ένα εργαλείο που μπορεί σε πραγματικό χρόνο να εντοπίζει τη θέση του εκτελεστή «πάνω στην παρτιτούρα», καθιστώντας δυνατό το συγχρονισμό της ερμηνείας με στοιχεία που έχουν παραχθεί από τον υπολογιστή. Είναι εφικτή, έτσι, η αυτόματη συνοδεία μουσικών σε ζωντανή εκτέλεση (κάτι που θα μπορούσε να εξυπηρετήσει και εκπαιδευτικά), η διαδραστική ηλεκτρονική μουσική σύνθεση κ.α.
- **Acousmograph [22]** και **Eanalysis [23]**, προγράμματα που επιτρέπουν την γραφική αναπαράσταση ηλεκτρονικής μουσικής.
- **INScore [24]**, ένα περιβάλλον για τη σχεδίαση Διαδραστικών Επαυξημένων Μουσικών Κειμένων (Interactive Augmented Music Scores).

Παρόλα αυτά, η μεγάλη πλειοψηφία των μουσικών συνεχίζει να ερμηνεύει με βάση την παρτιτούρα στη συνηθισμένη της μορφή και κατ' επέκταση ο αριθμός των προγραμμάτων που επικεντρώνονται στην κατασκευή τους είναι σαφώς μεγαλύτερος από εναλλακτικές σαν τις παραπάνω.

Παρακάτω επιχειρούμε μία σύντομη επισκόπηση των δημοφιλέστερων scorewriters, διαχωρίζοντάς τους καταρχήν με βάση τον τρόπο διανομής τους.²

2.2. Proprietary Software Scorewriter

Τα προγράμματα **Sibelius [25]** και **Finale [26]** είναι κατά κάποιο τρόπο οι «βετεράνοι» του χώρου. Πρόκειται για τις μόνες εμπορικές εφαρμογές αυτού του τύπου που συνεχίζουν αδιάλειπτα την πορεία τους από της αρχές της δεκαετίας του 90. Για πολλά χρόνια και μέχρι πολύ πρόσφατα ανταγωνίζονταν ουσιαστικά μόνο μεταξύ τους για το μεγαλύτερο μερίδιο της αγοράς. Συνεχίζουν να έχουν την αναγνώριση και την επίσημη υποστήριξη πολλών μουσικών σχολών ανά τον κόσμο.

Πάντως, οι εξελίξεις στο ελεύθερο λογισμικό και ορισμένες ανακατατάξεις στο εσωτερικό των εταιριών που παράγουν τα προϊόντα αυτά αφήνουν τους τελευταίους μήνες τη διάχυτη εντύπωση ότι οι ισορροπίες στην αγορά των scorewriters σύντομα θα αλλάξουν [27].

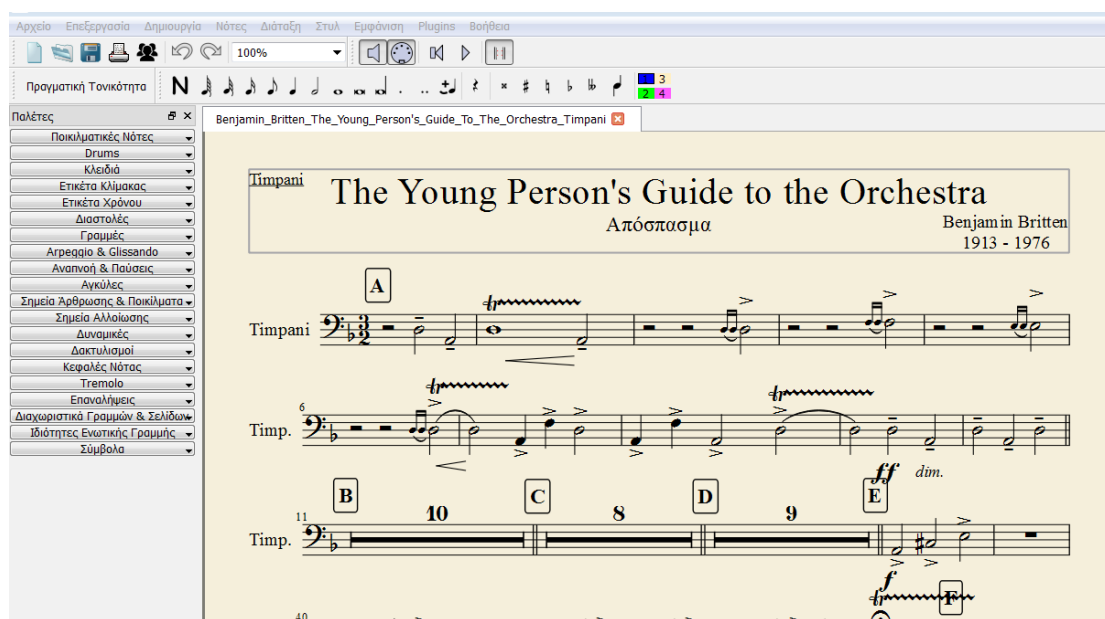
Μερικά γνωστά αλλά λιγότερα διαδεδομένα proprietary προγράμματα είναι το **Encore [28]**, το **Guitar Pro [29]** (που απευθύνεται προπάντων σε κιθαρίστες), το **Maestro Notation [30]** για το MS Word (με το οποίο μπορεί κανείς να εισάγει μουσικό κείμενο απευθείας σε ένα αρχείο κειμένου του MS Word), το **MUS2 [31]** (ειδικευμένο σε συνθέσεις βασισμένες σε μακάμια), κτλ.

Αν και επιλογές όπως η παραπάνω έχουν και τις μεγαλύτερες δυνατότητες, προτιμήθηκε η χρήση **εφαρμογών ελεύθερου λογισμικού (Free and Open – Source Software)**. Αν κι ενδέχεται έτσι να βρεθούμε να χρησιμοποιούμε ένα scorewriter που μειονεκτεί ποιοτικά σε ορισμένους, τουλάχιστον, τομείς, κερδίζουμε την ελευθερία επέμβασης στο λογισμικό, προκειμένου να το μελετήσουμε ή ακόμα και να το τροποποιήσουμε. Επιπλέον, ο κώδικας που συνοδεύει την εργασία μπορεί να διατεθεί στην κοινότητα των χρηστών του λογισμικού προς χρήση, βελτίωση και σχολιασμό. Εξάλλου, ένα επιπλέον κίνητρο για αυτή την επιλογή είναι ότι, όπως ομολογεί και το μεγαλύτερο ποσοστό των χρηστών, οι τιμές των proprietary scorewriters πολύ συχνά κρίνονται υπερβολικές, δεδομένης της πολιτικής τιμολόγησης που ακολουθείται σε άλλου τύπου εφαρμογές. Παρατηρείται, λοιπόν, μία στροφή σοβαρού μεριδίου της αγοράς προς κάποια λύση ελεύθερου λογισμικού, τάση η οποία, όπως είναι λογικό, ωθεί ταυτόχρονα στην ολοένα και ποιοτικότερη εξέλιξη των FOSS προϊόντων.

² Πέραν των scorewriters, μπορεί κανείς να δημιουργήσει και να επεξεργαστεί παρτιτούρες και σε ορισμένα Digital Audio Workstations, όμως με αρκετά πιο περιορισμένες, συνήθως, δυνατότητες.

2.3. FOSS Scorewriter

Για την παρούσα εργασία, επελέγη σαν ιδανικότερη λύση το **musescore** [32].



Εικόνα 10 Snapshot του musescore, έκδοση 1.3

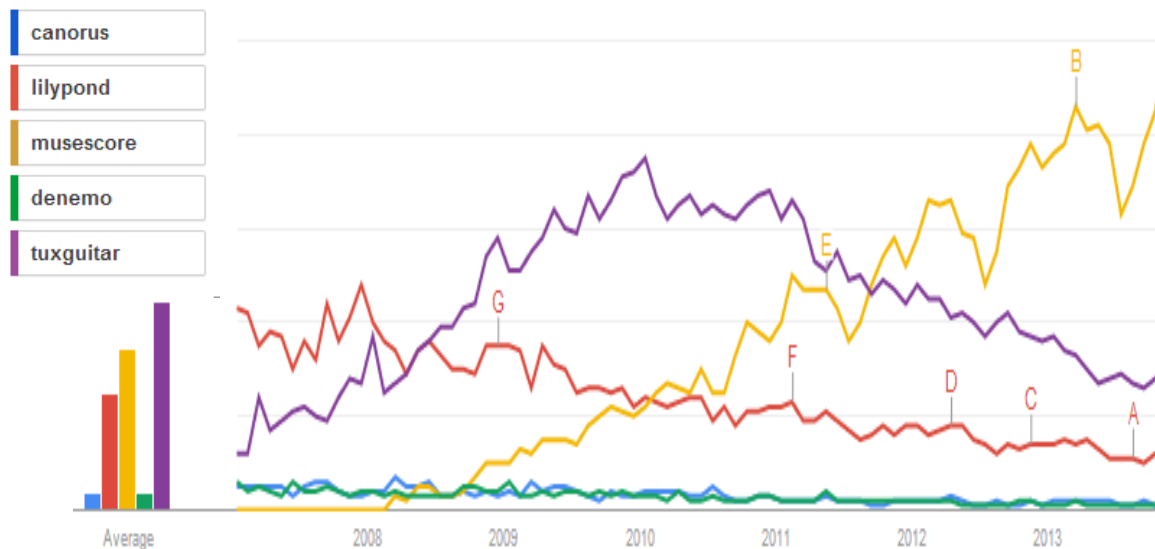
Πρόκειται για ένα generalised (κι όχι πχ guitar specific) και cross - platform notation software. Αναπτύσσεται σταθερά από το 2007 (έχει φτάσει στην 1.3 έκδοσή του και εντός του επόμενου χρόνου αναμένεται η 2.0 έκδοση) και περιλαμβάνει σοβαρό ποσοστό των δυνατοτήτων που προσφέρουν τα proprietary λογισμικά. Δίνουμε παρακάτω τους βασικότερους λόγους που οδήγησαν στην επιλογή αυτή.

- Δημοφιλία

Μπορεί βέβαια να στηρίξει κανείς ότι αυτή τη στιγμή το musescore είναι το πλέον δημοφιλές ελεύθερο scorewriter.

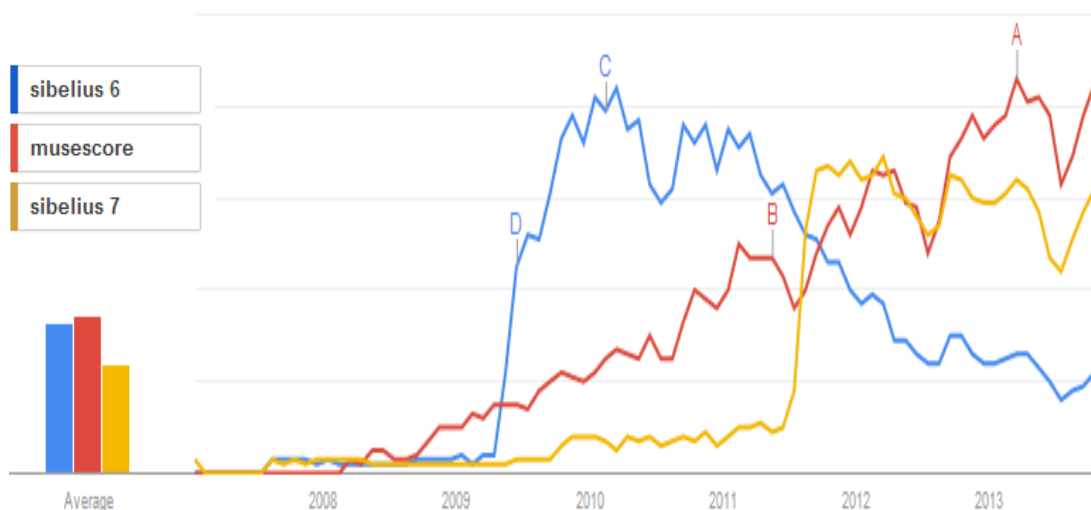
Όπως φαίνεται και στην Εικόνα 11, ανάμεσα στα πλέον γνωστά προγράμματα του χώρου, το musescore έχει σταθερά τις περισσότερες αναζητήσεις στο google ήδη από τα τέλη του 2011. Από εκείνη την περίοδο κι έπειτα παρατηρείται και η σταθερά πτωτική τάση του **TuxGuitar** [33]. Το γεγονός δεν είναι τυχαίο, μιας κι έκτοτε το development της εφαρμογής φαίνεται να έχει εγκαταλειφθεί. Το **lilypond** [34], στο οποίο θα αναφερθούμε παρακάτω εκτενέστερα, δείχνει να χάνει το προβάδισμά του την περίοδο της κυκλοφορίας του musescore 0.9.6, της πρώτης, δηλαδή, έκδοσης, που προέβλεπε τη χρήση όλων των ήχων του general MIDI στην έξοδό του (έως τότε η ηχητική απόδοση της παρτιτούρας πραγματοποιούνταν μόνο με έναν ήχο πιάνου).

Τα άλλα δύο εργαλεία που εμφανίζονται στο διάγραμμα (**canorus [35]** και **denemo [36]**) έχουν σαφώς μικρότερη απήχηση, όπως εξάλλου κι άλλα προγράμματα που δεν φαίνονται εδώ (**Frescobaldi [37]**, **NoteEdit [38]**, **Laborejo [39]**, κτλ).



Εικόνα 11 Αναζητήσεις του musescore στο Google σε σύγκριση με άλλα ελεύθερα scorewriters. Από το [google trends](#) , 12/11/13

Παρακάτω, βλέπουμε ένα αντίστοιχο διάγραμμα, αυτή τη φορά με στοιχεία για τις τελευταίες δύο εκδόσεις ενός εκ των πρωταγωνιστικών proprietary software (sibelius 6 και sibelius 7), εν συγκρίσει με το musescore.



Εικόνα 12 Αναζητήσεις του musescore στο Google σε σύγκριση με τις τελευταίες εκδόσεις του sibelius. Από το [google trends](#) , 12/11/13

Αν και η εξαγωγή συμπερασμάτων με κριτήριο το πλήθος των google αναζητήσεων είναι επισφαλής, μπορούμε να συγκρατήσουμε μία γενική εικόνα των τάσεων στην εν λόγω αγορά.

Ακόμα πιο ενδιαφέρουσα είναι η επίσημη ανακοίνωση της ομάδας των προγραμματιστών του musescore ότι, μέσα στο πρώτο εξάμηνο του 2013, η τελευταία έκδοση της εφαρμογής άγγιξε τα 600.000 downloads [40].

Η κοινότητα των χρηστών του musescore, εκτός από πολυπληθής είναι και αρκετά δραστήρια. Επομένως, υπάρχει μία διαρκής κίνηση ιδεών κι εφαρμογών, παρουσιάζονται κάθε τόσο καινούριες επεκτάσεις και περαιτέρω βελτιώσεις επί του υπάρχοντος προϊόντος, ενώ μπορεί οποιοσδήποτε επεξεργάζεται τον κώδικα της εφαρμογής να εκλάβει ανά πάσα στιγμή βοήθεια και σχόλια για τη δουλειά του.

Επιπλέον, το musescore έχει μεταφραστεί σε 48 γλώσσες (μεταξύ των οποίων και τα ελληνικά) και είναι επισήμως ο scorewriter που χρησιμοποιείται από τους φοιτητές του Ιονίου Πανεπιστημίου στην Ελλάδα και πολλά ακόμα πανεπιστήμια του εξωτερικού [41].

- Import/Export formats.

Τα αρχεία παρτιτούρας τα οποία επεξεργάζονται οι scorewriters είναι κατά κανόνα γραμμένα σε κάποια **γλώσσα σήμανσης (mark up language)**. Το πλέον διαδεδομένο format, εφόσον υποστηρίζεται από σχεδόν όλες τις εφαρμογές (ανάμεσά τους και το musescore), είναι το **MusicXML [42]**, που προέκυψε ως παραλλαγή της γλώσσας **XML (Extensible Markup Language) [43]**, ειδικά για χρήσεις μουσικής σημειογραφίας.

Επιπλέον, για τα περισσότερα προγράμματα, έχουν αναπτυχθεί εγγενείς τύποι (native format). Ο τύπος **.ly** του LilyPond είναι ένα τέτοιο παράδειγμα format. Αν και αρχεία .ly μπορούν να δοθούν στην έξοδο κι άλλων scorewriters, το μόνο λογισμικό που μπορεί να διαβάσει το συντακτικό τους είναι το LilyPond. Το χαρακτηριστικό αυτό φαντάζει εν πρώτοις ως σοβαρό εμπόδιο στην καθιέρωση του εν λόγω format το οποίο όμως, συνεχίζει χρόνια μετά την πρώτη του εμφάνιση (1996) να χρησιμοποιείται ευρέως. Η διαχρονικότητα της απήχησης του LilyPond αφορά πριν και πάνω από όλα το αισθητικά ανώτερο αποτέλεσμα της εξόδου του. Οι δημιουργοί του επικεντρώθηκαν εξ αρχής στην τελική προς εκτύπωση εικόνα της παρτιτούρας προκειμένου να εξασφαλίσουν ότι θα είναι εύκολη στην ανάγνωση από τους μουσικούς εκτελεστές [44]. Πράγματι, το LilyPond είναι πριν από όλα ένα εργαλείο που επιχειρεί να προσομοιώσει τις παλιότερες παρτιτούρες που παράγονταν με την τέχνη της χαρακτικής. Προκειμένου να επιτευχθεί αυτό, εκτελείται σε ένα αρχείο κειμένου που περιγράφει τη μουσική και καταλήγει στην προς εκτύπωση μορφή. Θα λέγαμε ότι μοιάζει περισσότερο με εξειδικευμένη γλώσσα προγραμματισμού παρά με πρόγραμμα επεξεργασίας παρτιτούρας όπως το

musescore ή το finale στα οποία ο χρήστης παρεμβαίνει στο αρχείο της παρτιτούρας μέσω ενός γραφικού περιβάλλοντος χρήστη (GUI).

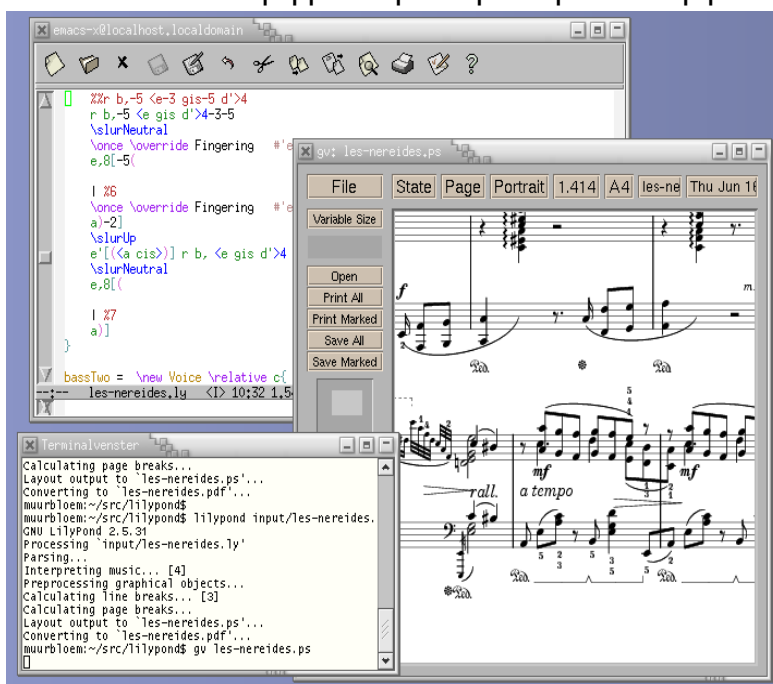
Παράλληλα, έχουν αναπτυχθεί ορισμένα προγράμματα που χρησιμοποιούν το λογισμικό του LilyPond έχοντας ταυτόχρονα προβλέψει για την ευχρηστία της εκάστοτε εφαρμογής, επιχειρώντας να συνδυάσουν τα πλεονεκτήματα διαφορετικών μεθόδων.

Ορισμένα από αυτά (τα πιο διαδεδομένα) είναι τα: **Frescobaldi**, **Denemo** και **Rosegarden**.

Όλα τα παραπάνω είναι λογισμικά που δημιουργήθηκαν πρώτα από όλα για χρήση στο περιβάλλον των λειτουργικών Linux, γεγονός που συνυπολογίστηκε αρνητικά (προτιμήθηκαν προγράμματα που λειτουργούν καλύτερα στα Windows). Το Frescobaldi, ιδιαίτερα, λειτουργεί προπαντός ως Digital Audio Workstation (DAW) και «περιφερειακά» ως scorewriter.

Επιπλέον των ως άνω επιχειρημάτων, πρέπει να σημειωθεί πως, σχεδόν όλα τα scorewriters υποστηρίζουν αποθήκευση σε midi, οπότε μας δίνουν τη δυνατότητα να ακούσουμε αυτό στο οποίο ανταποκρίνεται το κείμενο. Το musescore, όμως, είναι το μόνο από τα ελεύθερα λογισμικά που μπορεί να δώσει στην έξοδό του απευθείας synthesized αρχείο ήχου σε format .wav, .ogg και .flac, που κάνει, δηλαδή μόνο του τη σύνθεση του midi αρχείου σε ήχο. Το χαρακτηριστικό αυτό, αν κι όχι απολύτως απαραίτητο, διευκολύνει κατά πολύ το σκοπό μας.

Γίνεται σαφές, λοιπόν, ότι συνολικά, το musescore έχει το προβάδισμα έναντι όλων των άλλων υποψηφίων για την παρούσα εργασία.



Εικόνα 13: Snapshot του LilyPond

3. Η Ηχογράφηση

Το έτερο αρχείο εισόδου θα είναι μία ηχογράφηση/εκτέλεση της ως άνω παρτιτούρας.

Υπάρχει ένας βασικός περιορισμός στον οποίο θα πρέπει να υπόκειται η ηχογράφηση κι αυτό είναι να μην αποκλίνει σοβαρά ως προς το τονικό περιεχόμενο συγκριτικά με την synthesized παρτιτούρα (το αρχείο μουσικής, δηλαδή, στο οποίο μεταφέρονται τα περιεχόμενα του αρχείου παρτιτούρας, βλ. παράγραφο 4.1.). Με απλούστερα λόγια, δε θέλουμε ο εκτελεστής να κάνει φάλτσα δεδομένου ότι η ποιότητα του alignment συνδέεται με την πιστότητα της εκτέλεσης.

Επιπλέον, προτιμήθηκαν αποσπάσματα μικρής διάρκειας (γύρω στο ένα λεπτό) που εκτελούνται από ένα μοναδικό μουσικό όργανο, προκειμένου να κρατηθεί κατά το δυνατό χαμηλή η πολυπλοκότητα των υπολογισμών. Ο περιορισμός αυτός ισχύει προφανώς τόσο για το αρχείο ηχογράφησης όσο και γι' αυτό της παρτιτούρας.

Το format του αρχείου της ηχογράφησης μπορεί να είναι οποιοδήποτε, αρκεί να το αναγνωρίζει το matlab. Το κατά πόσο η σύγκριση στη συνέχεια θα πραγματοποιηθεί για παράδειγμα μεταξύ ενός αρχείου με κατάληξη .wav κι ενός με .mp3, ή ενός με .flac κι ενός με .ogg δεν απασχολεί καθόλου. Το γιατί γίνεται κατανοητό στην παράγραφο 4.1.3 όπου φανερώνεται και ο αντίκτυπος των διαφορετικών επιπέδων συμπίεσης.

Η πρόσβαση στα αρχεία ήχου γίνεται με χρήση της συνάρτησης `audioread` [45] του Dan Ellis.

Οι τύποι αρχείων που δέχεται η συνάρτηση αυτή ως είσοδο είναι οι εξής: Waveform Audio File Format, MP3 format, Free lossless audio codec και Ogg container format.

4. Preprocessing

Με τον όρο Preprocessing ή προεπεξεργασία εννοούμε το σύνολο των εργασιών που πραγματοποιούμε προκειμένου να «ετοιμάσουμε» τα αρχεία εισόδου για τη μετέπειτα αναμεταξύ τους σύνδεση κι αντιπαραβολή.

4.1. Σύνθεση Παρτιτούρας σε Audio

Έχουμε ήδη πει ότι το musescore έχει τη δυνατότητα να εξάγει τα περιεχόμενα μίας ψηφιακής παρτιτούρας σε αρχείο ήχου. Το format της υπό επεξεργασία παρτιτούρας κανονικά δεν ενδιαφέρει, αρκεί να αναγνωρίζεται από τον scorewriter. Παρόλα αυτά, εμείς θα φροντίζουμε ώστε να είναι σε μορφή musicXML (αν δεν είναι ήδη, εξάγεται σε .xml από το musescore). Μας παρέχεται έτσι η δυνατότητα να εξασφαλίσουμε ότι το αρχείο που πρόκειται να συντεθεί θα έχει σταθερή ταχύτητα. Παρόλο που η απαίτηση αυτή αναγκαστικά ικανοποιείται πριν από τη σύνθεση σε ήχο, το σκεπτικό που βρίσκεται πίσω της καθώς κι ο τρόπος με τον οποίο ικανοποιείται αναλύονται αργότερα, στην παράγραφο 4.7.3.1, για λόγους συνέχειας. Προς το παρόν θα σταθούμε μόνο στη καθαυτή διαδικασία του synthesis. Πρόκειται για μία λειτουργία με δύο βασικά βήματα [46]:

1. Δημιουργία μιας midi αναπαράστασης της παρτιτούρας
2. Σύνθεση του midi σε audio

Ξεκινώντας από το πρώτο βήμα, πρέπει καταρχήν να ορίσουμε τι είναι το midi.

4.1.1 MIDI

Με τον όρο **MIDI** (συντομογραφία του **Musical Instrument Digital Interface**) εννοείται ένα τεχνικό πρότυπο που αναπτύχθηκε πρωτίστως προκειμένου να επιτραπεί η επικοινωνία μεταξύ διαφορετικών ψηφιακών μουσικών οργάνων [47]. Πρωτοεμφανίστηκε στις αρχές της δεκαετίας του 80 και, τριάντα χρόνια μετά, αν κι έχει αλλάξει ελάχιστα, η παρουσία του σε κάθε είδους μουσικές εφαρμογές είναι καθολική.

Τα MIDI δεδομένα αποθηκεύονται σε ένα **Standard MIDI File (SMF)** [48]. Μία συνηθισμένη παρανόηση είναι ότι τα MIDI files είναι αρχεία ήχου. Το MIDI file δεν περιέχει δεδομένα ήχου αλλά τις οδηγίες που χρειάζεται ένα synthesizer για να παράγει ήχους [49]. Πρόκειται για μία συλλογή στατιστικών πληροφοριών που συγκεντρώνονται σε μία δομή εν είδει πίνακα και περιγράφουν πλήρως το αντίστοιχο μουσικό απόσπασμα.³

³ Εξάλλου, τα αρχεία MIDI καταλαμβάνουν χιλιάδες φορές μικρότερο χώρο μνήμης απ' ό,τι τα αρχεία ήχου τα οποία περιγράφουν. Μία παράπλευρη συνέπεια της ιδιότητας αυτής ήταν ότι σε εποχές που η πρόσβαση σε ευρυζωνικά δίκτυα και οι δίσκοι πολλών gigabytes δεν αποτελούσαν ακόμη κοινό τόπο δινόταν η δυνατότητα σε χρήστες να

Ένα αρχείο MIDI μπορεί να περιγράψει μία μουσική εκτέλεση προερχόμενη από κάποιου τύπου (software ή hardware) synthesizer που υποστηρίζει το MIDI πρότυπο. Μπορεί, επιπλέον, να έχει προγραμματιστεί με σκοπό να συντεθεί στη συνέχεια σε ήχο. Σύμφωνα με αυτή τη δεύτερη λογική, τα περιεχόμενα ενός αρχείου κάποιου scorewriter, γραμμένα σε έναν τύπο γλώσσας σήμανσης, «επαναδιατυπώνονται» με την ίδια ακρίβεια σε MIDI format πλέον, το οποίο θα περιλαμβάνει για κάθε δεδομένο στιγμιότυπο της παρτιτούρας πληροφορίες όπως: ποιες νότες εντοπίζονται στο στιγμιότυπο αυτό, με τι ένταση, τι διάρκεια, σε ποιο σημείο πάνω στον άξονα του χρόνου, κτλ.

Ένα αρχείο MIDI περιλαμβάνει ένα κομμάτι **κεφαλίδας (header chunk)** κι ένα ή περισσότερα **track chunks** [52].

Στην κεφαλίδα καταγράφονται βασικές πληροφορίες που χαρακτηρίζουν συνολικά το MIDI αρχείο. Τα κυριότερα από τα δεδομένα που αντλούνται είναι τα ακόλουθα :

file format: παίρνει τιμές 0, 1 και 2. Τιμή 0 συνεπάγεται ότι ακολουθεί ένα μόνο track chunk που περιλαμβάνει όλη την MIDI πληροφορία. Τιμή 1 σημαίνει ότι έπονται 2 ή περισσότερα track chunks, με το πρώτο να περιλαμβάνει meta γεγονότα (κείμενο, στίχους, tempo, end of track, κτλ) και τα υπόλοιπα όλα τα μουσικά γεγονότα. Όταν, τελικά, έχουμε τιμή 2, θα υπάρχουν πολλαπλά track chunks, καθένα εκ των οποίων όμως θα αντιστοιχεί και σε μία ξεχωριστή, ανεξάρτητη midi ακολουθία.

time division: υποδεικνύει τη διάρκεια ενός MIDI tick σε δευτερόλεπτα. Τα MIDI ticks είναι οι μονάδες που χρησιμοποιεί το MIDI για να τοποθετήσει στον άξονα του χρόνου τα διάφορα μουσικά γεγονότα.

Στα track chunks καταγράφονται ακολουθίες από MIDI γεγονότα, δηλαδή, MIDI μηνύματα [53] που συνοδεύονται από την εκάστοτε χρονική στιγμή κατά την οποία αυτά λαμβάνουν χώρα.

Ιδού ένα παράδειγμα ενός τέτοιου MIDI γεγονότος [54]:

0xA0 0x91 0x69 0x28

Πρόκειται για τέσσερις δεκαεξαδικούς αριθμούς που καταλαμβάνουν 4 bytes. Ο πρώτος, 0xA0, ισούται με τον δεκαδικό 160 και μας υποδεικνύει το πλήθος των MIDI ticks που παρεμβάλλονται μεταξύ του τρέχοντος και του προηγούμενου γεγονότος. Αν λοιπόν, το μήκος ενός tick σε δευτερόλεπτα έχει δοθεί στο header ίσο με 4.6875 ms, για παράδειγμα, το τρέχον γεγονός συμβαίνει $160 \times 4.6875 = 750\text{ms}$ ή $3/4$ του sec μετά το προηγούμενο.

μοιράζονται μουσική. Για το σκοπό αυτό, κατά τη δεκαετία του 90, κυκλοφορούσαν ευρέως δισκέτες με MIDI αρχεία. Το ίδιο διάστημα, οι οικονομικές κάρτες ήχου που κυριαρχούσαν στην αγορά έκαναν χρήση τεχνολογιών χαμηλών δυνατοτήτων, οδηγώντας στην τελική παραγωγή ενός κατά κοινή ομολογία κακόηχου αποτελέσματος. Αυτή η περίοδος εκτεταμένης διακίνησης μουσικής μέσω MIDI, η οποία έφτανε στο τελευταίο στάδιο της επεξεργασίας να αναπαράγεται από hardware χαμηλών προδιαγραφών, οδήγησε στη δημιουργία ενός ολότελα λαθμεμένου συνειρμού (που εντοπίζεται ακόμα και σήμερα σε μεγάλο μερίδιο του κοινού), ότι το MIDI είναι τύπος ηχητικού αρχείου χαμηλής ποιότητας. Στην πραγματικότητα, βέβαια, τα MIDI δεδομένα δεν έχουν καμία επίδραση στην ποιότητα του τελικού ηχητικού αποτελέσματος [50][51].

Τα επόμενα τρία bytes (0x91 0x69 0x28) αποτελούν το MIDI μήνυμα. Το πρώτο, 0x91, είναι το byte κατάστασης του μηνύματος. Ο αριθμός 9 δηλώνει πως έχει δοθεί μήνυμα έναρξης νότας (note on message) και ο αριθμός 1 πως η νότα αυτή πρέπει να παιχτεί στο κανάλι 1. Το 0x69 υποδεικνύει ότι πρέπει να παιχτεί η νότα με MIDI note number ίσο με 69. Πρόκειται για το μεσαίο Λα (A4, κατά σύμβαση 440 Hz. Σύμφωνα με το MIDI πρωτόκολλο, οι νότες αριθμούνται από 0 έως 127). Το δεκαεξαδικό 0x28, ίσο με το δεκαδικό 40, υποδεικνύει την ένταση της νότας που εκτελείται.⁴

Αν σε ένα γεγονός όπως το παραπάνω προσθέσουμε την πληροφορία που λαμβάνουμε από τα meta γεγονότα (tempo, όργανα, κ.α.) που εμφανίζεται σε άλλα μέρη του track chunk (ή των track chunks, αναλόγως), συμπεραίνουμε ότι το MIDI πρότυπο παρέχει όλα τα απαραίτητα εργαλεία για την πλήρη περιγραφή ενός μουσικού αποσπάσματος.

4.1.2 Από το MIDI σε Audio

Η απλή κωδικοποίηση που παρέχει το MIDI και η καθολική του υποστήριξη από τα software synthesizers το καθιστά το ιδανικό ενδιάμεσο εργαλείο για τη μετάβαση από την ψηφιακή παρτιτούρα γραμμένη σε markup language σε αρχείο ήχου.⁵

Στο επόμενο βήμα, λοιπόν, έχουμε το **synthesis**. Η λειτουργία αυτή υλοποιείται από ένα **software synthesizer** και συνίσταται στη σύνδεση της ακολουθίας των MIDI γεγονότων με τους ήχους που κωδικά αυτά περιγράφουν.

Το musescore χρησιμοποιεί ως προεπιλεγμένο synthesizer το **FluidSynth** [55] Πρόκειται για ένα FOSS cross – platform synthesizer το οποίο κάνει χρήση κυματομορφών που έχουν παραχθεί με δειγματοληψία από πραγματικά όργανα (**sample – based synthesis**).⁶

Οι κυματομορφές αντλούνται από αρχεία τύπου **SoundFont** [56][57] τα οποία παρέχουν τράπεζες από ήχους που έχουν ψηφιοποιηθεί με τη μέθοδο της παλμοκωδικής διαμόρφωσης (PCM, για περισσότερες λεπτομέρειες βλέπε παρακάτω).

⁴ Για την ένταση συνηθίζεται να χρησιμοποιείται ο όρος velocity, κυριολεκτικά ταχύτητα. Προκύπτει από τη λογική ότι, το πόσο δυνατά θα ακουστεί μία νότα εξαρτάται από την ταχύτητα με την οποία πιέζεται ένα πλήκτρο ενός MIDI οργάνου. Λαμβάνει τιμές από 0 έως 127.

⁵ Πολλές φορές, ορισμένα χαρακτηριστικά του μουσικού κειμένου, όπως αυτό αποτυπώνεται δια μέσου ενός scorewriter, δεν «φτάνουν» έως και τη MIDI διατύπωσή του. Για παράδειγμα, μία σημείωση rallentando (επιβράδυνση) στο musescore καταγράφεται μόνο σαν κείμενο επί της παρτιτούρας χωρίς να οδηγεί στην αποθήκευση κάποιας εντολής επιβράδυνσης στο MIDI αρχείο. Με αυτό το ζήτημα θα ασχοληθούμε αργότερα στην εργασία.

⁶ Ο τρόπος σύνθεσης με samples έρχεται σε αντιπαράθεση με άλλες μεθόδους όπως οι additive και subtractive synthesis που παράγουν ήχους με χρήση θεμελιωδών κυματομορφών, για παράδειγμα με το ημίτονο και τον τετραγωνικό παλμό.

Έχοντας, πλέον, συνθέσει το αρχείο παρτιτούρας στους αντίστοιχους ήχους, αποστέλλουμε στην έξοδο του scorewriter το αποτέλεσμα σε κάποιο αρχείο ήχου, που εν προκειμένω μπορεί να έχουν κατάληξη .wav, .flac ή .ogg.

4.1.3 Format Synthesized Παρτιτούρας

Σε αυτό το σημείο θα εξετάσουμε τα format εξόδου του Fluidsynth synthesizer. Η περιγραφή του καθενός εξ αυτών γίνεται σε συνδυασμό με την εξέταση του αλγορίθμου με τον οποίο κωδικοποιούνται (για το wav) είτε συμπιέζονται (για τα flac και ogg vorbis) τα περιεχόμενα δεδομένα τους.

4.1.3.1 WAV / PCM

Το .wav format (η ονομασία προκύπτει ως συντόμευση του '**Waveform Audio File Format**') χρησιμοποιείται για την αποθήκευση ήχου σε αρχεία και αναπτύχθηκε από συνεργασία της Microsoft με την IBM [58]. Η υποστήριξη των αρχείων τύπου wav προβλέπεται από όλες τις εκδόσεις των λειτουργικών windows από την έκδοση 95 κι έπειτα. Κατά συνέπεια, είναι στην πράξη ο κανόνας στην αποθήκευση αρχείων ήχου που δεν έχουν υποστεί συμπίεση. Είναι εφαρμογή του τύπου **RIFF (Resource Interchange File Format) [59]**, ένα είδος **περιέκτη (container format)** που αναπτύχθηκε από τη Microsoft για την αποθήκευση δεδομένων σε «**τεμάχια με ετικέτα**» (**tagged chunks**).⁷

Τα container formats δεν περιγράφουν με ποιον τρόπο κωδικοποιούνται τα δεδομένα, παρά μόνο τα «τακτοποιούν» σε «κομμάτια» πληροφορίας (chunks, atoms, packets, pages είναι κάποια από τα ονόματα που δίδονται σε αυτές τις μονάδες, αναλόγως το format) και παρέχουν τα μέσα για την αποθήκευση και το συγχρονισμό διαφορετικών ειδών δεδομένων και meta – δεδομένων.

Η κωδικοποίηση ήχου προς αποθήκευση σε WAV πραγματοποιείται με τη μέθοδο **της παλμοκωδικής διαμόρφωσης (PCM, Pulse – Code Modulation)**.

Η μέθοδος αυτή ακολουθεί τα εξής τρία βήματα:

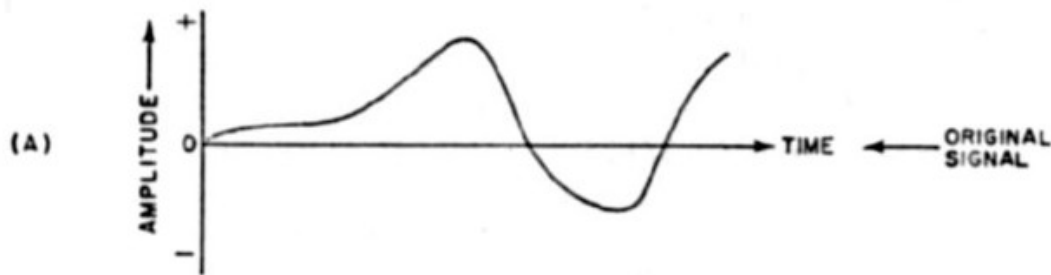
- 1) Δειγματοληψία
- 2) Κβάντωση⁸
- 3) Κωδικοποίηση

Βλέπουμε σχηματικά το φαινόμενο παρακάτω.

Λαμβάνουμε ένα σήμα ήχου, ένα μικρό απόσπασμα του οποίου θα μπορούσε να παρουσιάζει την παρακάτω εικόνα.

⁷ αντίστοιχος τύπος αρχείου για υπολογιστές macintosh είναι το AIFF (Audio Interchange File Format).

⁸ Συναντάται κι ως κβάντιση, κβαντισμός ή διακριτοποίηση.



Εικόνα 14 Σύντομο απόσπασμα ήχου στο πεδίο του χρόνου. Πηγή εικόνας: [Audioengineering Society](#)

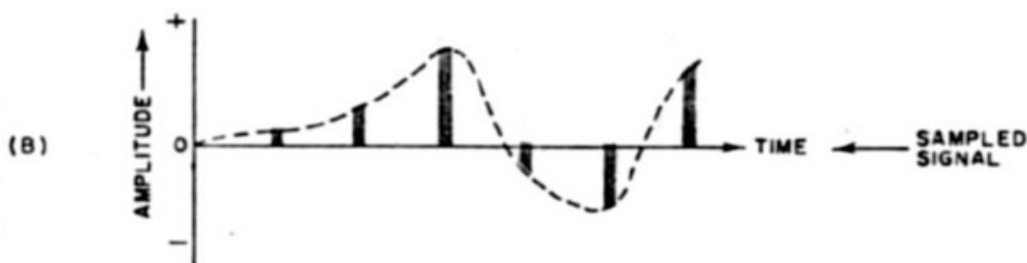
Στη συνέχεια, λαμβάνουμε δείγματά του, προκειμένου να το φέρουμε στη μορφή παλμών τάσης με πλάτος ανάλογο των αντίστοιχων στιγμιαίων πλατών του κύματος ηχητικής πίεσης. Εδώ πρέπει να υπεισέλθουμε και στην έννοια του **ρυθμού δειγματοληψίας (sampling rate)**, του αριθμού, δηλαδή, των δειγμάτων του σήματος που λαμβάνονται ανά δευτερόλεπτο. Πρόκειται για καθοριστικό παράγοντα της ποιότητας του ψηφιακού σήματος. Η πλέον συνηθισμένη τιμή του που αντιστοιχεί και στην ποιότητα CD είναι στα 44100 Hz (samples per second).

Με βάση τη διαπίστωση ότι κατά τη δειγματοληψία περιοδικών συναρτήσεων είναι απαραίτητη η λήψη τουλάχιστον δύο δειγμάτων ανά κύκλο προκειμένου να μη χαθεί η πληροφορία που αφορά την τιμή της φυσικής συχνότητας ενός σήματος, διατυπώνεται το **θεώρημα της δειγματοληψίας [60]**.

Ο μαθηματικό τύπος που διέπει το θεώρημα είναι ο εξής:

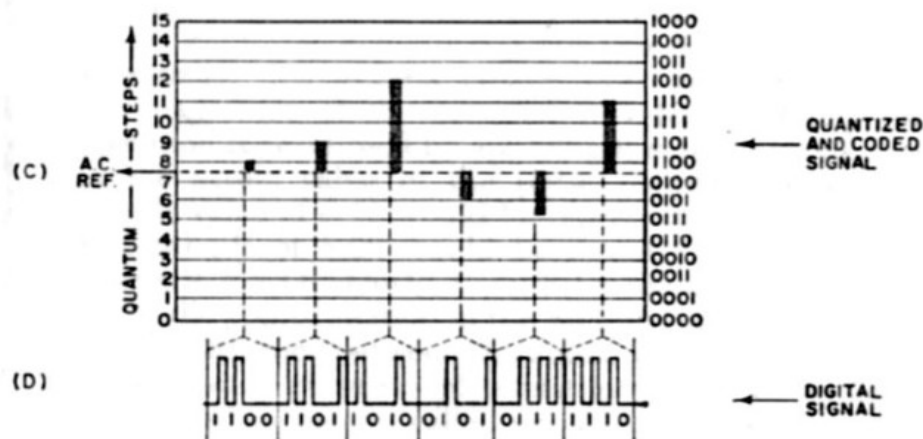
$$f_{max} \leq \frac{f_s}{2}$$

Αυτό που μας λέει η παραπάνω σχέση είναι ότι, προκειμένου να μην έχουμε ανεπιθύμητα φαινόμενα κατά την ψηφιοποίηση σημάτων, είναι απαραίτητο ο ρυθμός δειγματοληψίας να είναι τουλάχιστον ο διπλάσιος από τη μέγιστη φυσική συχνότητα του δειγματοληπτούμενου σήματος. Αντιστρόφως, η μέγιστη συχνότητα ενός ψηφιακού σήματος είναι ίση με το μισό του ρυθμού με τον οποίον λήφθηκαν τα δείγματά του.



Εικόνα 15 Δειγματοληψία. Πηγή εικόνας [Audioengineering Society](#)

Οι τιμές – δείγματα στη συνέχεια υπόκεινται σε κβάντωση⁹ και τελικά κωδικοποιούνται. Η ποιότητα της κβάντωσης, η ποσότητα, δηλαδή των δυνατών ρητών τιμών πλάτους στις οποίες μπορούν να αντιστοιχηθούν τα δείγματα (**amplitude resolution**), εξαρτάται από το μήκος λέξης, το πλήθος δηλαδή των bit που χρησιμοποιούνται για την αποθήκευση κάθε δείγματος (ποιότητα CD: 16 bits). Παράγωγο του μεγέθους αυτού και του ρυθμού δειγματοληψίας είναι ο πιο συνηθισμένος όρος **bit rate**, που αντιστοιχεί στα bits ανά δευτερόλεπτο που απαιτούνται για όλη τη διαδικασία της κωδικοποίησης.



Εικόνα 16 Κβάντωση και Κωδικοποίηση. Πηγή εικόνας: Audioengineering Society, <http://www.aes.org/>

4.1.3.2 Vorbis OGG/Lossy Compression και Ψυχοακουστική

Το ogg είναι ένας τύπος ελεύθερου κι ανοιχτού λογισμικού – περιέκτη¹⁰ (free and open source -FOSS- container format) που μπορεί να περιέχει δεδομένα πολυμέσων κωδικοποιημένα με διαφόρων ειδών **κωδικοποιητές/αποκωδικοποιητές (codecs)**. Στην προκειμένη μας ενδιαφέρει η χρήση του ogg με τον ελεύθερο codec **Vorbis [61]**.

Δουλειά του Vorbis είναι να δέχεται την audio είσοδο, να τη χωρίζει σε παράθυρα και να συμπιέζει αυτά τα παράθυρα σε ανεπεξέργαστα, μη διαμορφωμένα «πακέτα». Στη λειτουργία του ως αποκωδικοποιητής, δέχεται αυτά τα ανεπεξέργαστα πακέτα, τα αποκωδικοποιεί και συνθέτει από αυτά παράθυρα ήχου που όταν επανασυναρμολογηθούν σχηματίζουν ένα αντίγραφο του πρωτότυπου αρχείου ήχου [61].

⁹ Εν προκειμένω έχουμε γραμμική/ομοιόμορφη κβάντωση, όπου οι τιμές κβάντωσης του πλάτους έχουν αναμεταξύ τους την ίδια απόσταση, σε αντίθεση με άλλες μορφές κβάντωσης που χρησιμοποιούν κωδικοποίηση πχ A-Law ή μ-Law, οι οποίες χρησιμοποιούνται προκειμένου να μειωθεί το σφάλμα κβάντωσης στις περιοχές χαμηλού πλάτους.

¹⁰ Η αρχιτεκτονική ενός ogg αρχείου είναι παρόμοια με αυτή ενός wav δεδομένου ότι είναι κι αυτό ένα container format, αποτελούμενο από μία ακολουθία από «τεμάχια με ετικέτα» τα οποία ονομάζονται σελίδες ogg (ogg pages).

Στο σημείο αυτό μας ενδιαφέρει το κομμάτι της συμπίεσης, που για το Ogg Vorbis θα είναι **απωλεστική (lossy compression, συμπίεση με απώλειες)**.

Οι αλγόριθμοι υλοποίησης της συμπίεσης ηχητικών δεδομένων με απώλειες στηρίζονται στην **ψυχοακουστική**, στη μελέτη δηλαδή του τρόπου με τον οποίο το ανθρώπινο ακουστικό σύστημα αντιλαμβάνεται τους ήχους. Ένα κεντρικό συμπέρασμα της ψυχοακουστικής είναι το ότι δεν είναι δυνατόν για τον άνθρωπο να συλλάβει όλα τα δεδομένα που φτάνουν στα αυτιά του και κατά συνέπεια υπάρχει κάποιος όγκος πληροφορίας ο οποίος μπορεί να αφαιρεθεί από το κωδικοποιημένο ηχητικό σήμα, μειώνοντας σημαντικά τον όγκο του.

Σε τι συνίσταται όμως αυτή η «πλεονάζουσα» πληροφορία; Παραθέτουμε παρακάτω μία σειρά από έννοιες που βοηθούν στην απάντηση αυτού του ερωτήματος.¹¹

α) Ακουστός ήχος

Ο άνθρωπος μπορεί να ακούσει ένα φάσμα ήχων που κυμαίνεται περίπου από 20 έως 20000 Hz, όταν η ένταση της εκάστοτε συχνότητας βρίσκεται πάνω από το κατώφλι της ακουστότητας. Οι τιμές αυτές δεν έχουν απόλυτη ισχύ, απεναντίας, μεταβάλλονται ανάλογα με την ηλικία του ακροατή. Έχει διαπιστωθεί ότι είναι σπάνιο φτάνοντας κανείς στη μέση ηλικία να μπορεί να ακούσει συχνότητες άνω των 16kHz [62].

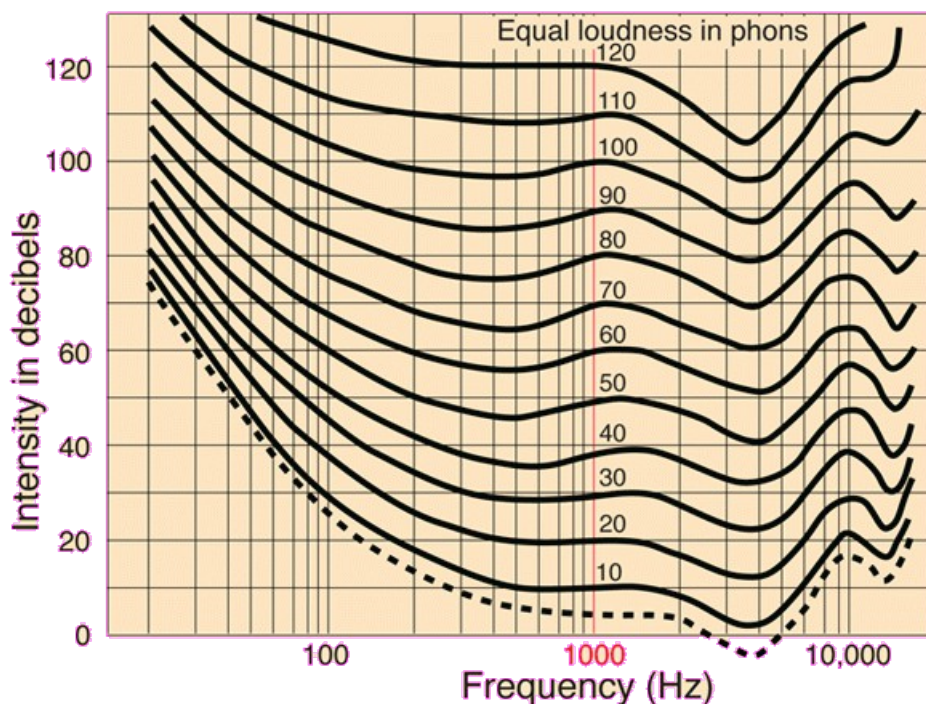
Επιπλέον, το ανθρώπινο ακουστικό σύστημα δεν ερεθίζεται στον ίδιο βαθμό από όλες τις συχνότητες. Το γεγονός αυτό αποτυπώνεται στις καμπύλες ίσης ακουστότητας.

Οι καμπύλες ίσης ακουστότητας καταδεικνύουν τη διαφοροποίηση που υπάρχει μεταξύ της **υποκειμενικής ηχητικής έντασης (loudness**, το μέτρο του πόσο δυνατά ή σιγά αντιλαμβάνεται ένας άνθρωπος κάποιον ήχο) στην οποία αναφερόμαστε με τον όρο **ακουστότητα** και της **φυσικής, αντικειμενικής έννοιας της έντασης** (όπου εδώ εννοείται ως ο ρυθμός μεταβίβασης ενέργειας ηχητικού κύματος διαμέσου μιας επιφάνειας).

Κάθε μία καμπύλη της εικόνας 17 καταγράφει μία περιοχή ίσης ακουστότητας ως συνάρτηση της έντασης σε κλίμακα decibel και της συχνότητας του ήχου. Παρατηρούμε πως οι πιο μπάσες συχνότητες ακούγονται λιγότερο δυνατά συγκριτικά με τις πιο πρίμες συχνότητες στα ίδια επίπεδα decibel¹². Το φαινόμενο εντείνεται όσο κινούμαστε προς πιο

¹¹ Η περιγραφή αυτή είναι μία σύντομη εισαγωγή στα ιδιαίτερα χαρακτηριστικά της ανθρώπινης ακοής που εκμεταλλεύονται οι διάφοροι απωλεστικοί αλγόριθμοι, προκειμένου να γίνει καλύτερα κατανοητό ποια στοιχεία του αρχικού ακουστικού σήματος παραλλάσσονται και τελικά να μπορούμε να συμπεράνουμε με ασφάλεια εάν η συμπίεση των αρχείων είναι επιζήμια για την παρούσα διαδικασία. Η εις βάθος ανάλυση των αλγορίθμων με τους οποίους υλοποιείται η συμπίεση σε κάθε τύπο αρχείου δεν ενδιαφέρει στην παρούσα φάση.

«απαλούς» ήχους. Επιπλέον, το επίπεδο ευαισθησίας του ακουστικού συστήματος κορυφώνεται στην περιοχή μεταξύ 3.5 και 4 kHz .¹³



Εικόνα 17 Καμπύλες ίσης ακουστότητας. Με διακεκομμένη γραμμή, το μέσο κατώφλι της ακοής. Πηγή Εικόνας: Σημειώσεις του καθηγητή Rod Nave, Georgia State University, Department of Physics and Astronomy

Εφόσον, λοιπόν, η ακουστότητα μετράται ως συνάρτηση τόσο της έντασης όσο και της συχνότητας, θα πρέπει να αναφερόμαστε σε αυτή με χρήση κάποιας συχνότητας αναφοράς. Το πλέον σύνηθες είναι η συχνότητα αυτή να επιλέγεται στα 1000 Hz. Έτσι, μπορούμε να προσδιορίσουμε επακριβώς μία καμπύλη ίσης ακουστότητας αν πούμε ότι έχει για παράδειγμα 60 decibel στα 1000 Hz. Ισοδύναμη διατύπωση είναι ότι μία καμπύλη έχει (σε όλο της το μήκος) ακουστότητα 60 phons.¹⁴

¹² Ως ένταση αναφοράς στο λογάριθμο χρησιμοποιείται η ποσότητα $I_0=10^{-12}w/m^2$, η ελάχιστη αντιληπτή από το αυτί ένταση.

¹³ Το μέγιστο αυτό έχει να κάνει με το γεγονός ότι σ' αυτή την περιοχή εντοπίζεται η ιδιοσυχνότητα του ακουστικού καναλιού. Μία ακόμα μικρή κορύφωση που παρατηρείται γύρω στα 13.5 kHz προκύπτει εξαιτίας του τρίτου αρμονικού της ιδιοσυχνότητας

¹⁴ Εδώ αξίζει να αναφέρουμε τον εμπειρικό κανόνα που λέει πως ο δεκαπλασιασμός της έντασης οδηγεί σε διπλασιασμό της ακουστότητας. Αριθμητικά, πολλαπλασιασμός της έντασης (μονάδες: w/m^2) επί 10 ισοδυναμεί με πρόσθεση 10 μονάδων στην κλίμακα decibel. Χρησιμοποιώντας την ισοδυναμία της διατύπωσης 10 dB στα 1000 Hz = 10 phons, μπορούμε να πούμε ότι μία αύξηση 10 phons ενός ήχου αντιστοιχεί σε διπλασιασμό της ακουστότητάς του. Παρόλο που οι μονάδες phons επαρκούν για την περιγραφή του επιπέδου ακουστότητας, δεν μεταβάλλονται ανάλογα με αυτή (για να γίνει ακόμα πιο σαφές: μία αύξηση κάποιου ήχου κατά 30 phons αντιστοιχεί σε $2 \times 2 \times 2 = 8$ φορές αύξηση της ακουστότητας). Για το λόγο αυτό, έχει εισαχθεί μία ακόμα μονάδα, τα

Το μέσο κατώφλι της ακοής (κάτω από το οποίο δε δύναται να ακούσει ο άνθρωπος), στα 0 phons, αποτυπώνεται στο γράφημα με διακεκομμένη γραμμή.

Οι συχνότητες, λοιπόν, που βρίσκονται εκτός του ακουστού φάσματος ή/και έχουν ακουστότητα κάτω από το ελάχιστο κατώφλι θεωρούνται περιττές κι αφαιρούνται.

β) Ταυτόχρονη επικάλυψη

Αν δύο ήχοι με παραπλήσιες συχνότητες και ίση ακουστότητα αναπαραχθούν ταυτόχρονα, θα προκύψει ένας ήχος του οποίου η συνολική ακουστότητα θα είναι ελάχιστα πιο μεγάλη συγκριτικά με αυτή καθενός από τους αρχικούς ήχους. Εάν, αντιθέτως, οι συχνότητες των δύο ήχων διαφέρουν αρκετά, το αθροιστικό αποτέλεσμα φτάνει στην αντίληψη του ακροατή με σημαντικά μεγαλύτερη ακουστότητα σε σχέση με την πρώτη περίπτωση. Το φαινόμενο αυτό είναι γνωστό ως φαινόμενο της **ταυτόχρονης επικάλυψης**. Η αιτία της εμφάνισής του εντοπίζεται στο γεγονός ότι κοντινές συχνότητες «ανταγωνίζονται» για τις ίδιες νευρικές απολήξεις στη βασική μεμβράνη του έσω αυτιού, με αποτέλεσμα η δυνατότητα ανταπόκρισης των απολήξεων αυτών να περιορίζεται [63].

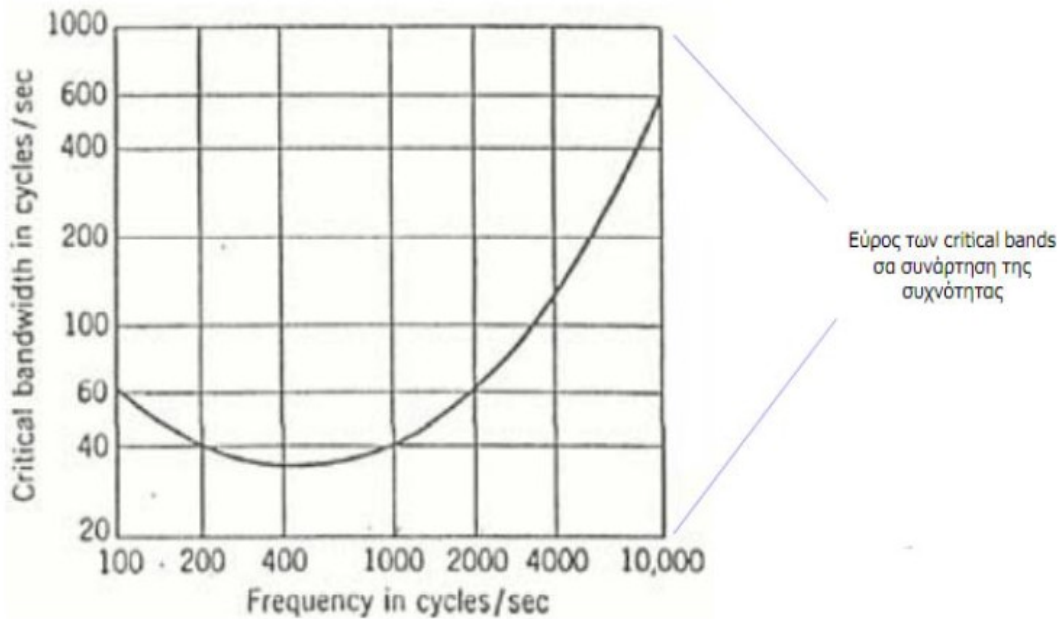
Το ζητούμενο εδώ είναι η ποσοτικοποίηση των αποστάσεων των συχνοτήτων, να προσδιορίσουμε δηλαδή πότε δύο συχνότητες είναι αρκετά κοντά ώστε να αρχίσουν να «θολώνουν».

Ο προσδιορισμός αυτός επιτυγχάνεται με το χωρισμό των συχνοτήτων σε «μονάδες αντίληψης», τις λεγόμενες **κρίσιμες ζώνες (critical bands)**. Οι κρίσιμες ζώνες μπορούν να παρομοιαστούν με ζωνοδιαβατά φίλτρα καθένα εκ των οποίων έχει διαφορετικό εύρος, ανάλογα με την περιοχή συχνοτήτων στην οποία βρισκόμαστε. Το εύρος στις χαμηλές συχνότητες βρίσκεται κοντά στα 90Hz ενώ στις υψηλότερες κυμαίνεται μεταξύ ενός μουσικού τόνου και το 1/3 της οκτάβας [64].

Πίνακας 1 Κεντρικές συχνότητες κρίσιμων ζωνών και τα αντίστοιχα εύρη συχνοτήτων

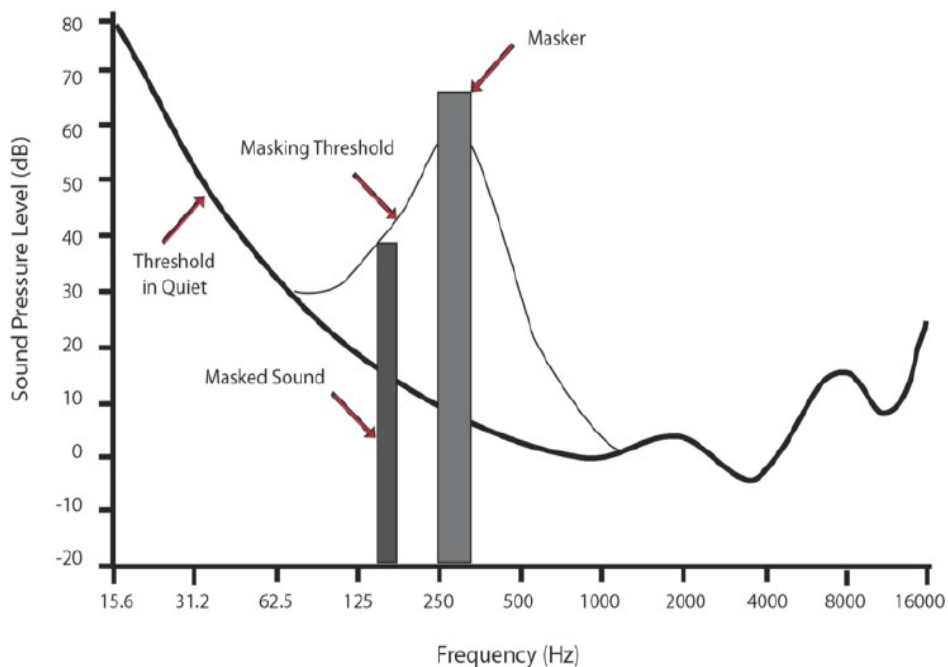
Center Freq (Hz)	Critical Bandwidth (Hz)
100	90
200	90
500	110
1000	150
2000	280
5000	700
10000	1200

sones. 1 sone έχει τυχαία επιλεγεί να είναι ίσο με 40 phons. Έτσι, ακολουθώντας την πραγματική αύξηση της ακουστότητας, 50 phons αντιστοιχούν σε 2 sones, 60 phons σε 4 sones κ.ο.κ.



Εικόνα 18 Εύρος των κρίσιμων ζωνών ως συνάρτηση της συχνότητας. Πηγή Εικόνας: Σημειώσεις του καθηγητή Ανδρέα Φλώρου για το μάθημα Ακουστική & Ψυχοακουστική, Ιόνιο Πανεπιστήμιο, Τμήμα Τεχνών Ήχου και Εικόνας

Βλέπουμε παραπάνω, στον πίνακα 1, μερικές ενδεικτικές κεντρικές συχνότητες συνοδεία του αντίστοιχου εύρους ζώνης και, επιπλέον, στην εικόνα 18 ένα γράφημα που περιγράφει το εύρος των critical bands συναρτήσει της συχνότητας.



Εικόνα 19 Ταυτόχρονη Επικάλυψη. Πηγή Εικόνας: Μετάδοση περιφερικού ήχου μέσω πρωτόκολλου mp3, Διπλωματική Εργασία του Κουκούλη Ιωάννη, Θεσσαλονίκη, Ιανουάριος 2011

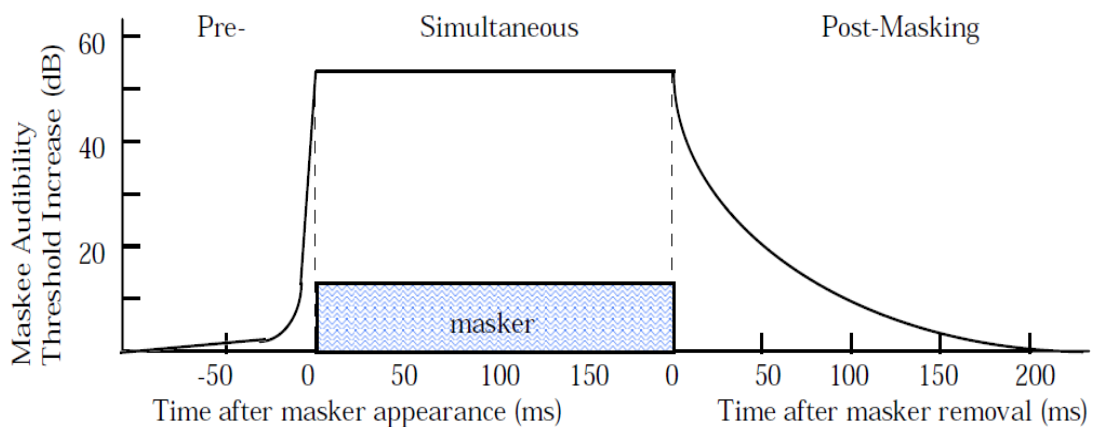
Έτσι λοιπόν, αφού το φάσμα του προς συμπίεση ήχου χωριστεί σε ζώνες συχνοτήτων, εντοπίζονται εντός τους κυρίαρχες συχνότητες και δημιουργείται γύρω από κάθε μία ένα **κατώφλι επικάλυψης**. Το κατώφλι αυτό βέβαια σχηματίζεται επιπρόσθετα στην καμπύλη ακοής. Η λειτουργία αυτή παρουσιάζεται γραφικά στην εικόνα 19.

Όπως βλέπουμε και παραπάνω, εάν μία συχνότητα βρεθεί εντός της ίδιας κρίσιμης ζώνης με μία δεύτερη, ισχυρότερη συχνότητα, ενδέχεται να αφαιρεθεί (να επικαλυφθεί) ως μη δυνάμενη να γίνει αντιληπτή από τον άνθρωπο, λόγω του ότι βρίσκεται κάτω από το κατώφλι επικάλυψης της δεύτερης, αν και βρίσκεται πάνω από το κατώφλι ακοής.

γ) Χρονική επικάλυψη

Μία συχνότητα που εμφανίζεται με πολύ κοντινή χρονική απόκλιση από κάποια δεύτερη, αρκετά ισχυρότερη συχνότητα, ενδέχεται να επικαλυφθεί ακόμα κι αν ανήκει σε κάποια μακρινή κρίσιμη ζώνη.

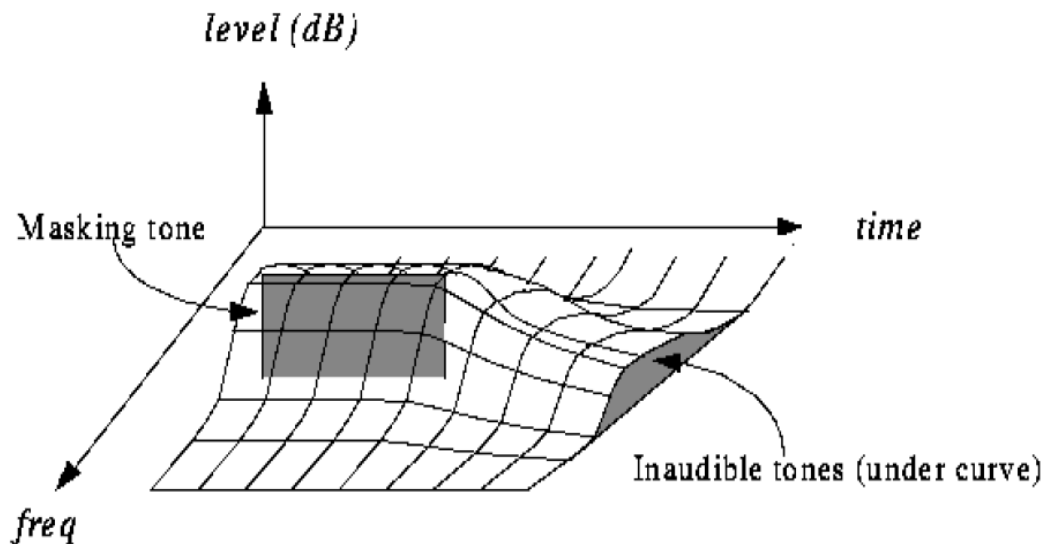
Το φαινόμενο της επικάλυψης εμφανίζεται και στο πεδίο του χρόνου. Απότομα σήματα σύντομης διάρκειας, όπως για παράδειγμα ο ήχος κρούσης κάποιου κρουστού οργάνου, δημιουργούν περιοχές στο χρόνο κατά τις οποίες ο ακροατής δεν μπορεί να αντιληφθεί άλλα σήματα των οποίων η ένταση βρίσκεται κάτω από το νέο, ανυψωμένο κατώφλι ακουστότητας.



Εικόνα 20: Χρονική Επικάλυψη. Πηγή Εικόνας: "A Review of Algorithms for Perceptual Coding of Digital Audio Signals", Ted Painter, Andreas Spanias

Το φαινόμενο δεν εντοπίζεται μόνο κατά τη διάρκεια της εμφάνισης του ήχου - «επικαλυπτή» (masker) αλλά και λίγο πριν και λίγο μετά. Η προηγηθείσα επικάλυψη έχει διάρκεια περίπου 5 ms ενώ αυτή που έπεται κυμαίνεται μεταξύ 50 και 300 ms.

Έχουμε, λοιπόν, άλλη μία περίπτωση όπου ήχοι οι οποίοι μπορούν να εντοπιστούν από το σύστημα κωδικοποίησης δε γίνονται αντιληπτοί από το άνθρωπο και κατά συνέπεια αφαιρούνται κατά την συμπίεση.



Εικόνα 21: Και τα δύο είδη επικάλυψης μαζί. Πηγή εικόνας : Multimedia Communications: Coding, Systems and Networking, Prof. Tsuhan Chen

Τα παραπάνω φαινόμενα είναι που εκμεταλλεύονται οι αλγόριθμοι συμπίεσης που κάνουν χρήση του ψυχοακουστικού μοντέλου ώστε να συμπεράνουν ποιες συχνότητες μπορούν να αφαιρεθούν και να επιφέρουν την επιθυμητή μείωση του όγκου των ηχητικών αρχείων. Μαζί με τον όγκο βέβαια, ενδέχεται να μειωθεί και η ποιότητα του ήχου που φτάνει στα αυτιά μας.

Η ποιότητα του τελικού ηχητικού αποτελέσματος εξαρτάται αφενός από παράγοντες ανεξάρτητους από τη συμπίεση, όπως το ρυθμό δειγματοληψίας, το μήκος λέξης και τον αριθμό καναλιών της ασυμπίεστης εισόδου αλλά και αστάθμητους παράγοντες όπως ο περιβάλλον θόρυβος και αφετέρου από τον ίδιο τον αλγόριθμο συμπίεσης και το επιλεγμένο bit rate του εκάστοτε λογισμικού με τον οποίον αυτός υλοποιείται.

Πάντως, το αντιληπτό αποτέλεσμα της συμπίεσης δεδομένων δύναται να είναι όμοιο με την ασυμπίεστη είσοδο του συστήματος. Σε αυτή την περίπτωση η συμπίεση ονομάζεται **διαφανής (transparent)**. Αν και πρόκειται για υποκειμενικό μέτρο, έχουν οριστεί πειραματικά κατώφλια όπου διαφορετικά format επιτυγχάνουν τη διαφάνεια. Για παράδειγμα, τα σφάλματα MP3 συμπίεσης θεωρούνται ασήμαντα σε bit rate 192 kbit/s και sampling rate 44.1 kHz. Το Vorbis επιτυγχάνει διαφάνεια συμπιεσμένων δεδομένων και σε χαμηλότερα bit rates.

Σε περίπτωση όμως που το συμπιεσμένο αρχείο εμφανίζει ελλείψεις και πλασματικές τιμές, το σήμα εξόδου μπορεί να ηχεί [65]:

- Παραμορφωμένο

Μία ενδεχόμενη ανεπάρκεια σε bits, λόγω πχ χαμηλού bit rate, μπορεί να οδηγήσει σε απώλεια εύρους ζώνης, πράγμα που συνεπάγεται τη διαγραφή ορισμένων υψηλών συχνοτήτων.

- Θορυβώδεις

Ένας codec συμπίεσης, αφού εφαρμόσει στο bitstream εισόδου το ψυχοακουστικό μοντέλο που περιγράφεται πιο πάνω, κβαντώνει τις προκύπτουσες φασματικές συνιστώσες, προκειμένου μετά να τις κωδικοποιήσει. Η μέθοδος με την οποία πραγματοποιούνται τα βήματα αυτά διαφέρει για κάθε αλγόριθμο. Σε κάθε περίπτωση, η διακριτοποίηση άρρητων φασματικών τιμών θα εμπεριέχει κάποιο σφάλμα, λόγω της διαφοράς μεταξύ των πραγματικών τιμών και των επιπέδων κβάντωσης. Το σφάλμα αυτό μπορεί να ιδωθεί ως ένα επιπρόσθετο σήμα και δη, εξαιτίας της στοχαστικής του συμπεριφοράς, ως θόρυβος, ο επονομαζόμενος θόρυβος κβάντωσης. Κατά τη φάση της αποκωδικοποίησης στην οποία συντίθεται το συμπιεσμένο ηχητικό σήμα, ο θόρυβος κβάντωσης γενικά κρύβεται εντός του σήματος. Αν, όμως, το ηχητικό σήμα περιλαμβάνει κάποια αύξηση στην ένταση, λόγω για παράδειγμα κάποιας δυνατής κρούσης, το σφάλμα κβάντωσης αυξάνεται παρομοίως. Επιπλέον, το σήμα του σφάλματος (θόρυβος) απλώνεται σε όλη τη (χρονική) διάρκεια του παραθύρου σύνθεσης που χρησιμοποιεί η τράπεζα φίλτρων του αποκωδικοποιητή. Αν, λοιπόν, η προαναφερθείσα αύξηση της έντασης έπεται σιωπής, ενδέχεται το αυξημένο σφάλμα να προηγείται της ίδιας της αιτίας ύπαρξής του. Σε αυτή την περίπτωση, ο θόρυβος εκτίθεται κατά το προηγηθέν απόσπασμα σιωπής, δημιουργώντας ένα ακουστικό σφάλμα που ονομάζεται pre - echo.

- Τραχύ

Χαμηλό bit rate μπορεί να συνεπάγεται και χαμηλή χρονική ανάλυση από μεριάς του κωδικοποιητή, τέτοια που να οδηγήσει σε προβλήματα χρονισμού κατά τη σύνθεση του συμπιεσμένου σήματος ήχου. Το αποτέλεσμα μπορεί να γίνει αντιληπτό στον ακροατή κυρίως σε ηχογραφήσεις ομιλίας, όπου η ανθρώπινη φωνή ακούγεται ελαφρά «ρομποτική», τραχιά, είτε σαν να έχει προκύψει από δύο επικαλυμμένες ηχογραφήσεις του ίδιου αποσπάσματος (απ' όπου και ο όρος 'Double - Speak').

Μετά από την παράθεση των παραπάνω ατελειών που μπορεί να συναντήσουμε στην έξοδο ενός codec συμπίεσης, διαφαίνεται ότι και στις χειρότερες των περιπτώσεων, **οι κύριες συχνότητες ενός μουσικού αποσπάσματος, τα πλάτη τους και οι χρονικές στιγμές που αυτές εμφανίζονται δεν επηρεάζονται από τη συμπίεση.**

Πέραν του Vorbis, το πλέον διαδεδομένο format κωδικοποίησης ήχου που χρησιμοποιεί απωλεστική συμπίεση είναι το MPEG -1 ή MPEG-2 Audio Layer III, γνωστό στο ευρύ κοινό με το όνομα MP3.

Άλλα format που χρησιμοποιούν το ψυχοακουστικό μοντέλο είναι τα AAC, WMA, MPEG-1 Layer II, κτλ.

4.1.3.3 FLAC / Lossless Compression

Η επέκταση .flac είναι συντομογραφία του “**Free Lossless Audio Codec**” [66].

Η λέξη Free δηλώνει ότι το πρόκειται για ελεύθερο λογισμικό ανοιχτού κώδικα.

Η λέξη Lossless δηλώνει ότι η συμπίεση την οποία υφίστανται τα δεδομένα ήχου δεν συνεπάγεται απώλεια πληροφορίας. Η συμπίεση πραγματοποιείται με **κωδικοποίηση εντροπίας [67]**. Βασικό χαρακτηριστικό της κωδικοποίησης εντροπίας είναι η θεώρηση των δεδομένων ως μία σειρά από bytes των οποίων η σημασιολογία αγνοείται. Οι τεχνικές με τις οποίες αυτή εφαρμόζεται είναι :

α) Ο περιορισμός των επαναλαμβανόμενων ακολουθιών.

β) Η στατιστική κωδικοποίηση (η βασική αρχή της οποίας είναι η αντικατάσταση των ακολουθιών χαρακτήρων με κωδικοποιημένες αναπαραστάσεις των οποίων το μήκος είναι αντίστροφο από τη συχνότητα εμφάνισης των αντίστοιχων ακολουθιών).

Δε χρειάζεται να εμβαθύνουμε στις μεθόδους με τις οποίες εφαρμόζεται η μη απωλεστική συμπίεση, αυτό το οποίο μετράει είναι ότι η αποκωδικοποιημένη έξοδος είναι εντελώς όμοια (bit- for – bit identical) με την παλμοκωδικοποιημένη είσοδο.

Βέβαια, η ποιοτική ανωτερότητα του ήχου που αποδίδει ένα flac αρχείο σε σύγκριση με τα απωλεστικά format έχει το κόστος της, καθώς το ποσοστό συμπίεσης που μπορεί να επιτύχει είναι σαφώς χαμηλότερα (το πόσο χαμηλότερα είναι συνάρτηση του bit rate του ασυμπίεστου audio και του bit rate του εκάστοτε αλγόριθμου συμπίεσης).

4.2. Sample Rate Conversion

Το sampling rate καθενός από τα δύο αρχεία δεν μας είναι εκ των προτέρων γνωστό. Η περίπτωση να είναι διαφορετικά θα έχει σαν αποτέλεσμα τα frames της ανάλυσής μας που έχουν το ίδιο μήκος σε samples να αντιστοιχούν σε διαφορετικά μήκη σε δευτερόλεπτα.

Προκειμένου να διασφαλίσουμε ότι σε κάθε περίπτωση θα έχουμε την ίδια χρονική ανάλυση κάνουμε resample τα δύο αρχεία εισόδου στα 44100 Hz [68].

4.3. Downmix

Η απαιτούμενη πληροφορία για την μετέπειτα επεξεργασία του σήματος πρέπει να είναι διαθέσιμη σε ένα κανάλι μόνο. Κάνουμε, λοιπόν, downmix το αρχείο ήχου σε μονοφωνικό.

$$x(i) = \frac{1}{C} \sum_0^{C-1} x_c(i)$$

όπου x το downmixed σήμα, x_c τα samples, C τα κανάλια.

4.4. Zero Padding

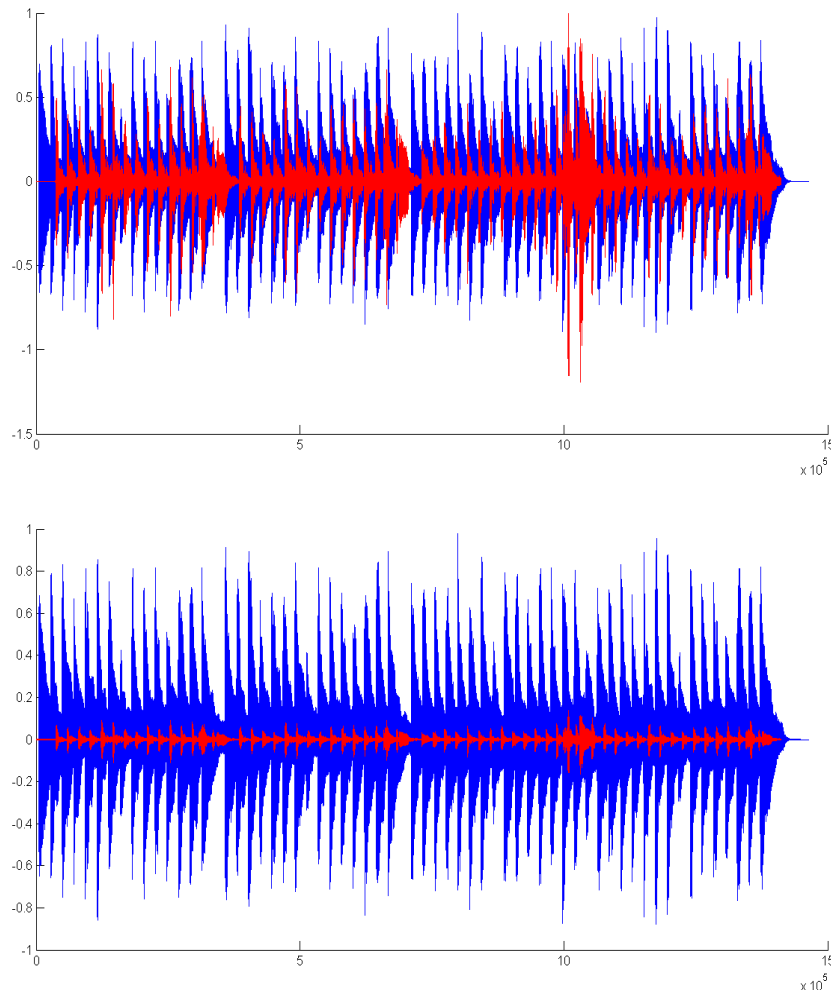
Προκειμένου να διευκολύνουμε το alignment το οποίο θα πραγματοποιήσουμε είναι απαραίτητο τα frames στην αρχή και το τέλος του ενός αρχείου να είναι όμοια με τα αντίστοιχα του δεύτερου. Η συνθήκη αυτή διασφαλίζεται με την τοποθέτηση frames με μηδενικά samples (σιωπή) στην αρχή και το τέλος και των δύο αρχείων.

4.5. Normalization

Ενίοτε οι εντάσεις μεταξύ των σημάτων εισόδου μπορεί να διαφέρουν σημαντικά, γεγονός το οποίο μπορεί να επηρεάσει τα αποτελέσματα της επεξεργασίας. Χρειάζεται, λοιπόν να προχωρήσουμε στην κάτωθι εξομάλυνση των ηχητικών σημάτων (peak normalization):

$$x_{norm} = \frac{x}{|x_{max}|}$$

Όπου x το σήμα εισόδου, $|x_{max}|$ η κατ' απόλυτο μέγιστη τιμή του και x_{norm} το normalized σήμα.



Εικόνα 22: Η synthesized παρτιτούρα (μπλε) και η ηχογραφημένη εκδοχή (κόκκινο) της παιδικής σύνθεσης Ah vous direz-je maman πριν και μετά το peak normalization.

4.6. Audio Feature Extraction

Στο επόμενο βήμα υπολογίζουμε τα short time φασματικά χαρακτηριστικά των σημάτων, εφαρμόζοντας μετασχηματισμό Fourier με παράθυρα (**Short Time Fourier Transform, STFT**) και συνεχίζουμε ή κρατώντας μόνο το magnitude spectrum (οπότε αγνοούμε τη φάση), το λεγόμενο **φασματογράφημα (spectrogram)** ή κάνοντας χρήση του **chromagram**.

Ακολουθεί μία σύντομη επισκόπηση ενός μέρους της θεωρίας του Fourier, με κατάληξη στον υπολογισμό του φασματογραφήματος και του παράγωγου του chromagram, ώστε να γίνει πιο σαφές το τι ακριβώς απεικονίζουν αυτά και κατ' επέκταση που εξυπηρετούν στην παρούσα ανάλυση.

4.6.1 Σειρά Fourier – Μετασχηματισμός Fourier

Η ανάλυση Fourier έχει σαν θεμέλιο τη δήλωση του Γάλλου μαθηματικού Jean Baptiste Joseph Fourier πως οποιαδήποτε περιοδική συνάρτηση μπορεί να εκφραστεί ως ένα άθροισμα αρμονικά συσχετισμένων ημιτονοειδών, καθένα εκ των οποίων πολλαπλασιάζεται με έναν διαφορετικό συντελεστή (αυτό το άθροισμα σήμερα είναι γνωστό ως **σειρά Fourier [69]**):

$$x(t) = \sum_{n=-\infty}^{\infty} c_n \exp\left(\frac{2\pi n}{T} t\right)$$

με τους συντελεστές c_n να ορίζονται ως

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \exp\left(-j \frac{2\pi n}{T} t\right) dt, \text{ για τιμές } n=0, \pm 1, \pm 2, \dots$$

Πρακτικά, θα λέγαμε ότι τα c_n δείχνουν το «βαθμό συμμετοχής» κάθε κύματος κυκλικής συχνότητας $2\pi n/T$ στο τελικό σήμα.

Ο πρώτος όρος από τους παραπάνω περιγράφει μία σειρά μιγαδικών τιμών που, μέσω της ταυτότητας Euler ($e^{jx} = \cos x + j \sin x$), παριστάνουν διακεκριμένες ημιτονοειδείς συναρτήσεις με συχνότητες πολλαπλάσιες μιας θεμελιώδους συχνότητας. Παρόλο που εκτείνεται στο άπειρο, η σειρά αυτή δε θα μπορούσε να περιγράψει ένα μη περιοδικό σήμα, εφόσον αυτό δεν περιέχει διακεκριμένες συχνότητες αλλά όλες τις συχνότητες.

Εξάλλου, από τον ορισμό των συντελεστών c_n , βλέπουμε πως όταν για ένα σήμα ισχύει $T \rightarrow \infty$ θα έχουμε $c_n \rightarrow 0$.

Ομοίως, η ποσότητα Δf που ορίζει την απόσταση μεταξύ διαδοχικών συχνοτήτων του γραμμωτού φάσματος και έχει τιμή

$$\Delta f = \frac{1}{T}(n+1) - \frac{1}{T}n = \frac{1}{T}$$

θα τείνει στο μηδέν για το απειροδικό σήμα (όπως και η ποσότητα $\Delta \omega$ για την κυκλική συχνότητα). Το συμπέρασμα αυτό αποτελεί ουσιαστικά μία αναδιατύπωση των προηγουμένων, ότι δηλαδή σήμα άπειρης περιόδου περιέχει ένα συνεχές συχνοτήτων.

Αντί αυτών χρησιμοποιείται το μέγεθος

$$\frac{c_n}{\Delta f} = c_n T = \int_{-T/2}^{T/2} f(t) \exp(-j \frac{2\pi n}{T} t) dt$$

το οποίο για $T \rightarrow \infty$ μας οδηγεί στο εξής

σήμα:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \exp(-j\omega t) dt$$

Πρόκειται για το **Μετασχηματισμό Fourier (Fourier Transform)** [70], την έκφραση δηλαδή ενός απειροδικού σήματος ως συνεχή συνάρτηση της συχνότητας.¹⁵

Κατ' αντιστοιχία με τους όρους c_n , ο Μ/Σ Fourier είναι η συνάρτηση που μετράει πόσο έντονη είναι η παρουσία διαφορετικών συχνοτήτων στο σήμα $x(t)$. Συνδυάζοντας τα κύματα αυτών των συχνοτήτων με χρήση ολοκληρώματος έχουμε τη δυνατότητα να επαναφέρουμε το αρχικό σήμα στο χρόνο:

$$x(t) = \int_{-\infty}^{\infty} X(\omega) \exp(j\omega t) d\omega$$

4.6.2 DTFT – DFT

Όπως είδαμε ο Μ/Σ Fourier αφορά σήματα συνεχούς χρόνου και δεν προϋποθέτει κάποιο περιορισμό αναφορικά με τη διάρκειά τους. Όμως, η επεξεργασία σημάτων στο πεδίο της συχνότητας με ψηφιακά μέσα προϋποθέτει ότι τα σήματα αυτά είναι διακριτά κι έχουν πεπερασμένη διάρκεια.

Αν έχουμε, λοιπόν, N δείγματα κάποιου σήματος $x[n]$, αυτό θα έχει τον ακόλουθο **Μετασχηματισμό Fourier Διακριτού Χρόνου (Discrete Time Fourier Transform, DTFT)**:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \quad 16$$

Η σχέση της έκφρασης αυτής με το ολοκλήρωμα του Μ/Σ Fourier συναρτήσεων συνεχούς χρόνου είναι εμφανής.

Συχνά αντί για $X(e^{j\omega})$ συναντάμε τη γραφή $X(\omega)$ που υποδεικνύει ότι ο Μ/Σ είναι συνεχής συνάρτηση της συχνότητας ω .

Επιπλέον, επειδή η $X(\omega)$ είναι μιγαδική, γράφεται και ως εξής:

$$X(\omega) = |X(\omega)| e^{j\phi(\omega)}$$

Εισάγουμε, έτσι, το μετασχηματισμό ως έκφραση του πλάτους $|X(\omega)|$ και της φάσης $\phi(\omega)$.

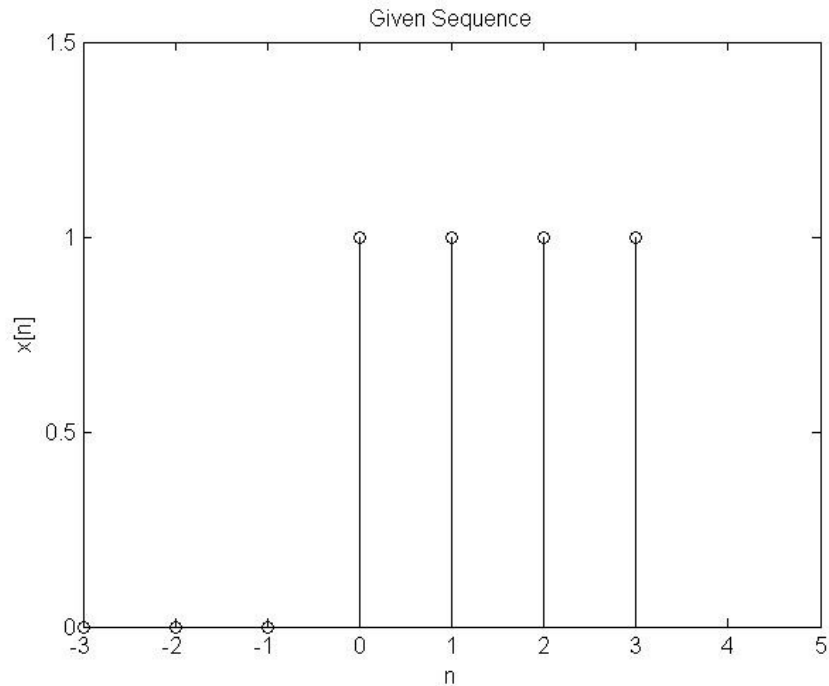
Ας εξετάσουμε για παράδειγμα τη συνάρτηση $x[n] = u[n] - u[n-3]$,

$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

όπου $u[n]$ η μοναδιαία βηματική συνάρτηση :

¹⁵Οι συνθήκες για την ύπαρξη του ολοκληρώματος του Μ/Σ Fourier είναι αρκετά περίπλοκες. Στην πράξη όμως, το ζήτημα αυτό σπάνια αποτελεί αντικείμενο μελέτης, εκτός από την περίπτωση ιδανικών σημάτων, όπως τα ημιτονοειδή.

¹⁶Ο DTFT στη γενική του μορφή ορίζει ότι η παραπάνω άθροιση γίνεται από $-\infty$ έως ∞ . Για σήμα πεπερασμένου μήκους, ο τύπος παίρνει την αναγραφόμενη μορφή.

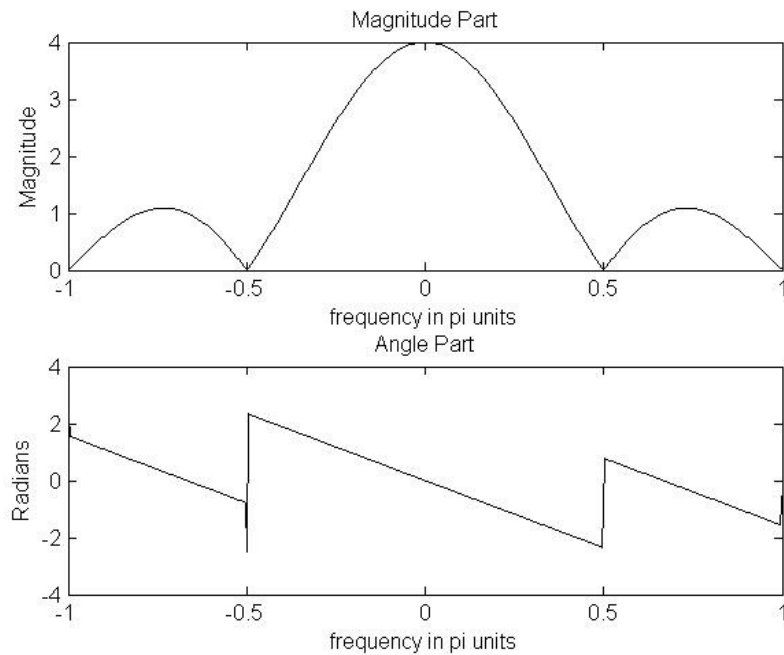


Εικόνα 23: Το σήμα $x[n]=u[n]-u[n-3]$

Εφαρμόζοντας τον δοσμένο τύπο για τον DTFT έχουμε:

$$X(e^{j\omega}) = \sum_{n=0}^3 1 * e^{-j\omega n} = \frac{1 - e^{-j4\omega}}{1 - e^{-j\omega}} = e^{-j3\omega/2} \frac{\sin(2\omega)}{\sin(\omega/2)}$$

Παρατίθενται παρακάτω τα **φάσματα πλάτους και συχνότητας**, όπως ονομάζονται οι καμπύλες $|X(\omega)|$ και $\varphi(\omega)$, αντίστοιχα, συναρτήσεως της συχνότητας.



Εικόνα 24 Φάσμα πλάτους και φάσμα συχνότητας του $x[n]$

Η κεντρική δυσκολία στον υπολογισμό των παραπάνω έγκειται στο γεγονός ότι η συχνότητα ω είναι συνεχής. Το πρόβλημα ξεπερνιέται εύκολα εάν επιλέξουμε έναν πεπερασμένο αριθμό από διακριτές τιμές της ω .

Συγκεκριμένα, θα λάβουμε $k=0,1,2,\dots,N-1$ δείγματα της ω στις τιμές:.

$\omega_k = \frac{2\pi}{N}k$ Αντικαθιστώντας όπου ω τις τιμές ω_k λαμβάνουμε και την

αντίστοιχη ακολουθία δειγμάτων του DTFT, σχηματίζοντας, έτσι, τη «διακριτή εκδοχή» του μετασχηματισμού του $x[n]$ επί του πεδίου της συχνότητας.

Η ακολουθία αυτή είναι ο **Διακριτός Μετασχηματισμός Fourier (Discrete Fourier Transform, DFT [71])** :

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Επιστρέφοντας στο παράδειγμά μας, παρατηρούμε ότι έχουμε $N=4$ δείγματα.

Αν εφαρμόσουμε, λοιπόν τον τύπο του DFT λαμβάνουμε:

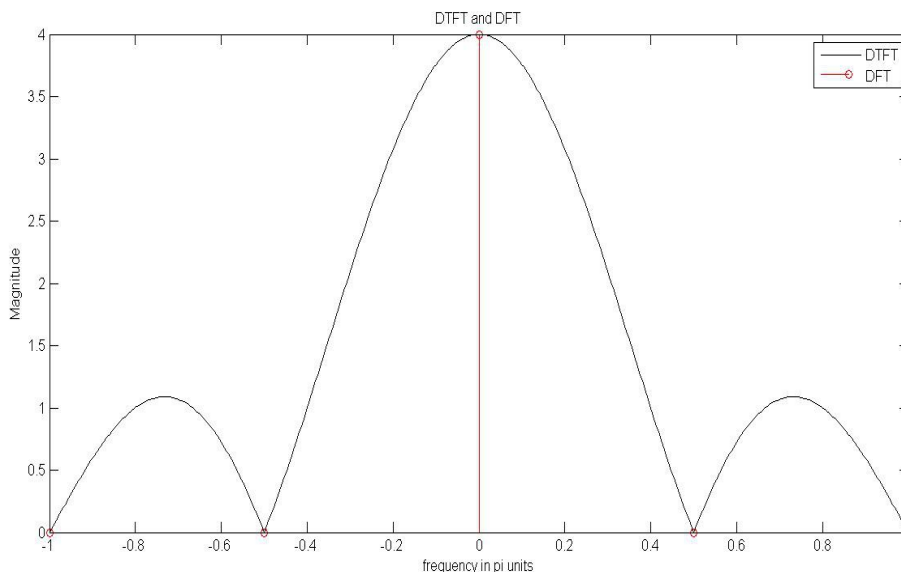
$$X[k] = \sum_{n=0}^3 1 * e^{-j\frac{2\pi}{4}kn} = 1 + e^{-j\frac{\pi}{2}k} + e^{-j\pi k} + e^{-j\frac{3\pi}{2}k} \begin{cases} 4, k=0 \\ 0, k=1,2,3 \end{cases}$$

Εξάλλου, αν αντιμετωπίσουμε τον DFT ως το αποτέλεσμα δειγματοληψίας επί του DTFT, αν αντικαταστήσουμε, δηλαδή, όπου ω τις

τιμές $\omega_k = \frac{2\pi}{4}k$, θα λάβουμε:

$$X(e^{j\frac{2\pi}{4}k}) = e^{-j\frac{3\pi}{4}k} \frac{\sin(\pi k)}{\sin(\pi/4k)} = \begin{cases} 4, k=0 \\ 0, k=1,2,3 \end{cases}$$

Στο ακόλουθο γράφημα, αποτυπώνονται οι τιμές του DFT επί του DTFT φάσματος πλάτους.



Εικόνα 25 Μέτρο των DFT και DTFT του $x[n]$

Για πραγματικά σήματα όπως το παραπάνω, το πραγματικό μέρος του DFT είναι άρτια και το φανταστικό μέρος περιττή συνάρτηση της συχνότητας. Αποτέλεσμα είναι ότι ο μετασχηματισμός περιέχει μόνο μία μοναδικότητα για $\frac{N}{2}$ σημεία. Λόγω των προηγούμενων συμμετριών για πραγματικά σήματα (όπως δηλαδή τα σήματα που επεξεργαζόμαστε στην παρούσα εργασία) το μέτρο του DFT θα είναι άρτια συνάρτηση της συχνότητας και θα έχουμε:

$$|X(k)| = |X(N-k)|$$

Από το πεδίο της συχνότητας μπορούμε να επιστρέψουμε σε αυτό του χρόνου μέσω του **Αντίστροφου Διακριτού Μετασχηματισμού Fourier (Inverse Discrete Fourier Transform, IDFT)**.

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\left(\frac{2\pi}{N}k\right)n}$$

Όπως σε όλες τις Fourier αναπαραστάσεις, το σήμα χρόνου $x[n]$ μπορεί να ιδωθεί ως ένας συνδυασμός σταθμισμένων ημιτονοειδών. Τα ημιτονοειδή αυτά έχουν συχνότητες $(2\pi/N) \cdot k$ και πλάτη $X[k]/N$.

4.6.3 STFT

Το συχνотικό περιεχόμενο των σημάτων ήχου που ενδιαφέρουν στην παρούσα εργασία μεταβάλλεται ως συνάρτηση του χρόνου. Για την ανάλυση σημάτων χρονικά μεταβλητής συχνότητας (μουσικής, λόγου, σεισμικών δονήσεων, κτλ) χρησιμοποιείται ο Μετασχηματισμός **Fourier Βραχέος Χρόνου (Short – Time Fourier Transform, STFT)** και το παράγωγό του **Φασματογράφημα (Spectrogram)**[72].

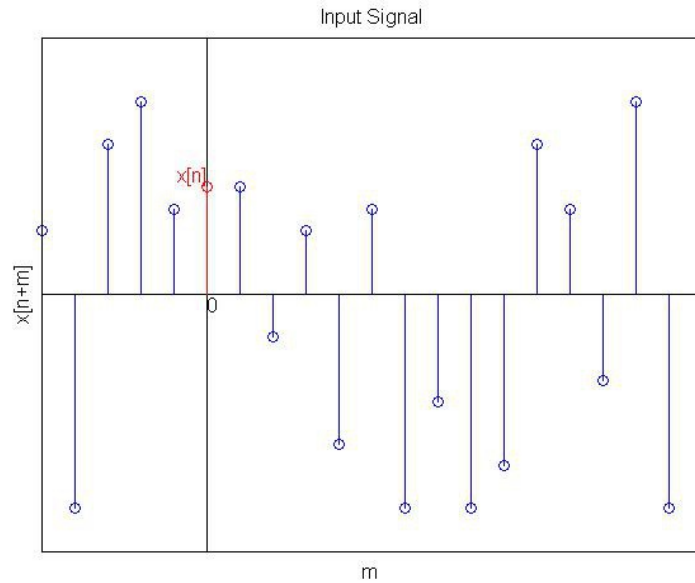
Ο STFT δεν είναι παρά μία επέκταση του DFT. Αυτό που συμβαίνει είναι ότι ξεκινώντας από την αρχή του σήματος λαμβάνουμε ένα απόσπασμα/παράθυρο, εφαρμόζουμε πάνω του το Διακριτό Fourier και συνεχίζουμε ομοίως, κυλίνοντας το παράθυρο κατά μήκος του σήματος. Κατ' αυτό τον τρόπο λαμβάνουμε το φάσμα από αλληπάλληλα στιγμιότυπα του σήματος, σχηματίζοντας έτσι μία εικόνα για την εξέλιξη του συχνотικού περιεχομένου του στο χρόνο.

Καταλήγουμε, με λίγα λόγια, στην παράσταση του σήματός μας ως συνάρτηση τόσο του χρόνου όσο και της συχνότητας.

Δίνεται παρακάτω μία απλή σχηματική αναπαράσταση του πώς εξάγεται ένα παράθυρο από ένα απεριοδικό ψηφιακό σήμα προκειμένου να υποστεί ST Fourier μετασχηματισμό, ένα βήμα μόνο, δηλαδή, από την όλη διαδικασία.

Είναι γενικά πιο χρήσιμο, για λόγους σημειογραφίας, να σκεφτούμε πως η συνάρτηση παραθύρου παραμένει σταθερή με το σήμα εισόδου να κυλιέται επί αυτής, παρά το αντίστροφο.

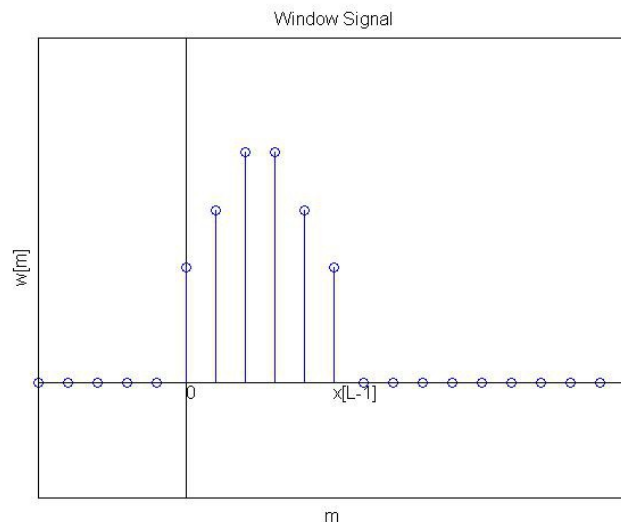
Βλέπουμε, λοιπόν, παρακάτω, ένα απόσπασμα ενός απλού σήματος χρονικά μεταβλητής συχνότητας.



Εικόνα 26 Τυχαίο διακριτό σήμα $x[m]$ μετατοπισμένο στο χρόνο κατά n .

Το σήμα είναι το $x[n+m]$, όπου n ο βαθμός μετατόπισης του και m η εξαρτημένη μεταβλητή.

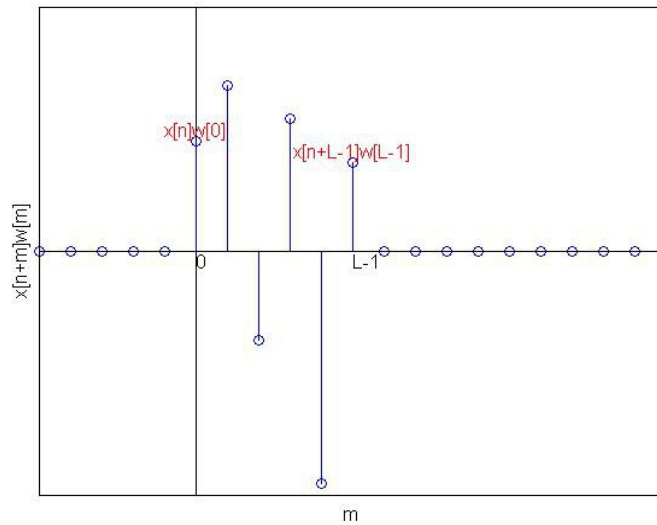
Το παράθυρό μας θα έχει μήκος L . Οι τιμές του επελέγησαν όπως φαίνονται παρακάτω. Γενικά, προτιμώνται παράθυρα με σταδιακά μειωνόμενες τιμές, αφού τα ορθογώνια παράθυρα έχουν ασυνέχειες στα άκρα τους που δημιουργούν προβλήματα στο πεδίο των συχνοτήτων.



Εικόνα 27: Συνάρτηση παραθύρου για εφαρμογή STFT.

Όπως βλέπουμε, τη χρονική στιγμή μηδέν ($m=0$) το σήμα $x[n+m]$ λαμβάνει την τιμή $x[n]$ (με άλλα λόγια, την τιμή του x για την ποσότητα κατά την οποία αυτό έχει μετατοπιστεί).

Αν, λοιπόν, πολλαπλασιάσουμε το σήμα μας με το παράθυρο $w[m]$, θα λάβουμε το αποτέλεσμα που φαίνεται παρακάτω.



Εικόνα 28 Το σήμα $x[n+m]$ πολλαπλασιασμένο με το παράθυρο $w[m]$.

Το τελευταίο αυτό γράφημα μας δίνει την εικόνα ενός και μόνο μέρους του αρχικού σήματος όπως αυτό προκύπτει από την εφαρμογή της συνάρτησης παραθύρου (πρόκειται για παραλλαγμένο απόσπασμα του σήματος εισόδου, εφόσον το παράθυρο δεν είναι ορθογώνιο).

Ας εφαρμόσουμε DTFT στο $x[n+m]w[m]$:

$$X[n,\lambda) = \sum_{m=-\infty}^{\infty} x[n+m]w[m]e^{-j\lambda m} \quad 17$$

n : «θέση στο χρόνο»

λ : συχνότητα

Έπειτα, περνάμε στον Διακριτό Μ/Σ Fourier λαμβάνοντας δείγματα της λ στις τιμές: $\lambda_k = \frac{2\pi}{N} k$

Προκειμένου, δηλαδή, να φτάσουμε στο Μετασχηματισμό Fourier Βραχέος Χρόνου, πρέπει να πραγματοποιήσουμε δειγματοληψία του $X[n,\lambda)$ επί του μοναδιαίου κύκλου. Η τιμή N μας λέει ότι ο DFT μας έχει N δείγματα και το διάστημα δειγματοληψίας της συχνότητας θα είναι $2\pi/N$.

Έχοντας προϋποθέσει ότι $N > L$, καταλήγουμε στον STFT:

$$X[n,k] = \sum_{m=0}^{L-1} x[n+m]w[m]e^{-jk\frac{2\pi}{N}m}$$

Πρόκειται ξεκάθαρα για μία συνάρτηση του χρόνου n αλλά και της συχνότητας k .

Ο STFT μπορεί κι αυτός να αντιστραφεί:

$$x[n+m] = \frac{1}{Nw[m]} \sum_{k=0}^{N-1} X[n,k]e^{j\frac{2\pi}{N}km}$$

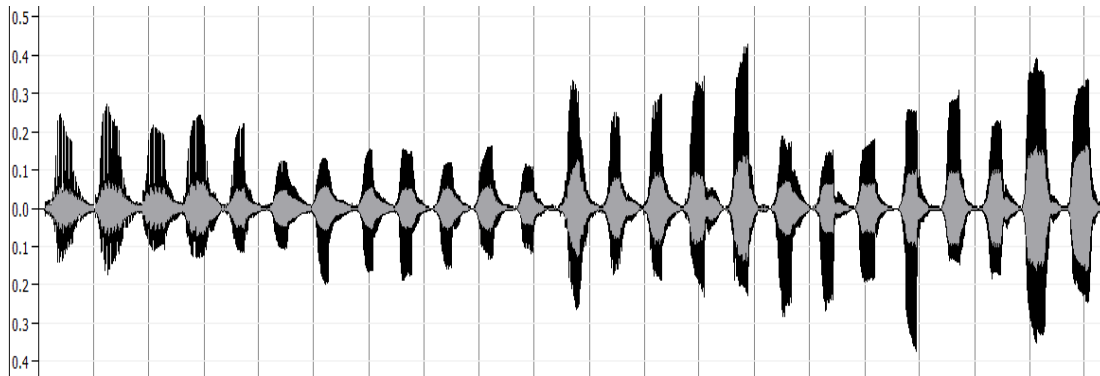
¹⁷ Η γραφή $[n,\lambda)$ επιλέγεται προκειμένου να υποδείξουμε ότι η n έχει διακριτές τιμές ενώ η λ είναι συνεχής.

4.6.4 Spectrogram

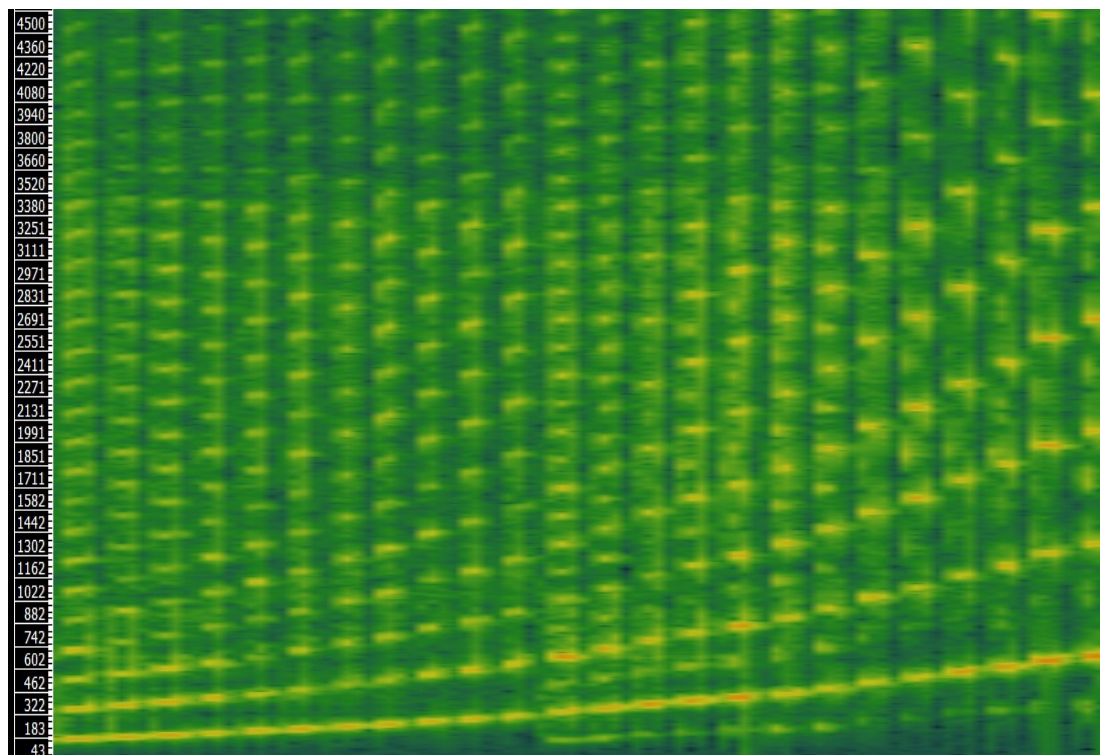
Το φασματογράφημα ορίζεται ως το πλάτος του STFT στο τετράγωνο:

$$|X[n, k]|^2$$

Παρατίθεται παρακάτω ένα παράδειγμα πραγματικού ήχου. Πρόκειται για μία ανοδική ντο μείζονα κλίμακα ηχογραφημένη από άλτο σαξόφωνο.



Εικόνα 29: Άλτο Σαξόφωνο, C4 major scale, 44100 Hz, 705 Kbps. Η επεξεργασία του ήχου έγινε στο audacity. Πηγή ήχου: freesound.org . Η κυματομορφή δημιουργήθηκε στο sonic visualiser .



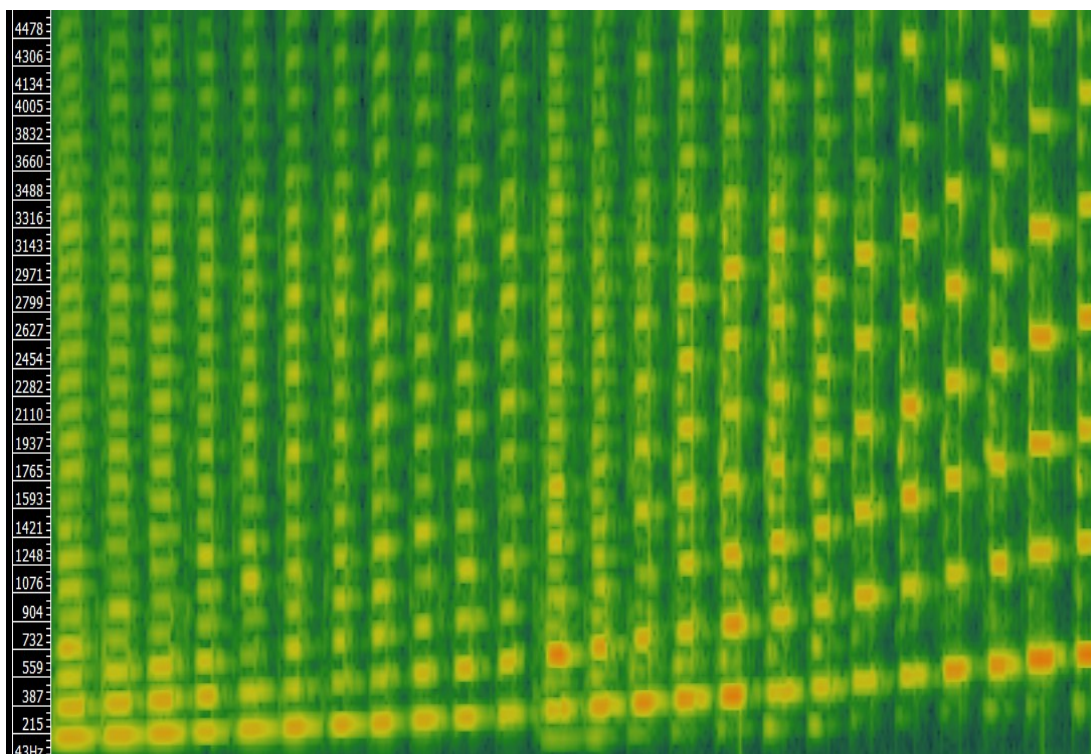
Εικόνα 30: Spectrogram του παραπάνω ήχου με window size 4096 samples και 50% overlap. Δημιουργήθηκε στο sonic visualiser

Βλέπουμε παραπάνω την απεικόνιση του ήχου ως κυματομορφή στο χρόνο και σε μορφή φασματογραφήματος. Το φασματογράφημα έχει στον οριζόντιο άξονα το χρόνο, διαιρεμένο σε παράθυρα μήκους 4096 samples και στον κατακόρυφο τις συχνότητες. Λόγω του θεωρήματος της

δειγματοληψίας θα πρέπει η μέγιστη καταγεγραμμένη συχνότητα να είναι ίση με $\text{sampling rate}/2$ (βλ. 4.1.3.1). Επιπλέον, ο DFT περιέχει μόνο μία μοναδικότητα για $N/2$ σημεία κι έχει φάσμα πλάτους άρτια συνάρτηση της συχνότητας(βλ. 4.6.2). Άρα, ο άξονας των συχνοτήτων έχει (window size) /2 σημεία (εν προκειμένω 2048) κι εκτείνεται από 0 έως 22050 Hz (στην εικόνα 30 έχει γίνει zoom ώστε να βλέπουμε τις σημαντικότερες συχνότητες, έως τα 4500 Hz). Στην τρίτη διάσταση έχουμε τα πλάτη των συχνοτήτων που καταγράφονται με διαβάθμιση στα χρώματα.

Φαίνονται εμφανώς οι διαφορετικές συχνότητες που παίζει το σαξόφωνο. Στο χαμηλότερο επίπεδο μπορούμε να δούμε με πιο έντονες "θερμές" αποχρώσεις τις θεμελιώδεις συχνότητες που αντιστοιχούν στις νότες που ακούγονται στο ηχητικό απόσπασμα (από C4 έως C5). Παραπάνω, σε σταθερά πολλαπλάσια και με μειούμενη ένταση φαίνονται καθαρά οι αρμονικοί κάθε νότας. Έχουμε εν ολίγοις μία σαφή απεικόνιση της κατανομής των συχνοτήτων. Παρατηρώντας όμως τις θεμελιώδεις συχνότητες, θα διαπιστώσουμε ότι η ακριβείς χρονικές στιγμές που αρχίζει και τελειώνει η κάθε νότα είναι αρκετά θολές, σε βαθμό που η αλληλουχία των συχνοτήτων τείνει να σχηματίσει μία συνεχή γραμμή.

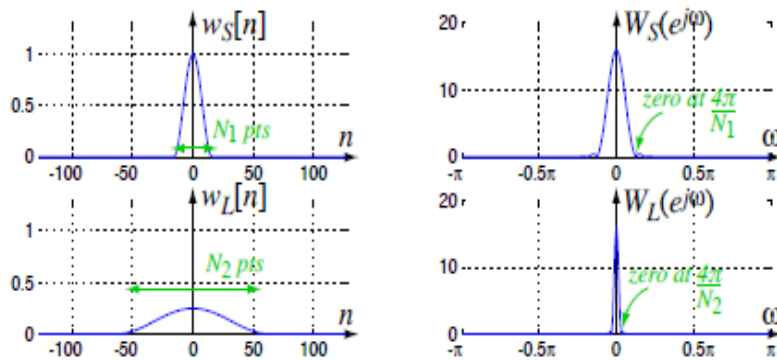
Παρακάτω βλέπουμε τον ίδιο ήχο αποτυπωμένο σε φασματογράφημα διαφορετικής ανάλυσης. Συγκεκριμένα, έχει μήκος παραθύρου 1024 samples και κατ'έπεκταση οι συχνότητες θα χωρίζονται σε διαστήματα μήκους 22050/1024 το καθένα.



Εικόνα 31: Spectrogram του παραπάνω ήχου με window size 1024 samples και 50% overlap. Δημιουργήθηκε στο sonic visualiser

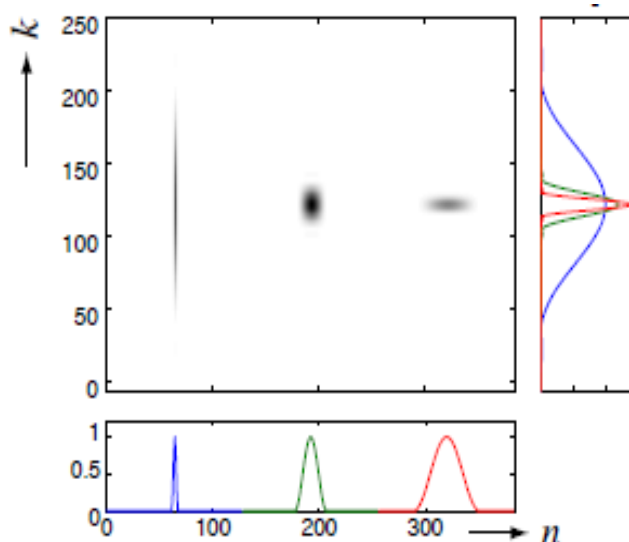
Βλέπουμε πως το συχνοτικό περιεχόμενο είναι πλέον πιο ασαφές, αλλά μπορούμε να διακρίνουμε με ακρίβεια τις χρονικές στιγμές που ξεκινάει και τελειώνει η κάθε μία νότα.

Όπως φαίνεται και παρακάτω το μήκος του παραθύρου στο χρόνο είναι αντιστρόφως ανάλογο του πλάτους του κύριου λοβού στη συχνότητα.



Εικόνα 32 Δύο παράθυρα διαφορετικού μήκους αποτυπωμένα στο χρόνο (αριστερά) και στη συχνότητα (δεξιά). Πηγή εικόνας: Dan Ellis, σημειώσεις από το μάθημα Digital Signal Processing, Department of Electrical Engineering - Columbia University

Κατά την ανάλυση με STFT, λοιπόν, όπου το σήμα αποτυπώνεται στο πεδίο χρόνου-συχνότητας, είναι απαραίτητο να επιλέξουμε κάποιον συμβιβασμό μεταξύ των δύο μεγεθών, προκειμένου να επιτύχουμε, αν όχι την καλύτερη, τότε τουλάχιστον τη «λιγότερο κακή» ανάλυση. Το γεγονός αποτυπώνεται και στην παρακάτω εικόνα, με τον απεικονιζόμενο δίσκο να αλλάζει μήκος και να «θολώνει», διατηρώντας όμως σταθερό το εμβαδό του.



Εικόνα 33 Τρία διαφορετικά σήματα παραθύρου με τρία χρώματα, αποτυπωμένα στο χρόνο (οριζόντιος άξονας), στη συχνότητα (κατακόρυφος άξονας) και στα δύο πεδία ταυτόχρονα. Πηγή εικόνας: Dan Ellis, σημειώσεις από το μάθημα Digital Signal Processing, Electrical Engine

4.6.5 Chromagram

Το **chromagram [73]** είναι επίσης μία τρισδιάστατη απεικόνιση του χρόνου, της συχνότητας και του ενεργειακού περιεχομένου της συχνότητας. Διαφέρει ως προς το spectrogram σ'ό,τι αφορά την κατανομή των συχνοτήτων: αντί για (window size/2) bins έχουμε 12 bins, καθένα από τα οποία αντιστοιχεί και σε ένα ημιτόνιο της χρωματικής κλίμακας. Έτσι, σε κάθε ένα από τα 12 bins τοποθετούνται τα magnitudes των συχνοτήτων που αντιστοιχούν στο ίδιο pitch class (η αλλιώς pitch class).¹⁸

Π. χ. στο bin 3 έχουμε το ενεργειακό περιεχόμενο των συχνοτήτων που ταιριάζουν στη νότα D (ρε), ανεξαρτήτως οκτάβας.

Τα έργα τα οποία αναλύονται στην παρούσα εργασία είναι γραμμένα για συγκεκριμένα όργανα, με βάση δηλαδή τους 12 φθόγγους που αναφέρθηκαν. Η κατηγοριοποίηση των συχνοτήτων, λοιπόν, σε 12 ομάδες κοινού pitch class βοηθάει την ανάλυσή μας. Επιπλέον, η spectrogram αναπαράσταση περιλαμβάνει συχνότητες έως και το μισό του sampling rate, εν προκειμένω έως 22050 Hz. Τα όργανα όμως που χρησιμοποιούνται στη δυτική, «κλασσική» μουσική δεν έχουν σχεδιαστεί για να αγγίζουν τέτοια μεγέθη η θεμελιώδης συχνότητα της υψηλότερης νότας σε ένα πιάνο, για παράδειγμα, δεν υπερβαίνει τα 3951 Hz ενώ η αντίστοιχη της marimba τα 2093 Hz (Εικόνα 34).

Υπάρχει, λοιπόν, στο spectrogram ένας «κενός χώρος» υψηλών συχνοτήτων η επεξεργασία κι αναπαράσταση του οποίου δεν εξυπηρετεί το σκοπό μας. Το ζήτημα αυτό δεν απασχολεί καν στο chromagram όπου οι εν λόγω συχνότητες δε γίνονται ορατές.

Συμπεραίνουμε, λοιπόν, ότι στη δική μας περίπτωση το chromagram είναι, από υπολογιστική άποψη, φθηνότερη λύση, ενώ επιπλέον περιέχει λιγότερη άχρηστη πληροφορία.

Από την άλλη πλευρά, πρέπει να σημειωθεί ότι ο μετασχηματισμός chromagram, αντίθετα με τον STFT, δεν είναι αντιστρέψιμος.¹⁹

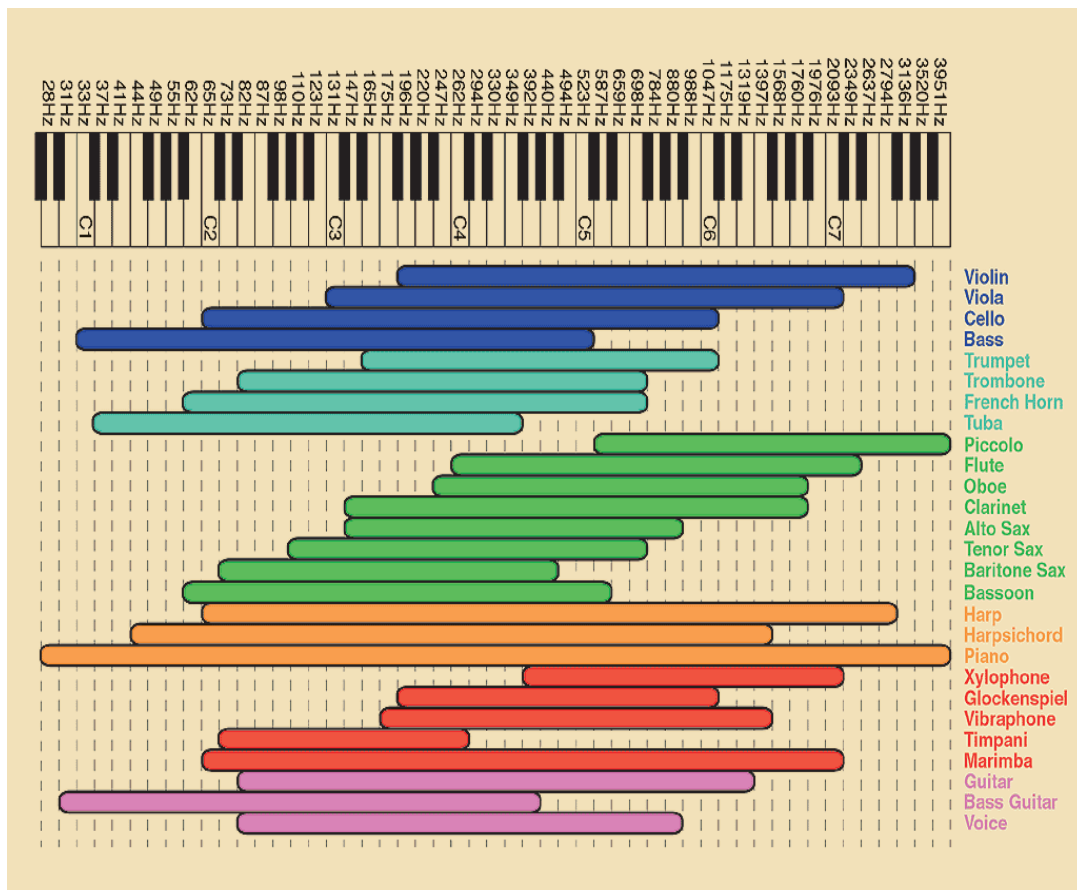
¹⁸ Η κοντινότερη μετάφραση για τη λέξη pitch είναι τονικό ύψος και κατ' επέκταση θα μεταφράζαμε το pitch class ως «τάξη τονικού ύψους». Με τον όρο pitch class στη μουσική εννοούμε ένα το σύνολο των pitches που έχουν μεταξύ τους απόσταση φυσικό αριθμό από οκτάβες, π.χ. το pitch class C (ντο) αποτελείται από τα Cs σε όλες τις οκτάβες.

Θα γράφαμε την τάξη τονικού ύψους 'C' ως το σύνολο:

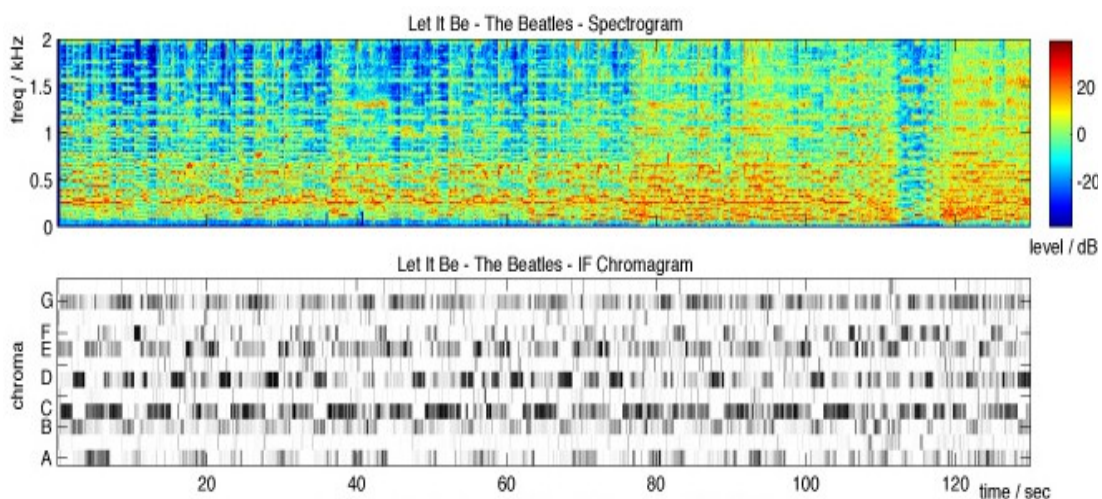
$$\{C_n, n \in \mathbb{N}\} = \{\dots C_{-2}, C_{-1}, C_0, C_1, C_2, \dots\}$$

Παρόλο που δεν υπάρχει επισήμως όριο για την ακολουθία ούτε προς τα κάτω ούτε προς τα πάνω, ο αριθμός των pitches που γίνονται αντιληπτά από το ανθρώπινο αυτί είναι περιορισμένος [74].

¹⁹Ο χρήστης έχει την επιπλέον επιλογή να χρησιμοποιήσει την παράγωγο του chromagram. Με τη μέθοδο αυτή έχουμε τη δυνατότητα να παρατηρούμε με μεγαλύτερη λεπτομέρεια απότομες μεταβολές, άρα και πιο έντονα τη στιγμή που έρχεται μια νέα νότα.



Εικόνα 34 Διάγραμμα συχνοτήτων επιλεγμένων μουσικών οργάνων (<http://obiaudio.com/2010/07/11/eq-chart/>).



Εικόνα 35: Σύγκριση ανάμεσα στη spectrogram και τη chromagram αναπαράσταση του Let It Be των Beatles. Το chromagram είναι σε grayscale κι έχει στον κατακόρυφο άξονά του τις νότες (από A έως G). <http://labrosa.ee.columbia.edu/projects/coversonas/overview.html>

Σε κάθε περίπτωση, ο κατά περίπτωση ποιοτικός (ή/και ποσοτικός, εάν υπάρχει δυνατότητα) έλεγχος του τελικού αποτελέσματος του alignment θα κρίνει ποιά μέθοδος θα επιλεγεί.

Χρησιμοποιήθηκε για το chromagram το implementation `chromagram_IF` του Dan Ellis.

Στον STFT έγιναν δοκιμές με παράθυρα Hanning μήκους 46ms, 23ms και 12ms (2048, 1024 και 512 samples αντίστοιχα, για sampling rate 44100) και overlap 23ms, 12ms και 6ms (50% σε κάθε περίπτωση).

4.7. Score Feature Extraction

Προκειμένου να μελετήσουμε τη ρυθμική αγωγή της ηχογράφησής μας, θα πρέπει καταρχήν να αποφασίσουμε πώς θα υπολογίζουμε την ταχύτητα της μουσικής ερμηνείας. Το πιο πρόσφορο και διαδεδομένο μέτρο για αυτό το σκοπό είναι το **Beats Per Minute (bpm)** που χρησιμοποιείται καθολικά από τους μουσικούς.

Η μέτρησή του προϋποθέτει να γνωρίζουμε τη διάρκεια κάθε ενός beat της ηχογράφησης. Η διάρκεια αυτή δεν είναι σταθερή και για να μπορέσουμε να την εξάγουμε, χρειαζόμαστε τις χρονικές στιγμές της έναρξης των beat (**beat onsets**)²⁰.

Η εύρεση των χρονικών στιγμών έναρξης των beat της ηχογράφησης περνάει μέσα από τον υπολογισμό των αντίστοιχων μεγεθών για τη synthesized παρτιτούρα. Το γιατί θα γίνει καλύτερα κατανοητό στη συνέχεια, μετά και το κεφάλαιο του alignment.

Η εξαγωγή των χαρακτηριστικών της παρτιτούρας περιλαμβάνει επιπλέον τον υπολογισμό του αριθμού των μέτρων της και του μήκους καθενός μέτρου σε beat, την εύρεση των διαφόρων σημειώσεων / οδηγιών που τη συνοδεύουν καθώς και τις θέσεις στις οποίες εντοπίζονται αυτές.

Σε όλες τις διαδικασίες απαιτείται η επεξεργασία ενός αρχείου τύπου MusicXML και γι' αυτό το λόγο παραθέτουμε στη συνέχεια μία εισαγωγή στα χαρακτηριστικά του format αυτού καθώς και στη σχετική με αυτό σύμβαση DOM.

4.7.1 MusicXML και DOM

Ο τύπος αρχείου **musicXML** [42] αναπτύχθηκε με βάση τη γλώσσα **XML(extensible markup language)** [43] με σκοπό τη αναπαράσταση διαδραστικών μουσικών κειμένων. Πρωτοεκδόθηκε τον Ιανουάριο του 2004 κι από τον Αύγουστο του 2011 κυκλοφορεί η πλέον πρόσφατη εκδοχή της, η 3.0 .

²⁰Στη μουσική το **beat** ορίζεται ως η βασικότερη μονάδα χρόνου, μία ρυθμική υποδιαίρεση που ορίζει τον παλμό εν μέσω ενός μουσικού μέτρου. Το **μέτρο** στη μουσική είναι το ελάχιστο τμήμα στο οποίο μπορούμε να χωρίσουμε ένα μουσικό έργο. Τα όριά του υποδεικνύονται με κατακόρυφες γραμμές και η διάρκειά του δίδεται στην αρχή του κομματιού. Ενδέχεται αυτή η διάρκεια να αλλάξει, σε αυτή την περίπτωση απαιτείται η δήλωση της καινούριας.

Το πιο σύνθητες είναι ένα beat να ισούται με ένα τέταρτο, ειδικά σε μέτρα που έχουν ένδειξη 2/4 , 3/4 , 4/4 κι άλλα πολλαπλάσια του τετάρτου. Σε άλλου τύπου μέτρα, όμως , το beat μπορεί να ισούται με ένα όγδοο, τρία όγδοα, δύο τέταρτα, κτλ.

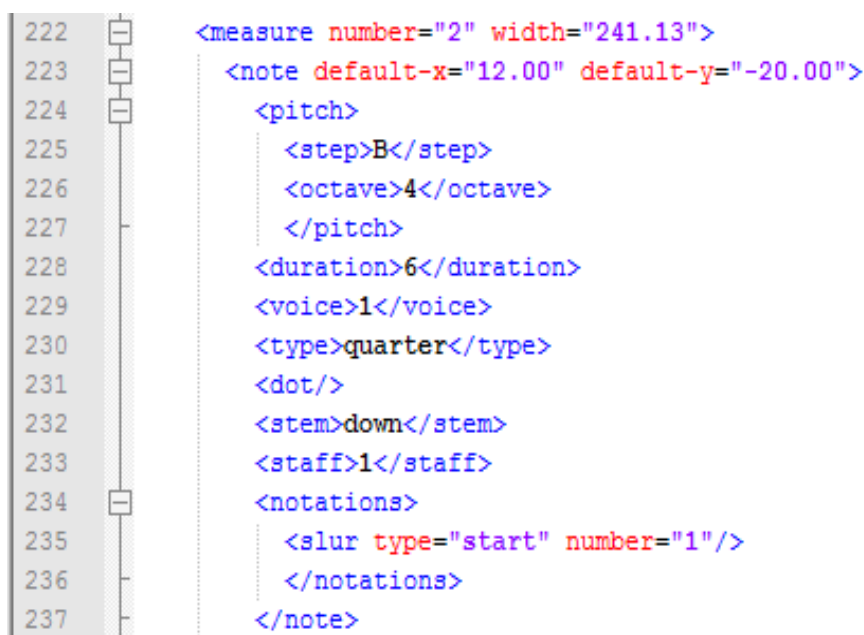
Τον Αύγουστο του 2013, υπολογίζεται ότι το πρότυπο musicXML υποστηρίζεται από τουλάχιστον 170 εφαρμογές [75] στις παρακάτω κατηγορίες:

- Μουσική σημειογραφία (Musescore, Finale, Sibelius κτλ)
- Αυτόματη αναγνώριση χαρακτήρων παρτιτούρας²¹ (SmartScore, PhotoScore, κτλ)
- Music Sequencing (Cubase, Logic Pro, Rosegarden, κτλ)

Βασικό πλεονέκτημα των αρχείων που ακολουθούν το πρότυπο musicXML (όπως εξάλλου κι όλα τα format που στηρίζονται στην XML) είναι η απλότητα και η αναγνωσιμότητα όχι μόνο από μηχανή αλλά κι από άνθρωπο . Εξάλλου, χαρακτηρίζονται από την ευκολία με την οποία υπόκεινται σε αυτόματη ανάλυση κι επεξεργασία από εφαρμογές όπως αυτές που αναφέρονται παραπάνω.

Η δενδρική δομή που ακολουθούν τα xml έγγραφα είναι βολική για την περίπτωση της μουσικής σημειογραφίας, αφού διευκολύνει την αναπαράσταση της ιεραρχικής κατανομής των επί μέρους της στοιχείων (κομμάτι → σύστημα → μέτρο → νότες). Επιπλέον, οι οδηγίες που συνοδεύουν αυτά τα τελευταία μέσα σε μια παρτιτούρα αποθηκεύονται ή σε **ετικέτες (tags)** ειδικού σκοπού ή με τη μορφή **ιδιότητας (attribute)** κάποιας ετικέτας.

Βλέπουμε παρακάτω κι ένα μικρό παράδειγμα, απόσπασμα από ένα musicXML αρχείο.



Εικόνα 36 Τμήμα .xml αρχείου μιας παρτιτούρας του musescore. Αντιστοιχεί στην πρώτη νότα του δεύτερου μέτρου, μία μεσαία σι (B4) διάρκειας 6/16 (τέταρτο παρεσιγμένο)

²¹ **Music Optical Character Recognition (Music OCR)**: Αυτόματη αναγνώριση χαρακτήρων παρτιτούρας, η εφαρμογή της αυτόματης αναγνώρισης χαρακτήρων κειμένου για τη μετατροπή προς- εκτύπωση - παρτιτούρας (πχ pdf) σε κάποια εκτελέσιμη ή / και επεξεργάσιμη μορφή (εν προκειμένω σε music xml).



Εικόνα 37 Η εικόνα που περιγράφεται από τον παραπάνω κώδικα, αποκομμένη από την υπόλοιπη παρτιτούρα

Μπορούμε στο ως άνω κείμενο να παρατηρήσουμε τα βασικότερα δομικά στοιχεία ενός xml αρχείου. Τέτοια είναι :

- Οι **ετικέτες (tags)**. Πρόκειται για στοιχεία σήμανσης που χωρίζονται σε **ετικέτες αρχής** (πχ < pitch> , σειρά 224) , **ετικέτες τέλους** (πχ </pitch> , σειρά 227) καθώς και σε **ετικέτες χωρίς περιεχόμενο**, τύπος ο οποίος δεν εμφανίζεται στο παράδειγμα μας.
- Τα **στοιχεία (elements)**. Κάθε απόσπασμα του κειμένου που ξεκινάει από μια ετικέτα αρχής και καταλήγει σε μία ετικέτα τέλους (πχ όλο το κομμάτι μεταξύ της σειράς 224 και της σειράς 227) ή αποτελείται αποκλειστικά και μόνο από μία ετικέτα χωρίς περιεχόμενο. Οτιδήποτε βρίσκεται εντός του στοιχείου συνιστά το **περιεχόμενό** του. Αυτό μπορεί να είναι **περιεχόμενο κειμένου (text content)** (πχ το B στη σειρά 225 ή το quarter στη σειρά 230) αλλά και κάποιο **στοιχείο – παιδί** (πχ τα <step> και <octave> για το στοιχείο – γονιό pitch).
- Τα **χαρακτηριστικά (attributes)**. Αποτελούνται από ένα ζεύγος **όνομα / τιμή (name / value)** και τοποθετούνται στην ετικέτα αρχής ενός στοιχείου (πχ number="2" εντός της ετικέτας measure, σειρά 222).

Τέλος, αναφέρουμε ότι το musicXML πρότυπο προβλέπει και την αποθήκευση αρχείων σε συμπιεσμένη μορφή με την κατάληξη **.mxl** περιορίζοντάς τα στο ένα εικοστό περίπου του αρχικού τους μεγέθους [76].

Για να μπορέσουμε να αντλήσουμε από το XML που έχει εξαγάγει το musescore τις πληροφορίες που μας χρειάζονται χρησιμοποιούμε τη συνάρτηση `xmlread` [77] η οποία επιστρέφει τα δεδομένα ενός XML σε ένα **Document Object Model (DOM, Μοντέλο Αντικειμένου Εγγράφου)**.

Το DOM είναι η πιο κοινή **σύμβαση (convention)** για την αναπαράσταση δεδομένων μέσα σε αρχεία HTML, XHTML και XML ως αντικείμενα. Πρόκειται για μία **προγραμματιστική διεπαφή (API, Application Programming Interface)** σχεδιασμένη για να μπορεί να χρησιμοποιηθεί με οποιαδήποτε γλώσσα προγραμματισμού και με οποιοδήποτε λειτουργικό σύστημα. Το DOM ορίζει τη λογική διάρθρωση μιας αναπαράστασης των εγγράφων μας, βάσει της οποίας κάθε μονάδα του XML απεικονίζεται ως ένα αντικείμενο – κόμβος εντός μιας δεντρικής δομής. Ο τρόπος με τον οποίο θα προσπελάσουμε και θα επεξεργαστούμε

τους κόμβους αυτούς (οι μέθοδοι δηλαδή της διεπαφής) ακολουθούν πρότυπα ορισμένα από το World Wide Web consortium.

Έστω το κομμάτι XML κώδικα στην εικόνα 36. Οι βασικότεροι τύποι κόμβων στους οποίους μεταφράζεται η περιεχόμενη πληροφορία εντός ενός DOM είναι οι ακόλουθοι:

- **Κόμβοι Στοιχείων (Element nodes)**- Όλα τα ονόματα ετικετών, εν προκειμένω:
 - `measure`
 - `note`
 - `pitch`
 - `step`
 - `octave`
 - `duration`
 - `voice`
 - `type`
 - `dot`
 - `stem`
 - `staff`
 - `notations`
 - `slur`

Το στοιχείο `note` είναι γονιός του κόμβου – παιδιού `pitch` αλλά και των δικών του παιδιών. Κάθε DOM ξεκινάει από έναν κόμβο στοιχείου που θα ορίζεται ως η ρίζα του DOM δέντρου. Για τα MusicXML που δημιουργούνται στο musescore, ο κόμβος αυτός παίρνει πάντα το όνομα `<score - partwise>`.

- **Κόμβοι κειμένου (Text nodes)** – Κάθε κόμβος κειμένου είναι παιδί κάποιου κόμβου στοιχείου. Για παράδειγμα, ο κόμβος `quarter` είναι παιδί του `type`.

- **Κόμβοι Χαρακτηριστικών (Attribute nodes)** – Χαρακτηρίζουν τον κόμβο στοιχείου με τον οποίο συνδέονται. Εμφανίζονται με τη μορφή ζεύγους όνομα – τιμή. Δεν είναι ούτε παιδιά ούτε γονιοί άλλων κόμβων.

Για την υλοποίηση του DOM χρησιμοποιούμε τη **xerces**, τη μητρική βιβλιοθήκη (native library) της matlab για την ανάλυση XML εγγράφων με χρήση java [78]. Θα χρησιμοποιήσουμε της εντολές της xerces για την προσθήκη, αφαίρεση και τροποποίηση κόμβων του DOM.

Ένας ακόμα τρόπος παρέμβασης στο MusicXML αρχείο δίχως την άμεση χρήση java είναι με χρήση της συνάρτησης `xml_read` της βιβλιοθήκης `xml_io_tools` [79] η οποία λαμβάνει τα δεδομένα που επιστρέφει η `xmlread` και τα γράφει σε μία δομή `struct`.

Αν, λοιπόν, δώσουμε στην `xml_read` ως είσοδο το XML αρχείο από τον κώδικα του οποίου έχει παρθεί το παράδειγμα στην Εικόνα 36, αυτή θα δώσει στην έξοδό της μία `struct`. Αν, χάριν του παραδείγματος, ονομάσουμε τη `struct` αυτή `s`, η νότα που περιγράφεται στο παράδειγμα

θα «αποθηκευτεί» στο field : `s.part.measure(2,1).note(1,1)`, ακολουθώντας τη δένδρική δομή που την καθιστά παιδί του `measure(2,1)` το οποίο με τη σειρά του έχει για γονιό το αντικείμενο `part κ. ο. κ.` Η `note` θα είναι επίσης struct με πεδία τα δικά της παιδιά (`pitch`, `duration`, `stem`, κτλ) αλλά και τα attributes που τη χαρακτηρίζουν (`default - x` και `default - y`).

Πράγματι, ξεκινώντας τη διαδικασία εξαγωγής των χαρακτηριστικών της παρτιτούρας, δίνουμε το xml αρχείο ως είσοδο στην `xml_read` (έχει προηγηθεί έλεγχος του `format` της παρτιτούρας, αν έχει κάποιον άλλο αποδεκτό τύπο πέραν του `.xml` φροντίζουμε για τη μετατροπή της) και, στη συνέχεια, επεμβαίνουμε στην προκύπτουσα struct για να λάβουμε την κάτωθι πληροφορία:

- Συνολική διάρκεια του κομματιού σε beat
- Διάρκεια καθενός μέτρου σε beat
- Μέτρα στα οποία εμφανίζονται σημειώσεις/οδηγίες που αναφέρονται στο tempo καθώς και το είδος των σημειώσεων αυτών.

Η χρησιμότητα των πρώτων δύο από τα παραπάνω θα φανεί καλύτερα στο κεφάλαιο που αναφέρεται στην περιγραφή της εκτέλεσης. Σχετικά με τις σημειώσεις μπαίνουμε σε περισσότερες λεπτομέρειες στην αμέσως επόμενη παράγραφο.

4.7.2 Εξαγωγή Μουσικών Σημειώσεων/Οδηγιών

Κάθε σημείωση/οδηγία δεσμεύεται στο μέτρο στο οποίο είχε (χειροκίνητα, στη διαδραστική παρτιτούρα) τοποθετηθεί. Μπορεί, λοιπόν, να εντοπιστεί στο xml αρχείο εντός του node που αντιστοιχεί στο μέτρο αυτό.

Οι σημειώσεις τοποθετούνται σε μία σειρά ενός **cell array (πίνακας κελιών)** ονόματι `tempo`, με κάποια μορφή που καθορίζεται από το είδος της σημείωσης, όπως ξεκαθαρίζεται και στη συνέχεια. Σε μία ακόμα σειρά κρατάμε το μέτρο στο οποίο βρέθηκε η εκάστοτε σημείωση. Στην περίπτωση που κάποιο μέτρο δε συνδέεται με καμία οδηγία, το αντίστοιχο κελί του πίνακα παραμένει κενό. Το τελικό μήκος του cell array είναι τουλάχιστον ίσο με το πλήθος των μέτρων (υπάρχει περίπτωση να υπερβαίνει τον αριθμό των μέτρων, αν σε κάποια μέτρα έχουν τοποθετηθεί περισσότερες της μίας σημειώσεις).

Οι σημειώσεις που αναζητούμε χωρίζονται στις εξής κατηγορίες:

a) «μετρονομική» οδηγία

Εδώ δίδεται ξεκάθαρα στον εκτελεστή η ταχύτητα με την οποία επιθυμεί ο συνθέτης να αποδοθεί το απόσπασμα, μετρημένη σε bpm. Αναλόγως με

το μήκος του beat του αποσπάσματος μπορεί να εμφανιστεί ως ♩ = ... ,
♪ = ... , κτλ.

Μία μετρονομική οδηγία αποτυπώνεται στο αντίστοιχο DOM σε έναν κόμβο με ετικέτα <metronome>.

```
76 | [ ] | <direction placement="above">
77 | [ ] |   <direction-type>
78 | [ ] |     <metronome parentheses="no">
79 | [ ] |       <beat-unit>quarter</beat-unit>
80 | [ ] |       <per-minute>80</per-minute>
81 | [ ] |     </metronome>
82 | [ ] |   </direction-type>
83 | [ ] | </direction>
```

Εικόνα 38 Απόσπασμα MusicXML που αντιστοιχεί στην οδηγία ♩ = 80 . Το ♩ ορίζεται εντός του <beat - unit> και ο αριθμός των bpm στο <per - minute>.

b) «ποιοτική» οδηγία tempo

Ως ποιοτικές οδηγίες εννοούμε μία σειρά από έννοιες οι οποίες (όπως σημειώνεται και στην εισαγωγή) ενέχουν ένα βαθμό ασάφειας. Συχνά αναφέρονται στη διάθεση με την οποία πρέπει να αποδοθεί το απόσπασμα (χαρούμενα, ζωδώς, πένθιμα, κτλ) αφήνοντας στον εκτελεστή την ευχέρεια να συμπεράνει ποιο είναι το κατάλληλο tempo εκτέλεσης. Η καλύτερη δυνατή προσέγγιση με χρήση κάποιου μετρήσιμου μεγέθους (bpm) μπορεί να γίνει επιλέγοντας ένα μέγιστο κι ένα ελάχιστο ποσό, δηλαδή ένα πάνω κι ένα κάτω όριο έξω από τα οποία μπορούμε να πούμε ότι ο εκτελεστής αποκλίνει από το επιθυμητό tempo.

Το musescore δίνει στο χρήστη τη δυνατότητα να κάνει μία επιλογή από μία λίστα οδηγιών τέτοιου τύπου και να την τοποθετήσει στην παρτιτούρα. Κάθε μία εξ αυτών συνοδεύεται από ένα συγκεκριμένο, σταθερό αριθμό από bpm ο οποίος υποδεικνύει και την ταχύτητα αναπαραγωγής της synthesized παρτιτούρας. Αυτή η ταχύτητα αναπαραγωγής όμως δεν έχει αντίκρισμα στην εργασία μας οπότε δε μας απασχολεί. Επιπλέον, όπως ήδη ξεκαθαρίστηκε, η σύνδεση μίας ποιοτικής ρυθμικής οδηγίας με έναν και μοναδικό αριθμό bpm θεωρείται εσφαλμένη.

Εξάλλου, μπορεί κανείς να πληκτρολογήσει μία τέτοιου τύπου οδηγία και να την τοποθετήσει στην παρτιτούρα με τη μορφή κειμένου. Τελικά, το ποια από τις δύο ανωτέρω δυνατότητες θα επιλέξει ο χρήστης δεν μας απασχολεί. Ο λόγος είναι ότι, σε αμφότερες τις περιπτώσεις, η σημείωση εμφανίζεται στο DOM σαν κόμβος κειμένου του οποίου οι άμεσοι πρόγονοι είναι πάντα οι ίδιοι.

```

80 <direction placement="above">
81 <direction-type>
82 <words>Allegro</words>
83 </direction-type>
84 <staff>1</staff>
85 </direction>

```

Εικόνα 39 Η οδηγία **Allegro**, όπως αποτυπώνεται σε ένα MusicXML που έχει εξαχθεί από το musescore.

Ο XML κώδικας που αποτυπώνεται στην Εικόνα 39, σε μια struct – έξοδο της συνάρτησης `xml_read` (έστω `s`) θα αποθηκεύεται ως `s.part.measure(5,1).direction.direction_DASH_type.words`. Αυτό που τελικά εμείς κρατάμε είναι η τιμή της καταχώρησης (εν προκειμένω 'Allegro', τύπου `char`), καθώς και το σημείο που αυτή εμφανίζεται (στο πέμπτο μέτρο, φαίνεται από το field `measure(5,1)`).

Ακολουθεί πίνακας που περιλαμβάνει το σύνολο των σημειώσεων αυτού του τύπου που λάβαμε υπόψη κατά την υλοποίηση της εφαρμογής.

Τα στοιχεία έχουν ληφθεί από το άρθρο **Tempo** της wikipedia [80].

Οδηγία	Πίνακας 2 Βασικές Ρυθμικές Σημειώσεις Ερμηνεία/Μετάφραση	Ελάχιστ α bpm	Μέγιστα bpm
Larghissimo	Πάρα πολύ αργά		19
Grave	Αργά κι επίσημα, τελετουργικά	20	40
Lento	Αργά	40	45
Largo	«Απλωμένα»	45	50
Larghetto	Σχετικά «απλωμένα»	50	55
Adagio	Αργά κι επιβλητικά (κυριολεκτικά «με άνεση»)	55	65
Adagietto	Σχετικά αργά	65	69
Andante moderato	Λίγο πιο αργά από ότι στο Andante	69	72
Andante	Σε ρυθμό περπατήματος	73	77
Andantino	Ελαφρώς πιο γρήγορο από το Andante	78	83

Οδηγία	Ερμηνεία/Μετάφραση	Ελάχιστ α bpm	Μέγιστα bpm
Marcia moderato	«Μετριοπαθώς» και σε εμβατηριακό ύφος	83	85
Moderato	«Μετριοπαθώς»	86	98
Allegretto	Μέτρια γρήγορα	98	109
Allegro	Γρήγορα και «λαμπερά»	109	132
Vivace	Γρήγορα και με ζωντάνια	132	140
Vivacissimo	Πολύ γρήγορα και με ζωντάνια	140	150
Allegroissimo	Πολύ γρήγορα	150	167
Presto	Υπερβολικά γρήγορα	168	177
Prestissimo	Πιο γρήγορα κι από το Presto	178	

Στο κελί της πρώτης σειράς του cell array tempo που αντιστοιχεί στο μέτρο όπου εντοπίζεται η σημείωση τοποθετείται ένα cell που περιέχει δύο integer, έναν ίσο με το μέγιστο κι έναν ίσο με το ελάχιστο δυνατό πλήθος bpm.

c) a tempo

Κυριολεκτικά «στο tempo» ή, πιο περιγραφικά, «πίσω στο tempo». Εδώ εννοείται ότι υπάρχει ένα «κυρίαρχο» tempo (δεν υπάρχουν πολλές ή/και συχνές αλλαγές), αυτό το οποίο δίδεται στην αρχή του αποσπάσματος. Θα έπεται κάποιας άλλης εντολής ρυθμικής διαφοροποίησης, πριν από την οποία μπορούμε να εντοπίσουμε το tempo στο οποίο πρέπει η μουσική να επιστρέψει. Για εμάς, λοιπόν, συμπίπτει με τον όρο tempo primo («πίσω στο πρώτο tempo»)²². Όπως και οι σημειώσεις της προηγούμενης κατηγορίας, εμφανίζεται στον κώδικα μέσα σε μια ετικέτα <words>.

Στο κελί της πρώτης σειράς του cell array tempo που αντιστοιχεί στο μέτρο όπου εντοπίζεται η σημείωση θα τοποθετείται το πλήθος των bpm στα οποία αναμένεται να επιστρέψει η μουσική εκτέλεση.²³

²²Η σύμβαση που διατυπώνουμε σε αυτό το σημείο, ότι, δηλαδή, ο όρος a tempo εμφανίζεται σε αποσπάσματα όπου κυριαρχεί ένα συγκεκριμένο tempo το οποίο μάλιστα θα είναι και το αρχικό, αποκτά μεγαλύτερη ισχύ αν αναλογιστούμε ότι η εργασία αυτή αναφέρεται προπάντων σε μουσικά αποσπάσματα μικρής χρονικής διάρκειας.

²³Το tempo στο οποίο ζητάμε από τον εκτελεστή να επανέλθει θα είναι αυτό που τηρούσε ο ίδιος πριν προβεί σε ρυθμικές διαφοροποιήσεις. Αυτό στο οποίο στεκόμαστε όταν βρεθούμε μπροστά σε μία εντολή a tempo είναι η πράξη της επιστροφής στο κυρίαρχο tempo όπως αυτό έχει προκύψει από την εκτέλεση. Αν αυτό το τελευταίο συνάδει ή όχι με τις οδηγίες της παρτιτούρας κρίνεται σε άλλη φάση, με τον τρόπο που εξηγείται στα a) και b).

d) Οδηγίες απότομης αύξησης ή μείωσης tempo

Πρόκειται για δύο συγκεκριμένες εντολές: **più mosso** και **meno mosso**, οι οποίες μεταφράζονται ως πιο γρήγορα και λιγότερο γρήγορα (κυριολεκτικά με περισσότερη και με λιγότερη κίνηση). Δε σχετίζονται με κάποιο αριθμό bpm, απλώς καθοδηγούν τον εκτελεστή στην ξαφνική, δίχως ενδιάμεσα στάδια αλλαγή της ταχύτητας εκτέλεσης προς τα πάνω ή προς τα κάτω.

Στο κελί της πρώτης σειράς του cell array tempo που αντιστοιχεί στο μέτρο όπου εντοπίζεται η σημείωση θα τοποθετείται το string 'piu mosso' ή 'meno mosso' κατά περίπτωση.²⁴

e) Οδηγίες σταδιακής αύξησης ή μείωσης tempo

Εδώ δίδεται στον μουσικό η εντολή να αυξήσει ή να μειώσει το tempo, όχι ξαφνικά αλλά σταδιακά, βήμα – βήμα. Αυτού του τύπου οι οδηγίες, όπως κι αυτές που περιγράφονται στα b) ,c) και d) συνδέονται με κόμβους κειμένου.

Παραθέτουμε παρακάτω όλες τις σημειώσεις σταδιακής αλλαγής tempo που λήφθηκαν υπόψη, χωρισμένες σε αυτές που δηλώνουν αύξηση και σε αυτές που δηλώνουν μείωση tempo. Σε κάθε μία κατηγορία θεωρούνται μεταξύ τους ισοδύναμες (παρόλο που υπάρχουν λεπτομέρειες που ενίοτε τις διαφοροποιούν, θεωρούνται αμελητέες). Αναφέρονται με τα πλήρη τους ονόματα, αλλά συχνά συναντώνται και σε συντομογραφία, περίπτωση για την οποία έχει επίσης ληφθεί πρόνοια. Επιπλέον, συναντώνται ορισμένες φορές μαζί με τη διευκρίνηση poco a poco που κυριολεκτικά μεταφράζεται ως «σιγά – σιγά», δεν επηρεάζει όμως με κάποιο τρόπο τη δική μας ανάλυση.

Αύξηση ταχύτητας:

- Stringendo
- Precipitando
- Accelerando

Μείωση ταχύτητας:

- Allargando
- Calando
- Lentando
- Rallentando
- Ritardando
- Ritenuto

²⁴Να σημειωθεί ότι, αν και η σωστή γραφή είναι più mosso, κρατήσαμε το piu mosso για την αποφυγή κακοτοπιών.

Στο κελί της πρώτης σειράς του cell array tempo που αντιστοιχεί στο μέτρο όπου εντοπίζεται η σημείωση θα τοποθετείται ένα string με όποια από τις παραπάνω εντολές έχουμε συναντήσει.

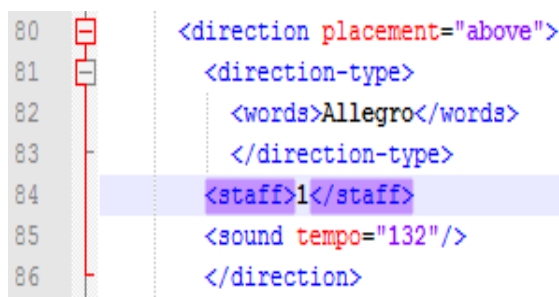
4.7.3 Beat onset extraction

Οι στιγμές κατά τις οποίες εκκινούν τα beat της synthesized παρτιτούρας θα υπολογιστούν μέσω των δεδομένων που λαμβάνουμε από το MIDI αρχείο. Προτού, όμως, προχωρήσουμε στην επεξεργασία του MIDI, πρέπει και να ξεκαθαρίσουμε ποια θα είναι η χρονική διάρκεια κάθε ενός beat.

4.7.3.1 Σταθεροποίηση διάρκειας beat synthesized παρτιτούρας

Οι scorewriter εφαρμογές δίνουν στο χρήστη τη δυνατότητα να επιλέξει ποια θα είναι η ταχύτητα με την οποία θα αναπαράγεται η synthesized παρτιτούρα (playback speed). Ενδέχεται, λοιπόν, το αρχείο παρτιτούρας το οποίο επεξεργαζόμαστε να αλλάζει tempo, ή, αναδιατυπώνοντας, να έχει beat μεταβλητής διάρκειας. Κάτι τέτοιο δημιουργεί εμπόδια στην ανάλυσή μας, η οποία απλοποιείται σημαντικά αν έχουμε σταθερή διάρκεια για όλα τα beat και κατ' επέκταση την ευχέρεια να αντιμετωπίσουμε με έναν κοινό τρόπο όλη τη synthesized παρτιτούρα, από την αρχή έως το τέλος. Στο επόμενο βήμα, λοιπόν, φροντίζουμε γι' αυτή την απλοποίηση παρεμβαίνοντας στο MusicXml αρχείο που την περιγράφει. Βέβαια, το αρχείο που μας δίδεται στην είσοδο μπορεί να έχει κάποια άλλο format, πχ .mscz. Σε αυτή την περίπτωση, φροντίζουμε για την μετατροπή του σε .xml μέσω του scorewriter. Όλα αυτά τα βήματα, όπως αναφέρεται και στην παράγραφο 4.1., προηγούνται του synthesis. Παρόλα αυτά προτιμήθηκε η σχετική παράγραφος να τοποθετηθεί σε αυτό το σημείο, για λόγους συνέχειας.

Οι όποιες εντολές ρυθμικών εναλλαγών αποτυπώνονται στο DOM μέσα σε nodes με ετικέτα sound. Σε περίπτωση που δεν έχουν δοθεί τέτοιες εντολές, η synthesized παρτιτούρα έχει την default σταθερή ταχύτητα των 120 bpm. Αρκεί η αφαίρεση αυτού του τύπου των κόμβων ώστε να εξασφαλίσουμε ότι η (νέα) synthesized παρτιτούρα θα έχει σταθερό tempo καθ' όλη τη διάρκειά της.



Εικόνα 40 Στη σειρά 85 φαίνεται ένας κόμβος τύπου sound, από αυτούς που αφαιρούμε στο τρέχον βήμα. Ο συγκεκριμένος δηλώνει ότι η synthesized παρτιτούρα θα εκτελείται σε tempo $J = 132$

Αυτό θα επιτευχθεί με μία σειρά από απλές εντολές σε java, παρμένες από τη βιβλιοθήκη xerces.

4.7.3.2 Beat onsets μέσω του MIDI

Αφού πλέον έχουμε εξασφαλίσει ότι τα beat έχουν σταθερή διάρκεια, θα λάβουμε τη διάρκεια αυτή από το MIDI της παρτιτούρας. Για να γίνει αυτό, θα πρέπει πρώτα να μεταφέρουμε τα MIDI δεδομένα σε μια δομή αναγνωρίσιμη από τη matlab. Η ανάγκη αυτή εξυπηρετείται μέσω της συνάρτησης `midi2nmat` της βιβλιοθήκης `miditoolbox` [81]. Η συνάρτηση αυτή θα μας δώσει στην έξοδό της έναν πίνακα ο οποίος, ανάμεσα σε άλλα, θα περιλαμβάνει και τους χρόνους έναρξης και λήξης κάθε νότας (note onset και note offset) της παρτιτούρας μετρημένους τόσο σε beat όσο και σε δευτερόλεπτα. Ο λόγος της διαφοράς δύο διαδοχικών onset υπολογισμένα σε beat προς την ίδια ποσότητα σε δευτερόλεπτα θα μας δώσει τη (σταθερή) διάρκεια ενός beat σε δευτερόλεπτα.

Μια μαθηματική διατύπωση της παραπάνω πρότασης για τις νότες υπ' αριθμό x και $x+1$ είναι η εξής:

$$(beat\ duration)_s = \frac{no_b^{(x+1)} - no_b^{(x)}}{no_s^{(x+1)} - no_s^{(x)}}$$

όπου no τα note onsets, με τους δείκτες b και s υποδεικνύουν μέτρηση σε second και beats αντίστοιχα.

Έτσι, ξεκινώντας από το πρώτο beat και προχωρώντας κατά ένα διάστημα $(beat\ duration)_s$ κάθε φορά, για το πλήθος των beat που υπολογίσαμε προηγουμένως (μέσω του MusicXml) ότι θα έχει το αρχείο μας, καταλήγουμε να έχουμε τη χρονική στιγμή έναρξης κάθε ενός beat της synthesized παρτιτούρας, μετρημένο σε second.

5. Alignment

Σκοπός μας σ' αυτό το στάδιο της εργασίας είναι η χρονική αντιστοίχιση (alignment, ευθυγράμμιση) των δύο αρχείων εισόδου, της synthesized παρτιτούρας ενός απλού, σύντομου έργου για ένα μόνο μουσικό όργανο και μιας ηχογράφησης του έργου αυτού.

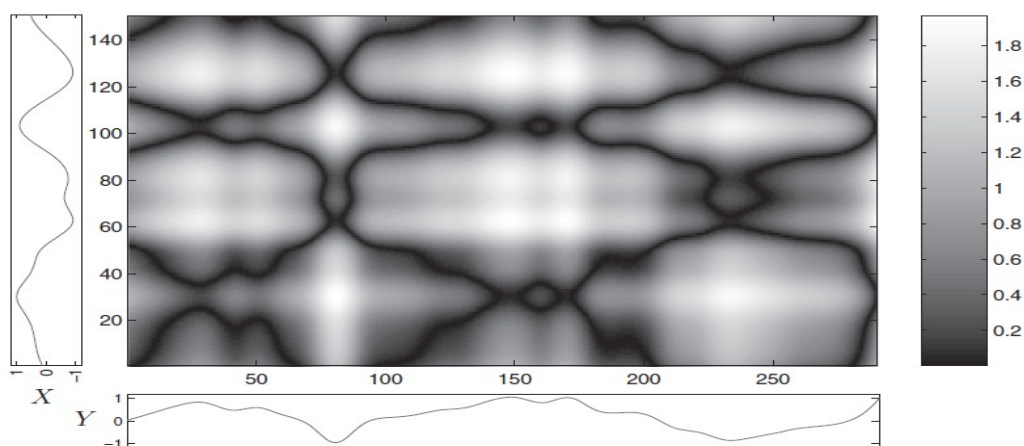
Η ευθυγράμμιση είναι το πρώτο προαπαιτούμενο για την επιμέρους σύγκριση των δύο αρχείων, εφόσον μας επιτρέπει να γνωρίζουμε σε πιο ακριβώς στιγμιότυπο της ηχογράφησης του εκάστοτε αποσπάσματος προκύπτει το κάθε τι που σημειώνεται επί της παρτιτούρας.

Η τεχνική που χρησιμοποιείται είναι γνωστή ως **Dynamic Time Warping (DTW)**. Αυτό που επιτυγχάνουμε με την τεχνική αυτή είναι η «στρέβλωση» (**warping**) του ενός εκ των δύο αρχείων ώστε να «προσαρμόζεται χρονικά» στο δεύτερο [82][83][84].

5.1. Similarity Matrix

Η διαδικασία υπολογισμού των short time φασματικών χαρακτηριστικών καταλήγει στην παραγωγή δύο χρονικά εξαρτημένων ακολουθιών $X \equiv (x_1, x_2, \dots, x_N)$ μήκους $N \in \mathbb{N}$ και $Y \equiv (y_1, y_2, \dots, y_M)$ μήκους $M \in \mathbb{N}$. Καθένα από τα στοιχεία τους είναι ένα διάνυσμα το οποίο περιέχει την ενέργεια κάθε bin. Το μήκος των διανυσμάτων είναι B (όπου B ο αριθμός των bins συχνοτήτων του Fourier Transform $B = (\text{window size})/2$) ή 12 στην περίπτωση που έχουμε chromagram.

Για να βρούμε τη βέλτιστη αντιστοίχιση μεταξύ των δύο αρχείων πρέπει καταρχήν να κατασκευάσουμε έναν **similarity matrix**, κάτι το οποίο προϋποθέτει τον ορισμό ενός μέτρου κόστους ή αλλιώς απόστασης.



Εικόνα 41: Πίνακας Ομοιότητας για τις πραγματικές ακολουθίες X και Y με χρήση της απόστασης Manhattan.

Έστω λοιπόν $F \subseteq \mathbb{R}^B$ ο χώρος των χαρακτηριστικών διανυσμάτων των παραπάνω ακολουθιών. Το μέτρο απόστασης δύο διανυσμάτων $x, y \in F$ είναι μια συνάρτηση $c: F \times F \rightarrow \mathbb{R}_{\geq 0}$. Το $c(x, y)$ είναι μικρό εάν τα x, y είναι «κοντά» το ένα με το άλλο και μεγάλο εάν είναι «μακριά».

Έχουμε χρησιμοποιήσει δύο ενδεχόμενα μέτρα απόστασης: την **ευκλείδεια** και την **συνημιτονική απόσταση** :

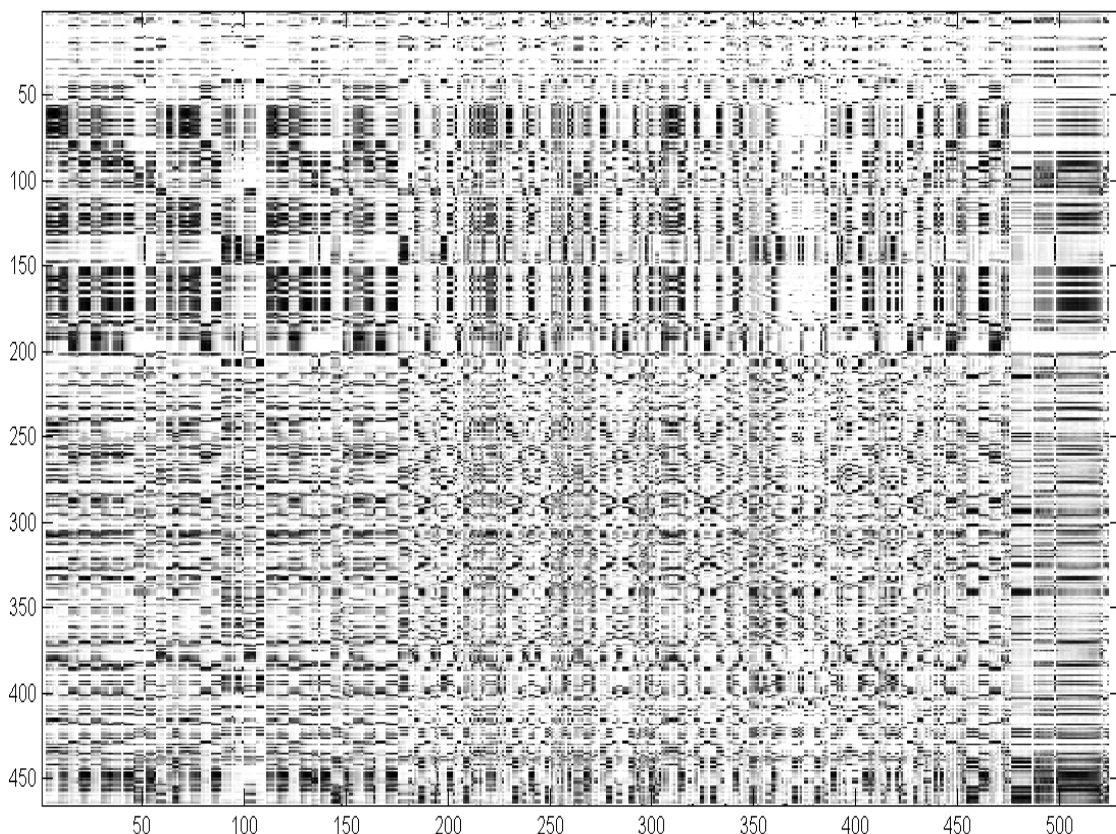
$$ED = \sqrt{\sum_{i=1}^k (a_i - b_i)^2}, CD = a * b / (||a|| * ||b||), \text{ όπου } a, b \in \mathbb{R}^k$$

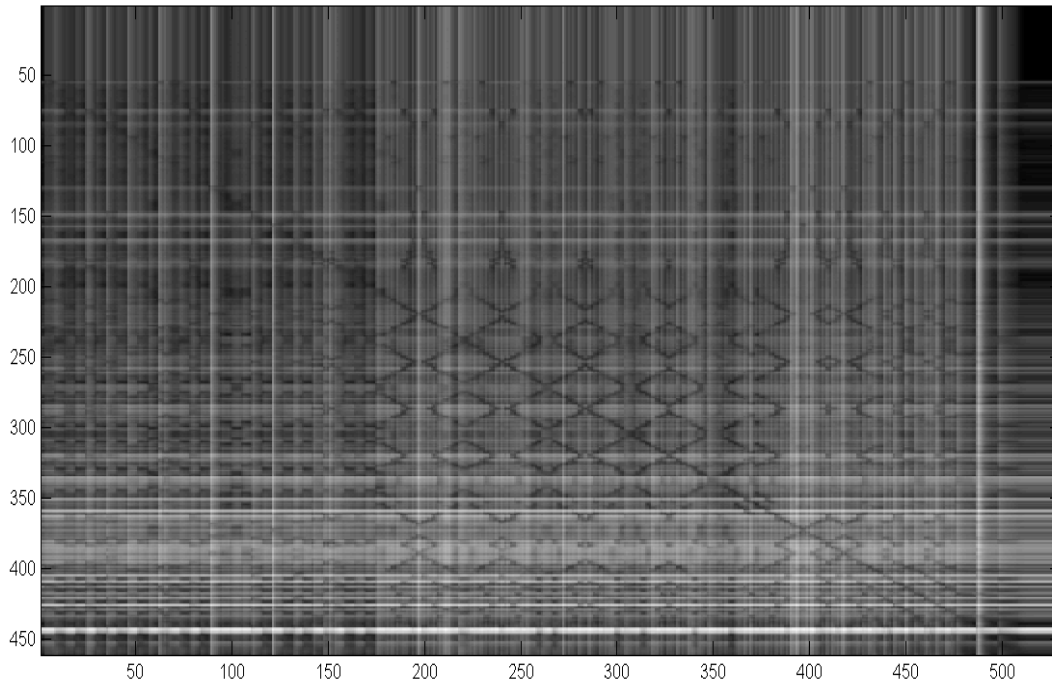
Εάν, λοιπόν, υπολογίσουμε την απόσταση μεταξύ όλων των πιθανών ζευγών μεταξύ των δύο ακολουθιών X και Y προκύπτει ο **Πίνακας Ομοιότητας (Similarity Matrix)**:

$$SM \in \mathbb{R}^{N \times M} \text{ με } SM(n, m) \equiv c(x_n, y_m)$$

Με δυο λόγια, χτίζοντας το similarity matrix, έχουμε έναν πίνακα που αποτυπώνει πόσο όμοιο είναι κάθε πιθανό ζεύγος frames μεταξύ των δύο ήχων.

Η υλοποίηση γίνεται μέσω των συναρτήσεων `simmx_euc` ή `simmx_cos`, αναλόγως με το μέτρο απόστασης που έχουμε επιλέξει.





Εικόνα 1 Πίνακας Ομοιότητας για απόσπασμα από τη σονάτα για πιάνο σε Ντο μείζονα του W. A. Mozart με χρήση (επάνω)chromagram και συνημιτονικής απόστασης και (κάτω)STFT και ευκλείδειας απόστασης. Διαφαιίνεται ήδη το optimal warping path, πρόκειται για τη διαγώνιο.

5.2. Lowest – Cost Path Calculation

Στη συνέχεια αυτό που μας απασχολεί είναι η εύρεση της αντιστοίχισης μεταξύ των στοιχείων των X και Y η οποία θα έχει το ελάχιστο συνολικό κόστος. Η απαίτηση του ελάχιστου κόστους μας προϊδεάζει για την αναζήτηση ζευγών με μικρή μεταξύ τους απόσταση. Συμπεραίνεται άμεσα ότι για την εξαγωγή της ζητούμενης αντιστοιχίας πρέπει να αναφερθούμε στον Πίνακα Ομοιότητας. Πιο συγκεκριμένα, θέλουμε να βρούμε ποια διαδρομή θα έπρεπε να ακολουθήσουμε επί του SM προκειμένου να εξάγουμε μια ακολουθία ελάχιστου αθροιστικού κόστους.

Τη διαδρομή αυτή ονομάζουμε **optimal warping path**.

Το μονοπάτι στρέβλωσης (warping path) είναι μια ακολουθία p , όπου

$$p = (p_1, p_2, \dots, p_L) \text{ με } p_l = (n_l, m_l) \in [1:N] \times [1:M] \text{ για } l \in [1:L]$$

Ορίζει, ουσιαστικά, το p , μία αντιστοίχιση μεταξύ των $x \equiv (x_1, x_2, \dots, x_N)$ και $y \equiv (y_1, y_2, \dots, y_M)$ αναθέτοντας το στοιχείο x_{n_l} του X στο στοιχείο y_{m_l} του Y .

Η p είναι αύξουσα ($n_1 \leq n_2 \leq \dots \leq n_L$ και $m_1 \leq m_2 \leq \dots \leq m_L$) ενώ επιπλέον ισχύει : $p_1 = (1, 1)$ και $p_L = (N, M)$ (οριακή συνθήκη).

Η συνθήκη μονοτονίας συνεπάγεται ότι, εάν ένα στοιχείο του X προηγείται ενός δεύτερου, τότε τα στοιχεία του Y με τα οποία αυτά ευθυγραμμίζονται θα πρέπει να ακολουθούν την ίδια σειρά κι αντιστρόφως.

Η οριακή συνθήκη αποτυπώνει την απαίτηση τα πρώτα και τα τελευταία στοιχεία του X να ευθυγραμμίζονται με τα αντίστοιχα του Y .

Θα πρέπει να ορίσουμε κι έναν περιορισμό για το μέγεθος του βήματος από p_i σε p_{i+1} : $p_{i+1} - p_i \in (1,0), (0,1), (1,1)$ για $i \in [1 : L-1]$

Η τελευταία αυτή συνθήκη, την οποία όπως θα δούμε μπορεί να παραλλάξουμε, εκφράζει το εξής: κανένα στοιχείο των X και Y δεν μπορεί να παραληφθεί και, επιπλέον, κάθε ζεύγος της ακολουθίας p είναι μοναδικό.

Εμάς όμως μας ενδιαφέρει να καταλήξουμε σ' ένα **ιδανικό (optimal) μονοπάτι**, το οποίο θα έχει το ελάχιστο δυνατό συνολικό κόστος c_p , όπου

$$c_p(X, Y) \equiv \sum_{i=1}^L c(x_{n_i}, y_{m_i})$$

Για την εξαγωγή του optimal warping path θα μπορούσαμε να συγκρίνουμε όλες τις διαδρομές εν μέσω του similarity matrix που ικανοποιούν τους παραπάνω περιορισμούς και να επιλέξουμε αυτή με το μικρότερο κόστος. Η διαδικασία θα ήταν εξαιρετικά χρονοβόρα, με την πολυπλοκότητά της να αυξάνει εκθετικά με την αύξηση των M, N . Αντ' αυτού, ακολουθούμε ένα αλγόριθμο δύο σταδίων ο οποίος στηρίζεται στο δυναμικό προγραμματισμό και παρατίθεται στη συνέχεια.

Στο πρώτο στάδιο, δημιουργούμε μέσω του Πίνακα Ομοιότητας έναν ακόμα πίνακα D , διαστάσεων $(N+1) \times (M+1)$, τον **Αθροιστικό Πίνακα Ομοιότητας**, ο οποίος στη βασική του μορφή ικανοποιεί τις παρακάτω συνθήκες:

$$D(n, 1) = \infty, n \in [1, N+1]$$

$$D(1, m) = \infty, m \in [1, M+1]$$

$$D(n, m) = \min\{D(n-1, m-1), D(n-1, m), D(n, m-1)\} + c(x_n, y_m)$$

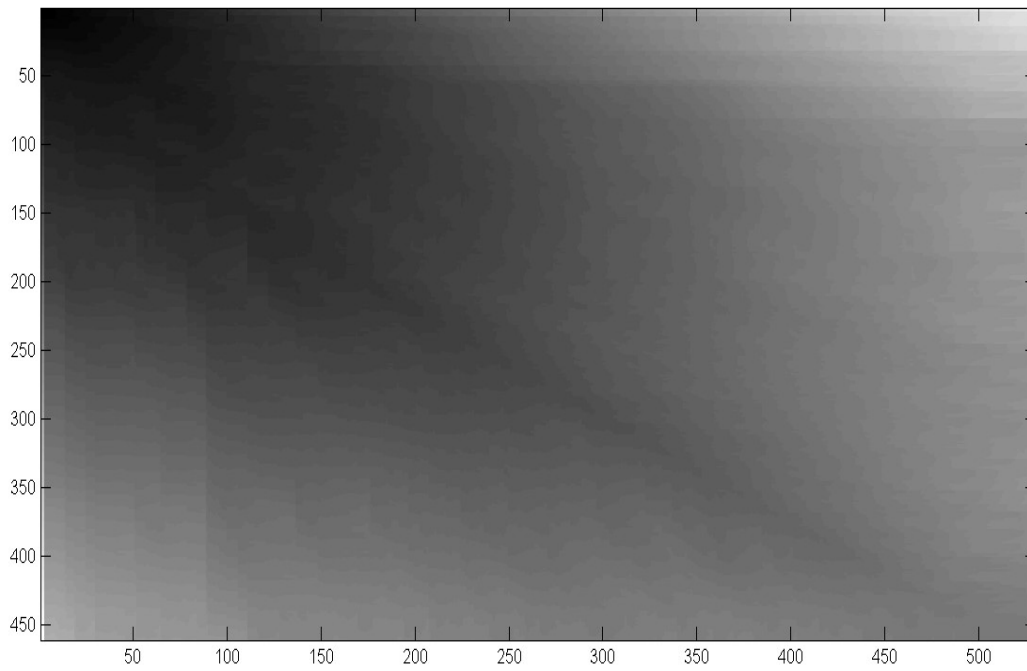
$$1 < n \leq N+1 \text{ και } 1 < m \leq M+1$$

Στο δεύτερο στάδιο, υπολογίζουμε το optimal μονοπάτι $p_{opt} = (p_1, p_2, \dots, p_L)$.

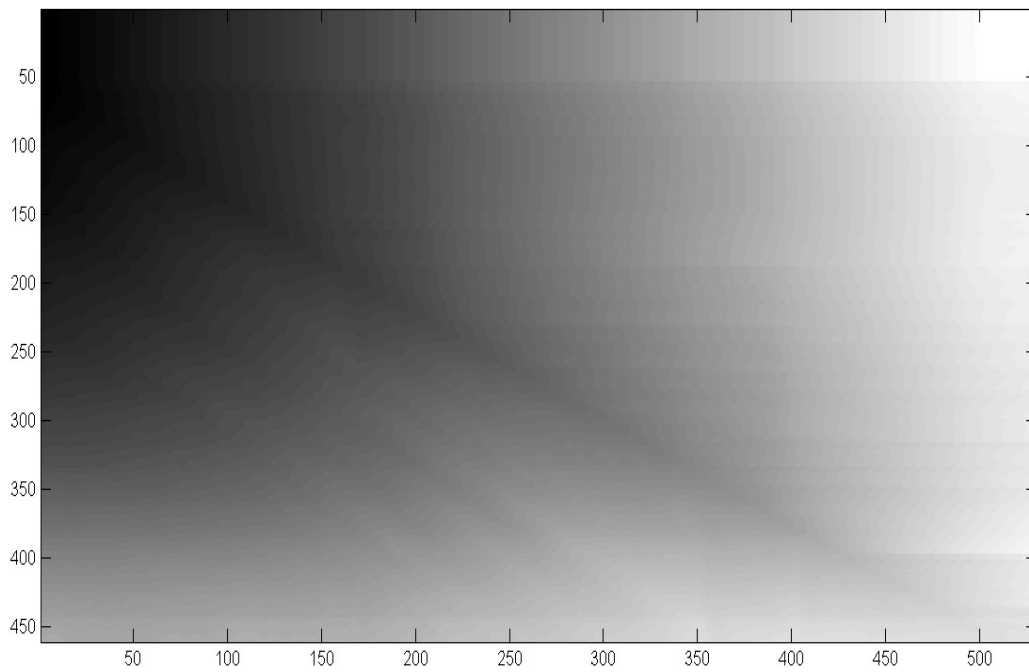
Ξεκινάμε από $p_L = (N, M)$ και συνεχίζουμε «προχωρώντας ανάποδα», με τα στοιχεία του path να υπολογίζονται ως εξής:

$$p_{l-1} \equiv \begin{cases} (1, m-1), n=1 \\ (n-1, 1), m=1 \\ \operatorname{argmin}\{D(n-1, m-1), D(n-1, m), D(n, m-1)\}, \text{ παντού αλλού} \end{cases}$$

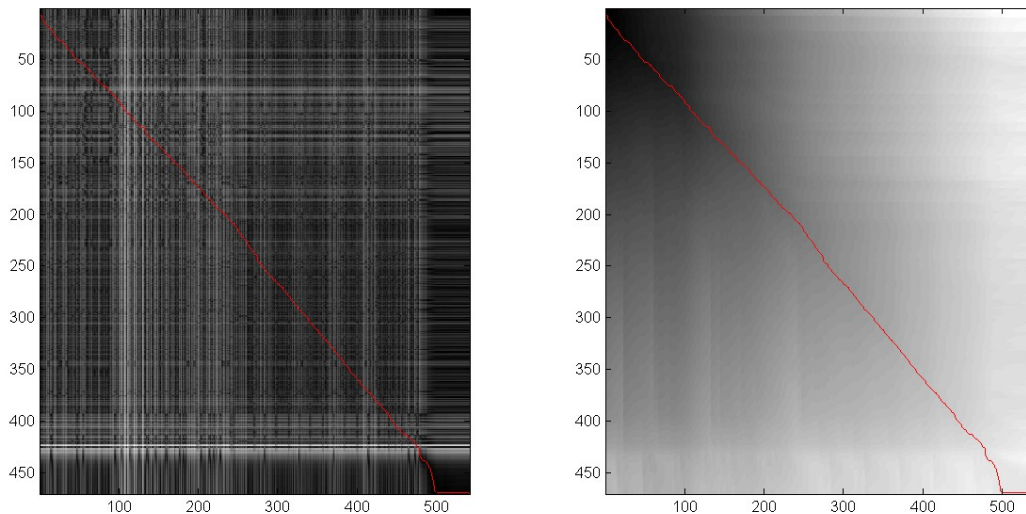
(α)



(β)



Εικόνα 43 Αθροιστικός Πίνακας Ομοιότητας για απόσπασμα από τη σονάτα για πιάνο σε Ντο μείζονα του W. A. Mozart με χρήση (α)chromagram και συνημιτονικής απόστασης και (β)STFT και ευκλείδειας απόστασης.

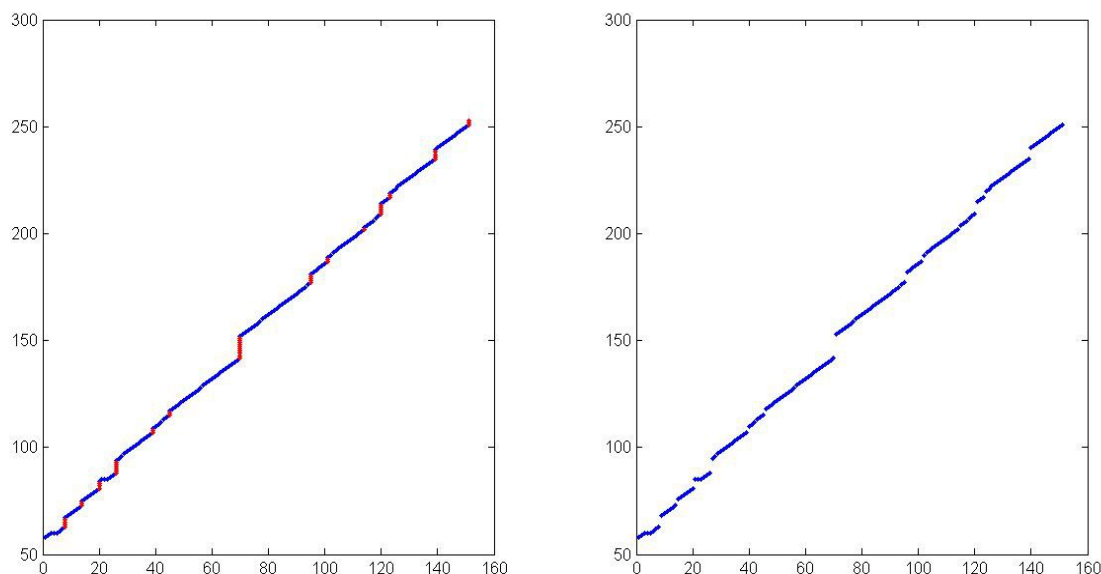


Εικόνα 44 Ο Πίνακας Ομοιότητας κι ο Αθροιστικός Πίνακας Ομοιότητας για απόσπασμα από το Dr Gradus Ad Parnassum του Claude Debussy. Με κόκκινο, το Optimal Warping Path

5.3. Alignment Path

Στο τελικό στάδιο της ευθυγράμμισης, θα προχωρήσουμε το «συνταίριασμα» των δύο φασματικών αναπαραστάσεων (έστω D1 και D2) ένα βήμα πιο πέρα. Κάθε ακολουθία θα επιτελεί διαφορετικό ρόλο, καθώς οι δείκτες της μίας (έστω D1) θα προσαρμόζονται σ' αυτούς της δεύτερης (έστω D2). Αν αφαιρέσουμε τις όποιες επαναλήψεις από τα στοιχεία n_i (δείκτες του D2) καθώς και τους δείκτες του D1 που αντιστοιχούνται στις επαναλήψεις αυτές, θα προκύψει μία παραλλαγμένη εκδοχή της ακολουθίας ζευγών $p_{opt} = (p_1, p_2, \dots, p_L)$ με λιγότερα στοιχεία. Οι εναπομείνουσες τιμές του n_i θα είναι όλοι οι φυσικοί αριθμοί από το 1 έως το N (όπου N το μήκος της D2) και μπορούν να ειπωθούν πλέον ως δείκτες ενός διανύσματος με μήκος N (αντί για L) και τιμές το εναπομείναν υποσύνολο του m_i (δείκτες του D1). Το διάνυσμα αυτό θα το ονομάζουμε **alignment path**:

$\text{alignmentpath}[n]: \mathbb{N} \rightarrow \mathbb{N}$ με $n=1, 2, \dots, N$, όπου $N \in \mathbb{N}$ το μήκος της D2 και $\text{alignmentpath}[n] \subseteq m_i$, με $m_i \in [1, M]$, όπου $M \in \mathbb{N}$ το μήκος της D1



Εικόνα 1 Σύγκριση των πρώτων τιμών του lowest-cost warping path (αριστερά) με αυτές του alignment path (δεξιά) μιας ηχογράφησης. Με κόκκινο επί του lowest-cost path σημειώνονται οι duplicate τιμές που αφαιρούνται κατά την κατασκευή του alignment path.

Όλα τα βήματα για την υλοποίηση του alignment εκτελούνται μέσω της συνάρτησης aligner. Η aligner λαμβάνει στην είσοδό της την ηχογράφηση, την παρτιτούρα, το είδος της φασματικής απεικόνισης που θα χρησιμοποιηθεί καθώς και το μήκος παραθύρου και το hop size που θα έχει η τελευταία.

Warped Version Synthesis

Σε αυτό το σημείο θέλουμε να εφαρμόσουμε μία πρακτική αξιολόγηση των αποτελεσμάτων της παραπάνω επεξεργασίας. Τροποποιούμε την ηχητική μορφή της παρτιτούρας ως προς τον τρόπο που διατάσσεται μέσα στο χρόνο, επηρεάζοντας όσο το δυνατόν λιγότερο τα φασματικά της χαρακτηριστικά. Αυτό επιτυγχάνεται μέσω του phase vocoder, μιας μεθόδου που εκτείνει και συμπιέζει χρονικά ένα αρχείο ήχου και υλοποιείται μέσω της συνάρτησης pvsample. Τα frames του σήματός μας που θα υποστούν την επεξεργασία αυτή θα προσδιοριστούν από το optimal warping path.

Καταλήγουμε, λοιπόν, να έχουμε μία warped εκδοχή της synthesized παρτιτούρας η οποία, όταν η ηχογράφηση προηγείται χρονικά, υπόκειται σε επιτάχυνση ενώ, όταν η ηχογράφηση «καθυστερεί», αναγκάζεται να «περιμένει».

Κρίνουμε, τελικώς, προσεγγιστικά δια της ακοής, εάν τα ηχητικά αρχεία που παράγουμε είναι πράγματι και σε ποιο βαθμό χρονικά συντονισμένα.

Μία ακριβής ποσοτική αξιολόγηση θα απαιτούσε να σημειώσουμε την αρχή και το τέλος κάθε νότας της ηχογράφησης με βάση το alignment και τελικά να υπολογίσουμε τη μέση διαφορά μεταξύ τους σε milliseconds (εν ολίγοις το μέσο σφάλμα του alignment). Αυτού του είδους η αξιολόγηση έχει γίνει ήδη αρκετά εκτενώς και δεν είναι στις προτεραιότητες της εργασίας να επαναληφθεί. Η «δι' ακοής» αξιολόγηση επαρκεί για να βεβαιωθούμε ότι έχουμε υλοποιήσει σωστά τον αλγόριθμο.

6. Περιγραφή Εκτέλεσης

Όπως έχουμε αναφέρει και προηγουμένως, το tempo της ηχογράφησης μας θα μετράται σε μονάδες Beats per Minute (bpm). Πιο συγκεκριμένα, θα υπολογίζουμε τα **beats per minute κάθε ενός μέτρου της ηχογράφησης**. Έτσι, κατατάχνουμε το κομμάτι σε μονάδες με νόημα για τη μουσική ορολογία (αντίθετα για παράδειγμα με τυχαίες ποσότητες από beat) οι οποίες μας επιτρέπουν ένα καλό βαθμό ακρίβειας: αν επιλέγαμε να υπολογίσουμε το tempo με βάση κάθε ένα beat ή κάθε μία νότα ξεχωριστά θα διακινδυνεύαμε την ακρίβεια της μέτρησης αφού, σε αποσπάσματα τα οποία, βάσει της παρτιτούρας, αναμένεται να έχουν σταθερό tempo, μικρές ρυθμικές αποκλίσεις από νότα σε νότα είναι σχετικά αναμενόμενες. Αντιθέτως, τέτοιες ανομοιότητες σε διαδοχικά μέτρα πολλαπλασιάζουν το σφάλμα, «θολώνουν» τον παλμό και γίνονται αισθητές στον ακροατή. Τουτέστιν, χρησιμοποιώντας ως βάση μέτρησης τα μέτρα μπορούμε να είμαστε πιο ασφαλείς για τα συμπεράσματά μας. Επιπλέον, αν διαλέγαμε περισσότερα του ενός μέτρα θα αφήναμε ανοιχτό το ενδεχόμενο να παραβλεφθεί κάποια αλλαγή στο tempo.

Αυτό που απαιτείται, λοιπόν, σε αυτή τη φάση, είναι να απομονώσουμε κάθε ένα μέτρο και να υπολογίσουμε τη διάρκειά του σε second. Σε αυτή την κατεύθυνση, θα αλλάξουμε τη βάση μέτρησης των beat onsets της synthesized παρτιτούρας από second σε παράθυρα δειγμάτων ώστε να μπορούν μέσω του alignment path να μας δώσουν τα αντίστοιχα μεγέθη για την ηχογράφηση.

Η μετατροπή από δευτερόλεπτα σε παράθυρα γίνεται με μία απλή πράξη:

$$(beat\ onsets)_w = (beat\ onsets)_s * \left(\frac{sampling\ rate}{hop\ size} \right)$$

Οι δείκτες w και s δηλώνουν τη μονάδα στην οποία μετράμε κάθε φορά τα onsets, $sampling\ rate$ και $hop\ size$ είναι μεγέθη που αναφέρονται στην STFT ανάλυση, όπως ορίζονται στην παράγραφο 4.6.3. Μετρώνται σε $(samples / second)$ και $(samples / window)$, αντίστοιχα.

Γνωρίζουμε πως το μονοπάτι ευθυγράμμισης συνδέει κάθε χρονική στιγμή της synthesized παρτιτούρας με την αντίστοιχη της ηχογράφησης. Αν, λοιπόν, το δεικτοδοτήσουμε με τις ως άνω τιμές, θα λάβουμε τα beat onsets της ηχογράφησης.

Εφόσον έχουμε λάβει προηγουμένως και το πλήθος των beat που αντιστοιχεί σε κάθε μέτρο από το MusicXML αρχείο, (βλ. κεφάλαιο 4) καταλήγουμε στη στιγμή έναρξης κάθε μέτρου και κατ' επέκταση (εφόσον η έναρξη κάθε μέτρου συμπίπτει με τη λήξη του προηγούμενου) στη διάρκεια καθενός εξ αυτών μετρημένη σε παράθυρα δειγμάτων. Αρκεί μία

μετατροπή αντίστροφη αυτής που φαίνεται παραπάνω για να καταλήξουμε στις διάρκειες των μέτρων σε second.

Τελικά, για να φτάσουμε στο ζητούμενο, εκτελούμε την παρακάτω πράξη:

$$(bpm \text{ per measure}) = \frac{beats \text{ per measure}}{(seconds \text{ per measure}/60)}$$

7. Αντιπαραβολή Εκτέλεσης - Παρτιτούρας

Στην παράγραφο 4.7.2 απαριθμήσαμε ποιες από τις σημειώσεις που μπορεί κανείς να συναντήσει σε μία παρτιτούρα καταγράφουμε ώστε να ελέγξουμε το βαθμό ομοιότητας των δύο αρχείων εισόδου ως προς το ρυθμό. Για να ελέγξουμε αν τα bpm της ηχογράφησης είναι κάθε φορά αυτά που πρέπει, και δεδομένου ότι στη synthesized παρτιτούρα έχει επιβληθεί σταθερό tempo από την αρχή έως το τέλος, θα πρέπει να συνδέσουμε όλα τα μέτρα για τα οποία κάθε σημείωση βρίσκεται σε ισχύ με ποσότητες bpm ή με κάποια ένδειξη που θα ερμηνεύεται ως απαίτηση αύξησης ή μείωσης των bpm.

7.1. Κωδικοποίηση Σημειώσεων Παρτιτούρας

Μία ρυθμική οδηγία είναι σε ισχύ από το μέτρο στο οποίο εμφανίζεται έως ότου παρουσιαστεί κάποια καινούρια σημείωση ή φτάσει το κομμάτι στο τέλος του. Στην cell array tempo (βλ. παράγραφο 4.1.2) προστίθεται άλλη μία σειρά, μέσα σε κάθε κελί της οποίας (πάντα συνδεδεμένο με το αντίστοιχο μέτρο) αποθηκεύεται συμβολικά η συναγόμενη επιθυμητή από τον εκτελεστή ρυθμική συμπεριφορά.

Κάθε κατηγορία μουσικών οδηγιών, χρίζει και διαφορετικής προσέγγισης:

a) «Μετρονομικές» Οδηγίες

Στα cells για τα οποία αυτές βρίσκονται σε ισχύ ανατίθεται ο σημειωμένος αριθμός bpm, όπως προκύπτει από την πρώτη σειρά του cell array tempo.

b) «Ποιοτικές» Οδηγίες.

Στα cells για τα οποία αυτές βρίσκονται σε ισχύ ανατίθεται ένα cell που περιέχει τα μέγιστα κι ελάχιστα αναμενόμενα bpm, όπως προκύπτει από την πρώτη σειρά του cell array tempo.

c) Οδηγία «a tempo»

Σε αυτή την περίπτωση αναζητείται ο αριθμός bpm στον οποίον ζητείται να επιστρέψει ο εκτελεστής. Από το τρέχον μέτρο και μέχρις ότου δοθεί νέα οδηγία ή τελειώσει το απόσπασμα ανατίθεται ο αριθμός αυτός.

d) Οδηγίες σταδιακής αύξησης και μείωσης του tempo

Στα cells για τα οποία αυτές βρίσκονται σε ισχύ ανατίθεται σε περίπτωση:

- αύξησης του tempo : ο αριθμός 1
- μείωσης του tempo : ο αριθμός -1

e) Οδηγίες απότομης αύξησης και μείωσης του tempo

Στο cell όπου μία από αυτές τίθεται σε ισχύ ανατίθεται σε περίπτωση:

- αύξησης του tempo : ο αριθμός 2
- μείωσης του tempo : ο αριθμός -2

Έχουμε, πλέον, αποθηκεύσει σε ένα cell array την πραγματική ταχύτητα με την οποία αποδίδει ο εκτελεστής το απόσπασμα καθώς και την επιθυμητή ταχύτητα ή κατεύθυνση της ταχύτητας (αυξητική ή ανασχετική) για κάθε ένα μέτρο. Αυτό που απαιτείται τώρα είναι να οριστεί ένα στατιστικός έλεγχος βάσει του οποίου θα κρίνουμε αν η ηχογράφηση υπηρετεί τα όσα ζητούνται από την παρτιτούρα αναφορικά με το ρυθμό.

7.2. Just Noticeable Difference στο tempo

Είναι απολύτως λογικό η ηχογράφηση να έχει μικρές ρυθμικές διακυμάνσεις ακόμα και σε σημεία που ο εκτελεστής επιδιώκει απόλυτη σταθερότητα στο tempo. Κάτι τέτοιο δε συνιστά απαραίτητα λάθος, απεναντίας, λάθος θα ήταν αν σε μία τέτοιου τύπου ανάλυση αγνοούσαμε τον ανθρώπινο παράγοντα και τις συνακόλουθες εκτελεστικές ατέλειες. Από ποιο σημείο κι έπειτα, λοιπόν, μπορούμε να συμπεράνουμε έγκυρα ότι έχουμε ξεπεράσει τα όρια του επιτρεπτού; Πότε είναι η ταχύτητα εκτέλεσης αρκετά μεγαλύτερη ή μικρότερη από την ζητούμενη ώστε να χρίζει διόρθωσης;

Η απάντηση που δώσαμε είναι: όταν η διαφορά μεταξύ των δύο ταχυτήτων μπορεί να γίνει αντιληπτή από το μέσο ακροατή.

Εδώ πρέπει να εισάγουμε την έννοια **«οριακά αντιληπτή διαφοροποίηση» (Just Noticeable Difference, JND) στο tempo**. Πρόκειται για ένα εργαλείο στατιστικού ελέγχου που μας δίνει ακριβώς αυτό που ψάχνουμε: τον ελάχιστο βαθμό στον οποίο χρειάζεται να αλλάξει το tempo για να αντιληφθεί ο ακροατής ότι πράγματι έχει συμβεί κάποια ρυθμική αλλαγή. Μετράται ως ποσοστό επί των bpm κι έχει υπολογιστεί πειραματικά ότι βρίσκεται μεταξύ 6% και 8% [85][86].

Εμείς μετά από αρκετές δοκιμές αποφασίσαμε να κρατήσουμε το 8% ως πιο έγκυρο για την προκειμένη περίπτωση.

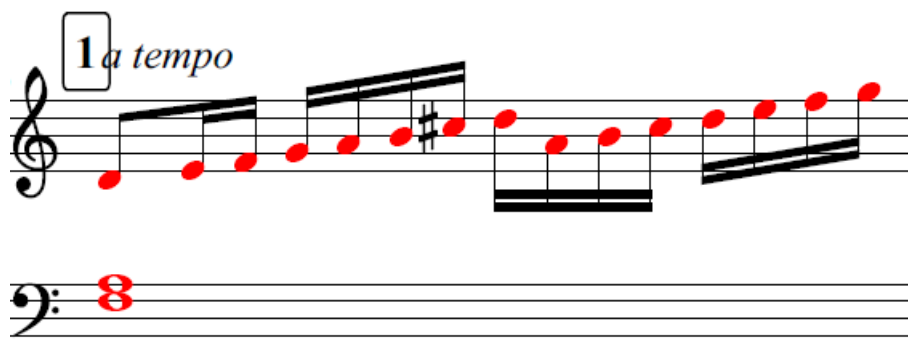
7.3. Σύγκριση μεταξύ bpm παρτιτούρας και ηχογράφησης - περιγραφή επαυξημένης παρτιτούρας

Έχουμε φτάσει πλέον στο σημείο όπου διαθέτουμε όλα τα απαιτούμενα εργαλεία προκειμένου να βγάλουμε ένα οριστικό συμπέρασμα για το κατά πόσο ο εκτελεστής είναι πιστός στις οδηγίες της παρτιτούρας που αναφέρονται στο tempo. Οι περιπτώσεις που αυτό δε συμβαίνει σημειώνονται σε μία νέα, επαυξημένη παρτιτούρα με τους εξής τρόπους:

- i. τοποθέτηση αριθμών/δεικτών στα σημεία που έχουν προκύψει σφάλματα
- ii. χρωματισμός των νοτών που περιλαμβάνονται στα αποσπάσματα για τα οποία ισχύει η εκάστοτε σημείωση
- iii. εμφάνιση μηνύματος λάθους ή προειδοποίησης στο τέλος της παρτιτούρας, συνοδεία του αντίστοιχου δείκτη (αυτού στον οποίον αναφερόμαστε παραπάνω)

Το μήνυμα στο τέλος της σελίδας (περίπτωση iii.) μπορεί να εμφανίζεται με τη μορφή ειδοποίησης σφάλματος (ERROR) ή προειδοποίησης (WARNING). Κατ' αντιστοιχία οι νότες θα χρωματίζονται (περίπτωση ii.) κόκκινες ή πορτοκαλί.

α)



β)

1) ERROR: a tempo (measure 9) was not executed

Εικόνα 46 Παράδειγμα σφάλματος: στο μέτρο αυτό δεν πραγματοποιήθηκε η εντολή «a tempo». Οι νότες του χρωματίζονται κόκκινες (α) και πάνω από το μέτρο τοποθετείται ο δείκτης 1 που μας παραπέμπει σε ένα μήνυμα σφάλματος (β) το οποίο τοποθετείται στο τέλος της επαυξημένης παρτιτούρας.

Περιγράφουμε αναλυτικότερα τι συμβαίνει για κάθε κατηγορία οδηγιών:

α) «μετρονομική» σημείωση

Αν ο μέσος αριθμός bpm για το διάστημα στο οποίο ισχύει η σημείωση είναι σε σύγκριση με τον υποδεικνυόμενο αριθμό bpm:

- ο μικρότερος ή μεγαλύτερος κατά περισσότερο από 8% (JND στο tempo, βλ. παράγραφο 7.2) και λιγότερο από 16%

εμφανίζεται μήνυμα προειδοποίησης και οι αντίστοιχες νότες χρωματίζονται πορτοκαλί.

- ο μικρότερος ή μεγαλύτερος κατά 16% (διπλάσιο του JND στο tempo)

εμφανίζεται μήνυμα σφάλματος και οι αντίστοιχες νότες χρωματίζονται κόκκινες.

Αν και μία διαφοροποίηση στα bpm της τάξης του 8% γίνεται αντιληπτή από τον ακροατή, λόγω του πλαισίου στο οποίο πραγματοποιείται η ανάλυσή μας, θεωρήθηκε πιο ρεαλιστικό να δοθεί στον εκτελεστή ένα μεγαλύτερο περιθώριο απομάκρυνσης από την παρτιτούρα προτού

χαρακτηρίσουμε την εκτέλεση εσφαλμένη. Έτσι, τα ποσοστά απόκλισης μεταξύ 8% και 16% μπορούν να αποδοθούν στην καλλιτεχνική αδεία και, κατ' επέκταση σημειώνονται μόνο με μία προειδοποίηση.

Επιπλέον, αν ο μέσος αριθμός bpm για το διάστημα στο οποίο ισχύει η σημείωση:

- ο *τείνει να είναι όμοιος με τον υποδεικνυόμενο αριθμό bpm αλλά παρόλα αυτά το πρώτο μέτρο του διαστήματος διαφέρει ρυθμικά από το τελευταίο τουλάχιστον κατά 8%*

εμφανίζεται μήνυμα σφάλματος και οι αντίστοιχες νότες χρωματίζονται κόκκινες.

Με αυτό τον τρόπο προσπαθούμε να προλάβουμε ορισμένες ιδιαίτερες περιπτώσεις που δεν προβλέπονται με τον υπολογισμό των μέσων bpm (πχ σε ένα απόσπασμα στο οποίο ο εκτελεστής επιταχύνει ξεκινώντας από τα 120 bpm και καταλήγει στα 160 bpm η μέση ταχύτητά του υπολογίζεται στα 140 bpm. Αν από την παρτιτούρα έχει δοθεί οδηγία για σταθερή ταχύτητα στα 140 bpm, θα είχαμε μία σημαντική απόκλιση η οποία δε θα εντοπιζόταν χωρίς τον τελευταίο αυτό έλεγχο).

b) «ποιοτική» οδηγία tempo

Αν ο μέσος αριθμός bpm για το διάστημα στο οποίο ισχύει η σημείωση είναι:

- ο *μικρότερος από το ελάχιστο υποδεικνυόμενο αριθμό bpm ή μεγαλύτερος από το μέγιστο υποδεικνυόμενο αριθμό bpm κατά περισσότερο από 8%*

εμφανίζεται μήνυμα προειδοποίησης και οι αντίστοιχες νότες χρωματίζονται πορτοκαλί.

Λόγω του ασαφούς χαρακτήρα των σημειώσεων αυτής της κατηγορίας, κρίναμε ότι είναι άστοχο να χαρακτηρίζει κανείς ως λάθος τέτοιου τύπου αποκλίσεις, δεδομένου ότι σε αυτές τις περιπτώσεις δίνεται εξ ορισμού στον εκτελεστή μεγάλος βαθμός ελευθερίας ως προς την επιλογή του tempo.

Επιπλέον, αν ο μέσος αριθμός bpm για το διάστημα στο οποίο ισχύει η σημείωση:

- ο *τείνει να βρίσκεται εντός των υποδεικνυόμενων οριακών αριθμών bpm αλλά παρόλα αυτά το πρώτο μέτρο του διαστήματος διαφέρει ρυθμικά από το τελευταίο τουλάχιστον κατά 8%*

εμφανίζεται μήνυμα προειδοποίησης και οι αντίστοιχες νότες χρωματίζονται πορτοκαλί.

Η λογική εδώ είναι η ίδια με αυτή που ακολουθείται στην αντίστοιχη περίπτωση για τις μετρονομικές οδηγίες. Ο λόγος που χρησιμοποιούμε μηνύματα προειδοποίησης κι όχι σφάλματος είναι αυτός που αναλύεται ακριβώς από πάνω.

c) a tempo

Αν ο μέσος αριθμός bpm για το διάστημα στο οποίο ισχύει η σημείωση είναι σε σύγκριση με τον υποδεικνυόμενο αριθμό bpm:

- ο μικρότερος ή μεγαλύτερος κατά 8%

εμφανίζεται μήνυμα σφάλματος και οι νότες του μέτρου στο οποίο αναμενόταν η αλλαγή χρωματίζονται κόκκινες.

Εδώ ο έλεγχος σφάλματος είναι πιο αυστηρός διότι τα bpm στα οποία οδηγείται ο εκτελεστής να επιστρέψει έχουν προηγουμένως καθοριστεί από τον ίδιο.

d) Οδηγίες σταδιακής αύξησης ή μείωσης του tempo

Στην προκειμένη πρέπει να υπολογιστούν τα εξής μεγέθη: το tempo με το οποίο ξεκινά το διάστημα σταδιακής αλλαγής, το tempo στο οποίο καταλήγει²⁵ και η μέση κλίση με την οποία αλλάζουν τα bpm κατά το διάστημα αυτό.

Όταν για το μέσο ρυθμό μεταβολής του tempo (έστω slope) ισχύει:

- ο $slope < 0$ όταν ζητείται αύξηση (*accelerando, precipitando, stringendo*)

ή

$slope > 0$ όταν ζητείται μείωση (*ritenuto, rallentando, ritardando, κτλ.*),

εμφανίζεται μήνυμα σφάλματος και οι αντίστοιχες νότες χρωματίζονται κόκκινες.

- ο $slope < 0$ όταν ζητείται μείωση αλλά υπάρχει τουλάχιστον ένα μέτρο κατά το οποίο το tempo παρουσιάζει αύξηση

ή

$slope > 0$ όταν ζητείται αύξηση αλλά υπάρχει τουλάχιστον ένα μέτρο κατά το οποίο tempo παρουσιάζει μείωση,

εμφανίζεται μήνυμα προειδοποίησης και οι αντίστοιχες νότες χρωματίζονται πορτοκαλί²⁶.

²⁵Το μέτρο από το οποίο θα αντλήσουμε τα bpm που αντιστοιχούν στο περί ου ο λόγος tempo κατάληξης μπορεί να είναι ή το τελευταίο μέτρο του διαστήματος ρυθμικής αλλαγής (όταν έπεται αλλαγή αντίθετης κατεύθυνσης, εντολή a tempo ή η λήξη του κομματιού) ή το αμέσως επόμενο (σε κάθε άλλη περίπτωση).

- *slope* < 0 όταν ζητείται μείωση αλλά το αρχικό tempo είναι μεγαλύτερο του τελικού τουλάχιστον κατά 8%
- ή
- *slope* > 0 όταν ζητείται αύξηση αλλά το τελικό tempo είναι μεγαλύτερο του αρχικού τουλάχιστον κατά 8%

εμφανίζεται μήνυμα σφάλματος και οι αντίστοιχες νότες χρωματίζονται κόκκινες.

e) Οδηγίες απότομης αύξησης ή μείωσης του tempo
Υπολογίζουμε το μέσο πλήθος bpm του διαστήματος που ξεκινά από το μέτρο στο οποίο εμφανίζεται η σημείωση μέχρις ότου τελειώσει το κομμάτι ή συναντήσουμε κάποια νέα ρυθμική οδηγία. Αν τόσο το μέγεθος αυτό όσο και τα bpm του ίδιου του μέτρου στο οποίο έχει τοποθετηθεί η οδηγία είναι:

- μικρότερα τουλάχιστον κατά %8 από τα bpm του μέτρου που βρίσκεται ακριβώς πριν την εμφάνιση της σημείωσης, αν αυτή είναι η *riu mosso*
- ή
- μεγαλύτερα τουλάχιστον κατά %8 από τα bpm του μέτρου που βρίσκεται ακριβώς πριν την εμφάνιση της σημείωσης, αν αυτή είναι η *meno mosso*

εμφανίζεται μήνυμα σφάλματος και οι νότες του μέτρου στο οποίο εμφανίζεται η σημείωση χρωματίζονται κόκκινες.

²⁶Αν υπάρχει μόνο ένα μέτρο το οποίο εξαιρείται από την γενικότερη αυξητική ή ανασχετική τάση, τότε χρωματίζεται μόνο αυτό, αν είναι περισσότερα τότε χρωματίζεται όλη η περιοχή και σημειώνεται ότι η αύξηση/μείωση είναι ασταθής.

8. Δημιουργία επαυξημένης παρτιτούρας

Περιγράψαμε παραπάνω ποια θα είναι κατά περίπτωση τα στοιχεία που προστίθενται στην επαυξημένη παρτιτούρα. Πρέπει τώρα να δούμε πώς θα συμβεί αυτό. Παρεμβαίνουμε στο .xml αρχείο της παρτιτούρας και, με χρήση μιας σειράς java εντολών από την βιβλιοθήκη xerces, τοποθετούμε εντός του νέους κόμβους ή αλλοιώνουμε τους ήδη υπάρχοντες.

Στη συνέχεια περιγράφουμε τη διαδικασία για κάθε μία περίπτωση.

- Οι δείκτες οι οποίοι θα αριθμούν τα σφάλματα, τοποθετημένοι πάνω από το πρώτο μέτρο του εκάστοτε διαστήματος όπου αυτά εμφανίζονται, θα μπαίνουν στην παρτιτούρα μέσω της προσθήκης κόμβων (nodes) στο .xml αρχείο. Εκμεταλλευόμαστε την επιλογή προσθήκης **σημείων πρόβας (rehearsal points)** του musescore, όπως ονομάζεται μία ειδική κατηγορία ενδείξεων που χρησιμοποιούνται από τους μουσικούς σαν «σημάδια» πάνω από το κομβικά σημεία της παρτιτούρας για τη διευκόλυνση της αναμεταξύ τους συνεννόησης σε πρόβες.



Εικόνα 47 Παράδειγμα σημείου πρόβας

Τα rehearsal points εμφανίζονται στο .xml αρχείο ως nodes με ετικέτα rehearsal. Κάθε τέτοιο στοιχείο θα έχει ως περιεχόμενο κειμένου τον αριθμό - δείκτη που θα το αντιστοιχεί στο κατάλληλο μήνυμα σφάλματος / προειδοποίησης στο τέλος της παρτιτούρας.

```
112 | □ | <direction placement="above">
113 | □ | <direction-type>
114 | □ | <rehearsal>1</rehearsal>
115 | □ | </direction-type>
116 | □ | </direction>
```

Εικόνα 48 Παράδειγμα σημείου πρόβας εντός του .xml αρχείου, όπως παρουσιάζεται στη δική μας εφαρμογή

- Τα μηνύματα σφάλματος ή προειδοποίησης στο τέλος της παρτιτούρας. Είναι σύντομες προτάσεις που επεξηγούν τον τύπο της ρυθμικής απόκλισης μεταξύ των αρχείων εισόδου καθώς και το

σημείο στο οποίο εμφανίζεται καθεμία. Του κάθε μηνύματος προηγείται ένας αριθμός που το φέρνει σε αντιστοιχία με το ως άνω εντός – της – παρτιτούρας δείκτη.

- 1) ERROR in measures 1 to 4 : tempo is too fast
- 2) ERROR in measures 5 to 10 : no acceleration was executed
- 3) WARNING in measures 11 to 12 : tempo is slower than indicated

Εικόνα 49 Παράδειγμα δύο μηνυμάτων σφάλματος κι ενός προειδοποίησης

Σε αυτή την περίπτωση κάνουμε χρήση της δυνατότητας προσθήκης σημείωσης τύπου **credit**, των συνοδευτικών δηλαδή γραμμών κειμένου που μπαίνουν στα άνω και κάτω περιθώρια της παρτιτούρας κι αναφέρονται στον τίτλο του έργου, στο συνθέτη κι άλλα meta στοιχεία. Ο λόγος είναι ότι έτσι μας δίνεται η δυνατότητα να βάζουμε τα μηνύματα στο κάτω μέρος της τελευταίας σελίδας της παρτιτούρας. Τα credit έχουν εντός του .xml τη μορφή κόμβων με ετικέτα `credit` και attribute τον αριθμό της σελίδας στην οποία αναγράφονται. Παιδί του `credit` είναι το `credit - words` που έχει σαν περιεχόμενο κειμένου το κείμενο που είναι ορατό πάνω στην παρτιτούρα και `attributes` τις ακριβείς συντεταγμένες εμφάνισής του, το είδος της χρησιμοποιούμενης γραμματοσειράς, το μέγεθος της κι άλλα.

```
69 <credit page="1">
70   <credit-words default-x="75.275" default-y="163.386" font-size="12" justify="center" >1 ) ERROR in measures 1 to 4 : tempo is too fast</credit-words>
71 </credit>
72 <credit page="1">
73   <credit-words default-x="75.275" default-y="138.386" font-size="12" justify="center" >2 ) ERROR in measures 5 to 10 : no acceleration was executed</credit-words>
74 </credit>
75 <credit page="1">
76   <credit-words default-x="75.275" default-y="113.386" font-size="12" justify="center" >3 ) WARNING in measures 11 to 12 : tempo is slower than indicated</credit-words>
77 </credit>
```

Εικόνα 50 Τα μηνύματα της εικόνας 49 όπως καταγράφονται στο .xml

□ Ο χρωματισμός με κόκκινο ή πορτοκαλί των νοτών του διαστήματος στο οποίο αναφέρονται τα παραπάνω.



Εικόνα 51 Παράδειγμα χρωματισμένων νοτών, περίπτωση προειδοποίησης (WARNING)

Αυτή τη φορά πρέπει να προσθέσουμε σε κάθε μία νότα έναν κόμβο – παιδί με ετικέτα `notehead`, attribute `color` με τιμή αυτή του χρώματος σε δεκαεξαδικό συμβολισμό (`#FF0000` για το κόκκινο και `#FFA500` για το πορτοκαλί) και περιεχόμενο κειμένου `normal` που υποδεικνύει ότι η νότα θα έχει σχήμα κανονικό.

```
4787 | <notehead color="#FFA500">normal</notehead>
```

Εικόνα 52 Ο κόμβος για την προσθήκη χρώματος πορτοκαλί σε μία νότα

9. Επίλογος: συμπεράσματα και προτάσεις για μελλοντικές επεκτάσεις

Το επιστέγασμα της παρούσας εργασίας είναι μία απλή κι εύχρηστη προγραμματιστική εφαρμογή που απευθύνεται κυρίως σε μουσικούς εκτελεστές. Μπορεί κανείς χάρη σε αυτή να διαπιστώσει με πολύ απλό τρόπο αν κάποια ηχογράφιση που έχει πραγματοποιήσει είναι σε ρυθμική συνάφεια με την παρτιτούρα την οποία επιχειρεί να αποδώσει. Το τελικό αποτέλεσμα / έξοδος της εφαρμογής έχει μορφή κατανοητή σε οποιονδήποτε έχει έστω και μικρή σχέση με την κλασική μουσική σημειογραφία. Ακόμα και σε περιπτώσεις που ο χρήστης έχει δώσει στην είσοδο κάποια ηχογράφιση κακής ηχητικής ποιότητας, με σημαντικές αυθαιρεσίες ως προς το ρυθμό (ή ενδεχομένως και σε ζητήματα όπως οι στιγμές έναρξης και λήξης της καθαυτής εκτέλεσης εν μέσω του αρχείου ήχου) το πρόγραμμα θα τη συγχρονίσει με την synthesized παρτιτούρα και θα δώσει έξοδο. Επιπλέον πλεονεκτήματα είναι η σχετική ευρύτητα επιλογών ως προς το format των αρχείων εισόδου καθώς και το γεγονός ότι ο κώδικας είναι αρκετά ανεξάρτητος από το χρησιμοποιούμενο scorewriter πρόγραμμα (musescore), πράγμα που σημαίνει ότι με λίγες παρεμβάσεις μπορεί να φτάσει να χρησιμοποιείται συνοδεία πολλών ακόμα άλλων εφαρμογών επεξεργασίας μουσικής σημειογραφίας.

Βέβαια, υπάρχει μεγάλο περιθώριο για βελτιώσεις και περαιτέρω προσθήκες στις δυνατότητες της εφαρμογής. Ένα παράδειγμα θα ήταν η επέκταση της σύγκρισης ηχογράφησης – παρτιτούρας πέραν του tempo και στις δυναμικές, στα επίπεδα έντασης, δηλαδή, της πραγματοποιηθείσας εκτέλεσης. Κάτι τέτοιο θα απαιτούσε την ανάπτυξη μίας καινούριας μεθοδολογίας αρκετά απομακρυσμένης από αυτή που ακολουθήθηκε για το tempo, πράγμα που σημαίνει ότι ενδεχομένως να ξεφεύγαμε από το σκοπό της παρούσας εργασίας. Για να γίνουμε πιο σαφείς: οι εντάσεις (και κατ' επέκταση η ακουστότητα, βλ. παράγραφο 4.1.3.2) της ηχογράφησης και της synthesized παρτιτούρας μας είναι γνωστές από τη στιγμή που λαμβάνουμε τα αρχεία εισόδου και πραγματοποιούμε το score synthesis. Θα είναι υπολογισμένες σε απόλυτα μεγέθη τα οποία μπορούν να μετατραπούν σε κλίμακα decibel και στη συνέχεια ενδεχομένως σε sones ή phones (βλ. υποσημείωση 14). Οι μετρήσεις θα έπρεπε να πραγματοποιηθούν σε επίπεδο νότας ή beat κι όχι μέτρου (όπως γίνεται για τη ρυθμική επεξεργασία, βλ. κεφάλαιο 6.), δεδομένου ότι μέσα σε ένα μέτρο μπορούν να συντελεστούν πολύ πιο ουσιώδεις αλλαγές στις εντάσεις από ότι στο tempo. Είναι απολύτως αναμενόμενο η ευθυγράμμιση των αρχείων εισόδου να περιλαμβάνει σφάλματα εξαιτίας πολλών, ανεξαρτήτων και συχνά αστάθμητων παραγόντων (ποιότητα ηχογράφησης, χρησιμοποιούμενα όργανα, κτλ.). Μιλάμε για σφάλματα σε μεγέθη milisecond, τα οποία επηρεάζουν ελάχιστα τα υπολογιζόμενα bpm και δε θέτουν σε κίνδυνο τη ρυθμική ανάλυση κι αντιπαραβολή. Στις εντάσεις, όμως, τα πράγματα είναι

διαφορετικά. Μία απόκλιση τέτοιου βαθμού, ειδικά σε όργανα που έχουν σύντομο attack, μπορεί να συνεπάγεται ένα πολύ σημαντικό σφάλμα μέτρησης, αφού η ένταση ενδέχεται μέσα σε ακόμα μικρότερο χρονικό διάστημα να «εκτιναχθεί» από τη σιωπή στο μέγιστό της. Οι διαφορετικές συμπεριφορές που αναμένονται κατά περίπτωση, ανάλογα με τα χρησιμοποιούμενα όργανα, την ταχύτητα εκτέλεσης, την ποιότητα ηχογράφησης, κτλ πολλαπλασιάζουν το αντίκτυπο ενός τέτοιου σφάλματος. Προκύπτει, τελικά, ότι προκειμένου να αντιμετωπιστεί το ζήτημα απαιτείται μία καινούρια ανάλυση από διαφορετική σκοπιά, κάτι το οποίο θα οδηγούσε την παρούσα εργασία μακριά από τον αντικειμενικό της σκοπό.

Επιπλέον, θα μπορούσαμε να επεκτείνουμε την επεξεργασία μας σε εισόδους μεγαλύτερης διάρκειας, δεδομένου ότι περιοριστήκαμε σε κομμάτια που δεν υπερβαίνουν τα 1 με 1.5 λεπτά, προκειμένου να μειώσουμε το βαθμό πολυπλοκότητας των υπολογισμών. Ένας ακόμη περιορισμός που τέθηκε αφορά το πλήθος και το είδος των χρησιμοποιούμενων οργάνων. Όλες μας οι δοκιμές, λόγω ανεπάρκειας χρόνου, έγιναν με χρήση ηχογραφήσεων από έναν πιανίστα.

Γενικεύοντας στο θέμα των δοκιμών, πρέπει να πούμε καταρχήν ότι περαιτέρω πειραματισμός με πολλά έργα και με ποικιλία οργάνων ενδεχομένως να οδηγούσε σε επιπλέον συμπεράσματα και σε βελτιστοποίηση των ήδη υπάρχουσών δυνατοτήτων της εφαρμογής. Η συγκρότηση μίας βάσης παραδειγμάτων μεγαλύτερης από την παρούσα θα βοηθούσε στην ακριβέστερη «κωδικοποίηση» των διαφόρων εκτελεστικών συμπεριφορών.

Ακόμη, μία τέτοια εμβάθυνση ανοίγει δυνητικά την προοπτική προσθήκης μίας επιπλέον χρήσιμης λειτουργίας: τον εντοπισμό ηθελημένων εκτελεστικών πρωτοβουλιών του μουσικού της ηχογράφησης και τη σημειώσή τους στην παρτιτούρα εξόδου. Πρόκειται στην ουσία για την «ανάποδη» λογική από αυτή που εφαρμόσαμε, στην οποία δεν αναζητούμε παρεκκλίσεις από τις σημειώσεις τις παρτιτούρας εν είδη σφάλματος, αλλά προσθήκες εν είδη πρότασης. Το σκεπτικό μίας τέτοιας επιπρόσθετης δυνατότητας είναι να δίδεται στο χρήστη η ευκαιρία να δει γραμμένες στο μουσικό κείμενο τις εκτελεστικές παρεμβάσεις κάποιου σολίστ οι οποίες αρχικά δεν αναγράφονταν σε αυτό, ώστε να βοηθηθεί στην προσωπική του μελέτη. Σε μια τέτοια περίπτωση όμως θα χρειαζόταν να εξετάσουμε τη μουσική και το αντίστοιχο κείμενο στην ολότητά τους (όχι μόνο , δηλαδή, στα αποσπάσματα με σημειώσεις) και να μπορούμε να αναγνωρίσουμε την όποια εμφάνιση παραλλαγής ως προς το ρυθμό ή/και την ένταση, εξ ου και η επακόλουθη απαίτηση για αυστηρότερη κωδικοποίηση των μουσικών συμπεριφορών.

Σε κάθε περίπτωση, κρίνουμε ότι η παρούσα ανάλυση και η απότοκη εφαρμογή είναι ως έχουν χρήσιμα εργαλεία σε όποιον επιθυμεί να προχωρήσει στην εμβάθυνση των ιδιαίτερων παραμέτρων της μουσικής εκτέλεσης, σε ό,τι τουλάχιστον αφορά το ρυθμό.

10. Βιβλιογραφία & Internet References

[1] Martin Litchfield West : "The Babylonian Musical Notation and the Hurrian Melodic Texts" (1994)

[2] Anne Kilmer, Anne Draffkorn, and Miguel Civil : "Old Babylonian Musical Instructions Relating to Hymnody" (1986)

[3] [Miles: The Autobiography, by Miles Davis, σελ 89](#)
<http://books.google.gr/books?id=xgAVXHhuNYgC&pg=RA1-PA89&lpg=RA1-PA89&dq=Bird+was+a+great+improviser+and+that+%27s+where+he+thought+great+music+came+from+and+what+great+musicians+were+about.+His+concept+was+#v=onepage&q&f=false>

[4] P. A. Scholes, chapter on notation, *The Oxford Companion to Music, 10th ed., Oxford University Press, 1975, pp. 688–689.*

[5] *Beethoven's Letters 1790–1826* in translation by Lady Wallace.

[6] Donington, Robert: *Baroque Music* (1982) WW Norton, 1982, σελ 33.

[7] [Σελίδα του AMPACT, \(Automated Music Performance Analysis and Comparison Toolkit για Matlab \)](#)
<http://ampact.tumblr.com/>

[8] [Η διατριβή Software-Based Extraction of Objective Parameters from MusicPerformances του A. Lerch σε .pdf](#)
http://www2.ak.tu-berlin.de/~akgroup/ak_pub/2008/Lerch%202008_Software-Based%20Extraction%20of%20Objective%20Parameters%20from%20Music%20Performances.pdf

[9] [Η εργασία Extracting Expressive Performance Information from Recorded Music του E. D. Scheirer σε pdf](#)
<http://docs.happycoders.org/unsorted/ai/music/thesis.pdf>

[10] [To Information Retrieval for Music and Motion του M. Müller από το site των εκδόσεων Springer](#)
<http://www.springer.com/computer/database+management+%26+information+retrieval/book/978-3-540-74047-6>

[11] [Η εργασία Music Compression Algorithms and Why You Should Care του Alexander Benjamin σε pdf](#)
http://ese.wustl.edu/ContentFiles/Research/UndergraduateResearch/CompletedProjects/WebPages/su10/AlexBenjamin_AudioCompression.pdf

[12] [To Audio Signal Processing and Coding από το site των εκδόσεων Wiley](http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471791474.html)

<http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471791474.html>

[13] [Σημειώσεις του καθηγητή Rod Nave του Georgia State University γύρω από τον ήχο και την ακοή](http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html)

<http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html>

[14] [Just Noticeable Difference and Tempo Change, paper του Kim Thomas σε pdf](http://www.psyencelab.com/images/Just_Noticeable_Difference_and_Tempo_Change.pdf)

http://www.psyencelab.com/images/Just_Noticeable_Difference_and_Tempo_Change.pdf

[15] [Music Perception, κεφάλαιο 6, Tempo and Rhythm του J. Devin McAuley σε pdf](http://psychology.msu.edu/TAPlab/PDF/McAuley%202010.pdf)

<http://psychology.msu.edu/TAPlab/PDF/McAuley%202010.pdf>

[16] [Alf Gabrielsson, από τη σελίδα του πανεπιστημίου της Uppsala](http://katalog.uu.se/empinfo/?languageId=1&id=XX552)

<http://katalog.uu.se/empinfo/?languageId=1&id=XX552>

[17] [Προσωπική σελίδα του Dan Ellis, Columbia University](http://www.ee.columbia.edu/~dpwe/)

<http://www.ee.columbia.edu/~dpwe/>

[18] [Βιβλιογραφία για τον τομέα της Επεξεργασίας Μουσικής Ερμηνείας από το AMPACT](http://ampact.tumblr.com/bibliography)

<http://ampact.tumblr.com/bibliography>

[19] Richard Sussman, Michael Abene, Mike Abene: Jazz Composition and Arranging in the Digital Age, σελ. 37-38

[20] [Notational alternatives beyond finale and sibelius](http://www.newmusicbox.org/articles/notational-alternatives-beyond-finale-and-sibelius/)

<http://www.newmusicbox.org/articles/notational-alternatives-beyond-finale-and-sibelius/>

[21] [επίσημη σελίδα του antescofo](http://repmus.ircam.fr/antescofo)

<http://repmus.ircam.fr/antescofo>

[22] [επίσημη σελίδα του acousmographe](http://www.inagrm.com/accueil/outils/acousmographe)

<http://www.inagrm.com/accueil/outils/acousmographe>

[23] [επίσημη σελίδα του Eanalysis](http://logiciels.pierrecouprie.fr/?page_id=402)

http://logiciels.pierrecouprie.fr/?page_id=402

[24] [επίσημη σελίδα του inscore](http://inscore.sourceforge.net/)

<http://inscore.sourceforge.net/>

- [25] [επίσημη σελίδα του Sibelius](http://www.sibelius.com/home/index_flash.html)
http://www.sibelius.com/home/index_flash.html
- [26] [επίσημη σελίδα του Finale](http://www.finalemusic.com/)
<http://www.finalemusic.com/>
- [27] [The World of Music Notation is Changing...](http://performingarts.uncg.edu/patech/?p=260)
<http://performingarts.uncg.edu/patech/?p=260>
- [28] [encore, από τη σελίδα του Passport Music Software](http://www.passportmusic.com/)
<http://www.passportmusic.com/>
- [29] [επίσημη σελίδα του GuitarPro](http://www.guitar-pro.com/en/index.php)
<http://www.guitar-pro.com/en/index.php>
- [30] [επίσημη σελίδα του Maestro Music Software](http://www.music-notation-software.com/)
<http://www.music-notation-software.com/>
- [31] [επίσημη σελίδα του MUS2](http://www.mus2.com.tr/en/)
<http://www.mus2.com.tr/en/>
- [32] [επίσημη σελίδα του musescore](http://musescore.org/)
<http://musescore.org/>
- [33] [επίσημη σελίδα του TuxGuitar](http://tuxguitar.herac.com.ar/)
<http://tuxguitar.herac.com.ar/>
- [34] [επίσημη σελίδα του LilyPond](http://www.lilypond.org/)
<http://www.lilypond.org/>
- [35] [canorus](http://sourceforge.net/apps/mediawiki/canorus/index.php?title=Main_Page)
[http://sourceforge.net/apps/mediawiki/canorus/index.php?
title=Main_Page](http://sourceforge.net/apps/mediawiki/canorus/index.php?title=Main_Page)
- [36] [επίσημη σελίδα του denemo](http://www.denemo.org/)
<http://www.denemo.org/>
- [37] [επίσημη σελίδα του frescobaldi](http://frescobaldi.org/)
<http://frescobaldi.org/>
- [38] [σελίδα του noteedit στο sourceforge](http://sourceforge.net/projects/noteedit.berlios/)
<http://sourceforge.net/projects/noteedit.berlios/>
- [39] [επίσημη σελίδα του Laborejo](http://laborejo.org/)
<http://laborejo.org/>
- [40] [MuseScore records its 5,000,000 download](#)

<http://techinmusiced.wordpress.com/2013/06/28/musescore-records-its-5000000-download-2/>

[41] [Schools & Universities using MuseScore](#)

<http://musescore.org/en/about/references>

[42] [επίσημη σελίδα του MusicXML](#)

<http://www.musicxml.com/>

[43] [XML, από το World Wide Web Consortium](#)

<http://www.w3.org/TR/2008/REC-xml-20081126/>

[44] [Essay on automated music engraving, από το LilyPond](#)

<http://www.lilypond.org/doc/v2.17/Documentation/essay/index.html>

[45] [audioread, από τη σελίδα του Columbia University για τον τομέα ece](#)

<http://www.ee.columbia.edu/~dpwe/resources/matlab/audioread/>

[46] [New audio output architecture for MuseScore 2.0](#)

[47] [An Introduction to MIDI by Andrew Swift](#)

http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol1/aps2/

[48] [Tutorial: MIDI and Music Synthesis, MIDI Sequencers and Standard MIDI Files](#)

http://www.midi.org/aboutmidi/tut_midimusicynth.php

[49] [Tutorial: MIDI and Music Synthesis, MIDI vs. Digitized Audio](#)

http://www.midi.org/aboutmidi/tut_midimusicynth.php

[50] Crawford, Walt. "MIDI and Wave: Coping with the Language". Online. Vol. 20, No. 1. Jan/Feb 1996

[51] Wiffen, Paul. "Synth School, Part 3: Digital Synthesis (FM, PD & VPM)". Sound on Sound Sep 1997

[52] [Header chunk \(of a MIDI file\)](#)

[http://www.recordingblogs.com/sa/tabid/88/Default.aspx?topic=Header+chunk+\(of+a+MIDI+file\)](http://www.recordingblogs.com/sa/tabid/88/Default.aspx?topic=Header+chunk+(of+a+MIDI+file))

[53] [Summary of MIDI Messages](#)

<http://www.midi.org/techspecs/midimessages.php>

[54] [MIDI event](#)

<http://www.recordingblogs.com/sa/tabid/88/Default.aspx?topic=MIDI+event>

- [55] [επίσημη σελίδα του fluidsynth](http://www.fluidsynth.org/)
<http://www.fluidsynth.org/>
- [56] [soundfont από τη σελίδα του musescore](http://musescore.org/en/handbook/soundfont)
<http://musescore.org/en/handbook/soundfont>
- [57] [SoundFont® Technical Specification](http://freepats.zenvoid.org/sf2/sfspec24.pdf)
<http://freepats.zenvoid.org/sf2/sfspec24.pdf>
- [58] [IBM Corporation and Microsoft Corporation \(August 1991\),
Multimedia Programming Interface and Data Specifications 1.0 \(TXT\)](http://www.kk.iij4u.or.jp/~kondo/wave/mpidata.txt)
<http://www.kk.iij4u.or.jp/~kondo/wave/mpidata.txt>
- [59] [Resource Interchange File Format](http://msdn.microsoft.com/en-us/library/windows/desktop/dd798636(v=vs.85).aspx)
[http://msdn.microsoft.com/en-us/library/windows/desktop/dd798636\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd798636(v=vs.85).aspx)
- [60] Εισαγωγή στα σήματα & συστήματα, Γ.Καραγιάννης, Κ.Τζιτζιράχου, Εκδόσεις Παπασωτηρίου, Αθήνα 2003, σελ 24, 108 - 113
- [61] [Vorbis I specification : Application and Assumptions](http://www.xiph.org/vorbis/doc/Vorbis_I_spec.html#x1-60001.1.3)
http://www.xiph.org/vorbis/doc/Vorbis_I_spec.html#x1-60001.1.3
- [62] [Frequency Range of Human Hearing, Edited by Glenn Elert --
Written by his students](http://hypertextbook.com/facts/2003/ChrisDAmbrose.shtml)
<http://hypertextbook.com/facts/2003/ChrisDAmbrose.shtml>
- [63] [Η κλασική και η σύγχρονη αντίληψη για τη λειτουργία του κοχλία,
Ανέστης Ψηφίδης, Αν. Καθηγητής Ωτορινολαρυγγολογίας ΑΠΘ](http://www.iatrikionline.gr/Orl_23/8.pdf)
http://www.iatrikionline.gr/Orl_23/8.pdf
- [64] Juan G. Roederer, The Physics and Psychophysics of Music
An Introduction, Springer, Σελ 40
- [65] MP3 AND AAC Explained, Karlheinz Brandenburg, Fraunhofer
Institute for Integrated Circuits FhG-IIS A, Erlangen, Germany
- [66] [Free Lossless Audio Codec](https://xiph.org/flac/)
<https://xiph.org/flac/>
- [67] [Βασικές αρχές συμπίεσης, από σημειώσεις του καθηγητή Νικήτα -
Μαρίνου Σγουρού](http://www.cslab.ece.ntua.gr/~sgouros/MM-Old/notes4.htm)
<http://www.cslab.ece.ntua.gr/~sgouros/MM-Old/notes4.htm>
- [68] [Digital Audio Resampling Home Page, από το Center for Computer
Research in Music and Acoustics του Stanford University](https://ccrma.stanford.edu/~jos/resample/)
<https://ccrma.stanford.edu/~jos/resample/>

[69] Εισαγωγή στα σήματα&συστήματα, Γ.Καραγιάννης, Κ.Τζιτζιράχου, Εκδόσεις Παπασωτηρίου, Αθήνα 2003, σελ 84

[70] Εισαγωγή στα σήματα&συστήματα, Γ.Καραγιάννης, Κ.Τζιτζιράχου, Εκδόσεις Παπασωτηρίου, Αθήνα 2003, σελ 95

[71] Εισαγωγή στα σήματα&συστήματα, Γ.Καραγιάννης, Κ.Τζιτζιράχου, Εκδόσεις Παπασωτηρίου, Αθήνα 2003, σελ 189

[72] [Dan Ellis, σημειώσεις από το μάθημα Digital Signal Processing, Department of Electrical Engineering - Columbia University, σελ. 19](http://www.ee.columbia.edu/~dpwe/e4810/lectures/L10-fft.pdf)
<http://www.ee.columbia.edu/~dpwe/e4810/lectures/L10-fft.pdf>

[73] [Chroma Feature Analysis and Synthesis](http://labrosa.ee.columbia.edu/matlab/chroma-ansyn/)
<http://labrosa.ee.columbia.edu/matlab/chroma-ansyn/>

[74] Arnold Whittall, The Cambridge Introduction to Serialism (New York: Cambridge University Press, 2008)

[75] [Chart which lists software that features at least some MusicXML interchange capability](http://www.musicxml.com/software/)
<http://www.musicxml.com/software/>

[76] [musicXML FAQ](http://www.musicxml.com/tutorial/faq/#Verbose)
<http://www.musicxml.com/tutorial/faq/#Verbose>

[77] [xmlread στο mathworks](http://www.mathworks.com/help/matlab/ref/xmlread.html)
<http://www.mathworks.com/help/matlab/ref/xmlread.html>

[78] [The Apache Xerces™ Project](http://xerces.apache.org/)
<http://xerces.apache.org/>

[79] [xml_io_tools στο matlab central](http://www.mathworks.com/matlabcentral/fileexchange/12907-xml-io-tools/content/html/xml_tutorial_script.html)
http://www.mathworks.com/matlabcentral/fileexchange/12907-xml-io-tools/content/html/xml_tutorial_script.html

[80] [Βασικές Ρυθμικές Σημειώσεις, από τη wikipedia](http://en.wikipedia.org/wiki/Tempo#Basic_tempo_markings)
http://en.wikipedia.org/wiki/Tempo#Basic_tempo_markings

[81] [midi toolbox](https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/miditoolbox) Eerola, T. & Toiviainen, P. (2004). *MIDI Toolbox: MATLAB Tools for Music Research*. University of Jyväskylä: Kopijyvä, Jyväskylä, Finland
<https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/miditoolbox>

[82] Ground-Truth Transcriptions of Real Music from Force – Aligned MIDI Syntheses, Robert J. Turetsky and Daniel P.W. Ellis, LabROSA, Dept. of Electrical Engineering - Columbia University

[83] Software-based extraction of objective parameters from music performances, Alexander Lerch, 2009, κεφάλαιο 3: Tempo Extraction

[84] Information Retrieval for Music and Motion, Meinard Müller, Springer Berlin Heidelberg, 2007, κεφάλαιο 4: Dynamic Time Warping

[85] J. Devin McAuley, Music Perception, Chapter 6, Tempo and Rhythm

[86] Just Noticeable Difference and Tempo Change, Kim Thomas, *University of Central Oklahoma*

[87] [GUIDE for Matlab GUIs](#)

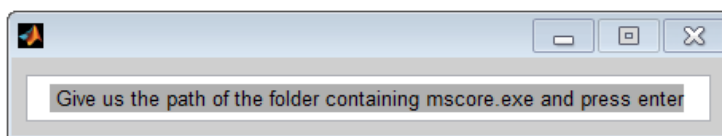
<http://www.mathworks.com/discovery/matlab-gui.html%20>

Παράρτημα 1

Περιγραφή του Graphic User Interface

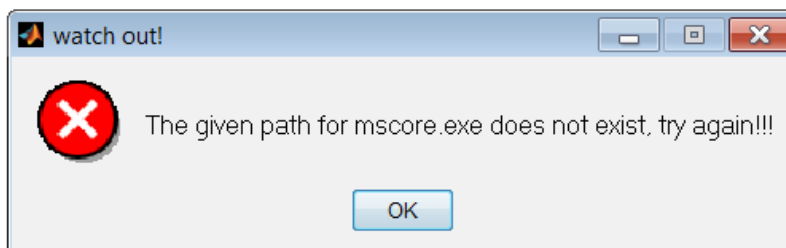
Η εργασία έρχεται μαζί και με μία προγραμματιστική εφαρμογή που δίνει τη δυνατότητα στον καθένα να εφαρμόσει τα όσα μελετήθηκαν παραπάνω. Ο χρήστης έρχεται σε επαφή με μία απλή γραφική διεπαφή (GUI) μέσω τις οποίας ελέγχει το βαθμό της ρυθμικής συνάφειας μεταξύ μιας παρτιτούρας και μίας ηχογράφησης της. Το GUI προγραμματίστηκε από την αρχή, χωρίς τη χρήση κάποιου εργαλείου – περιβάλλοντος ανάπτυξης (όπως πχ το GUIDE) [87].

Όταν εκκινεί το executable αρχείο, το πρώτο που εμφανίζεται είναι ένα παράθυρο που ζητάει το path στο οποίο έχει ο χρήστης αποθηκευμένο το mscore.exe, το executable, δηλαδή, του musescore.



Εικόνα 53 Πρώτο παράθυρο του GUI, ζητάει τη θέση αποθήκευσης του musescore

Αν η είσοδος που πληκτρολογείται είναι λαθεμένη (έχει μη αποδεκτό format, δεν υπάρχει, δεν περιέχει το mscore.exe) εμφανίζεται παράθυρο με μήνυμα σφάλματος.

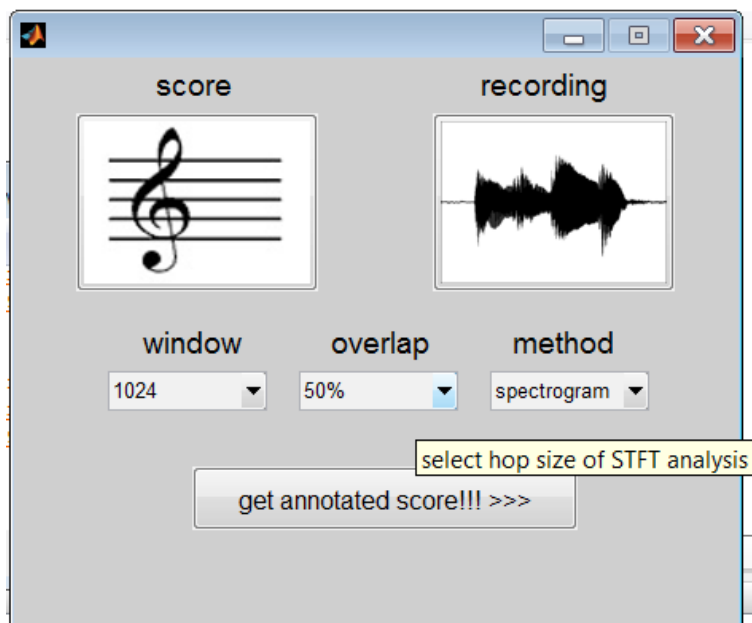


Εικόνα 54 Μήνυμα σφάλματος για λανθασμένη είσοδο – θέση αποθήκευσης του mscore.exe

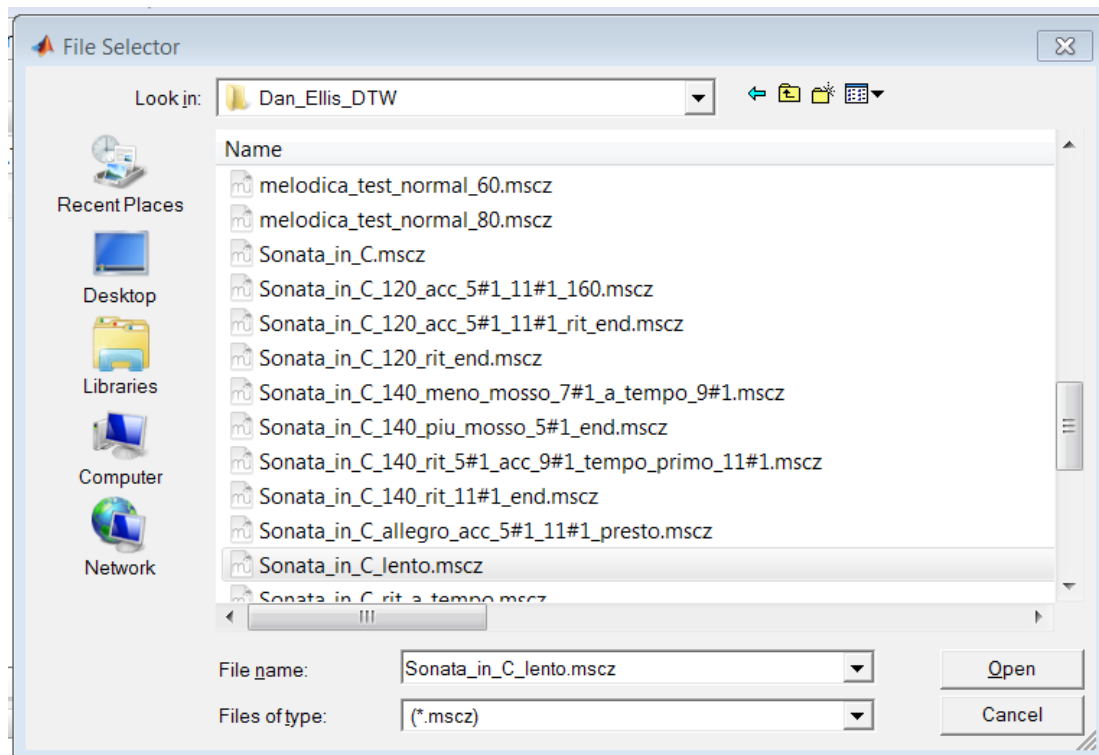
Ο χρήστης πατάει το OK που φαίνεται και στην Εικόνα 54 και η διαδικασία επαναλαμβάνεται έως ότου δοθεί αποδεκτή είσοδος. Το μονοπάτι που οδηγεί στο mscore.exe έχει πλέον αποθηκευτεί και, αν δεν αλλάξει από τον ίδιο το χρήστη, δεν ξαναζητείται.

Έπειτα, έρχεται στο προσκήνιο το panel της Εικόνα 55. Ζητούνται εδώ από το χρήστη δύο είσοδοι: η παρτιτούρα και η ηχογράφησης της. Με το πάτημα ενός εκ των δύο πλήκτρων στο πάνω μέρος του

panel (score και recording) ανοίγει ο file browser του συστήματος στον οποίο ο χρήστης πρέπει να εντοπίσει τα αρχεία που θέλει να δώσει για είσοδο (Εικόνα 56).



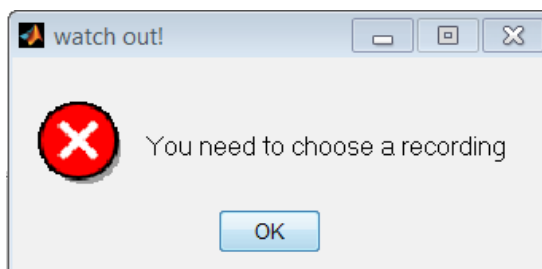
Εικόνα 55 Το κεντρικό panel του GUI



Εικόνα 56 Περιηγητής φακέλων για την επιλογή εισόδου παρτιτούρας

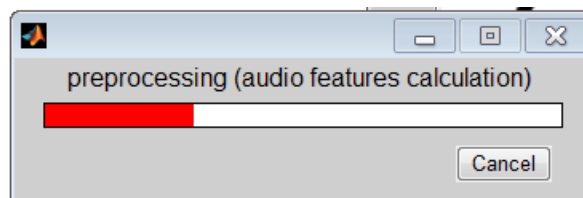
Περιλαμβάνονται, ακόμη, τρία drop down menus που επιτρέπουν στο χρήστη να επιλέξει από μία λίστα το επιθυμητό μήκος παραθύρου και overlap για το STFT καθώς και τη μέθοδο εφαρμογής του.

Στο κάτω μέρος του κεντρικού panel βλέπουμε το κουμπί με την ένδειξη create annotated score, με το πάτημα του οποίου ξεκινά η διαδικασία που θα μας επιστρέψει την τελική έξοδο, την επαυξημένη παρτιτούρα. Έχει ληφθεί, τέλος, πρόνοια για την περίπτωση που ο χρήστης επιχειρήσει να εκκινήσει τη διαδικασία αντιπαραβολής δίχως προηγουμένως να δώσει και τις δύο απαραίτητες εισόδους, οπότε κι εμφανίζεται μήνυμα λάθους.



Εικόνα 57 Μήνυμα λάθους για την περίπτωση που ο χρήστης πάει να ξεκινήσει τις διαδικασίες υπολογισμών δίχως προηγουμένως να έχει δώσει εισοδο ηχογράφησης

Τα βήματα της διεργασίας περιγράφονται κατά την εξέλιξή της σε μηνύματα πάνω σε ένα waitbar (Εικόνα 58)



Εικόνα 58 Στυγμιότυπο του waitbar

Παράρτημα 2

Ένα παράδειγμα χρήσης

Έστω ότι δίνουμε στο GUI μας ως είσοδο – παρτιτούρα αυτή που φαίνεται στην Εικόνα 59.

Πρόκειται για ένα απόσπασμα (για την ακρίβεια τα πρώτα 12 μέτρα) της σονάτας σε ντο μείζονα για πιάνο του W.A. Mozart. Έχει παραποιηθεί ώστε να ζητείται από τον εκτελεστή να ξεκινήσει παίζοντας στα 120 bpm, να αρχίσει να επιταχύνει στο πέμπτο μέτρο και να σταθεροποιείται οριστικά στο μέτρο 11, στα 160 bpm. Δίνουμε, επιπλέον, σαν δεύτερη είσοδο μία ηχογράφιση του ως άνω αποσπάσματος η οποία, όμως, εκτελείται σταθερά στα 140 bpm, παραβλέποντας όλες τις σημειώσεις της παρτιτούρας. Επιλέγουμε: μέθοδο εφαρμογής STFT το spectrogram, μήκος παραθύρου 1024 samples και hop size 512.

Απεικονίζονται παρακάτω μία σειρά εικόνες που φωτίζουν ορισμένα στιγμιότυπα από τις διαδικασίες που ακολουθούνται έως και το alignment.

Sonata_in_C_120_acc_5#1_11#1_160

♩=120

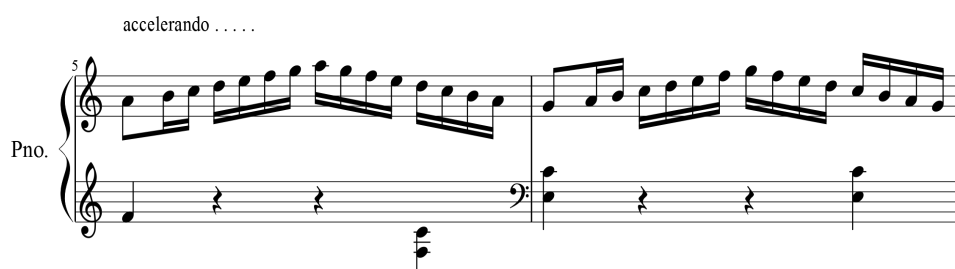
Piano



accelerando

5

Pno.



7

Pno.



9

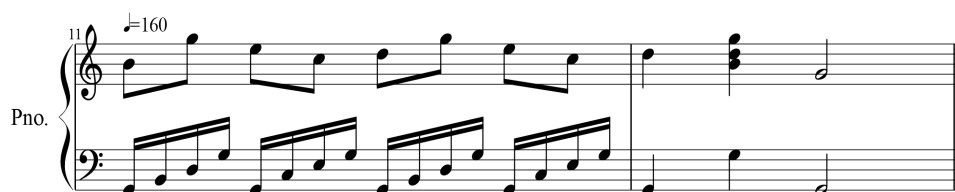
Pno.



11

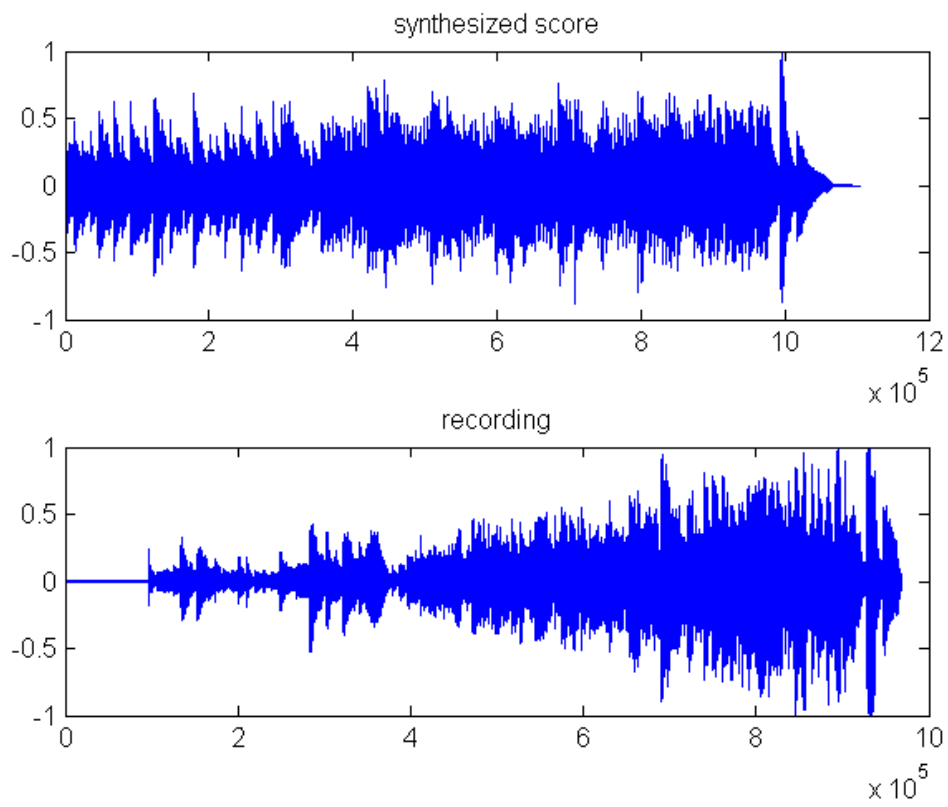
♩=160

Pno.

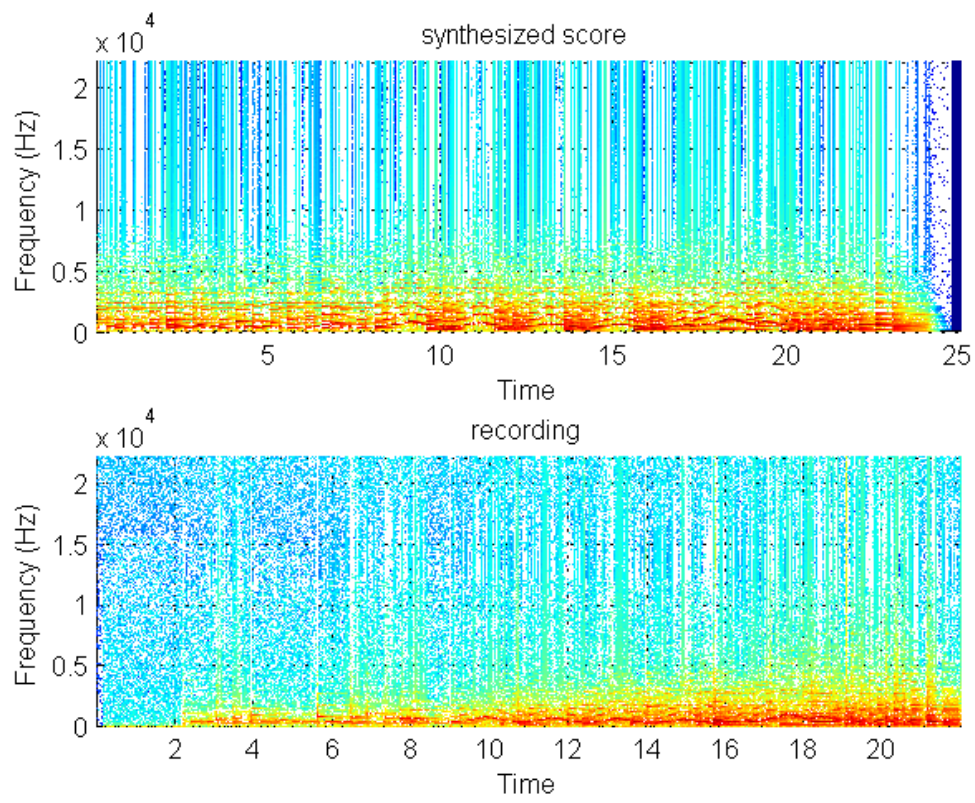


Εικόνα 59 Παρτιτούρα εισόδου

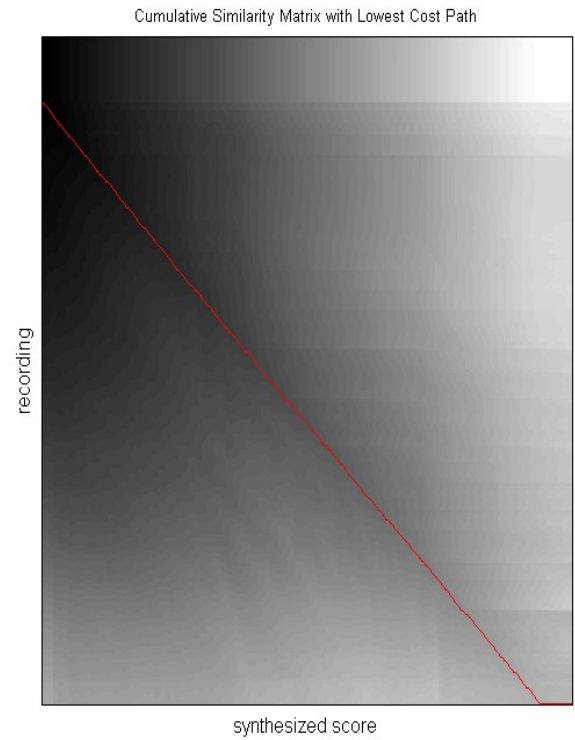
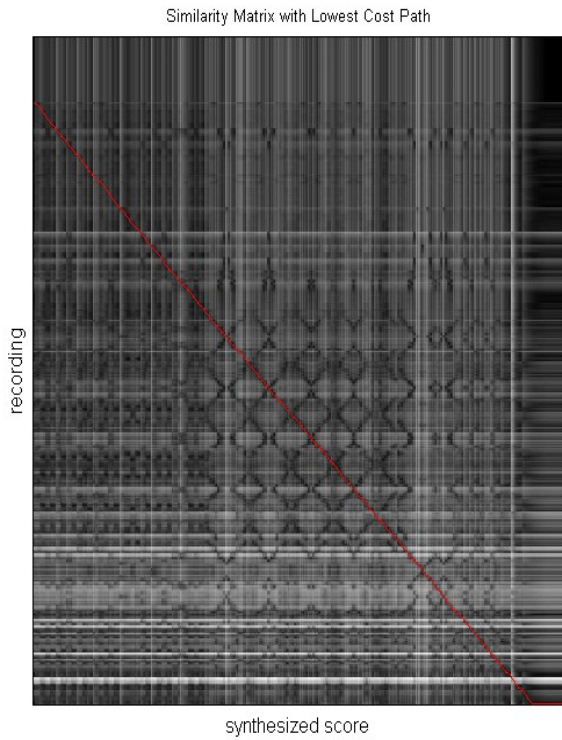
(a)



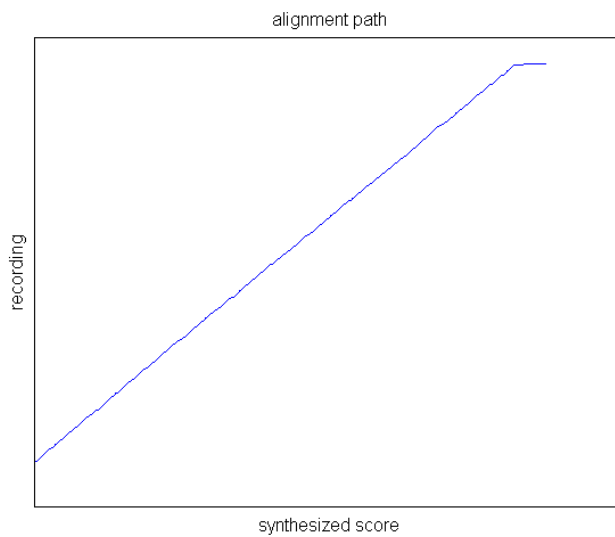
(β)



(γ)



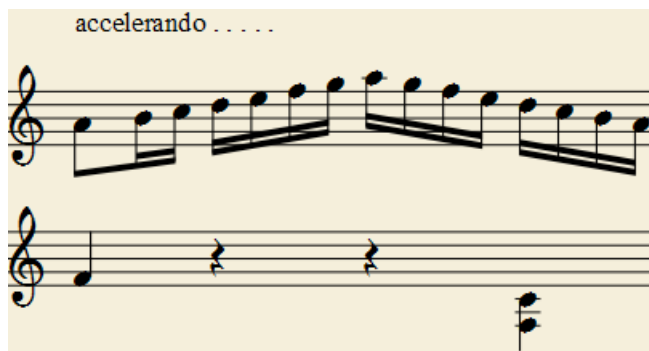
(δ)



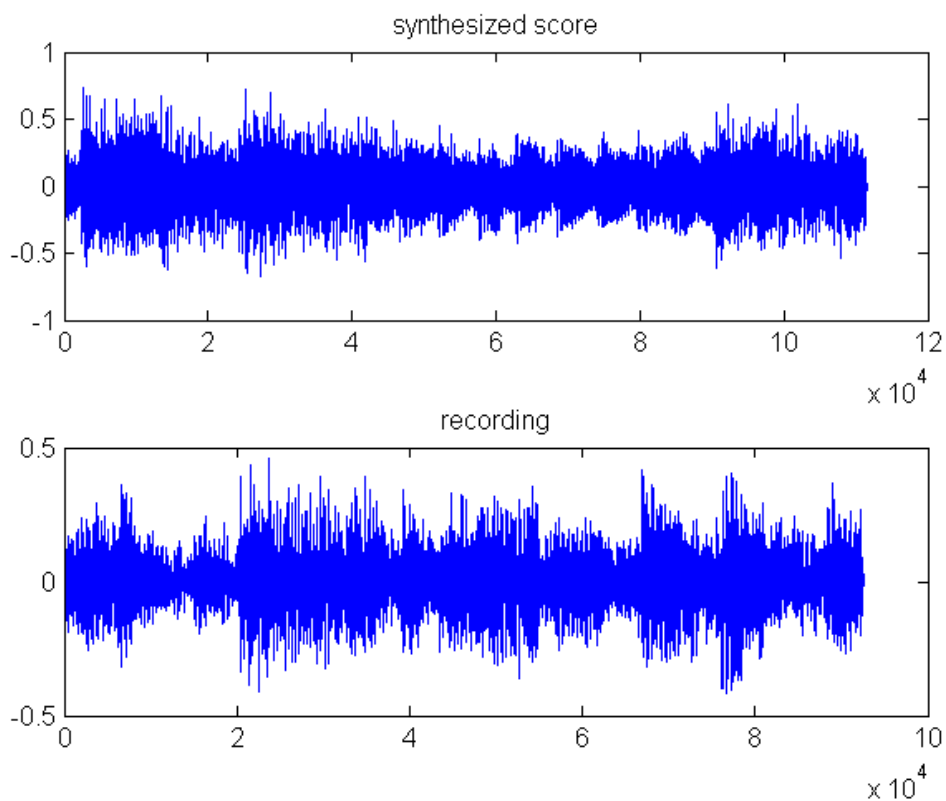
Εικόνα 60 (α) η κυματομορφή της synthesized παρτιτούρας σε αντιπαραβολή με αυτή της ηχογράφησης (β) τα spectrogram των ίδιων δύο αρχείων (γ) ο πίνακας ομοιότητας κι ο αθροιστικός πίνακας ομοιότητας πλάι - πλάι, με γραμμένο πάνω τους σε χρώμα κόκκινο το lowest - cost path (δ) το μονοπάτι ευθυγράμμισης

Για πιο λεπτομερή κατανόηση δίνονται μερικές εικόνες που αναφέρονται σε ένα μέτρο μόνο του υπό ανάλυση αποσπάσματος, το μέτρο 5.

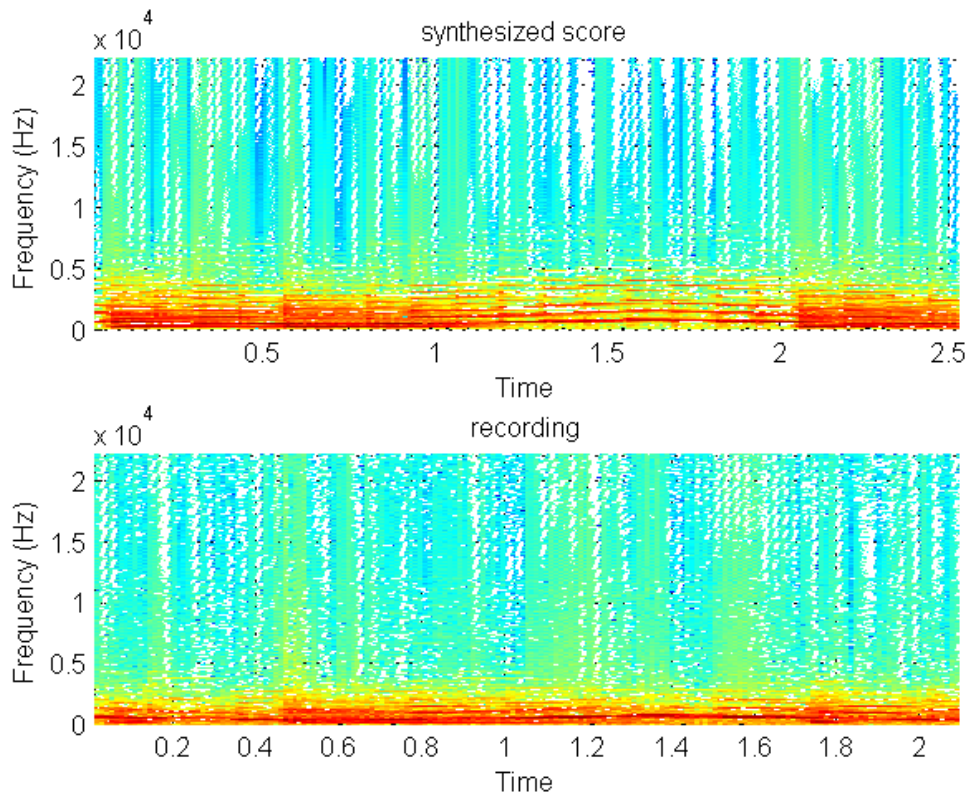
(α)



(β)



(γ)



(δ)

```
<measure number="5" width="518.07">
  <print new-system="yes">
  <direction placement="above">
    <direction-type>
      <words>accelerando . . . . . </words>
    </direction-type>
  </direction>
  <staff>1</staff>
  <note default-x="53.15" default-y="-25.00">
    <pitch>
      <step>A</step>
      <octave>4</octave>
    </pitch>
    <duration>2</duration>
    <voice>1</voice>
    <type>eighth</type>
    <stem>down</stem>
  </note>
  <beam number="1">begin</beam>
</measure>
```

Εικόνα 61(α) Το μέτρο 5 της παρτιτούρας εισόδου, αποκομμένο από το υπόλοιπο μουσικό κείμενο (β) οι κυματομορφές για το απόσπασμα που αντιστοιχεί στο ίδιο μέτρο παρμένες από την ηχογράφιση και τη synthesized παρτιτούρα (γ) όπως προηγουμένως αλλά με spectrogram αντί για κυματομορφές (δ) το κομμάτι του κώδικα του αρχείου MusicXML που αντιστοιχεί στο ίδιο μέτρο

Τέλος, παραθέτουμε σε μορφή εικόνας την επαυξημένη παρτιτούρα εξόδου.

Sonata_in_C_120_acc_5#1_11#1_160

Piano

$\text{♩} = 120$

Pno. *accelerando*

Pno.

Pno.

Pno. $\text{♩} = 160$

ERROR in measures 1 to 4 : tempo is too fast
ERROR in measures 5 to 10 : no acceleration was executed
WARNING in measures 11 to 12 : tempo is slower than it should be

Υπενθυμίζουμε ότι η ηχογράφηση είναι σταθερή στα 140 bpm. Έτσι, στα πρώτα 4 μέτρα όπου ζητείται η ταχύτητα να είναι στα 120 bpm σημειώνεται ότι η ηχογράφηση είναι πολύ γρήγορη (ERROR, εφόσον $140 > 1.08 * 120$), στα μέτρα 5 έως 10 όπου ζητείται *accelerando* σημειώνεται λάθος, εφόσον το tempo της ηχογράφησης παραμένει σταθερό και, τελικά, στα δύο τελευταία μέτρα όπου θέλουμε να σταθεροποιηθεί η ταχύτητα στα 160 bpm δίδεται μήνυμα προειδοποίησης (WARNING, εφόσον $0.85 * 160 < 140 < 0.92 * 160$).

Παρατίθενται παρακάτω δύο πίνακες με τα περιεχόμενα του cell array tempo το οποίο περιλαμβάνει τα εξής (βλ. παράγραφο 7.1) : στην πρώτη σειρά τις σημειώσεις της παρτιτούρας (όπου δεν υπάρχουν το κελί μένει κενό), στη δεύτερη το tempo ή την κατεύθυνση (αυξητική ή ανασχετική) του tempo που ζητείται για κάθε μέτρο μέσω των σημειώσεων της παρτιτούρας, στην τρίτη τα πραγματικά, μετρημένα bpm κάθε μέτρου της ηχογράφησης και στην τέταρτη ο αριθμός του κάθε μέτρου.

Μπορούμε να δούμε ότι το tempo της ηχογράφησης κρατιέται σταθερό με μικρές αποκλίσεις γύρω από τα 140 bpm παράτα όσα ζητά η παρτιτούρα.

Πίνακας 3 Ο cell array tempo για το παράδειγμά μας. Το μέρος που περιγράφει τα πρώτα 6 μέτρα...

120	[]	[]	[]	'accelerando.'	[]
				..'	
120	120	120	120	1	1
140.62	139.67	143.55	143.55	145.5766	140.625
50	48	47	47		0
1	2	3	4	5	6

Πίνακας 4 ... κι αυτό που περιγράφει τα 6 δεύτερα

[]	[]	[]	[]	160	[]
1	1	1	1	160	160
146.609	143.554	145.576	141.588	144.558	138.7374
0	7	6	2	6	
7	8	9	10	11	12

Για σύγκριση, δώσαμε ως είσοδο την ίδια παρτιτούρα και μία ηχογράφιση η οποία τηρεί τα ζητούμενα από τις οδηγίες. Δίδονται στους παρακάτω πίνακες τα περιεχόμενα του cell array tempo για αυτή την περίπτωση.

Πίνακας 5 Ο cell array tempo για την περίπτωση που έχουμε στην είσοδο το ίδιο αρχείο παρτιτούρας αλλά αυτή τη φορά ηχογράφιση η οποία ακολουθεί τις οδηγίες που δίδονται από την παρτιτούρα. Το μέρος που περιγράφει τα πρώτα 6 μέτρα...

120	[]	[]	[]	'accelerando.	[]
				..'	
120	120	120	120	1	1
122.3188	118.8039	123.0469	120.8882	130.0118	130.8347
1	2	3	4	5	6

Πίνακας 6 ... κι αυτό που περιγράφει τα 6 δεύτερα

[]	[]	[]	[]	160	[]
1	1	1	1	160	160
145.5766	146.6090	155.4276	157.8006	168.0640	159.0144
7	8	9	10	11	12

Παράρτημα 3

Κώδικας

Παραθέτουμε μόνο τα κομμάτια πρωτότυπου κώδικα, που γράφτηκαν για την εργασία (βασικές function: `lfteris`, `GUI_zero`, `GUI_first`, `preprocessing`, `align`, `performance_description`, `enhance`, `judge`).

```
function lfteris

if ~exist('musescore_path.txt')
    fileID = fopen('musescore_path.txt','w');

    pth= GUI_zero();

    fprintf(fileID,'%s\n',pth);
    fclose(fileID);
    fileID = fopen('musescore_path.txt','r');
else
    fileID = fopen('musescore_path.txt','r');
end

A=fileread('musescore_path.txt');
B=cellstr(A);
text=num2str(cell2mat(B));

if (exist('pth'))&&(strcmp(pth,'close the window'))
    fclose(fileID);
    delete('musescore_path.txt');
elseif ~exist(text)
    fclose(fileID);
    uiwait(errordlg('The given path for mscore.exe does not exist, try again!!!','watch out!','modal'));
    delete('musescore_path.txt');
    main;
elseif isempty(text)
    fclose(fileID);
    uiwait(errordlg('The given path for mscore.exe is empty, try again!!!','watch out!','modal'));
    delete('musescore_path.txt');
    main;
elseif ~(exist(text)==7)
    fclose(fileID);
    uiwait(errordlg('The given path for mscore.exe is not a folder, try again!!!','watch out!','modal'));
    delete('musescore_path.txt');
    main;
elseif strcmpi(fliplr(text),'\',1)
    if ~exist([text,'mscore.exe'])
        fclose(fileID);
        uiwait(errordlg('The given path for mscore.exe is wrong, try again!!!','watch out!','modal'));
        delete('musescore_path.txt');
```

```

    main;
else
    fclose(fileID);
    musescore_path=['',text,'mscore.exe'];
    GUI_first(musescore_path);
end
else
    if ~exist([text,'\mscore.exe'])
        fclose(fileID);
        uiwait(errordlg('The given path for mscore.exe is wrong, try
again!!!','watch out!','modal'));
        delete('musescore_path.txt');
        main;
    else
        fclose(fileID);
        musescore_path=['',text,'\mscore.exe'];
        GUI_first(musescore_path);
    end
end
end

```

```

function output= GUI_zero()

```

```

output='close the window';
Screen.fh = figure('units','pixels',...
    'position',[450 450 450 50],...
    'menubar','none',...
    'numbertitle','off',...
    'resize','off');

Screen.ed = uicontrol('style','edit',...
    'units','pix',...
    'position',[10 10 430 30],...
    'backgroundcolor','w',...
    'string',' Give us the path of the folder containing
mscore.exe and press enter',...
    'min',0,'max',1);%If Max - Min <= 1, then editable text
    %boxes accept only single line input.

```

```

uicontrol(Screen.ed) % Give the editbox control.
set(Screen.ed,'KeyPressFcn',{@keypress_call,Screen});

```

```

function [] = keypress_call(varargin)
delete('musescore_path.txt'); S = varargin{3};
key = get(gcf,'CurrentKey');
if(strcmp(key, 'return'))
    ed_call(S);
end
end

```

```

function [] = ed_call(varargin)
% Callback for edit.
S = varargin{1};
drawnow;
pth = get(S.ed,'string');
output=pth;

```

```

    % Closes the dialog when user presses return.
    close(S.fh)
end
    waitfor(Screen.fh);

end

function [] = GUI_first(mscore_path)
%%
% scnsz = get(0,'ScreenSize');
fhposition=[455 256 455 356];
x=imread('treble-clef.jpg');
I2=imresize(x, [102 142]);
y=imread('piano_notes.jpg');
I1=imresize(y, [102 142]);
%figure uicontrol
S.fh = figure('units','pixels',...
    'position',fhposition,...
    'menubar','none',...
    'numbertitle','off',...
    'resize','off');
%pushbutton uicontrol for recording choice
S.pb(1) = uicontrol('style','push',...
    'units','pixels',...
    'position',[264 210 152 112],...
    'fontSize',14,...
    'tooltip','select music file',...
    'cdata',I1);
S.tx(4) = uicontrol('style','text',...
    'unit','pix',...
    'position',[288 322 90 30],...
    'string','recording',...
    'fontSize',12,...
    'backgroundcolor',get(S.fh,'color'));
%pushbutton uicontrol for score choice
S.pb(2) = uicontrol('style','push',...
    'units','pixels',...
    'position',[40 210 152 112],...
    'fontSize',14,...
    'tooltip','select score file',...
    'cdata',I2);
S.tx(5) = uicontrol('style','text',...
    'unit','pix',...
    'position',[68 322 90 30],...
    'string','score',...
    'fontSize',12,...
    'backgroundcolor',get(S.fh,'color'));
%%pushbutton uicontrol initiating everything
S.pb(3) = uicontrol('style','push',...
    'units','pixels',...
    'position',[113 60 242 40],...
    'fontSize',10,...
    'str','get annotated score!!! >>>');
%pop up uicontrol with window size choices and explanatory text above it

```

```

S.pp(1) = uicontrol('style','pop',...
    'unit','pix',...
    'position',[60 125 100 35],...
    'tooltip','select window size of STFT analysis',...
    'string',{'512','1024','2048','4096'},...
    'value',2);
S.tx(1) = uicontrol('style','text',...
    'unit','pix',...
    'position',[77 160 70 30],...
    'string','window',...
    'fontsize',12,...
    'backgroundcolor',get(S.fh,'color'));
%pop up uicontrol with hop size choices and explanatory text above it
S.pp(2) = uicontrol('style','pop',...
    'unit','pix',...
    'position',[180 125 100 35],...
    'tooltip','select hop size of STFT analysis',...
    'string',{'25%','50%','75%','87.5%','93.75%'},...
    'value',2);
S.tx(2) = uicontrol('style','text',...
    'unit','pix',...
    'position',[195 160 70 30],...
    'string','overlap',...
    'fontsize',12,...
    'backgroundcolor',get(S.fh,'color'));
%pop up uicontrol with analyzing method choices and explanatory text above
it
S.pp(3) = uicontrol('style','pop',...
    'unit','pix',...
    'position',[300 125 100 35],...
    'tooltip','select specific method of STFT analysis',...
    'string',
    {'spectrogram','chromagram','diff(chromagram)'});
S.tx(3) = uicontrol('style','text',...
    'unit','pix',...
    'position',[310 160 70 30],...
    'string','method',...
    'fontsize',12,...
    'backgroundcolor',get(S.fh,'color'));
set([S.pb(1),S.pb(2),S.pb(3)],{'callback'},{@pb1_call,S};
{@pb2_call,S};...
{@pb3_call,S,mscore_path}); % Set callback
%%
function [] = pb1_call(varargin)
% callback for pushbutton 1
% S = varargin{3}; % Get the structure.
file1 = uigetfile({'*.wav'; '*.mp3'; '*.flac'; '*.ogg'; '*.m4a'; '*..*'},...
    'File Selector');
%store file1 (recording) in UserData for later use
set(gcbo,'UserData',file1);
%%
function [] = pb2_call(varargin)
% callback for pushbutton 2
% S = varargin{3}; % Get the structure.
file2 = uigetfile({'*.mscz'; '*.xml'; '*.mxl'; '*.mid'; '*..*'},...
    'File Selector');
%store file2 (score) in UserData for later use
set(gcbo,'UserData',file2);
%%

```

```

function [] = pb3_call(varargin)
% callback for pushbutton 3
S = varargin{3};
mscore_path = varargin{4};
file1=get(S.pb(1),'UserData');
file2=get(S.pb(2),'UserData');

%%
if isempty(file1)&&isempty(file2)
    %warning window appears in case neither a recording nor a score file
    have
    %been chosen
    errordlg('You need to choose a recording and a score','watch out!');
elseif isempty(file1)
    %warning window appears if no recording has been chosen
    errordlg('You need to choose a recording','watch out!');
elseif isempty(file2)
    %warning window appears if no score has been chosen
    errordlg('You need to choose a score','watch out!');
else

VL = get(S.pp(1),{'str','value'}); % User's choice from window size
popup.
% Now get the string updates and perform operations.
switch VL{1}{VL{2}} % User's string choice.
    case '512'
        w=512;
    case '1024'
        w=1024;
    case '2048'
        w=2048;
    case '4096'
        w=4096;
    otherwise
end

VL = get(S.pp(2),{'str','value'}); % User's choice from hop size popup.
% Now get the string updates and perform operations.
switch VL{1}{VL{2}} % User's string choice.
    case '25%'
        h=w*.25;
    case '50%'
        h=w*.5;
    case '75%'
        h=w*.75;
    case '87.5%'
        h=w*87.5;
    case '93.75%'
        h=w*93.75;
    otherwise
end

VL = get(S.pp(3),{'str','value'}); % User's choice from analyzing method
popup.
% Now get the string updates and perform operations.
switch VL{1}{VL{2}} % User's string choice.
    case 'spectrogram'
        method=1;
    case 'chromagram'

```

```

        method=2;
    case 'diff(chromagram)'
        method=3;
    otherwise
end

%%
%Preprocessing of both input files
watchon;
[AudioFeatures,ScoreFeatures,tempo,wait]=preprocessing(file1,file2,w,h,method,mscore_path,S);

%%
%align input files, return alignment path
[alignment,wait]=align(AudioFeatures,wait,S);

%%
%performance description
[performance,wait]=performance_description(ScoreFeatures,AudioFeatures,alignment,wait,S);

%%
%This final part includes the score - performance comparison and the
%production of the new, accumulated score

%enhance produces an enhanced version of the tempo cell array, containing
%the "expected" tempo or tempo change of each measure (in bpm) as given by
%the score directions (which are also added in the tempo cell array)
[tempo,wait]=enhance(tempo,performance,wait,S);

%the following function "judges" whether the score directions are followed
and
%proceeds to give a new corrected score in the output
wait=judge(tempo,ScoreFeatures,wait,S);

%new_score.xml is the new, accumulated score
if exist('new_score.xml')==2
    new_score='new_score.xml';
else
    new_score=ScoreFeatures.file2xml;
end

%we need to extract the new score in a pdf file
[file2PathName,file2Name,ext]=fileparts(new_score);
file2pdf=[file2Name,'.pdf'];
dos([mscore_path ' ' ' ' new_score ' ' ' -o ' ' ' file2PathName ...
    file2pdf ' ' ],'-echo');

% open(file2pdf);
% open('C:\Users\Gobalski\Documents\MATLAB\Dan_Ellis_DTW\new_score.pdf');
open([pwd, '\new_score.pdf']);

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

```

```

if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

waitbar(1,wait,'DONE');
pause(1);
delete(wait);
delete('new_score.xml');
delete('new_score.pdf');
watchoff;
end

function
[AudioFeatures,ScoreFeatures,tempo,wait]=preprocessing(file1,file2,w,h,method,mscore_path,S)

assignin('base','score',file2);
assignin('base','recording',file1);

%this function is responsible for the preprocessing of the two input files

%Check for input arguments. file1 and file2 can not be omitted. Default
%values for window size, hop size and method are 4096,2048 and 1
%(spectrogram) respectively.
if (nargin<3)
    w=4096;
end
if (nargin<4)
    h=w/2;
end
if (nargin<5)
    method=1;
end
%%
%extract xml and DOM files
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

wait=waitbar(0,'extract .xml and DOM files','CreateCancelBtn'...
, 'setappdata(gcf,'canceling',1)');
setappdata(wait,'canceling',0);
% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

```

```

%Extract the score (file2) in .xml format
if (strcmp(filext(file2),'.xml'))
    file2xml=file2;
else
    [file2PathName,file2Name,ext]=fileparts(file2);
    file2xml=[file2Name, '.xml'];
    dos(['mscore_path ' ' ' file2 ' ' ' -o ' ' ' file2PathName ...
        file2xml ' ''],'-echo');
end

%do not load external dtd
dbf = javaMethod('newInstance',
'javax.xml.parsers.DocumentBuilderFactory');
dbf.setFeature('http://apache.org/xml/features/nonvalidating/load-
external-dtd', false);
%read xml file into a DOM
scorexml=xmlread(file2xml,dbf);

%delete_sound_nodes deletes all nodes in the DOM file tagged as 'sound'
%this way we assure that the synthesized score file has a steady tempo
scorexml=delete_sound_nodes(scorexml);

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

%Convert the .xml file into a MATLAB structure
pref={};
file2struct= xml_read(scorexml,pref);

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    delete(wait);
    drawnow;
    % terminateExecution;
    % errordlg('execution terminated','watch out!');
    error('execution terminated');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    delete(wait);
    drawnow;
    terminateExecution;
    errordlg('execution terminated','watch out!');
end

```



```

%%
%check formats / synthesize score
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
waitbar(0.1,wait,'preprocessing(check formats / synthesize score)');

%Check format of file1 (recording, could be either .wav or .mp3)
if (strcmp(filext(file1),'.wav')||strcmp(filext(file1),'.mp3')||
(strcmp(filext(file1),'.flac')||strcmp(filext(file1),'.ogg'))
    [d1,sr1] = audioread(file1);
else
    errordlg('recording format is not recognised','watch out!!!')
end

%export xml file as .wav (SCORE SYNTHESIS)
file2PathName=fileparts(file2xml);
dos([mscore_path ' ' file2xml ' ' -o ' ' file2PathName
'tmpwav.wav' ' '],'-echo');
[d2,sr2] = audioread([file2PathName 'tmpwav.wav']);
dos(['del ' file2PathName 'tmpwav.wav']);

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

%%
%resample
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
waitbar(0.2,wait,'preprocessing (resample)');

%Resample d2 and d1 to 44100 Hz.
if (sr1~=44100)
    d1=resample(d1,44100,sr1);
end
if (sr2~=44100)
    d2=resample(d2,44100,sr2);
end

%%
%Downmix signals averaging over the channels
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

waitbar(0.23,wait,'processing (downmix)');
if (size(d1,2)>1)
    d1(:,1)=sum(d1,2)/size(d1,2);
    d1(:,2:end)=[];
end

```

```

if (size(d2,2)>1)
    d2(:,1)=sum(d2,2)/size(d2,2);
    d2(:,2:end)=[];
end

%Add zero frames to the beginnig and the end of both files.
dzero=zeros(w,1);
d2=[dzero; d2; dzero];
d1=[dzero; d1; dzero];

%%
%normalize
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
waitbar(0.26,wait,'preprocessing (normalize)');

%Vector normalization
d1=d1/abs(max(d1));
d2=d2/abs(max(d2));

%%
%audio features calculation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
waitbar(0.29,wait,'preprocessing (audio features calculation)');

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

% Calculate STFT features for both sounds
if (method==1)
D1 = spectrogram(d1,w,w-h,w,sr1);
D2 = spectrogram(d2,w,w-h,w,sr1);
elseif (method==2)
%Calculate chroma represantation
D1 = chromagram_IF(d1,sr1,w,h);
D2 = chromagram_IF(d2,sr1,w,h);
else
%Calculate the derivative of the chroma represantation
D1 = diff(chromagram_IF(d1,sr1,w,h),1,2);
D1 =[zeros(12,1) D1];
D2 = diff(chromagram_IF(d2,sr1,w,h),1,2);
D2 =[zeros(12,1) D2];
end

% Check for Cancel button press

```

```

if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

%%
%collect in one struct all the information about both audio files
%(recording and synthesisd score)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
AudioFeatures=struct;
AudioFeatures.D1=D1;
AudioFeatures.D2=D2;
AudioFeatures.d1=d1;
AudioFeatures.d2=d2;
AudioFeatures.w=w;
AudioFeatures.h=h;
AudioFeatures.sr=44100;
AudioFeatures.method=method;

assignin('base','AudioFeatures',AudioFeatures);

%%
%calculate score features
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

waitbar(0.32,wait,'calculate score features');

%Extract measure length in quarter notes (beats)
measure_length=file2struct.part.measure(1,1).attributes.time.beats;
if file2struct.part.measure(1,1).attributes.time.beat_DASH_type==8
    measure_length=measure_length/2;
elseif file2struct.part.measure(1,1).attributes.time.beat_DASH_type==2
    measure_length=measure_length*2;
end

%Extract total length in quarter notes (beats)
total_length=length(file2struct.part.measure);
total_length=total_length*measure_length;

%Export file2 (score) as a .mid file.
if (strcmp(filext(file2),'.mid'))
    [midiscore.score,midiscore.beatDur] = mid2score(file2);
elseif (strcmp(filext(file2),'.xml')||...
        (strcmp(filext(file2),'.mxl')||...
        (strcmp(filext(file2),'.mid')||...
        (strcmp(filext(file2),'.mscz')||...
        (strcmp(filext(file2),'.mp3')||...
        (strcmp(filext(file2),'.alignerStruct.wav')))
    file2PathName=fileparts(file2);

```

```

    dos(['mscore_path ' ' "' file2 "' ' -o ' "' file2PathName ...
        'tmpmid.mid' "''], '-echo');
% midiscore.score is a ("#of midi notes")x3 matrix.
% each column contains respectively the note onset, note offset, and midi
% note number of each midi note so practically midiscore.score is what its
% name suggests : the midi score
%midiscore.beatDur is beat duration in sec,1 beat by default=1 quarter
note

    [midiscore.score,midiscore.beatDur] =...
        mid2score(['file2PathName 'tmpmid.mid']);
    dos(['del ' file2PathName 'tmpmid.mid']);
end

%score_onsets contains the onset windows of each beat of the score
score_onsets=zeros(1,total_length);

for i=0:total_length-1

score_onsets(i+1)=(round(i*midiscore.beatDur*AudioFeatures.sr/AudioFeature
s.h));
end

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

%%
%collect all the available information regarding tempo indicated
%throughout the score directions and executed in the performance recording
%inside the 'tempo' cell array
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

waitbar(0.42,wait,'preprocessing (collect tempo information throughout
score) ');

%we ignore general details about the score file and collect the informtion
%stored in each separate measure (note pitch and intensity,
directions,etc)
%inside a new struct named measure
measure=file2struct.part.measure;

pin=1;
pend=length(measure);

%directions regarding tempo will be gathered in the struct tempo
tempo=cell(4,1000);

```

```

%in case there are more than one directions in one measure, we need to
keep
%track of the measure each direction belongs to, simply indexing the tempo
%and velocity arrays wont do

crnt_meas=pin;
i=pin;
while crnt_meas<=pend
%no field named 'direction' means no directions
    if isfield(measure(crnt_meas,1),'direction')
%there is a chance that there are more than one directions in a
%single measure
        if length(measure(crnt_meas,1).direction)>1
            j=1;
            while j<=length(measure(crnt_meas,1).direction)
                if ~isempty(measure(crnt_meas,1).direction(j,1))
%metronome direction
                    if isfield(measure(crnt_meas,1).direction(j,1)...
                        .direction_DASH_type,'metronome')
%bpm direction assigned to unit
                        bpm=measure(crnt_meas,1).direction(j,1)...
                            .direction_DASH_type.metronome...
                                .per_DASH_minute;
%unit in metronome direction (quarter note, eighth, etc)
                        if isfield(measure(crnt_meas,1).direction(j,1)...
                            .direction_DASH_type.metronome,'beat_DASH_unit_dot')
                            bpm=bpm*1.5;
                        else
                            switch measure(crnt_meas,1).direction(j,1)...
                                .direction_DASH_type.metronome.per_DASH_minute
                                case 'eighth'
                                    bpm=bpm/2;
                                case '16th'
                                    bpm=bpm/4;
                                case 'half'
                                    bpm=bpm*2;
                                end
                            end
                        end

                        tempo{1,i}=bpm;
                        tempo{4,i}=crnt_meas;
                        i=i+1;

%wedge direction standing either for diminuendo or crescendo
                    elseif isfield(measure(crnt_meas,1).direction(j,1)...
                        .direction_DASH_type,'wedge')
                        wedge=measure(crnt_meas,1).direction(j,1)...
                            .direction_DASH_type.wedge.ATTRIBUTE.type;

                    elseif isfield(measure(crnt_meas,1).direction(j,1)...
                        .direction_DASH_type,'dynamics')

dyn=cell2mat((fieldnames(measure(crnt_meas,1).direction(j,1)...
    .direction_DASH_type.dynamics)));

                    else isfield(measure(crnt_meas,1).direction(j,1)...

```

```

        .direction_DASH_type,'words')
text=measure(crnt_meas,1).direction(j,1)...
        .direction_DASH_type.words;
%there might exist an empty text direction which has to be ignored
if isempty(text)
    measure(crnt_meas,1).direction(j)=[];
    tempo{4,i}=crnt_meas;
    i=i+1;
    j=j-1;
else
    tmp=find(text~=' ',1);
    text=text(tmp:end);
    switch text(1:3)
        case 'acc'
            tempo{1,i}='accelerando';
        case 'rit'
            tempo{1,i}='ritenuto';
        case 'ral'
            tempo{1,i}='ritenuto';
        case 'dim'
        case 'cre'
        case 'dec'
    end

    switch text
        case 'stringendo'
            tempo{1,i}='accelerando';
        case 'Stringendo'
            tempo{1,i}='accelerando';
        case 'precipitando'
            tempo{1,i}='accelerando';
        case 'Precipitando'
            tempo{1,i}='accelerando';
        case 'allargando'
            tempo{1,i}='ritenuto';
        case 'Allargando'
            tempo{1,i}='ritenuto';
        case 'Calando'
            tempo{1,i}='ritenuto';
        case 'calando'
            tempo{1,i}='ritenuto';
        case 'lentando'
            tempo{1,i}='ritenuto';
        case 'Lentando'
            tempo{1,i}='ritenuto';
        case 'meno mosso'
            tempo{1,i}='meno mosso';
        case 'piu mosso'
            tempo{1,i}='piu mosso';
        case 'poco meno mosso'
            tempo{1,i}='meno mosso';
        case 'poco piu mosso'
            tempo{1,i}='piu mosso';
        case 'a tempo'
            tempo{1,i}='a tempo';
        case 'tempo primo'
            tempo{1,i}='a tempo';
        case 'Andante'
            tempo{1,i}='andante';

```

```

case 'andante'
    tempo{1,i}='andante';
case 'Lento'
    tempo{1,i}='lento';
case 'lento'
    tempo{1,i}='lento';
case 'Largo'
    tempo{1,i}='largo';
case 'largo'
    tempo{1,i}='largo';
case 'larghissimo'
    tempo{1,i}='larghissimo';
case 'Larghissimo'
    tempo{1,i}='larghissimo';
case 'Grave'
    tempo{1,i}='grave';
case 'grave'
    tempo{1,i}='grave';
case 'Larghetto'
    tempo{1,i}='larghetto';
case 'larghetto'
    tempo{1,i}='larghetto';
case 'Adagio'
    tempo{1,i}='adagio';
case 'adagio'
    tempo{1,i}='adagio';
case 'Adagietto'
    tempo{1,i}='adagietto';
case 'adagietto'
    tempo{1,i}='adagietto';
case 'Andante moderato'
    tempo{1,i}='andante moderato';
case 'andante moderato'
    tempo{1,i}='andante moderato';
case 'Andantino'
    tempo{1,i}='andantino';
case 'andantino'
    tempo{1,i}='andantino';
case 'Presto'
    tempo{1,i}='presto';
case 'presto'
    tempo{1,i}='presto';
case 'Prestissimo'
    tempo{1,i}='prestissimo';
case 'prestissimo'
    tempo{1,i}='prestissimo';
case 'Allegro'
    tempo{1,i}='allegro';
case 'allegro'
    tempo{1,i}='allegro';
case 'marcia moderato'
    tempo{1,i}='marcia moderato';
case 'Marcia moderato'
    tempo{1,i}='marcia moderato';
case 'moderato'
    tempo{1,i}='moderato';
case 'Moderato'
    tempo{1,i}='moderato';
case 'vivace'

```

```

        tempo{1,i}='vivace';
    case 'Vivace'
        tempo{1,i}='vivace';
    case 'vivicissimo'
        tempo{1,i}='vivicissimo';
    case 'Vivicissimo'
        tempo{1,i}='vivicissimo';
    case 'Allegrissimo'
        tempo{1,i}='allegrissimo';
    case 'allegrissimo'
        tempo{1,i}='allegrissimo';
    end
    tempo{4,i}=crnt_meas;
    i=i+1;
end
end
j=j+1;
else
    i=i+1;
end
end
else
    if ~isempty(measure(crnt_meas,1).direction)
%for a single measure direction
%metronome direction
        if isfield(measure(crnt_meas,1).direction...
            .direction_DASH_type,'metronome')
%bpm direction assigned to unit
            bpm=measure(crnt_meas,1).direction...
                .direction_DASH_type.metronome...
                    .per_DASH_minute;
%unit in metronome direction (quarter note, eighth, etc)
            if isfield(measure(crnt_meas,1).direction...
                .direction_DASH_type.metronome,'beat_DASH_unit_dot')
                    bpm=bpm*1.5;
            else
                switch measure(crnt_meas,1).direction...
                    .direction_DASH_type.metronome.per_DASH_minute
                case 'eighth'
                    bpm=bpm/2;
                case '16th'
                    bpm=bpm/4;
                case 'half'
                    bpm=bpm*2;
                end
            end
            tempo{1,i}=bpm;
            tempo{4,i}=crnt_meas;
            i=i+1;

%wedge direction standing either for diminuendo or crescendo
        elseif isfield(measure(crnt_meas,1).direction...
            .direction_DASH_type,'wedge')
            wedge=measure(crnt_meas,1).direction...
                .direction_DASH_type.wedge.Attributes.type;
        elseif isfield(measure(crnt_meas,1).direction...

```



```

        .direction_DASH_type,'dynamics')

dyn=cell2mat((fieldnames(measure(crnt_meas,1).direction...
        .direction_DASH_type.dynamics)));
    else isfield(measure(crnt_meas,1).direction...
        .direction_DASH_type,'words')
        text=measure(crnt_meas,1).direction...
        .direction_DASH_type.words;
    %there might exist an empty text direction which has to be ignored
    if isempty(text)
        measure(crnt_meas,1).direction=[];
        tempo{4,i}=crnt_meas;
        i=i+1;
    else
        tmp=find(text~= ' ',1);
        text=text(tmp:end);
        switch text(1:3)
            case 'acc'
                tempo{1,i}='accelerando';
            case 'rit'
                tempo{1,i}='ritenuto';
            case 'dim'
            case 'cre'
            case 'dec'
        end

        switch text
            case 'meno mosso'
                tempo{1,i}='meno mosso';
            case 'piu mosso'
                tempo{1,i}='piu mosso';
            case 'poco meno mosso'
                tempo{1,i}='meno mosso';
            case 'poco piu mosso'
                tempo{1,i}='piu mosso';
            case 'a tempo'
                tempo{1,i}='a tempo';
            case 'tempo primo'
                tempo{1,i}='a tempo';
            case 'Andante'
                tempo{1,i}='andante';
            case 'andante'
                tempo{1,i}='andante';
            case 'Lento'
                tempo{1,i}='lento';
            case 'lento'
                tempo{1,i}='lento';
            case 'Largo'
                tempo{1,i}='largo';
            case 'largo'
                tempo{1,i}='largo';
            case 'larghissimo'
                tempo{1,i}='larghissimo';
            case 'Larghissimo'
                tempo{1,i}='larghissimo';
            case 'Grave'
                tempo{1,i}='grave';
            case 'grave'

```

```

        tempo{1,i}='grave';
    case 'Larghetto'
        tempo{1,i}='larghetto';
    case 'larghetto'
        tempo{1,i}='larghetto';
    case 'Adagio'
        tempo{1,i}='adagio';
    case 'adagio'
        tempo{1,i}='adagio';
    case 'Adagietto'
        tempo{1,i}='adagietto';
    case 'adagietto'
        tempo{1,i}='adagietto';
    case 'Andante moderato'
        tempo{1,i}='andante moderato';
    case 'andante moderato'
        tempo{1,i}='andante moderato';
    case 'Andantino'
        tempo{1,i}='andantino';
    case 'andantino'
        tempo{1,i}='andantino';
    case 'Presto'
        tempo{1,i}='presto';
    case 'presto'
        tempo{1,i}='presto';
    case 'Prestissimo'
        tempo{1,i}='prestissimo';
    case 'prestissimo'
        tempo{1,i}='prestissimo';
    case 'Allegro'
        tempo{1,i}='allegro';
    case 'allegro'
        tempo{1,i}='allegro';
    case 'marcia moderato'
        tempo{1,i}='marcia moderato';
    case 'Marcia moderato'
        tempo{1,i}='marcia moderato';
    case 'moderato'
        tempo{1,i}='moderato';
    case 'Moderato'
        tempo{1,i}='moderato';
    case 'vivace'
        tempo{1,i}='vivace';
    case 'Vivace'
        tempo{1,i}='vivace';
    case 'vivacissimo'
        tempo{1,i}='vivacissimo';
    case 'Vivacissimo'
        tempo{1,i}='vivacissimo';
    case 'Allegrissimo'
        tempo{1,i}='allegrissimo';
    case 'allegrissimo'
        tempo{1,i}='allegrissimo';
    end
    tempo{4,i}=crrt_meas;
    i=i+1;
end
else
    end

```

```

        i=i+1;
    end
end
else
    i=i+1;
end
if isempty(tempo{4,i-1})
    tempo{4,i-1}=crnt_meas;
end
crnt_meas=crnt_meas+1;
end
l=length(tempo);
tempo(:,i:l)=[];

assignin('base','tempo',tempo);

%%
%collect in one struct all the information about the score file (xml file,
$xml data in a struct, length of a measure in beats, total length in
beats,
%beat onsets and midiscore, a matrix with midi data as described above
ScoreFeatures=struct;
ScoreFeatures.file2xml=file2xml;
ScoreFeatures.file2struct=file2struct;
ScoreFeatures.scorexml=scorexml;
ScoreFeatures.measure_length=measure_length;
ScoreFeatures.total_length=total_length;
ScoreFeatures.score_onsets=score_onsets;
ScoreFeatures.midiscore=midiscore;

assignin('base','ScoreFeatures',ScoreFeatures);

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

function [alignment,wait]=align(AudioFeatures,wait,S)
%this function takes the spectral representations of the input files from
%the AudioFeatures struct and aligns them by calculating their Similarity
%Matrix, their Cost Matrix (cumulative similarity matrix) and eventually
%their alignment path

D1=AudioFeatures.D1;
D2=AudioFeatures.D2;
method=AudioFeatures.method;

```

```

w=AudioFeatures.w;
h=AudioFeatures.h;
sr=AudioFeatures.sr;

%%
%calculate similarity matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
waitbar(0.45,wait,'alignment (calculate similarity matrix)');
% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
%Generate the simmlarity matrix using either cosine or euclidean distance
% SM = simmx_cos(abs(D1),abs(D2));
SM = simmx_euc(abs(D1),abs(D2));
% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
%%
%caclulate cost matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
waitbar(0.55,wait,'alignment (caclulate cost matrix)');

% Use dynamic programming to find the lowest-cost path between the
% opposite corners of the cost matrix
[p,q,C,s] = dpfast(SM);
% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

```

```

end
%%
%If chroma representation has been used, calculate the spectrogram as well
%in order to resynthesize accurately the warped version.
if (method==2) || (method==3)
    D2 = spectrogram(d2,w,w-h,w,sr);
end

%%
%calculate alignment path
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
waitbar(0.65,wait,'alignment (calculate alignment path)');
% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
%alignment_path: vector aligning the spectral representations of the
%recording and the synthesized score.
%starting_frame and ending_frame: anything before and after those frames
is
%silence.
alignment_path = zeros(1, size(D2,2));
for i = 1:length(alignment_path );
    alignment_path (i) = p(min(find(q >= i)));
end

[starting_frame,ending_frame]=boundary_values...
    (AudioFeatures.dl,Dl,AudioFeatures.w,AudioFeatures.h);

%%
%collect in one file all the information calculated above
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
alignment.alignment_path=alignment_path;
alignment.starting_frame=starting_frame;
alignment.ending_frame=ending_frame;
alignment.SM=SM;
alignment.C=C;
alignment.p=p;
alignment.q=q;

assignin('base','alignment',alignment);

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');

```

```

end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

```

```

function

```

```

[performance,wait]=performance_description(ScoreFeatures,AudioFeatures,alignment,wait,S)

```

```

%initializations

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

```

```

alignment_path=alignment.alignment_path;
starting_frame=alignment.starting_frame;

```

```

h=AudioFeatures.h;
sr=AudioFeatures.sr;

```

```

score_onsets=ScoreFeatures.score_onsets;
score=ScoreFeatures.midiscore.score;
beatDur=ScoreFeatures.midiscore.beatDur;
measure_length=ScoreFeatures.measure_length;
total_length=ScoreFeatures.total_length;

```

```

%%
%the vector rec_onsets will be filled with the onset windows of each beat
%of the recording
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

```

```

rec_onsets=zeros(1,total_length);
waitbar(0.75,wait,'performance description(calculate recording beat
onsets) ');

```

```

%If the condition: midiscore.score(1)~=0 is true, the midi notes dont
begin
%at midi time 0 (zero) which means that the score begins with a rest! In
%this case we have to approximate the first rec_onset values.
%In case that the score sets off at the midi time 0 the value starting
%frame will be assigned to rec_onset(1)

```

```

if score(1)~=0
    k=ceil(score(1)+1);
    for i=k:total_length
        rec_onsets(i)=alignment_path(score_onsets(i));
    end
    x=round(mean(diff(rec_onsets(4:end))));
    for i=1:k-1
        rec_onsets(k-i)=rec_onsets(k-i+1)-x;
        if rec_onsets(k-i)<0
            rec_onsets(k-i)=1;
        end
    end
end
end

```

```

else
    rec_onsets(1)=starting_frame;
    for i=2:total_length
        rec_onsets(i)=alignment_path(score_onsets(i));
    end
end
end
% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
end
%%
%%iMi stands for inter measure intervals, it is a struct containing the
%%length of each measure calculated in seconds, windows and beats as well
as
%%the onset of each measure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
iMi=struct;
waitbar(0.78,wait,'performance description(calculate measure
information)');

j=1;
for i=1:measure_length:length(rec_onsets)
    iMi.measure_onsets(j)=rec_onsets(i);
    j=j+1;
end

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
end
x=rec_onsets(total_length)-rec_onsets(total_length-1);
x=rec_onsets(total_length)+x;
y=horzcat(iMi.measure_onsets,x);
iMi.win=diff(y);

%Measure length in seconds
iMi.sec=iMi.win*h/sr;

%Measure length in beats
iMi.beat=iMi.sec/beatDur;

```

```

%k equals 60 (seconds) * measure length in beats
k=measure_length*60;

for i=1:length(iMi.beat)
iMi.bpm(i)=k./iMi.sec(i);
end

assignin('base','iMi',iMi);

%%
%performance is the struct containing the main performance (recording)
%characteristics : tempo of each measure in beats per minute (bpm),
measure
%onsets and beat onsets
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
performance.bpm=iMi.bpm;
performance.rec_beat_onsets=rec_onsets;
performance.rec_measure_onsets=iMi.measure_onsets;

assignin('base','performance',performance);
% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

function [tempo,wait]=enhance(tempo,performance,wait,S)
%This function produces an enhanced version of the tempo cell array,
%containing the "expected" tempo or tempo change of each measure (in bpm)
%as given by the score directions (which are also added in tempo cell
array)

waitbar(0.81,wait,'score - performance comparison');

bpm=performance.bpm;

pin=1;
pend=length(tempo);
crnt_tempo=0;

for i=pin:pend
    if isempty(tempo{1,i})
        tempo{3,i}=bpm(tempo{4,i});
        tempo{2,i}=crnt_tempo;
    elseif isa(tempo{1,i},'numeric')

```



```

    crnt_tempo=tempo{1,i};
    tempo{3,i}=bpm(tempo{4,i});
    tempo{2,i}=crnt_tempo;
else isa(tempo{1,i},'char')
    if strcmp(tempo{1,i},'accelerando')
        crnt_tempo=1;
    elseif strcmp(tempo{1,i},'ritenuto')
        crnt_tempo=-1;
    elseif strcmp(tempo{1,i},'a tempo')
        j=i-1;
        if (tempo{2,j}===-1) || (tempo{2,j}==1) || (tempo{2,j}==
2) || ...
            (tempo{2,j}==2)
        while (tempo{2,j}===-1) || (tempo{2,j}==1) ||
(tempo{2,j}===-2) || ...
            (tempo{2,j}==2)
            j=j-1;
        end

        if j==1
            crnt_tempo=tempo{3,1};
        else
            crnt_tempo=[tempo{3,j},tempo{3,j-1}];
            crnt_tempo=mean(crnt_tempo);
        end

    elseif tempo{2,j}==0
        if j==1
            crnt_tempo=tempo{3,1};
        else
            crnt_tempo=[tempo{3,j},tempo{3,j-1}];
            crnt_tempo=mean(crnt_tempo);
        end
    else
        crnt_tempo=tempo{3,j};
    end
elseif strcmp(tempo{1,i},'meno mosso')
    crnt_tempo=-2;
elseif strcmp(tempo{1,i},'piu mosso')
    crnt_tempo=2;
else
    switch tempo{1,i}
    case 'lento'
        min=40;
        max=45;
    case 'presto'
        min=168;
        max=177;
    case 'allegro'
        min=109;
        max=132;
    case 'andante'
        min=73;
        max=77;
    case 'largo'
        min=45;
        max=50;
    case 'larghissimo'
        min=0;

```

```

        max=19;
    case 'grave'
        min=20;
        max=40;
    case 'larghetto'
        min=50;
        max=55;
    case 'adagio'
        min=55;
        max=65;
    case 'adagietto'
        min=65;
        max=69;
    case 'andante moderato'
        min=69;
        max=72;
    case 'andantino'
        min=78;
        max=83;
    case 'prestissimo'
        min=178;
        max=300;
    case 'marcia moderato'
        min=83;
        max=85;
    case 'moderato'
        min=86;
        max=98;
    case 'vivace'
        min=132;
        max=140;
    case 'vivacissimo'
        min=140;
        max=150;
    case 'allegro'
        min=150;
        max=167;
    end
    crnt_tempo={min,max};
end
tempo{3,i}=bpm(tempo{4,i});
tempo{2,i}=crnt_tempo;
end
assignin('base','tempo',tempo);
end

% Check for Cancel button press
if getappdata(wait,'canceling')
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end
if ~ishandle(S.fh) % Check if the figure exists.
    watchoff;
    drawnow;
    delete(wait);
    errordlg('execution terminated','watch out!');
end

```

end

```
function wait=judge(tempo,ScoreFeatures,wait,S)
%this function "judges" whether the score directions are followed and
%proceeds to export a new, augmented score

waitbar(0.84,wait,'score - performance comparison (message creation)');

file2xml=ScoreFeatures.file2xml;
scorexml=ScoreFeatures.scorexml;

pend=length(tempo);
i=1;
msg=cell(pend,2);
msg_ind=1;

while i<=pend
    % Check for Cancel button press
    if getappdata(wait,'canceling')
        watchoff;
        drawnow;
        delete(wait);
        errordlg('execution terminated','watch out!');
    end
    if ~ishandle(S.fh) % Check if the figure exists.
        watchoff;
        drawnow;
        delete(wait);
        errordlg('execution terminated','watch out!');
    end
    %this measure has a tempo indicated by a qualitative direction, e.g.
    %adante,allegro, ...
    if iscell(tempo{2,i})
        j=i;
        i=i+1;

        %the direction should be followed in measures j throught i
        while i<=length(tempo)
            if ~isequal(tempo{2,i},tempo{2,j})
                break
            end
            i=i+1;
        end

        i=i-1;

        %tempo indicated by the score
        score_tmp=tempo{2,j};
        %actual mean tempo of the performance
        rec_tmp=mean([tempo{3,j:i}]);

        if rec_tmp>1.08*cell2mat(score_tmp(2))
```

```

        %store message warning that tempo is too fast
        msg{msg_ind,1}=[num2str(msg_ind),' ) WARNING for
measures',num2str(tempo{4,j}),' to ',num2str(tempo{4,i}),' : tempo is too
fast'];
        msg{msg_ind,2}=tempo{4,j};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FFA500';
        color_notes(tempo{4,j},tempo{4,i},scorexml,color);

    elseif rec_tmp<.92*cell2mat(score_tmp(1))
        %store message warning that tempo is too slow
        msg{msg_ind,1}=[num2str(msg_ind),' ) WARNING for measures '...
        ,num2str(tempo{4,j}),' to ',num2str(tempo{4,i}),' : tempo
is too slow'];
        msg{msg_ind,2}=tempo{4,j};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FFA500';
        color_notes(tempo{4,j},tempo{4,i},scorexml,color);
    elseif (tempo{3,j}>1.08*tempo{3,i})||(tempo{3,j}<.92*tempo{3,i})
        %store message warning that tempo is not steady (although it
        %appears to be) since the tempo in the first measure is
        %noticeably different than the one in the last
        msg{msg_ind,1}=[num2str(msg_ind),' ) WARNING in measures '...
        ,num2str(tempo{4,j}),' to ',num2str(tempo{4,i}),' : tempo
does not appear to be steady'];
        msg{msg_ind,2}=tempo{4,j};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FFA500';
        color_notes(tempo{4,j},tempo{4,i},scorexml,color);
    end

%accelerando
    elseif tempo{2,i}==1
        %the tempo should be accelerated from measure accin till measure i
        accin=i;
        i=i+1;
        while i<=length(tempo)
            if ~isequal(tempo{2,i},1)
                break
            end
            i=i+1;
        end
        i=i-1;

        %calculate slope of accelerating measures
        if i==length(tempo)||isequal(tempo{2,i+1},-1)||
strcmp(tempo{1,i+1},'a tempo')
            acc=round(cell2mat({tempo{3,accin:i}}));
        else
            acc=round(cell2mat({tempo{3,accin:i+1}}));
        end
        slope=diff(acc);

        %calculate tempo right before accelerando

```

```

if accin==1
    intempo=tempo{3,1};
elseif accin==2
    intempo=[tempo{3,1},tempo{3,2}];
    intempo=mean(intempo);
else
    intempo=[tempo{3,accin-1},tempo{3,accin-2}];
    intempo=mean(intempo);
end

%calculate tempo right after accelerando
if i==length(tempo)||isequal(tempo{2,i+1},-1)||
strcmp(tempo{1,i+1},'a tempo')
    outmeasure=i;
    outtempo=tempo{3,outmeasure};
elseif i==length(tempo)-1
    outmeasure=i+1;
    outtempo=tempo{3,outmeasure};
else
    outmeasure=i+1;
    outtempo=[tempo{3,outmeasure},tempo{3,outmeasure+1}];
    outtempo=mean(outtempo);
end

%store error messages
if mean(slope)<=0
    msg[msg_ind,1]=[num2str(msg_ind),' ) ERROR in measures '...
        ,num2str(tempo{4,accin}),' to ',num2str(tempo{4,i}),' : no
acceleration was executed'];
    msg[msg_ind,2]=tempo{4,accin};
    msg_ind=msg_ind+1;
    %color the corresponding noteheads
    color='#FF0000';
    color_notes(tempo{4,accin},tempo{4,i},scorexml,color);
%store message warning that, although the tempo is accelerated,
%acceleration is not steadily executed
elseif ~isempty(find(slope<0,1))
    if length(find(slope<0))>1
        %there are more than one decelerating measure in an
        %accelerating passage
        msg[msg_ind,1]=[num2str(msg_ind),' ) WARNING for measures
'...
            ,num2str(tempo{4,accin}),' to ',num2str(tempo{4,i}),' :
acceleration is not steady'];
        msg[msg_ind,2]=tempo{4,accin};
        msg_ind=msg_ind+1;
        color='#FFA500';
        color_notes(tempo{4,accin},tempo{4,i},scorexml,color);
    else
        %there is only one decelerating measure in the
        %accelerating passage
        ind=accin+find(slope<0);
        msg[msg_ind,1]=[num2str(msg_ind),' ) WARNING for measure
'...
            ,num2str(tempo{4,ind}),' :decelerating measure inside
an accelerating passage'];
        msg[msg_ind,2]=tempo{4,ind};
        msg_ind=msg_ind+1;
        color='#FFA500';

```

```

        color_notes(tempo{4,ind},tempo{4,ind},scorexml,color);
    end
end

%store error message that, although the passage accelerates, the
%measure coming after it is not as fast as it should be
if mean(slope)>0&&outtempo<=1.08*intempo
    if outmeasure==i
        msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR in measure '...
            ,num2str(tempo{4,outmeasure}),' :accelerating passage
does not end with a tempo as fast as expected '];
        msg{msg_ind,2}=tempo{4,outmeasure};
    else
        msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR in measure
',num2str(...
            tempo{4,outmeasure}),'...
            ' : tempo following acceleration is not as fast as it
should be'];
        msg{msg_ind,2}=tempo{4,outmeasure};
    end
    msg_ind=msg_ind+1;
    %color the corresponding noteheads
    color='#FF0000';

color_notes(tempo{4,outmeasure},tempo{4,outmeasure},scorexml,color);
end

%ritenuto
elseif tempo{2,i}==-1
    %the tempo should be decelerated from measure rit till measure i
    ritin=i;
    i=i+1;
    while i<=length(tempo)
        if ~isequal(tempo{2,i},-1)
            break
        end
        i=i+1;
    end

    i=i-1;

    %calculate slope of decelerating measures
    if i==length(tempo)||strcmp(tempo{1,i+1},'a tempo')||
isequal(tempo{2,i+1},1)
        rit=round(cell2mat({tempo{3,ritin:i}}));
    else
        rit=round(cell2mat({tempo{3,ritin:i+1}}));
    end
    slope=diff(rit);

    %calculate tempo right before ritenuto
    if ritin==1
        intempo=tempo{3,1};
    elseif ritin==2
        intempo=[tempo{3,1},tempo{3,2}];
        intempo=mean(intempo);
    else

```

```

        intempo=[tempo{3,ritin-1},tempo{3,ritin-2}];
        intempo=mean(intempo);
    end

    %calculate tempo right after ritenuto
    if i==length(tempo)||strcmp(tempo{1,i+1},'a tempo')||
isequal(tempo{2,i+1},1)
        outmeasure=i;
        outtempo=tempo{3,outmeasure};
    elseif i==length(tempo)-1
        outmeasure=i+1;
        outtempo=tempo{3,outmeasure};
    else
        outmeasure=i+1;
        outtempo=[tempo{3,outmeasure},tempo{3,outmeasure+1}];
        outtempo=mean(outtempo);
    end

    %store warning and error messages
    if mean(slope)>=0
        msg[msg_ind,1]=[num2str(msg_ind),' ) ERROR in measures '...
            ,num2str(tempo{4,ritin}),' to ',num2str(tempo{4,i}),' : no
deceleration was executed'];
        msg[msg_ind,2]=tempo{4,ritin};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FF0000';
        color_notes(tempo{4,ritin},tempo{4,i},scorexml,color);
        %store message warning that, although the tempo is decelerated,
        %deceleration is not steadily executed
    elseif ~isempty(find(slope>0,1))
        if length(find(slope>0))>1
            %there are more than one accelerating measure in a
            %decelerating passage
            msg[msg_ind,1]=[num2str(msg_ind),' ) WARNING for measures
'...
                ,num2str(tempo{4,ritin}),' to ',num2str(tempo{4,i}),' :
deceleration is not steady'];
            msg[msg_ind,2]=tempo{4,ritin};
            msg_ind=msg_ind+1;
            color='#FFA500';
            color_notes(tempo{4,accin},tempo{4,i},scorexml,color);
        else
            %there is only one accelerating measure in the
            %decelerating passage
            ind=ritin+find(slope>0);
            msg[msg_ind,1]=[num2str(msg_ind),' ) WARNING for measure
'...
                ,num2str(tempo{4,ind}),' :accelerating measure inside a
decelerating passage'];
            msg[msg_ind,2]=tempo{4,ind};
            msg_ind=msg_ind+1;
            color='#FFA500';
            color_notes(tempo{4,ind},tempo{4,ind},scorexml,color);
        end
    end

    %store error message that, although the passage decelerates, the

```

```

%measure coming after it is not as slow as it should be
if mean(slope)<0&&intempo<=1.08*outtempo
    if outmeasure==i
        msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR in measure '...
            ,num2str(tempo{4,outmeasure}),' :decelerating passage
does not end with a tempo as slow as expected'];
        msg{msg_ind,2}=tempo{4,outmeasure};
    else
        msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR in measure
'...
            ,num2str(tempo{4,outmeasure}),' : tempo following
deceleration is not as slow as it should be'];
        msg{msg_ind,2}=tempo{4,outmeasure};
    end
    msg_ind=msg_ind+1;
    %color the corresponding noteheads
    color='#FF0000';

color_notes(tempo{4,outmeasure},tempo{4,outmeasure},scorexml,color);
end

%piu mosso
elseif tempo{2,i}==2
    in=i;
    i=i+1;
    while i<=length(tempo)
        if ~isequal(tempo{2,i},2)
            break
        end
        i=i+1;
    end

    i=i-1;

    m=[tempo{3,in:i}];
    rec_tmp=mean(m);
    %first_msr is the measure where the "piu mosso" direction appears
    first_msr=tempo{3,in};
    %score_tmp is the "expected" tempo, %8 faster than the previous
    %tempo
    score_tmp=1.08*tempo{3,in-1};

    %store ERROR message in case that the tempo in the measure where
piu
%mosso appears is slower than expected AND the mean tempo of the
%following measures is slower than expected
if (rec_tmp<score_tmp)&&(first_msr<score_tmp)
    msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR in measure '...
        ,num2str(tempo{4,in}),' : tempo did not get any faster'];
    msg{msg_ind,2}=tempo{4,in};
    msg_ind=msg_ind+1;
    color='#FF0000';
    color_notes(tempo{4,in},tempo{4,in},scorexml,color);
end

%meno mosso

```



```

elseif tempo{2,i}==-2
    in=i;
    i=i+1;
    while i<=length(tempo)
        if ~isequal(tempo{2,i},-2)
            break
        end
        i=i+1;
    end

    i=i-1;

    m=[tempo{3,in:i}];
    %actual mean tempo of the performance
    rec_tmp=mean(m);
    %first_msr is the measure where the "piu mosso" direction appears
    first_msr=tempo{3,in};
    %score_tmp is the "expected" tempo, %8 slower than the previous
    %tempo
    score_tmp=0.92*tempo{3,in-1};

    %store ERROR message in case that the tempo in the measure where
    %meno mosso appears is faster than expected AND the mean tempo of
    %the following measures is faster than expected
    if (rec_tmp>score_tmp)&&(first_msr>score_tmp)
        msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR in measure '...
            ,num2str(tempo{4,in}),' : tempo did not get any slower'];
        msg{msg_ind,2}=tempo{4,in};
        msg_ind=msg_ind+1;
        color='#FF0000';
        color_notes(tempo{4,in},tempo{4,in},scorexml,color);
    end

%direction indicating that tempo should be kept steady and equal to a
%specific number of beats per minute (e.g. 1 quarter = 120 bpm)
elseif tempo{2,i}~=0
    j=i;
    i=i+1;
    while i<=length(tempo)
        if ~isequal(tempo{2,i},tempo{2,j})
            break
        end
        i=i+1;
    end

    i=i-1;

    %tempo indicated by the score
    score_tmp=tempo{2,j};
    %actual mean tempo of the performance
    rec_tmp=mean([tempo{3,j:i}]);

    %'a tempo' is a subcase
    if strcmp(tempo{1,j},'a tempo')
        if or(rec_tmp>1.08*score_tmp,...
            rec_tmp<.92*score_tmp)
            msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR: a tempo
(measure '...

```

```

        ,num2str(tempo{4,j}),' ) was not executed'];
    msg{msg_ind,2}=tempo{4,j};
    msg_ind=msg_ind+1;
    %color the corresponding noteheads
    color='#FF0000';
    color_notes(tempo{4,j},tempo{4,j},scorexml,color);
    end
%tempo is faster than it should be
    elseif rec_tmp>1.16*score_tmp
        msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR in measures '...
            ,num2str(tempo{4,j}),' to ',num2str(tempo{4,i}),' : tempo
is too fast'];
        msg{msg_ind,2}=tempo{4,j};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FF0000';
        color_notes(tempo{4,j},tempo{4,i},scorexml,color);
    elseif rec_tmp>1.08*score_tmp
        msg{msg_ind,1}=[num2str(msg_ind),' ) WARNING in measures '...
            ,num2str(tempo{4,j}),' to ',num2str(tempo{4,i}),' : tempo
is faster than indicated'];
        msg{msg_ind,2}=tempo{4,j};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FFA500';
        color_notes(tempo{4,j},tempo{4,i},scorexml,color);

%tempo is slower than it should be
    elseif rec_tmp<.84*score_tmp
        msg{msg_ind,1}=[num2str(msg_ind),' ) ERROR in measures '...
            ,num2str(tempo{4,j}),' to ',num2str(tempo{4,i}),' : tempo
is too slow'];
        msg{msg_ind,2}=tempo{4,j};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FF0000';
        color_notes(tempo{4,j},tempo{4,i},scorexml,color);
    elseif rec_tmp<.92*score_tmp
        msg{msg_ind,1}=[num2str(msg_ind),' ) WARNING in measures '...
            ,num2str(tempo{4,j}),' to ',num2str(tempo{4,i}),' : tempo
is slower than indicated'];
        msg{msg_ind,2}=tempo{4,j};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FFA500';
        color_notes(tempo{4,j},tempo{4,i},scorexml,color);

%mean tempo appears to be steady yet the first measure differs
%noticeably in tempo compared to the last one
    elseif (tempo{3,j}>1.08*tempo{3,i})||(tempo{3,j}<.92*tempo{3,i})
        msg{msg_ind,1}=[num2str(msg_ind),' ) WARNING in measures '...
            ,num2str(tempo{4,j}),' to ',num2str(tempo{4,i}),' : tempo
does not appear to be steady'];
        msg{msg_ind,2}=tempo{4,j};
        msg_ind=msg_ind+1;
        %color the corresponding noteheads
        color='#FFA500';
        color_notes(tempo{4,j},tempo{4,i},scorexml,color);
    end

```

```

        end
        i=i+1;
    end

waitbar(0.94,wait,'extract new score');

%remove empty cells
a=msg';
msg = reshape(a(~cellfun(@isempty,a)),2,[]);

%display the stored messages
display_messages(scorexml,msg);

    % Check for Cancel button press
    if getappdata(wait,'canceling')
        watchoff;
        drawnow;
        delete(wait);
        errordlg('execution terminated','watch out!');
    end
    if ~ishandle(S.fh) % Check if the figure exists.
        watchoff;
        drawnow;
        delete(wait);
        errordlg('execution terminated','watch out!');
    end
end

```