



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ**

**Αυτόματη Καταγραφή Μουσικής από  
Μονοφωνικό Πνευστό Όργανο**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

της

**ΚΟΛΙΟΠΟΥΛΟΥ Ν. ΕΥΑΓΓΕΛΙΑΣ**

**Επιβλέπων :** ΚΑΡΑΓΙΑΝΝΗΣ ΓΕΩΡΓΙΟΣ  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2010





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

## Αυτόματη Καταγραφή Μουσικής από Μονοφωνικό Πνευστό Όργανο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

**ΚΟΛΙΟΠΟΥΛΟΥ Ν. ΕΥΑΓΓΕΛΙΑΣ**

**Επιβλέπων :** ΚΑΡΑΓΙΑΝΝΗΣ ΓΕΩΡΓΙΟΣ  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18<sup>η</sup> ΟΚΤΩΒΡΙΟΥ 2010 .

.....  
ΚΑΡΑΓΙΑΝΝΗΣ ΓΕΩΡΓΙΟΣ  
Καθηγητής Ε.Μ.Π.

.....  
ΣΕΛΛΗΣ ΤΙΜΟΛΕΩΝ  
Καθηγητής Ε.Μ.Π.

.....  
ΤΑΜΠΟΥΡΑΤΖΗΣ ΓΕΩΡΓΙΟΣ  
Ερευνητής στο Ινστιτούτο  
Επεξεργασίας του Λόγου / Ε.Κ. "Αθηνά"

Αθήνα, Οκτώβριος 2010

.....  
**ΚΟΛΙΟΠΟΥΛΟΥ Ν. ΕΥΑΓΓΕΛΙΑ**

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

© 2010 – All rights reserved

## Περίληψη

Το αντικείμενο της διπλωματικής είναι η υλοποίηση μίας αυτόματης διαδικασίας με την οποία θα δημιουργείται η παρτιτούρα (\*.MusicXml) η οποία απαντά σε μία ηχογράφιση μονοφωνικού πνευστού οργάνου (\*.wav). Ξεκινώντας με τη μελέτη βασικών εννοιών της θεωρίας επεξεργασίας σήματος και αναγνώρισης προτύπων, η εργασία αυτή συνεχίζει με τη συλλογή των απαραίτητων στοιχείων για την υλοποίηση του «αναγνωριστή» βασικών παραμέτρων για τη δημιουργία μίας παρτιτούρας. Με την αξιοποίηση των πληροφοριών αυτών σχεδιάζεται και υλοποιείται η διαδικασία μέσω της οποίας δημιουργείται αυτόματα ένα MusicXml αρχείο το οποίο περιγράφει με το βέλτιστο τρόπο το ακουστικό σήμα, διατηρώντας ταυτόχρονα χαμηλή πολυπλοκότητα στο πλαίσιο της αποφυγής λανθασμένων εκτιμήσεων. Η εργασία συνοδεύεται τέλος από δυο ξεχωριστές διαδικασίες: αυτή της αξιολόγησης της παρτιτούρας εξόδου μετά από σύγκρισή της με την παρτιτούρα αναφοράς για μία σειρά δεδομένων ελέγχου και αυτή της αναδημιουργίας της παρτιτούρας μετά τη διόρθωση της ακολουθίας από νότες που έχει παραχθεί.

**Λέξεις Κλειδιά:** << αυτόματη καταγραφή μουσικής, επεξεργασία μουσικού σήματος, παρτιτούρα/μουσική σημειογραφία, MusicXml, αναγνώριση θεμελιώδους συχνότητας μουσικού σήματος (pitch) >>



## **Abstract**

The subject of the thesis is the implementation of an automatic process by which the score (\*.MusicXml) that responds to a mono wind instrument recording (\*.wav) is created. After studying the basic principles of digital signal processing and pattern recognition theory, we continue by collecting the necessary data for the creation of the recognizer which will offer us the basic parameters in order to create a score. Finally, by using these information we present the conceptualization and the development of a system that automatically create a music xml file, which defines precisely the acoustic signal while having low complexity so as to avoid misestimates. The system is accompanied by two separate processes: the first one help us re-transcript the score after correcting the sequence of notes produced and the other allows us to evaluate the output score by comparing it with the reference score.

**Keywords:** << automatic transcription of music, music signal processing, score/music notation/sheet music, MusicXml, pitch recognition >>





## **Ευχαριστίες**

*Καταρχήν, θα ευχαριστήσω την κ. Β. Καντερέ για την άμεση ανταπόκρισή της στο ενδιαφέρον που εξέφρασα για την ενασχόλησή μου με ένα συναφές θέμα διπλωματικής το οποίο είχε προταθεί σε προηγούμενο έτος και η οποία με παρότρυνε να απευθυνθώ στον κ. Τ. Σελλή.*

*Εκφράζω τις θερμές μου ευχαριστίες στον κ. Τ. Σελλή για την συνεργασία μου με το Ινστιτούτο Επεξεργασίας του Λόγου του Ερευνητικού Κέντρου “Αθηνά” και την όλη προσφορά του για την πραγματοποίηση της παρούσας εργασίας.*

*Επιπλέον, οφείλω θερμότερες ευχαριστίες στον κ. Γ. Ταμπουρατζή ο οποίος με την υπομονή, τη θέλησή του και τις συνεχείς και άοκνες νουθεσίες και υποδείξεις του, με κατεύθυνε και με κατετόπιζε στα διάφορα προκύπτοντα ζητήματα της εργασίας μου. Φυσικά θέλω ακόμη να ευχαριστήσω από την ομάδα του Ι.Ε.Λ. τον κ. Θ. Γουλιά για την πολύτιμη συμβολή του στην υλοποίηση της παρούσας εργασίας, καθώς και τον κ. Κ. Περήφανο για την επιδειχθείσα προθυμία του να ασχοληθεί με το θέμα μου.*

*Τέλος, είμαι ευγνώμων στον κ. Γ. Καραγιάννη για την αποδοχή εκ μέρους του, της ανάθεσής μου του παρόντος θέματος διπλωματικής εργασίας.*



# Πίνακας Περιεχομένων

|                                                                                    |           |
|------------------------------------------------------------------------------------|-----------|
| <b>1 Εισαγωγή.....</b>                                                             | <b>1</b>  |
| 1.1 Αυτόματη Καταγραφή Μουσικής.....                                               | 1         |
| 1.2 Αντικείμενο Διπλωματικής.....                                                  | 3         |
| 1.3 Οργάνωση Κειμένου.....                                                         | 4         |
| <b>2 Ο Υπολογιστής ως Μουσικό Εργαλείο.....</b>                                    | <b>5</b>  |
| 2.1 Διαφορετικές Προσεγγίσεις της Μουσικής Επεξεργασίας στον Πραγματικό Κόσμο..... | 5         |
| 2.2 Αυτόματη Μουσική Καταγραφή και Μουσική Σημειογραφία από Ηλεκτρονικό Υπολογιστή | 6         |
| 2.2.1 Εισαγωγικές Έννοιες.....                                                     | 6         |
| 2.2.2 Η Φυσική Διάσταση των Βασικών Εννοιών της Μουσικής.....                      | 8         |
| 2.2.3 Αυτόματη Καταγραφή Μουσικής - Σχετικές Εργασίες.....                         | 9         |
| 2.2.4 Εκμάθηση Μουσικής με Χρήση Υπολογιστή - Ψηφιακή Μουσική Σημειογραφία.....    | 12        |
| 2.2.5 Συνεισφορά Εργασίας.....                                                     | 16        |
| <b>3 Θεωρητικό υπόβαθρο και εργαλεία.....</b>                                      | <b>17</b> |
| 3.1 Επεξεργασία Μουσικού Σήματος με χρήση του Praat.....                           | 17        |
| 3.2 Αναγνώριση Προτύπων.....                                                       | 21        |
| 3.2.1 Αλγόριθμοι αναγνώρισης προτύπων.....                                         | 22        |
| 3.2.2 Η μέθοδος του κοντινότερου γείτονα (Nearest Neighbor Classification).....    | 22        |
| 3.3 Οι προδιαγραφές του προτύπου MusicXml.....                                     | 23        |
| 3.4 Μετρική Ομοιότητας Συμβολοσειρών (Levenshtein distance).....                   | 27        |
| <b>4 Ιδέα και Υλοποίηση .....</b>                                                  | <b>29</b> |
| 4.1 Στάδιο πρώτο.....                                                              | 30        |
| 4.1.1 Τεχνικά χαρακτηριστικά κλαρινέτου.....                                       | 30        |
| 4.1.2 Συλλογή πραγματικών δεδομένων.....                                           | 31        |
| 4.1.3 Παράδειγμα διαδικασίας.....                                                  | 32        |
| 4.1.4 Συμπεράσματα.....                                                            | 36        |
| 4.2 Στάδιο δεύτερο.....                                                            | 36        |
| 4.2.1 Εφαρμογή 3-Nearest-Neighbors.....                                            | 36        |
| 4.2.2 Παράδειγμα εκτέλεσης.....                                                    | 37        |
| 4.2.3 Συμπεράσματα.....                                                            | 39        |
| 4.3 Στάδιο τρίτο.....                                                              | 39        |
| 4.3.1 Σχεδίαση συστήματος.....                                                     | 39        |
| 4.3.2 Έλεγχος.....                                                                 | 46        |
| 4.3.3 Αξιολόγηση.....                                                              | 50        |
| 4.3.4 Δυνατότητα διόρθωσης παρτιτούρας.....                                        | 52        |
| <b>5 Επίλογος.....</b>                                                             | <b>53</b> |
| 5.1 Σύνοψη & Συμπεράσματα.....                                                     | 53        |
| 5.2 Μελλοντικές Επεκτάσεις.....                                                    | 54        |

|                                                    |            |
|----------------------------------------------------|------------|
| <b>Παράρτημα Α – Το Ερευνητικό Έργο VEMUS.....</b> | <b>55</b>  |
| <b>Παράρτημα Β – Πληροφορίες Υλοποίησης.....</b>   | <b>58</b>  |
| • <b>B1. k - Nearest - Neighbors.....</b>          | <b>58</b>  |
| • <b>B2. Αυτόματη δημιουργία score.xml.....</b>    | <b>69</b>  |
| • <b>B3. Αξιολόγηση αρχείου εξόδου.....</b>        | <b>87</b>  |
| • <b>B4. Διόρθωση αρχείου εξόδου.....</b>          | <b>90</b>  |
| <b>6 Βιβλιογραφία.....</b>                         | <b>102</b> |

# 1

## Εισαγωγή

### 1.1 Αυτόματη Καταγραφή Μουσικής

Πριν από 50 περίπου χρόνια οι άνθρωποι επιχείρησαν να χρησιμοποιήσουν τον ηλεκτρονικό υπολογιστή προς όφελος της μουσικής. Το βασικό πρόβλημα το οποίο αντιμετώπιζαν ωστόσο ήταν ότι ούτε το μουσικό σήμα ούτε η κλασική μέθοδος απεικόνισης της μουσικής, συμβατικά η μουσική σημειογραφία, μπορούσε να είναι είτε είσοδος είτε έξοδος κάποιου υπάρχοντος συστήματος. Από τότε, και οι τέσσερις πτυχές του προβλήματος – ο ήχος ως είσοδος, ο ήχος ως έξοδος, η (τυπωμένη) μουσική σημειογραφία ως είσοδος (optical music recognition/music scanning), η (ψηφιακή) μουσική σημειογραφία ως έξοδος – έχουν μελετηθεί με διαφορετικό βαθμό επιτυχίας η κάθε μία [Byrd84].

Η αυτόματη καταγραφή της μουσικής είναι σε μεγάλο βαθμό συγκρίσιμη με την αυτόματη αναγνώριση φωνής, αν και η αναγνώριση φωνής έχει σίγουρα προσελκύσει, τόσο ακαδημαϊκά όσο και εμπορικά, πολύ περισσότερο ενδιαφέρον κι έτσι έχει μελετηθεί περισσότερο. Χαρακτηριστικό γνώρισμα και των δύο αυτών εφαρμογών αποτελεί το γεγονός ότι εφαρμόζονται σε δεδομένα του πραγματικού κόσμου, δηλαδή που παράγονται φυσικά. Έτσι είναι περιορισμένος ο αριθμός των ξεχωριστών γεγονότων που μπορούν να ορίσουν δομές. Στο λόγο, τα φωνήματα<sup>1</sup> χρησιμοποιούνται για τη δημιουργία λέξεων και προτάσεων και αντίστοιχα, στη μουσική, μεμονωμένοι ήχοι συνδυάζονται για τη δημιουργία μελωδιών, ρυθμών και τραγουδιών. Μιά βασική διαφορά των δύο, ωστόσο, είναι ότι η φωνή είναι ουσιαστικά μονοφωνική (ένας ομιλητής), ενώ η μουσική συχνά είναι πολυφωνική. Από την άλλη, ο λόγος διαφοροποιείται ταχύτερα και ταυτόχρονα οι ιδιότητες του ήχου που μεταφέρει πληροφορία φωνής είναι εγγενώς πολυδιάστατες, ενώ το pitch και η διάρκεια ενός (μονοφωνικού) μουσικού σήματος είναι μονοδιάστατες ποσότητες (§1.2.5 [KD06]).

---

<sup>1</sup> Τα **φωνήματα** (phonemes) αποτελούν τα ελάχιστα στοιχεία/μονάδες μιάς γλώσσας, που παρέχουν διακριτική/διαφοροποιητική λειτουργία στο φωνητικό επίπεδο, για το νόημα του γλωσσικού ήχου.

Η βασική δυσκολία κατά την ανάπτυξη συστημάτων φωνής είναι η δυναμική ποικιλομορφία των σημάτων φωνής διαφορετικών γλωσσολογικών και ακουστικών περιεχομένων, ακόμη και στην περίπτωση ενός ομιλητή. Για την επαρκή μοντελοποίηση αυτής της ευρύτητας, χρειάζεται η σχολαστική οργάνωση μεγάλων βάσεων πραγματικών δεδομένων φωνής και η αξιοποίησή τους στην εκπαίδευση στατιστικών μοντέλων τα οποία αναπαριστούν τα ακουστικά χαρακτηριστικά των φωνημάτων και λέξεων.

Στη μουσική καταγραφή, οι κυρίαρχες δυσκολίες πηγάζουν από τη *Συνδυαστική (Combinatorics)*. Ήχοι από διαφορετικά μουσικά όργανα εμφανίζονται σε ποικίλους συνδυασμούς και δημιουργούν τις μουσικές φράσεις και τα μουσικά κομμάτια. Από την άλλη, η δυναμική διαφοροποίηση και πολυπλοκότητα ενός μεμονωμένου μουσικού ήχου δεν είναι τέτοια όπως των ήχων φωνής και αυτό συνεπάγεται την πιο εύκολη συλλογή δεδομένων αλλά και τη χρήση ακόμη και συνθετικών δεδομένων εκπαίδευσης. Ακουστικές μετρήσεις μεμονωμένων μουσικών ήχων που στη συνέχεια μπορούν να συνδυαστούν και να εμπλουτιστούν με ηχητικά εφέ μας παρέχουν μία ακριβή αναφορά του συνόλου υπό εξέταση.

Ωστόσο, οι τεχνολογίες που έχουν αναπτυχθεί στον τομέα της αναγνώρισης φωνής και ομιλητή μπορούν να προσφέρουν μία πλούσια βιβλιογραφία και φυσικά μία ποικιλία τεχνικών στην κατεύθυνση της έρευνας για την αυτόματη αναγνώριση μουσικής και μουσικών οργάνων. Το αντικείμενο της αναγνώρισης ήχου αποτελεί τομή της Ψηφιακής Επεξεργασίας Σήματος (Digital Signal Processing) και της Μηχανικής Μάθησης (Machine Learning). Η Ψηφιακή Επεξεργασία Σήματος στοχεύει στην εξαγωγή γνώσης από το ακουστικό σήμα, δηλαδή στη συλλογή μαθηματικών ποσοτήτων οι οποίες μπορούν να περιγράψουν με αποτελεσματικό τρόπο για τη δεδομένη περίπτωση το ίδιο το σήμα. Η Μηχανική Μάθηση είναι μία σύγχρονη επιστήμη που πραγματεύεται τον τρόπο με τον οποίο τα υπάρχοντα δεδομένα (δεδομένα εκπαίδευσης ή μάθησης) τα οποία επιλύουν το υπό εξέταση πρόβλημα μπορούν να χρησιμοποιηθούν ώστε να μας επιτρέπουν, μελλοντικά, την επίλυση του ίδιου προβλήματος για διαφορετικά αλλά της ίδια φύσης δεδομένα.

Ως αυτόματη καταγραφή της μουσικής ορίζεται η διαδικασία της ανάλυσης ενός μουσικού σήματος η οποία οδηγεί στην καταγραφή των παραμέτρων της μουσικής που λαμβάνουν χώρα σε αυτό. Τέτοιες παράμετροι είναι ο *τόνος (pitch)*, η *χρονική στιγμή έναρξης (onset time)*, η *διάρκεια (duration)* και η *πηγή (source)* κάθε ήχου που εμφανίζεται στο ακουστικό σήμα. Η καταγραφή παίρνει τη μορφή της συμβατικής μουσικής σημειογραφίας ή οποιασδήποτε συμβολικής αναπαράστασης η οποία παρέχει τις απαραίτητες πληροφορίες για την εκτέλεση ενός μουσικού κομματιού. Στην περίπτωση της Δυτικής ή Ευρωπαϊκής μουσικής με την οποία θα ασχοληθούμε στην παρούσα μελέτη οι *μουσικές νότες* αποτελούν τα σύμβολα μέσω των οποίων καθορίζονται οι παράμετροι που αναφέρθηκαν.

Συμβατικά, η εργασία της μουσικής καταγραφής στο πεντάγραμμο γίνεται από εκπαιδευμένους μουσικούς οι οποίοι ακούνε το μουσικό κομμάτι και στη συνέχεια καταγράφουν οι ίδιοι τις κατάλληλες νότες ή συγχορδίες, κάτι που απαιτεί χρόνο και προχωρημένες μουσικές γνώσεις (σχετικό μάθημα μουσικής το οποίο εξετάζει την ικανότητα αυτή της καταγραφής αποτελεί το **dictée**). Μία αυτοματοποιημένη υπολογιστική καταγραφή επιτρέπει αρκετές λειτουργίες. Καταρχήν, παρέχει έναν εύκολο τρόπο για την απόκτηση της μουσικής περιγραφής μίας ηχογράφησης και γνώσεις για την κατάλληλη κωδικοποίησή της. Ακόμη, η παραχθείσα μουσική καταγραφή επιτρέπει την αναπαραγωγή και εκτέλεση του μουσικού κομματιού, την μουσικολογική ανάλυσή του, την εξαγωγή μουσικής πληροφορίας (MIR: Music Information Retrieval) από μεγάλες βάσεις δεδομένων μουσικής ή την επεξεργασία του ακουστικού σήματος αναφορικά με το μουσικό περιεχόμενο. Η διαδικασία της αυτόματης καταγραφής θα μπορούσε τέλος να ενσωματωθεί σε Διαδραστικά μουσικά συστήματα ή συστήματα απομακρυσμένης εκμάθησης (Distance Learning Environment), επιτρέποντας τη διαμοίραση μεταξύ των χρηστών αρχείων παρτιτούρας τα οποία έχουν παραχθεί με αυτόματη καταγραφή. Με τη διαδικασία αυτή λοιπόν, οι καθηγητές μουσικής ή οι χρήστες λογισμικών μουσικής σημειογραφίας και άλλων σχετικών εφαρμογών θα μπορούν εύκολα και σε σαφώς λιγότερο χρόνο να εισάγουν κάποιο μουσικό κομμάτι και να προχωρούν σε περαιτέρω επεξεργασία (για παράδειγμα σε διορθώσεις ή συμπληρώσεις) και χρήση αυτού ανάλογα με τις δυνατότητες του εργαλείου το οποίο χρησιμοποιούν (για παράδειγμα θα μπορούν να αποστείλουν το αρχείο στον μαθητή προς μελέτη). Συμπερασματικά, η αυτόματη δημιουργία της μουσικής σημειογραφίας για ένα δοθέν ακουστικό σήμα θα διευκόλυνε σημαντικά το έργο της εκμάθησης μουσικής και θα διεύρυνε τις δυνατότητες των χρηστών λογισμικών μουσικής.

## 1.2 Αντικείμενο Διπλωματικής

Η παρούσα εργασία μελετά τη διαδικασία μέσω της οποίας μπορούμε να εξάγουμε την παρτιτούρα της οποίας η εκτέλεση συμφωνεί με την υπό εξέταση ηχογράφιση από μονοφωνικό πνευστό μουσικό όργανο, στην περίπτωση μας το κλαρινέτο.

Η παρτιτούρα, περιγράφει τη μουσική με έναν γραφικό τρόπο κάνοντας χρήση συγκεκριμένων συμβόλων. Παρέχει λοιπόν τη δυνατότητα σε όποιον έχει πρόσβαση σε αυτή να εκτελέσει (ο ίδιος) ή να αναπαράγει (με χρήση υπολογιστή) το μουσικό κομμάτι που αναπαρίσταται. Παραδοσιακά, η μουσική σημειογραφία περιλαμβάνει τις νότες και τις διάρκειες αυτών, και άλλες πληροφορίες σχετικές με το ύφος της εκτέλεσης και την ερμηνεία του οργάνου για το οποίο προορίζεται. Κατά την καταγραφή της μουσικής λοιπόν θα πρέπει να αναγνωριστεί η νότα (ή οι νότες στην περίπτωση της πολυφωνικής μουσικής) που αναπαράγεται κάθε στιγμή (pitch), η αρχή της (onset time), η χρονική της διάρκεια και το μουσικό όργανο από το οποίο προήλθε. Συμπληρωματικά, θα ήταν χρήσιμη η αναγνώριση κάποιων χαρακτηριστικών που έχουν να κάνουν με την αρμονία<sup>2</sup> της μουσικής, όπως για παράδειγμα η κλίμακα στην οποία είναι γραμμένο το μουσικό κομμάτι, και άλλα χαρακτηριστικά που περιγράφουν το είδος στο οποίο αυτό ανήκει και συγκεκριμένες οδηγίες που αφορούν το εκάστοτε μουσικό όργανο. Ο στόχος της αυτόματης μουσικής καταγραφής είναι να εξάγει αυτές τις παραμέτρους δοθέντος μόνο ενός αρχείου ηχογράφησης μουσικής.

Όπως γίνεται σαφές από τα παραπάνω, η αυτόματη καταγραφή της μουσικής, δηλαδή η σύλληψη μιάς βασικής ιδέας και ο σχεδιασμός ενός συστήματος γενικής εφαρμογής, αποτελεί ένα περίπλοκο αντικείμενο έρευνας καθώς εμπεριέχει ένα σύνολο από ξεχωριστά θέματα εργασιών.

Η παρούσα εργασία πραγματοποιήθηκε σε συνεργασία με το *Ινστιτούτο Επεξεργασίας του Λόγου* του Ερευνητικού κέντρου *Αθηνά* - <http://www.ilsp.gr/> - με αφορμή το έργο *VEMUS (Virtual European Music School)* - <http://www.vemus.org/> - το οποίο αποτέλεσε ένα έργο έρευνας και ανάπτυξης που συγχρηματοδοτήθηκε από την Ευρωπαϊκή Επιτροπή στο πλαίσιο του Προγράμματος των Τεχνολογιών της Κοινωνίας της Πληροφορίας του Έκτου Προγράμματος-Πλαισίου.

Συγκεκριμένα, το VEMUS παρέχει ένα πλαίσιο μουσικής διδασκαλίας για δημοφιλή μουσικά όργανα, όπως το φλάουτο, το κλαρινέτο, το σαξόφωνο και η φλογέρα, το οποίο είναι ανοικτό, διαδικτυακό και πολύγλωσσο προσφέροντας υψηλή διαδραστικότητα καθώς και ένα σύνολο από καινοτόμα εργαλεία ηλεκτρονικής μάθησης, μιά συνολική παρουσίαση του έργου γίνεται στο Παράρτημα Α.

Η αυτόματη καταγραφή των παραμέτρων της μουσικής που λαμβάνουν χώρα σε ένα ακουστικό σήμα και η αναπαράστασή τους ως μιά ψηφιακή παρτιτούρα θα επιτρέπει στους δασκάλους χρήστες της εφαρμογής αφού εκτελέσουν και ηχογραφήσουν ένα μουσικό κομμάτι (ηχογράφιση αναφοράς) να μπορούν να εισάγουν άμεσα την παρτιτούρα που θα μπορεί να τη συνοδεύει. Διευκολύνεται έτσι η διαδικασία της εισαγωγής νέων μουσικών κομματιών και τα δυο αρχεία (ηχογράφιση αναφοράς, παραχθείσα παρτιτούρα) θα μπορούν στη συνέχεια να διατεθούν στους μαθητές.

Αντλώντας αρχικά δεδομένα από το Vemus και χρησιμοποιώντας τελικά βασικές λειτουργίες του, η εργασία αυτή θα μελετήσει τα χαρακτηριστικά των ήχων που μπορούν να αναπαραχθούν από το κλαρινέτο μέσα από μιά ποικιλία ηχογραφήσεων, με στόχο την υλοποίηση ενός συστήματος το οποίο θα επιτρέπει την αυτόματη δημιουργία μιάς παρτιτούρας. Η παρτιτούρα αυτή θα παρέχει τις απαραίτητες για τη μουσική εκτέλεση πληροφορίες ενός μουσικού κομματιού το οποίο δίνεται με τη μορφή ηχογράφησης. Το κλαρινέτο επιλέχθηκε ως ένα τυπικό παράδειγμα μονοφωνικού πνευστού μουσικού οργάνου. Η εργασία βασίζεται στην *αυτόματη καταγραφή μονοφωνικών μουσικών σημάτων*. Το σύστημα το οποίο σχεδιάστηκε και υλοποιήθηκε μπορεί στη συνέχεια να γενικευτεί και σε άλλα μονοφωνικά μουσικά όργανα.

<sup>2</sup> Στην μουσική, με τον όρο **αρμονία** αναφερόμαστε στην ύπαρξη ταυτόχρονων pitch ή συγχορδιών και στους κανόνες που τα διέπουν. Η αρμονία περιγράφει την "κάθετη" ανάλυση της μουσικής και διαφοροποιείται από την μελωδική γραμμή, την "οριζόντια" ανάλυση.

### ***1.3 Οργάνωση Κειμένου***

Στο Κεφάλαιο 2 που ακολουθεί αναφερόμαστε στην ιστορία της σχέσης μουσικής – υπολογιστή και στις σχετικές εργασίες που έχουν λάβει χώρα. Στο Κεφάλαιο 3 παρουσιάζονται οι μεθοδολογίες, τα εργαλεία και οι τεχνικές τις οποίες η εργασία θα χρησιμοποιήσει. Το Κεφάλαιο 4 περιγράφει αναλυτικά τα βήματα που πραγματοποιήθηκαν κατά το σχεδιασμό του συστήματος αυτόματης καταγραφής της μουσικής. Περιλαμβάνει τα στάδια που προηγήθηκαν του βασικού συστήματος και τα οποία παρέχουν τις απαραίτητες γνώσεις για τη λειτουργία της διαδικασίας της αυτόματης καταγραφής, την ιδέα η οποία υλοποιήθηκε, καθώς και τον έλεγχο και την αξιολόγηση των αποτελεσμάτων. Τέλος, στο Κεφάλαιο 5 γίνεται μιά συνολική αξιολόγηση της εργασίας και ενθαρρύνεται περαιτέρω έρευνα σχετική με το αντικείμενο.



# 2

## *Ο Υπολογιστής ως Μουσικό Εργαλείο*

### *2.1 Διαφορετικές Προσεγγίσεις της Μουσικής Επεξεργασίας στον*

#### *Πραγματικό Κόσμο*

Η επεξεργασία της μουσικής πληροφορίας με χρήση ψηφιακών μέσων έχει αναπτυχθεί ιδιαίτερα τα τελευταία χρόνια. Φυσικά, τεχνολογίες προσανατολισμένες προς τους μουσικούς μελετούνται εδώ και χρόνια, συμπεριλαμβανομένων της σύνθεσης ήχου με χρήση synthesizers, της μουσικής παραγωγής με χρήση προγραμμάτων μουσικής εγγραφής (MIDI sequencers) και διαφόρων άλλων ειδών υποστήριξης για τη σύνθεση, εκτέλεση και καταγραφή της μουσικής. Τα εργαλεία αυτά έχουν ήδη γίνει ένα αναπόσπαστο κομμάτι της διαδικασίας της μουσικής παραγωγής. Πιο πρόσφατα, όμως, η προσοχή έχει στραφεί από τα συμβατικά αυτά εργαλεία προς νέες τεχνολογίες που στοχεύουν στην άμεση απόλαυση της μουσικής από τους τελικούς χρήστες, οι οποίοι δεν είναι μουσικοί, και αν και έχουν μάλλον ελάχιστη επιθυμία για μουσική δημιουργία, εκδηλώνουν ιδιαίτερο ενδιαφέρον για την απόκτηση της αγαπημένης τους μουσικής. Πρόσφατα θέματα έρευνας στο χώρο της μουσικής επεξεργασίας αντανακλούν αυτές τις απαιτήσεις των τελικών χρηστών. Ο στόχος της επεξεργασίας ξεφεύγοντας από τα όρια ενός μεμονωμένου μουσικού κομματιού επικεντρώνεται στο γενικότερο πλαίσιο του συνόλου μουσικών κομματιών.

Τυπικά παραδείγματα τεχνολογιών στην κατεύθυνση αυτή αποτελούν τεχνολογίες αναγνώρισης ομοιότητας μεταξύ κομματιών, μέθοδοι ανάκτησης και ταξινόμησης της μουσικής αναφορικά με το είδος της μουσικής που ομάδες χρηστών μιάς εφαρμογής προτιμούν και επιλογή μουσικών κομματιών σύμφωνα με αυτό. Το πεδίο αυτό της έρευνας ονομάζεται **Music Information Retrieval (MIR)** και αντικείμενό της αποτελούν μιά σειρά από θέματα που βασίζονται στην ανάκτηση πληροφοριών μετά από επεξεργασία του ακουστικού σήματος, σε αντίθεση με την αναζήτηση βασισμένη σε βιβλιογραφικές πληροφορίες (όπως τίτλοι και ονόματα καλλιτεχνών από μιά CDDb, μιά online βάση δεδομένων με πληροφορίες CD). Αναφορικά, τέτοιες μεθόδους αποτελούν η αναζήτηση βάσει μιάς μελωδίας (Research on Melody: Query by Humming – QBH), η αναζήτηση βάσει αποσπάσματος (Research on Music Fragments) και η αναζήτηση βάσει ομοιότητας μεταξύ μουσικών κομματιών (Research on Entire Musical Pieces) [GH04].

Τα τελευταία δεκαπέντε χρόνια σημαντική πρόοδο έχει σημειώσει η έρευνα σχετικά με την κατανόηση του μουσικού ακουστικού σήματος. Εκτός από την συνήθη μέθοδο του διαχωρισμού και εξαγωγής των ξεχωριστών οντοτήτων του ήχου που συνθέτουν ένα ακουστικό σήμα (**Sound Source Segregation**) και τη χρήση των πληροφοριών αυτών για την αυτόματη καταγραφή της μουσικής (**Automatic Transcription**), το 1997 [GM97], σε επανεξέταση του πως οι ίδιοι οι άνθρωποι αντιλαμβάνονται τη μουσική, μία νέα ερευνητική προσέγγιση προτάθηκε για τη μουσική αντίληψη, η οποία βασίζεται στην άποψη ότι οι ακροατές κατανοούν τη μουσική χωρίς το διαχωρισμό των ηχητικών πηγών και χωρίς τη νοητική αναπαράσταση των ηχητικών σημάτων με κάποιο συμβολικό τρόπο. Θέματα ερευνών που συμφωνούν με την παρούσα προσέγγιση είναι η εκτίμηση του ρυθμού (**Rhythm Tracking/Tempo and Beat Tracking**), η αναγνώριση της μελωδίας (**Melody Recognition**) και η ανάλυση της μουσικής δομής (**Music Scene**). [GH04]

Συνοψίζοντας, η σημαντική εξέλιξη στην κατανόηση των μουσικών σημάτων έχει ενισχυθεί σημαντικά με την πρόοδο του υλικού του υπολογιστή και τις τεχνικές επεξεργασίας σημάτων ήχου. Πριν από δεκαπέντε χρόνια ήταν ακόμα δύσκολος ο υπολογισμός του **Fast Fourier Transform (FFT)** σε πραγματικό χρόνο, αλλά σήμερα η εκτέλεσή του μπορεί να γίνει τόσο γρήγορα ώστε ο χρόνος που απαιτείται για τους υπολογισμούς μπορεί ουσιαστικά να αγνοηθεί. Αυτό το άλμα στις επιδόσεις της επεξεργασίας επέτρεψε στους ερευνητές να επινοήσουν υπολογιστικά εντατικές προσεγγίσεις που δεν μπορούσαν να χρησιμοποιηθούν κατά το παρελθόν, καθώς έχει προωθήσει επίσης τη χρήση ενός φάσματος **στατιστικών τεχνικών**. Για παράδειγμα, έχουν χρησιμοποιηθεί στην κατεύθυνση αυτή τεχνικές που βασίζονται σε πιθανοτικά μοντέλα, όπως το Μαρκοβιανό μοντέλο (Hidden Markov Model - HMM), και ποικίλες τεχνικές οι οποίες κάνουν χρήση των εκτιμητών Μέγιστης Πιθανότητας (maximum likelihood estimation) και Bayes. [GH04]

Στο ευρύ αυτό πεδίο ενδιαφέροντος έχουν πραγματοποιηθεί και κάποιες ακόμη διπλωματικές εργασίες, διαφορετικών προσεγγίσεων και σε συνεργασία με περισσότερους του ενός τομέα, γεγονός που ενισχύει την διεπιστημονικότητα του πεδίου της κατανόησης και προώθησης της μουσικής με χρήση ή μέσω των ηλεκτρονικών υπολογιστών. Σχετικές εργασίες, για παράδειγμα *Αυτόματη Αναγνώριση Μουσικών Οργάνων* [DT06H], *Εφαρμογές της Μουσικής Τεχνολογίας στην Μουσική Εκπαίδευση* [DT07S], *ΜΟΥΣΙΚΟΣΥΝΘΕΤΗΣ σύστημα ημιαυτόματης σύνθεσης Βυζαντινής μουσικής* [DT08L], *Εγκεφαλική Λειτουργία και Μουσική: Καταγραφή και Επεξεργασία Δεδομένων Ηλεκτροεγκεφαλογραφίας σε Αρμονικά και Δυσαρμονικά Ακουστικά Ερεθίσματα* [DT08I], μπορούν να αναζητηθούν στην βάση Διπλωματικών Εργασιών της σχολής καθώς παρέχουν σημαντικές γνώσεις για την κατανόηση των πρόσφατων τάσεων ([http://artemis.cslab.ntua.gr/el\\_thesis/](http://artemis.cslab.ntua.gr/el_thesis/)).

## 2.2 Αυτόματη Μουσική Καταγραφή και Μουσική Σημειογραφία από

### Ηλεκτρονικό Υπολογιστή

#### 2.2.1 Εισαγωγικές Έννοιες

Για την καλύτερη κατανόηση του θέματος της αυτόματης καταγραφής μουσικής, είναι σκόπιμο να δώσουμε έναν διευκρινιστικό συνοπτικό ορισμό σε κάποιες έννοιες.

**Αυτόματη καταγραφή μουσικής** ονομάζεται η διαδικασία κατά την οποία ένα πρόγραμμα δέχεται ως είσοδο μία ηχογράφηση και δίνει ως έξοδο πληροφορίες για το ποιές νότες και σε ποιά χρονική στιγμή έχουν εκτελεστεί.

**Ακουστική αναπαράσταση** της μουσικής μπορεί να αποτελέσει μία ψηφιακή ηχογράφηση, από την οποία οι συχνότητες οι οποίες ακούγονται μπορούν να οριστούν για την κάθε χρονική στιγμή. Κάτι τέτοιο μπορεί να πάρει την μορφή mp3, CD εγγραφής ή οποιαδήποτε άλλη αναπαράσταση των φυσικών ήχων.

**Μουσική σημειογραφία** ονομάζεται ένα οποιοδήποτε σύστημα το οποίο αναπαριστά την ακουστική αντίληψη της μουσικής με χρήση γραπτών συμβόλων. Η παρούσα εργασία αναφέρεται στην Ευρωπαϊκή μουσική και επαγωγικά ως σύστημα μουσικής σημειογραφίας θα αναφέρεται η σύμφωνη με τους κανόνες της μουσικής **παρτιτούρα**.

Η **μουσική αναπαράσταση ή καταγραφή** είναι μιά απεικόνιση αρκετά κοντά στην παρτιτούρα. Περιέχει συγκεκριμένες πληροφορίες σε επίπεδο μουσικής, για το ποιές νότες ακούγονται και σε ποιά χρονική στιγμή, αλλά δεν ορίζει με ακρίβεια το ακουστικό αποτέλεσμα. Η μουσική αναπαράσταση συνήθως παίρνει τη μορφή ενός Midi<sup>3</sup> αρχείου.

Η παρούσα εργασία ασχολείται με την *αυτόματη καταγραφή μουσικής* και την αναπαράστασή της ως *παρτιτούρα*. Ως αρχείο εισόδου του συστήματος αυτόματης καταγραφής χρησιμοποιείται ένα wav αρχείο ηχογράφησης μουσικής εκτέλεσης, το οποίο αποτελεί μιά ακατέργαστη και ασυμπιεστη μορφή *ακουστικής αναπαράστασης*.



**Εικόνα 1: Οπτική αναπαράσταση μουσικής πληροφορίας. Περιγραφή:** Η πρώτη εικόνα παρουσιάζει την καθιερωμένη μουσική σημειογραφία (παρτιτούρα) και κάτω από αυτή βρίσκεται μιά ακριβής αναπαράσταση της αρχής και του τέλους της νότας. Η πληροφορία αυτή μπορεί να μην είναι η πλέον κατάλληλη όταν πρόκειται να χρησιμοποιηθεί για τη μουσική ανάγνωση και εκτέλεση αλλά έχει ενδιαφέρον σε περιπτώσεις παρουσίας του ρυθμού περίπλοκων συνδυασμών από νότες. Στα δεξιά απεικονίζεται το φάσμα συχνοτήτων. Ανάλογα με τα προηγούμενα, και στην περίπτωση αυτή, ο οριζόντιος άξονας είναι ο άξονας των χρόνων ενώ οι γραμμές του ανώτερου επιπέδου αναπαριστούν νότες υψηλότερων συχνοτήτων. Κάπως απλοϊκά στη μουσική ορολογία χρησιμοποιείται ο όρος "ύψος" αντί του "pitch" ή "τόνου" μιάς νότας. [L!!06]

<sup>3</sup> Ο όρος **MIDI** προέρχεται από τα αρχικά των λέξεων *Musical Instrument Digital Interface* το οποίο σε ελληνική απόδοση σημαίνει *Ψηφιακή διασύνδεση Μουσικών Οργάνων* και αναφέρεται στον εξ' αποστάσεως έλεγχο και την επικοινωνία ανάμεσα σε ηλεκτρονικά μουσικά όργανα και άλλες συσκευές όπως ηλεκτρονικούς υπολογιστές με λογισμικό μουσικής εγγραφής (*Sequencers*), ρυθμομηχανές (*drum-machines*), δειγματολήπτες (*samplers*), συνθετητές με δυνατότητα μουσικής εγγραφής (*Workstation Synthesizers*), συσκευές συγχρονισμού και άλλα, ανεξαρτήτως κατασκευαστή.

## 2.2.2 Η Φυσική Διάσταση των Βασικών Εννοιών της Μουσικής

Αντιλαμβανόμαστε έναν ήχο παρατηρώντας στιγμιαίες διαφορές στην ένταση του αέρα γύρω από τα αυτιά μας. Είναι αυτές οι εναλλαγές της έντασης που αναπαράγονται από ένα μουσικό όργανο ή καταγράφονται από ένα μικρόφωνο. Τα πρότυπα που παράγονται από αυτές τις εντάσεις είναι ενδεικτικά των ιδιοτήτων που έχουν οι νότες: *τόνος (pitch)*, *δυναμική (dynamic)* και *ηχώχρωμα (timbre)*. Τα πρότυπα αυτά ταλαντεύονται σε ένα συγκεκριμένο πεδίο τιμών και έτσι συσχετίζονται άμεσα με κάποιο από τα χαρακτηριστικά της κυματομορφής του ήχου - το pitch με τη συχνότητα, η δυναμική με το πλάτος και το ηχώχρωμα με το φάσμα. Πιο αναλυτικά, [KD06]

Το **pitch** είναι μία ιδιότητα εύκολα αντιληπτή από τον άνθρωπο που επιτρέπει την κατηγοριοποίηση των ήχων σε μία κλίμακα συχνοτήτων μεταξύ χαμηλών και υψηλών τιμών. Ακριβέστερα, το pitch ορίζεται ως η συχνότητα μιάς ημιτονοειδούς κυματομορφής που ταιριάζει με τον ήχο που αντιλαμβανόμαστε [Hart96]. Η **θεμελιώδης συχνότητα (F0)** είναι η αντιστοιχούσα φυσική σημασία του όρου και ορίζεται μόνο για περιοδικά ή σχεδόν περιοδικά σήματα. Για αυτή την κατηγορία σημάτων, η θεμελιώδης συχνότητα ορίζεται ως το αντίστροφο της περιόδου και είναι στενά συνδεδεμένη με το pitch. Σε διφορούμενες καταστάσεις, η περίοδος που αντιστοιχεί στην **αντιλαμβανόμενη συχνότητα** είναι αυτή που επιλέγεται. Κάθε pitch μπορεί να προσδιοριστεί από την τιμή της θεμελιώδους συχνότητας (Hz) στην οποία αντιστοιχεί ή από ένα γράμμα (A-G) το οποίο προσδιορίζει το όνομα της νότας (συνοδευόμενο από το σύμβολο της dieses ή ύφεσης κατά την αλλοίωση της νότας) και έναν αριθμό (1-7) ο οποίος προσδιορίζει την οκτάβα στην οποία εμφανίζεται η νότα αυτή (για παράδειγμα ο συμβολισμός A#4 / B b 4 αντιπροσωπεύει τη νότα A# της 4ης οκτάβας η οποία ταυτίζεται με την B b της ίδιας οκτάβας και η θεωρητική τιμή της συχνότητας η οποία αντιστοιχεί στη νότα αυτή είναι ίση με 415Hz για το κλαρινέτο). Οι νότες οι οποίες σημειώνονται με το ίδιο γράμμα (το οποίο αντιστοιχεί στο όνομα της νότας) αλλά διαφορετικό αριθμό (ο οποίος αντιστοιχεί στην οκτάβα της νότας) ηχούν όμοια (για παράδειγμα C4 και C5). Το φαινόμενο αυτό εξηγείται στην παράγραφο 3.1.

Η αντίληψη της **δυναμικής ή ηχηρότητας**<sup>4</sup> ενός ακουστικού σήματος δεν συνδέεται με έναν αντίστοιχα απλό τρόπο με τις φυσικές ιδιότητές του, και τα σχετικά υπολογιστικά μοντέλα για την αντίληψή της αποτελούν βασικό πεδίο έρευνας της **ψυχοακουστικής**. Παρόλα αυτά, κατά την επεξεργασία της μουσικής είναι βολικό να εκφράζουμε την ηχηρότητα ενός ακουστικού σήματος ως τη μέση τετραγωνική τιμή του (ισχύς) εκφρασμένη σε λογαριθμική κλίμακα (decibel). Και αυτό διότι για παράδειγμα ένας ήχος γίνεται αντιληπτός από τους ανθρώπους ως δύο φορές πιο δυνατός όταν η έντασή του είναι δέκα φορές πιο μεγάλη. Η υποκειμενική αίσθηση της **έντασης** ενός ήχου λοιπόν συνδέεται με την ισχύ του ηχητικού σήματος που διεγείρει το αυτί μας. Σημειώνεται πως στην μουσική η δυναμική ή ένταση δεν αποτελεί χαρακτηριστικό γνώρισμα κάθε νότας ξεχωριστά, αλλά ορίζεται βάσει γενικότερων οδηγιών για την εκτέλεση.

Τέλος, το **ηχώχρωμα ή χροιά** μιάς νότας είναι στενά συνδεδεμένο με την πηγή από την οποία έχει παραχθεί ο εκάστοτε ήχος, χαρακτηριστικό γνώρισμα το οποίο μελετάται για το λόγο αυτό σε εργασίες **κατηγοριοποίησης των μουσικών οργάνων** [DT06H]. Δύο ήχοι που αναπαράγονται από διαφορετικά όργανα, αν και μπορεί να είναι πανομοιότυποι όσον αφορά τις τιμές pitch και δυναμικής, παραμένουν εύκολα διαχωρίσιμοι λόγω του διαφορετικού τους ηχοχρώματος. Ο λόγος δεν μπορεί να στηριχθεί σε κάποια ξεχωριστή ιδιότητα του ακουστικού σήματος αλλά εξαρτάται κυρίως από το πόσο 'τραχύ' είναι το ενεργειακό φάσμα του σήματος στο χρόνο και τη χρονική του συνάρτηση. Καθορίζεται ουσιαστικά από τη συμμετοχή των αρμονικών όρων πέραν της θεμελιώδους ή βασικής συχνότητας του ήχου και από τα χαρακτηριστικά κορυφώματα της περιβάλλουσας του φάσματος. Ενώ λοιπόν το pitch και η δυναμική ενός σήματος μπορούν με απλό τρόπο να αναπαρασταθούν σε δυο διαστάσεων κλίμακα, το ηχώχρωμα αποτελεί ουσιαστικά πολυδιάστατη παράμετρο, και αναπαρίσταται τυπικά από ένα χαρακτηριστικό διάνυσμα.

<sup>4</sup> Σε μία κάπως απλή προσέγγιση, από την πλευρά ενός εκπαιδευμένου μουσικού αυτιού και μόνο, θα λέγαμε πως η ένταση του ήχου και η ποιότητά του, ή ακριβέστερα το πόσο λαμπρός είναι ένας ήχος, είναι χαρακτηριστικά γνωρίσματα τα οποία αποτελούν σύνθετες έννοιες ικανές να περιγραφούν μέσα από την επίδραση εναλλαγών του pitch και της δυναμικής του ήχου. Αυτό σημαίνει ότι πρακτικά μία μείωση στο pitch με ταυτόχρονη αύξηση της δυναμικής θα γίνει αντιληπτή ως αύξηση της έντασης, όταν μία αύξηση του pitch με ταυτόχρονη αύξηση της δυναμικής θα μας δώσει έναν πιο λαμπρό ήχο.

### 2.2.3 Αυτόματη Καταγραφή Μουσικής - Σχετικές Εργασίες

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

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

Αντίθετα, κάποιο εμπορικό σύστημα **αυτόματης καταγραφής πολυφωνικής μουσικής** γενικής εφαρμογής δεν έχει ακόμα εκδοθεί. Σε ένα άρθρο [Klar04], ο Anssi Klauri, συζητά το πρόβλημα της καταγραφής μουσικής. Δηλώνει πως παρά τον σημαντικό αριθμό των προσπαθειών για την επίλυση του προβλήματος, ένα πρακτικά γενικού σκοπού εφαρμόσιμο σύστημα καταγραφής δεν έχει υπάρξει, και αυτό θα μπορούσε να υποστηριχθεί μέχρι και σήμερα. Ωστόσο, ερευνητές ανά τον κόσμο έχουν παρουσιάσει ποικίλες προσεγγίσεις για την καταγραφή συγκεκριμένων κατηγοριών ήχου της μουσικής.

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

- Εκτίμηση των θεμελιωδών συχνοτήτων μονοφωνικών και πολυφωνικών μουσικών ήχων
- Εκτίμηση της χρονικής διάρθρωσης των ήχων (του μουσικού ρυθμού)
- Αναγνώριση των μουσικών οργάνων
- Καταγραφή των κρουστών
- Αρμονική ανάλυση και καταγραφή των συγχορδιών
- Υψηλότερο επίπεδο μουσικολογικής δομής για την επίλυση διαφορούμενων καταστάσεων.

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

Μεγάλος αριθμός ερευνητών οι οποίοι επιχειρήσαν την αυτόματη καταγραφή πολυφωνικής μουσικής έχουν παρουσιαστεί τα τελευταία τουλάχιστον 30 χρόνια, μετά την δουλειά του Mougey τη δεκαετία του '70 [Moog75]. Τα διάφορα συστήματα ανωτάτου επιπέδου τεχνικής παρέμειναν χωρίς αμφιβολία φτωχά σε σύγκριση με τις δυνατότητες των ίδιων των μουσικών ως προς την ακρίβεια και την ευελιξία. Τα τελευταία 15 χρόνια ωστόσο έχει παρατηρηθεί μιά σημαντική πρόοδος.

Οι **διαφορετικές προσεγγίσεις της αυτόματης καταγραφής της μουσικής** περιλαμβάνουν, εκτός από το πολύ βασικό πεδίο έρευνας της **επίλυσης μείξεων ήχων μουσικών οργάνων τα οποία παράγουν τόνο/pitch** (σε αντιδιαστολή με τα μουσικά όργανα που παράγουν κρότο και ονομάζονται άτονα) (**polyphonic pitch recognition/detection**), το οποίο βασίζεται σε τεχνικές επεξεργασίας σήματος και προτείνει μεθόδους **εκτίμησης των ταυτόχρονων θεμελιωδών συχνοτήτων (F0s) ή διαχωρισμού των ήχων που συνυπάρχουν**, μελέτες αναφορικά με την **καταγραφή των κρουστών (percussion transcription)** και την **κατηγοριοποίηση των μουσικών οργάνων**. Μιά ακόμη προσέγγιση αποτελεί η ανάλυση του **μουσικού τοπίου ή δομής (music scene)**. Οι προτεινόμενες μέθοδοι στην περίπτωση αυτή στηρίζονται στην κατανόηση του μουσικού ακουστικού σήματος στο επίπεδο των μουσικά ανειδίκευτων ακροατών. Αντίθετα δηλαδή με την προσέγγιση η οποία βασίζεται σε τεχνικές επεξεργασίας σήματος, και στοχεύει στην ανάλυση και κατανόηση του ακουστικού σήματος στο επίπεδο των καταρτισμένων μουσικών ακροατών, είτε αναγνωρίζοντας το σύνολο από νότες που αποτελούν ένα μουσικό κομμάτι είτε διαχωρίζοντας τα επιμέρους σήματα που το συνθέτουν, η περιγραφή του μουσικού τοπίου έγκειται στην **αναγνώριση των διακριτικών γεγονότων που χαρακτηρίζουν μία μουσική εκτέλεση**, όπως η **μελωδική γραμμή (melodic line)**, τα **μπάσα (bass line)**<sup>5</sup>, ο **ρυθμός (rhythm ή tempo tracking)**, το **ρεφρέν** και η **επανάληψη φράσεων**, καθώς και η **ποιότητα του ήχου (ηχώχρωμα/timbre)** των μουσικών οργάνων [KD06].

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

Σε αυτό το σημείο είναι χρήσιμο να αναφερθούμε στην **ιστορία της αυτόματης μουσικής καταγραφής** που ακολούθησε μετά την εμφάνιση του Moorer και του συστήματος καταγραφής για δύο φωνές που υλοποίησε [Moor75], [Moor77], όπως παρουσιάζεται συνολικά στο πρώτο κεφάλαιο (§1.Introduction) του βιβλίου “Signal Processing Methods for Music Transcription” των Anssi Klauri και Manuel Davy (2006) [KD06].

“Οι Chafe et al. [Chafe85], Piszcalski [Pizsc86] και Maher [Maher89] συνέχισαν τη δουλειά του Moorer κατά τη δεκαετία του '80. Στα πρώιμα αυτά συστήματα, ο αριθμός των ταυτόχρονων φωνών ήταν περιορισμένος στις δύο και οι σχέσεις των ταυτόχρονων pitch περιοριστικές με διάφορους τρόπους. Στο κομμάτι της ανάλυσης του ρυθμού, ο πρώτος αλγόριθμος εκτίμησης του ρυθμού σε γενικού περιεχομένου ακουστικά σήματα προτάθηκε από τους Goto και Muraoka [GM94b] την επόμενη δεκαετία, του '90, παρότι είχε προηγηθεί αξιοσημείωτος όγκος δουλειάς στην εκτίμηση του ρυθμού παραμετρικών δεδομένων από νότες και ο αλγόριθμος του Schloss [Sch85] για την εκτίμηση του ρυθμού κρουστών οργάνων. Οι πρώτες απόπειρες καταγραφής των κρουστών οργάνων έγιναν στα μέσα του '80 από τον Schloss [Sch85] και αργότερα από τον Bilmes [Bilm93]. Και οι δυο κατηγοριοποιούσαν διαφορετικού τύπου “χτύπους” σε συνεχείς ηχογραφήσεις. Η καταγραφή πολυφωνικών κρουστών ήχων πραγματοποιήθηκε αργότερα από τους Goto και Muraoka [GM94]. Μιά αναλυτική παρουσίαση των πρώτων αυτών σταδίων της εξέλιξης της μουσικής καταγραφής έχει γίνει από τον Tanguiane [Tangui93, pp.3-6].

<sup>5</sup> Ο όρος **bass line** χρησιμοποιείται για να περιγράψει το μέρος των μουσικών εκτελέσεων της ορχήστρας στα χαμηλά pitch, το οποίο αναλαμβάνει το ρυθμικό συνοδευτικό ρόλο της μουσικής και εκτελείται συνήθως από το ηλεκτρικό μπάσο (ηλεκτρικό τετράχορδο σε σχήμα κιθάρας), το κοντραμπάσο (οικογένεια βιολιού) ή από κάποιο πληκτροφόρο όργανο.

Από την αρχή της δεκαετίας του '90, το ενδιαφέρον για την μουσική καταγραφή έχει μεγαλώσει ταχύτατα και δεν είναι εφικτό να γίνει ο συνολικός απολογισμός της δουλειάς στα πλαίσια αυτής της εργασίας. Για το λόγο αυτό, θα αναφερθούμε μόνο σε συγκεκριμένες τάσεις και επιτυχημένες προσεγγίσεις που έχουν σημειωθεί. Μιά από αυτές είναι η αξιοποίηση **στατιστικών μεθόδων (statistical methods)**. Σημαντικά παραδείγματα χρήσης στατιστικών μεθόδων στην ανάλυση των πολλαπλών pitch της πολυφωνικής μουσικής αποτελούν οι μέθοδοι που προτάθηκαν από τους Kashimo et al [Kal95], Goto [Goto01], Davy και Godsill [DG02], και Ryyanen και Klapuri [RK05]. Στο πεδίο της εκτίμησης του ρυθμού, στατιστικές μέθοδοι εφαρμόστηκαν από τους Cemgil και Karpen [CK03], Hainsworth και MacLeod [HM04], και Klapuri et al. [Kal06], ενώ στη μελέτη της καταγραφής κρουστών οργάνων από τους Gillet και Richard [GR04] και Paulus και Klapuri [PK03]. Η χρήση όμως στατιστικών μεθόδων αναγνώρισης προτύπων επικράτησε στο πεδίο της κατηγοριοποίησης των μουσικών οργάνων [HPD03].

Μιά ακόμη τάση αποτέλεσε η χρησιμοποίηση **υπολογιστικών μοντέλων του ανθρώπινου ακουστικού συστήματος (computational models of the human auditory system)**. Τέτοιες τεχνικές για την μουσική καταγραφή παρουσιάστηκαν από τον Martin [Mar96], και οδήγησαν στην πρόταση σχετικών μεθόδων για την ανάλυση των pitch πολυφωνικών ήχων από τους Karjalainen και Tolonen [KT00] και Klapuri [Klap05], και μεθόδων για την εκτίμηση του ρυθμού από τον Scheiner [Sche98], για παράδειγμα.

Μιά σπουδαία προσέγγιση ήταν αυτή της μοντελοποίησης του **μουσικού τοπίου (auditory scene analysis - asa)**, η οποία μελετά τα όσα γίνονται εύκολα αντιληπτά από κάθε άνθρωπο. Οι αρχές του asa εφαρμόστηκαν στην ανάλυση των pitch πολυφωνικών μουσικών σημάτων από τους Mellinger [Mel91] και Kashimo και Tanaka [KT93], και αργότερα από τους Godsmark και Brown [GB99] και τους Sterian et al. [SSW99]. Πιο πρόσφατα, κάποιες μέθοδοι **μάθησης χωρίς επίβλεψη (unsupervised learning)** έχουν προταθεί κατά τις οποίες ένας ελάχιστος αριθμός από πρωταρχικές υποθέσεις λαμβάνονται για το υπό ανάλυση σήμα.

Μέθοδοι βασισμένες στην **ανάλυση ανεξάρτητων συνιστωσών (independent component analysis)** προτάθηκαν από τον Casey [Cas98], [CW00] και άλλες διαφορετικές μέθοδοι παρουσιάστηκαν αργότερα από τους Lepain [Lep99], Smaragdis [Sma01], [SB03], Abdallh [Abd02], [AP04], Virtanen (§9. Unsupervised Learning Methods for Source Separation in Monaural Music Signals), FitzGerald [Fitz04], [FCL03] και Paulus και Virtanen [PV05].”

Εκτός από την αναφορά στις σημαντικότερες τάσεις και εργασίες που έχουν παρουσιαστεί κρίνεται εξίσου απαραίτητη η παρουσίαση της συγκριτικής **αξιολόγησης** αυτών όπως παρουσιάζεται στη συνέχεια του ίδιου κεφαλαίου του βιβλίου “Signal Processing Methods for Music Transcription” των Anssi Klapuri και Manuel Davy (2006) [KD06].

“ Ένα πρακτικά γενικού σκοπού εφαρμόσιμο σύστημα καταγραφής δεν έχει ακόμη υπάρξει. Παρόλα αυτά έχει επιτευχθεί μιά κάποια επιτυχία στην καταγραφή πολυφωνικής μουσικής θέτοντας περιορισμούς στην πολυπλοκότητα. Στην καταγραφή οργάνων τα οποία παράγουν τόνους (pitch), τυπικό περιορισμό αποτελεί ο μέγιστος αριθμός των ταυτόχρονων ήχων [KT00], [DG02], η παρεμβολή ντράμς και άλλων κρουστών δεν επιτρέπεται [KNS04], ή μόνο ένα συγκεκριμένο όργανο λαμβάνεται υπόψη [Mar04]. Κάποια ελπιδοφόρα αποτελέσματα στην καταγραφή μουσικής του πραγματικού κόσμου από ηχογραφήσεις CD έχουν να επιδείξουν οι Goto [Goto01] και οι Ryunanen και Klaruri [RK05]. Στην καταγραφή των κρουστών έχει επιτευχθεί αρκετά καλή ακρίβεια η οποία περιλαμβάνει περιορισμένο αριθμό μουσικών οργάνων και άτονα μουσικά όργανα (μουσικά όργανα τα οποία δεν παράγουν τόνο/pitch αλλά κρότο) [GR04], [PV05]. Επίσης υποσχόμενα αποτελέσματα έχουν παρουσιαστεί στην καταγραφή των μπάσων και του τυμπάνου σε πραγματικού κόσμου ηχογραφήσεις, αλλά αυτό αποτελεί ένα ακόμη ανοιχτό πρόβλημα (Zils et al. [Zal02], FitzGerald et al. [Fal03], Yoshii et al. [Yal04]). Η εκτίμηση του ρυθμού πολύπλοκων ακουστικών σημάτων μπορεί να γίνει με αρκετή αξιοπιστία με τις δεδομένες ανωτάτου επιπέδου τεχνικές, αλλά οι δυσκολίες παραμένουν ιδιαίτερα κατά την ανάλυση της κλασικής μουσικής και ρυθμικά περίπλοκου υλικού. Μιά συγκριτική αξιολόγηση των εργασιών εκτίμησης του ρυθμού παρουσιάζεται στα [HM04], [Kal05] και [KEA06]. Οι έρευνες αναφορικά με την κατηγοριοποίηση των μουσικών οργάνων έχουν επικεντρωθεί στο επίπεδο των μεμονωμένων ήχων, παρότι πιο πρόσφατα επιχειρούνται και σε πολυφωνικά ακουστικά σήματα [KM99], [BE01], [EB04], [VR04]. ”

Το βασικότερο ίσως συμπέρασμα που προκύπτει από τα όσα έχουν αναφερθεί είναι πως στην αναζήτηση της δημιουργίας ενός ολοκληρωμένου και αποτελεσματικού συστήματος προς όφελος της μουσικής επιστήμης και τέχνης, απαιτείται συνδυαστική γνώση από περισσότερα του ενός πεδία έρευνας - επεξεργασία σήματος, ακουστική, επιστήμη των υπολογιστών, μουσική, γλωσσολογία και πειραματική ψυχολογία [Klar04T].

#### **2.2.4 Εκμάθηση Μουσικής με Χρήση Υπολογιστή - Ψηφιακή Μουσική Σημειογραφία**

Αμέτρητα είναι τα στοιχεία που επιβεβαιώνουν την ποσοτική και ποιοτική αλλαγή που έχει επιφέρει η μουσική τεχνολογία στην παραγωγή και την κατανάλωση της μουσικής. Η παρουσία λογισμικών μουσικής σημειογραφίας είναι πια τόσο σημαντική που η σκέψη της υποβολής μιάς χειρόγραφης παρτιτούρας προς δημοσίευση φαντάζει απαρχαιωμένη. Με την εξέλιξη αυτή προκύπτει εύλογα το ερώτημα από πλευράς των διδασκόντων μουσικής (των δεκτικών στις νέες τεχνολογίες), οι οποίοι έχουν την ευθύνη της προετοιμασίας των μαθητών για έναν διαφορετικό κόσμο μουσικής εμπειρίας, σχετικά με το ποιος είναι τελικά ο βαθμός στον οποίο η εμπειρία αυτή έχει ήδη ή πρόκειται να αλλάξει. Γίνεται ξεκάθαρη σιγά σιγά η αναγκαιότητα της ενσωμάτωσης της χρήσης ηλεκτρονικών υπολογιστών στην εκπαιδευτική διαδικασία προς διευκόλυνση και ενίσχυση του έργου τόσο των δασκάλων μουσικής (μεγαλύτερος έλεγχος και παρακολούθηση προόδου, περισσότερες δυνατότητες) όσο και των μαθητευόμενων (για παράδειγμα εξ' αποστάσεως διδασκαλία).

Σημαντικό ρόλο στην κατεύθυνση αυτή κατέχει η σημειογραφία της κοινής Ευρωπαϊκής μουσικής στην οθόνη ενός Υπολογιστή. Πέρα από τη δημοσίευση ολόκληρων μουσικών έργων ή μερών παρτιτούρας στο Διαδίκτυο, η μουσική σημειογραφία έχει κωδικοποιηθεί στον υπολογιστή με διάφορους τρόπους τα τελευταία 40 χρόνια. Το βιβλίο *Beyond Midi: The handbook of Musical Codes (1997)* περιγράφει περισσότερους από 20 διαφορετικούς τρόπους κωδικοποίησης. Υπάρχουν βέβαια και αρκετοί μη δημοσιευμένοι ιδιόκτητοι κώδικες.



Κάνοντας μία ανασκόπηση, παρατηρούμε πως η αγορά της διάθεσης παρτιτούρας στο Διαδίκτυο είχε ως εμπόδιο την εξάρτησή της από δυαδικά (binary format) ιδιόκτητα αρχεία. Το πλέον σύνθηες μέσο κωδικοποίησης, το Portable Document Format (PDF), δεν περιέχει κάποια μουσική σημασιολογία και δίνει τη δυνατότητα απλής ανάγνωσης και εκτύπωσης της παρτιτούρας. Εταιρίες όπως οι Sunhawk, MusicNotes, Sibelius και Noteheads, διέθεταν όλες διαφορετική ιδιόκτητη μουσική κωδικοποίηση για τους φορείς μουσικής σημειογραφίας τους. Η αγορά υλικού από αυτές ή συνεργάτες τους επέτρεπε επομένως την εκτέλεση, ανάγνωση ή εκτύπωση του μόνο από το συγκεκριμένο μουσικό πρόγραμμα. Αυτό επιτρέπει ελάχιστες δυνατότητες στον καταναλωτή-χρήστη συγκριτικά με την συμβατική τυπωμένη παρτιτούρα. Από το **2001** αυτό έχει αλλάξει χάρη στην εμφάνιση του προτύπου **MusicXml** από την **Recordare LLC** [Good01], το οποίο θα αποτελέσει μία φιλική προς το χρήστη του Διαδικτύου κωδικοποίηση της μουσικής σημειογραφίας και παρτιτούρας. Δε θα αναφερθούμε σε περισσότερες τεχνικές λεπτομέρειες της τεχνολογίας αυτής καθώς γίνεται μία ολοκληρωμένη παρουσίαση στην σχετική ενότητα (§3.3).

Η ηλεκτρονική μουσική, για παράδειγμα η χρήση synthesizer, αντιμετώπιζε παρόμοιο πρόβλημα τη δεκαετία του '80, καθώς δεν ήταν ακόμη δυνατή η σύμφωνη εκτέλεση και μείξη μουσικών οργάνων διαφορετικών προμηθευτών. Η παρουσίαση του πρωτοκόλλου **Musical Instrument Digital Interface (MIDI, 1996)** έλυσε αυτό το πρόβλημα και οδήγησε στην σημαντική ανάπτυξη της αγοράς των ηλεκτρικών μουσικών οργάνων, επιτρέποντας ταυτόχρονα την ανάπτυξη στην συμβατότητα και συναλλαγή δεδομένων γεγονός που οδήγησε στην επέκταση της μουσικής βιομηχανίας. Σήμερα, το διασυνδεδετικό Midi παραμένει η μοναδική μορφή συμβολικής επικοινωνίας μουσικής με ευρεία χρηστική εφαρμογή. Αλλά, το πρωτόκολλο Midi έχει σχεδιαστεί για την επίλυση προβλημάτων αναφορικά με την αναπαραγωγή μιάς εκτέλεσης από τον υπολογιστή, και δεν περιέχει αρκετή πληροφορία σχετικά με την μουσικολογική περιγραφή πίσω από το ακουστικό αποτέλεσμα, δηλαδή δεν περιλαμβάνει πληροφορίες οι οποίες μπορούν να οργανωθούν και να μελετηθούν από τους μουσικούς. Το MusicXml επιχειρεί να προσφέρει στην διαδικτυακή παρουσία της παρτιτούρας και τα διάφορα λογισμικά μουσικής ο,τι το Midi αποτέλεσε για την ηλεκτρονική μουσική βιομηχανία. Το πρότυπο MusicXml αποτελεί ένα εύχρηστο και φιλικό στο χρήστη του Διαδικτύου μέσο ανταλλαγής για εφαρμογές μουσικής σημειογραφίας, μουσικολογικής ανάλυσης, εξαγωγής μουσικής πληροφορίας και φυσικά μουσικής εκτέλεσης. Οι ίδιοι λόγοι που καθιστούν την xml<sup>6</sup> κωδικοποίηση ενδιαφέρουσα - διαδικτυακή χρηστικότητα, ευκολία σύνταξης του αρχείου, φιλική προς το χρήστη και ευανάγνωστη μορφή - ισχύουν αντίστοιχα και για την MusicXml κωδικοποίηση.

Ένας περιοριστικός παράγοντας στην ανάπτυξη λογισμικού σχετικού με τη μουσική ήταν η στενή σχέση της μουσικής κωδικοποίησης με τα εργαλεία ανάπτυξης. Για παράδειγμα, το μουσικό πρόγραμμα Finale απαιτεί C/C++ προγραμματιστικές δεξιότητες και το Humdrum toolkit προαπαιτεί μία εξοικείωση με το Unix. Η στενή λοιπόν σύζευξη του προγραμματιστικού περιβάλλοντος με την αναπαράσταση των δεδομένων περιόριζε σημαντικά τις δυνατότητες και την παραγωγικότητα των υπεύθυνων ανάπτυξης λογισμικού μουσικής.

Η ευελιξία και ευρύτητα των διαθέσιμων xml εργαλείων, επιτρέπουν στους προγραμματιστές που χρησιμοποιούν το πρότυπο MusicXml να επιλέξουν ανάμεσα σε μία σαφώς μεγαλύτερη ποικιλία εργαλείων ανάπτυξης. Ο ίδιος ο **Michael Good**, ιδρυτής της Recordare και της τεχνολογίας MusicXml, συνοδεύει αυτό τον ισχυρισμό με κάποια δείγματα προγραμματιστικών εφαρμογών που συντάχθηκαν μόλις σε μία μέρα με χρήση των Visual Basic, ActiveX controls και MusicXml [Good01s].

Η νέα αυτή τεχνολογία έχει γίνει το πιο επιτυχημένο πρότυπο για την εγγραφή και ανάγνωση της μουσικής σημειογραφίας μετά το Midi. Η εικόνα που ακολουθεί παρουσιάζει τα προϊόντα στα οποία έχει υιοθετηθεί και στα οποία πραγματοποιείται εγγραφή και ανάγνωση MusicXml, μόνο εγγραφή και μόνο ανάγνωση αντίστοιχα.

---

<sup>6</sup> Η **xml** (**extensible markup language**) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων.

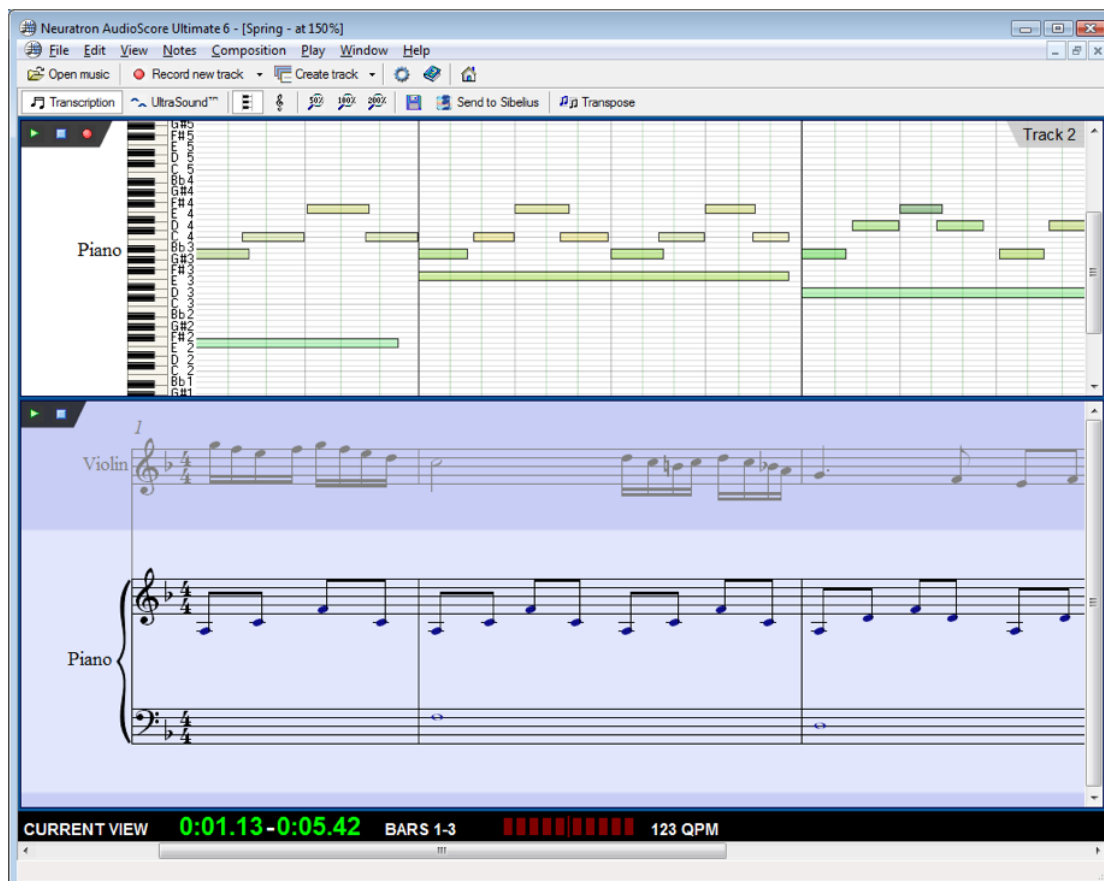


Εικόνα 2: Η εμπορική χρήση του προτύπου MusicXml

Περισσότερες πληροφορίες σχετικά με το πρότυπο MusicXml μπορούν να αναζητηθούν στην επίσημη ιστοσελίδα:

<http://www.recordare.com/xml.html>

Από τα παραπάνω θα αναφερθούμε ενδεικτικά στο προϊόν **AudioScore Ultimate** της εταιρίας **Neuratron** (η πρώτη του έκδοση κυκλοφόρησε τον 1999), το οποίο εκτός από την αναγνώριση και μουσική καταγραφή μονοφωνικών μουσικών σημάτων (\*.wav) καθιστά ικανή πλέον και την καταγραφή πολυφωνικής μουσικής από CD και mp3 επιτρέποντας ταυτόχρονα μεγάλο βαθμό ελέγχου και παρακολούθησης της διαδικασίας. Το προϊόν είναι συμβατό με διάφορα μουσικά προγράμματα, όπως τα Sibelius, Finale και Cubase επιτρέποντας έτσι μεγάλη ευελιξία.



**Εικόνα 3: Παράδειγμα εκτέλεσης του προγράμματος AudioScore Ultimate 6 της Neuratron. Αναγνώριση και καταγραφή πολυφωνικής μουσικής**

Κάτι που πρέπει τέλος να αναφερθεί είναι το πόσο ξεκάθαρο είναι πως σε εργασίες αυτής της φύσης η μη επαρκής γνώση μπορεί να οδηγήσει σε προβλήματα ή παραλείψεις. Ένα από τα πιο ουσιαστικά παραδείγματα αυτού έχουν αποτελέσει τα διάφορα λογισμικά μουσικής σημειογραφίας. Καθώς την δεκαετία του '90 σημειώθηκε σημαντική εξέλιξη στα συστήματα σύνθεσης παρτιτούρας, παρατηρήθηκε μιά τάση αυτοματοποίησης της διαδικασίας η οποία λόγω παραλείψεων οδηγούσε σε περιορισμούς και λάθη, όπως έχει σχολιάσει ο Donald Byrd [Byrd94], δημιουργός του συστήματος μουσικής μετάφρασης/καταγραφής SMUT (1977). Το σύστημα αυτό δεχόταν ως είσοδο μιά αλφαριθμητική ακολουθία συγκεκριμένης κωδικοποίησης και παρήγαγε την αντιστοιχούσα μουσική σημειογραφία. Το σύστημα αποτέλεσε ένα από τα πρώτα στο είδος του.

“ Ο βασικός λόγος φαίνεται να είναι πως τα όσα λαμβάνονται υπόψη από τα διάφορα λογισμικά, σχετικά με έννοιες όπως ο ρυθμός, οι περίπλοκες συγχορδίες, βασικά χαρακτηριστικά μουσικών οργάνων και λοιπά, σε κάποιες περιπτώσεις αποδεικνύονται ανεπαρκή σε σύγκριση με τα όσα το μεγαλείο της μουσικής έχει να επιδείξει. ” , συμπληρώνει ο Donald Byrd. “ Η καταγραφή της μουσικής αποτελεί ένα ιδιαίτερα περίπλοκο θέμα με πολλές εξαιρέσεις και λεπτολογίες. Θα πρέπει λοιπόν το σύστημα να γνωρίζει σε ικανοποιητικό βαθμό να εκτελέσει το σωστό και αναμενόμενο χωρίς να στηρίζεται σε εσφαλμένες υποθέσεις. ”

### 2.2.5 Συνεισφορά Εργασίας

Σε αυτό το σημείο κρίνεται απαραίτητη η σύνδεση της παρούσας εργασίας με τα όσα έχουν παρουσιαστεί στην παρούσα ενότητα (§2.2).

Στην παράγραφο 2.2.3 έγινε μιά παράθεση πληροφοριών και σχετικών μεθόδων/εργασιών του ευρύτερου πεδίου της Αυτόματης Μουσικής Καταγραφής χωρίς να γίνεται περαιτέρω ανάλυση αυτών. Αυτό γίνεται διότι η εργασία πραγματοποιήθηκε στα πλαίσια του έργου Vemus, ενός πλαισίου μουσικής διδασκαλίας δημοφιλών πνευστών μονοφωνικών μουσικών οργάνων, όπως το κλαρινέτο. Κατά συνέπεια η επεξεργασία της ακουστικής πληροφορίας στην κατεύθυνση της αναγνώρισης μουσικών γεγονότων που πραγματοποιείται στα πλαίσια της εργασίας περιορίζεται στην εκτίμηση των απλούστερων παραμέτρων, του pitch και της έντασης, της μονοφωνικής μουσικής, κάνοντας χρήση υπάρχοντος εργαλείου και έτσι δεν αποτελεί πρωτότυπη συνεισφορά της εργασίας.

Το βασικό πρόβλημα το οποίο καλείται να επιλύσει η εργασία είναι η οργάνωση αυτών των παραμέτρων και η κατάλληλη κωδικοποίησή τους με στόχο τη δημιουργία της βέλτιστης παρτιτούρας εξόδου. Για το λόγο αυτό παρατίθενται και κάποια βασικά στοιχεία σχετικά με την ιστορία και χρήση της ψηφιακής παρτιτούρας/μουσικής σημειογραφίας από υπολογιστή (§2.2.4).

Η μεθοδολογία που ακολουθείται περιλαμβάνει την επεξεργασία μονοφωνικών μουσικών σημάτων και την οργάνωση των μαθηματικών ποσοτήτων που παράγονται από τη διαδικασία αυτή έτσι ώστε να φέρουν μουσική πληροφορία. Τα δεδομένα κωδικοποιούνται κατάλληλα με γνώμονα τα χαρακτηριστικά της μουσικής που μιά παρτιτούρα ορίζει. Μετά την ολοκλήρωση της διαδικασίας αυτής το σύστημα είναι σε θέση να παράγει ένα αρχείο εξόδου το οποίο είναι εύκολα κατανοητό από τους μουσικούς, μιά ψηφιακή παρτιτούρα, επιτρέποντας την περαιτέρω επεξεργασία (διορθώσεις/συμπληρώσεις) και χρήση της προς την επιθυμητή κατεύθυνση (μουσικολογική ανάλυση, μουσική εκπαίδευση μέσω Διαδικτύου και άλλα.). Συμπληρωματικά, προτείνεται η μελέτη της ροής του προγράμματος όπως παρουσιάζεται στο τέλος της παραγράφου 4.3.1.

Σημειώνεται πως η διαδικασία της αναδημιουργίας της παρτιτούρας, η οποία κρίθηκε απαραίτητη για την ολοκληρωμένη λειτουργία του συστήματος αυτόματης καταγραφής και παρουσιάζεται σε συνέχεια της βασικής μεθοδολογίας (§4.3.4), θυμίζει την περιγραφή του συστήματος μουσικής μετάφρασης/καταγραφής SMUT του Donald Byrd το οποίο αναφέρεται στην ίδια ενότητα (§2.2.4). Η διαδικασία αυτή παρέχει έναν άμεσο τρόπο διόρθωσης του αρχείου εξόδου σε περίπτωση που ο χρήστης δε διαθέτει κάποιο άλλο εργαλείο επεξεργασίας παρτιτούρας (music score editor).

# 3

## *Θεωρητικό υπόβαθρο και εργαλεία*

Στην παρούσα ενότητα παρουσιάζονται τα διαθέσιμα εργαλεία καθώς και οι αλγόριθμοι ή μεθοδολογίες που έχουν προταθεί για την επίλυση συγκεκριμένων προβλημάτων και δεν αποτελούν συνεισφορά της παρούσας εργασίας. Το υλικό αυτό θα προσαρμοστεί στη συνέχεια (στο Κεφάλαιο 4) στις απαιτήσεις του συστήματος το οποίο υλοποιείται.

Για την ανάλυση του μουσικού σήματος και τη συλλογή δεδομένων εκπαίδευσης από διαφορετικές ηχογραφήσεις μουσικής (στάδιο πρώτο, §4.1) θα κάνουμε χρήση του λογισμικού Praat (§3.1). Η εφαρμογή του κανόνα απόφασης του κοντινότερου γείτονα (§3.2.2) με χρήση των δεδομένων εκπαίδευσης που συλλέχθηκαν θα αποτελέσει τη βάση της δημιουργίας του αναγνωριστή των pitch που μπορούν να αναπαραχθούν από το μουσικό όργανο υπό εξέταση, το κλαρινέτο (στάδιο δεύτερο, §4.2). Κατά την οργάνωση και κωδικοποίηση των γνώσεων που στο τρίτο στάδιο διαθέτουμε (§4.3), θα επιλέξουμε το πρότυπο MusicXml (§3.3) για την κωδικοποίηση της ψηφιακής παρτιτούρας που αποτελεί το αρχείο εξόδου του συνολικού συστήματος. Τέλος, για τον έλεγχο του βαθμού επιτυχίας του αρχείου παρτιτούρας που δημιουργείται από το σύστημα αυτόματης καταγραφής υλοποιείται μία διαδικασία αξιολόγησης (§4.3.3) η οποία βασίζεται στη μετρική του Levenshtein (Levenshtein distance) (§3.4).

### *3.1 Επεξεργασία Μουσικού Σήματος με χρήση του Praat*

Το σήμα ορίζεται ως μία φυσική ποσότητα η οποία διαφοροποιείται με το χρόνο, χώρο ή κάποια άλλη ανεξάρτητη ποσότητα ή ποσότητες. Ο ήχος θεωρείται ως ένα μονοδιάστατο σήμα, δηλαδή μία συνάρτηση του χρόνου, που αναπαριστά την ένταση του αέρα στο ακουστικό κανάλι. Η ταλάντωση της χορδής ή του αέρα σε σωλήνα, για έγχορδα ή πνευστά αντίστοιχα, οδηγεί στη δημιουργία κυμάτων τα οποία μέσω του σώματος του μουσικού οργάνου μετατρέπονται σε ακουστικά κύματα. Τα κύματα αυτά μπορούν να έχουν μία συγκεκριμένη συχνότητα, πολλαπλάσιά της ή συνδυασμό αυτών. Τα κύματα λαμβάνονται από τον ακροατή ως ήχοι ενός συγκεκριμένου τόνου, προσδίδοντάς τους έτσι κάποιο pitch.

Συνεπώς, τα δυο πεδία στα οποία μπορεί ένα μουσικό σήμα να αναλυθεί είναι ο χρόνος και η συχνότητα. Εάν ο χρόνος είναι το πεδίο στο οποίο η μουσική μπορεί να εκτελεστεί και ηχογραφηθεί, η συχνότητα είναι το πεδίο στο οποίο μπορεί να αναπαρασταθεί και να γίνει κατανοητή.

Όπως έχει ήδη αναφερθεί το “ύψος” μιάς νότας σε μιά παρτιτούρα αντιστοιχεί στη θεμελιώδη συχνότητα  $F_0$  του ακουστικού σήματος. Το φάσμα της ανθρώπινης ακοής εκτείνεται προσεγγιστικά από τα 20 Hz έως τα 20 kHz, και αυτό αντιστοιχεί σε 10 περίπου οκτάβες αντιληπτών pitch. Οι νότες που διαφέρουν κατά μιά οκτάβα (Εικόνα 4) δίνουν τη αίσθηση του ίδιου τόνου με μιά μεγαλύτερη ή μικρότερη οξύτητα, και οι συχνότητές τους είναι η μιά διπλάσια της άλλης. Η σχέση αυτή της οκτάβας αποτελεί ένα φυσικό φαινόμενο το οποίο θεωρείται ως το θαύμα της μουσικής.



**Εικόνα 4: Διάστημα οκτάβας C4 (έστω  $f_1$ ) και C5 (τότε  $f_2 = 2 \cdot f_1$ )**

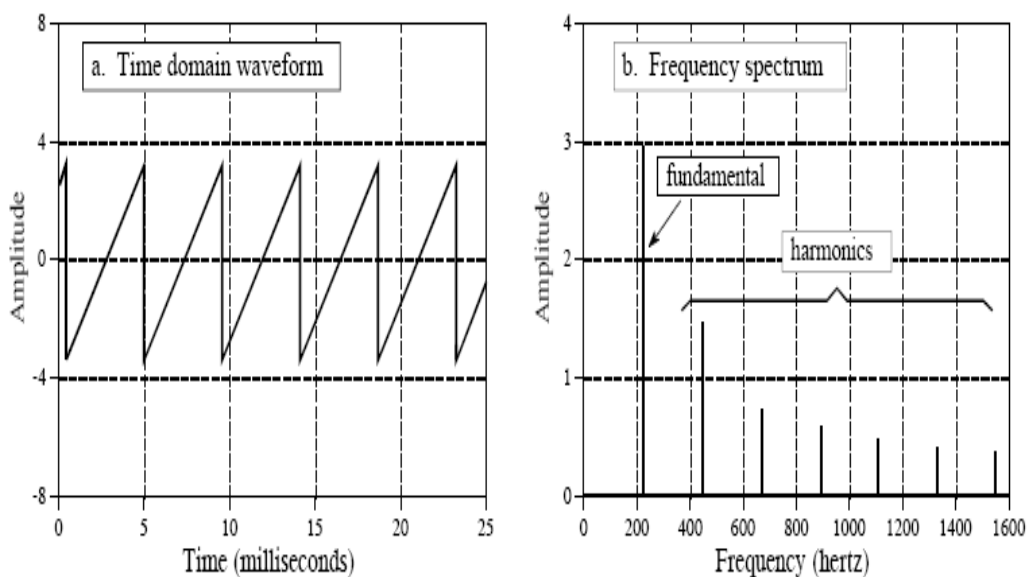
Από μαθηματικής άποψης, η συχνότητα υπολογίζεται μέσω του Μετασχηματισμού Fourier. Παρόλα αυτά ο ορισμός αυτός δεν είναι πρακτικά εφαρμόσιμος στην περίπτωση μας, καθώς η συχνότητα ορίζεται μόνο για απείρου μήκους, στάσιμα, συνεχή και τέλεια ημιτονοειδή κύματα, κατηγορία στην οποία δεν ανήκουν τα σήματα του πραγματικού κόσμου.

Καθώς μεγέθη όπως ο χρόνος ή η συχνότητα από μόνα τους δεν αρκούν για την μουσική αναπαράσταση, στρεφόμαστε σε αναπαραστάσεις οι οποίες επιτρέπουν το συνδυασμό των δύο. Αυτή η προσέγγιση εφαρμόζεται στη λογική των παραθύρων (frames), αποσπάσματα δηλαδή του σήματος ισχυρά τοπικά ως προς το χρόνο. Με αυτόν τον τρόπο έχουμε τη δυνατότητα να ορίσουμε μιά απλή χρονοσυχνοτική αναπαράσταση, το φασματογράφημα (spectrogram) ή φάσμα (cepstral) του σήματος.

Γιατί όμως μιά τέτοια μουσική αναπαράσταση μας είναι πρακτικά χρήσιμη; Καταρχήν, διότι μας δίνει τη δυνατότητα μιάς ανάλυσης η οποία δεν είναι εφικτό να πραγματοποιηθεί από το αυτί ή τον εγκέφαλό μας. Η αναλυτική οπτική αναπαράσταση μιάς μελωδίας μας παρέχει ταυτόχρονα ακριβείς τιμές των pitch και χρήσιμες πληροφορίες σχετικά με τη διάρκεια και το ρυθμό αυτών. Επιπρόσθετα, όταν ένας μουσικολόγος πραγματοποιεί μιά τέτοια ανάλυση, για παράδειγμα τη μουσική καταγραφή στο πεντάγραμμο, κανείς δεν γνωρίζει ποιά είναι η ακριβής διαδικασία η οποία ακολουθείται. Δεν είμαστε σε θέση να γνωρίζουμε τη λογική διαδικασία που συμβαίνει στον εγκέφαλό μας κατά την αντίληψη του μουσικού ήχου και τη μεταφορά του σε γραπτό μουσικό κείμενο. Αντιθέτως, όταν αυτή πραγματοποιείται μέσω μιάς υπολογιστικής διαδικασίας μπορούμε τουλάχιστον να γνωρίζουμε τον τρόπο με τον οποίο δουλεύει, ιδιαίτερα όταν κοινοποιείται. Οι δημιουργοί του λογισμικού praat [2], του εργαλείου το οποίο θα χρησιμοποιήσουμε, έχουν δημοσιεύσει αρκετά άρθρα στα οποία εξηγούν τους αλγόριθμους που έχουν υλοποιήσει ή προτείνουν επιπλέον δυνατότητες [Praat03], [SendP], [ScriptP]. Το praat, που στα Ολλανδικά σημαίνει “συζήτηση”, είναι ένα ελεύθερου κώδικα λογισμικό ανάλυσης και σύνθεσης σήματος φωνής, το οποίο δημιουργήθηκε από τους Paul Boersma και David Weenink, στο Institute of Phonetic Sciences του Πανεπιστημίου του Άμστερνταμ. Το λογισμικό αυτό δουλεύει θαυμάσια και για την ανάλυση συγκεκριμένης κατηγορίας μουσικών σημάτων.

Το **praat** [2] δουλεύει με αρχεία mono (απλής εγγραφής, μονοφωνικά), ενώ σε περίπτωση που διαθέταμε στερεοφωνικά αρχεία θα χρειαζόταν απλά να διαχωρίσουμε τα κανάλια. Στη δική μας περίπτωση χρησιμοποιούμε αρχεία ήχου **wav** (**w**aveform audio file format), τα οποία στη συνήθη μορφή τους περιέχουν ασυμπίεστη ακουστική πληροφορία με τη μορφή LPCM (linear pulse code modulation) και έτσι μπορούν με ευκολία να συνταχθούν και αξιοποιηθούν για την ανάλυση του ήχου που φέρουν. Σημειώνεται επίσης ότι το praat δεν μπορεί να χρησιμοποιηθεί ικανοποιητικά για την ανάλυση μουσικών οργάνων στα οποία αντηχούν μία σειρά από χορδές κατά την εκτέλεση μιάς εξ' αυτών, για παράδειγμα αναπαραγωγή μερικών (partials) συχνοτήτων που δεν αποτελούν αρμονικές (πολλαπλάσια της θεμελιώδους), όπως το πιάνο, ή ηχογραφήσεις οργάνων τα οποία αναπαράγουν συγχορδίες – ταυτόχρονη ύπαρξη περισσότερων του ενός pitch – όπως η κιθάρα, το πιάνο ή το βιολί. Στην περίπτωσή μας χρησιμοποιούμε ηχογραφήσεις ενός δημοφιλούς πνευστού μονοφωνικού μουσικού οργάνου, του κλαρινέτου, του οποίου τα βασικά χαρακτηριστικά παρουσιάζονται στην παράγραφο 4.1.1.

Τα τελευταία χρόνια η εξαγωγή του **pitch**, το οποίο αποτελεί τη βασική πληροφορία που απαιτείται για την καταγραφή της μουσικής από το σύστημά μας, γίνεται εκτιμώντας με χρήση κάποιου αλγορίθμου τη θεμελιώδη συχνότητα. Η κυματομορφή ενός μουσικού ήχου παρουσιάζει μία κάποια πολυπλοκότητα καθώς συνυπάρχουν πέραν της θεμελιώδους (fundamental) και μία σειρά από αρμονικές (harmonics) (Εικόνα 5). Εάν για παράδειγμα η θεμελιώδης συχνότητα είναι 220 Hz (tonic), που αντιστοιχεί στην θεωρητική τιμή της νότας B3 για το κλαρινέτο, θα συνυπάρχουν και οι συχνότητες των 440 Hz, η δεύτερη αρμονική που αντιστοιχεί στη νότα B4 (octave), 660 Hz, η τρίτη αρμονική που αντιστοιχεί στη νότα F#5 (fifths above octave), 880 Hz, η τέταρτη αρμονική που αντιστοιχεί στη νότα B5 (double octave) κ.τ.λ. (μέχρι και x14). Για περισσότερα από 15 χρόνια οι ερευνητές προσπαθούσαν να εξάγουν τη θεμελιώδη συχνότητα με απλή εφαρμογή κάποιων φίλτρων απευθείας στο σήμα. Για διάφορους λόγους, αυτό δεν αποτελεί πια πρακτικό τρόπο εκτίμησης της θεμελιώδους συχνότητας. Αντίθετα, σήμερα η εκτίμηση του pitch βασίζεται στη μελέτη των μερικών (partials) συχνοτήτων, τον υπολογισμό των σχετικών τους αποστάσεων και την εφαρμογή ενός αλγορίθμου υπολογισμού της θέσης της θεμελιώδους συχνότητας.



Εικόνα 5: Κυματομορφή στο πεδίο του χρόνου και της συχνότητας

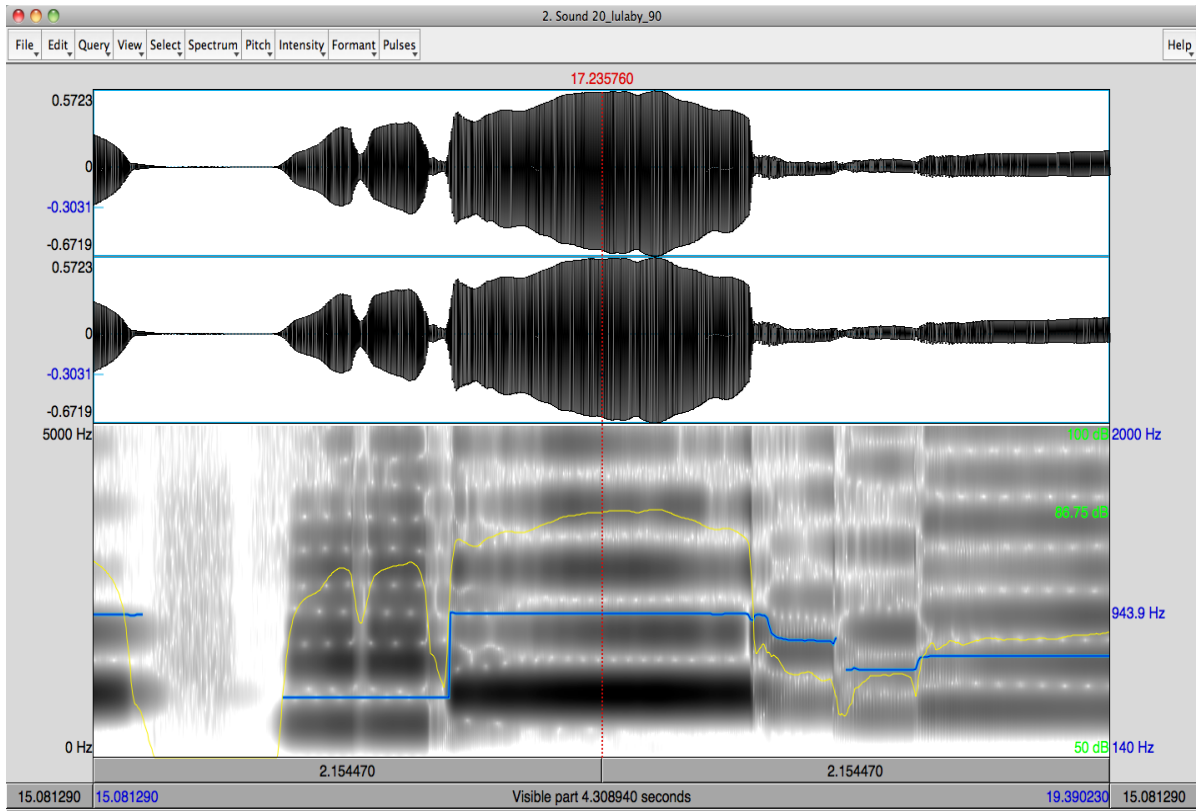
Με παρόμοιο τρόπο, το *praat* είναι σε θέση να διαχωρίσει την βασική πληροφορία στην περίπτωση ύπαρξης μιάς βασικής φωνής, και να κάνει μιά πολύ καλή εκτίμηση της θεμελιώδους συχνότητας αυτής. Στην περίπτωση όμως που υπάρχουν δύο φωνές της ίδιας δυναμικής που έχουν για παράδειγμα σχέση πέμπτης μεταξύ τους, λόγω χάρη με θεμελιώδεις συχνότητες 220 και 330 Hz, το *praat* θα θεωρήσει ως μιά φωνή τα δεδομένα, καθώς μόνο μιά συχνότητα εκτιμάται, και ενοποιώντας τις τιμές αυτές θα υπολογίσει την θεμελιώδη συχνότητα που αυτές ορίζουν. Στην συγκεκριμένη περίπτωση θα εκτιμηθεί η τιμή 110 Hz ως η θεμελιώδης συχνότητα η οποία μπορεί να συμπεριλάβει τις αρμονικές και των δυο φωνών (110 – 220 – 330 – 440 – 660 – 880..). Όμοια είναι τα προβλήματα που παρουσιάζονται κατά την ανάλυση των μουσικών οργάνων που αναπαράγουν συγχορδίες ή συντονισμό περισσοτέρων χορδών.

Μετά τον καθορισμό του εκτιμώμενου *pitch* το ενδιαφέρον μας στρέφεται στη διάρκεια κατά την οποία μιά νότα αναπαράγεται. Το αντιληπτό **onset** σε ένα μουσικό σήμα ορίζεται ως η αντιληπτή αρχή ενός διακριτού γεγονότος, που καθορίζεται από μιά αξιοπρόσεκτη αύξηση στην ένταση, ή από μιά ξαφνική αλλαγή στο *pitch* ή τη χροιά. Ο όρος **onset detection** αναφέρεται στη βιβλιογραφία ως η ανίχνευση των αρχών των διακριτών γεγονότων σε ένα ακουστικό σήμα. Οι κρουστικοί ήχοι, όπως τα τύμπανα, εμφανίζουν συχνά αιχμηρά *attacks*, σύντομα χρονικά διαστήματα κατά τα οποία παρατηρείται μιά ξαφνική αύξηση της ενέργειας στην κυματομορφή τους και μιά ευρείας ζώνης έκρηξη στο φάσμα τους. Οι χροιές άλλων οργάνων, όπως των πνευστών ή των εγχόρδων, παρουσιάζουν ομαλές μεταβάσεις από μιά νότα σε άλλη, και ο εντοπισμός αυτών των αλλαγών είναι πιο δυσδιάκριτος. Το *onset* στα μή-κρουστά όργανα αναφέρεται ως τονική αρχή (*tonal onset*). Ο χαρακτηρισμός των αρχών δεν είναι πάντα εύκολος, δεδομένου ότι μπορούν να καθοριστούν από τις αλλαγές στην ηχηρότητα, το ύψος και τη χροιά καθώς η μεταβολή της ενέργειας του σήματος μπορεί να υποδηλώνει την παρουσία *tremolo* ενώ της συχνότητας *vibrato*. Αυτές οι βαθμιαίες αλλαγές θεωρούνται ως μεταβολές στο εύρος ή τη συχνότητα, αλλά όχι ως διακριτά γεγονότα. [DT07S]

Ενδιαφέρουσα είναι η μεθοδολογία της ανάκτησης και ανάλυσης πληροφορίας σχετικά με το μουσικό **ρυθμό**, βάσει της οπτικής παρατήρησης και διαχωρισμού των μουσικών γεγονότων όπως περιγράφεται από την μουσικολογική ανάλυση με χρήση του *praat* του Μουσικού τμήματος του Πανεπιστημίου του Άμστερνταμ [PraatM]. Η μεθοδολογία κρίνεται πως ενδείκνυται για την ανάλυση του ρυθμού σε ηχογραφήσεις με μικρή διάρκεια ή σε τμήματα αυτών και δε θα αναλυθεί περαιτέρω στα πλαίσια αυτής της εργασίας.

Η παρούσα εργασία κάνει χρήση του *praat* για τις ανάγκες της ανάκτησης πληροφοριών αναφορικά με το ακριβές **pitch** το οποίο αναπαράγεται κάθε χρονική στιγμή - ή τη μή ύπαρξη κάποιου *pitch* στο εύρος που μας ενδιαφέρει, που ισοδυναμεί με 'ησυχία' ή παύση - αλλά και των τιμών που ορίζουν την περιβάλλουσα του πλάτους που μας παρέχει πληροφορίες σχετικά με την ένταση/δυναμική του ακουστικού σήματος (**intensity**), μιά απαραίτητη παράμετρο για το διαχωρισμό των ξεχωριστών γεγονότων (§4.3). Ακολουθεί ένα παράδειγμα της χρήσης του (Εικόνα 6).





**Εικόνα 6: Οπτική αναπαράσταση σήματος από ηχογράφηση κλαρινέτου με χρήση του εργαλείου praat. Περιγραφή:** Η παραπάνω εικόνα απεικονίζει ένα στιγμιότυπο του σήματος και της ανάλυσης αυτού με τη βοήθεια του praat. Το ακουστικό σήμα προέκυψε από την ηχογράφηση του μουσικού κομματιού lullaby από κλαρινέτο. Στο επάνω τμήμα μπορούμε να παρατηρήσουμε τη γραφική αναπαράσταση της κυματομορφής του σήματος στο πεδίο του χρόνου και στο κάτω μέρος το φάσμα του. Οι οριζόντιες γραμμές μπλε χρώματος αναπαριστούν την εκτιμώμενη θεμελιώδη συχνότητα/pitch (σε Hz) και η κίτρινη συνεχόμενη καμπύλη την περιβάλλουσα του πλάτους του σήματος που αντιστοιχεί στην ένταση/δυναμική του σήματος (σε decibel).

### 3.2 Αναγνώριση Προτύπων

Τα δεδομένα τα οποία συλλέχθηκαν χρησιμοποιώντας το λογισμικό επεξεργασίας σήματος praat είναι απαραίτητο να οργανωθούν και να κωδικοποιηθούν ώστε να μας παρέχουν αξιοποιήσιμη γνώση. Η γνώση αυτή θα μας επιτρέψει να προχωρήσουμε στις απαιτούμενες ενέργειες ανάλογα με το τί καθένα από τα δεδομένα αυτά αντιπροσωπεύει. Η διαδικασία της περιγραφής και κατάταξης αντικειμένων σε κατηγορίες με σαφώς ορισμένα χαρακτηριστικά αποτελεί αντικείμενο της Αναγνώρισης Προτύπων.

### 3.2.1 Αλγόριθμοι αναγνώρισης προτύπων

Η ευκολία με την οποία αναγνωρίζουμε ένα πρόσωπο, αντιλαμβανόμαστε τις λέξεις και τους ήχους ή κατανοούμε το γραπτό κείμενο σίγουρα δεν είναι ενδεικτική της σύνθετης διαδικασίας που ενυπάρχει σε αυτές τις εργασίες αναγνώρισης προτύπων. Η αναγνώριση προτύπων είναι η απαραίτητη αυτή διαδικασία κατά την οποία τα υπάρχοντα ακατέργαστα δεδομένα κατηγοριοποιούνται με βάση κάποιο ή κάποια κριτήρια. Η κατηγορία στην οποία εκτιμάται ότι ανήκουν τα δεδομένα καθορίζει τις περαιτέρω επιλογές και αποφάσεις μας.

Οι αλγόριθμοι αναγνώρισης προτύπων αναφέρονται κατά βάση στην κατηγοριοποίηση δεδομένων. Υπάρχουν δυο κατηγορίες τέτοιων τεχνικών, η **μάθηση με επίβλεψη (supervised learning)** και η **μάθηση χωρίς επίβλεψη (unsupervised learning)**. Στην πρώτη περίπτωση μιά σειρά από δεδομένα εκπαίδευσης είναι διαθέσιμα, τα οποία λαμβάνονται υπόψη για την εκτίμηση της κλάσης στην οποία τα νέα δεδομένα θα κατηγοριοποιηθούν. Αντίθετα, μιά σειρά από τεχνικές μάθησης χωρίς επίβλεψη μας δίνουν τη δυνατότητα συγκέντρωσης και ομαδοποίησης (clustering) δεδομένων, τα οποία αποτελούν μιά αυθαίρετη διανομή, σε κατηγορίες, τις κλάσεις.

Τα μουσικά δεδομένα διαθέτουν κάποια σαφώς ορισμένα χαρακτηριστικά, αλλά μπορούμε να παρατηρήσουμε μιά μεγάλη διαφοροποίηση μεταξύ διαφορετικών οργάνων ή περιβαλλόντων ηχογράφησης. Τα πιθανοτικά ή γεωμετρικά μοντέλα αποτελούν ένα πολύ χρήσιμο εργαλείο για την μοντελοποίηση των μουσικών ήχων και των διαφόρων χαρακτηριστικών τους ακριβώς διότι μπορούν να χειριστούν αυτή την ποικιλομορφία.

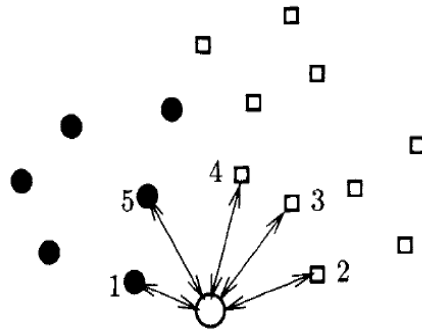
### 3.2.2 Η μέθοδος του κοντινότερου γείτονα (Nearest Neighbor Classification)

Ο **κανόνας απόφασης του κοντινότερου γείτονα (Nearest Neighbor Classification)** αποτελεί μιά απλή μέθοδο μάθησης με επίβλεψη. Απαιτεί τη μέτρηση των αποστάσεων  $d(x, y_j)$  μεταξύ του δείγματος  $x$  και των δεδομένων εκπαίδευσης  $y_j$ , για όλα τα  $j$ . Επιλέγεται η κατηγορία  $n$  για την οποία ικανοποιείται η μικρότερη απόσταση  $d(x_i, y_j)$  και  $j \in n$ .

Έστω  $k$  ένας θετικός ακέραιος (περιττός αριθμός για την αποφυγή διφορούμενων καταστάσεων), τότε η μέθοδος λαμβάνει τη μορφή **k-Nearest-Neighbors** σύμφωνα με την οποία επιλέγεται η κατηγορία η οποία κατέχει την πλειοψηφία των κοντινότερων αποστάσεων μεταξύ των  $k$  που έχουν υπολογιστεί. Η επιλογή  $k > 1$  όπως γίνεται εύκολα αντιληπτό οδηγεί σε μεγαλύτερη ακρίβεια ταξινόμησης (Εικόνα 7).

Η εφαρμογή του αλγορίθμου παρουσιάζει συνέπεια ως προς τα αποτελέσματα και ενδείκνυται σε περιπτώσεις δεδομένων του πραγματικού κόσμου. Καθώς το  $k$  αυξάνεται το ποσοστό σφάλματος πλησιάζει σταδιακά την κατώτατη εφικτή τιμή, αυτή του Bayes.

Ακολουθεί ένα σχηματικό παράδειγμα της εφαρμογής του 5-Nearest-Neighbors (Εικόνα 7).



**Εικόνα 7: 5-Nearest-Neighbors. Περιγραφή:** Από τις δυο κλάσεις των δυο διαστάσεων στοιχείων βάσει πλειοψηφίας το δείγμα 0 κατηγοριοποιείται ως □ (παρότι η μικρότερη απόσταση είναι αυτή του άλλου δείγματος και το αποτέλεσμα θα ήταν διαφορετικό στην περίπτωση εφαρμογής του απλού Nearest Neighbor).

### 3.3 Οι προδιαγραφές του προτύπου MusicXml

Η μουσική σημειογραφία είναι μιά συμβολική αναπαράσταση της μουσικής η οποία μας παρέχει πληροφορίες αναφορικά με το ακουστικό αποτέλεσμα μιάς εκτέλεσης. Περιλαμβάνει πληροφορίες σχετικά με το σύνολο από νότες που συνθέτουν το μουσικό αποτέλεσμα με βασικότερο χαρακτηριστικό αυτό των *pitch* που αντιλαμβανόμαστε ως ακροατές, αλλά και εξίσου καθοριστικές για το ακουστικό αποτέλεσμα ιδιότητες αυτών όπως η *διάρκεια* και η οργάνωσή τους στο χρόνο, στοιχεία τα οποία ορίζουν το *ρυθμό* ενός μουσικού κομματιού. Λιγότερο άμεσα σχετιζόμενη με το ακουστικό αποτέλεσμα είναι η *αρμονία* που διέπει τη σύνθεση του κομματιού η οποία συνδέεται με τη μουσικολογική ανάλυση αυτού, ενώ η *δυναμική* αποτελεί χαρακτηριστικό γνώρισμα των φράσεων, δηλαδή τμημάτων της μουσικής εκτέλεσης, και όχι κάθε ξεχωριστής νότας.

Το πρότυπο MusicXml [3] αποτελεί έναν φιλικό προς το Διαδίκτυο τρόπο δημιουργίας, δημοσίευσης και επεξεργασίας της ψηφιακής παρτιτούρας. Η 'μετάφραση' των μαθηματικών ποσοτήτων που προκύπτουν από την επεξεργασία του ακουστικού σήματος σε μουσικές παραμέτρους η οποία πραγματοποιείται από το σύστημα της παρούσας εργασίας θα πραγματοποιηθεί σύμφωνα με το πρότυπο MusicXml, το οποίο αποτελεί ένα ελεύθερο και κοινό πρότυπο και έτσι επιτρέπει την ανταλλαγή δεδομένων από διαφορετικά μέσα διευκολύνοντας σημαντικά τους χρήστες.

Στην παράγραφο 2.2.4 αναφέραμε πως οι ίδιοι λόγοι που καθιστούν την xml κωδικοποίηση ενδιαφέρουσα - διαδικτυακή χρησιμότητα, ευκολία σύνταξης του αρχείου, φιλική προς το χρήστη και ευανάγνωστη μορφή - ισχύουν αντίστοιχα και για την MusicXml κωδικοποίηση. Και αυτό θα γίνει πιο κατανοητό με ένα απλό παράδειγμα (Εικόνες 8a-8b).

Το παράδειγμα παρουσιάζεται στην επίσημη ιστοσελίδα της Recordare [3], και έχει τίτλο "Hello world in MusicXML" σε αντιστοιχία με το πρόγραμμα τυπώματος της φράσης "Hello world" ως το πρώτο πρόγραμμα που προτείνεται κατά την εκμάθηση μιάς νέας γλώσσας προγραμματισμού. Το πιο απλό πρόγραμμα στην περίπτωση του προτύπου MusicXml αποτελεί το 'τύπωμα' ενός μουσικού κομματιού ενός μόνο μέτρου με τη νότα C4 διάρκειας ενός ολόκληρου σε ρυθμό 4/4. Παρόμοιος θα είναι ο τρόπος με τον οποίο θα χρειαστεί να κωδικοποιήσουμε τα δεδομένα μας κατά τη σχεδίαση του συστήματος το οποίο δημιουργεί το αντίστοιχο αρχείο για το μουσικό κομμάτι που ο χρήστης έχει εκτελέσει.



Εικόνα 8α: Μουσική σημειογραφία - νότα C4

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 2.0 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="2.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

Εικόνα 8β: MusicXml κώδικας 'τυπώματος' της μουσικής αναπαράστασης της Εικόνας 8α

Ας αναλύσουμε τη δομή του αρχείου.

```
<?xml version="1.0" standalone="no"?>
```

Στην αρχή κάθε αρχείου xml είναι απαραίτητη η παραπάνω δήλωση. Η τιμή "no" της παραμέτρου standalone σημαίνει πως το έγγραφο ορίζεται με μία εξωτερική δήλωση σε ένα άλλο αρχείο, το οποίο αναφέρεται στη συνέχεια.

```
<!DOCTYPE score-partwise PUBLIC  
"-//Recordare//DTD MusicXML 1.0 Partwise//EN"  
"http://www.musicxml.org/dtds/partwise.dtd">
```

Σε αυτό το σημείο αναφέρουμε ότι χρησιμοποιούμε το πρότυπο MusicXml, και συγκεκριμένα τη μορφή partwise score κατά την οποία ένα σύνολο από μέτρα περιέχονται στα μέρη. Κάνουμε μία δημόσια (public) δήλωση συμπεριλαμβάνοντας τη Διαδικτυακή διεύθυνση για το σχετικό DTD (Document Type Definition) όπου περιλαμβάνεται το σύνολο των κανόνων τους οποίους θα πρέπει να ακολουθήσουμε.

```
<score-partwise version="2.0">
```

Αυτή είναι η ρίζα του Document Type. Το score-partwise στοιχείο αποτελείται από μέρη, και καθένα από αυτά αποτελείται από μέτρα. Υπάρχει επίσης και η επιλογή score-timewise όπου το κάθε στοιχείο αποτελείται από μέτρα τα οποία αποτελούνται από μέρη. Η ιδιότητα version επιτρέπει στο πρόγραμμα να αναγνωρίσει πιο εύκολα ποιά εκδοχή του MusicXml χρησιμοποιείται (εάν γίνεται χρήση αρχείων MusicXml 1.0 η ιδιότητα μπορεί να παραλειφθεί).

```
<part-list>  
<score-part id="P1">  
  <part-name>Part 1</part-name>  
</score-part>  
</part-list>
```

Όποια μορφή κι αν υιοθετείται (scorewise ή partwise) ένα αρχείο MusicXml αρχικοποιείται με μία επικεφαλίδα η οποία απαριθμεί τα διαφορετικά μουσικά μέρη στην παρτιτούρα. Το παρόν παράδειγμα αποτελεί την πιο απλή λίστα από μέρη (part-list) καθώς περιέχει ένα μέρος (το score-part), την απαραίτητη ιδιότητα ταυτοποίησης αυτού (id) και το στοιχείο με το όνομά του (part-name).

```
<part id="P1">
```

Στο σημείο αυτό ξεκινά το πρώτο (και το μοναδικό στην περίπτωσή μας) μέρος του εγγράφου. Η ιδιότητα id θα πρέπει να αναφέρεται σε μία id μιάς επικεφαλίδας του μέρους (score-part).

```
<measure number="1">
```

Το πρώτο μέτρο του πρώτου μέρους ξεκινά.

```
<attributes>
```

Το στοιχείο attributes περιλαμβάνει βασικές πληροφορίες απαραίτητες για την σωστή ερμηνεία του συνόλου από νότες και μουσικά δεδομένα που ακολουθούν στο συγκεκριμένο μέρος.

```
<divisions>1</divisions>
```

Κάθε νότα στο MusicXml διαθέτει ένα στοιχείο duration (διάρκεια). Το στοιχείο divisions παρέχει τη μονάδα του μέτρου για το στοιχείο duration με την έννοια των υποδιαιρέσεων ανά τέταρτο.

Από τη στιγμή που στο παράδειγμά μας το μόνο που έχουμε είναι ένα ολόκληρο, δε θα χρειαστεί ποτέ να διαιρέσουμε ένα τέταρτο, και έτσι θέτουμε την τιμή της παραμέτρου duration ίση με 1.

Οι διάρκειες στη μουσική τυπικά εκφράζονται ως κλάσματα, όπως  $\frac{1}{4}$  ('ένα τέταρτο') και  $\frac{1}{8}$  ('ένα όγδοο'). Οι διάρκειες στο MusicXml αναπαρίστανται κατά αντίστοιχο τρόπο ως κλάσματα. Καθώς ο παρονομαστής σπάνια χρειάζεται να μεταβληθεί, ορίζεται ξεχωριστά από το στοιχείο divisions κι έτσι μόνο ο αριθμητής μένει να οριστεί για κάθε ξεχωριστή νότα. Αυτή η λογική είναι όμοια με αυτή που χρησιμοποιείται στη Midi αναπαράσταση της διάρκειας κάθε νότας.

```
<key>
  <fifths>0</fifths>
</key>
```

Το στοιχείο key χρησιμοποιείται για την αναπαράσταση του οπλισμού, δηλαδή του συνόλου από διέσεις ή υφέσεις της κλίμακας στην οποία έχει γραφτεί ένα μουσικό κομμάτι. Στο παράδειγμά μας είμαστε στην C major (Ντο μείζονα) και συνεπώς δεν υπάρχουν υφέσεις ή διέσεις και το στοιχείο fifths είναι 0. Το όνομα fifths προέρχεται από την αναπαράσταση του οπλισμού ανά ένα κυκλικό διάστημα πέμπτης. Μας επιτρέπει την αναπαράσταση του οπλισμού με ένα μόνο στοιχείο περιλαμβάνοντας ταυτόχρονα το σύνολο από διέσεις ή υφέσεις δηλώνοντάς το με τον αντίστοιχο θετικό ή αρνητικό αριθμό αντίστοιχα.

```
<time>
  <beats>4</beats>
  <beat-type>4</beat-type>
</time>
```

Το στοιχείο time αναπαριστά το μουσικό ρυθμό, τον τρόπο δηλαδή κατά τον οποίο οργανώνονται οι νότες στο χρόνο. Αποτελείται από δυο συστατικά στοιχεία, το beat και το beat-type, τον αριθμητή και τον παρονομαστή αντίστοιχα του μουσικού ρυθμού (εδώ 4/4).

```
<clef>
  <sign>G</sign>
  <line>2</line>
</clef>
```

Το πρότυπο MusicXml επιτρέπει την αναπαράσταση μιάς ποικιλίας μουσικών κλειδιών. Από αυτά το πλέον σύνθηες είναι το 'κλειδί του Σολ' (G clef) της δεύτερης (2) γραμμής του πενταγράμμου.

```
</attributes>
<note>
```

Στο σημείο αυτό έχουμε ολοκληρώσει τις βασικές ιδιότητες και μπορούμε να ξεκινήσουμε με την πρώτη (και μοναδική στην περίπτωση μας) νότα.

```
<pitch>
  <step>C</step>
  <octave>4</octave>
</pitch>
```

Το στοιχείο pitch θα πρέπει αναγκαστικά να ορίζει τα στοιχεία step και octave. Προαιρετικά μπορεί να περιλαμβάνει και το στοιχείο alter στην περίπτωση αλλοιωμένης νότας (νότας με ύφεση ή δέση). Αυτά τα στοιχεία καθορίζουν το ακουστικό αποτέλεσμα. Για το λόγο αυτό θα πρέπει πάντα να περιλαμβάνεται το στοιχείο alter όταν πρόκειται για αλλοιωμένη νότα, ακόμη και στην περίπτωση που η συγκεκριμένη αλλοίωση περιλαμβάνεται στον οπλισμό. Στα πλαίσια της παρούσας εργασίας ωστόσο η παράμετρος αυτή παραλείπεται χάριν απλότητας, όπως θα εξηγηθεί και στην σχετική παράγραφο (§4.3.2 pp.47, όπου αναφέρεται πως "...δε λαμβάνονται υπόψη αρμονικές σχέσεις..").

```
<duration>4</duration>
```

Η τιμή του στοιχείου divisions έχει οριστεί ίση με 1 (υποδιαίρεση ενός τέταρτου), οπότε το στοιχείο duration της νότας του παραδείγματος θα οριστεί ίσο με 4, που ισοδυναμεί με διάρκεια τεσσάρων τετάρτων.

```
<type>whole</type>
```

Το στοιχείο `type` υποδηλώνει ότι η νότα συμβολίζεται ως ένα ολόκληρο (whole). Αυτό βέβαια μπορούμε να το συμπεράνουμε στην περίπτωση μας και από τη διάρκεια της νότας, αλλά όταν τα δεδομένα αυτά είναι ορισμένα ξεχωριστά είναι πιο εύκολο να εργαζόμαστε με εφαρμογές τόσο σημειογραφίας όσο και εκτέλεσης.

Σε κάποιες περιπτώσεις, μπορεί να είναι επιθυμητό τα δεδομένα μεταξύ σημειογραφίας και εκτέλεσης να μην είναι σε απόλυτη συμφωνία (περιπτώσεις τις οποίες δε θα εξετάσουμε καθώς στα πλαίσια της παρούσας εργασίας ο ρυθμός θεωρείται γνωστός και σταθερός σε όλη τη διάρκεια του μουσικού κομματιού). Τέτοια παραδείγματα αποτελούν jazz εκτελέσεις οι οποίες χαρακτηρίζονται από μία swing διάθεση ή άλλες εκφραστικές και άρα λιγότερο χρονικά αυστηρά καθορισμένες εκτελέσεις [Dix01]. Ακόμη και η μουσική του Bach περιλαμβάνει σημειογραφία η οποία υποδηλώνει ερμηνεία διαφορετική από αυτή που οι νότες αυτές καθαυτές ορίζουν.

`</note>`

Η νότα έχει οριστεί πλήρως.

`</measure>`

Το μέτρο έχει ολοκληρωθεί.

`</part>`

Το μέρος έχει ολοκληρωθεί.

`</score-partwise>`

Κι έτσι, η παρτιτούρα είναι πλήρως ορισμένη.

Ο μόνος περιορισμός που έχει μεγάλη σημασία είναι πως ναι μεν έχουμε την ελευθερία να επιλέξουμε τα στοιχεία που κρίνουμε απαραίτητα και να παραλείψουμε άλλα, μα θα πρέπει να διατηρούμε τη δεδομένη διάταξη. Στο αρχείο DTD το οποίο δηλώνεται στην αρχή του εγγράφου θα πρέπει να είναι σαφώς ορισμένη αυτή η διάταξη.

### 3.4 Μετρική Ομοιότητας Συμβολοσειρών (*Levenshtein distance*)

Μετά την ολοκλήρωση του συστήματος αυτόματης καταγραφής και στα πλαίσια της αξιολόγησης της λειτουργίας του θα χρησιμοποιήσουμε έναν αλγόριθμο ο οποίος υπολογίζει την ομοιότητα μεταξύ δυο πεδίων, στην περίπτωση μας δύο ακολουθιών από συμβολοσειρές, τις αλφαριθμητικές αναπαραστάσεις του συνόλου από νότες του αυτόματα παραχθέντος αρχείου και του επιθυμητού (§4.3.3).

Θα κάνουμε χρήση μιάς συνάρτησης η οποία ποσοτικοποιεί τις διαφορές μεταξύ δυο ακολουθιών. Στη βιβλιογραφία η διαδικασία αυτή αναφέρεται ως **edit distance**. Στη βασική της μορφή, όπως προτάθηκαν από τον **Levenshtein** (1965) [Levdist], η απόσταση δύο συμβολοσειρών ορίζεται ως ο αριθμός των χαρακτήρων που πρέπει να εισαγάγουμε, διαγράψουμε ή αντικαταστήσουμε προκειμένου να μεταβούμε από την μιά συμβολοσειρά στην άλλη. Το κόστος (penalty cost) καθεμιάς εκ των τριών διαδικασιών (εισαγωγή, διαγραφή και αντικατάσταση) επιλέγεται ίσο με 1, διατηρώντας έτσι την απλούστερη μορφή του αλγορίθμου, ώστε το αποτέλεσμα που λαμβάνεται να αντιπροσωπεύει το σύνολο των διορθώσεων στις οποίες ο χρήστης θα πρέπει να προχωρήσει μετά την αυτόματη δημιουργία του αρχείου παρτιτούρας. Όπως προκύπτει, όσο μικρότερη είναι η απόσταση που επιστρέφεται τόσο πιο όμοια είναι τα πεδία, και συνεπώς μεγαλύτερο το ποσοστό επιτυχίας του συστήματός μας.

Ακολουθεί ένα παράδειγμα υπολογισμού της απόστασης μεταξύ δυο αλφαριθμητικών ακολουθιών (Πίνακας 1) μετά την εφαρμογή του προσαρμοσμένου στα δεδομένα που μελετώνται κώδικα (Παράρτημα Β, Β3). Ο συμβολισμός που επιλέχθηκε εξηγείται στη σχετική παράγραφο §4.3.3.

|      |   |      |      |      |      |      |      |
|------|---|------|------|------|------|------|------|
|      |   | C4-1 | C4-2 | G4-1 | D4-1 | G4-1 | G4-1 |
|      | 0 | 1    | 2    | 3    | 4    | 5    | 6    |
| C4-1 | 1 | 0    | 1    | 2    | 3    | 4    | 5    |
| C4-1 | 2 | 1    | 1    | 2    | 3    | 4    | 5    |
| G4-1 | 3 | 2    | 2    | 1    | 2    | 3    | 4    |
| D4-1 | 4 | 3    | 3    | 2    | 1    | 2    | 3    |
| F4-1 | 5 | 4    | 4    | 3    | 2    | 2    | 3    |
| G4-1 | 6 | 5    | 5    | 4    | 3    | 2    | 2    |
| E4-1 | 7 | 6    | 6    | 5    | 4    | 3    | 3    |

**Πίνακας 1:** Πίνακας με το αποτέλεσμα της εφαρμογής της μετρικής Levenshtein σε δυο αλφαριθμητικές ακολουθίες. **Περιγραφή:** Κάθε στοιχείο της ακολουθίας αντιμετωπίζεται σαν μία οντότητα. Η πρώτη ακολουθία (πρώτη στήλη) οδηγεί στην δεύτερη (πρώτη σειρά) μετά από δυο αντικαταστάσεις (οι οποίες σημειώνονται με πορτοκαλί ανοιχτό) και μία διαγραφή (η οποία σημειώνεται με σκούρο πορτοκαλί). Συνολικά οι διορθώσεις στις οποίες ο χρήστης θα πρέπει να προχωρήσει για την απόκτηση του επιθυμητού αρχείου είναι τρεις, όπως υπολογίζεται και μετά την εφαρμογή της μετρικής στα δεδομένα (στοιχείο της κάτω δεξιά γωνίας του πίνακα).



# 4

## Ιδέα και Υλοποίηση

Στο κεφάλαιο αυτό θα αναλύσουμε την κεντρική ιδέα και τα βήματα της επεξεργασίας που ακολουθήσαμε για τον σχεδιασμό ενός συστήματος το οποίο είναι σε θέση να δημιουργεί αυτόματα την παρτιτούρα, η οποία αντιστοιχεί στο ακουστικό σήμα μιάς ηχογράφησης κλαρινέτου.

Στο **πρώτο στάδιο** μελετώνται τα τεχνικά χαρακτηριστά του μουσικού οργάνου υπό εξέταση, του κλαρινέτου. Η βασική παράμετρος η οποία μας ενδιαφέρει στην παρούσα εργασία είναι το εκτιμώμενο pitch. Οι θεωρητικές τιμές των pitch που αναπαράγονται από ένα μουσικό όργανο βεβαίως μας παρέχουν σημαντική πληροφορία. Όμως η επεξεργασία σημάτων του πραγματικού κόσμου είναι σαφές πως δεν μπορεί να βασιστεί μόνο σε αυτό. Για να προσδιορίσουμε την απόκλιση μεταξύ θεωρητικών και πρακτικά παρατηρούμενων τιμών, με τη βοήθεια του Praat, προχωρήσαμε στη συλλογή ενός αντιπροσωπευτικού δείγματος διαφορετικών στιγμιότυπων για κάποιες από τις 46 διαφορετικές νότες που μπορούν να αναπαραχθούν από το κλαρινέτο και στηριζόμενοι στη μαθηματική σχέση που τις διέπει κατασκευάσαμε έναν Πίνακα θεωρητικών-πραγματικών τιμών.

Στο **δεύτερο στάδιο**, τα δεδομένα αυτά (Πίνακας πρώτου σταδίου) θα αξιοποιηθούν ως δεδομένα μάθησης στην εφαρμογή του κανόνα απόφασης του κοντινότερου γείτονα σε ολοκληρωμένες ηχογραφήσεις. Με τον τρόπο αυτό είμαστε σε θέση να γνωρίζουμε το pitch το οποίο αναπαράγεται κάθε χρονική στιγμή και να συμπεράνουμε τη διάρκεια αυτού από το συνολικό αριθμό των συνεχόμενων εμφανίσεών του.

Το **τρίτο στάδιο** είναι εκείνο της οργάνωσης των δεδομένων και της 'μετάφρασής' τους σε νότες συγκεκριμένων χαρακτηριστικών, δηλαδή συγκεκριμένων τιμών pitch και διάρκειας, με σκοπό τη δημιουργία της ολοκληρωμένης παρτιτούρας η οποία μπορεί να περιγράψει την ηχογράφηση που έχει δοθεί. Οργανώνουμε τις τιμές pitch που διαθέτουμε από την εφαρμογή της μεθόδου του κοντινότερου γείτονα (στάδιο δεύτερο) ανά ομάδες με βάση το ρυθμό (δοθείσα τιμή παραμέτρου tempo) στον οποίο έχει γίνει η ηχογράφηση του μουσικού κομματιού και την ελάχιστη διάρκεια που εμφανίζεται σε αυτό (δοθείσα τιμή παραμέτρου min). Κάθε ομάδα τιμών που προκύπτει αντιστοιχεί στην εμφάνιση ενός pitch με ελάχιστη διάρκεια. Για κάθε νότα που θα 'τυπωθεί' θα πρέπει να ορίσουμε το pitch και τη συνολική διάρκεια αυτού (συνεχόμενη εμφάνιση ίδιου pitch με ταυτόχρονη διατήρηση της ενέργειας του σήματος σε υψηλό επίπεδο) προτού προχωρήσουμε στην αποθήκευση του συνόλου από νότες οργανωμένες στο χρόνο σύμφωνα με το ρυθμό της ηχογράφησης (δοθείσες τιμές beat και beat\_type).

## 4.1 Στάδιο πρώτο

### 4.1.1 Τεχνικά χαρακτηριστικά κλαρινέτου

Το **κλαρινέτο** είναι ένα πνευστό μουσικό όργανο. Στη σημερινή του μορφή εμφανίστηκε το 19ο αιώνα. Κατέχει βασική θέση στη συμφωνική ορχήστρα, και ανήκει στην κατηγορία των ξύλινων πνευστών. Πολύ σύνηθες είναι το κλαρινέτο και ως μέλος ορχηστρών της τζαζ. Στην Ελλάδα, αλλά και σε πολλές χώρες των Βαλκανίων, αποτελεί ένα από τα βασικά όργανα της παραδοσιακής μουσικής.

Ο συνδυασμός των καλυπτόμενων και αποκαλυπτόμενων οπών που διαθέτει επιτρέπουν την αναπαραγωγή μεγάλης έκτασης ήχων, μεγαλύτερη από τρεις οκτάβες, ανάλογα με τις ικανότητες του οργανοπαίκτη. Υπάρχει μιά αρκετά μεγάλη οικογένεια από κλαρινέτα, διαχωριζόμενα ανάλογα με την τονικότητά τους. Το πιο συνηθισμένο είναι το **κλαρινέτο σε Σι b / B b**. Το παραδοσιακό κλαρινέτο της Ελλάδας είναι το κλαρινέτο σε Ντο (C), το οποίο ηχεί όπου γράφεται σε αντίθεση με τα υπόλοιπα.

Ακολουθεί απεικόνιση της έκτασης του οργάνου που θα μελετήσουμε (Εικόνα 9) και ο αναλυτικός πίνακας (Πίνακας 2) με την αντιστοιχία pitch και των θεωρητικών τιμών αυτών. [CIPitch]



**Εικόνα 9: Η έκταση του κλαρινέτου σε Σι b / B b . Περιγραφή:** οπτική αναπαράσταση βάσει μουσικής σημειογραφίας, όπου παρουσιάζεται η χαμηλότερη και η υψηλότερη συχνότητα (δίπλα σημειώνεται το όνομα και η τιμή συχνότητας (Hz) των δυο pitch) τις οποίες μπορεί να παράγει το όργανο.

|            |             |            |             |            |            |             |            |             |            |             |            |
|------------|-------------|------------|-------------|------------|------------|-------------|------------|-------------|------------|-------------|------------|
|            |             |            |             | E3<br>147  | F3<br>156  | F#3<br>165  | G3<br>175  | G#3<br>185  | A3<br>196  | A#3<br>208  | B3<br>220  |
| C4<br>233  | C#4<br>247  | D4<br>262  | D#4<br>277  | E4<br>294  | F4<br>331  | F#4<br>330  | G4<br>349  | G#4<br>370  | A4<br>392  | A#4<br>415  | B4<br>440  |
| C5<br>466  | C#5<br>494  | D5<br>523  | D#5<br>554  | E5<br>587  | F5<br>622  | F#5<br>659  | G5<br>698  | G#5<br>740  | A5<br>784  | A#5<br>831  | B5<br>880  |
| C6<br>932  | C#6<br>988  | D6<br>1047 | D#6<br>1109 | E6<br>1175 | F6<br>1245 | F#6<br>1319 | G6<br>1397 | G#6<br>1480 | A6<br>1568 | A#6<br>1661 | B6<br>1760 |
| C7<br>1865 | C#7<br>1976 |            |             |            |            |             |            |             |            |             |            |

**Πίνακας 2: Τα pitch με τις αντίστοιχες θεωρητικές τιμές συχνότητας σε Hz της συνολικής έκτασης του κλαρινέτου σε Σι b / B b .**

### 4.1.2 Συλλογή πραγματικών δεδομένων

Κατά τη συλλογή των πρακτικά παρατηρούμενων τιμών συχνότητας με χρήση του praat χρειάστηκε να το παραμετροποιήσουμε κατάλληλα έτσι ώστε να λάβουμε σωστά και ακριβή αποτελέσματα. Θέσαμε λοιπόν το **εύρος του pitch** σε **140 – 2000 Hz** έτσι ώστε να μπορούμε να έχουμε μία καλή εκτίμηση όλων των τιμών του Πίνακα 2.

Επιπλέον, η τιμή της παραμέτρου **octave cost** επιλέχθηκε ίση με **0.06**. Η παράμετρος **octave cost**, όπως αναφέρεται κατά την παρουσίαση του ακριβούς αλγορίθμου βάσει του οποίου γίνεται η εκτίμηση της θεμελιώδους συχνότητας από το praat [Praat03], αποτελεί μία παράμετρο η οποία ευνοεί τις υψηλότερες θεμελιώδεις συχνότητες. Σύμφωνα με τον αλγόριθμο που εφαρμόζεται, η βέλτιστη υποψήφια τιμή της περιόδου ενός ακουστικού σήματος μπορεί να υπολογιστεί από τη θέση του μεγίστου της συνάρτησης αυτοσυσχέτισης (autocorrelation) αυτού και ο βαθμός της περιοδικότητας από το σχετικό ύψος του μεγίστου. Το ύψος (local strength) καθεμιάς από τις υποψήφιες θέσεις μεγίστου που εκτιμώνται υπολογίζεται βάσει ενός μαθηματικού τύπου. Η παράμετρος octave cost αποτελεί μία εκ των παραμέτρων του μαθηματικού αυτού τύπου [Praat03].

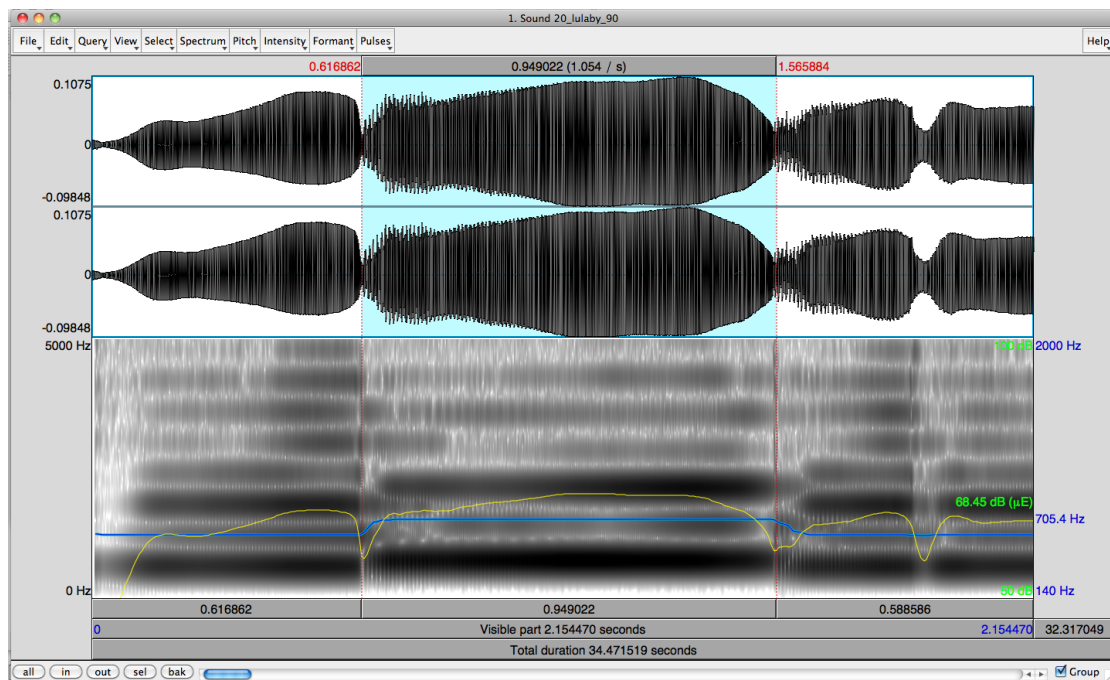
Μετά από παρατήρηση των αποτελεσμάτων για διαφορετικές τιμές καταλήξαμε στην επιλογή της τιμής 0.06 καθώς με την default τιμή (0.01) για κάποιες νότες η εκτίμηση του pitch διέφερε από το αντιλαμβανόμενο κατά μία οκτάβα ακριβώς, δηλαδή η εκτιμώμενη αριθμητική τιμή της θεμελιώδους συχνότητας ήταν υποδιπλάσια, το όνομα της νότας το ίδιο, ενώ η οκτάβα στην οποία σημειώνεται ήταν μία φορά χαμηλότερη. Μετά από δοκιμές καταλήξαμε στο συμπέρασμα πως η αύξηση της τιμής οδηγεί γενικά σε καλύτερα αποτελέσματα (δοκιμάστηκαν οι τιμές 0.06, 0.08, 0.1 και 0.15), ωστόσο σημαντική αύξηση της τιμής (για παράδειγμα, μεγαλύτερης ή ίσης από 0.1) σε ορισμένες περιπτώσεις ηχογραφήσεων οδηγεί σε μεταβολή της διαφορετικά ορθής εκτίμησης του pitch. Υπήρξε παράδειγμα ηχογράφησης για την οποία μία τέτοια αύξηση της τιμής octave cost είχε ως αποτέλεσμα να εκτιμάται pitch του οποίου η αριθμητική τιμή της συχνότητας ήταν 3 φορές μεγαλύτερη της επιθυμητής και για τιμές μικρότερες του 0.1 σύμφωνης με το αντιλαμβανόμενο. Μία τιμή για την οποία δεν παρατηρήθηκαν τέτοια προβλήματα ενώ ταυτόχρονα οδηγεί σε ικανοποιητική βελτίωση των διαφορετικά λανθασμένων εκτιμήσεων είναι η τιμή 0.06, η οποία και επιλέχθηκε.

Η εκτίμηση του pitch πραγματοποιείται με βραχυπρόθεσμη (short-term) ανάλυση η οποία πραγματοποιείται για καθένα από τα αποσπάσματα (frames) τα οποία ανακτώνται από το σήμα με βήμα ίσο με την παράμετρο time step. Για τις απαιτήσεις της παρούσας εργασίας, το **βήμα δειγματοληψίας**, ή διαφορετικά το **εύρος του παραθύρου** που μελετάται είναι ίσο με την τιμή **0.01sec** (παράμετρος του praat **time step = 0.01** , επιλογή **fixed**).

Μετά τη συλλογή των στοιχείων αυτών χρησιμοποιήσαμε το πρόγραμμα στατιστικής ανάλυσης **SPSS** για την εξαγωγή βασικών στατιστικών παραμέτρων για κάθε pitch από τα υπό μελέτη βασιζόμενοι σε ξεχωριστά στιγμιότυπα ηχογραφήσεων, αφού πρώτα διαγράψαμε τις ακραίες τιμές. Θεωρήσαμε ικανοποιητικά αντιπροσωπευτικές τη **μέγιστη** και **κεντρική** τιμή του συνόλου των **κεντρικών τιμών** των διαφορετικών στιγμιότυπων της εκάστοτε νότας.

### 4.1.3 Παράδειγμα διαδικασίας

Στο ίδιο μουσικό κομμάτι το οποίο παρουσιάστηκε ανωτέρω (Εικόνα 6, §3.1), με χρήση του praat λαμβάνουμε για το μαρκαρισμένο στιγμιότυπο της αναπαράστασης του σήματος (γαλάζιο τμήμα κυματομορφής), το οποίο αντιστοιχεί σε μία δεδομένη νότα, τη λίστα από τα εκτιμώμενα pitch με βήμα 0.01sec (επιλογή του praat **pitch listing**).



**Εικόνα 10:** Παράδειγμα συλλογής εκτιμώμενων pitch για τη νότα G5 του μουσικού κομματιού lullaby από κλαρινέτο. **Περιγραφή:** Η λίστα τιμών που ακολουθεί αντιστοιχεί στο τμήμα που έχει επιλεγεί (η κυματομορφή που περιλαμβάνεται στο γαλάζιο φόντο).

Λίστα τιμών χρόνου (time\_s) και εκτιμώμενης θεμελιώδους συχνότητας (F0\_Hz):

| Time_s   | F0_Hz      | ..                  |
|----------|------------|---------------------|
| 0.622596 | 614.171568 | 1.152596 708.285499 |
| 0.632596 | 652.345653 | 1.162596 708.314982 |
| 0.642596 | 682.198499 | 1.172596 708.236651 |
| 0.652596 | 692.795342 | 1.182596 708.189087 |
| 0.662596 | 698.771669 | 1.192596 708.080245 |
| 0.672596 | 700.780849 | 1.202596 708.082163 |
| 0.682596 | 703.060178 | 1.212596 708.065451 |
| 0.692596 | 705.522998 | 1.222596 707.985053 |
| 0.702596 | 708.000824 | 1.232596 707.949940 |
| 0.712596 | 706.508065 | 1.242596 707.950118 |
| 0.722596 | 706.419228 | 1.252596 707.935169 |
| 0.732596 | 706.880481 | 1.262596 707.911338 |
| 0.742596 | 707.413302 | 1.272596 707.865070 |
| 0.752596 | 706.761175 | 1.282596 707.811602 |
| 0.762596 | 707.292216 | 1.292596 707.730340 |
| 0.772596 | 707.704641 | 1.302596 707.701618 |
| 0.782596 | 708.078169 | 1.312596 707.645036 |
| 0.792596 | 707.740230 | 1.322596 707.627376 |
| 0.802596 | 707.944910 | 1.332596 707.585382 |
| 0.812596 | 708.297545 | 1.342596 707.572454 |
| 0.822596 | 708.458205 | 1.352596 707.515187 |
| 0.832596 | 708.731593 | 1.362596 707.457725 |
| 0.842596 | 708.623734 | 1.372596 707.364429 |
| 0.852596 | 708.591704 | 1.382596 707.292526 |
| 0.862596 | 708.507353 | 1.392596 707.145179 |
| 0.872596 | 708.208405 | 1.402596 707.010934 |
| 0.882596 | 708.017188 | 1.412596 706.778537 |
| 0.892596 | 707.817619 | 1.422596 706.611823 |
| 0.902596 | 708.044518 | 1.432596 706.447519 |
| 0.912596 | 707.975490 | 1.442596 706.482007 |
| 0.922596 | 708.491034 | 1.452596 706.509177 |
| 0.932596 | 708.775350 | 1.462596 706.590972 |
| 0.942596 | 708.864654 | 1.472596 706.683431 |
| 0.952596 | 708.667325 | 1.482596 706.771010 |
| 0.962596 | 708.596370 | 1.492596 706.883135 |
| 0.972596 | 708.348489 | 1.502596 706.912739 |
| 0.982596 | 708.342858 | 1.512596 706.981249 |
| 0.992596 | 708.270121 | 1.522596 706.907116 |
| 1.002596 | 708.400213 | 1.532596 706.678629 |
| 1.012596 | 708.523500 | 1.542596 705.961256 |
| 1.022596 | 708.574596 | 1.552596 704.653527 |
| 1.032596 | 708.712736 | 1.562596 700.052341 |
| 1.042596 | 708.690383 |                     |
| 1.052596 | 708.670261 |                     |
| 1.062596 | 708.599629 |                     |
| 1.072596 | 708.633439 |                     |
| 1.082596 | 708.623259 |                     |
| 1.092596 | 708.554264 |                     |
| 1.102596 | 708.528113 |                     |
| 1.112596 | 708.515750 |                     |
| 1.122596 | 708.485878 |                     |
| 1.132596 | 708.423918 |                     |
| 1.142596 | 708.389307 |                     |

Οι πρώτες επτά και η τελευταία τιμή, οι οποίες σημειώνονται με πορτοκαλί χρώμα στη λίστα, είναι κατανοητό πως δεν αποτελούν αντιπροσωπευτικές τιμές του δείγματος και έτσι μπορούν να διαγραφούν. Από αυτές η τιμή 703.060178 αποτελεί outlier, τιμή η οποία παρεκκλίνει από τα υπόλοιπα μέλη του δείγματος, ενώ οι υπόλοιπες αποτελούν extreme τιμές, τιμές οι οποίες παρεκκλίνουν αρκετά από τη διάμεσο του συνόλου τιμών, σύμφωνα και με την στατιστική ανάλυση με χρήση του SPSS (boxplot).

Η κεντρική τιμή μετά την απομόνωση των επιθυμητών τιμών του δείγματος για το pitch G5 υπολογίζεται ίση με 707.9501Hz, τιμή η οποία διαφέρει κατά έναν μικρό αλλά όχι ασήμαντο όρο (περίπου 1.4%) από την ιδανική απόκριση (698Hz). Η σημαντικότερη παρατήρηση ωστόσο είναι πως η διαφορά της από την ιδανική τιμής της επόμενης νότας είναι περίπου τριπλάσια αυτής (η επόμενη νότα είναι η G#5 = 740Hz και η απόκλιση της από αυτή είναι περίπου ίση με 4.3%).

Για την ίδια νότα εξετάστηκαν και αρκετά ακόμη στιγμιότυπα από διαφορετικές ηχογραφήσεις και υπολογίστηκαν οι κεντρικές τιμές αυτών. Από το σύνολό τους υπολογίστηκαν η μέγιστη και η κεντρική τιμή που προέκυψαν. Η ίδια διαδικασία πραγματοποιήθηκε για κάποια ακόμη pitch (καλύφθηκε μιά οκτάβα από μη αλλοιωμένες νότες καθώς οι αλλοιωμένες νότες εμφανίζονται σπάνια ή καθόλου στις ηχογραφήσεις που εξετάστηκαν και δεν μπορούσαν να αποτελέσουν αντιπροσωπευτικό δείγμα).

Βάσει των δεδομένων που συλλέχθηκαν με τον τρόπο αυτό και της μαθηματικής σχέσης που συνδέει τα pitch :

$$f = 2^{n/12} f_{ref} \quad \text{όπου } f_{ref} \text{ η συχνότητα αναφοράς και } n \text{ τα ημιτόνια}^7 \text{ μεταξύ των} \\ \text{συχνοτήτων } f_{ref} \text{ και της επιθυμητής συχνότητας } f$$

συντάχθηκε ο παρακάτω πίνακας:

| pitch | Θεωρητική τιμή | Μέγιστη τιμή | Κεντρική τιμή |
|-------|----------------|--------------|---------------|
| E3    | 147            | 149.2263     | 147.99495     |
| F3    | 156            | 158.25       | 157.99495     |
| F#3   | 165            | 167.6335481  | 166.52105     |
| G3    | 175            | 178.0897     | 176.5029      |
| G#3   | 185            | 189.8309635  | 188.6794646   |
| A3    | 196            | 201.1189     | 199.2082      |
| A#3   | 208            | 213.0780521  | 210.37925     |
| B3    | 220            | 222.15785    | 222.4466      |
| C4    | 233            | 237.8895     | 235.653       |
| C#4   | 247            | 249.8826     | 249.4769      |
| D4    | 262            | 266.5421     | 264.6688      |
| D#4   | 277            | 282.391518   | 281.7017427   |
| E4    | 294            | 298.4526     | 295.9899      |
| F4    | 311            | 316.45       | 314.2306      |
| F#4   | 330            | 335.2670962  | 333.0421      |
| G4    | 349            | 356.1794     | 353.0058      |
| G#4   | 370            | 379.661927   | 377.3589293   |

<sup>7</sup> Το μισό της απόστασης μεταξύ δυο διαδοχικών τόνων της μουσικής κλίμακας. Πρακτικά, στον Πίνακα 2, κάθε νότα απέχει από την προηγούμενη και την επόμενη κατά ένα ημιτόνιο.

|     |      |             |             |
|-----|------|-------------|-------------|
| A4  | 392  | 402.2378    | 398.4164    |
| A#4 | 415  | 426.1561043 | 420.7585    |
| B4  | 440  | 444.8932    | 444.3157    |
| C5  | 466  | 475.779     | 471.306     |
| C#5 | 494  | 499.7652    | 498.9538    |
| D5  | 524  | 533.0842    | 529.3376    |
| D#5 | 554  | 564.7830361 | 563.4034854 |
| E5  | 588  | 596.9052    | 591.9798    |
| F5  | 622  | 632.9       | 628.4612    |
| F#5 | 660  | 670.5341924 | 666.0842    |
| G5  | 698  | 712.3588    | 706.0116    |
| G#5 | 740  | 759.323854  | 754.7178585 |
| A5  | 784  | 804.4756    | 796.8328    |
| A#5 | 830  | 852.3122085 | 841.517     |
| B5  | 880  | 889.7864    | 888.6314    |
| C6  | 932  | 951.558     | 942.612     |
| C#6 | 988  | 999.5304    | 997.9076    |
| D6  | 1048 | 1066.1684   | 1058.6752   |
| D#6 | 1108 | 1129.566072 | 1126.806971 |
| E6  | 1176 | 1193.8104   | 1183.9596   |
| F6  | 1244 | 1265.8      | 1256.9224   |
| F#6 | 1320 | 1341.068385 | 1332.1684   |
| G6  | 1396 | 1424.7176   | 1412.0232   |
| G#6 | 1480 | 1518.647708 | 1509.435717 |
| A6  | 1568 | 1608.9512   | 1593.6656   |
| A#6 | 1660 | 1704.624417 | 1683.034    |
| B6  | 1760 | 1779.5728   | 1777.2628   |
| C7  | 1864 | 1903.116    | 1885.224    |
| C#7 | 1976 | 1999.0608   | 1995.8152   |

**Πίνακας 3: Θεωρητικές και πραγματικές τιμές των θεμελιωδών συχνοτήτων για όλα τα pitch που μπορούν να αναπαραχθούν από το κλαρινέτο σε Σι b /Bb . Περιγραφή:** Η πρώτη στήλη περιέχει το αναγνωριστικό όνομα του pitch, η δεύτερη στήλη τη θεωρητική τιμή της θεμελιώδους συχνότητας που αντιστοιχεί σε αυτό, και οι επόμενες δύο στήλες τη μέγιστη και κεντρική, αντίστοιχα, τιμή των κεντρικών τιμών όλων των στιγμιοτύπων της νότας, μετά τον αποκλεισμό των ακραίων τιμών. Οι τιμές των οποίων το φόντο έχει αφεθεί λευκό έχουν προκύψει από παρατήρηση και όλες οι υπόλοιπες έχουν υπολογιστεί βάσει του δοθέντος τύπου.

#### 4.1.4 Συμπεράσματα

Τα αποτελέσματα είναι ικανοποιητικά. Παρατηρούμε πως οι πραγματικές τιμές έχουν μία αυξητική τάση απομάκρυνσης από τις θεωρητικές τιμές καθώς πηγαίνουμε από χαμηλές σε υψηλότερες συχνότητες, ανάλογη όμως με την κατανομή και των θεωρητικών τιμών (το βήμα μεγαλώνει καθώς οι συχνότητες μεγαλώνουν). Οι αποκλίσεις από τις ιδανικές τιμές κυμαίνονται μεταξύ των τιμών 0.98% και 2.69%. Οι τιμές ανά κλάση παρουσιάζουν μία ομοιομορφία ενώ ταυτόχρονα δεν παρατηρούνται επικαλύψεις μεταξύ διαφορετικών κλάσεων.

## 4.2 Στάδιο δεύτερο

### 4.2.1 Εφαρμογή 3-Nearest-Neighbors

Στην προηγούμενη παράγραφο (4.1) παρουσιάσαμε τον τρόπο με τον οποίο μπορούμε να λάβουμε μία εκτίμηση των θεμελιωδών συχνοτήτων που αντιστοιχούν στο αντιλαμβανόμενο pitch ενός μουσικού σήματος. Επαναλαμβάνοντας τη διαδικασία αυτή συντάχθηκε τελικά ο Πίνακας 3, ο οποίος περιλαμβάνει τρεις διαφορετικές τιμές συχνοτήτων – τη θεωρητική, τη μέγιστη και την κεντρική τιμή – αντιπροσωπευτικές για καθένα από τα pitch που μπορεί να αναπαράγει το κλαρινέτο σε Σι b /B b .

Σε αυτό το σημείο λοιπόν είμαστε σε θέση εφαρμόζοντας τον κανόνα απόφασης k-Nearest-Neighbours, θέτοντας  $k=3$ , να εκτιμήσουμε το αναγνωριστικό όνομα του pitch στο οποίο κατηγοριοποιείται μία οποιαδήποτε συχνότητα μεταξύ των τιμών 140-2000Hz. Ο Πίνακας 3 θα αποτελέσει τα δεδομένα εκπαίδευσης για το σκοπό αυτό. Το αναγνωριστικό όνομα των pitch θα αντιστοιχεί λοιπόν στο αναγνωριστικό όνομα της εκάστοτε κλάσης ( $n$ ), και το σύνολο των τιμών στα  $y_j$ , σύμφωνα με τον αλγόριθμο που παρουσιάστηκε στην παράγραφο 3.2.2. Το  $k$  λαμβάνει την τιμή 3 καθώς με τα δεδομένα εκπαίδευσης που διαθέτουμε η πλειοψηφία με την οποία θα εκλεγεί το  $n$  δεν μπορεί να είναι μεγαλύτερη αυτού του αριθμού.

Η υλοποίηση του αναγνωριστή των pitch του κλαρινέτου υλοποιήθηκε σε γλώσσα προγραμματισμού C++ και ο σχετικός κώδικας συνοδευόμενος από τα απαραίτητα αρχεία εισόδου παρουσιάζονται στο Παράρτημα B, B1. Σημειώνεται πως η παράμετρος  $k$  του κανόνα απόφασης δίνεται ως είσοδος του συστήματος και σε περίπτωση ύπαρξης διαφορετικών δεδομένων εκπαίδευσης μπορεί να εφαρμοστεί για οποιοδήποτε άλλο  $k$ .

Το πρόγραμμα δέχεται ως εισόδους δύο αρχεία με τις πληροφορίες σχετικά με τα δεδομένα εκπαίδευσης, το ένα αρχείο (names.in) περιλαμβάνει τα αναγνωριστικά ονόματα των pitch, συνοδευόμενο από ένα αλφαριθμητικό id για το διαχωρισμό της θεωρητικής (a), κεντρικής (b) και μέγιστης (c) τιμής από την οποία απέχει το ελάχιστο (για παράδειγμα, C4 9c), και το άλλο (data.in) τη λίστα με τις τιμές θεμελιωδών συχνοτήτων όπως ορίζονται στον Πίνακα 3.

Η εκτέλεσή του δημιουργεί ένα αρχείο εξόδου, με όνομα output.txt, το οποίο περιλαμβάνει τα ονόματα των pitch στα οποία οι τιμές του δείγματος που δίνονται ως είσοδος (test.in) κατηγοριοποιήθηκαν. Το όνομα του pitch συνοδεύεται από το αλφαριθμητικό id του στοιχείου από το οποίο η τιμή του δείγματος απέχει την μικρότερη απόσταση (το id είναι αυτό του αρχείου names.in).



#### 4.2.2 Παράδειγμα εκτέλεσης

Ακολουθεί ένα απόσπασμα του αρχείου εξόδου της εφαρμογής του κώδικα 3-Nearest-Neighbors για τη δεύτερη νότα (G5) του κομματιού lullaby από κλαρινέτο (Εικόνα 10), η οποία ακολουθεί αλλά και προηγείται της F5. Το πρώτο στοιχείο κάθε γραμμής είναι η συχνότητα εισόδου (test.in) και δίπλα της δίνεται το όνομα του pitch συνδεδεμένο από το id (θεωρητική (a), κεντρική (b) και μέγιστης τιμή (c)) το οποίο επιστρέφεται ως το κοντινότερο μετά την εφαρμογή του αλγορίθμου 3-Nearest-Neighbors (output.txt).

Για παράδειγμα, η τιμή 'F5 (26c)' που επιστρέφεται για την τιμή θεμελιώδους συχνότητας 633.56 (πρώτη γραμμή στη λίστα που ακολουθεί) ορίζει ότι το pitch στο οποίο αντιστοιχείται η μαθηματική ποσότητα είναι το F5 και η τιμή εκπαίδευσης από την οποία απέχει το ελάχιστο είναι η '26c', δηλαδή η 26η νότα από τις 46 που μπορούν να αναπαραχθούν (η F5) και η τιμή της συχνότητας από την οποία απέχει το ελάχιστο είναι αυτή της μέγιστης (c) βάσει του Πίνακα 3 (η οποία είναι ίση με 632.9Hz)

|                  |                 |                  |
|------------------|-----------------|------------------|
| ...              | 708.68 G5 (28b) | 706.86 G5 (28b)  |
| 633.56 F5 (26c)  | 708.02 G5 (28b) | 706.98 G5 (28b)  |
| 656.90 F#5 (27a) | 708.69 G5 (28b) | 706.97 G5 (28b)  |
| 677.42 F#5 (27c) | 708.03 G5 (28b) | 706.89 G5 (28b)  |
| 685.40 F#5 (27c) | 707.98 G5 (28b) | 706.79 G5 (28b)  |
| 695.25 G5 (28a)  | 707.99 G5 (28b) | 706.62 G5 (28b)  |
| 695.02 G5 (28a)  | 707.95 G5 (28b) | 706.28 G5 (28b)  |
| 699.51 G5 (28a)  | 707.94 G5 (28b) | 705.71 G5 (28b)  |
| 702.06 G5 (28b)  | 707.96 G5 (28b) | 704.77 G5 (28b)  |
| 700.53 G5 (28a)  | 707.93 G5 (28b) | 705.93 G5 (28b)  |
| 704.73 G5 (28b)  | 707.92 G5 (28b) | 703.63 G5 (28b)  |
| 702.71 G5 (28b)  | 707.91 G5 (28b) | 692.96 G5 (28a)  |
| 706.42 G5 (28b)  | 707.89 G5 (28b) | 681.07 F#5 (27c) |
| 705.76 G5 (28b)  | 707.87 G5 (28b) | 680.06 F#5 (27c) |
| 707.16 G5 (28b)  | 707.83 G5 (28b) | 674.42 F#5 (27c) |
| 708.56 G5 (28b)  | 707.81 G5 (28b) | 670.54 F#5 (27c) |
| 704.76 G5 (28b)  | 707.78 G5 (28b) | 659.09 F#5 (27a) |
| 707.75 G5 (28b)  | 707.73 G5 (28b) | 637.62 F5 (26c)  |
| 704.40 G5 (28b)  | 707.73 G5 (28b) | 624.78 F5 (26a)  |
| 707.12 G5 (28b)  | 707.69 G5 (28b) | 622.50 F5 (26a)  |
| 705.26 G5 (28b)  | 707.66 G5 (28b) | ...              |
| 706.28 G5 (28b)  | 707.65 G5 (28b) |                  |
| 707.08 G5 (28b)  | 707.63 G5 (28b) |                  |
| 705.27 G5 (28b)  | 707.64 G5 (28b) |                  |
| 707.03 G5 (28b)  | 707.61 G5 (28b) |                  |
| 705.71 G5 (28b)  | 707.58 G5 (28b) |                  |
| 707.34 G5 (28b)  | 707.57 G5 (28b) |                  |
| 707.51 G5 (28b)  | 707.55 G5 (28b) |                  |
| 707.39 G5 (28b)  | 707.51 G5 (28b) |                  |
| 708.31 G5 (28b)  | 707.48 G5 (28b) |                  |
| 707.98 G5 (28b)  | 707.46 G5 (28b) |                  |
| 708.79 G5 (28b)  | 707.41 G5 (28b) |                  |
| 707.93 G5 (28b)  | 707.36 G5 (28b) |                  |
| 708.12 G5 (28b)  | 707.31 G5 (28b) |                  |
| 708.79 G5 (28b)  | 707.27 G5 (28b) |                  |
| 708.10 G5 (28b)  | 707.20 G5 (28b) |                  |
| 709.00 G5 (28b)  | 707.11 G5 (28b) |                  |
| 708.24 G5 (28b)  | 707.07 G5 (28b) |                  |
| 708.34 G5 (28b)  | 706.94 G5 (28b) |                  |
| 708.66 G5 (28b)  | 706.84 G5 (28b) |                  |
| 707.69 G5 (28b)  | 706.70 G5 (28b) |                  |
| 708.62 G5 (28b)  | 706.64 G5 (28b) |                  |
| 707.91 G5 (28b)  | 706.52 G5 (28b) |                  |
| 708.29 G5 (28b)  | 706.45 G5 (28b) |                  |
| 708.24 G5 (28b)  | 706.47 G5 (28b) |                  |
| 708.06 G5 (28b)  | 706.48 G5 (28b) |                  |
| 708.76 G5 (28b)  | 706.48 G5 (28b) |                  |
| 707.87 G5 (28b)  | 706.52 G5 (28b) |                  |
| 708.67 G5 (28b)  | 706.56 G5 (28b) |                  |
| 708.09 G5 (28b)  | 706.61 G5 (28b) |                  |
| 708.04 G5 (28b)  | 706.64 G5 (28b) |                  |
| 706.91 G5 (28b)  | 706.71 G5 (28b) |                  |
| 706.91 G5 (28b)  | 706.76 G5 (28b) |                  |
| 706.93 G5 (28b)  | 706.78 G5 (28b) |                  |

### 4.2.3 Συμπεράσματα

Τα αποτελέσματα της εφαρμογής της μεθόδου του κοντινότερου γείτονα σε διάφορα δείγματα οδηγούν στο συμπέρασμα πως διαθέτουμε μία πολύ καλή εκτίμηση του **pitch** της νότας που αναπαράγεται κάθε χρονική στιγμή καθώς και μία ικανοποιητική προσέγγιση της **διάρκειάς** της (Η διάρκεια μιάς νότας μπορεί να υπολογιστεί από το συνολικό αριθμό εμφάνισης του ίδιου pitch στο αρχείο εξόδου πολλαπλασιασμένο με το παράθυρο/frame στο οποίο εκτιμάται η θεμελιώδης συχνότητα, δηλαδή 0.01sec. Κατά τη μετάβαση από μία νότα σε άλλη παρατηρούνται μεταβατικά φαινόμενα και γιαυτό αναφέρεται πως ο υπολογισμός της διάρκειας μπορεί να εκτιμηθεί κατά *προσέγγιση*).

Σε διαδοχικές παρατηρήσεις του ίδιου pitch το/τα στοιχείο/α που έπεται του πρώτου εκτιμάται στις περισσότερες περιπτώσεις με ακρίβεια 100% ενώ λόγω μεταβάσεων μεταξύ διαφορετικών pitch (και παρουσίας παύσεων) το ποσοστό επιτυχίας κυμαίνεται μεταξύ 80% και 97%. Τα χαμηλότερα ποσοστά, τιμές κοντινότερες στην κρίσιμη τιμή 50%, παρατηρούνται μόνο στις περιπτώσεις όπου η εκτίμηση της θεμελιώδους συχνότητας έχει προκύψει υποδιπλάσια της αντιλαμβανόμενης για κάποιες τιμές του συνόλου που αντιστοιχεί σε μία νότα (για περισσότερες πληροφορίες ο αναγνώστης παραπέμπεται στην ενότητα 4.1.2, στη σχετική με την παράμετρο octave cost παράγραφο). Λίγες είναι οι περιπτώσεις όπου το ποσοστό είναι χαμηλότερο της κρίσιμης αυτής τιμής (50%) και η νότα εμφανίζεται με διαφορετικό pitch από το αναμενόμενο, αυτό που αντιστοιχεί στην υποδιπλάσια τιμή της θεμελιώδους συχνότητας (για παράδειγμα G3 αντί G4, δηλαδή εκτιμάται σωστά το όνομα της νότας αλλά όχι η οκτάβα στην οποία σημειώνεται).

Αξίζει να σημειωθεί πως η τιμή της θεωρητικής συχνότητας ως κοντινότερης εμφανίζεται σπάνια, και αυτό συμβαίνει στα σημεία μετάβασης από μία νότα σε μία άλλη διαφορετικού pitch. Το γεγονός αυτό επιβεβαιώνει την αναγκαιότητα ύπαρξης εκπαιδευτικών δεδομένων και, ή ίσως αποκλειστικά, από τον πραγματικό κόσμο για μεγαλύτερη ακρίβεια των υπολογισμών.

## 4.3 Στάδιο τρίτο

### 4.3.1 Σχεδίαση συστήματος

Η αυτόματη χρονική κατανομή του συνόλου από νότες σε ένα μουσικό κομμάτι απαιτεί την παρακολούθηση του ρυθμού η οποία πρακτικά δε διατηρείται απόλυτα σταθερή. Η έναρξη και λήξη μιάς νότας αναφέρεται στη βιβλιογραφία με τον όρο *onset*. Τα χαρακτηριστικά γνώρισμα της μεθόδου ανίχνευσης των αρχών των διακριτικών γεγονότων στα ακουστικά σήματα (onset detection) όμως δεν μπορούν να οριστούν σαφώς στην περίπτωση των πνευστών οργάνων όπως έχει ήδη αναφερθεί (§3.1). Οι μελωδίες που εξετάστηκαν αντλήθηκαν από τη βάση μουσικών κομματιών του Vemus, η ομάδα-στόχος χρηστών του οποίου είναι σπουδαστές πνευστών οργάνων κατά τη διάρκεια των πρώτων 4-5 ετών της μουσικής τους εκπαίδευσης. Έτσι είναι θα λέγαμε απλές και αυστηρού ύφους, σε αντιδιαστολή με την ελευθερία και την εκφραστικότητα ως χαρακτηριστικό γνώρισμα μιάς πιο προχωρημένης ερμηνείας. Και αυτό ακριβώς είναι το γεγονός το οποίο εκμεταλλευόμαστε κατά τη θεώρηση σταθερού και γνωστού από πριν ρυθμού της ηχογράφησης. Έτσι, στα πλαίσια της εργασίας θεωρούμε το ρυθμό (δοθείσα παράμετρος **tempo** η οποία ορίζεται στον σχετικό κώδικα υλοποίησης του συστήματος που παρουσιάζεται στο Παράρτημα Β, Β2) γνωστό και σταθερό για το υπό εξέταση μουσικό κομμάτι.

Οι χρονικές αξίες/διάρκειες που εμφανίζονται στην Ευρωπαϊκή μουσική είναι οι ακόλουθες: 1/256°, 1/128°, 1/64°, 1/32°, 1/16°, ένα όγδοο, ένα τέταρτο, μισό, ολόκληρο, breve και long. Για λόγους απλότητας αλλά και για την αποφυγή λανθασμένων εκτιμήσεων θα θεωρήσουμε γνωστή την ελάχιστη διάρκεια (δοθείσα παράμετρος **min**: μέγιστη υποδιαίρεση της αξίας αναφοράς, συνήθως του ενός τετάρτου) η οποία εμφανίζεται στο υπό μελέτη μουσικό κομμάτι ή φράση. Με τον τρόπο αυτό είμαστε σε θέση να χωρίσουμε μία ηχογράφιση σε ομάδες τιμών που αντιπροσωπεύουν μία διάρκεια ίση με την ελάχιστη, για τις οποίες μπορούμε να συμπεράνουμε το επικρατέστερο pitch το οποίο εμφανίζεται, βάσει πλειοψηφίας (majority voting rule).

Δεδομένες θεωρούμε επίσης και κάποιες τιμές οι οποίες θα μας επιτρέψουν την ορθή οπτική και ακουστική αναπαράσταση της παρτιτούρας και οι οποίες συνδέονται με το πρότυπο MusicXml το οποίο παρουσιάστηκε στην προηγούμενη ενότητα (§3.3). Αυτές οι τιμές αντιστοιχούν στις παρακάτω παραμέτρους (οι οποίες ορίζονται στον σχετικό κώδικα του Παραρτήματος Β, Β2) :

**divisions**: Το στοιχείο divisions παρέχει τη μονάδα του μέτρου για το στοιχείο duration με την έννοια των υποδιαίρεσεων ανά τέταρτο

**beats**: Αριθμητής του μουσικού ρυθμού

**beatType**: Παρονομαστής του μουσικού ρυθμού

**sign**: Το μουσικό κλειδί

**line** : Η γραμμή του πενταγράμμου στην οποία σημειώνεται το μουσικό κλειδί

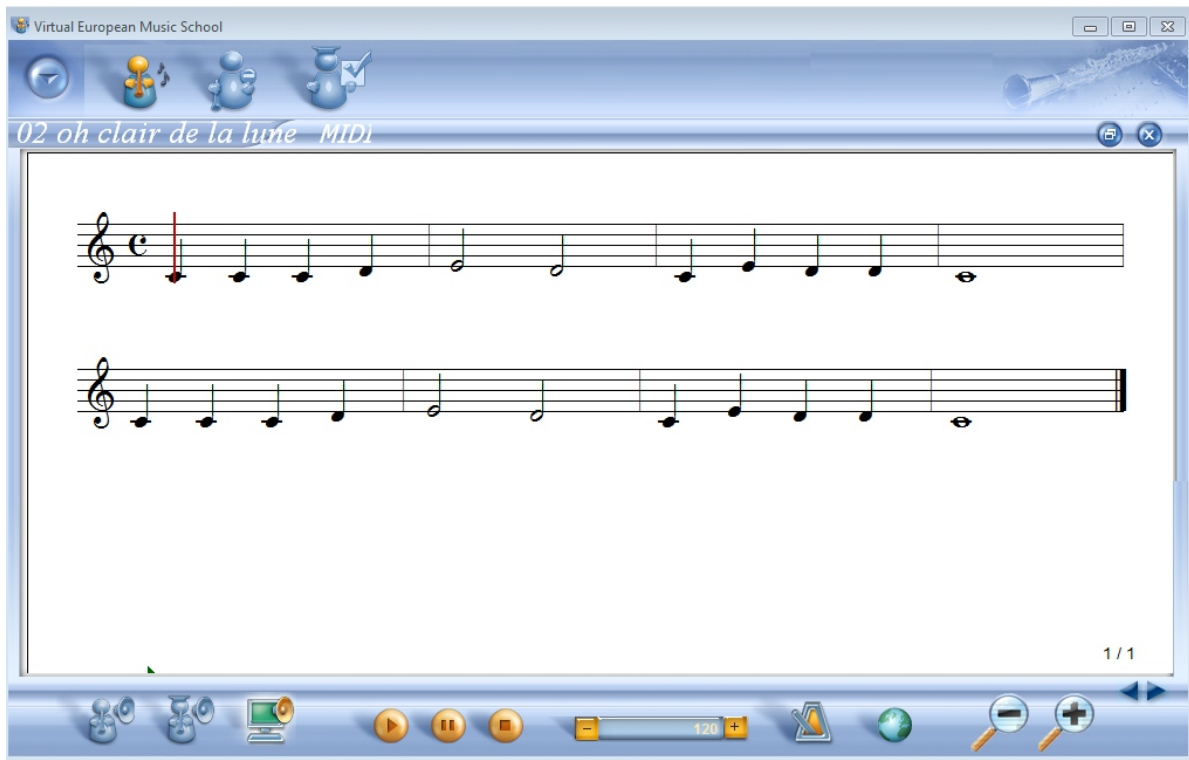
**instrument**: Το μουσικό όργανο (για παράδειγμα το κλαρινέτο)

**program**: Midi πληροφορία του οργάνου

ενώ προσθέτουμε μία ακόμη παράμετρο, τη χρονική στιγμή του μέτρου κατά την οποία το στιγμιότυπο ξεκινά (**moment**). Η παράμετρος αυτή είναι ιδιαίτερα σημαντική για μουσικές εκτελέσεις οι οποίες ξεκινούν με ελλιπές μέτρο, δηλαδή μέτρο το οποίο δεν ικανοποιεί την αναμενόμενη διάρκεια.

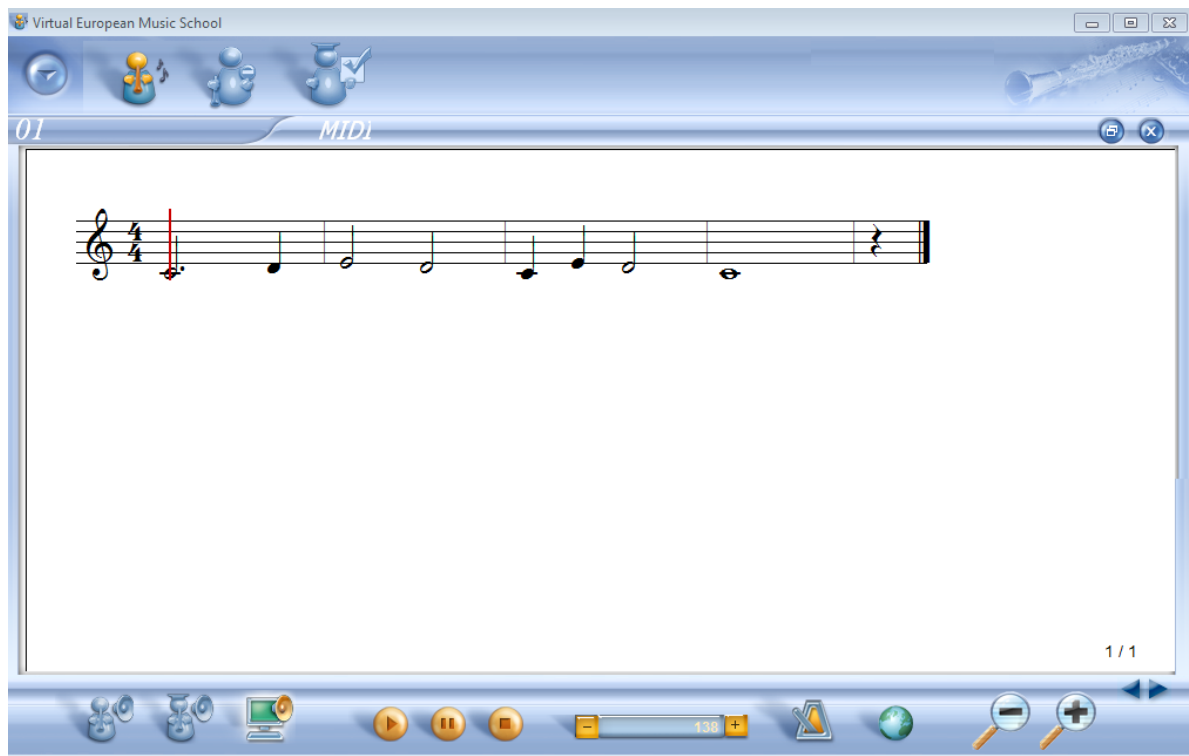
Στο στάδιο αυτό και για να γίνουν κατανοητά τα όσα έχουν περιγραφεί θα δώσουμε ένα παράδειγμα του τι είμαστε σε θέση να παράγουμε.

Έστω ότι θέλουμε να καταγράψουμε την παρτιτούρα η οποία αντιστοιχεί στην ηχογράφιση της πρώτης φράσης (πρώτη γραμμή) του παρακάτω μουσικού κομματιού (Εικόνα 11).



Εικόνα 11: Παρτιτούρα μουσικού κομματιού "oh clair de la lune" που περιέχεται στη βάση μουσικών κομματιών του Vemus

Θεωρώντας το τέμπο της ηχογράφησης σταθερό και ίσο με *138 τέταρτα το λεπτό*, όπως διαπιστώσαμε με τη χρήση χρονομέτρου, ομαδοποιούμε τις τιμές των pitch που η εκτέλεση του κώδικα της προηγούμενης ενότητας μας δίνει ανά τέταρτο, το οποίο αποτελεί την ελάχιστη διάρκεια η οποία εμφανίζεται σε ολόκληρο το μουσικό κομμάτι. Δουλεύουμε ως εξής: η έκφραση *138 τέταρτα το λεπτό* σημαίνει πως ένα τέταρτο έχει διάρκεια ίση με  $1/138$  λεπτά ή  $60/136$  δευτερόλεπτα. Η δειγματοληψία των pitch έχει οριστεί να πραγματοποιείται κάθε  $0,01$  δευτερόλεπτα (§4.1.2). Συνεπώς, ένα τέταρτο αντιστοιχεί σε  $(60/136)/0,01 \approx 43$  τιμές. Ομαδοποιούμε τις τιμές εξόδου του αλγορίθμου 3-Nearest-Neighbors ανά 43 και στη συνέχεια μπορούμε να αναπαραστήσουμε κάθε τέταρτο ξεχωριστά ή να ενώσουμε τα όμοια pitch και να αποτυπώσουμε τη συνολική τους διάρκεια όπως παρακάτω (Εικόνα 12).



Εικόνα 12: score.xml μετά την εκτέλεση του αρχικού κώδικα αυτόματης καταγραφής της πρώτης φράσης της ηχογράφησης του μουσικού κομματιού "oh clair de la lune"

Όπως γίνεται εύκολα αντιληπτό το αποτέλεσμα αυτό δεν είναι αντιπροσωπευτικό της ηχογράφησης αλλά παρέχει οπτικά την πληροφορία η οποία περιγράφει μόνο τα pitch που αναπαράγονται. Κρίνεται λοιπόν αναγκαία η αναζήτηση κάποιου χαρακτηριστικού το οποίο θα μας παρέχει την απαραίτητη πληροφορία βάσει της οποίας θα αποφασίζουμε πότε δυο νότες ελάχιστης διάρκειας του ίδιου pitch αποτελούν μία νότα της αμέσως μεγαλύτερης διάρκειας.

Αυτή η πληροφορία μπορεί να εξαχθεί μετά από μελέτη της περιβάλλουσας του μουσικού σήματος. Όπως έχει ήδη αναφερθεί (§3.1), η περιβάλλουσα του πλάτους μας παρέχει πληροφορίες σχετικά με την ένταση του ακουστικού σήματος (**intensity**), μία απαραίτητη παράμετρο για το διαχωρισμό των ξεχωριστών γεγονότων καθώς ανάμεσα σε δυο νότες του ίδιου pitch που ακολουθούν η μία την άλλη παρατηρείται μία ελάττωση στην ενέργεια του σήματος.

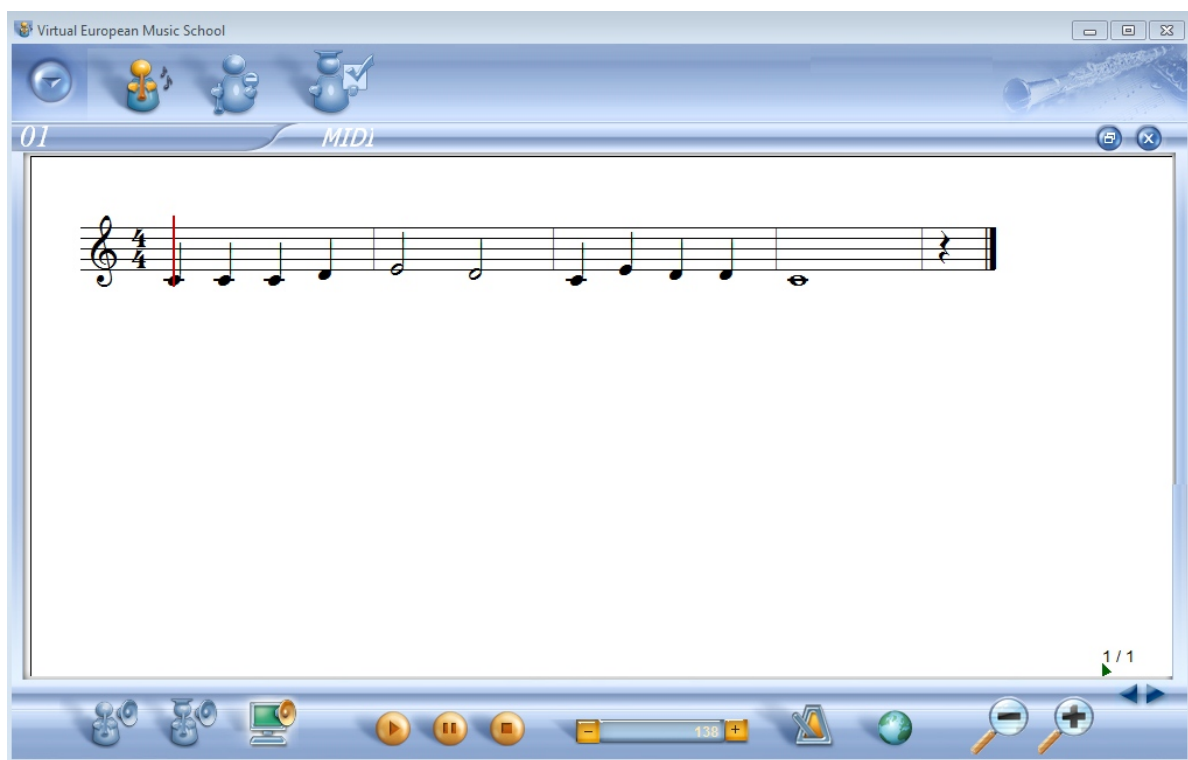
Το praat παρέχει ένα αρχείο το οποίο περιλαμβάνει τριάδες τιμών που αντιστοιχούν στα μεγέθη time, pitch και intensity με χρονικό βήμα το ίδιο με προηγουμένως, ήτοι 0.01 sec, μετά την εκτέλεση ενός δημοσιευμένου script [ScriptP].

Ελέγχοντας λοιπόν την πτώση που παρατηρείται στην περιβάλλουσα του πλάτους για το σύνολο των τιμών που αντιστοιχούν σε δυο συνεχόμενα και όμοια pitch ελάχιστης διάρκειας μπορούμε να αποφασίσουμε για το διαχωρισμό ή όχι αυτών και αντίστοιχα προχωρούμε στην σύμφωνη με το ακουστικό σήμα αναπαράσταση της διάρκειάς τους (Εικόνα 13).

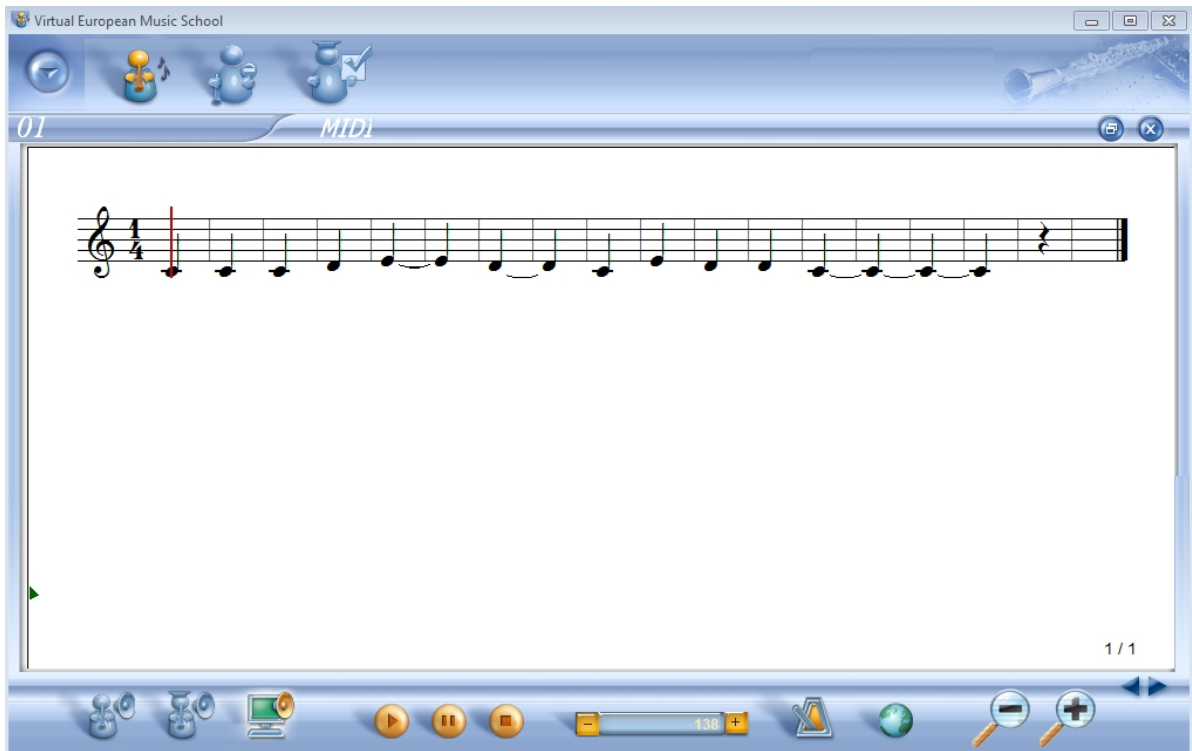
Ως το κατώφλι για την απόφαση του διαχωρισμού, μετά από παρακολούθηση των δεδομένων, επιλέξαμε τα 6db. Μία αύξηση των 6db αντιστοιχεί κατά προσέγγιση σε τετραπλασιασμό της ενέργειας ή διπλασιασμό του πλάτους του ακουστικού σήματος. Η τιμή επιλέχθηκε μετά από παρατήρηση της μέγιστης διακύμανσης της τιμής της έντασης κατά τη διάρκεια μίας νότας (η διαφορά μεταξύ μέγιστης και ελάχιστης τιμής έντασης ήταν μικρότερη των 4db για το σύνολο του δείγματος που εξετάστηκε) και της πτώσης που παρατηρείται κατά τη μετάβαση από μία νότα σε μία άλλη (η διαφορά στην περίπτωση αυτή ξεπερνά τα 10db και σε αρκετές περιπτώσεις είναι μεγαλύτερη και από 20 db).

Καθώς όμως εργαζόμαστε με δεδομένα του πραγματικού κόσμου και έτσι ο ρυθμός δεν είναι απόλυτα σταθερός σε αρκετές περιπτώσεις παρατηρείται πως στα δεδομένα υπό εξέταση συμπεριλαμβάνονται τιμές που αντιστοιχούν στο τέλος της δεύτερης από τις δυο νότες ελάχιστης διάρκειας του ίδιου pitch και για τις οποίες ελέγχεται το ενδεχόμενο σύνδεσής τους. Οι ακραίες αυτές τιμές αντιστοιχούν συνήθως στην ολοκλήρωση μιάς νότας όπου η ένταση εξασθενεί σημαντικά. Το φαινόμενο αυτό όπως είναι φυσικό γίνεται πιο έντονο καθώς απομακρυνόμαστε από την αρχή μιάς ηχογράφησης (λόγω σταθερού αριθμού ομαδοποίησης των δεδομένων και ασταθούς ρυθμού). Στις εξαιρετικές αυτές περιπτώσεις παρατηρήσαμε πως όσο μεγαλύτερη είναι η κρίσιμη τιμή που επιλέγεται, για παράδειγμα 8, 10 ή 12db, το αποτέλεσμα βελτιώνεται (παράδειγμα ηχογράφησης: long ago του Πίνακα 4 της §4.3.3). Κατά κανόνα βέβαια, τιμές μεγαλύτερες των 6db οδηγούν στο ίδιο ή χαμηλότερο ποσοστό επιτυχίας (Πίνακας 4, §4.3.3).

Η ηχογράφηση oh clair de la lune χάρη και στην μικρή της διάρκεια εμφανίζει ποσοστό επιτυχίας 100% όπως φαίνεται και στις δύο εικόνες που ακολουθούν (Εικόνες 13 – 14).



Εικόνα 13: score.xml μετά την εκτέλεση του τελικού κώδικα αυτόματης καταγραφής της πρώτης φράσης της ηχογράφησης του μουσικού κομματιού "oh clair de la lune"



**Εικόνα 14:** `score.xml` μετά την εκτέλεση του τελικού κώδικα αυτόματης καταγραφής της πρώτης φράσης της ηχογράφησης του μουσικού κομματιού "oh clair de la lune".

**Περιγραφή:** Μεταβάλλουμε το μουσικό ρυθμό από 4/4 σε 1/4 για τον έλεγχο της σωστής σύνδεσης (παράμετρος `tied` του προτύπου MusicXml) μίας νότας κατά τη συνεχόμενη αλλαγή μέτρου.

Ακολουθεί η παρουσίαση της ροής του προγράμματος. Ο σχετικός κώδικας παρουσιάζεται στο Παράρτημα Β, Β2.



## ΡΟΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

### ΕΙΣΟΔΟΣ

- **data.in** και **names.in** : Αρχεία τα οποία περιλαμβάνουν τις συχνότητες και τα ονόματα των pitch για την εφαρμογή του 3-Nearest-Neighbors με 1-1 αντιστοιχία στις γραμμές/στοιχεία τους
- **test.in** : Αρχείο εξόδου εκτέλεσης σχετικού script του praat για πληροφορίες 'time - pitch - intensity' της ηχογράφησης [ScriptP]. Πρώτο στοιχείο αποτελεί το k του kNN (k=3)
- **input.in** : Αρχείο με πληροφορίες της ηχογράφησης που απαιτούνται για την δημιουργία του score.xml. Αναλυτικά<sup>8</sup>:  
*divisions*: How many divisions per quarter note for a duration (int)  
*min* : Η μέγιστη υποδιαίρεση του beatType στην ηχογράφηση, δηλαδή η ελάχιστη αξία (int) (min = minimum type / beatType)  
*beats*: Time signature numerator (int)  
*beatType*: Time signature denominator (int)  
*sign*: Clef sign. Values: G, F, C, percussion, TAB, none (char)  
*line* : Staff line numbered from bottom to top for clefs (int)  
*tempo*: Tempo in quarter notes per minute (int)  
*moment*: Ακριβής αξία ελλιπούς μέτρου (double) (0.0 εάν όχι ελλιπές)  
*instrument*: Full name of instrument (eg. "Clarinet") (char)  
*program*: MIDI program number from 1 to 128 (eg. 72) (int)



### ΕΠΕΞΕΡΓΑΣΙΑ

1. Ανάγνωση δεδομένων εισόδου.
2. Υπολογισμός 3-Nearest-Neighbors & Δημιουργία **output.out** με το αποτέλεσμα του 3-NN
  - Ευκλείδεια απόσταση μεταξύ των pitch του δείγματος (test.in) και των τιμών αναφοράς (data.in)
3. Δημιουργία **score.xml** βάσει αποτελεσμάτων output.out και δεδομένων intensity που περιέχονται στο αρχείο test.in
  - Ομαδοποίηση τιμών output.out ανά ελάχιστη διάρκεια νότας προς αναζήτηση
  - Εύρεση pitch κάθε ομάδας βάσει πλειοψηφίας (majority voting rule)
  - Έλεγχος γειτονικών όμοιων pitch και της περιβάλλουσας αυτών για πιθανή ανάγκη ένωσης (δηλαδή τότε δύο τέταρτα αποτελούν ένα μισό ή τότε απαιτείται tied note (ιδιότητα 'tied' του προτύπου MusicXml) κατά την αλλαγή μέτρου)
  - Εγγραφή στο score.xml



### ΕΞΟΔΟΣ

- **output.out**: Αρχείο εξόδου αναγνωριστή των pitch του οργάνου σε απάντηση του αρχείου test.in με ζεύγη 'pitch - id' (id αυτό των πινάκων data.in και names.in)
- **note.out**: Αρχείο εξόδου με την αλφαριθμητική ακολουθία από pitch ελάχιστης διάρκειας που 'τυπώνονται' στην παρτιτούρα (για παράδειγμα 'C4-1 C4-2' : νότα με pitch C4 και διάρκεια 2\*ελάχιστη διάρκεια)
- **score.xml** : Αναπαράσταση παρτιτούρας

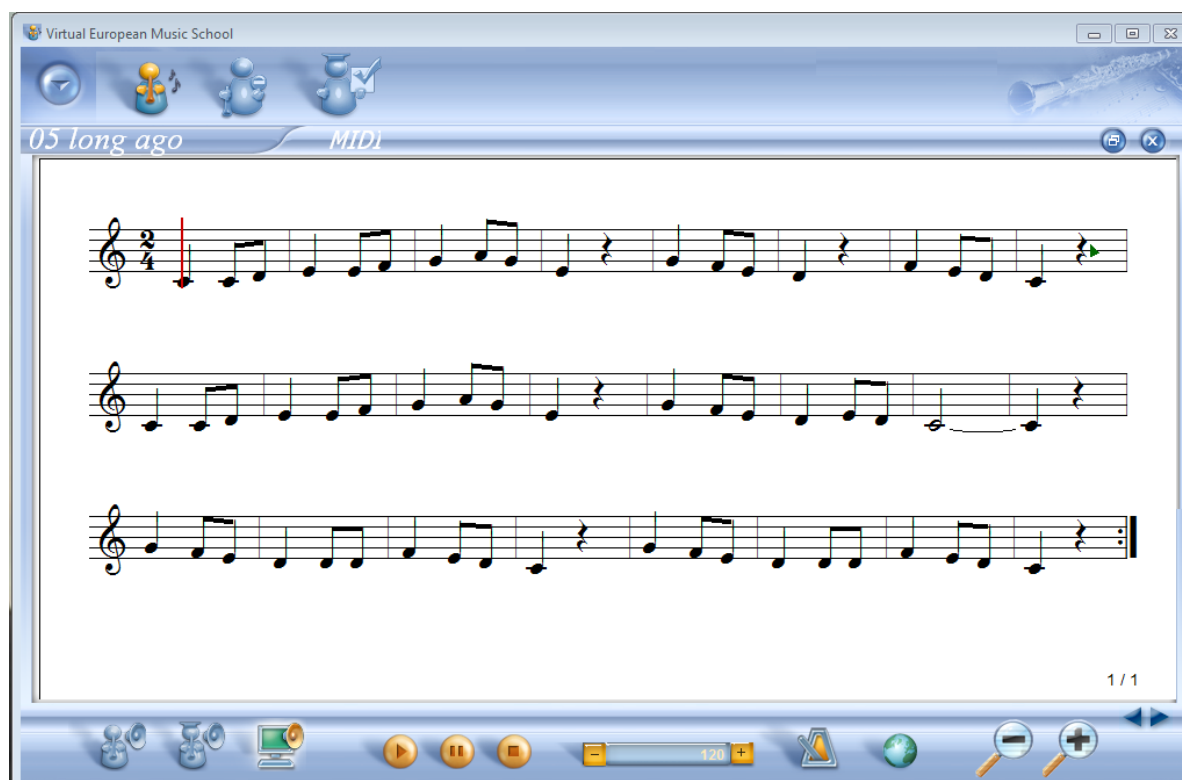
<sup>8</sup> Οι παράμετροι των οποίων η περιγραφή δίνεται στα αγγλικά αποτελούν ιδιότητες που ορίζονται από το πρότυπο MusicXML, σε αντιδιαστολή με τις παραμέτρους που κρίθηκαν ακόμη απαραίτητες για την επίτευξη του επιθυμητού αποτελέσματος.

### 4.3.2 Έλεγχος

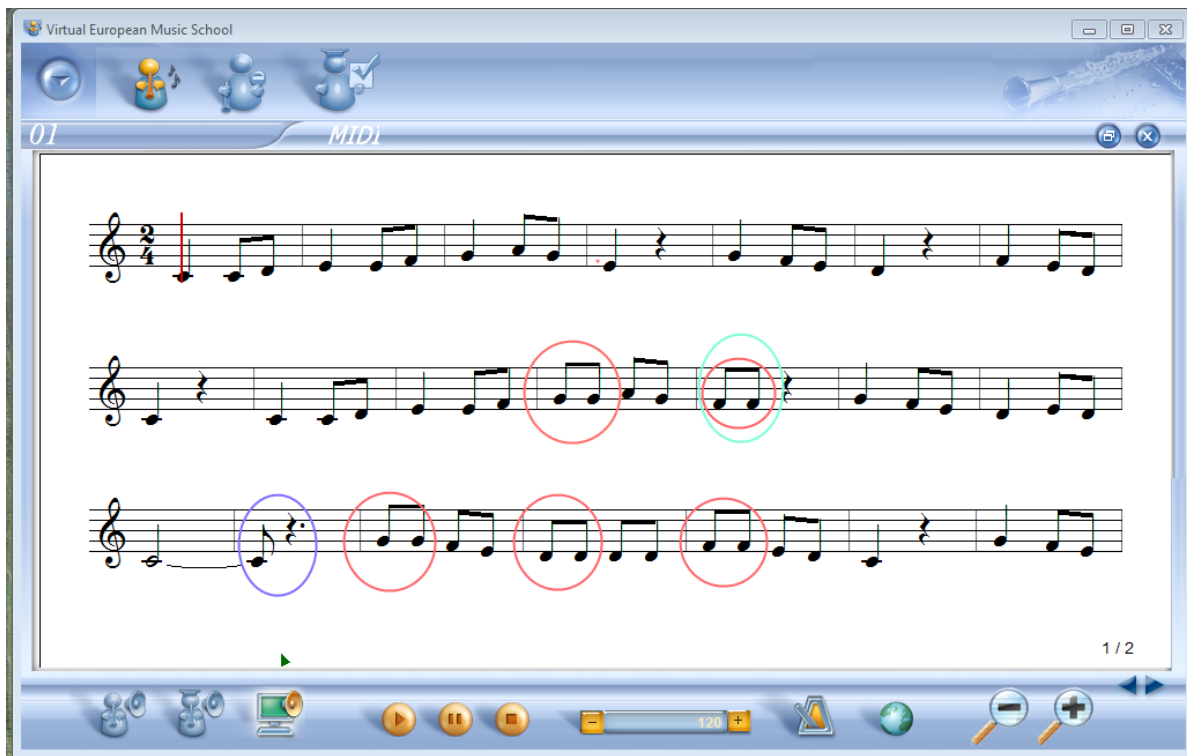
Για τον έλεγχο του προγράμματος εφαρμόσαμε τον κώδικα που παρουσιάστηκε στην προηγούμενη ενότητα (Παράρτημα Β, Β2) σε κάποιες ακόμη ηχογραφήσεις.

Όπως είναι αναμενόμενο, μιά ηχογράφηση η οποία έχει δημιουργηθεί με φυσικό τρόπο, δηλαδή μετά από την ανθρώπινη εκτέλεση ενός μουσικού κομματιού και τη χρήση ενός μικροφώνου, όσο αυστηρά κι αν τηρούνται οι οδηγίες της εκτέλεσης μπορεί να παρουσιάσει κάποιες ίσως όχι αντιληπτές από τους εκτελεστές ή ακροατές αποκλίσεις. Είναι αυτές οι περιπτώσεις κατά τις οποίες το αποτέλεσμα οπτικά κυρίως (διαφορές στην παρτιτούρα οι οποίες γίνονται εύκολα αντιληπτές), αλλά και ακουστικά (μερικές φορές απαιτείται συνειδητή προσπάθεια για να γίνουν κατανοητά κάποια λάθη), δεν είναι το επιθυμητό.

Ένα παράδειγμα αποτελεί η παρακάτω ηχογράφηση (Εικόνες 15-16).

The image shows a screenshot of a software application titled "Virtual European Music School". The main window displays a musical score for a piece titled "05 long ago" in MIDI format. The score is written on three staves in 2/4 time. The first staff begins with a red double bar line. The interface includes a toolbar at the top with icons for navigation and a playback control bar at the bottom with buttons for play, pause, stop, and volume control. The score is displayed in a clean, professional layout.

Εικόνα 15: Παρτιτούρα μουσικού κομματιού "long ago" που περιέχεται στη βάση μουσικών κομματιών του Vemus



Εικόνα 16: score.xml της ηχογράφησης του μουσικού κομματιού "long ago"

Όπως παρατηρούμε τα δυο κομμάτια παρουσιάζουν διαφορές καθώς απομακρυνόμαστε από την έναρξη της ηχογράφησης. Οι διαφορές αυτές αποδίδονται σε αποκλίσεις του δοθέντος tempo της ηχογράφησης. Η εκτίμηση της διάρκειας μιάς νότας προκύπτει λανθασμένη (σειρά 3 μέτρο 2, Εικόνας 17, σημειώνεται με λιλά), καθώς επίσης η απόφαση σχετικά με το διαχωρισμό των φθόγγων βάσει της έντασης του ήχου οδηγείται σε λάθη (σημεία τα οποία είναι σημειωμένα με κόκκινο). Σημειώνεται πως το pitch στη σειρά 2 μέτρο 5 (σημειώνεται με γαλάζιο) συμφωνεί με την ηχογράφηση και άρα αποτελεί ανθρώπινο λάθος στην εκτέλεση του μουσικού κομματιού και όχι σφάλμα του προγράμματος αναγνώρισης.

Όσον αφορά το μουσικολογικό επίπεδο της παρτιτούρας δεν λαμβάνονται υπόψη αρμονικές σχέσεις (Η αρμονία αποτελεί βασικό κομμάτι της πολυφωνικής μουσικής. Μιά μελωδία υπονοεί ωστόσο μιά συγκεκριμένη αρμονική δομή) και η μελωδία εκτιμάται με γνώμονα το ακουστικό αποτέλεσμα και μόνο. Αυτό σημαίνει πως ενώ μιά μελωδία μπορεί να αναπαρασταθεί με περισσότερους του ενός τρόπους όλοι εκ των οποίων θα οδηγήσουν στο ίδιο ακουστικό αποτέλεσμα, υπάρχει μιά μοναδική μορφή η οποία είναι σύμφωνη με τους κανόνες της αρμονίας τους οποίους ο συνθέτης έχει ακολουθήσει. Στην περίπτωση μας, αρκούμαστε σε μιά απεικόνιση της μελωδίας, η αναπαραγωγή της οποίας συμφωνεί με το αρχικό αρχείο ήχου το οποίο μελετάται. Παραλείπονται έτσι χαρακτηριστικά όπως ο οπλισμός και η αρμονική σύνδεση των γεγονότων. Η μουσικολογική ανάλυση μιάς μελωδίας απαιτεί τη συνολική παρακολούθηση ολόκληρων μουσικών φράσεων, η μελέτη των οποίων θα οδηγούσε στην εισαγωγή σημαντικής πολυπλοκότητας υπολογισμών και εκτιμήσεων που θα οδηγούσαν πιθανότατα σε λάθος αποφάσεις, καθώς είναι δύσκολο να ληφθεί υπόψη στα πλαίσια αυτής τη εργασίας το σύνολο των κανόνων και εξαιρέσεων που διέπουν την Ευρωπαϊκή μουσική.

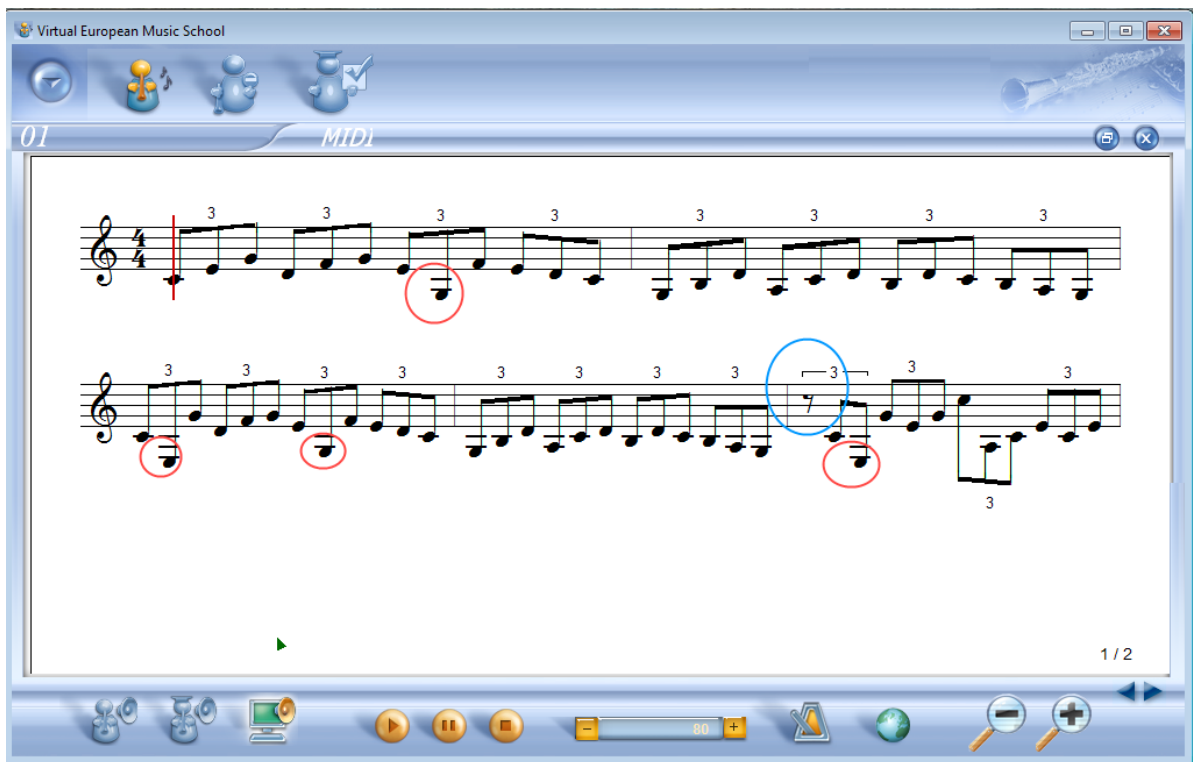
Μιλώντας για κανόνες και εξαιρέσεις, θα πρέπει να αναφερθούμε στις χρονικές αξίες που μπορούν να υπάρξουν στην Ευρωπαϊκή μουσική. Όπως έχει αναφερθεί και νωρίτερα, οι συνήθεις χρονικές αξίες είναι οι παρακάτω:  $1/256^{\circ}$ ,  $1/128^{\circ}$ ,  $1/64^{\circ}$ ,  $1/32^{\circ}$ ,  $1/16^{\circ}$ , ένα όγδοο, ένα τέταρτο, μισό, ολόκληρο, breve και long. Τιμή αναφοράς αποτελεί το ολόκληρο (η αξία του ολόκληρου θεωρείται ίση με 1) και οι υπόλοιπες αξίες αποτελούν άρτια πολλαπλάσια ή υποπολλαπλάσια αυτής (για παράδειγμα,  $1/256^{\circ} = 1/256$ , ένα τέταρτο =  $1/4$ , breve =  $2*1$ ).

Η Ευρωπαϊκή μουσική περιλαμβάνει ακόμη μιά κατηγορία από κανονικές και ακανόνιστες υποδιαίρεσεις ενός φθόγγου οι οποίες αντιμετωπίζονται ως 'ελεύθεροι χρόνοι'. Ο αριθμός των υποδιαίρεσεων αυτών υποδηλώνεται από το όνομά τους, για παράδειγμα ένα τρίγχο αποτελείται από 3 ξεχωριστούς φθόγγους (κανονική υποδιαίρεση) (Εικόνα 17). Το τρίγχο είναι το πιο σύνηθες από την κατηγορία αυτή και έτσι κρίθηκε απαραίτητη η μελέτη του. Η συνολική χρονική αξία των τριών φθόγγων του τρίχχου είναι ίση με την αξία των δύο εξ' αυτών. Αυτό σημαίνει πως εάν για παράδειγμα θέλουμε να εκτελέσουμε ένα τρίγχο ογδόων η συνολική του διάρκεια θα πρέπει να είναι ίση με δυο όγδοα ή ένα τέταρτο. Κατά συνέπεια, ο κάθε φθόγγος έχει διάρκεια ίση με το  $1/3$  της συνολικής διάρκειας, συνεπώς ίση με  $2/3$ . Προκύπτει έτσι ένα κλάσμα το οποίο δεν αποτελεί άρτιο πολλαπλάσιο του ολόκληρου.

Εικόνα 17: Παρτιτούρα μουσικού κομματιού "lefevre" - τρίγχα ογδού

Από την περιγραφή της κατηγορίας αυτής 'ελεύθερων χρόνων', η διαφορετικότητα εντοπίζεται στον ξεχωριστό τρόπο κατά τον οποίο γίνεται επιτρεπτή η υποδιαίρεση ενός φθόγγου. Η πληροφορία σχετικά με τον τρόπο της υποδιαίρεσης αναφορικά με το πρότυπο MusicXml μπορεί να εκφραστεί μέσω των παραμέτρων *divisions* και *duration*, ενώ στο κώδικά μας βασική είναι η παράμετρος *min* που ορίζει τη μέγιστη υποδιαίρεση της αξίας αναφοράς (*beatType*) και κατά συνέπεια την ελάχιστη αξία προς αναζήτηση. Έτσι, εάν για παράδειγμα το *beatType* είναι ίσο με 4, δηλαδή η αξία αναφοράς είναι ένα τέταρτο, και το *divisions* οριστεί ίσο με 6, οι πιθανές υποδιαίρεσεις θέτοντας *min* = 6, αναζητώντας δηλαδή όλες τις πιθανές υποδιαίρεσεις που η παράμετρος *divisions* μας επιτρέπει, οδηγούν στις ακόλουθες πιθανές τιμές της παραμέτρου *duration*: 1 (δυο τρίγχα δεκάτων έκτων), 2 (ένα τρίγχο ογδόων), 3 (δυο όγδοα), 4 (τρίγχο τετάρτου με όγδοο), 5 (τρίγχο τετάρτου παρεστιγμένου με δέκατο έκτο) και 6 (ένα τέταρτο), όλα εκ των οποίων έχουν συνολική αξία ενός τετάρτου.

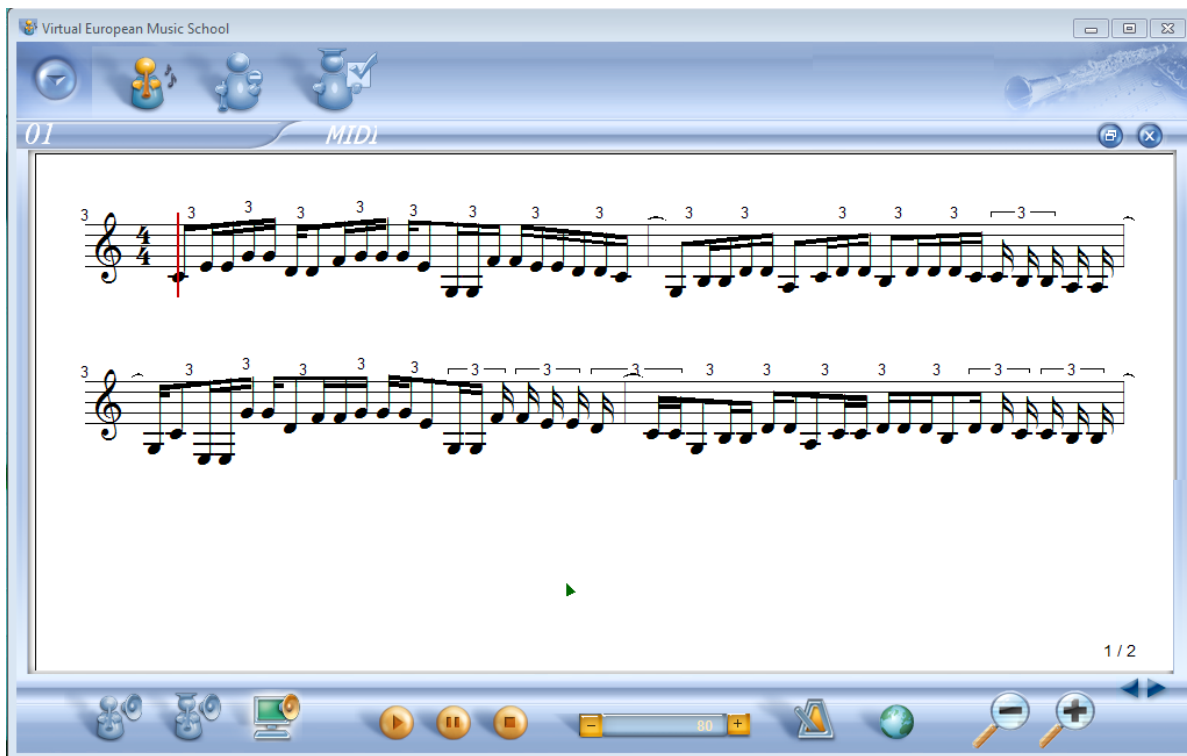
Για την καταγραφή της ηχογράφησης του συγκεκριμένου κομματιού (Εικόνα 17), έχουμε ορίσει τις σχετικές παραμέτρους όπως ακολούθως: *divisions*=6 και *min*=3. Ακολουθεί η έξοδος του προγράμματός μας (Εικόνα 18).



Εικόνα 18: score.xml του κομματιού "lefevre" (min=3)

Όπως παρατηρούμε το αποτέλεσμα είναι πολύ κοντά στο επιθυμητό. Ένα βασικό πρόβλημα είναι η εκτίμηση που το praat μας παρέχει για τις νότες Σολ(G), G3 αντί G4, και Μι(E), E3 αντί E4. Σφάλματα αυτού του τύπου εμφανίζονται παρά τη ρύθμιση της παραμέτρου προτίμησης υψηλότερων συχνοτήτων octave cost (§4.1.2) καθώςον εξετάζονται ηχογραφήσεις από διαφορετικά μουσικά όργανα (δεν χρησιμοποιείται πάντα το ίδιο κλαρινέτο) ή εκτελεστές. Το πρόβλημα στην συγκεκριμένη ηχογράφιση δεν λύνεται ούτε με την αύξηση της παραμέτρου octave cost, περίπτωση κατά την οποία οι λάθος εκτιμώμενες τιμές αντικαθίσταται από παύσεις (δεν εκτιμάται κάποια τιμή pitch στο καθορισμένο εύρος). Το δεύτερο σφάλμα το οποίο παρατηρείται στο συγκεκριμένο παράδειγμα είναι πως μεταξύ πρώτης και δεύτερης μουσικής φράσης ο εκτελεστής παίρνει μία αναπνοή η οποία μουσικά μας βοηθάει στο να αντιληφθούμε την 'αλλαγή' η οποία πραγματοποιείται, αλλά η καθυστέρηση που προστίθεται οδηγεί στην εμφάνιση μιάς παύσης και σαν αποτέλεσμα η δεύτερη φράση καταγράφεται μετατοπισμένη κατά ένα όγδοο τριήχου.

Στη συνέχεια δοκιμάζουμε να εκτελέσουμε το πρόγραμμα αυτόματης καταγραφής για την ίδια ηχογράφιση αυτή τη φορά μεταβάλλοντας την τιμή min = 6 (Εικόνα 19), τιμή την οποία θα επιλέγαμε εάν περιμέναμε την εμφάνιση και δυο ογδών ως υποδιαίρεση ενός τετάρτου. Στην Εικόνα 19 παρατηρούμε πως τα γεγονότα δεν ενοποιούνται ανά δυο όπως θα αναμέναμε και αποδίδεται στην ποιότητα της εκτέλεσης της μελωδίας – staccato – , όπως δηλώνεται από τη μικρή τελεία που σημειώνεται κάτω από κάποιους φθόγγους στις οδηγίες της εκτέλεσης (Εικόνα 17). Στο staccato το παίξιμο είναι κοφτό, νευρικό και δημιουργεί παύσεις, ενώ έχουμε την αίσθηση της ζωηρότητας και του απροσδόκητου. Ένα ακόμη λάθος το οποίο εισάγεται έγκειται στο γεγονός πως όσο πιο μεγάλη είναι η τιμή του min τόσο πιο αυστηρό θα πρέπει να είναι το παίξιμο του εκτελεστή ώστε να οδηγούμαστε στο επιθυμητό αποτέλεσμα.



Εικόνα 19: score.xml του κομματιού "lefevre" (min=6)

### 4.3.3 Αξιολόγηση

Για τη διαδικασία της αξιολόγησης των αρχείων εξόδου εφαρμόσαμε τον κανόνα του Levenshtein (§3.4) μεταξύ της συμβολικής αλφαριθμητικής ακολουθίας αναπαράστασης του συνόλου από pitch ελάχιστης διάρκειας που καταγράφονται στην παρτιτούρα εξόδου και της αντίστοιχης ακολουθίας αναφοράς (η οποία θα οδηγούσε στο επιθυμητό αποτέλεσμα). Στην ακολουθία αυτή κάθε pitch ελάχιστης διάρκειας συμβολίζεται συνοδευόμενο από έναν ακέραιο θετικό αριθμό ο οποίος αυξάνεται όταν η νότα 'δένεται' με την προηγούμενη της, δηλαδή αποτελεί μέρος της συνολικής διάρκειας μιάς νότας του συγκεκριμένου pitch, διαφορετικά έχει την τιμή 1 (για παράδειγμα η ακολουθία 'C4-1 C4-2' αναπαριστά μιά νότα C της 4ης οκτάβας με συνολική διάρκεια τη διπλάσια της ελάχιστης που εμφανίζεται στο μουσικό κομμάτι ενώ η ακολουθία 'C4-1 C4-1' αναπαριστά δύο ξεχωριστές νότες C της 4ης οκτάβας που η κάθε μιά έχει διάρκεια ίση με την ελάχιστη).

Ένα λεπτό σημείο της λογικής αυτής είναι ότι το αρχείο αναφοράς το οποίο χρησιμοποιείται για τη σύγκριση διαφέρει ελάχιστα από αυτό που θα χρησιμοποιηθεί στην επόμενη παράγραφο ως το διορθωμένο για τη δημιουργία της επιθυμητής παρτιτούρας (§4.3.4). Τα pitch ελάχιστης διάρκειας που έχουν εκτιμηθεί σωστά λαμβάνουν ως θετικό αριθμό αυτόν που συμφωνεί με το αρχικό αρχείο. Για παράδειγμα, ας θεωρήσουμε τις ακολουθίες αρχική και επιθυμητή όπως ακολούθως:

| αρχική | επιθυμητή | αναφοράς    |
|--------|-----------|-------------|
| C#4-1  | C4-1      | <b>C4-1</b> |
| C4-1   | C4-2      | <b>C4-1</b> |
| C4-2   | C4-3      | <b>C4-2</b> |

Η επιθυμητή ακολουθία λαμβάνεται υπόψη για να οριστεί η ακολουθία αναφοράς που συμφωνεί με την αρχική ακολουθία όταν το pitch είναι το σωστό.

Ο κώδικας και τα σχετικά αρχεία παρουσιάζονται στο Παράρτημα Β, Β3.

Ακολουθεί ο πίνακας με πληροφορίες από την αξιολόγηση της αυτόματης δημιουργίας παρτιτούρας κάποιων ηχογραφήσεων (Πίνακας 4).

| Όνομα ηχογράφησης (*.wav) | Levenshtein distance | Σύνολο από pitch ελάχιστης διάρκειας στο αρχείο αναφοράς | Ποσοστό επιτυχίας (%) | Σχόλια                                                                                                                                                               |
|---------------------------|----------------------|----------------------------------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| lefevre                   | 12                   | 96                                                       | <b>87.50</b>          | -                                                                                                                                                                    |
| lullaby                   | 15                   | 98                                                       | <b>84.69</b>          | Χειροτέρευση αποτελέσματος για μεγαλύτερη τιμή κατωφλίου του διαχωρισμού γεγονότων (>6db) (74.49% για 12db)                                                          |
| fegaraki mou lampro       | 8                    | 64                                                       | <b>87.50</b>          | Χειροτέρευση αποτελέσματος για μεγαλύτερη τιμή κατωφλίου του διαχωρισμού γεγονότων (>6db) (75% για 12db)                                                             |
| oh clair de la lune       | 1                    | 16                                                       | 93.75                 | <b>100%</b> ποσοστό επιτυχίας μετά τη διαγραφή επιπλέον τιμών στο αρχείο εισόδου οι οποίες προσθέτουν μία παύση μετά το τέλος της ολοκλήρωσης της φράσης (Εικόνα 13) |
| long ago                  | 18                   | 160                                                      | <b>88.75</b>          | Βελτίωση ποσοστού επιτυχίας μετά από αύξηση της τιμής κατωφλίου του διαχωρισμού γεγονότων ( <b>94.38%</b> για 12db )                                                 |

**Πίνακας 4: Ενδεικτικά αποτελέσματα αξιολόγησης**

Το σύνολο από pitch ελάχιστης διάρκειας (3η στήλη) είναι ενδεικτικό της συνολικής διάρκειας μιάς ηχογράφησης, και καθώς αυξάνεται είναι φυσικό να αυξάνεται και η τιμή επιστροφής της εφαρμογής του κανόνα Levenshtein (περισσότερες διαγραφές, εισαγωγές ή αντικαταστάσεις). Η απόσταση Levenshtein είναι ενδεικτική των αλλαγών στις οποίες ο χρήστης της εφαρμογής θα πρέπει να προχωρήσει για τη διόρθωση της παρτιτούρας.

Από την παρακολούθηση όλης της διαδικασίας και από τα στοιχεία του Πίνακα 4 μπορούμε να συμπεράνουμε πως τα αποτελέσματα της αυτόματης δημιουργίας της παρτιτούρας είναι ικανοποιητικά και το αρχείο score.xml το οποίο δημιουργείται είναι πολύ κοντά στο επιθυμητό. Τα συνήθη λάθη τα οποία υπάρχουν πηγάζουν είτε από την ίδια την εκτέλεση (για παράδειγμα, ασταθές tempo ή εκφραστικότητα της εκτέλεσης) είτε από την εκτίμηση χαμηλότερου κατά μία οκτάβα pitch από το αντιλαμβανόμενο.

Η στήλη με τα 'Σχόλια' σημειώνεται σε συμπλήρωση θεμάτων τα οποία έχουν αναφερθεί στην παράγραφο 4.3.1 κατά την παρουσίαση της σχεδίασης του συστήματος.

#### **4.3.4 Δυνατότητα διόρθωσης παρτιτούρας**

Το αρχείο score.xml που δημιουργείται με την αυτόματη διαδικασία που έχει περιγραφεί στην ενότητα αυτή μπορεί πολύ εύκολα να τροποποιηθεί και να διαμορφωθεί με γραφικό τρόπο από κάθε πρόγραμμα μουσικής σημειογραφίας το οποίο υποστηρίζει το πρότυπο MusicXml και διαθέτει music score editor (για παράδειγμα το Finale). Για λόγους πληρότητας της εργασίας αλλά και διευκόλυνσης της διαδικασίας αξιολόγησης της προηγούμενης παραγράφου (§4.3.3) υλοποιήθηκε συμπληρωματικά ένα πρόγραμμα το οποίο δέχεται ως είσοδο τη διορθωμένη αλφαριθμητική ακολουθία (επιθυμητή) από pitch ελάχιστης διάρκειας και αναδημιουργεί την παρτιτούρα. Ο σχετικός κώδικας παρουσιάζεται στο Παράρτημα Β, Β4.



# 5

## *Επίλογος*

### *5.1 Σύνοψη & Συμπεράσματα*

Στο Κεφάλαιο 2 αναφερθήκαμε στο ευρύτερο πλαίσιο στο οποίο εντάσσεται η εργασία αυτή, αυτό της αυτόματης καταγραφής της μουσικής και της δημιουργίας της παρτιτούρας που μπορεί να περιγράψει ικανοποιητικά μία ηχογράφιση μουσικής. Το ερευνητικό ενδιαφέρον εντοπίζεται σε περισσότερα του ενός επίπεδα και διαφορετικές προσεγγίσεις έχουν προσφέρει σημαντικές γνώσεις. Η καταγραφή της μονοφωνικής μουσικής είναι σαφώς απλούστερη και μπορεί να θεωρηθεί ένα πρόβλημα το οποίο έχει επιλυθεί. Η πολυφωνική μουσική και η δυναμική της ποικιλομορφία δυσκολεύουν την αναζήτηση ενός γενικά εφαρμόσιμου και αποτελεσματικού συστήματος.

Καθώς η εργασία αυτή είχε στόχο την ενσωμάτωσή της στο μουσικό πλαίσιο διδασκαλίας Vemus [1], η σχεδίαση του συστήματος που πραγματοποιήθηκε, είχε ως γνώμονα τη βέλτιστη επίλυση του προβλήματος της αυτόματης καταγραφής των μουσικών κομματιών που το Vemus διαχειρίζεται. Τα αποτελέσματα τα οποία προέκυψαν αποδεικνύουν ότι αυτό έχει επιτευχθεί. Το επόμενο βήμα προς την κατεύθυνση της μεγίστης αυτοματοποίησης του συστήματος, θα ήταν η ανεξαρτησία του από κάποιες, αν όχι όλες, τις παραμέτρους εισόδου και προς την κατεύθυνση της γενίκευσής του, η εφαρμογή της διαδικασίας της αυτόματης καταγραφής και για ηχογραφήσεις από άλλα μονοφωνικά μουσικά όργανα, πλέον του κλαρινέτου.

Η ψηφιακή παρτιτούρα που παράγεται, μπορεί γρήγορα και εύκολα να διορθωθεί και να συμπληρωθεί με επιπλέον στοιχεία, σύμφωνα με τις επιθυμίες του δασκάλου και στη συνέχεια να διατεθεί στους μαθητές προς μελέτη, συνοδευόμενη από την ηχογράφιση αναφοράς. Το σύστημα αυτόματης καταγραφής το οποίο σχεδιάστηκε, θα διευκολύνει λοιπόν τη διαδικασία της εισαγωγής νέων μουσικών κομματιών στο Vemus και κατ' επέκταση θα ενισχύσει την εκπαιδευτική διαδικασία.

## 5.2 Μελλοντικές Επεκτάσεις

Οι παράμετροι που αφορούν το μουσικό κλειδί (παράμετροι sign και line που ορίζονται στον κώδικα που υλοποιήθηκε και παρουσιάζεται στο Παράρτημα Β, Β2) με βάση το οποίο ορίζεται ότι θα αποτυπωθούν οι νότες, και το μουσικό ρυθμό (παράμετροι beat και beat type, πχ. 4/4 ή 6/8) σύμφωνα με τον οποίο αυτές οργανώνονται στο χρόνο, αποτελούν απαραίτητα δεδομένα του αρχείου MusicXml που δεν μπορούν να προκύψουν με κάποιον τρόπο από το ακουστικό σήμα. Η παράμετρος η οποία ορίζει το μουσικό όργανο το οποίο αναπαράγεται (instrument) προϋποθέτει την ενσωμάτωση στο σύστημα γνώσεων από το ξεχωριστό πεδίο έρευνας της αναγνώρισης των μουσικών οργάνων. Η παράμετρος του ρυθμού (για παράδειγμα 120 τέταρτα το λεπτό) (tempo) απαιτεί επίσης γνώσεις οι οποίες μπορούν να προκύψουν από την παρακολούθηση του ρυθμού (tempo and beat tracking). Οι πληροφορίες από την παρακολούθηση του ρυθμού (tempo) σε συνδυασμό με το μουσικό ρυθμό στον οποίο θέλουμε να απεικονίσουμε τις νότες (παράμετροι beat και beat type) θα ήταν ίσως αρκετά για να συμπεράνουμε τη διάρκεια κάθε νότας ξεχωριστά. Με τον τρόπο αυτό δε θα ήταν απαραίτητο να ορίσουμε τις παραμέτρους οι οποίες καθορίζουν την ελάχιστη διάρκεια που εμφανίζεται (min) και τη χρονική στιγμή κατά την οποία ξεκινά το μουσικό κομμάτι σε περίπτωση ελλιπούς μέτρου (moment). Τέλος, μία κατάλληλη τιμή για την παράμετρο divisions, η οποία θα επιτρέπει την εκτέλεση της σωστής διάρκειας κάθε νότας που εμφανίζεται, θα μπορούσε εύκολα να επιλεγεί. Προκύπτει λοιπόν η απαίτηση του συνδυασμού γνώσης από διαφορετικά πεδία έρευνας για τη δημιουργία ενός πιο 'έξυπνου' συστήματος.

Ενδιαφέρον θα παρουσίαζε ο έλεγχος της λειτουργίας του μέρους της καταγραφής της παρτιτούρας (στάδιο τρίτο, §4.3) με χρήση του αναγνωριστή pitch του οργάνου όπως έχει σχεδιαστεί και πραγματοποιείται από το Vemus. Ταυτόχρονα, με τον τρόπο αυτό, το μέρος αυτό της υλοποίησης θα μπορούσε να εφαρμοστεί και για την αυτόματη δημιουργία της παρτιτούρας σε ηχογραφήσεις των υπόλοιπων μουσικών οργάνων για τα οποία έχει σχεδιαστεί η μουσική αυτή εφαρμογή.

Τέλος, η ενσωμάτωση της υλοποίησης στο Vemus, θα ήταν απαραίτητο να συνοδευτεί από τη δημιουργία ενός score editor, για τη διευκόλυνση της γραφικής επεξεργασίας και διόρθωσης της παρτιτούρας, η οποία παράγεται αυτόματα.

## Παράρτημα Α – Το Ερευνητικό Έργο VEMUS

The poster for the VEMUS project features a central blue figure of a person playing a flute, surrounded by musical notes and a large yellow sun. The background is orange with various small photos of people playing instruments. The text is in white and blue, with a blue footer containing logos and project details.

www.vemus.org

# VEMUS

## Virtual European Music School

Exploration of new teaching practices in music

VEMUS aims at developing an open, highly interactive, and networked multilingual music tuition framework for popular wind instruments such as the flute, the clarinet, the saxophone and the trumpet.

**Objectives**

- Integration & pedagogical innovation through complementary environments and novel features
- VEMUS will develop and validate a unified innovative, pedagogically-innovated e-learning environment seamlessly blended with traditional face-to-face lessons.

**Environments**

- self-practicing environments
- automatic performance evaluation
- constructive feedback
- maximizing the effectiveness of practicing
- group/interactive settings
- collaborative learning
- group activities
- enriching learning experience
- a distance learning platform
- content repository
- communication and progress monitoring tools
- virtual meeting, practicing and remote coaching

**Validation**

user-centred iterative approach

VEMUS will scale-up the validation activities to obtain more coherent and statistically meaningful results and to identify the critical factors for wider deployment putting special focus on the new requirements and new potentials in the enlarged Europe.

VEMUS adopts a strongly user-centred iterative approach, with user groups actively participating throughout the lifetime of the project, from requirements, to field-tests, evaluation activities, and over.

Contact Information: Institute for Language and Speech Processing [www.ilsp.gr](http://www.ilsp.gr) Artemidos 6 & Epidavrou GR-15125, Maroussi, Athens, GREECE

**PROJECT PARTNERS**

Project name: VEMUS - Virtual European Music School / Ref. No: IST-27952 / Programme: FP6-2004-IST-4  
Activity: IST-3304-2.4.13. Strengthening the Integration of the ICT research efforts in an Enlarged Europe / Starting date: 1/10/2005 / Duration: 36 months / Website: [www.vemus.org](http://www.vemus.org)

Εικόνα : VEMUS project

Το VEMUS (Virtual European Music School) είναι ένα ερευνητικό και αναπτυξιακό έργο που συγχρηματοδοτήθηκε από την Ευρωπαϊκή Επιτροπή στο πλαίσιο του Προγράμματος των Τεχνολογιών της Κοινωνίας της Πληροφορίας (IST - Information Society Technologies) του Έκτου Προγράμματος-Πλαισίου (FP6 - Sixth Framework Program).

Το έργο αυτό ξεκίνησε τον Οκτώβριο του 2005 με στόχο την ανάπτυξη και αξιολόγηση πρωτοποριακών τεχνολογικών στοιχείων σχεδιασμένων βάσει παιδαγωγικών κριτηρίων. Πιο συγκεκριμένα, είχε στόχο τη δημιουργία ενός πλαισίου μουσικής διδασκαλίας για δημοφιλή μουσικά όργανα που θα είναι ανοικτό, διαδικτυακό και πολύγλωσσο και θα προσφέρει υψηλή αλληλεπιδραστικότητα καθώς και ένα σύνολο από καινοτόμα εργαλεία ηλεκτρονικής μάθησης. Η ομάδα-στόχος χρηστών είναι σπουδαστές πνευστών οργάνων κατά τη διάρκεια των πρώτων 4-5 ετών της μουσικής εκπαίδευσής τους. Το VEMUS θα μπορούσε να θεωρηθεί ως εικονικός βοηθητικός δάσκαλος που καθοδηγεί το σπουδαστή ενώ εξασκείται.

Το περιβάλλον του VEMUS ενσωματώνει καινοτομικά, παιδαγωγικά στοιχεία ηλεκτρονικής μάθησης ώστε να επεκτείνει την παραδοσιακή διδασκαλία της μουσικής σε διαφορετικά μαθησιακά περιβάλλοντα:

**Αυτό-εξάσκηση:** Ένα σύνολο εφαρμογών έχουν σχεδιαστεί, οι οποίες ενισχύουν την εξάσκηση, στο σπίτι ή στην τάξη, ώστε να γίνονται οι συνεδρίες εξάσκησης περισσότερο επιμορφωτικές, δημιουργικές και ψυχαγωγικές. Η αυτόματη αξιολόγηση της εκτέλεσης και η υψηλού επιπέδου ανάδραση, προσαρμοσμένη σε κάθε σπουδαστή, ενισχύουν την αποτελεσματικότητα της εξάσκησης στο σπίτι.

**Αίθουσα διδασκαλίας Μουσικής:** Το VEMUS ερευνώντας και αξιολογώντας καινοτόμα εργαλεία στοχεύει στην υποστήριξη της διδασκαλίας της μουσικής σε περιβάλλον ομάδας, με διδακτικά βοηθήματα που υποστηρίζουν τον δάσκαλο, την ομαδική μάθηση και τις ομαδικές δραστηριότητες. Αυτά τα εργαλεία αποτελούν τη βάση της εμφάνισης του VEMUS στην αίθουσα διδασκαλίας ενισχύοντας τις διαδικασίες διδασκαλίας της μουσικής και την εμπειρία εκμάθησης του μαθητή.

**Επεκτάσεις μάθησης εξ' αποστάσεως :** Το περιβάλλον του VEMUS παρέχει επεκτάσεις μάθησης εξ' αποστάσεως, συντηρώντας και υποστηρίζοντας τη διαχείριση μιάς πηγής ανοικτού περιεχομένου και προσφέροντας εργαλεία συγγραφής και επεξεργασίας, παρέχοντας εργαλεία επικοινωνίας και επιτρέποντας διδασκαλία από απόσταση των μαθητών και έλεγχο της προόδου τους με το πέρασμα του χρόνου. Αυτά τα γνωρίσματα βοηθούν στην περαιτέρω ενίσχυση της σχέσης δασκάλου-μαθητή ενώ καθιστούν δυνατή τη συμμετοχή μαθητών που διαφορετικά θα ήταν αδύνατη εξαιτίας γεωγραφικών ή άλλων περιορισμών.

Το VEMUS στηρίχθηκε στην γνώση και τα αποτελέσματα του επιτυχημένου έργου IMUTUS που υλοποίησε ένα αποδοτικό περιβάλλον μουσικής εξάσκησης για τη φλογέρα. Επέκτεινε την τεχνολογική και παιδαγωγική βάση του IMUTUS ώστε να καλύψει επιπρόσθετα δημοφιλή μουσικά όργανα όπως το φλάουτο, το κλαρινέτο και το σαξόφωνο, επιπρόσθετα περιβάλλοντα εκμάθησης και ενισχυμένα εκπαιδευτικά σενάρια. Το σύστημα απευθύνεται σε αρχάριους έως μέσου επιπέδου σπουδαστές. Τα προκαταρκτικά αποτελέσματα από την αξιολόγηση του IMUTUS κατέστησαν φανερή την ισχυρή δυναμική της προσέγγισης που υιοθετήθηκε. Το VEMUS διευρύνει τις δραστηριότητες αξιολόγησης, επιδιώκοντας αφενός τη συλλογή πυκνότερων και στατιστικά πληρέστερων παρατηρήσεων και αφετέρου τον προσδιορισμό των κρίσιμων εκείνων παραγόντων που θα επιτρέψουν μιά ευρύτερης κλίμακας αξιοποίηση του συστήματος.

Μιά από τις βασικές επιδιώξεις του έργου VEMUS ήταν να συνεισφέρει στην ενίσχυση και την ολοκλήρωση του Ευρωπαϊκού ερευνητικού πεδίου στον τομέα των τεχνολογιών της Κοινωνίας της Πληροφορίας, μέσω της ενεργούς συμμετοχής φορέων από τα Νέα Κράτη μέλη. Για τον σκοπό αυτό, στην Κοινοπραξία του Έργου συμμετείχαν 8 οργανισμοί από 6 χώρες, 3 από τις οποίες είναι Νέα ή Συνεργαζόμενα Κράτη. Το VEMUS συγκέντρωσε όλη την απαραίτητη πραγματογνωμοσύνη από όλους τους τομείς που ενέχονται και υιοθετεί καθαρά μιά ευρωπαϊκή διάσταση με τη συμμετοχή εταίρων από 6 χώρες. Έχοντας τη μουσική ως κύριο όχημα και εμπλέκοντας ομάδες σε διαφορετικές χώρες και διάφορες γλώσσες, το VEMUS αναζητά να πλησιάσει ανθρώπους με διαφορετική ιστορία και κουλτούρα, δημιουργώντας πολιτιστικούς δεσμούς και διασφαλίζοντας μιά μεγαλύτερη συμμετοχή στη γνώση και στα αποτελέσματα της έρευνας καθώς και ένα μεγαλύτερο αντίκτυπο.

Το VEMUS υιοθετεί μία προσέγγιση ισχυρά προσανατολισμένη στο χρήστη, με ομάδες χρηστών (user groups) που συμμετέχουν ενεργά σε όλη τη διάρκεια του έργου: από προϋποθέσεις, σε δοκιμές σε διάφορα πεδία, αξιολόγηση και άλλα. Η συμμετοχή ομάδων χρηστών στην κοινοπραξία επιβεβαιώνει μία ισχυρά προσανατολισμένη στο χρήστη διαδικασία εφαρμογής και δίνει την ευκαιρία στο έργο να αναπτύξει την προσέγγισή του σε ακόμη μεγαλύτερη κλίμακα και να καταπιαστεί με σημαντικούς παράγοντες που θα αποτελέσουν τη βάση για ευρύτερες μεταγενέστερες πρωτοβουλίες.

# Παράρτημα Β – Πληροφορίες Υλοποίησης

## B1. k - Nearest - Neighbors

### KNN.cpp

```
#include <iostream> // Header file gia leitourgies eisodou/e3odou
#include <fstream> // Header file gia xeirismo anagnwshs/egrafhs apo/se arxeia
#include <cmath> // Header file gia vasikes ma8hmatikes pra3eis
#include <string> // Header file gia xeirismo string

// Omadopoihsh klasewn, antikeimenwn kai synarthsewn ypo ena onoma
using namespace std; // std: standard C++ library

// Functions prototypes
void sortFirstK (int k, int testSize, int dataSize, double** distanceArray, int** indexArray);
void computeKnn (double* dataArray, double* testArray, int dataSize, int testSize, int k, char** namesArray,
char** namesIdArray);

int main ()
{

int k,i; // To k antiproswpeuei to k-Nearest-Neighbors kai to i metavlth vroxou

// -----
// Anagnwsh dedomenwn ws input stream apo to arxeio data.in
ifstream inDataFile("data.in", ios::in); // To data.in periexei tis dedomenes syxnothtes

// Exit program if can not open file
if(!inDataFile)
{
cerr << "File could not be opened!" << endl;
exit(1);
}

// Ypologismos plh8ous stoxeiwn sto arxeio data.in
int dataSize=0; // Mege8os arxeiou data.in
double num; // Proswrinh apo8hkeush ka8e stoxeiou tou arxeiou
for(;;)
{
inDataFile >> num;
if ( inDataFile.eof() )
{
break;
}
dataSize++;
}

inDataFile.clear(); // Set error state flags
inDataFile.seekg(0, ios::beg); // Ka8orismos tou get pointer sthn arxh tou arxeiou (Offset 0 from the
beginning of the stream's buffer)

//-----
```

```

/// Kataskeuh pinaka dataArray kai apo8hkeush tw n stoixeiwn tou arxeiou data.in
double* dataArray;
dataArray = new double[dataSize];

for(i = 0; i < dataSize; i++)
{
    inDataFile >> dataArray[i];
}

inDataFile.close();    // Kleisimo arxeiou data.in

/// -----
/// Anagnwsh dedomenwn ws input stream apo to arxeio test.in
ifstream inTestFile("test.in", ios::in);    // To test.in periexei tis syxnothtes tou deigmatos pros
kathgoriopoihsh

/// exit program if can not open file.
if(!inTestFile)
{
    cerr << "File could not be opened." << endl;
    exit(1);
}

/// Ypologismos plh8ous stoixeiwn sto arxeio test.in
int testSize = 0; // Mege8os arxeiou test.in
char num2[14];    // Proswrinh apo8hkeush stoixeiwn tou arxeiou
inTestFile >> num2;    // To prwto stoixeio einai to k kai metrame ta ypoloipa stoixeia pou 8a
apo8hkeutoun ston pinaka testArray[testSize]
for(;;)
{
    inTestFile >> num2 >> num2;

    if (inTestFile.eof())
    {
        break;
    }
    testSize++;
}

inTestFile.clear();    // Set error state flags
inTestFile.seekg(0, ios::beg);    // Ka8orismos tou get pointer sthn arxh tou arxeiou (Offset 0 from the
beginning of the stream's buffer)

/// San prwto stoixeio tou arxeiou dinetai to k (k-Nearest-Neighbors)
inTestFile >> k;

/// Kataskeuh pinaka testArray kai apo8hkeush tw n stoixeiwn tou arxeiou test.in (to arxeio periexei zeugh
"time-pitch")
double* testArray;
testArray = new double[testSize];

for(i = 0; i < testSize; i++)

```

```

    {
        inTestFile >> num2 >> num2;    // To prwto stoixeio ka8e zeugous einai to time (xronos) kai
den xrhsimopoieitai

        char* und = "--undefined--";    // Se periptwsh pou den ektimatai kapoia timh to praat
epistrefei "undefined" to opoio antistoixoume sthn timh mhden (0)
        if ( strcmp(num2,und) == 0 )
            testArray[i] = 0.0;
        else
            testArray[i] = atof(num2);
    }

    inTestFile.close(); // Kleisimo arxeiou test.in

    /// -----
    /// Anagnwsh dedomenwn ws input stream apo to arxeio names.in me ta pitch pou antistoixoun stis
dedomenes syxnothtes tou data.in
    ifstream inNamesFile("names.in", ios::in);

    /// Exit program if can not open file
    if(!inNamesFile)
    {
        cerr << "File could not be opened." << endl;
        exit(1);
    }

    /**
     *   Kataskeuh pinakwn
     *   namesArray[dataSize] gia apo8hkeush tw n pitch
     *   kai namesIdArray[dataSize] gia ta id autwn
     *   apo to arxeio names.in
     *   To mege8os autwn tw n pinakwn vasei antistoixias me ta stoixeia tou dataArray
     *   einai iso me dataSize
     */
    char** namesArray; // Pinakas me ta pitch tw n syxnothtw n tou dataArray
    namesArray = new char*[dataSize];
    char** namesIdArray; // Pinakas me ta id tw n pitch se 1-1 antistoixia me ton namesArray
    namesIdArray = new char*[dataSize];

    for(i = 0; i < dataSize; i++)
    {
        namesArray[i] = new char[3];    // e.g. "C#4"
        namesIdArray[i] = new char[3]; // e.g. "10c"
        inNamesFile >> namesArray[i] >> namesIdArray[i];
    }

    inNamesFile.close(); // Kleisimo arxeiou names.in

    /// -----
    /// Klhsh synarthshs ypologismou k-Nearest-Neighbors tw n stoixeiwn testArray - dataArray
    computeKnn( (double*)dataArray, (double*)testArray, dataSize, testSize, k, (char **)namesArray, (char
**) namesIdArray);

    delete dataArray;    // Free memory pointed to by dataArray

```



```

delete testArray;      // Free memory pointed to by testArray
delete namesArray;    // Free memory pointed to by namesArray
delete namesIdArray;  // Free memory pointed to by namesIdArray

return 0;// The program ended normally
}

// *****
// Synarthsh ypologismou k-Nearest-Neighbors
/** Ypologizei thn Eukleidia apostash meta3y twn timwn tou deigmatos (testArray)
 *   kai twn timwn anaforas (dataArray)
 */
void computeKnn (double* dataArray, double* testArray, int dataSize, int testSize, int k, char** namesArray,
char** namesIdArray)
{
    // Kataskeuh pinaka ston opoio 8a apo8hkeytoun oi apostaseis meta3y timwn deigmatos kai timwn
    anaforas
    double** distanceArray;
    distanceArray = new double*[testSize];
    int i;    // metavlhth vroxou
    for (i = 0; i < testSize; ++i)
    {
        distanceArray[i] = new double[dataSize];
    }

    // Kastaskeuh pinaka o opoios 8a periexei ta index twn stoixeiwn tou pinaka dataArray pou lhf8hkan gia
    thn ka8e apostash
    int** indexArray;
    indexArray = new int*[testSize];
    for (i = 0; i < testSize; ++i)
    {
        indexArray[i] = new int[dataSize];
    }

    // Ypologismos eukleidias apostashs meta3y twn timwn toy deigmatos (testArray) kai twn timwn
    anaforas (dataArray)
    for (i=0; i <= testSize - 1; i++)
    {
        for (int j=0; j <= dataSize - 1; j++)
        {
            distanceArray[i][j] = abs(testArray[i] - dataArray[j]);    // H synarthsh 'abs' einai
            ypewrtwmenh gia typous kinhths ypodiastolhs (cmath)
            indexArray[i][j] = j; // To j apotelei to index tou stoixeiou dataArray pou lhf8hke
        }
    }

    // Euresh k mikroterwn apostaswn gia ka8e stoixeio tou testArray
    /** Meta thn klhsh ths sortFirstK, gia ka8e stoixeio tou testArray (seira)
     *   oi k prwtes sthles tou indexArray 8a periexoun tis 8eseis twn k stoixeiwn tou dataArray
     *   pou exoun dwsei tis k mikroteres eukleidies apostaseis
     */
    sortFirstK( k, testSize, dataSize, (double**)distanceArray, (int**)indexArray);
}

```

```

/**      Gia ka8e stoixeio tou deigmatos testArray a8roizoume (sum) tis emfaniseis ka8e pitch
 *      Ean to sum einai megalytero tou akeraiou merous tou k/2
 *      (perissotera apo ta misa stoixeia)
 *      tote auto to pitch epistrefetai mazi me to id tou
 */
int sum;                // Metrhts plh8ous omoiwn stoixeiwn
ofstream myOutFile;    // Egrafh se arxeio ws output stream
myOutFile.open ("output.txt"); // Onoma arxeiou
char* pitch;           // Pitch pros sygrish
char* anotherPitch;    // Ena apo ta ypoloipa pitch pros sygrish
char* id;               // Id tou pitch pleiopshfias

for (int i = 0; i <= testSize - 1; i++) // Vroxos gia kathgoriopoihsh ka8e stoixeiou tou testArray
{
    // Sygrhsh tw n k prwtwn stoixeiwn kai egrafh tou stoixeiou pleiopshfias sto output arxeio
    for (int j = 0; j <= k - 1; j++)
    {
        sum = 1;
        pitch = namesArray[(indexArray[i][j])]; // To pitch tou pinaka namesArray me index
        // thn timh tou j-stoixeiou apo ta k tou pinaka indexArray gia to stoixeio i
        for (int l = j + 1; l <= k - 1; l++) // Sygrish j stoixeiou me ta epomena
        {
            anotherPitch = namesArray[(indexArray[i][l])]; // To pitch tou pinaka
            // namesArray me index thn timh tou l-stoixeiou apo ta k tou pinaka indexArray gia to stoixeio i
            // Ean ta string einai isa (to idio pitch) au3anetai o metrhts sum
            if ( strcmp(pitch,anotherPitch) == 0 )
            {
                sum++;
            }
        }

        // Elegxos pleiopshfias kai egrash pitch kai id autou sto output arxeio se periptwsh
        // ikanopoihshs
        if ( sum > k / 2)
        {
            id = namesIdArray[ (indexArray[i][j]) ]; // To id tou pitch pleiopshfias (ths
            // prwths tou emfanishs, dld gia thn mikroterh apostash)

            // Egrafh 'pitch (id)' pleiopshfias
            myOutFile << pitch << " (" << id << ")" << endl;
            break;
        }
    } // Kathgoriopoihsh enos stoixeiou
} // Telos kathgoriopoihshs
myOutFile.close(); // Kleisimo output arxeiou
} //end of computeKnn

// *****
// Synarthsh eureshs tw n k mikroterwn apostasewn gia ka8e stoixeio tou deigmatos
/** Ta stoixeia tou pinaka indexArray einai se 1-1 antistoixia me ta stoixeia tou distanceArray
 * Oi k prwtes sthles ka8e grammhs tou pinaka distanceArray periexoun meta thn klhsh ths synarthshs
 * ta index tou pinaka dataArray kai synepws namesArray pou mas endiaferoun
 */

```

```

void sortFirstK (int k, int testSize, int dataSize, double** distanceArray, int** indexArray)
{
    int tempInt;           // Προσwrinh apo8hkeush stoixeiou tou pinaka indexArray
    double tempDouble;    // Προσwrinh apo8hkeush stoixeiou tou pinaka distanceArray

    for (int i = 0; i <= testSize - 1; i++) // Vroxos eureshs tw n k mikroterwn apostasewn gia ka8e stoixeio tou
testArray
    {
        for (int j = 0; j <= k - 1; j++) // Vroxos eureshs tw n k mikroterwn apostasewn tou stoixeiou i
        {
            for (int m = j + 1; m <= dataSize - 1; m++) // Vroxos gia th sygrish tou j stoixeiou me ola
ta epomena
            {
                if (distanceArray[i][j] > distanceArray[i][m]) // Ean vre8ei mikroterh apostash
ginetai swap
                {
                    /// Swap tw n apostasewn (distanceArray)
                    tempDouble = distanceArray[i][j];
                    distanceArray[i][j] = distanceArray[i][m];
                    distanceArray[i][m] = tempDouble;

                    /// Swap tw n index (indexArray)
                    tempInt = indexArray[i][j];
                    indexArray[i][j] = indexArray[i][m];
                    indexArray[i][m] = tempInt;
                }
            } // Epomeno stoixeio gia sygrish me to j
        } // Oi k mikroteres apostaseis tou stoixeiou j ypologistikan
    } // Oi k mikroteres apostaseis exoun vre8ei gia ola ta stoixeia
} // end of sortFirstK

```

\*Αρχεία εισόδου (δεδομένα εκπαίδευσης της κατηγοριοποίησης):

| <b>data.in</b> |
|----------------|
| 0              |
| 0              |
| 0              |
| 147            |
| 147.99495      |
| 149.2263       |
| 155.5          |
| 157.1153       |
| 158.225        |
| 165            |
| 166.52105      |
| 167.6335481    |
| 174.5          |
| 176.5029       |
| 178.0897       |
| 185            |
| 188.6794646    |
| 189.8309635    |
| 196            |

199.2082  
201.1189  
207.5  
210.37925  
213.0780521  
220  
222.15785  
222.4466  
233  
235.653  
237.8895  
247  
249.4769  
249.8826  
262  
264.6688  
266.5421  
277  
281.7017427  
282.391518  
294  
295.9899  
298.4526  
311  
314.2306  
316.45  
330  
333.0421  
335.2670962  
349  
353.0058  
356.1794  
370  
377.3589293  
379.661927  
392  
398.4164  
402.2378  
415  
420.7585  
426.1561043  
440  
444.3157  
444.8932  
466  
471.306  
475.779  
494  
498.9538  
499.7652  
524  
529.3376  
533.0842  
554

563.4034854  
564.7830361  
588  
591.9798  
596.9052  
622  
628.4612  
632.9  
660  
666.0842  
670.5341924  
698  
706.0116  
712.3588  
740  
754.7178585  
759.323854  
784  
796.8328  
804.4756  
830  
841.517  
852.3122085  
880  
888.6314  
889.7864  
932  
942.612  
951.558  
988  
997.9076  
999.5304  
1048  
1058.6752  
1066.1684  
1108  
1126.806971  
1129.566072  
1176  
1183.9596  
1193.8104  
1244  
1256.9224  
1265.8  
1320  
1332.1684  
1341.068385  
1396  
1412.0232  
1424.7176  
1480  
1509.435717  
1518.647708  
1568

1593.6656  
1608.9512  
1660  
1683.034  
1704.624417  
1760  
1777.2628  
1779.5728  
1864  
1885.224  
1903.116  
1976  
1995.8152  
1999.0608

**names.in**

NO 00  
NO 00  
NO 00  
E3 1a  
E3 1b  
E3 1c  
F3 2a  
F3 2b  
F3 2c  
F#3 3a  
F#3 3b  
F#3 3c  
G3 4a  
G3 4b  
G3 4c  
G#3 5a  
G#3 5b  
G#3 5c  
A3 6a  
A3 6b  
A3 6c  
A#3 7a  
A#3 7b  
A#3 7c  
B3 8a  
B3 8b  
B3 8c  
C4 9a  
C4 9b  
C4 9c  
C#4 10a  
C#4 10b  
C#4 10c  
D4 11a  
D4 11b  
D4 11c  
D#4 12a

D#4 12b  
D#4 12c  
E4 13a  
E4 13b  
E4 13c  
F4 14a  
F4 14b  
F4 14c  
F#4 15a  
F#4 15b  
F#4 15c  
G4 16a  
G4 16b  
G4 16c  
G#4 17a  
G#4 17b  
G#4 17c  
A4 18a  
A4 18b  
A4 18c  
A#4 19a  
A#4 19b  
A#4 19c  
B4 20a  
B4 20b  
B4 20c  
C5 21a  
C5 21b  
C5 21c  
C#5 22a  
C#5 22b  
C#5 22c  
D5 23a  
D5 23b  
D5 23c  
D#5 24a  
D#5 24b  
D#5 24c  
E5 25a  
E5 25b  
E5 25c  
F5 26a  
F5 26b  
F5 26c  
F#5 27a  
F#5 27b  
F#5 27c  
G5 28a  
G5 28b  
G5 28c  
G#5 29a  
G#5 29b  
G#5 29c  
A5 30a

A5 30b  
A5 30c  
A#5 31a  
A#5 31b  
A#5 31c  
B5 32a  
B5 32b  
B5 32c  
C6 33a  
C6 33b  
C6 33c  
C#6 34a  
C#6 34b  
C#6 34c  
D6 35a  
D6 35b  
D6 35c  
D#6 36a  
D#6 36b  
D#6 36c  
E6 37a  
E6 37b  
E6 37c  
F6 38a  
F6 38b  
F6 38c  
F#6 39a  
F#6 39b  
F#6 39c  
G6 40a  
G6 40b  
G6 40c  
G#6 41a  
G#6 41b  
G#6 41c  
A6 42a  
A6 42b  
A6 42c  
A#6 43a  
A#6 43b  
A#6 43c  
B6 44a  
B6 44b  
B6 44c  
C7 45a  
C7 45b  
C7 45c  
C#7 46a  
C#7 46b  
C#7 46c



## B2. Αυτόματη δημιουργία score.xml

### Construction.cpp

```
#include <iostream> // Header file gia leitourgies eisodou/e3odou
#include <fstream> // Header file gia xeirismo anagnwshs/egrafhs apo/se arxeia
#include <cmath> // Header file gia vasikes mathmatikes pra3eis
#include <string> // Header file gia xeirismo string

/// Omadopoihsh klasewn, antikeimenwn kai synarthsewn ypo ena onoma
using namespace std; // std: standard C++ library

/// Function prototypes
void sortFirstK (int k, int testSize, int dataSize, double** distanceArray, int** indexArray);
void computeKnn (double* dataArray, double* testArray, int dataSize, int testSize, int k, char** namesArray,
char** namesIdArray, double* intensityArray);
void constructXml (char** pitchArray, int testSize, double* intensityArray);
string chooseBeatType (double x);
void storeNote (char* element, int duration, string type, string tieprev, string tie);
bool join (double* testArray, int x, int y);

int main ()
{
    int k,i; // To k antistoixei sto k-Nearest-Neighbors kai to i apotelei metavlth vroxou

    /// -----
    /// Anagnwsh dedomenwn apo to arxeio data.in (input stream)
    ifstream inDataFile("data.in", ios::in); // To data.in periexei tis dedomenes syxnothtes

    /// Exit program if can not open file
    if(!inDataFile)
    {
        cerr << "File could not be opened!" << endl;
        exit(1);
    }

    /// Ypologismos plh8ous stoixeiwn sto arxeio data.in
    int dataSize = 0; // Mege8os arxeiou data.in
    double temp; // Proswrinh apo8hkeush stoixeiwn tou arxeiou
    for(;;)
    {
        inDataFile >> temp;
        if ( inDataFile.eof() )
        {
            break;
        }
        dataSize++;
    }

    inDataFile.clear(); // Set error state flags
    inDataFile.seekg(0, ios::beg); // Ka8orismos tou get pointer sthn arxh tou arxeiou (Offset 0 from the
beginning of the stream's buffer)

    /// -----
```

```

/// Kataskeuh pinaka dataArray kai apo8hkeush tw n stoixeiwn tou arxeiou data.in
double* dataArray;
dataArray = new double[dataSize];

for(i = 0; i < dataSize; i++)
{
    inDataFile >> dataArray[i];
}

inDataFile.close();    // Kleisimo arxeiou data. in

/// -----
/// Anagnwsh dedomenwn apo to arxeio test.in (input stream)
ifstream inTestFile("test.in", ios::in );    // To test.in perixei tis syxnothtes tou deigmatos pros
kathgoriopoihsh

/// exit program if can not open file.
if(!inTestFile)
{
    cerr << "File could not be opened." << endl;
    exit(1);
}

/// Ypologismos plh8ous stoixeiwn sto arxeio test.in
int testSize = 0; // Mege8os arxeiou test.in
char num[14];    // Proswrinh apo8hkeush stoixeiwn tou arxeiou
inTestFile >> num;    // To prwto stoixeio einai to k kai metrame ta ypoloipa stoixeia pou 8a
apo8hkeutoun stous pinakes testArray[testSize] - intensityArray[testSize]
for(;;)
{
    inTestFile >> num >> num >> num;

    if (inTestFile.eof())
    {
        break;
    }
    testSize++;
}

inTestFile.clear();    // Set error state flags
inTestFile.seekg(0, ios::beg);    // Ka8orismos tou get pointer sthn arxh tou arxeiou (Offset 0 from the
beginning of the stream's buffer)

/// To prwto stoixeio tou arxeiou einai to k (k-Nearest-Neighbors)
inTestFile >> k;

/// Kataskeuh pinaka testArray
double* testArray;
testArray = new double[testSize];

/// Kataskeuh pinaka intensityArray
double* intensityArray;
intensityArray = new double[testSize];

```

```

    /// Apouhkeush timwn pitch kai intensity stoys pinakes testArray kai intensityArray antistoixa.
    for(i = 0; i < testSize; i++)
    {
        inTestFile >> num >> num;        // To prwto stoixeio einai to time (xronos) kai den
        xrhsimopieitai - To arxeio perilamvanei triades timwn (time - pitch - intensity)

        char* und = "--undefined--";    // Se periptwsh pou den ektimatai kapoia timh to praat
        epistrefei "undefined" to opoio antistoixoume sthn timh mhden (0)
        if ( strcmp(num,und) == 0 )
            testArray[i] = 0.0;
        else
            testArray[i] = atof(num);

        inTestFile >> num;

        if (strcmp(num,und) == 0)
            intensityArray[i] = 0.0;
        else
            intensityArray[i] = atof(num);
    }

    inTestFile.close(); // Kleisimo arxeiou test.in

    /// -----
    /// Anagnwsh dedomenwn apo to arxeio names.in (input stream) me ta pitch pou antistoixoun stis
    dedomenes syxnothtes tou data.in
    ifstream inNamesFile("names.in", ios::in);

    /// Exit program if can not open file
    if(!inNamesFile)
    {
        cerr << "File could not be opened." << endl;
        exit(1);
    }

    /**
     *   Kataskeuh pinakwn
     *   namesArray[dataSize] gia apo8hkeush tw n pitch
     *   kai namesIdArray[dataSize] gia ta id autwn
     *   apo to arxeio names.in
     *   To mege8os autwn tw n pinakwn vasei antistoixias me ta stoixeia tou dataArray
     *   einai iso me dataSize
     */
    char** namesArray; // Pinakas me ta pitch tw n syxnothtw n tou dataArray
    namesArray = new char*[dataSize];
    char** namesIdArray; // Pinakas me ta id tw n pitch se 1-1 antistoixia me ton namesArray
    namesIdArray = new char*[dataSize];

    for(i = 0; i < dataSize; i++)
    {
        namesArray[i] = new char[3];    // e.g. namesArray[i] = "C#4"
        namesIdArray[i] = new char[3];  // e.g. namesIdArray[i] = "10c"
        inNamesFile >> namesArray[i] >> namesIdArray[i];
    }

```

```

    }
    inNamesFile.close(); // Kleisimo arxeiou names.in

    /// -----
    /// Klhsh synarthshs ypologismou k-Nearest-Neighbors: testArray (object classification), dataArray
(training data)
    computeKnn((double*)dataArray, (double*)testArray, dataSize, testSize, k, (char **)namesArray, (char **)
namesIdArray, intensityArray);

    /// -----
    /// When done, free memory
    delete dataArray;
    delete testArray;
    delete namesArray;
    delete namesIdArray;

    return 0;// The program ended normally
}

/// *****
/// Synarthsh ypologismou k-Nearest-Neighbors
/** Ypologizei thn Eukleidia apostash meta3y twn timwn tou deigmatos testArray
 * kai twn timwn anaforas (dataArray)
 */
void computeKnn (double* dataArray, double* testArray, int dataSize, int testSize, int k, char** namesArray,
char** namesIdArray, double* intensityArray)
{
    /// Kataskeuh pinaka ston opoio 8a apo8hkeytoun oi apostaseis meta3y timwn deigmatos kai timwn
anaforas
    double** distanceArray;
    distanceArray = new double*[testSize];
    int i; // metavlthth vroxou
    for (i = 0; i < testSize; ++i)
    {
        distanceArray[i] = new double[dataSize];
    }

    /// Kastaskeuh pinaka o opoios 8a periexei ta index twn stoixeiwn tou pinaka dataArray pou lhf8hkan gia
thn ka8e apostash
    int** indexArray;
    indexArray = new int*[testSize];
    for (i = 0; i < testSize; ++i)
    {
        indexArray[i] = new int[dataSize];
    }

    /// Ypologismos eukleidias apostashs meta3y twn timwn toy deigmatos testArray kai twn timwn anaforas
(dataArray)
    for (i=0; i <= testSize - 1; i++)
    {
        for (int j=0; j <= dataSize - 1; j++)
        {
            distanceArray[i][j] = abs(testArray[i] - dataArray[j]); // H synarthsh 'abs' einai

```

```

yperformtwmenh gia typous kinhths ypodiastolhs (cmath)
        indexArray[i][j] = j; // To j apotelei to index tou stoixeiou dataArray pou lhf8hke
    }
}

//-----
// Euresh k mikroterwn apostaswn gia ka8e stoixeiou tou testArray
/** Meta thn klhsh ths sortFirstK, gia ka8e stoixeiou tou testArray (seira)
 * oi k prwtes sthles tou indexArray 8a periexoun tis 8eseis tw n k stoixeiwn tou dataArray
 * pou exoun dwsei tis k mikroteres eukleidies apostaseis
 */
sortFirstK( k, testSize, dataSize, (double**)distanceArray, (int**)indexArray);

int sum;                // Metrhts plh8ous omoiwn stoixeiwn
char* pitch;           // Pitch pros sygrish
char* anotherPitch;    // Ena apo ta ypoloipa pitch pros sygrish
char* id;              // Id tou pitch pleiopshfias

// -----
// Egrafh se arxeio me onoma output.out tw n pitch pou prokyptoun apo thn efarmogh tou algori8mou
KNN
ofstream myOutFile;
myOutFile.open ("output.out");

// -----
// Kataskeuh pinaka pitchArray ston opoio 8a apo8hkeutoun ta pitch pou prokyptoun apo thn efarmogh
tou algori8mou KNN
char** pitchArray;
pitchArray = new char*[testSize];

// -----
/** Gia ka8e stoixeiou tou deigmatos testArray a8roizoume (sum) tis emfaniseis ka8e pitch
 * Ean to sum einai megalytero tou akeraiou merous tou k/2
 * (perissotera apo ta misa stoixeia)
 * tote auto to pitch epistrefetai mazi me to id tou
 */
for (int i = 0; i <= testSize - 1; i++) // Vroxos gia kathgoriopoishh ka8e stoixeiou tou testArray
{
    // Sygrhsh tw n k prwtwn stoixeiwn kai egrafh tou stoixeiou pleiopshfias sto output arxeio
    for (int j = 0; j <= k - 1; j++)
    {
        sum = 1;
        pitch = namesArray[(indexArray[i][j])]; // To pitch tou pinaka namesArray me index
        thn timh tou j-stoixeiou apo ta k tou pinaka indexArray gia to stoixeiou i
        for (int l = j + 1; l <= k - 1; l++) // Sygrish j stoixeiou me ta epomena
        {
            anotherPitch = namesArray[(indexArray[i][l])]; // To pitch tou pinaka
            namesArray me index thn timh tou l-stoixeiou apo ta k tou pinaka indexArray gia to stoixeiou i

            // Ean ta string einai isa (to idio pitch) au3anetai o metrhts sum
            if ( strcmp(pitch,anotherPitch) == 0 )
            {
                sum++;
            }
        }
    }
}

```

```

    }

    // Elegxos pleiopshfias kai egrash pitch kai id autou sto output arxeio se periptwsh
    ikanopoihshs
    if ( sum > k / 2)
    {
        id = namesIdArray[ (indexArray[i][j]) ];    // To id tou pitch pleiopshfias (ths
    prwths tou emfanishs, dld gia thn mikroterh apostash)

        // Egrafh 'pitch (id)' pleiopshfias sto arxeio output.out
        myOutFile << pitch << " (" << id << ')' << endl;

        // Apo8hkeush tou pitch pleiopshfias ston pinaka pitchArray
        pitchArray[i] = new char[3];
        pitchArray[i] = pitch;

        break; // Epomeno pitch
    }
    // end of kathgoriopoihsh enos stoixeiou
} // end of kathgoriopoihsh olwn tw n stoixeiwn
myOutFile.close(); // Kleisimo arxeiou output.out

// -----
// Klhsh synarthshs kataskeuhs tou arxeiou score.xml
constructXml ((char**) pitchArray, testSize, intensityArray);
} //end of computeKnn

// *****
// Synarthsh eureshs tw n k mikroterwn apostasewn gia ka8e stoixeio tou deigmatos
/** Ta stoixeia tou pinaka indexArray einai se 1-1 antistoixia me ta stoixeia tou distanceArray
 * Oi k prwths sthles ka8e grammhs tou pinaka distanceArray periexoun meta thn klhsh ths synarthshs
 * ta index tou pinaka dataArray kai synepw s namesArray pou mas endiaferoun
 */
void sortFirstK (int k, int testSize, int dataSize, double** distanceArray, int** indexArray)
{
    int tempInt; // Proswrinh apo8hkeush stoixeiou tou pinaka indexArray
    double tempDouble; // Proswrinh apo8hkeush stoixeiou tou pinaka distanceArray

    for (int i = 0; i <= testSize - 1; i++) // Vroxos eureshs tw n k mikroterwn apostasewn gia ka8e stoixeio tou
testArray
    {
        for (int j = 0; j <= k - 1; j++) // Vroxos eureshs tw n k mikroterwn apostasewn tou stoixeiou i
        {
            for (int m = j + 1; m <= dataSize - 1; m++) // Vroxos gia th sygrish tou j stoixeiou me ola
ta epomena
            {
                if (distanceArray[i][j] > distanceArray[i][m]) // Ean vre8ei mikroterh apostash
ginetai swap
                {
                    // Swap tw n apostasewn (distanceArray)
                    tempDouble = distanceArray[i][j];
                    distanceArray[i][j] = distanceArray[i][m];
                }
            }
        }
    }
}

```

```

        distanceArray[i][m] = tempDouble;

        /// Swap twn index (indexArray)
        tempInt = indexArray[i][j];
        indexArray[i][j] = indexArray[i][m];
        indexArray[i][m] = tempInt;
    }
    } // Epomeno stoixeio gia sygrish me to j
} // Oi k mikrotres apostaseis tou stoixeiou j ypologistikan
} // Oi k mikrotres apostaseis exoun vre8ei gia ola ta stoixeia
// end of sortFirstK

/// *****
/// Synarthsh kataskeuhs arxeiou score.xml
/** Dexetai ws eisodous tous pinakes pitchArray (apotelesma computeKNN) kai intensityArray (apo arxeio test.in)
 * kai to mege8os twn pinakwn autwn
 */
void constructXml (char** pitchArray, int testSize, double* intensityArray)
{
    /// Anagnwsh dedomenwn apo to arxeio input.in
    /** To input.in arxeio perixei times vasikwn parametrwn gia th dhmiourgia tou xml arxeiou :
 * divisions : How many divisions per quarter note for a duration (int)
 * min : H megisth diaresh tou beatType sthn hxografhsh (int) (min =
minimum type / beatType)
 * beats : Time signature numerator (int)
 * beatType : Time signature denominator (int)
 * sign : Clef sign. Values: G, F, C, percussion, TAB, none (char)
 * line : Staff line numbered from bottom to top for clefs (int)
 * tempo : Tempo in quarter notes per minute (int)
 * moment : Ean h hxografhsh 3ekina me ellipes metro orizoume thn akrivh a3ia
autou (double)
 * instrument : Full name of instrument (eg. "Clarinet") (char)
 * program : MIDI program number from 1 to 128 (eg. 72) (int)
 */
    ifstream inputFile("input.in", ios::in);

    /// Exit program if can not open file
    if(!inputFile)
    {
        cerr << "File could not be opened!" << endl;
        exit(1);
    }

    int divisions, min, beats, beatType, line, tempo, program;
    double tempoPerSecond, SecondsPerBeatType, moment;
    char sign, instrument[10];

    inputFile >> divisions; // How many divisions per quarter note for a duration
    inputFile >> min; // TH megisth diaresh tou beatType sthn hxografhsh - eg. ean beatType = 4
(quarter) tote gia minimum to eighth(8) to min = 2

    inputFile >> beats; // Time signature numerator
    inputFile >> beatType; // Time signature denominator

```

```

inputFile >> sign;           // Clef sign. Values: G, F, C, percussion, TAB, none
inputFile >> line;           // Staff line numbered from bottom to top for clefs

inputFile >> tempo;           // Tempo in quarter notes per minute (8ewrw : "in beatType notes" ean
byteType diaforetiko tou quarter)
tempoPerSecond = (double) tempo / 60; // Tempo in quarter notes per second (Omoia.. beatType notes
per second)
SecondsPerBeatType = 1.0 / tempoPerSecond; // Seconds per quarter (Omoia.. seconds per
beatType)

inputFile >> moment; // Ean to kommati 3ekina me ellipes metro orizoume thn akrivh a3ia autou

inputFile >> instrument; // Full name of instrument
inputFile >> program; // MIDI program number

inputFile.close(); // Kleisimo arxeiou input.in

/// -----
/// Dhmiourgia arxeiou score.xml (output stream)
ofstream myXmlFile;
myXmlFile.open ("score.xml", ofstream::out);

/// Exit program if can not open file
if(!myXmlFile)
{
    cerr << "File could not be opened!" << endl;
    exit(1);
}

/// -----
/// Egrafh sto arxeio score.xml tw n arxikwn parametrwn
myXmlFile << "<?xml version='1.0' standalone='no'?">"
    << "\n<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 1.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd ">"
    << "\n\n<score-partwise>"
    << "\n  <identification>"
    << "\n      <encoding>"
    << "\n          <software>Score Processing Library</software>"
    << "\n      </encoding>"
    << "\n  </identification>"
    << "\n  <part-list>"
    << "\n      <score-part id='P1'>"
    << "\n          <part-name/>"
    << "\n          <score-instrument id='P1-I1'>"
    << "\n              <instrument-name>" << instrument <<
"</instrument-name>"
    << "\n          </score-instrument>"
    << "\n          <midi-instrument id='P1-I1'>"
    << "\n              <midi-channel>1</midi-channel>"
    << "\n              <midi-program>" << program << "</midi-
program>"
    << "\n          </midi-instrument>"
    << "\n      </score-part>"

```



```

        << "\n </part-list>"
        << "\n <part id=\"P1\">"
        << "\n         <measure number=\"1\">"
        << "\n         <attributes>"
    << "\n         <divisions>" << divisions << "</divisions>"
    << "\n         <key>"
    << "\n         <fifths>0</fifths>"
    << "\n         </key>"
    << "\n         <time>"
    << "\n         <beats>" << beats << "</beats>"
    << "\n         <beat-type>" << beatType <<
"</beat-type>"
    << "\n         </time>"
    << "\n         <clef>"
    << "\n         <sign>" << sign << "</sign>"
    << "\n         <line>" << line << "</line>"
    << "\n         </clef>"
    << "\n         </attributes>"
    << "\n         <sound tempo=\"\" << tempo << "\"/>";

    /// -----
    /// Anagnwrish ths melwdias - Kataskeuh pinaka xmlPitchArray me omadopoihsh tw n dedomenwn tou
PitchArray ana groupMinNote

    int next; // Metavlthh vroxou
    int position = 0; // Index tou pinaka ston opoio 8a apo8hkeutoun oi notes
    char* element; // Stoixeio tou pitchArray pros sygrish me ta ypoloipa
    double timeStep = 0.01; // time step tw n stoixeiwn tou pitchArray (parametros time tou praat sto fixed)
    int count; // Metrthhs gia to synolo tw n omoiwn timwn

    double groupBeatType = SecondsPerBeatType / timeStep; // (seconds per beatType / seconds per
step) = ari8mos timwn pou apoteloun ena beatType
    int groupMinNote = (int) groupBeatType / min; // ari8mos timwn pou apoteloun mia minimum duration
nota

    int elements = ceil((double) testSize/groupMinNote); // synolikos ari8mos apo notes pros anazhthsh
( ceil: Round up value <- testSize/groupMinNote oxi panta teleia diaresh!)

    /// Kataskeuh pinaka ston opoio 8a apo8hkeutoun oi notes pou prokryptoun omadopoiwntas ta dedomena
tou PitchArray ana groupMinNote
    char** xmlPitchArray;
    xmlPitchArray = new char*[elements];

    /// Omadopoihsh dedomenwn tou pitchArray ana groupMinNote
    for (next = 0; next <= (testSize - groupMinNote); next += groupMinNote) // Vroxos omadopoihshs
    {
        for (int i = 0; i <= (groupMinNote - 1); i++) // Vroxos evreshs tou pitch mias notas
        {

            element = pitchArray[next+i]; // Stoixeio pros sygrish
            count = 1; // Prwth emfanish tou stoixeiou element

            /// Vroxos sygrishs me ta ypoloipa ths idias omadas

```

```

        for (int j = i + 1; j <= (groupMinNote - 1); j++)
        {
            if ( strcmp(element,pitchArray[next+j]) == 0 )      // Sygrish tou pitch me to
            {
                count++;      // Ean ta pitch einai idia au3anetai o metrhts count
            }
        }      // Telos vroxou sygrishs

        /// Pleiopshfia tou stoixeiou element gia thn dedomenh omada timwn
        if ( count >= (groupMinNote / 2) )
        {
            xmlPitchArray[position] = element;      // Apo8hkeush autou tou pitch
            break; // Epomenh omada timwn
        }

        /// Den yparxei stoixeio pleiopshfias
        if ( i == (groupMinNote / 2) + 1 )
        {
            xmlPitchArray[position] = "Bad"; // Apo8hkeush timhs "Bad" gia auth thn
            break; // Epomenh omada timwn
        }
    }      // end of evresh enos pitch
    position++;      // index tou pinaka XmlPitchArray se epomeno stoixeio
}

/// To tempo mias performance den einai panta akrives kai h diaresh testSize/groupMinNote prokypetei
// Evresh notas vasei twv timwn pou apomenoun ( < minDuration) - Omoia me prin
for (int i = 0; i <= (testSize - next - 1); i++)
{
    element = pitchArray[next+i];
    count = 1;

    for (int j = i + 1; j <= (testSize - next - 1); j++)
    {
        if ( strcmp(element,pitchArray[next+j]) == 0 )
        {
            count++;
        }
    }

    if ( count >= ((testSize - next)/ 2))
    {
        xmlPitchArray[position] = element;
        break;
    }

    if ( i == ((testSize - next)/ 2) + 1 )
    {
        xmlPitchArray[position] = "Bad";
        break;
    }
}

```

```

    }
}

///
-----
/// Dhmiourgia arxeiou note.out (output stream) opou 8a apo8hkeutoun oi plhrofories me ta pitch kai tis
diarkeies (alfari8mhtikh akolou8ia apo notes) opws apo8hkeuontai sto score.xml (gia dior8wsh-a3iologhsh)
ofstream noteFile;
noteFile.open ("note.out", ofstream::out);

/// -----
/// Egrafh sto score.xml ths melwdias

int j = 0;          // Metavlhth eswterikou vroxou
int duration;      // Note duration in divisions
int measureNumber = 1; // measure-numbering
double pitchType;  // (double) Pitch type - orisma ths chooseBeatType (timh anaforas to
whole = 1.0)
string type;        // (String) Pitch type - apotelesma ekteleshs chooseBeatType, Values
= {256th, 128th, 64th, 32nd, 16th, eighth, quarter, half, whole, breve, long}
string tieprev, tie; // Tied notas se periptwsh allaghs metrou (opou xreiazetai)

/// Vroxos kataskeuhs ths melwdias
for ( int i = 0; i < elements; i = j )
{
    /** O pinakas XmlPitchArray periexei ta pitch pou proekypsan omadopoiwntas ta dedomena
vasei ths elaxisths diarkeias pou anazhtoume
    * Omadopoioume ta omoia pitch diathrwntas ston metrth count to synoliko ari8mo autwn
    */
    element = xmlPitchArray[i];
    count = 1;

    /// Arxikopoihsh twv tieprev kai tie gia endexomenh enwsh ths idias notas se allagh metrou
    tieprev = "null";
    tie = "null";

    for ( j = i + 1; j <= elements; j++)
    {
        int x = (j - 1) * groupMinNote; // arxiko index pinaka intensityArray pros elegxo - arxh
prwths omadas elaxisths diarkeias
        int y = (j + 1) * groupMinNote; // teliko index pinaka intensityArray pros elegxo - telos
deuterhs omadas elaxisths diarkeias

        if ( ( j < elements) && (strcmp (element,xmlPitchArray[j]) == 0) && (join(intensityArray, x,
y))) // Au3hsh metrth ean to element den htan to teleutaio pitch kai einai idio me to epomeno tou meta apo
elegxo ths perivallousas (klhsh ths join)
        {
            count++;
        }

        else // Diaforetika proxwroume sthn egrafh autou sto score.xml

```

```

        {
            /// Υπολογισμός pitchType
            /** count: Αριθμός από minimum type pitch (e.g. 4 * eighth = 1 * half)
             * min: μέγιστη υποδιαίρεση του beatType (e.g. minimum type to eighth <->
beatType=4 (quarter), min=2)
            */
            pitchType = (double) count / min; // e.g. pitchType = 4/2 = 2

            pitchType /= beatType; // Διαίρεση με beatType: Το ορίσματος της
chooseBeatType ορίζεται με shmeio αναφοράς το whole = 1.0 (e.g. 2/4 = 0.5 ---από chooseType---> half)
            type = chooseBeatType(pitchType); // (String) note type

            pitchType *= beatType; // Επαναφορά pitchType
            duration = pitchType*divisions; // Note duration "in divisions"

            /** Στο αρχείο note.out αποθηκεύεται η ακολουθία από ελαστικές διάρκειες pitch
synodeuomena από έναν 8ετικο αριθμό
            * Ο αριθμός αυτός αυξάνει σε περπτώση ιδίου pitch το όποιο "denetai"
me to prohgomeno tou
            * Διαφορετικά έχει την τιμή 1
            */
            for (int k = 1; k <= count; k++)
            {
                noteFile << element << "-" << k << endl; // e.g. "C4-1" και "C4-1" δύο
notes με pitch=C4 με 3εξίστα duration=1 ΚΑΙ "C4-1" "C4-2" μία nota C4 με duration=2 (διπλασία διάρκειας από
tis prohgomenes η tied notes)
            }

            /// Περίπτωση 1: Η nota έχει διάρκεια που δεν 3επερνά το μέτρο

            if ((moment + pitchType) <= beats )
            {

                moment += pitchType; // Χρονική στιγμή από την αρχή του μέτρου
meta την εγραφή της notas

                /// Εγραφή της notas στο αρχείο score.xml με κλήση της storeNote
myXmlFile.close(); // Κλείσιμο αρχείου πριν την κλήση της
storeNote η οποία πραγματοποιεί εγραφή στο ίδιο αρχείο
                storeNote (element, duration, type, tieprev, tie);
myXmlFile.open ("score.xml", ofstream::app); // Ανοίγματος του
arxeiou score.xml για προσθήκη στοιχείων (append)

                /// Αλλάξη μέτρου στο score.xml
                if ((beats - moment < 1e-014) && (j != elements)) // Στήν περίπτωση
ypodiareshs se trixa (e.g.min=6) εισάγεται ένα sfalma στην τιμή moment της ταξέως του 1e-015
            {
                myXmlFile << "\n </measure>"
                << "\n <measure
number=\"\" << ++measureNumber << "\>";

                moment = 0.0; // Νέο μέτρο
            }
        }

```

```

    }

    /// Periptwsh 2: H diardeia ths notas 3eperna to metro sto opoio vriskomaste

    else
    {
        double temp = beats - moment; // Diardeia ths notas pou xwra sto
arxiko metro
        double temp2 = pitchType - temp; // Ypoloipo ths diardeia ths
notas pou 8a apouhkeutei ston neo metro

        /// Sthn periptwsh ypodiareshs diardeias se triixa (e.g.min=6) eisagetai
ena sfalma sthn timh moment, kai ara kai temp, temp2. Auto lynei to provlhma gia triixa apo notes elaxisths
diardeias

        if ((temp > beatType/min) && (temp < 2*beatType/min))
        {
            temp = beatType/min;
            temp2 = pitchType - temp;
        }

        tie = "start"; // Arxh enwshs notas

        /// To temp ginetai iso h mikrotero tou 0 afou h nota exei katagrafei
while (temp > 0)
    {
        temp /= beatType; // Diaresh me beatType: To orisma
ths chooseBeatType orizetai me shmeio anaforas to whole = 1.0
        type = chooseBeatType(temp); // (String) Note type
        temp *= beatType; // Epanafora temp
        duration = temp*divisions; // Note duration "in
divisions"

        moment += temp; // Xronikh stigmh apo thn arxh tou
metrou meta thn egraphh ths notas

        /// Egraphh ths notas sto arxeio score.xml me klhsh ths
storeNote
        myXmlFile.close(); // Kleisimo arxeiou prin thn klhsh ths
storeNote h opoia pragmatopoiei egraphh sto idio arxeio
        storeNote (element, duration, type, tieprev, tie);
        myXmlFile.open ("score.xml", ofstream::app); // Anoigma
tou arxeiou score.xml gia pros8hkh stoixeiwn (append)

        /// Allagh metrou sto arxeio score.xml
        if (beats - moment < 1e-014) // Sthn periptwsh
ypodiareshs se triixa (e.g.min=6) eisagetai ena sfalma sthn timh moment ths ta3ews tou 1e-015
        {
            myXmlFile << "\n </measure>"
number="\n" << ++measureNumber << "\n";
            << "\n <measure
            moment=0; // Neo metro
        }
    }

```

```

// Diarkeia tou ypoloipou ths notas akoma megalyterh tou
metrou
if (temp2 > beats)
{
    // Apo8hkeush ths notas me diarkeia ish me ena
    metro (beats)
    double metro = (double) beats / beatType; // Diaresh
me beatType: To orisma ths chooseBeatType orizetai me shmeio anaforas to whole = 1.0
    type = chooseBeatType(metro); // (String) Note type

    duration = beats*divisions; // Note
duration "in divisions"

    // Egrafh ths notas sto arxeio score.xml me klhsh ths
storeNote
    myXmlFile.close(); // Kleisimo arxeiou prin thn
klhsh ths storeNote h opoia pragmatopoei egrafh sto idio arxeio
    tieprev = "stop"; // Kleisimo prohgomens enwshs
    tie = "start"; // Arxh epomenhs
    storeNote (element, duration, type, tieprev, tie);
    tie = "null"; // h nota 8a exei olokhrw8ei
(timh arxikopoihs)
    tieprev = "stop"; // Kleisimo
    myXmlFile.open ("score.xml", ofstream::app); //
Anoigma tou arxeiou score.xml gia pros8hkh stoixeiwn (append)

    // Allagh metrou sto arxeio score.xml
    myXmlFile << "\n </measure>"
    << "\n <measure
number=\"\" << ++measureNumber << "\">";

    moment = 0; // Neo metro
}
else if (temp2 > 0)
{
    temp2 /= beatType; // Diaresh me beatType: To
orisma ths chooseBeatType orizetai me shmeio anaforas to whole = 1.0
    type = chooseBeatType(temp2); // (String) Note type
    temp2 *= beatType; // Epanafora temp2
    duration = temp2*divisions; // Note
duration "in divisions"

    // Egrafh ths notas sto arxeio score.xml me klhsh ths
storeNote
    myXmlFile.close(); // Kleisimo arxeiou prin thn
klhsh ths storeNote h opoia pragmatopoei egrafh sto idio arxeio
    tieprev = "stop"; // Telos prohgomens
    tie = "null"; // Timh opws arxikopoihs
    storeNote (element, duration, type, tieprev, tie); //
Enallagh tie <-> tieprev gia olokhrwsh
    myXmlFile.open ("score.xml", ofstream::app); //
Anoigma tou arxeiou score.xml gia pros8hkh stoixeiwn (append)

```

```

thn temp2
moment = temp2; // To metro meta thn apo8hkeush

// Allagh metrou sto score.xml
if ((beats - moment < 1e-014) && (j != elements)) //
Sthn periptwsh ypodiareshs se triixa (e.g.min=6) eisagetai ena sfalma sthn timh moment ths ta3ews tou 1e-015
{
    myXmlFile << "\n
    </measure>" << "\n <measure
number=\"\" << ++measureNumber << "\">";
    moment = 0.0; // Neo metro
}
break; // Epomenh nota
}
// Periptwsh sthn opoia h nota apaitei ki allo/alla metro/a -
xwrismos omoia me prin
temp = temp2 - beats;
// Ean h temp exei diarkeia megalyterh tou metrou
if (temp > beats)
{
    temp2 = temp - beats;
    temp = beats;
    tie = "start";
    tieprev = "stop";
}
// Ean mikroterh tou metrou menei mono h egrafh auths
else
{
    temp2 = 0;
    tie = "null";
    tieprev = "stop";
}
} // end of while gia periptwsh notas me diarkeia megalyterh tou
metrou
tie = "null"; // Timh opws arxikopoihsh
tieprev = "null"; // Timh opws arxikopoihsh
} // end of else if egrafhs notas diarkeias megalyterhs tou metrou
break; // Epomenh nota
} // end of else egrafhs mias notas
} // end of eswteriko for loop
} // oles oi notes exoun apo8hkeutei sto arxeio score.xml

// -----
// Olokhrwsh tou arxeiou score.xml
myXmlFile << "\n <barline location=\"right\">
<< "\n <bar-style>light-heavy</bar-style>"

```

```

        << "\n                </barline>"
        << "\n                </measure>"
        << "\n    </part>"
        << "\n</score-partwise>";

    myXmlFile.close();                // Kleisimo arxeiou score.xml
    delete xmlPitchArray;            // Free memory pointed to by XmlPitchArray
}

/// *****
/// Synarthsh epistrofhs (string) type of note
/** Dextetai ws eisodo enan double pou antistoixei sth diarkeia ths notas (eg. whole = 1/1, quarter = 1/4)
 *     Elegxetai kai h periptwsh parestigmenhs notas (one dot only) kai trihxou (synolikhs diarkeias eighth kai
quarter).
 */
string chooseBeatType (double x)
{
    if (x == 1.0/256)
        return("256th");
    else if (x == 1.0/128 || x == 1.0/128 + 1.0/256)
        return("128th");
    else if (x == 1.0/64 || x == 1.0/64 + 1.0/128)
        return("64th");
    else if (x == 1.0/32 || x == 1.0/32 + 1.0/64)
        return("32nd");
    else if (x == 1.0/16 || x == 1.0/24 || x == 1.0/16 + 1.0/32)
        return("16th");
    else if (x == 1.0/8 || x == 1.0/12 || x == 1.0/8 + 1.0/16)
        return("eighth");
    else if (x == 1.0/4 || x == 1.0/4 + 1.0/8)
        return("quarter");
    else if (x == 1.0/2 || x == 1.0/2 + 1.0/4)
        return("half");
    else if (x == 1.0/1 || x == 1.0/1 + 1.0/2)
        return("whole");
    else if (x == 2.0/1 || x == 3.0/1)
        return("breve");
    else
        return("long");
} // end of chooseBeatType

/// *****
/// Synarthsh egrafhs sto arxeio score.xml tou pitch element ws nota me symvolo type kai diarkeia duration
void storeNote (char* element, int duration, string type, string tieprev, string tie)
{
    ofstream myXmlFile;                // Egraphh sto arxeio score.xml (output stream)
    myXmlFile.open ("score.xml", ofstream::app); // Anoigma tou arxeiou score.xml gia pros8hkh
stoixeiwn (append)
}

```



```

myXmlFile << "\n" <note>;

/// Ean to stoixeio element exei thn timh "NO" h "Bad" tote pros8etoume thn atistoixhs diarkeias paush
if ( strcmp(element,"NO") == 0 || strcmp(element,"Bad") == 0 )
{
    myXmlFile << "\n" <rest/>;
}
else // Egrafh tou pitch
{
    /// Egrafh prwtou xarakthra -> step (eg. 'A' ean "A#4" h "A4")
    myXmlFile << "\n" <pitch>"
        << "\n" <step>" << element[0] <<
"</step>";

    /// Egrafh tou teleutaiou xarakthra -> octave ('4') kai ths # mesw tou alter = +1 (eg. "A#4")
    if ( element[1] == '#' )
    {
        myXmlFile << "\n" <alter>+1</alter>"
            << "\n" <octave>" <<
element[2] << "</octave>"
            << "\n" </pitch>";
    }
    /// Egrafh tou teleutaiou xarakthra -> octave (eg. '4' ean "A4")
    else
    {
        myXmlFile << "\n" <octave>" << element[1] <<
"</octave>"
            << "\n" </pitch>";
    }
} // end of egrafh pitch

/// Egrafh diarkeias ths notas/paushs
myXmlFile << "\n" <duration>" << duration << "</duration>";

/// Egrafh typou ths notas/paushs
/// Pi8anh anagh enwshs ths notas se allagh metrou (ean den einai paush!)
if (((tie != "null") || (tieprev != "null")) && !(strcmp(element,"NO") == 0) || (strcmp(element,"Bad") == 0))
{
    /// Enwsh me prohgomenh kai epomenh
    if ((tieprev != "null") && (tie != "null"))
    {
        myXmlFile << "\n" <tie type="" << tieprev << "\/>"
            << "\n" <tie type="" << tie << "\/>"
            << "\n" <type>" << type << "</type>"
            << "\n" <notations>"
            << "\n" <tied type="" <<
tieprev << "\/>"
            << "\n" <tied type="" << tie
<< "\/>"
            << "\n" </notations>";
    }

    /// Enwsh me epomenh( tie == "start") h prohgomenh( tieprev == "stop")
    else

```

```

        {
            if (tieprev != "null")
                tie = tieprev;

            myXmlFile << "\n" << "tie type=\"" << tie << "\"/>"
                << "\n" << "<type>" << type << "</type>"
                << "\n" << "<notations>"
                << "\n" << "<tied type=\"" << tie
<< "\"/>"
                << "\n" << "</notations>";

        }
    }

else
    myXmlFile<< "\n" << "<type>" << type << "</type>";

myXmlFile << "\n" << "</note>";

myXmlFile.close(); // Kleisimo arxeiou score.xml
} // end of storeNote

/// *****
/// Synarthsh elegxou mh shmantikhs ptwshs ths perivallousas tou shmatos
bool join (double* testArray, int x, int y)
{
    double tempMin = 100; // Proswrinh apo8hkeush elaxisths timhs
    double tempMax = 0; // Proswrinh apo8hkeush megisths timhs

    /// Den elegxw tis prwtes kai teleutaies times ka8ws den periexoun shmantikh plhroforia
    int first = x + (y - x) / 4;
    int last = y - (y - x) / 4;

    int mini = first; // Arxikopoihsh index elaxisths timhs

    for (int i = first; i <= last; i++)
    {
        /// Euresh elaxisths timhs
        if (testArray[i] < tempMin)
        {
            tempMin = testArray[i];
        }

        /// Euresh megisths timhs
        if (testArray[i] > tempMax)
        {
            tempMax = testArray[i];
        }
    }

    /// Syn8hkh apofashs
    /** Ean (tempMin < 50) prokeitai gia paush opote join = true
    * 8ewrw shmantikh ptwsh > 6db opote join = false
    */
}

```

```

    if ((tempMin > 50) && ((tempMax - tempMin) > 6))
        return false;
    else
        return true;
}

```

\*Τα αρχεία εισόδου name.in και data.in είναι τα ίδια με πριν (B1). Συμπληρωματικά απαιτείται το input.in με τις βασικές παραμέτρους της ηχογράφησης για τη δημιουργία του αρχείου score της αυτόματης καταγραφής (B2)

### B3. Αξιολόγηση αρχείου εξόδου

```

Levenshtein.cpp

#include <iostream> // Header file gia leitourgies eisodou/e3odou
#include <fstream> // Header file gia xeirismo anagnwshs/egrafhs apo/se arxeia
#include <string> // Header file gia xeirismo string

// Omadopoihsh klasewn, antikeimenwn kai synarthsewn ypo ena onoma
using namespace std; // std: standard C++ library

// Synarthsh ypologismou ths elaxisths timhs 3 orismatwn
int minimum (int a, int b, int c)
{
    int min = a;
    if (b<a)
        min = b;
    if (c<min)
        min = c;

    return min;
}

int main()
{
    // -----
    // Anagnwsh dedomenwn apo to arxeio note.out (input stream)

    ifstream noteFile("note.out", ios::in);

    // Exit program if can not open file
    if(!noteFile)
    {
        cerr << "File could not be opened!" << endl;
        exit(1);
    }

    // Ypologismos plh8ous stoxeiwn sto arxeio note.out
    int n = 0; // n : Mege8os arxeiou note.out
    char* temp; // Proswrinh apo8hkeush stoxeiwn tou arxeiou
    for(;;)
    {
        temp = new char[5]; // e.g. temp = "C#4-1"
    }
}

```

```

        noteFile >> temp;
        if ( noteFile.eof() )
        {
            break;
        }
        n++;
    }

    noteFile.clear(); // Set error state flags
    noteFile.seekg(0, ios::beg); // Καθ'ορισμός του get pointer sthn arxh tou arxeiou (Offset 0 from the
beginning of the stream's buffer)

//-----
    /// Kataskeuh pinaka noteArray kai apo8hkeush tw n stoixeiwn tou arxeiou note.out
    char** noteArray;
    noteArray = new char*[n];

    for(int i = 0; i < n; i++)
    {
        noteArray[i] = new char[5]; // e.g. temp = "C#4-1"
        noteFile >> noteArray[i];
    }

    noteFile.close(); // Kleisimo arxeiou note.out

    /// -----
    /// Anagnwsh dedomenwn apo to arxeio pitch.in (input stream) pou antistoixei sto score.xml anaforas
    /** To arxeio pitch.in perilamvanei thn akolou8ia apo elaxisths diarkeias pitch synodeuomena apo enan
8etiko ari8mo (alfari8mhtikh akolou8ia apo notes)
    * O ari8mos autos au3anei otan to idio pitch "denetai" me to prohgomeno tou
    * Diaforetika exei thn timh 1
    */
    /** Shmantikh parathrhsh!! To arxeio auto diaferei apo to omonymo arxeio ths diadikasias Re-
transcription
    * O 8etikos ari8mos, se periptwseis diagrafs/eisagwghs, den au3anetai me ton anamenomeno
tropo
    * alla akolou8ei thn ari8mhsh tou note.out arxeiou otan to pitch einai swsto
    * e.g. akolou8ia pitch.in "C4-1 C4-1 C4-2" gia akolou8ia note.out "C#4-1 C4-1 C4-2" otan h
anaforas einai "C4-1 C4-2 C4-3"
    */
    ifstream dataFile("pitch.in", ios::in);

    /// Exit program if can not open file
    if(!dataFile)
    {
        cerr << "File could not be opened!" << endl;
        exit(1);
    }

    /// Ypologismos plh8ous stoixeiwn sto arxeio pitch.in
    int m = 0; // m : Mege8os arxeiou pitch.in
    for(;;)
    {
        temp = new char[5]; // e.g. temp = "C#4-1"

```

```

        dataFile >> temp;
        if ( dataFile.eof() )
        {
            break;
        }
        m++;
    }

    dataFile.clear(); // Set error state flags
    dataFile.seekg(0, ios::beg); // Καθ'ορισμος του get pointer sthn arxh tou arxeiou (Offset 0 from the
beginning of the stream's buffer)

//-----
    /// Kataskeuh pinaka dataArray kai apo8hkeush tw'n stoixeiwn tou arxeiou note.out
    /** To arxeio note.out perilamvanei thn akolou8ia apo elaxisths diarkeias pitch synodeuomena apo enan
8etiko ari8mo
    *   O ari8mos autos au3anei otan to idio pitch "denetai" me to prohgooumeno tou
    *   Diaforetika exei thn timh 1
    */
    char** dataArray;
    dataArray = new char*[m];

    for(int i = 0; i < m; i++)
    {
        dataArray[i] = new char[5]; // e.g. temp = "C#4-1"
        dataFile >> dataArray[i];
    }

    noteFile.close(); // Kleisimo arxeiou note.out

    /// Computing Levenshtein distance

    /// d is a table with n+1 rows and m+1 columns
    int** d;
    d = new int*[n+1];
    for (int i = 0; i < n + 1; i++)
        d[i] = new int[m+1];

    for (int i = 0; i < n + 1; i++)
        d[i][0] = i; // deletion
    for (int j = 0; j < m + 1; j++)
        d[0][j] = j; // insertion

    for (int j = 1; j < m + 1; j++)
    {
        for (int i = 1; i < n + 1; i++)
        {
            if (strcmp(noteArray[i-1],dataArray[j-1]) == 0) // Sygrish tw'n string pou
antiproswpeuoun ta pitch kai tis diarkeies gia sta score apo automath katagrafh kai anaforas
                d[i][j] = d[i-1][j-1];
            else
                d[i][j] = minimum
                (
                    d[i-1][j] + 1, // deletion

```

```

        d[i][j-1] + 1, // insertion
        d[i-1][j-1] + 1 // substitution
    );
    }
}

// Epistrophh tous sfalmatos, tw n kai m kai tou posostou epityxias ths automaths katagrafhs partitouras
cout << d[n][m] << " " << "(" << n << ", " << m << ")" << endl
<< 100*(double(m - d[n][m]))/m << "%" << endl;
system("pause");

//return d[m][n];
}

```

\*Δέχεται ως εισόδους δύο αρχεία. Το note.out που έχει προκύψει από την εφαρμογή του Construction.cpp (B2) και του pitch.in αρχείου μετά τη διόρθωση της ακολουθίας του note.out.

#### B4. Διόρθωση αρχείου εξόδου

```

Edited.cpp

#include <iostream> // Header file gia leitourgies eisodou/e3odou
#include <fstream> // Header file gia xeirismo anagnwshs/egrafh apo/se arxeia
#include <cmath> // Header file gia vasikes mathmatikes pra3eis
#include <string> // Header file gia xeirismo string

// Omadopoihsh klasewn, antikeimenwn kai synarthsewn ypo ena onoma
using namespace std; // std: standard C++ library

// Function prototypes
string chooseBeatType (double x);
void storeNote (char* element, int duration, string type, string tieprev, string tie);
bool join (double* testArray, int x, int y);

int main ()
{
    // Anagnwsh dedomenwn apo to arxeio input.in
    /** To input.in arxeio periexei times vasikwn parametrwn gia th dhmiourgia tou xml arxeiou :
    * divisions : How many divisions per quarter note for a duration (int)
    * min : H megisth diairesi tou beatType sthn hxografhsh (int) (min =
minimum type / beatType)
    * beats : Time signature numerator (int)
    * beatType : Time signature denominator (int)
    * sign : Clef sign. Values: G, F, C, percussion, TAB, none (char)
    * line : Staff line numbered from bottom to top for clefs (int)
    * tempo : Tempo in quarter notes per minute (int)
    * moment : Ean h hxografhsh 3ekina me ellipes metro orizoume thn akrivh a3ia
autou (double)
    * instrument : Full name of instrument (eg. "Clarinet") (char)
    * program : MIDI program number from 1 to 128 (eg. 72) (int)
    */
    ifstream inputFile("input.in", ios::in);

```

```

/// Exit program if can not open file
if(!inputFile)
{
    cerr << "File could not be opened!" << endl;
    exit(1);
}

int divisions, min, beats, beatType, line, tempo, program;
double moment;
char sign, instrument[10];

inputFile >> divisions; // How many divisions per quarter note for a duration
inputFile >> min;        // TH megisth diaresh tou beatType sthn hxografsh - eg. ean beatType = 4
(quarter) tote gia minimum to eighth(8) to min = 2

inputFile >> beats;      // Time signature numerator
inputFile >> beatType; // Time signature denominator

inputFile >> sign;       // Clef sign. Values: G, F, C, percussion, TAB, none
inputFile >> line;       // Staff line numbered from bottom to top for clefs

inputFile >> tempo;      // Tempo in quarter notes per minute (8EWRW!! : in beatType notes
ean byteType diaforetiko tou quarter)

inputFile >> moment;     // Ean to kommati 3ekina me ellipes metro orizoume thn akrivh a3ia autou
inputFile >> instrument; // Full name of instrument
inputFile >> program;    // MIDI program number

inputFile.close(); // Kleisimo arxeiou input.in

/// -----
/// Dhmiourgia arxeiou score.xml (output stream)
ofstream myXmlFile;
myXmlFile.open ("score.xml", ofstream::out);

/// Exit program if can not open file
if(!myXmlFile)
{
    cerr << "File could not be opened!" << endl;
    exit(1);
}

/// -----
/// Egrafh sto arxeio score.xml tw n arxikwn parametrwn
myXmlFile << "<?xml version='1.0' standalone='no'?">"
<< "\n<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 1.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd ">"
<< "\n\n<score-partwise>"
<< "\n  <identification>"
<< "\n      <encoding>"
<< "\n          <software>Score Processing Library</software>"
<< "\n      </encoding>"
<< "\n  </identification>"

```

```

        << "\n <part-list>"
        << "\n <score-part id=\"P1\">"
        << "\n <part-name/>"
        << "\n <score-instrument id=\"P1-I1\">"
        << "\n <instrument-name>" << instrument <<
"/instrument-name>"
        << "\n </score-instrument>"
        << "\n <midi-instrument id=\"P1-I1\">"
        << "\n <midi-channel>1</midi-channel>"
        << "\n <midi-program>" << program << "</midi-
program>"
        << "\n </midi-instrument>"
        << "\n </score-part>"
        << "\n </part-list>"
        << "\n <part id=\"P1\">"
        << "\n <measure number=\"1\">"
        << "\n <attributes>"
        << "\n <divisions>" << divisions << "</divisions>"
        << "\n <key>"
        << "\n <fifths>0</fifths>"
        << "\n </key>"
        << "\n <time>"
        << "\n <beats>" << beats << "</beats>"
        << "\n <beat-type>" << beatType <<
"/beat-type>"
        << "\n </time>"
        << "\n <clef>"
        << "\n <sign>" << sign << "</sign>"
        << "\n <line>" << line << "</line>"
        << "\n </clef>"
        << "\n </attributes>"
        << "\n <sound tempo=\"\" << tempo << "\"/>";

/// -----
/// Anagnwsh dedomenwn apo to arxeio pitch.in (input stream)
/** To arxeio pitch.in periexei thn akolou8eia apo elaxisths diarkeias pitch synodeuomena apo enan
8etiko ari8mo (alfari8mhtikh akolou8ia apo notes)
* O ari8mos autos au3anei se periptwsh idiou pitch to opoio "denetai" me to prohgoymeno tou
* Diaforetika exei thn timh 1
*/
ifstream noteFile("pitch.in", ios::in);

/// Exit program if can not open file
if(!noteFile)
{
    cerr << "File could not be opened!" << endl;
    exit(1);
}

/// Ypologismos plh8ous stoxeiwn sto arxeio pitch.in
int elements = 0; // Mege8os arxeiou pitch.in
char* temp; // Proswrinh apo8hkeush stoxeiwn tou arxeiou
for(;;)
{

```



```

        temp = new char[5];    // e.g. "C#4-1"
        noteFile >> temp;
        if ( noteFile.eof() )
        {
            break;
        }
        elements++;
    }

    noteFile.clear(); // Set error state flags
    noteFile.seekg(0, ios::beg); // Ka8orismos tou get pointer sthn arxh tou arxeiou (Offset 0 from the
beginning of the stream's buffer)

//-----
    /// Kataskeuh pinaka xmlPitchArray kai apo8hkeush tw'n stoixeiw'n tou arxeiou pitch.in
    char** xmlPitchArray;
    xmlPitchArray = new char*[elements];

    int* count;
    count = new int[elements];

    char temp2;    // proswrinh apo8hkeush xarakthra

    for (int i = 0; i < elements; i++)
    {
        xmlPitchArray[i] = new char[3];
        noteFile >> xmlPitchArray[i][0] >> xmlPitchArray[i][1] >> temp2;    // Sto xmlPitchArray[i] exoun
apo8hkeutei oi 2 prwtioi xarakthres tou stoixeiou

        if (temp2 != '-') // Periptwsh allowmenhs notas e.g. stoixeio "C#4-1" opote xmlPitchArray[i] =
"C#4" kai count[i] = 1
        {
            xmlPitchArray[i][2] = temp2;
            noteFile >> temp2 >> count[i];
        }
        else // Periptwsh e.g. stoixeio"C4-1" opote xmlPitchArray[i] = "C4" kai count[i] = 1
        {
            noteFile >> count[i];
        }
    }

    noteFile.close(); // Kleisimo arxeiou note.out

// -----
    /// Egrafh sto score.xml ths melwdias
    int j = 0;    // Metavlth eswterikou vroxou
    char* element;    // Pitch pros sygrish me to epomeno
    int elementCount;    // Diarkeia enos stoixeiou apo thn prwth tou emfanish
    int duration;    // Note duration in divisions
    int measureNumber = 1; // measure-numbering
    double pitchType;    // (double) Pitch type - orisma ths chooseBeatType, Timh me shmeio
anaforas to whole = 1.0
    string type;    // (String) Pitch type - apotelesma ekteleshhs chooseBeatType, Values
= {256th, 128th, 64th, 32nd, 16th, eighth, quarter, half, whole, breve, long}

```

```

string tieprev, tie; // Tied notas se periptwsh allaghs metrou (opou xreiazetai)

/// Vroxos kataskeuhs ths melwdias
for ( int i = 0; i < elements; i = j )
{
    element = xmlPitchArray[i]; // Pitch elaxisths diarkeias gia sygrish me to epomeno
    elementCount = count[i]; // parametros pou sxetizetai me th synolikh diarkeia tou pitch

    /// Arxikopoihsh tw n tie kai tieprev gia endexomenh enwsh ths idias notas se allagh metrou
    tieprev = "null";
    tie = "null";

    for ( j = i + 1; j <= elements; j++)
    {
        if ( ( j < elements) && (strcmp (element,xmlPitchArray[j]) == 0) && (count[j] >
elementCount)) // Au3hsh metrth ean to element den htan to teleutaio pitch kai einai idio me to epomeno tou
meta apo elegxo ths perivallousas (klsh ths join)
        {
            element = xmlPitchArray[j];
            elementCount = count[j];
        }
        else // Diaforetika proxwroume sthn egrafh autou sto score.xml
        {
            /// Ypologismos pitchType
            /** count: Ari8mos apo minimum type pitch (e.g. 4 * eighth = 1 * half)
* min: megisth ypodiaresh toy beatType (e.g. minimum type to eighth <->
beatType=4 (quarter), min=2)
*/
            pitchType = (double) elementCount / min; // e.g. pitchType = 4/2 = 2

            pitchType /= beatType; // Diaresh me beatType: To orisma ths
chooseBeatType orizetai me shmeio anaforas to whole = 1.0 (e.g. 2/4 = 0.5 ---apo chooseType---> half)
            type = chooseBeatType(pitchType); // (String) note type

            pitchType *= beatType; // Epanafora pitchType
            duration = pitchType*divisions; // Note duration "in divisions"

            /// Periptwsh 1: H nota exei diarkeia pou den 3eperna to metro

            if ((moment + pitchType) <= beats )
            {

                moment += pitchType; // Xronikh stigmh apo thn arxh tou metrou
meta thn egrafh ths notas

                /// Egrafh ths notas sto arxeio score.xml me klsh ths storeNote
myXmlFile.close(); // Kleisimo arxeiou prin thn klsh ths
storeNote h opoia pragmatopoiiei egrafh sto idio arxeio
                storeNote (element, duration, type, tieprev, tie);
myXmlFile.open ("score.xml", ofstream::app); // Anoigma tou
arxeiou score.xml gia pros8hkh stoixeiwn (append)

                /// Allagh metrou sto score.xml
                if ((beats - moment < 1e-014) && (j != elements)) // Sthn periptwsh

```

```

ypodiareshs se triixa (e.g.min=6) eisagetai ena sfalma sthn timh moment ths ta3ews tou 1e-015
    {
        myXmlFile    << "\n          </measure>"
                    << "\n          <measure
number=\"\" << ++measureNumber << "\">";

        moment = 0.0; // Neo metro
    }
}

/// Periptwsh 2: H diarkeia ths notas 3eperna to metro sto opoio vriskomaste

else
{
    double temp = beats - moment; // Diarkeia ths notas pou xwra sto
arxiko metro
    double temp2 = pitchType - temp; // Ypoloipo ths diarkeia ths
notas pou 8a topo8eth8ei ston neo metro

    /// Sthn periptwsh ypodiareshs diarkeias se triixa (e.g.min=6) eisagetai
ena sfalma sthn timh moment, kai ara kai temp, temp2. Auto lynei to provlhma gia triixa elaxisths diarkeias
    if ((temp > beatType/min) && (temp < 2*beatType/min))
    {
        temp = beatType/min;
        temp2 = pitchType - temp;
    }

    tie = "start"; // Arxh enwshs

    /// To temp ginetai iso h mikrotero tou 0 afou h nota exei katagrafei
    while (temp > 0)
    {
        temp /= beatType; // Diaresh me beatType: To orisma
ths chooseBeatType orizetai me shmeio anaforas to whole = 1.0
        type = chooseBeatType(temp); // (String) Note type
        temp *= beatType; // Epanafora temp
        duration = temp*divisions; // Note duration "in
divisions"

        moment += temp; // Xronikh stigmh apo thn arxh tou
metrou meta thn egraph ths notas

        /// Egraph ths notas sto arxeio score.xml me klsh ths
storeNote
        myXmlFile.close(); // Kleisimo arxeiou prin thn klsh ths
storeNote h opoia pragmatopoiei egraph sto idio arxeio
        storeNote (element, duration, type, tieprev, tie);
        myXmlFile.open ("score.xml", ofstream::app); // Anoigma
tou arxeiou score.xml gia pros8hkh stoixeiwn (append)

        /// Allagh metrou sto arxeio score.xml
        if (beats - moment < 1e-014) // Sthn periptwsh
ypodiareshs se triixa (e.g.min=6) eisagetai ena sfalma sthn timh moment ths ta3ews tou 1e-015
        {

```

```

myXmlFile    << "\n          </measure>"
              << "\n          <measure
number=\"\" << ++measureNumber << "\">";

              moment=0;    // Neo metro
          }

          /// Diarkeia tou ypoloipou ths notas akoma megalyterh tou
metrou

          if (temp2 > beats)
          {
              /// Apo8hkeush ths notas me diarkeia ish me ena
metro (beats)

              double metro = (double) beats / beatType; // Diaresh
me beatType: To orisma ths chooseBeatType orizetai me shmeio anaforas to whole = 1.0
              type = chooseBeatType(metro); // (String) Note type

              duration = beats*divisions;           // Note
duration "in divisions"

              /// Egrafh ths notas sto arxeio score.xml me klhsh ths
storeNote

              myXmlFile.close();    // Kleisimo arxeiou prin thn
klhsh ths storeNote h opoia pragmatopoeiei egrafh sto idio arxeio
              tieprev = "stop"; // Kleisimo prohgoumenhs enwshs
              tie = "start";      // Arxh epomenhs
              storeNote (element, duration, type, tieprev, tie);
              tie = "null";       // 8a exei oloklhrw8ei (timh
arxikopoihshs)

              tieprev = "stop"; // Kleisimo
              myXmlFile.open ("score.xml", ofstream::app);    //
Anoigma tou arxeiou score.xml gia pros8khk stoixeiwn (append)

              /// Allagh metrou sto arxeio score.xml
myXmlFile    << "\n          </measure>"
              << "\n          <measure
number=\"\" << ++measureNumber << "\">";

              moment = 0;    // Neo metro

          }
          else if (temp2 > 0)
          {
              temp2 /= beatType;    // Diaresh me beatType: To
orisma ths chooseBeatType orizetai me shmeio anaforas to whole = 1.0
              type = chooseBeatType(temp2); // (String) Note type
              temp2 *= beatType;    // Epanafora temp2
              duration = temp2*divisions;           // Note
duration "in divisions"

              /// Egrafh ths notas sto arxeio score.xml me klhsh ths
storeNote

              myXmlFile.close();    // Kleisimo arxeiou prin thn
klhsh ths storeNote h opoia pragmatopoeiei egrafh sto idio arxeio

```

```

    tieprev = "stop"; // Telos prohgomoumenhs
    tie = "null";      // Timh opws arxikopoihsh
    storeNote (element, duration, type, tieprev, tie); //

Enallagh tie <-> tieprev gia olokrhrwsh
    myXmlFile.open ("score.xml", ofstream::app); //
Anoigma tou arxeiou score.xml gia pros8hkh stoixeiwn (append)

    moment = temp2; // To metro meta thn apo8hkeush
ths temp2

    /// Allagh metrou sto score.xml
    if ((beats - moment < 1e-014) && (j != elements)) //
Sthn periptwsh ypodiaireshs se triixa (e.g.min=6) eisagetai ena sfalma sthn timh moment ths ta3ews tou 1e-015
    {
        myXmlFile    << "\n
        << "\n      <measure
    number=\"\" << ++measureNumber << "\">";

        moment = 0.0; // Neo metro
    }

    break; // Epomenh nota
}

/// Periptwsh sthn opoia h nota apaitei ki allo/alla metro/a -
xwrismos omoia me prin
    temp = temp2 - beats;

    /// Ean h temp exei diarkeia megalyterh tou metrou
    if (temp > beats)
    {
        temp2 = temp - beats;
        temp = beats;
        tie = "start";
        tieprev = "stop";
    }
    /// Ean h temp exei diarkeia mikroterh tou metrou menei mono
h egraphh auths
    else
    {
        temp2 = 0;
        tie = "null";
        tieprev = "stop";
    }

} // end of while gia periptwsh notas me diarkeia megalyterh tou
metrou

    tie = "null";      // Timh opws arxikopoihsh
    tieprev = "null"; // Timh opws arxikopoihsh

} // end of else if egraphhs notas diarkeias megalyterhs tou metrou
break; // Epomenh nota

```

```

        } // end of else egrafhs mias notas
    } // end of eswteriko for loop
} // oles oi notes exoun apo8hkeutei sto arxeio score.xml

/// -----
/// Olokhrwsh tou arxeiou score.xml
myXmlFile << "\n                <barline location=\"right\">"
                << "\n                                <bar-style>light-heavy</bar-style>"
                << "\n                                </barline>"
                << "\n                </measure>"
                << "\n </part>"
                << "\n</score-partwise>";

myXmlFile.close(); // Kleisimo arxeiou score.xml
delete xmlPitchArray; // Free memory pointed to by XmlPitchArray
}

// *****
/// Synarthsh epistrofhs (string) type of note
/** Dexetai ws eisodo enan double pou antistoixei sth diarkeia ths notas (eg. whole = 1/1, quarter = 1/4)
 * Elegxetai kai h periptwsh parestigmenhs notas (one dot only) kai trihxou (eighth,quarter).
 */
string chooseBeatType (double x)
{
    if (x == 1.0/256)
        return("256th");
    else if (x == 1.0/128 || x == 1.0/128 + 1.0/256)
        return("128th");
    else if (x == 1.0/64 || x == 1.0/64 + 1.0/128)
        return("64th");
    else if (x == 1.0/32 || x == 1.0/32 + 1.0/64)
        return("32nd");
    else if (x == 1.0/16 || x == 1.0/24 || x == 1.0/16 + 1.0/32)
        return("16th");
    else if (x == 1.0/8 || x == 1.0/12 || x == 1.0/8 + 1.0/16)
        return("eighth");
    else if (x == 1.0/4 || x == 1.0/4 + 1.0/8)
        return("quarter");
    else if (x == 1.0/2 || x == 1.0/2 + 1.0/4)
        return("half");
    else if (x == 1.0/1 || x == 1.0/1 + 1.0/2)
        return("whole");
    else if (x == 2.0/1 || x == 3.0/1)
        return("breve");
    else
        return("long");
} // end of chooseBeatType

// *****
/// Synarthsh egrafhs sto arxeio score.xml tou pitch element ws nota me symvolo type kai diarkeia duration
void storeNote (char* element, int duration, string type, string tieprev, string tie)

```

```

{
    ofstream myXmlFile; // Egrafh sto arxeio score.xml (output stream)
    myXmlFile.open ("score.xml", ofstream::app); // Anoigma tou arxeiou score.xml gia pros8hkh
    stoixeiwn (append)

    myXmlFile << "\n" << " <note>";

    /// Ean to stoixeio element exei thn timh "NO" h "Bad" tote pros8etoume thn atistoixhs diarkeias paush
    if ( element[0] == 'N' )
    {
        myXmlFile << "\n" << " <rest/>";
    }
    else // Egrafh tou pitch
    {
        /// Egrafh prwtou xarakthra -> step (eg. 'A' ean "A#4" h "A4")
        myXmlFile << "\n" << " <pitch>"
        << "\n" << " <step>" << element[0] <<
"</step>";

        /// Egrafh tou teleutaiou xarakthra -> octave ('4') kai ths # mesw tou alter = +1 (eg. "A#4")
        if ( element[1] == '#' )
        {
            myXmlFile << "\n" << " <alter>+1</alter>"
            << "\n" << " <octave>" <<
element[2] << "</octave>"
            << "\n" << " </pitch>";
        }
        /// Egrafh tou teleutaiou xarakthra -> octave (eg. '4' ean "A4")
        else
        {
            myXmlFile << "\n" << " <octave>" << element[1] <<
"</octave>"
            << "\n" << " </pitch>";
        }
    } // end of egrafh pitch

    /// Egrafh diarkeias ths notas/paushs
    myXmlFile << "\n" << " <duration>" << duration << "</duration>";

    /// Egrafh typou ths notas/paushs

    /// Pi8anh anagh enwshs ths notas se allagh metrou (ean den einai paush!)
    if (((tie != "null") || (tieprev != "null")) && !(strcmp(element,"NO") == 0) || (strcmp(element,"Bad") == 0))
    {
        /// Enwsh me prohgomenh kai epomenh
        if ((tieprev != "null") && (tie != "null"))
        {
            myXmlFile << "\n" << " <tie type=\"\" << tieprev << "\"/>"
            << "\n" << " <tie type=\"\" << tie << "\"/>"
            << "\n" << " <type>" << type << "</type>"
            << "\n" << " <notations>"
            << "\n" << " <tied type=\"\" <<
tieprev << "\"/>"

```

```

        << "\/>"
        << "\n
        <tie type=\"\" << tie

    }

    // Enwsh me epomenh( tie == "start") h prohgoumenh( tieprev == "stop")
    else
    {
        if (tieprev != "null")
            tie = tieprev;

        myXmlFile << "\n
        << "\n
        << "\n
        << "\n
        <tie type=\"\" << tie << "\/>"
        <type>" << type << "</type>"
        <notations>"
        <tie type=\"\" << tie

    << "\/>"
        << "\n
        </notations>";

    }

}

else
    myXmlFile<< "\n
        <type>" << type << "</type>";

    myXmlFile << "\n
        </note>";

    myXmlFile.close(); // Kleisimo arxeiou score.xml
} // end of storeNote

// *****
// Synarthsh elegxou mh shmantikhs ptwshs ths perivalousas tou shmatos
bool join (double* testArray, int x, int y)
{
    double tempMin = 100; // Proswrinh apo8hkeush elaxisths timhs
    double tempMax = 0; // Proswrinh apo8hkeush megisths timhs

    // Den elegxw tis akraies times ka8ws den periexoun shmantikh plhroforia
    int first = x + (y - x) / 4;
    int last = y - (y - x) / 4;

    int mini = first; // Arxikopoihsh index elaxisths timhs

    for (int i = first; i <= last; i++)
    {
        // Euresh elaxisths timhs
        if (testArray[i] < tempMin)
        {
            tempMin = testArray[i];
        }

        // Euresh megisths timhs
        if (testArray[i] > tempMax)
        {
            tempMax = testArray[i];
        }
    }
}

```



```
    }  
}  
  
/// Syn8hkh apofashs  
/** Ean (tempMin < 50) prokeitai gia paush opote join = true  
 *      8ewrw shmantikh ptwsh > 6 opote join = false  
 */  
if ((tempMin > 50) && ((tempMax - tempMin) > 6))  
    return false;  
else  
    return true;  
}
```

\*Δέχεται ως εισόδους το αρχείο με την διορθωμένη αλφαριθμητική ακολουθία (pitch.in) καθώς και το αρχείο input.in με τις βασικές παραμέτρους της ηχογράφησης για τη δημιουργία του αρχείου score της αυτόματης καταγραφής (B2).

# 6

## Βιβλιογραφία

- [1] **Vemus** : <http://www.vemus.org/>
- [2] **Praat** : <http://www.fon.hum.uva.nl/praat/>
- [3] **MusicXml** : <http://www.recordare.com/xml.html>
- [Abd02] © S.A. Abdallah 2002  
*Towards Music Perception by Redundancy Reduction and Unsupervised Learning in Probabilistic Models*  
PhD thesis, Department of Electronic Engineering, King's College London
- [AP04] © S.A. Abdallah and M.D. Plumbley 2004  
*Polyphonic transcription by non-negative sparse coding of power spectra*  
International Conference on Music Information Retrieval, pp. 318-325, Barcelona, Spain
- [BE01] © A.L. Berenzweig and D.P.W. Ellis 2001  
*Locating singing voice segments within music signals*  
IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 119-122, New Paltz, USA

- [Bilm93] © J.A. Bilmes 1993  
***Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm***  
Master's thesis, Massachusetts Institute of Technology
- [Byrd84] © Donald Alvin Byrd 1984  
***Music Notation by Computer***  
available @ <http://www.informatics.indiana.edu/donbyrd/>
- [Byrd94] © Donald Alvin Byrd 1994  
***Music Notation Software and Intelligence***  
Computer Music Journal, 18:1, pp.17-20, MIT
- [Cas98] © M.A. Casey 1998  
***Auditory Group Theory with Applications to Statistical Basis Methods for Structured Audio***  
PhD thesis, Massachusetts Institute of Technology
- [Chafe85] © C. Chafe, J. Kashima, B. Mont-Reynaud, and J. Smith 1985  
***Techniques for note identification in polyphonic music***  
International Computer Music Conference pp. 399-405, Vancouver, Canada
- [CK03] © A.T. Cemgil and B. Kappen 2003  
***Monte Carlo methods for tempo tracking and rhythm quantization***  
Journal of Artificial Intelligence Research, 18:45-81
- [CIPitch] <http://www.phys.unsw.edu.au/music/clarinet/E3.html>
- [CW00] © M.A. Casey and A. Westner 2000  
***Separation of mixed audio sources by independent subspace analysis***  
International Computer Music Conference, Berlin, Germany
- [DG02] © M. Davy and S. Godsill 2002  
***Bayesian harmonic models for musical signal analysis***  
Seventh Valencia International meeting Bayesian statistics 7, Tenerife, Spain
- [DH] © Richard O Duda, Peter E. Hart, David G. Stork  
***Pattern Classification (2<sup>nd</sup> edition)***

- [Dix01] © Simon Dixon 2001  
**Automatic Extraction of Tempo and Beat from Expressive Performances**  
Austrian Research Institute for Artificial Intelligence, Wien, Austria
- [DT06H] © Άγγελος Γκιόκας, Διπλωματική εργασία, 2006  
**Αυτόματη Αναγνώριση Μουσικών Οργάνων**  
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής
- [DT07S] © Βαγια Ουρανία-Ειρήνη, Διπλωματική εργασία, 2007  
**Εφαρμογές της Μουσικής Τεχνολογίας στην Μουσική Εκπαίδευση**  
Τομέας Σημάτων, Ελέγχου & Ρομποτικής
- [DT08I] © Κουλουμπή Ηλέκτρα, Διπλωματική εργασία 2008  
**Εγκεφαλική Λειτουργία και Μουσική: Καταγραφή και Επεξεργασία Δεδομένων Ηλεκτροεγκεφαλογραφίας σε Αρμονικά και Δυσαρμονικά Ακουστικά Ερεθίσματα**  
Τομέας Συστημάτων Μετάδοσης Πληροφορίας & Τεχνολογίας Υλικών
- [DT08L] © Θεόδωρος Καρβούνης, Διπλωματική εργασία, 2008  
**ΜΟΥΣΙΚΟΣΥΝΘΕΤΗΣ σύστημα ημιαυτόματης σύνθεσης Βυζαντινής μουσικής**  
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών
- [EB04] © J. Eggink and G.J. Brown 2004  
**Instrument recognition in accompanied sonatas and concertos**  
IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 217-220, Montreal, Canada
- [Fal03] © D. FitzGerald, R. Lawlor, and E. Coyle 2003  
**Drum transcription in the presence of pitched instruments using prior subspace analysis**  
Irish Signals & Systems Conference 2003, Limerick, Ireland
- [FCL03] © D. FitzGerald, E. Coyle, and B. Lawlor 2003  
**Prior subspace analysis for drum transcription**  
Audio Engineering Society 114th Convention, Amsterdam, Netherlands, March
- [Fitz04] © D. FitzGerald 2004  
**Automatic Drum Transcription and Source Separation**  
PhD thesis, Dublin Institute of Technology

- [GB99] © D. Godsmark and G.J. Brown 1999  
*A blackboard architecture for computational auditory scene analysis*  
Speech Communication, 27(3):351-366  
© Mastaka Goto, Keiji Hirata 2004
- [GH04] *Recent studies on music information processing*  
Acoust. Sci. & Tech. 25, 6 (2004)  
available @ <http://staff.aist.go.jp/m.goto/PAPER/JAS200411goto/index.html>
- [GM94] © Goto and Y. Muraoka 1994  
*A sound source separation system for percussion instruments*  
Transactions of the Institute of Electronics, Information and Communication Engineers D-II, J77-D-II(5):901-911 (in Japanese)
- [GM94b] © Goto and Y. Muraoka 1994  
*A beat tracking system for acoustic signals of music.*  
ACM International Conference on Multimedia, pp. 365-372, San Fransisco, California
- [GM97] © M. Goto and Y. Muraoka 1997  
*Real-time rhythm tracking for drumless audio signals — chord change detection for musical decisions —*  
Working Notes of the IJCAI-97 Workshop on Computational Auditory Scene Analysis, pp. 135–144
- [Good01] © Michael Good 2001  
*MusicXML: An Internet-Friendly Format for Sheet Music*  
available @ <http://www.recordare.com/good/xml2001.html>
- [Good01s] © Good Michael 2001  
*MusicXML for Notation and Analysis, The Virtual Score: Representation, Retrieval, Restoration*  
ed. Walter B. Hewlett and Eleanor Selfridge-Field (Cambridge, MA: MIT Press)
- [Goto01] © M. Goto 2001  
*A predominant-F estimation method for real-world musical audio signals: Map estimation for incorporating prior knowledge about F0s and tone models*  
Proc. Workshop on Consistent and Reliable Acoustic Cues for Sound Analysis, Aalborg, Denmark

- [GR04] © O. Gillet and G. Richard 2004  
*Automatic transcription of drum loops*  
IEEE International Conference on Acoustics, Speech, and Signal Processing,  
Montreal, Canada
- [Hart96] © W.M. Hartmann 1996  
*Pitch, periodicity, and auditory organization*  
Journal of the Acoustical Society of America, 100(6):3491-3502
- [HM04] © S.W. Hainsworth and M.D. Macleod 2004  
*Particle filtering applied to musical tempo tracking*  
Journal of Applied Signal Processing, 15:2385-2395
- [HPD03] © P. Herrera, G. Peeters, and S. Dubnov 2003  
*Automatic classification of musical instrument sounds*  
Journal of New Music Research, 32:3-21
- [Kal05] © F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and  
P. Cano 2005  
*An experimental comparison of audio tempo induction algorithms.*  
IEEE Transactions on Speech and Audio Processing
- [Kal06] © A. Klapuri, A Eronen, and J. Astola 2006  
*Analysis of the meter of acoustic musical signals*  
IEEE Transactions on Speech and Audio Processing, 14(1)
- [Kal95] © K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka 1995  
*Organisation of hierarchical perceptual sounds: Music scene analysis with  
autonomous processing modules and a quantitative information integration  
mechanism*  
International Joint Conference on Artificial Intelligence, pp. 158-164, Montreal,  
Quebec
- [KD06] © A. Klapuri, M. Davy 2006  
*Signal Processing Methods for Music Transcription*  
Springer, New York  
available @ <http://www.cs.tut.fi/~klap/amt.html> and  
<http://www.springerlink.com/home/main.mpx>

- [KEA06] © A. Klapuri, A Eronen, and J. Astola 2006  
*Analysis of the meter of acoustic musical signals*  
IEEE Transactions on Speech and Audio Processing, 14(1)
- [Klap04] © Anssi P. Klapuri 2004  
*Automatic music transcription as we know it today*  
Journal of New Music Research, Vol. 33, No. 3, pp. 269–282  
available @ <http://www.cs.tut.fi/~klap/iiro>
- [Klap04T] © Anssi P. Klapuri 2004  
*Signal Processing Methods for the Automatic Transcription of Music*  
Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Auditorium S1, at Tampere University of Technology, Tampere  
available @ <http://www.cs.tut.fi/~klap/iiro>
- [Klap05] © A.P. Klapuri 2005  
*A perceptually motivated multiple-FO estimation method for polyphonic music signals*  
IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA
- [KM99] © K. Kashino and H. Murase 1999  
*A sound source identification system for ensemble music based on template adaptation and music stream extraction*  
SpeechCommunication, 27:337-349
- [KNS04] © H. Kameoka, T. Nishimoto, and S. Sagayama 2004  
*Separation of harmonic structures based on tied Gaussian mixture model and information criterion for concurrent sounds*  
In IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada
- [KT00] © T. Tolonen and M. Karjalainen 2000  
*A computationally efficient multipitch analysis model*  
IEEE Transactions on Speech and Audio Processing, 8(6):708-716
- [KT93] © K. Kashino and H. Tanaka 1993  
*A sound source separation system with the ability of automatic tone modeling*  
International Computer Music Conference, pp. 248-255, Tokyo, Japan

- [KV09] © Anssi Klapuri, Tuomas Virtanen 2009  
*Handbook of Signal Processing in Acoustics, part IV, chapter 1: Automatic Music Transcription*  
Springer  
available @ <http://www.springerlink.com/home/main.mpx>
- [Lu06] © David Lu 2006  
*Automatic Music Transcription Using Genetic Algorithms and Electronic Synthesis*
- [Lep99] © P. Lepain 1999  
*Polyphonic pitch extraction from musical signals*  
Journal of New Music Research, 28(4):296-309
- [LevDist] **Levenshtein Distance**  
[http://en.wikipedia.org/wiki/Levenshtein\\_distance](http://en.wikipedia.org/wiki/Levenshtein_distance)
- [Mah90] © R.C. Maher 1990  
*Evaluation of a method for separating digitized duet signals*  
Journal of the Audio Engineering Society, 38(12):956-979
- [Maher89] © R.C. Maher 1989  
*An Approach for the Separation of Voices in Composite Music Signals*  
PhD thesis, Univ. of Illinois, Urbana
- [Mar04] © Marolt 2004  
*A connectionist approach to transcription of polyphonic piano music*  
IEEE Transactions on Multimedia, 6(3):439-449
- [Mar96] © K.D. Martin 1996  
*Automatic transcription of simple polyphonic music: Robust front end processing.*  
Technical Report 399, MIT Media Laboratory Perceptual Computing Section
- [Mel91] © D.K. Mellinger 1991  
*Event Formation and Separation of Musical Sound.* PhD thesis  
Stanford University, Stanford, USA



- [Moor75] © J. A. Moorer 1975  
***On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer***  
PhD thesis, Department of Music, Stanford University, Distributed as Department of Music report No. STAN-M-3
- [Moor77] © J. A. Moorer 1977  
***On the transcription of musical sound by computer***  
Computer Music Journal
- [Piszc86] © M. Piszczalski 1986  
***A Computational Model of Music Transcription***  
PhD thesis, Univ. of Michigan, Ann Arbor
- [PK03] © J.K. Paulus and A.P. Klapuri 2003  
***Conventional and periodic N-grams in the transcription of drum sequences***  
IEEE International Conference on Multimedia and Expo, Volume 2, pp. 737-740, Baltimore, Maryland, USA
- [PM] © John G. Proakis, Dimitris G. Manolakis  
***Digital Signal Processing (3<sup>rd</sup> edition)***  
Prentice Hall
- [Praat03] © Paul Boersma 1993  
***ACCURATE SHORT-TERM ANALYSIS OF THE FUNDAMENTAL FREQUENCY AND THE HARMONICS-TO-NOISE RATIO OF A SAMPLED SOUND***  
Available @ <http://music.cs.northwestern.edu/courses/eecs352/papers/pitch%20tracking%20-%20boersma.pdf>
- [PraatM] ***PRAAT TUTORIAL FOR MUSICOLOGISTS***  
available @ [http://www.musicology.nl/wm/research/praat\\_musicologists.htm](http://www.musicology.nl/wm/research/praat_musicologists.htm)
- [PV05] © J. Paulus and T. Virtanen 2005  
***Drum transcription with non-negative spectrogram factorisation***  
European Signal Processing Conference, Antalya, Turkey
- [RK05] © Ryynanen and A. Klapuri 2005  
***Polyphonic music transcription using note event modeling***  
IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA

- [RS] © L. R. Rabiner, R.W. Schafer  
***Digital Processing of Speech Signal***  
Prentice Hall
- [SB03] © P. Smaragdis and J.C. Brown 2003  
***Non-negative matrix factorization for polyphonic music transcription***  
IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA
- [Sch85] © W.A. Schloss 1985  
***On the Automatic Transcription of Percussive Music – from Acoustic Signal to High-Level Analysis***  
PhD thesis, Center for Computer Research in Music and Acoustics, Stanford University, Stanford, California, USA
- [Sche98] E.D. Scheirer 1998  
***Tempo and beat analysis of acoustical musical signals***  
Journal of the Acoustical Society of America, 103(1):588-601
- [ScriptP] © Paul Boersma  
***Script for listing time|--F0|--intensity***  
available @ [http://www.fon.hum.uva.nl/praat/manual/Script\\_for\\_listing\\_time\\_--F0\\_--intensity.html](http://www.fon.hum.uva.nl/praat/manual/Script_for_listing_time_--F0_--intensity.html)
- [SendP] © Paul Boersma  
Sendpraat: sending messages to a Praat shell program  
available @ <http://www.fon.hum.uva.nl/praat/sendpraat.html>
- [Sma01] © P. Smaragdis 2001  
***Redundancy Reduction for Computational Audition, a Unifying Approach***  
PhD thesis, Massachusetts Institute of Technology
- [Smith] © Steven W. Smith  
***The Scientist and Engineer's Guide to Digital Signal Processing***  
Ph.D
- [SSW99] © A. Sterian, M.H. Simoni, and G.H. Wakefield 1999  
***Model-based musical transcription***  
In International Computer Music Conference, Beijing, China

- [Tangui93] © A.S. Tanguiane 1993  
*Artificial Perception and Music Recognition*  
Springer, Berlin Heidelberg  
available @ <http://www.springerlink.com/home/main.mpx>
- [VR04] © E. Vincent and X. Rodet  
*Instrument identification in solo and ensemble music using independent subspace analysis*  
International Conference on Music Information Retrieval, Barcelona, Spain
- [Webs07] © Peter R. Webster 2007  
*Computer-based Technology and Music Teaching and Learning: 2000–2005*  
International Handbook of Research in Arts Education  
Springer  
available @ <http://www.springerlink.com/home/main.mpx>
- [Yal04] © K. Yoshii, M. Goto, and H.G. Okuno 2004  
*Drum sound identification for polyphonic music using template adaptation and matching methods*  
ISC A Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, Jeju, Korea
- [Zal02] © A. Zils, F. Pachet, O. Delerue, and F. Gouyon 2002  
*Automatic extraction of drum tracks from polyphonic music signals*  
International Conference on Web Delivering of Music, Darmstadt, Germany