Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπυξη μεθόδου εντοπισμού κρουστικών συμβάντων σε ροές ήχου και υλοποίηση σε περιβάλλον Android
Ονοματεπώνυμο Φοιτητή	Ζαβιτσάνος Φώτιος-Δημήτριος
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	ΜΠΠΛ/ 09034
Επιβλέπων	Πικράκης Άγγελος, Λέκτορας

Ημερομηνία Παράδοσης Μάρτιος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Άγγελος Πικράκης Λέκτορας	Χάρης Κωνσταντόπουλος Επίκουρος Καθηγητής	Γεώργιος Τσιχριντζής Καθηγητής

Περίληψη
Η παρούσα διπλωματική εργασία πραγματεύεται τη διαδικασία κατάτμησης περιοχών κρουστικών ήχων σε ροές ήχου, την ομαδοποίηση των περιοχών αυτών βάσει των χαρακτηριστικών τους, και, τέλος την αναπαραγωγή τους με τη χρήση συνθετικών ήχων – έτοιμων ηχητικών δειγμάτων.

Για την παρουσίαση της υπό μελέτη μεθοδολογίας παρατίθεται εφαρμογή επίδειξης σε πλατφόρμα Android η οποία αναπτύχθηκε με τη χρήση του Android SDK.

Abstract
The current work is concerned with the tasks of detecting impulsive sounds in user-driven audio streams, clustering the detected events based on appropriately selected features and reproducing the events in synthetic audio streams using predefined audio samples. For demonstration purposes, an Android application has been implemented with the Android SDK.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Πίνακας Περιεχομένων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

1. Εισαγωγή

Η σύγχρονη ψηφιακή τεχνολογία έχει πραγματοποιήσει βήματα μεγίστης σημασίας στον τομέα της Ψηφιακής Επεξεργασίας Σήματος (Digital Signal Processing). Λογικό επακόλουθο της προαναφερθείσας προόδου είναι η ταχεία εκμετάλλευσή της και στον τομέα της Μουσικής Πληροφορικής.

Τόσο σε επίπεδο επεξεργασίας του μουσικού σήματος (ένα επίπεδο το οποίο είναι ευρέως γνωστό στο ευρύ κοινό με την ορολογία effects), όσο και σε επίπεδο ανάλυσης του ηχητικού σήματος και εξαγωγής των χαρακτηριστικών αυτού, η ψηφιακή τεχνολογία έχει συνεισφέρει τα μέγιστα. Σε φαινομενικά "απλά" effects που τη σημερινή εποχή είναι διαθέσιμα ακόμα και σε ένα φορητό MP3 player, υπάρχει αυτή η πλάνη, ακριβώς επειδή η ψηφιακή τεχνολογία τα κατέστησε προσιτά και οικονομικά, ενώ στο όχι και πολύ μακρινό παρελθόν ήταν αρκετά δύσκολα -αν όχι αδύνατον- να πραγματοποιηθούν.

Εκτός, όμως, από τα effects, σημαντική πρόοδος έχει επιτευχθεί και στον τομέα της εξαγωγής ορισμένων χαρακτηριστικών ενός ηχητικού αποσπάσματος[1]. Για του λόγου το αληθές, αρκεί ο αναγνώστης να αναλογιστεί το πως φτάσαμε από τους συντονιστές του Helmholtz (resonators)[2] στα ταχύτατα φασματογραφήματα που χρησιμοποιούνται για οπτικοποίηση του ήχου στις συσκευές αναπαραγωγής των αυτοκινήτων.

Από τη στιγμή που υπάρχει η δυνατότητα γρήγορης εξαγωγής των χαρακτηριστικών ενός ηχητικού αποσπάσματος, το επόμενο -και ίσως πιο συναρπαστικό- βήμα είναι η αξιοποίηση τεχνικών από άλλους τομείς της Πληροφορικής, όπως τον τομέα της Αναγνώρισης Προτύπων (Pattern Recognition) για την εξαγωγή κάποιου είδους συμπερασμάτων σχετικά με τον υπό επεξεργασία ήχο. Μια τέτοιου είδους προσέγγιση θα έχει την ευκαιρία να διαπιστώσει ο αναγνώστης στις επόμενες σελίδες του κειμένου.

1.1 Στόχος Εργασίας

Στόχος της παρούσας εργασίας είναι η αξιοποίηση ορισμένων τεχνικών επεξεργασίας ηχητικού σήματος, εξαγωγής χαρακτηριστικών αλλά και της τεχνικής της Ομαδοποίησης (Clustering), ούτως ώστε να επιτευχθεί κατάτμηση των κρουστικών ήχων ενός ηχητικού σήματος και να προσεγγιστεί ο ρυθμός μέσω της ομαδοποίησης.

Απώτερος στόχος αποτελεί η δυνατότητα ανασύνθεσης του αρχικού προσλαμβανόμενου ρυθμού με τη χρήση προ ηχογραφημένων δειγμάτων κρουστών ήχων. Στο σημείο αυτό θα ήταν χρήσιμο να τονιστεί ότι σκοπός της παρούσας εργασίας δεν είναι να προτείνει κάποια καινοτόμα τεχνική στα επιμέρους λογικά βήματα της υλοποίησης, όπως αυτά προκύπτουν από ένα πρωταρχικό σκαρίφημα, όπως φαίνεται και στην Εικόνα 1, αλλά να συνδυαστούν οι βασικές τεχνικές στα προαναφερθέντα βήματα, με οικονομικούς -από άποψη επεξεργαστικού φόρτου- αλγόριθμους.

Ο σχεδιασμός που πραγματοποιήθηκε και περιγράφεται στο Κεφάλαιο 2 στοχεύει σε υλοποιήσεις για προσωπικές, "έξυπνες" συσκευές, όπως κινητά τηλέφωνα (smartphones) και άλλες μικροσυσκευές. Για το σκοπό αυτό, σε κάθε διακριτό βήμα του σχεδιασμού γίνεται προσπάθεια υιοθέτησης τεχνικών που έχουν ήδη προταθεί σε σχετική βιβλιογραφία.

[1] Εννοείται, εδώ, ηχητικό απόσπασμα σε ψηφιακή μορφή. Για λόγους οικονομίας χώρου, στο παρόν κείμενο, θεωρείται από τον γράφων ότι ο αναγνώστης διαθέτει ένα ικανό επίπεδο εξοικείωσης με την έννοια του ψηφιακού σήματος, συνεπώς οι έννοιες της δειγματοληψίας (sampling) και της κβάντοποίησης (quantization), που χαρακτηρίζουν ένα ψηφιακό σήμα, δεν θα αναλυθούν.

[2] Συντονιστές ή Συνηχητές Helmholtz: Ένα από τα πρώτα μέσα μετασχηματισμού Fourier. Πλέον, οι ηλεκτρονικές συσκευές χρησιμοποιούν αλγόριθμους FFT (Fast Fourier Transform), που είναι υλοποιήσεις προσέγγισης του μαθηματικού τύπου του Διακριτού Μετασχηματισμού Fourier (Discrete Fourier Transform -DFT)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Εικόνα 1: Βασικό διάγραμμα ροής

1.2 Παρόμοια Προβλήματα – Σχετική Έρευνα

Η παρούσα εργασία έχει εμπνευσθεί από το διεθνή διαγωνισμό Query By Tapping της MIREX. Ο σχετικός σύνδεσμος στο διαδίκτυο βρίσκεται στη διεύθυνση http://www.music-ir.org/mirex/wiki/Query_by_Tapping. Ο διαγωνισμός αυτός εστιάζει στον εντοπισμό των σημείων onset ενός ηχητικού δείγματος, συνεπώς αποτελεί παρόμοιο πρόβλημα με το πρώτο μέρος της παρούσας εργασίας, ήτοι τον εντοπισμό – κατάτμηση των κρουστικών τμημάτων.

Επιπροσθέτως, όπως θα δειχθεί στην Ενότητα 2.1, ορισμένες από τις πηγές του διαγωνισμού Query by Tapping χρησιμοποιήθηκαν προκειμένου να αξιολογηθεί η αποδοτικότητα του σχεδιασμένου αλγόριθμου κατάτμησης.

1.3 Οργάνωση Εργασίας

Το παρόν τεκμήριο είναι οργανωμένο σε κεφάλαια, όπου το καθένα αντιπροσωπεύει τις διακριτές φάσεις σχεδιασμού και υλοποίησης της τελικής εφαρμογής.

Πιο συγκεκριμένα, στο Κεφάλαιο 2, γίνεται περιγραφή του σχεδιασμού του αλγόριθμου, καθώς και μια σύντομη παράθεση των αποτελεσμάτων από τη διαδικασία της προτυποποίησης με τη χρήση του προγραμματιστικού εργαλείου GNU Octave. Ιδιαίτερη έμφαση δίνεται στην αξιολόγηση της διαδικασίας κατάτμησης – εντοπισμού των κατάλληλων τμημάτων του ηχητικού σήματος (βλ. Ενότητα 2.1). Επίσης, στην Ενότητα 2.2 γίνεται παρουσίαση του αλγόριθμου εξαγωγής των απαραίτητων, για τη διαδικασία της ομαδοποίησης, χαρακτηριστικών του ήχου, ενώ στην Ενότητα 2.3 περιγράφεται ο αλγόριθμος της ομαδοποίησης.

Καθώς στην παρούσα εργασία δεν αποδίδεται ιδιαίτερο βάρος στην τελική σύνθεση – παραγωγή του ήχου, δεν κρίθηκε σκόπιμο από τον γράφων να συμπεριληφθεί ειδική ενότητα για την φάση της σύνθεσης.

Στο Κεφάλαιο 3 παρουσιάζεται η τελική υλοποίηση με στόχο φορητές μικροσυσκευές[3] με λειτουργικό σύστημα Android, με τη χρήση της προγραμματιστικής βιβλιοθήκης του Android. Στο ίδιο κεφάλαιο περιγράφονται τα προβλήματα που ανέκυψαν κατά στη διαδικασία των δοκιμών της τελικής εφαρμογής.

[3] Για παράδειγμα: κινητά τηλέφωνα, tablets

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τέλος, στο Κεφάλαιο 4, επιχειρείται η εξαγωγή ορισμένων συμπερασμάτων. Επίσης, στο ίδιο κεφάλαιο γίνεται αναφορά στα σημεία της υλοποίησης που χρήζουν επέκτασης και περαιτέρω έρευνας.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

2. Σχεδιασμός-Μοντελοποίηση

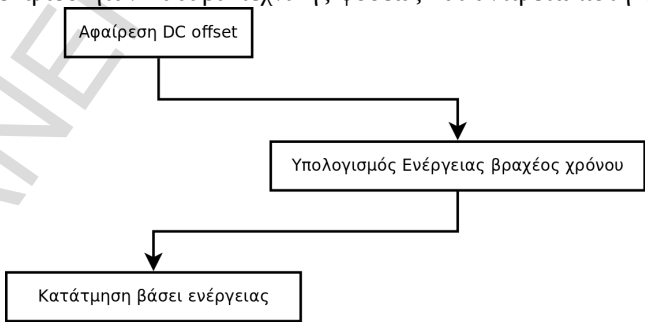
Όπως προκύπτει και από τη διατύπωση του στόχου της εργασίας (βλ. Ενότητα 1.1), καθίσταται σαφές ότι πρόκειται για πρόβλημα πολυδιάστατο, καθώς απαιτείται ανάλυση των διακριτών σταδίων (βλ. Εικόνα 1) και ξεχωριστή αντιμετώπισή τους.

Ενώ η είσοδος του υπό εξέταση ηχητικού σήματος και η ψηφιοποίηση μπορούν να θεωρηθούν λυμένα από το υποκείμενο υλικό (κάρτα ήχου), οι διαδικασίες της κατάτμησης, της εξαγωγής χαρακτηριστικών και της ομαδοποίησης των αποτελεσμάτων αποτελούν αντικείμενα ειδικής πραγμάτευσης. Σε κάθε ένα από αυτά θα πρέπει να επιτευχθεί η προσδοκώμενη ισορροπία μεταξύ αποτελεσματικότητας και αποδοτικότητας του τελικού αλγορίθμου. Στόχος δεν είναι η απόλυτη ακρίβεια των αποτελεσμάτων. Το επιθυμητό είναι ένας επαρκής αλγόριθμος, ελάχιστα απαιτητικός σε κατανάλωση υπολογιστικών πόρων.

Κάθε μία από τις ενότητες που ακολουθούν (2.1, 2.2 και 2.3) πραγματεύεται ένα από τα παραπάνω προβλήματα. Πιο συγκεκριμένα, η Ενότητα 2.1 αναφέρεται στη διαδικασία κατάτμησης του ηχητικού σήματος, ούτως ώστε να απομονωθεί το χρήσιμο σήμα από τις παύσεις. Στην Ενότητα 2.2 περιγράφεται η διαδικασία εξαγωγής των απαραίτητων χαρακτηριστικών των χρήσιμων τμημάτων του σήματος. Τα χαρακτηριστικά αυτά αποτελούν τα κριτήρια σύμφωνα με τα οποία θα γίνει η ομαδοποίηση των τμημάτων σε ξεχωριστές ομάδες, διαδικασία που περιγράφεται στην Ενότητα 2.3. Υποθέτοντας ότι τα τμήματα του ήχου είναι κρουστικοί ήχοι που προκαλούνται με οποιοδήποτε μέσο διαθέτει ο τελικός χρήστης, ο σκοπός της ομαδοποίησης είναι η απόδοση κάθε τέτοιου κρουστικού ήχου στο κατάλληλο τύμπανο. Επίσης περιγράφεται ο τρόπος αντιμετώπισής του στη φάση του σχεδιασμού της τελικής εφαρμογής.

Ο οξυδερκής αναγνώστης ίσως να παρατήρησε το γεγονός ότι κατά την ανάλυση δεν γίνεται ρητή μνεία στο τελικό στάδιο της εφαρμογής, ήτοι την σύνθεση – αναπαραγωγή του τελικού ήχου. Αυτό συμβαίνει επειδή στόχος είναι απλώς μια ενδεικτική αναπαραγωγή, ώστε να καταστεί δυνατή η με κάποιο τρόπο πληροφόρηση από την τελική εφαρμογή στον χρήστη για τα αποτελέσματα της προηγηθείσας επεξεργασίας. Μια τέτοιου είδους πληροφόρηση θα μπορούσε να γίνει γραφικά, με την παράθεση κάποιων πινάκων με τα αποτελέσματα της ομαδοποίησης, ή με τη χρήση ενός μουσικού πενταγράμμου. Ωστόσο, εν τέλει προτιμήθηκε μια -έστω υποτυπώδης- αναπαραγωγή του τελικού αποτελέσματος με τη χρήση έτοιμων ηχητικών δειγμάτων, αφού με αυτόν τον τρόπο είναι πιο εύκολη και άμεση η αξιολόγηση από τον τελικό χρήστη.

Συνοψίζοντας, τα όποια ειδικά προβλήματα προέκυψαν κατά την υλοποίηση της αναπαραγωγής του τελικού ηχητικού δοκιμίου ήταν καθαρά τεχνικής φύσεως και αντιμετωπίστηκαν στην τελική φάση της



Εικόνα 2: Βασικό διάγραμμα ροής για την κατάτμηση

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

υλοποίησης για την πλατφόρμα Android, ενώ οι θεμελιώδεις μεθοδεύσεις που απαιτούνται κατά τη διαδικασία του ηχητικού μοντάζ έτοιμων δειγμάτων[4] αντιμετωπίζονται από την προγραμματιστική βιβλιοθήκη του Android.

2.1 Εντοπισμός κρουστικών τμημάτων του σήματος

Η διαδικασία της κατάτμησης (segmentation) κρίνεται απαραίτητη στην παρούσα ανάλυση καθώς μέσω αυτής επιτελείται ο διαχωρισμός των χρήσιμων ηχητικών αποσπασμάτων από τις παύσεις. Ο σκοπός αυτής της κατάτμησης είναι η αντιμετώπιση κάθε ηχητικού συμβάντος (το οποίο οριοθετείται από ένα τμήμα του ήχου που έχει προκύψει από την κατάτμηση) ως ξεχωριστή οντότητα κατά τα επόμενα στάδια της εξαγωγής χαρακτηριστικών και ομαδοποίησης.

Κρίσιμη έννοια εδώ είναι η έννοια του ηχητικού συμβάντος ή γεγονότος (event). Με τον όρο αυτό εννοείται ακριβώς ένα τμήμα του ήχου όπου θεωρείται χρήσιμο για τον αλγόριθμο. Στην παρούσα πραγμάτευση ως συμβάν θεωρείται οποιοσδήποτε κρουστικός ήχος. Ωστόσο, η υπό ανάλυση προσέγγιση λαμβάνει ως παραδοχή ότι κάθε συμβάν αντιστοιχεί σε κρουστικό ήχο και το αντιμετωπίζει ως τέτοιο.

Μια ακόμα παραδοχή που υιοθετήθηκε είναι ότι τα πρώτα 100 msec[5] του ηχογραφημένου ήχου οπωσδήποτε δεν περιέχουν χρήσιμο σήμα, δεδομένου ότι σε αυτό το μικρό για τα ανθρώπινα δεδομένα χρονικό διάστημα δεν έχει προλάβει να παραχθεί κάποιο χρήσιμο ηχητικό συμβάν από την πλευρά του χρήστη. Ο αλγόριθμος, λοιπόν, εκμεταλλεύεται το γεγονός αυτό ούτως ώστε να πραγματοποιήσει μια ανάλυση σε επίπεδο ενέργειας του θορύβου υποβάθρου (background noise). Ουσιαστικά, σκοπός εδώ είναι η εξαγωγή μιας μέσης στάθμης θορύβου υποβάθρου που θα χρησιμέψει στον διαχωρισμό του χρήσιμου σήματος από το θόρυβο υποβάθρου. Σε αυτό το σημείο υπεισέρχεται και η τρίτη παραδοχή, σύμφωνα με την οποία το χρήσιμο σήμα θα πρέπει να είναι τουλάχιστον κατά 10 dB ισχυρότερο από το θόρυβο υποβάθρου. Η κλίμακα των decibel, ως γνωστόν είναι σχετικό μέγεθος και εξαρτάται από ένα κατώφλι αναφοράς, ήτοι:

$$x\,dB \;=\; 10\cdot\log\left(\frac{x_{\mu\varepsilon\tau\rho o\acute{u}\mu\varepsilon\nu o}}{x_{\alpha\nu\alpha\varphi o\rho\acute{\alpha}\varsigma}}\right)$$

Συμπερασματικά, στην παρούσα υλοποίηση θεωρείται ότι ο θόρυβος υποβάθρου πρέπει να είναι τουλάχιστον κατά 10 dB ασθενέστερος από το χρήσιμο σήμα, διαφορετικά ο αλγόριθμος κατάτμησης αποτυγχάνει. Αυτό σημαίνει ότι ο υπό παρουσίαση αλγόριθμος δεν είναι κατάλληλος για πολύ θορυβώδη σήματα.

Επίσης, ο αλγόριθμος κατάτμησης δεν είναι ικανός να ξεχωρίσει επικαλυπτόμενα ηχητικά συμβάντα, ή τις περιπτώσεις πολύ γρήγορων εναλλαγών μεταξύ κρουστικών ήχων. Σε αυτές τις περιπτώσεις, τα επικαλυπτόμενα ή τα πολύ κοντινά στο πεδίο του χρόνου ηχητικά συμβάντα προσμετρώνται ως ένα, του οποίου ο χρόνος έναρξης ισοδυναμεί με τον χρόνο έναρξης του προπορευόμενου χρονικά συμβάντος, ενώ ο χρόνος λήξης ισοδυναμεί με τον χρόνο λήξης του τελευταίου χρονικά συμβάντος.

Ήδη από τα προαναφερθέντα μπορεί να συναχθεί ότι προκειμένου να γίνει η κατάτμηση των συμβάντων, θα πρέπει να έχουμε στη διάθεσή μας τουλάχιστον ένα χαρακτηριστικό του υπό εξέταση ηχητικού αποσπάσματος. Το βασικότερο χαρακτηριστικό που είναι απαραίτητο για την επιτυχή κατάτμηση σε τυπικές περιπτώσεις ηχητικών σημάτων είναι το μέγεθος της ενέργειας του σήματος. Η ενέργεια ενός διακριτού σήματος χαρακτηρίζεται από το βαθμό στον οποίο αποκλίνουν οι διακριτές τιμές του από το σημείο μηδέν ή σημείο ισορροπίας. Πρόκειται δηλαδή για ένα μέγεθος που εκφράζει την ένταση ή ισχύ ενός τμήματος του σήματος και υπολογίζεται από τον παρακάτω τύπο:

[4] π.χ. Αντιμετώπιση ασυνεχειών του σήματος

[5] Παραδοχή που υιοθετήθηκε από τον αλγόριθμο κατάτμησης ομιλίας του Rabiner.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

$E = \frac{1}{N}\sum_{i=1}^{N} x_i^2$, όπου N: το πλήθος των δειγμάτων του σήματος και x_i : η τιμή του i-οστού σε σειρά δείγματος του σήματος.

Φυσικά μία και μοναδική τιμή ενέργειας που θα χαρακτηρίζει ολόκληρο το σήμα δεν θα ήταν καθόλου χρήσιμη για την κατάτμηση του ήχου. Η συνήθης πρακτική κατά την ανάλυση ενός ήχου είναι ο υπολογισμός χαρακτηριστικών όπως η ενέργεια σε σύνολα διαδοχικών δειγμάτων, τα λεγόμενα πλαίσια (frames). Το σύνολο των τιμών της ενέργειας για κάθε πλαίσιο του σήματος καλείται *ενέργεια βραχέος χρόνου (short-time energy)*.

Τα πλαίσια χαρακτηρίζονται από δύο μεγέθη:

- **Μέγεθος πλαισίου**: Το μέγεθος του πλαισίου σε πλήθος δειγμάτων ή σε χρονική διάρκεια
- **Ολίσθηση πλαισίου**: Θετικό μέγεθος εκφρασμένο σε πλήθος δειγμάτων ή σε χρονική διάρκεια που υποδεικνύει τη μετατόπιση κάθε επόμενου πλαισίου σε σχέση με το προηγούμενο στο πεδίο του χρόνου. Εάν η ολίσθηση πλαισίου < μέγεθος πλαισίου, τότε τα πλαίσια είναι ως ένα βαθμό αλληλοεπικαλυπτόμενα.

Για παράδειγμα, έστω ότι ορίζουμε το μέγεθος πλαισίου στα 400 δείγματα και την ολίσθηση πλαισίου στα 100 δείγματα. Το πρώτο πλαίσιο θα εκτείνεται από το 1ο έως και το 400οστό δείγμα, το δεύτερο από το 101ο δείγμα έως και το 500οστό κ.ο.κ.

Αρκετά συνηθισμένες τιμές των δύο αυτών παραμέτρων είναι 40 msec για το μέγεθος του πλαισίου και 10 msec έως 20 msec για την ολίσθηση του πλαισίου. Στην παρούσα προσέγγιση, κατόπιν και πειραματισμών το μέγεθος του πλαισίου ορίσθηκε στα 40 msec και η ολίσθηση του πλαισίου στα 20 msec. Οι παραπάνω τιμές διαπιστώθηκε ότι παρείχαν ικανοποιητικά αποτελέσματα για το είδος της επεξεργασίας που τίθεται ως στόχος στην παρούσα εργασία.

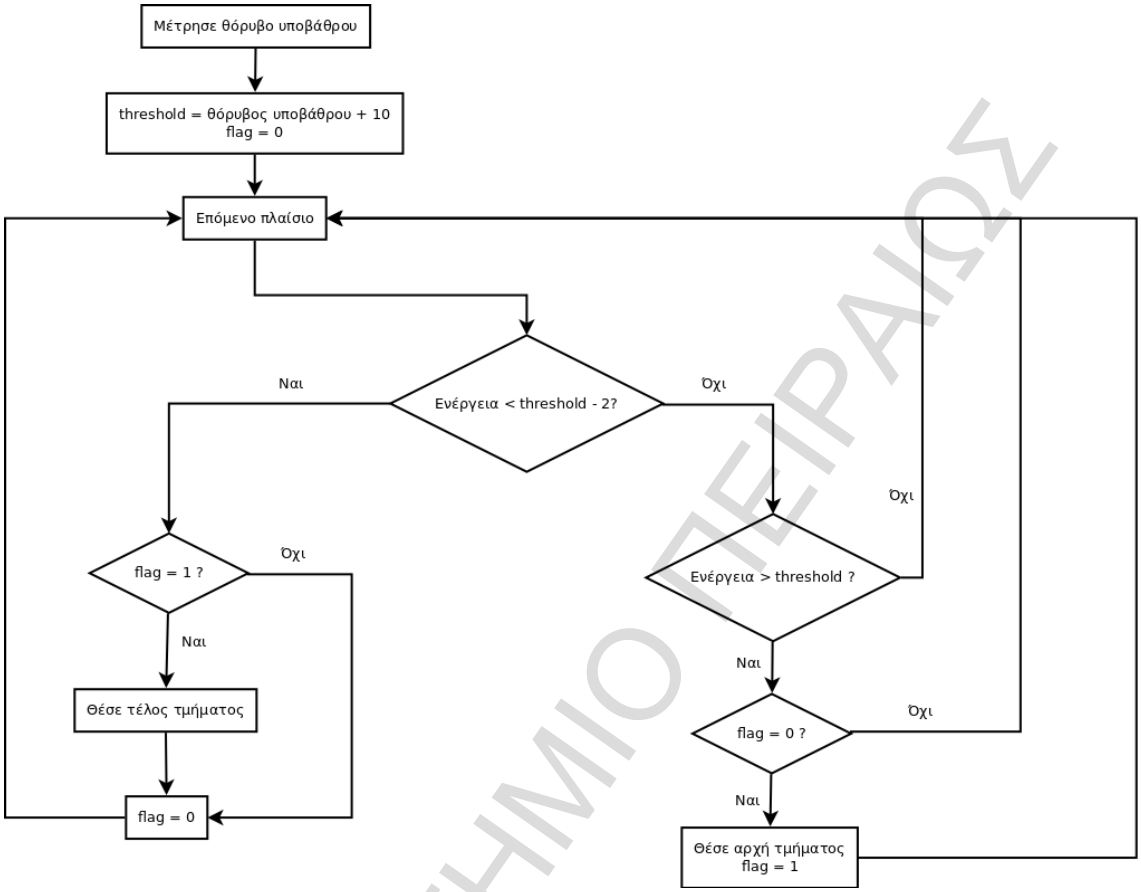
Ουσιαστικά, υπάρχει μια δεδομένη και αναγκαία επικάλυψη μεταξύ του δεύτερου σταδίου της εργασίας (δηλαδή του σταδίου της εξαγωγής των χαρακτηριστικών) και του πρώτου σταδίου που παρουσιάζεται εδώ, δηλαδή του σταδίου της κατάτμησης των ηχητικών συμβάντων. Το χαρακτηριστικό της ενέργειας βραχέος χρόνου είναι ένα χαρακτηριστικό που θα ήταν απαραίτητο να υπολογιστεί κατά τη διαδικασία της εξαγωγής των χαρακτηριστικών, καθώς θα είναι ένα από τα χαρακτηριστικά που θα ληφθεί υπόψη για την διαδικασία της ομαδοποίησης των συμβάντων. Το γεγονός ότι είναι απαραίτητο και για τη διαδικασία της κατάτμησης ελάχιστα διαφοροποιεί τον σχεδιασμό. Απλώς οι τιμές της ενέργειας διατηρούνται ώστε να χρησιμοποιηθούν και στην διαδικασία της ομαδοποίησης, έτσι ώστε να μην χρειάζεται να υπολογιστούν εκ νέου.

Πριν τον υπολογισμό της ενέργειας βραχέος χρόνου, όπως φαίνεται και από το βασικό διάγραμμα στην Εικόνα 2, προηγείται μια άλλου είδους επεξεργασία, γνωστή ως “κεντράρισμα” της κυματομορφής ή ως *αφαίρεσης του DC offset*. Κατά την ηχογράφηση ενός ηχητικού σήματος, μπορεί να συμβεί η κυματομορφή να είναι κατά κάποιο τρόπο μετατοπισμένη στον κάθετο άξονα, ούτως ώστε αν είχαμε μια απλή περιοδική ταλάντωση αυτή να μην ήταν γύρω από το μηδέν, αλλά γύρω από το +0.2, για παράδειγμα. Τότε, λέγεται ότι το σήμα έχει DC offset 0.2, υποθέτοντας ότι ενώ η αιτία της διακύμανσης της τιμής του σήματος είναι μια εναλλασσόμενη (AC) πηγή, ενώ η αιτία της κάθετης μετατόπισης της κυματομορφής είναι μια DC πηγή. Το παραπάνω είναι ένα συνηθισμένο φαινόμενο κατά την ηχογράφηση και οφείλεται στο υποκείμενο υλικό.

Η καλύτερη πρακτική είναι η αφαίρεση του DC offset καθώς μπορεί να οδηγήσει σε λανθασμένο υπολογισμό των χαρακτηριστικών του σήματος.

Εφόσον έχουν πραγματοποιηθεί τα προπαρασκευαστικά βήματα (αφαίρεση DC offset) και έχουν υπολογιστεί οι τιμές της ενέργειας βραχέος χρόνου, είναι ώρα για την εφαρμογή του βασικού αλγόριθμου για τον εντοπισμό των κατάλληλων τμημάτων του ήχου.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



Εικόνα 3: Διάγραμμα ροής για τον εντοπισμό των τμημάτων του ήχου

Πρώτο βήμα είναι ο υπολογισμός της στάθμης του θορύβου βάθους, με τη χρήση των τιμών ενέργειας βραχέος χρόνου που αντιστοιχούν στα πρώτα $100 \cdot 10^{-3}\, sec$. Μέσω της παραπάνω στάθμης υπολογίζεται η στάθμη – κατώφλι, όπου τα τμήματα του σήματος που υπερβαίνουν το κατώφλι αυτό θα θεωρούνται ως χρήσιμο σήμα. Όπως ήδη αναφέρθηκε, η στάθμη αυτή ορίζεται να είναι κατά 10 decibel ισχυρότερη από τη στάθμη του θορύβου βάθους[6].

Χρησιμοποιώντας τη στάθμη κατωφλίου, στη συνέχεια, πραγματοποιείται γραμμική σάρωση όλων των τιμών ενέργειας. Σε περίπτωση που παρατηρηθεί τιμή ενέργειας μεγαλύτερη από τη στάθμη κατωφλίου, σηματοδοτείται η έναρξη ενός τμήματος χρήσιμου ήχου. Όταν, στη συνέχεια η στάθμη πέσει κάτω από τη στάθμη κατωφλίου – 2 dB[7], σηματοδοτείται η λήξη ενός τμήματος χρήσιμου ήχου.

[6] Στο σημείο αυτό η προσέγγιση του Rabiner λαμβάνει υπόψη της και άλλα χαρακτηριστικά του θορύβου βάθους, όπως είναι η τυπική απόκλιση που παρουσιάζουν οι στάθμες ενέργειας των πρώτων 100 msec. Για λόγους απλότητας, εφόσον τελικός στόχος είναι η μεταφορά του αλγορίθμου σε μικροσυσκευές με περιβάλλον Android, η προσέγγιση αυτή απλοποιήθηκε. Η στάθμη κατωφλίου ορίστηκε -κατόπιν πειραματισμού- να είναι +10dB από τη στάθμη του θορύβου βάθους.

[7] Αποφασίστηκε ο αλγόριθμος να είναι ελαφρώς πιο “ανεκτικός” όσον αφορά το τμήμα του χρόνου πτώσης του κρουστικού ήχου, καθώς έτσι λαμβάνεται υπόψη μεγαλύτερο κομμάτι των αντηχήσεων που μπορούν να διαμορφώσουν τα χαρακτηριστικά του τελικού ήχου. Η υπόθεση αυτή επιβεβαιώθηκε κατόπιν πειραματισμών με τα ηχητικά δοκίμια που διατίθενται μαζί με την τελική υλοποίηση.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Προκειμένου να αξιολογηθεί η απόδοση του αλγορίθμου χρησιμοποιήθηκε η σειρά των ηχογραφήσεων του διαγωνισμού Query by Tapping της MIREX (890 αρχεία ηχογραφήσεων κρουστικών ήχων), καθώς και η αντίστοιχη σειρά των onsets κάθε αρχείου (890 αρχεία κειμένου, όπου αναγράφεται ο χρόνος εμφάνισης κάθε onset σε millisecond).

Παρά το γεγονός ότι η προσέγγιση των onsets είναι διαφορετική από την προσέγγιση του υπό παρουσίαση αλγόριθμου (αφού στον υπό παρουσίαση αλγόριθμο δεν γίνεται εντοπισμός των onsets αλλά της έναρξης και της λήξης κάθε κρουστικού ήχου), μπορεί να επιτευχθεί μια κάποια σύγκριση, δεχόμενοι ως επιτυχία κάθε σημειωμένο onset να περιβάλλεται από τους αντίστοιχους χρόνους έναρξης και λήξης στα αποτελέσματα του υπό παρουσίαση αλγόριθμου. Δηλαδή, κάθε εντοπισμένο κρουστικό τμήμα από τον αλγόριθμο αρκεί να αντιστοιχεί σε ένα καταγεγραμμένο onset. Σε περίπτωση που ένα εντοπισμένο κρουστικό τμήμα δεν αντιστοιχεί σε κάποιο χρόνο onset, τότε αξιολογείται ως αστοχία του αλγόριθμου τύπου εσφαλμένης αποδοχής (False Alarm). Αντιστοίχως, εάν κάποιο σημειωμένο onset δεν αντιστοιχεί σε κάποιο εντοπισμένο τμήμα, τότε αξιολογείται ως αστοχία του αλγόριθμου τύπου εσφαλμένης απόρριψης (False Reject).

Τμήμα (ms)	**Onset (ms)**	**Σφάλμα?**
800 – 940	833	Όχι
1160 – 1300	1189.50	Όχι
1500 – 1660	-	False Alarm
-	1740	False Reject

Εικόνα 4: Παράδειγμα αξιολόγησης των αποτελεσμάτων. Όταν κάποιο onset δεν αντιστοιχεί σε κάποιο εντοπισμένο τμήμα, τότε έχουμε False Reject, ενώ όταν κάποιο εντοπισμένο τμήμα δεν αντιστοιχεί σε κάποιο onset τότε έχουμε False Alarm.

Όπως αναφέρθηκε και προηγουμένως, ο υπό παρουσίαση αλγόριθμος αποτυγχάνει σε περιπτώσεις πολύ γρήγορων διαδοχικών συμβάντων (πολύ κοντινά χρονικά onset στη σειρά ηχογραφήσεων), καθώς και σε θορυβώδη σήματα. Τα αποτελέσματα της αξιολόγησης επιβεβαιώνουν τις παραπάνω εκτιμήσεις.

Παρατηρήθηκαν αρκετά διαδοχικά onsets που αντιστοιχούσαν σε μόνο ένα εντοπισμένο από τον αλγόριθμο τμήμα. Αυτό συμβαίνει επειδή στην περίπτωση των πολύ κοντινών διαδοχικών onset, μετά την πάροδο του πρώτου συμβάντος, η στάθμη της ενέργειας βραχέος χρόνου δεν προλαβαίνει να πέσει κάτω από το θεωρούμενο από τον αλγόριθμο κατώφλι ενέργειας, πριν το δεύτερο συμβάν. Αυτό το πρόβλημα θα μπορούσε να αντιμετωπιστεί με τον υπολογισμό της παραγώγου της περιβάλλουσας της ενέργειας (σε συνδυασμό με κάποιο φίλτρο μεσαίων για την εξομάλυνση της καμπύλης της ενέργειας), μια προσέγγιση όμως που δεν προτιμήθηκε από την παρούσα υλοποίηση, προκειμένου να δοθεί περισσότερη βαρύτητα στην απόδοση.

Επίσης, στις περιπτώσεις των πιο θορυβωδών ηχογραφήσεων, ο αλγόριθμος απέτυχε να εντοπίσει τα τμήματα του χρήσιμου ήχου. Αυτό συνέβη εξαιτίας της υψηλής στάθμης του θορύβου βάθους που αντίστοιχα ανέβασε πολύ ψηλά την στάθμη κατωφλίου. Εκτός, όμως, από τα παραπάνω είδη αστοχιών του αλγόριθμου, υπήρξε ένα μεγάλο ποσοστό αστοχιών εσφαλμένης αποδοχής. Αυτές οφείλονται στα μεγάλα ποσοστά διακύμανσης της στάθμης του θορύβου βάθους, που ορισμένες φορές ξεπερνούσαν τη στάθμη κατωφλίου. Μια πιο συντηρητική στάθμη κατωφλίου θα περιόριζε αυτό το είδος αστοχιών, θα αύξανε, όμως, τις περιπτώσεις αστοχιών λόγω υψηλής στάθμης του θορύβου βάθους.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Συνολικά, σε σύνολο **27806 καταγεγραμμένων onset**, κατά την αξιολόγηση του αλγόριθμου παρατηρήθηκαν **3991 false alarms** και **1515 false rejects**. Δεδομένου ότι βασικός στόχος του αλγόριθμου είναι η ταχύτητα των επεξεργασιών τα αποτελέσματα κρίνονται ικανοποιητικά.

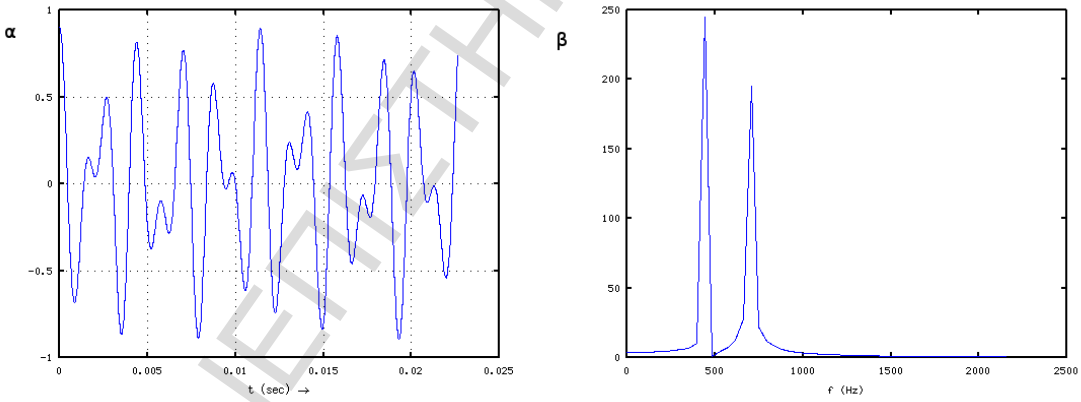
2.2 Εξαγωγή χαρακτηριστικών

Αφού έχει επιτευχθεί η κατάτμηση των κρουστικών συμβάντων, επόμενο βήμα είναι η εξαγωγή των χαρακτηριστικών των ηχητικών συμβάντων βάσει των οποίων θα πραγματοποιηθεί η ομαδοποίησή τους. Για τον σκοπό αυτό, ένα χρήσιμο χαρακτηριστικό είναι η ενέργεια βραχέος χρόνου. Το ευχάριστο είναι ότι η ενέργεια βραχέος χρόνου έχει ήδη υπολογισθεί από το πρώτο στάδιο του αλγόριθμου. Συνεπώς, προκειμένου να χρησιμοποιηθεί κατά την διαδικασία της ομαδοποίησης, αρκεί να γίνει χρήση των κατάλληλων μόνο τιμών, αυτών δηλαδή που αντιστοιχούν στα τμήματα που οριοθετούν τα ηχητικά συμβάντα.

Εκτός από την ενέργεια βραχέος χρόνου, υπάρχει πλήθος χαρακτηριστικών που είναι δυνατόν να εξαχθούν από τα συμβάντα προκειμένου να βοηθήσουν στην ομαδοποίηση. Εκτός από την ενέργεια, αποφασιστικό ρόλο στη διάκριση ενός ήχου από έναν άλλο παίζουν τα χαρακτηριστικά που παραπέμπουν στο συχνοτικό τους περιεχόμενο. Ουσιαστικά, η ανθρώπινη ακοή και αντίληψη μπορεί πολύ εύκολα να διακρίνει δύο ήχους βάσει του τονικού τους ύψους, δηλαδή της συχνότητάς τους. Για να λάβουμε πληροφορίες για το συχνοτικό περιεχόμενο ενός ψηφιακού σήματος x(n) θα πρέπει να εφαρμόσουμε τον διακριτό μετασχηματισμό Fourier:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot \mathrm{e}^{-j\omega kn}$$

Στην Εικόνα 5 φαίνεται ένα παράδειγμα μετασχηματισμού Fourier σε ένα σήμα.



Εικόνα 5: α) Ένα σήμα στο πεδίο του χρόνου β) Το ίδιο σήμα στο πεδίο των συχνοτήτων, μετά από μετασχηματισμό Fourier

Μέσω του μετασχηματισμένου σήματος, μπορούν να εξαχθούν χαρακτηριστικά, όπως η θεμελιώδης συχνότητα ή το φασματικό κεντροειδές. Η θεμελιώδης συχνότητα, συνήθως, αντιστοιχεί στη συχνότητα που παρουσιάζει το μεγαλύτερο πλάτος. Για παράδειγμα, στην Εικόνα 5β η θεμελιώδης συχνότητα βρίσκεται στα 440 Hz. Η θεμελιώδης συχνότητα δεν παρέχει καμία πληροφορία για το υπόλοιπο συχνοτικό περιεχόμενο. Ενώ για κάποιες εφαρμογές το χαρακτηριστικό της θεμελιώδους συχνότητας ίσως να ήταν αρκετό (π.χ. Έλεγχος δονήσεων μηχανημάτων, όπου καίριο ρόλο για την πρόβλεψη μελλοντικών αστοχιών υλικού παίζει η παρακολούθηση της συχνότητας περιστροφής των ρουλεμάν), για

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

τον τομέα της ανάλυσης ήχων χωρίς σαφές τονικό περιεχόμενο (όπως είναι η πλειονότητα των κρουστικών ήχων) η θεμελιώδης συχνότητα συχνά απουσιάζει.

Το φασματικό κεντροειδές υπολογίζεται μέσω του μετασχηματισμένου σήματος X(k). Το σήμα αυτό εύκολα συμπεραίνεται πως αποτελείται από τους συντελεστές X(k) για k = 0, 1, ...,N-1. Για τον υπολογισμό του φασματικού κεντροειδούς, ορίζεται το σήμα $X'(k) = X(k)$ *για* $k = 0,1,...m$, όπου $m = \frac{N-1}{2}$.

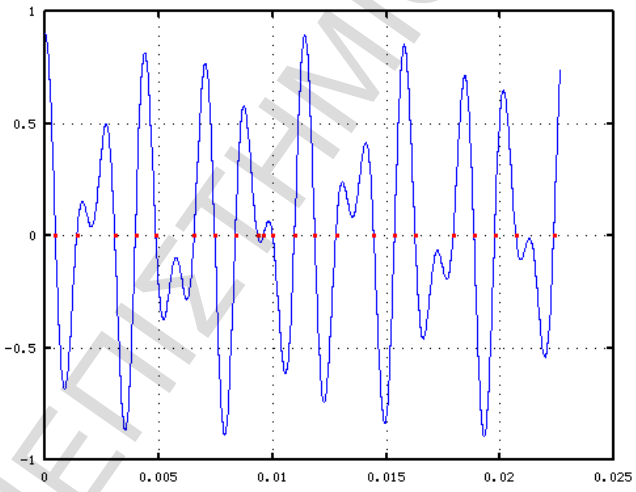
Στη συνέχεια, υπολογίζουμε τους κανονικοποιημένους συντελεστές λαμβάνοντας το μέτρο των μιγαδικών συντελεστών X'(k) και διαιρώντας τον καθένα με το άθροισμα όλων των συντελεστών:

$$\hat{X}(k) = \frac{|X'(k)|}{\sum_{l=0}^{m} |X(l)|}$$

Τέλος, υπολογίζεται το φασματικό κεντροειδές με τη χρήση του τύπου:

$fc = \sum_{k=0}^{k=m} k \frac{fs}{N} \cdot \hat{X}(k)$, όπου fs: η συχνότητα δειγματοληψίας.

Είναι εμφανές ότι το φασματικό κεντροειδές αποτελεί έναν πιο αντιπροσωπευτικό δείκτη του φασματικού περιεχομένου ενός σήματος. Ωστόσο, ο υπολογισμός του είναι αρκετά απαιτητικός σε επεξεργαστική ισχύ. Για αυτό το λόγο στην παρούσα υλοποίηση, αντί του φασματικού κεντροειδούς



Εικόνα 6: Ο ρυθμός διέλευσης από το μηδέν αποτελεί μια χρήσιμη ένδειξη για το συχνοτικό περιεχόμενο ενός σήματος. Στην εικόνα οι διελεύσεις από το μηδέν είναι σημειωμένες με κόκκινες τελείες.

χρησιμοποιήθηκε ο Ρυθμός Διέλευσης από το Μηδέν (Zero Crossing Rate - ZCR). Το ZCR είναι ένας αρκετά αξιόπιστος δείκτης και μάλιστα ο υπολογισμός του είναι πολύ πιο εύκολος. Παρόλο που το ZCR είναι ένα μέγεθος που παραπέμπει στο συχνοτικό περιεχόμενο ενός σήματος, υπολογίζεται στο πεδίο του χρόνου. Έτσι, δεν είναι απαραίτητος ο μετασχηματισμός Fourier.

Το ZCR ενός πλαισίου υπολογίζεται από τον παρακάτω τύπο:

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

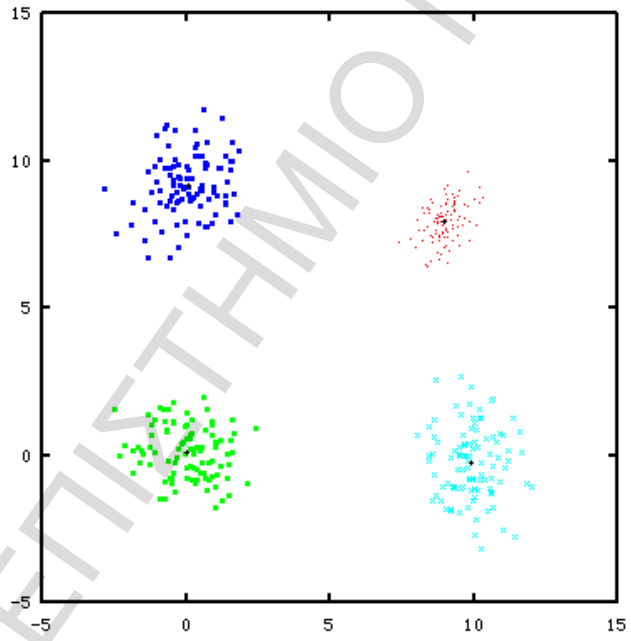
$ZCR = \frac{1}{2N}\sum_{n=1}^{N}|sign(x[n])-sign(x[n-1])|$, όπου N: το μέγεθος του πλαισίου σε πλήθος δειγμάτων και

$$sign(x)=\begin{cases}-1, & \alpha\nu\ x<0\\ +1, & \alpha\nu\ x\geqslant 0\end{cases}.$$

Ουσιαστικά, το ZCR εξαρτάται από το πλήθος των μεταβολών του πρόσημου στις τιμές των δειγμάτων. Εάν στο σήμα έχει προηγηθεί η επεξεργασία αφαίρεσης του DC offset, τότε το ZCR αποτελεί μια καλή ένδειξη για το συχνοτικό περιεχόμενο του σήματος.

Το Zero Crossing Rate χρησιμοποιείται αρκετά συχνά σε αλγόριθμους αναγνώρισης ομιλίας, ως μια μέθοδος διαχωρισμού – κατάτμησης του διαστήματος ομιλίας από τα διαστήματα όπου υπάρχει παύση. Εδώ, γίνεται χρήση της μεθόδου ZCR προκειμένου να αποφευχθεί η χρήση του Μετασχηματισμού Fourier.

Συνοψίζοντας, κάθε ηχητικό συμβάν που έχει εντοπισθεί μέσω της κατάτμησης πρόκειται να λειτουργήσει ως είσοδος στον αλγόριθμο ομαδοποίησης που περιγράφεται στην Ενότητα 2.3. Κάθε



Εικόνα 7: Αποτελέσματα ομαδοποίησης με τη χρήση του αλγόριθμου k-Means
Πηγή: Theodoridis S., Pikrakis A., Koutroumbas K., Cavouras D. "Introduction to Pattern Recognition"

τέτοιο συμβάν – παρατήρηση θα χαρακτηρίζεται από δύο χαρακτηριστικά:

- Ενέργεια
- Ρυθμός διέλευσης από το μηδέν (Zero Crossing Rate)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

2.3 Ομαδοποίηση

Το τελευταίο βήμα του αλγόριθμου είναι η ομαδοποίηση των συμβάντων – παρατηρήσεων προκειμένου η αναπαραγωγή του ρυθμού με τη χρήση έτοιμων ηχητικών δοκιμίων να γίνει με τη χρήση του κατάλληλου δοκιμίου για κάθε συμβάν.

Πρόκειται για κλασικό πρόβλημα ομαδοποίησης, καθώς δεν είναι δυνατή η ταξινόμηση των παρατηρήσεων σε ομάδες με ήδη γνωστά χαρακτηριστικά. Εν ολίγοις, τα χαρακτηριστικά των ήχων που θα ηχογραφηθούν δεν είναι εκ των προτέρων γνωστά, καθώς δεν είναι δυνατό να γίνει γνωστό το είδος των ήχων που θα χρησιμοποιηθεί από το χρήστη. Για παράδειγμα, δεν μπορούμε να ορίσουμε εκ των προτέρων αντιπροσωπευτικές ομάδες με συγκεκριμένες μέσες τιμές για την ενέργεια βραχέος χρόνου και το Zero Crossing Rate.

Προκειμένου να πραγματοποιηθεί η ομαδοποίηση των παρατηρήσεων, πρέπει να οριστούν τα δεδομένα των διανυσμάτων εισόδου για τον αλγόριθμο ομαδοποίησης. Δεδομένου ότι κάθε παρατήρηση αντιστοιχεί σε ένα ηχητικό συμβάν, όπως αυτά εντοπίσθηκαν μέσω της προσέγγισης που περιγράφηκε στην Ενότητα 2.1 και ότι κάθε παρατήρηση χαρακτηρίζεται και διαφοροποιείται από τις υπόλοιπες μέσω των χαρακτηριστικών που περιγράφηκαν στην Ενότητα 2.2, ως σύνολο για τον αλγόριθμο ομαδοποίησης θεωρείται ένα σύνολο διδιάστατων διανυσμάτων:

$$X = \{X_1, X_2, \ldots, X_N\},$$

όπου N: το πλήθος των εντοπισμένων συμβάντων – τμημάτων του ήχου.

Κάθε διάνυσμα X_n θεωρείται ως εξής:

$$X_n = \begin{bmatrix} E_n \\ ZCR_n \end{bmatrix},$$

όπου E_n : η μέση τιμή ενέργειας βραχέος χρόνου του n-οστού συμβάντος και ZCR_n : η μέση τιμή του ρυθμού διέλευσης από το μηδέν του n-οστού συμβάντος.

Δεδομένου ότι η ομαδοποίηση των κρουστικών συμβάντων πρέπει να είναι τύπου Hard Clustering -δηλαδή κάθε παρατήρηση πρέπει να ανήκει σε μια και μόνο ομάδα- και ότι -τουλάχιστον σε πρωταρχικό ερευνητικό στάδιο- μπορούμε να θεωρήσουμε ότι ο χρήστης της εφαρμογής της φορητής μικροσυσκευής δύσκολα θα έχει στη διάθεσή του πάνω από δύο πηγές ήχου, επιλέχθηκε ως αλγόριθμος ομαδοποίησης ο k-Means. Ο αλγόριθμος k-Means είναι αλγόριθμος για Hard Clustering και θεωρεί το πλήθος των ομάδων γνωστό εκ των προτέρων, συνεπώς φαίνεται κατάλληλος για το είδος της εφαρμογής.

Ένα ακόμη στοιχείο που οδήγησε στην επιλογή του k-Means ως αλγόριθμο ομαδοποίησης είναι η απλότητα του συγκεκριμένου αλγόριθμου που τον καθιστά ελάχιστα απαιτητικό σε υπολογιστικούς πόρους.

Επίσης, ο αλγόριθμος k-Means είναι κατάλληλος για τη διάκριση αρκετά “συμπυκνωμένων” στο χώρο-διάστημα των παρατηρήσεων ομάδων. Λόγω της φύσης του προβλήματος, όπου υπάρχει η υπόθεση ότι ο χρήστης δεν θα διαθέτει εξειδικευμένες πηγές ήχου για την παραγωγή του επιθυμητού ρυθμού, είναι πιθανό οι τιμές των χαρακτηριστικών να μην διαφέρουν σημαντικά.

Ο αλγόριθμος k-Means αρχικά θεωρεί ένα πλήθος διανυσμάτων

$\theta_1, \theta_2, \ldots, \theta_m$, όπου m: το πλήθος των ομάδων (όπως αναφέρθηκε και παραπάνω, το πλήθος των ομάδων είναι γνωστό εκ των προτέρων).

Στο σύνολο των διανυσμάτων αυτών αποδίδει τυχαίες αρχικές τιμές. Στη συνέχεια, ο αλγόριθμος λειτουργεί επαναληπτικά, όπου σε κάθε επανάληψη εισάγεται σε κάθε ομάδα η παρατήρηση που βρίσκεται πλησιέστερα στην αντίστοιχη τιμή θ_n , σε σχέση με τις υπόλοιπες, μη ομαδοποιημένες

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

παρατηρήσεις. Η τιμή θ_n ενημερώνεται ώστε να ισούται με την μέση τιμή των διανυσμάτων που έχουν εισαχθεί στη συγκεκριμένη ομάδα.

Αν σε κάποια επανάληψη δεν παρατηρηθεί καμία μεταβολή στις τιμές $\theta_{1,}\theta_{2,}\ldots,\theta_m$ ο αλγόριθμος τερματίζεται.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

3. Υλοποίηση

Στις ενότητες που ακολουθούν περιγράφεται η τελική υλοποίηση της εφαρμογής στην πλατφόρμα Android.

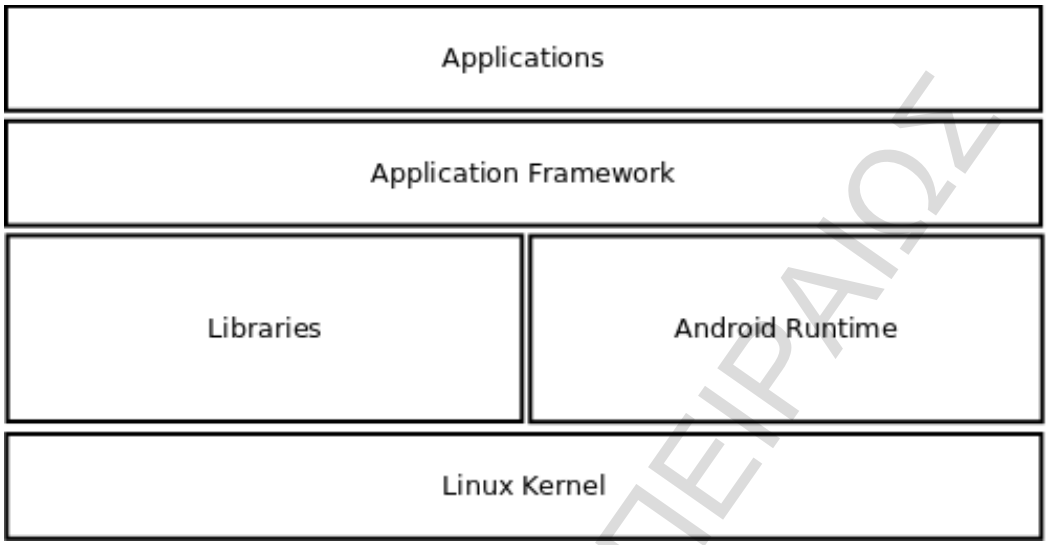
Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα με στόχο τις φορητές μικροσυσκευές. Αρκετά διαδεδομένο είναι στην αγορά των κινητών τηλεφώνων (smartphones). Μάλιστα, τα τελευταία χρόνια το Android παρουσιάζει ανοδική πορεία στο μερίδιο αγοράς, ενώ οι ανταγωνιστές του φαίνεται να κατακρημνίζονται. Υπάρχουν εκτιμήσεις που αποδίδουν ποσοστό κοντά στο 80% για το μερίδιο αγοράς του Android.

Εικόνα 8: Το εμπορικό λογότυπο του Android

Αρχικά η ανάπτυξη του Android προωθήθηκε από τη Google και στη συνέχεια από την Open Handset Alliance. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007 παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Μεταξύ των 48 αυτών εταιρειών, μέλη του οργανισμού αυτού, είναι και οι εταιρείες Sprint Nextel , T-Mobile, Motorola, Samsung, Sony Ericsson, Vodafone, Google, Verizon και Texas Instruments.

Το Android διαθέτει μια ευρεία γκάμα εφαρμογών για το περιβάλλον του. Επιπλέον το Android παρέχει ειδική προγραμματιστική βιβλιοθήκη (Application Programming Interface - API) για τους προγραμματιστές που επιθυμούν να αναπτύξουν λογισμικό για την πλατφόρμα Android.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



Εικόνα 9: Τα τέσσερα διαφορετικά επίπεδα της αρχιτεκτονικής του Android

Καθώς γράφονται οι γραμμές αυτές η προγραμματιστική βιβλιοθήκη του Android έχει φτάσει την έκδοση 18, η οποία αντιστοιχεί στην έκδοση 4.3 της πλατφόρμας.

3.1 Αρχιτεκτονική του Android

Στο Android διακρίνονται τέσσερα διαφορετικά επίπεδα λειτουργικότητας. Ουσιαστικά, κάθε επιπρόσθετο επίπεδο δημιουργεί ένα υψηλότερο επίπεδο αφαίρεσης του υποκείμενου υλικού και χρησιμοποιεί τις υπηρεσίες που παρέχονται από το κατώτερο επίπεδο.

Τα τέσσερα αυτά επίπεδα είναι τα παρακάτω:

- **Πυρήνας Linux (Linux Kernel)**: Το Android είναι βασισμένο στον πυρήνα Linux, ο οποίος παρέχει το αφαιρετικό επίπεδο υλικού και υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν λειτουργίες δικτύου, διαχείριση μνήμης, διαχείριση διεργασιών, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί, όπως και σε κάθε σύστημα, είναι υπεύθυνοι για την επικοινωνία του λογισμικού (software) με το υποκείμενο υλικό (hardware) της συσκευής. Υπάρχουν drivers για το σύστημα WiFi, Bluetooth, για το σύστημα ήχου, της κάμερας, της υποδοχής USB, του πληκτρολογίου, του συστήματος γραφικών, της διαχείρισης διεργασιών, της μνήμης, του συστήματος εξοικονόμησης ενέργειας κ.ο.κ.

  Πάνω από τον πυρήνα Linux δημιουργείται μια διεπαφή, το Επίπεδο Αφαίρεσης Υλικού (Hardware Abstraction Layer) μέσω του οποίου επικοινωνεί το δεύτερο επίπεδο της αρχιτεκτονικής του Android.

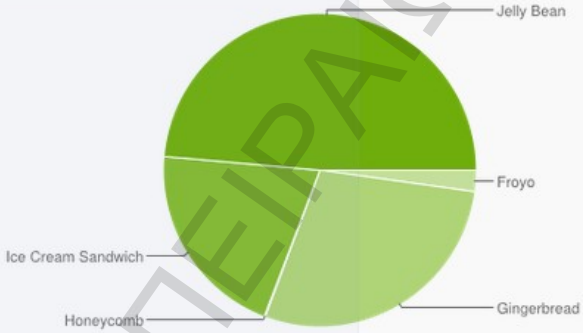
- **Βιβλιοθήκες Android (Libraries) και Περιβάλλον Εκτέλεσης (Runtime Environment)**: Το Android περιλαμβάνει ένα σύνολο βιβλιοθηκών (Εγγενείς Βιβλιοθήκες - Προηγμένες Βιβλιοθήκες Android), οι οποίες χρησιμοποιούνται από διάφορα συστατικά μέρη του συστήματος. Το σύνολο σχεδόν των βιβλιοθηκών είναι γραμμένο σε C και C++, οι οποίες έχουν μεταγλωττιστεί για τη χρήση τους από το λειτουργικό. Οι δυνατότητες αυτών των βιβλιοθηκών γίνονται διαθέσιμες στους προγραμματιστές μέσω του πλαισίου εφαρμογών του Android. Οι

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών, και άρα είναι αναπόσπαστο κομμάτι τους. Μερικές από τις βιβλιοθήκες του Android είναι:

- System C library: Είναι μια ενσωμάτωση της τυπικής βιβλιοθήκης συστήματος της C, για φορητές συσκευές βασισμένες στο Linux.

Version	Codename	API	Distribution
2.2	Froyo	8	2.2%
2.3.3 - 2.3.7	Gingerbread	10	28.5%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	20.6%
4.1.x	Jelly Bean	16	36.5%
4.2.x		17	10.6%
4.3		18	1.5%



Εικόνα 10: Οι τελευταίες εκδόσεις του Android και το μερίδιο της αγοράς που καταλαμβάνουν αυτές. Πηγή: http://developer.android.com

- Βιβλιοθήκες Πολυμέσων: Αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας,όπως: MPEG4, H.264, MP3, AAC, AMR, JPG και PNG.
- Surface Manager: Διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα διδιάστατα και τριδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.
- LibWebCore: Μία μοντέρνα μηχανή υποστήριξης πλοήγησης στο διαδίκτυο (browser engine) την οποία χρησιμοποιεί ο ενσωματωμένος browser του Android αλλά και οι WebViews, που ενσωματώνονται στις εφαρμογές.
- SGL: Μηχανή διδιάστατων γραφικών.
- Βιβλιοθήκες 3D: Μία υλοποίηση βασισμένη στα APIs του OpenGL ES 1. Οι βιβλιοθήκες χρησιμοποιούν είτε τρισδιάστατη επιτάχυνση υλικού, όπου αυτή είναι διαθέσιμη, είτε μια υψηλά βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση που η πρώτη δεν είναι διαθέσιμη.
- SQLite: Μια πολύ ελαφριά από άποψη υπολογιστικής ισχύος μηχανή σχεσιακών βάσεων δεδομένων, όπου κάθε βάση δεδομένων αποθηκεύεται σε ξεχωριστό αρχείο. Πρόκειται για single-threaded μηχανή, δηλαδή μόνο μια διεργασία ή νήμα (thread) μπορεί να πραγματοποιεί αλλαγές στη βάση δεδομένων ανά πάσα στιγμή. Για την τοπική εμβέλεια μιας φορητής μικροσυσκευής αυτό το μοντέλο είναι παραπάνω από αρκετό.
- FreeType: Παρέχει ευκρίνεια γραφικών στα bitmaps και τις γραμματοσειρές των εφαρμογών του συστήματος.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Το περιβάλλον εκτέλεσης (Android Runtime Environment) βρίσκεται στο ίδιο επίπεδο με τις εγγενείς βιβλιοθήκες. Εδώ, εκτός των βασικών βιβλιοθηκών της Java, βρίσκεται και η επονομαζόμενη Εικονική Μηχανή Dalvik (Dalvik Virtual Machine). Η Dalvik τρέχει .dex αρχεία, τα οποία είναι bytecodes που προέρχονται από αρχεία .class και .jar. Σε αντίθεση με τα class αρχεία, τα .dex είναι πολύ πιο συμπαγή και αποδοτικά, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη και μπαταρία. Το Android περιλαμβάνει ένα σύνολο βασικών βιβλιοθηκών που παρέχουν τις περισσότερες από τις διαθέσιμες λειτουργίες των βασικών βιβλιοθηκών της Java.

- **Πλαίσιο Εργασίας Εφαρμογών (Application Framework)**: Πάνω από τις βιβλιοθήκες και το περιβάλλον εκτέλεσης Android, είναι το πλαίσιο εργασίας εφαρμογών. Αυτό το επίπεδο παρέχει υψηλού επιπέδου δομικές μονάδες οι οποίες μπορούν να χρησιμοποιηθούν για την κατασκευή των εφαρμογών. Αυτό το πλαίσιο είναι προ-εγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια. Το Android όντας μία ανοιχτή πλατφόρμα ανάπτυξης, προσφέρει τη δυνατότητα δημιουργίας εξαιρετικά πλούσιων και καινοτόμων εφαρμογών. Η αρχιτεκτονική των εφαρμογών είναι έτσι σχεδιασμένη ώστε να απλοποιεί την επαναχρησιμοποίηση των διάφορων συστατικών μερών. Οποιαδήποτε εφαρμογή μπορεί να θέσει διαθέσιμες προς χρήση τις δυνατότητές της και οποιαδήποτε άλλη εφαρμογή που μπορεί να κάνει χρήση αυτών των δυνατοτήτων. Τα σημαντικότερα δομικά στοιχεία του πλαισίου αυτού είναι:
    - Activity Manager - Διαχειριστής δραστηριοτήτων: Υπεύθυνος για τον έλεγχο του χρόνου ζωής των εφαρμογών και για την διατήρηση μιας στοίβας που επιτρέπει την πλοήγηση του χρήστη σε προηγούμενες οθόνες.
    - Content Providers - Παροχείς Περιεχομένου: Επιτρέπουν στις εφαρμογές να έχουν πρόσβαση στα δεδομένα άλλων εφαρμογών ή να κοινοποιήσουν στις άλλες εφαρμογές τα δικά τους δεδομένα.
    - Resource Manager - Διαχειριστής Πόρων: Παρέχει πρόσβαση σε πόρους. Οι πόροι είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι κωδικοί χρωμάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
    - Location Manager - Διαχειριστής Τοποθεσίας: Χρησιμοποιείται για να μπορεί να ξέρει το τηλέφωνο που βρίσκεται ανά πάσα στιγμή.
    - Notification Manager - Διαχειριστής Ειδοποιήσεων: Επιτρέπει στις εφαρμογές να εμφανίζουν ειδοποιήσεις στη μπάρα κατάστασης, χωρίς τη διακοπή των υπόλοιπων διεργασιών.
- **Εφαρμογές (Applications)**: Το επίπεδο των εφαρμογών είναι το επίπεδο στο οποίο εγκαθίστανται όλες οι εφαρμογές, όπως ο περιηγητής για το διαδίκτυο (browser) , παιχνίδια, διαχειριστής επαφών κ.α.

  Σε αυτό το επίπεδο στοχεύουν οι προγραμματιστές, όπου, κάνοντας χρήση των στοιχείων του Application Framework, δημιουργούν τις δικές τους εφαρμογές.

Το Android παρέχει ειδική προγραμματιστική βιβλιοθήκη για ανάπτυξη εφαρμογών με τη χρήση της γλώσσας προγραμματισμού Java και ειδικό πακέτο ανάπτυξης (SDK) που διευκολύνει την ανάπτυξη και την διανομή των εφαρμογών. Επίσης, παρέχεται μηχανισμός δοκιμής και αποσφαλμάτωσης με τη χρήση εικονικών συσκευών Android.

Στο SDK παρέχεται επίσης τεκμηρίωση και ειδική διεπαφή για τη διαχείριση των εκδόσεων API.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

3.2 Προδιαγραφές Λειτουργίας της Τελικής Εφαρμογής

Η τελική εφαρμογή επίδειξης στο Android στοχεύει σε εκδόσεις του API 14 και άνω. Η λειτουργία της είναι πολύ απλή. Ο χρήστης, καθώς εισέρχεται στην εφαρμογή επιλέγει το πλήκτρο "Έναρξη Ηχογράφησης" (βλ. Εικόνα 11) προκειμένου να ηχογραφήσει το ρυθμό που επιθυμεί, χρησιμοποιώντας το μικρόφωνο της συσκευής.

Εικόνα 11: Σχεδιάγραμμα mockup της γραφικής διεπαφής της τελικής εφαρμογής

Όταν ο χρήστης ολοκληρώσει την ηχογράφηση, επιλέγει το πλήκτρο "Διακοπή Ηχογράφησης", προκειμένου να σταματήσει η ηχογράφηση από την εφαρμογή και να ξεκινήσουν οι διαδικασίες της κατάτμησης, της εξαγωγής χαρακτηριστικών και της ομαδοποίησης. Το τελικό αποτέλεσμα της κατάτμησης θα εμφανίζεται σε ειδικά κατασκευασμένο διάγραμμα της εφαρμογής, όπου θα φαίνεται η περιβάλλουσα ενέργειας του ηχογραφημένου δείγματος, καθώς και τα τμήματα που προέκυψαν από την διαδικασία κατάτμησης.

Αφού ολοκληρωθεί η κατάτμηση, θα γίνεται ο υπολογισμός του Zero Crossing Rate και η ομαδοποίηση βάσει των τιμών ενέργειας και ZCR. Τα αποτελέσματα των υπολογισμών αυτών δεν εμφανίζονται στον τελικό χρήστη, τα αποτελέσματα, όμως, της ομαδοποίησης διατηρούνται στη μνήμη, προκειμένου να χρησιμοποιηθούν κατά την αναπαραγωγή του τελικού ήχου, την οποία ο χρήστης ενεργοποιεί επιλέγοντας το πλήκτρο "Αναπαραγωγή". Κατά την αναπαραγωγή του τελικού ήχου, με τη χρήση, πλέον, έτοιμων ηχητικών δειγμάτων, ο χρήστης πληροφορείται για το αποτέλεσμα της

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

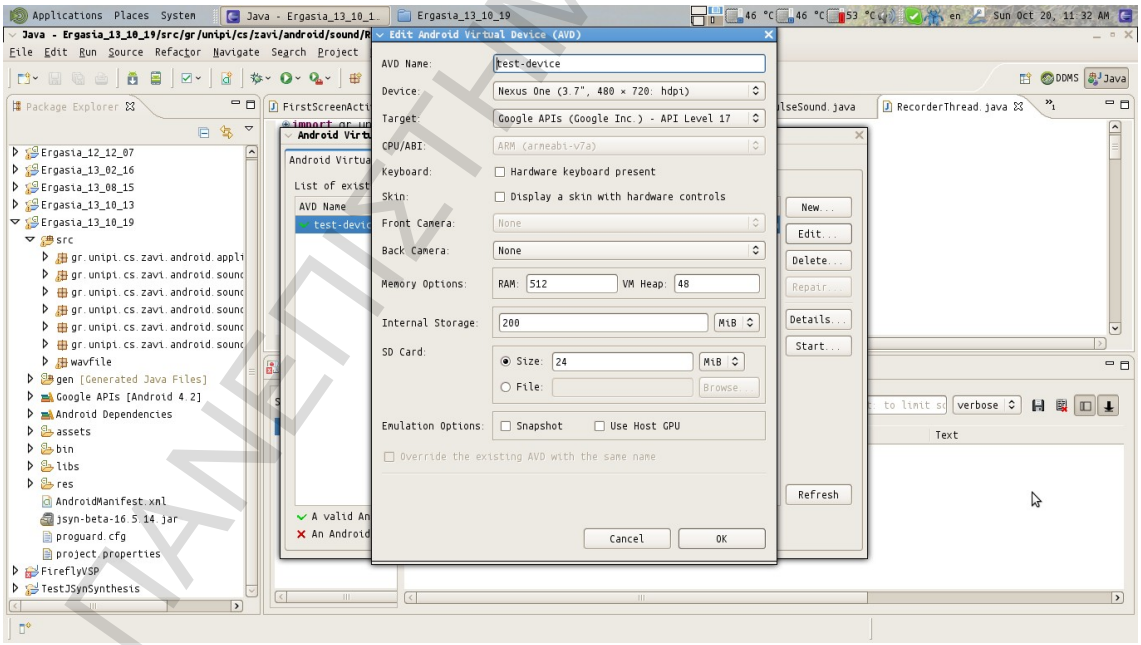
ομαδοποίησης μέσω του ηχητικού αποτελέσματος (η ομαδοποίηση γίνεται σε δύο ομάδες, θεωρώντας ότι στις περισσότερες περιπτώσεις ο χρήστης θα μπορεί να χρησιμοποιήσει τα δύο χέρια του για την παραγωγή του ήχου στην ηχογράφηση, συνεπώς οι δύο αυτές διαφορετικές πηγές ήχου θα πρέπει να αποδοθούν σε δύο διαφορετικές ομάδες και κατά την ομαδοποίηση).

3.3 Περιγραφή Υλοποίησης

Στη γλώσσα προγραμματισμού Java, συνιστάται η οργάνωση των καινούργιων κλάσεων που δημιουργούμε σε πακέτα (packages) προκειμένου να αποφευχθεί η πιθανότητα σύγκρουσης δύο ίδιων ονομάτων κλάσεων. Για παράδειγμα, εάν σε κάποιο project ορίσουμε μια κλάση με το όνομα StringBuffer, η Java δεν θα μπορούσε να ξεχωρίσει, κάθε φορά που θα κάναμε χρήση αυτής της κλάσης, αν θα αναφερόμασταν στην κλάση που έχουμε ορίσει ή στην κλάση με το ίδιο όνομα που διαθέτει το API της Java.

Για αυτόν τον λόγο, χρησιμοποιούνται ευρύτατα τα πακέτα, όπου ουσιαστικά πρόκειται για φακέλους στους οποίους αποθηκεύονται οι κλάσεις μας. Όταν έχουμε δημιουργήσει ένα πακέτο, το οποίο θα περιέχει τις κλάσεις που θα έχουμε συγγράψει, μπορούμε στη συνέχεια να χρησιμοποιήσουμε αυτές τις κλάσεις σε κάποιο άλλο πακέτο, εάν εισάγουμε (import) το πακέτο στον κώδικα του καινούργιου πακέτου.

Η χρήση των πακέτων μπορεί να διευκολύνει πολύ τη διαδικασία ανάπτυξης εφαρμογών, καθώς η τάση που φαίνεται να επικρατεί στον προγραμματισμό τα τελευταία χρόνια, είτε αυτός αναφέρεται σε εφαρμογές οποιουδήποτε ενδιαφέροντος, είτε σε εφαρμογές δικτύου, είναι η τάση του διαχωρισμού του περιεχομένου της εφαρμογής από τον τρόπο παρουσίασής της. Αυτό σημαίνει ότι οποιοδήποτε κομμάτι του κώδικα αφορά το ίδιο το περιεχόμενο (για παράδειγμα, σε μια εφαρμογή εμπορικής διαχείρισης, η διαχείριση της βάσης δεδομένων των πελατών), θα πρέπει να είναι σαφώς διαχωρισμένο από το κομμάτι



Εικόνα 12: Άποψη του Android Virtual Device Manager, χρησιμοποιούμενος μέσω του περιβάλλοντος ανάπτυξης λογισμικού (IDE) Eclipse

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

του κώδικα που αφορά τον τρόπο παρουσίασής του, δηλαδή την Γραφική Διεπαφή Χρήστη (Graphical User Interface - GUI). Με αυτόν τον τρόπο, σε μια μελλοντική έκδοση του προγράμματος, που θα υιοθετούσε νέες τεχνολογίες στον τομέα του GUI, αλλά δεν θα είχε τίποτε διαφορετικό να παρουσιάσει όσον αφορά την διαχείριση των δεδομένων, οι όποιες μεταβολές του κώδικα θα γίνονταν μόνο στο κομμάτι του κώδικα που αφορά το GUI.

Η τελική υλοποίηση ακολουθεί το προαναφερθέν πρότυπο οργάνωσης σε πακέτα. Τα πακέτα της τελικής υλοποίησης είναι τα παρακάτω:

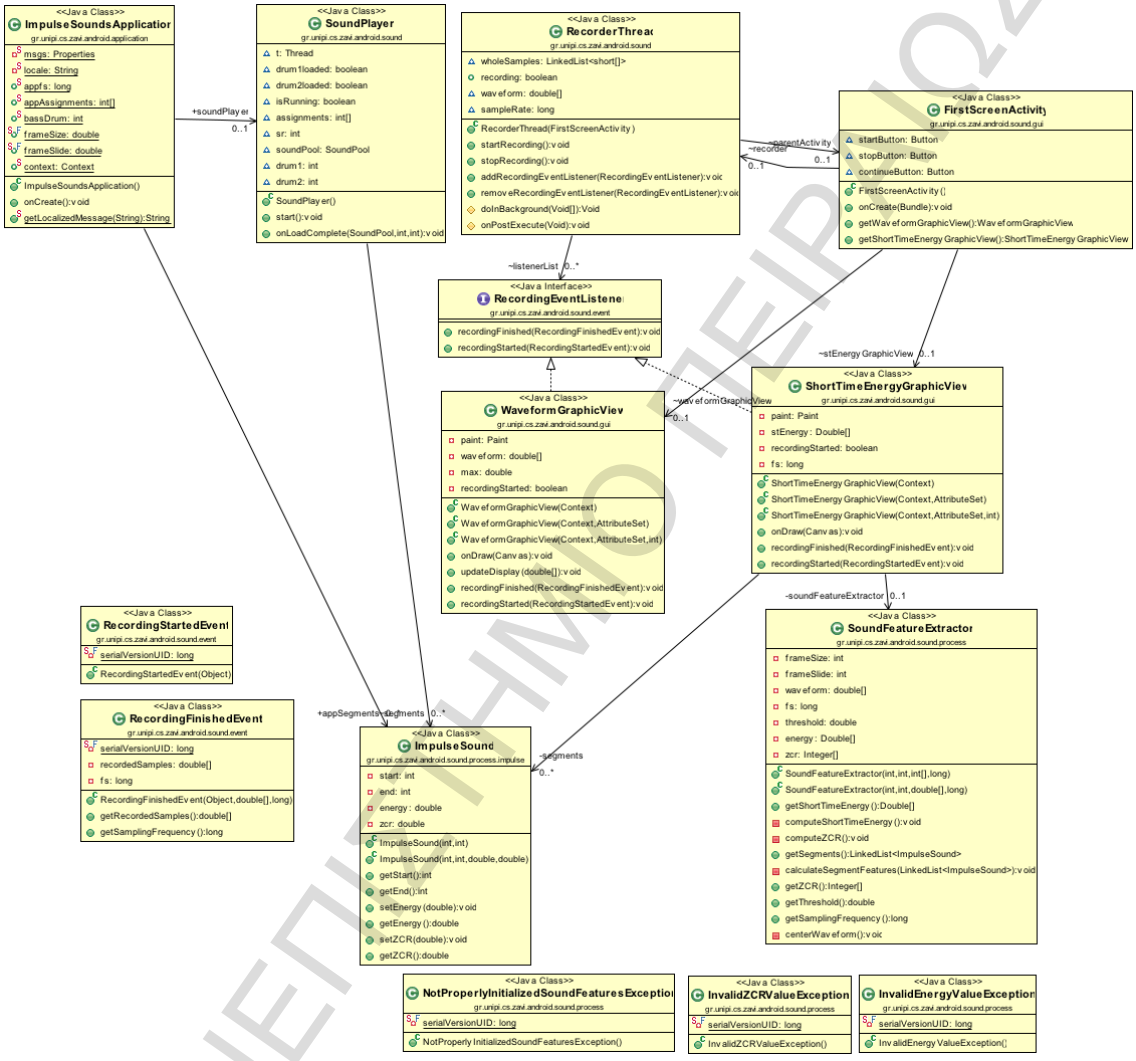
gr.unipi.cs.zavi.android.application	Περιέχει μια κλάση που επεκτείνει την κλάση Application του πακέτου android.app (παρέχεται από το API του Android). Αποτελεί το περιβάλλον της Android εφαρμογής.
gr.unipi.cs.zavi.android.sound	Περιέχει κλάσεις που αφορούν την είσοδο/έξοδο I/O στο επίπεδο του ήχου.
gr.unipi.cs.zavi.android.sound.event	Πρόκειται για εξειδικευμένη υλοποίηση του μοντέλου event – event listener. Χρησιμοποιείται προκειμένου να ειδοποιηθούν τα γραφικά στοιχεία της εφαρμογής (πακέτο gr.unipi.cs.zavi.android.sound.gui) όταν η ηχογράφηση ξεκινάει ή ολοκληρώνεται. Περιέχεται η διεπαφή (interface) RecordingEvent Listener[8].
gr.unipi.cs.zavi.android.sound.gui	Περιέχει κλάσεις που αφορούν την γραφική διεπαφή (GUI) της εφαρμογής. Πέραν της κλάσης που επεκτείνει την κλάση Activity του πακέτου android.app, περιέχει δύο κλάσεις που επεκτείνουν την κλάση View του πακέτου android.view και υλοποιούν της γραφικές αναπαραστάσεις της κυματομορφής και της ενέργειας βραχέος χρόνου.
gr.unipi.cs.zavi.android.sound.process	Περιέχει την θεμελιώδη για τη λειτουργικότητα κλάση SoundFeatureExtractor, η οποία αναλαμβάνει την υλοποίηση της κατάτμησης του ηχογραφημένου ηχητικού δείγματος, αλλά και τον υπολογισμό των χαρακτηριστικών. Επίσης, περιέχονται κλάσεις τύπου Exception, που αντιπροσωπεύουν πιθανές περιπτώσεις σφάλματος κατά την επεξεργασία του ήχου.
gr.unipi.cs.zavi.android.sound.process.impulse	Περιέχει την κλάση ImpulseSound που αντιπροσωπεύει ένα κρουστικό τμήμα του ήχου. Χρησιμοποιείται βοηθητικά από το πακέτο gr.unipi.cs.zavi.android.sound.process.

[8] Ένα interface είναι μια ειδική κλάση που περιέχει δηλώσεις μεθόδων, χωρίς σώμα κώδικα. Ένα interface ορίζει μια δυνατότητα που μπορεί να έχει μια άλλη κλάση και μπορεί να υλοποιηθεί (implement) από αυτήν την κλάση, προκειμένου αυτή να ορίσει τον τρόπο με τον οποίο υλοποιεί αυτή τη δυνατότητα. Τα interfaces χρησιμοποιούνται πολύ συχνά στη Java σε event handling routines (ρουτίνες χειρισμού γεγονότων). Σε αυτές, οι κλάσεις που θέλουν να είναι event handlers (δηλαδή που θέλουν να αποκρίνονται σε ένα συμβάν – γεγονός) πρέπει να υλοποιήσουν τις μεθόδους του αντίστοιχου interface.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τα παραπάνω πακέτα καλύπτουν την γραφική διασύνδεση του χρήστη με την εφαρμογή, την εξαγωγή των χαρακτηριστικών και την επεξεργασία του ήχου, των χειρισμό των νημάτων (threads), την ηχογράφηση και την αναπαραγωγή του ήχου.

Η ομαδοποίηση καλύφθηκε από εξωτερική βιβλιοθήκη, την βιβλιοθήκη Weka



Εικόνα 13: Διάγραμμα τάξεων (class diagram) της τελικής υλοποίησης

(http://www.cs.waikato.ac.nz/ml/weka/). Η βιβλοθήκη Weka είναι μια συλλογή αλγόριθμων machine learning και εξόρυξης δεδομένων (data mining).

Εξαιτίας της έλλειψης συσκευής με λειτουργικό σύστημα Android, η εφαρμογή ελέγχθηκε σε μία εικονική συσκευή που δημιουργήθηκε από το ειδικό εργαλείο Android Virtual Device Manager (βλ. Εικόνα 12).

Στην Εικόνα 13 παρατίθεται το ολοκληρωμένο διάγραμμα κλάσεων της εφαρμογής. Η κλάση ImpulseSoundsApplication αντιπροσωπεύει την εφαρμογή και περιέχει μεταβλητές και σταθερές

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

καθολικής εμβέλειας. Για παράδειγμα, περιέχει τις τιμές frameSize και frameSlide, το μέγεθος των πλαισίων σε sec και την ολίσθηση των πλαισίων. Επίσης, περιέχεται η μέθοδος getLocalizedMessage με την οποία πραγματοποιείται ανάκτηση των μηνυμάτων που περιέχονται στη γραφική διεπαφή της εφμαρμογής, βάσει της επιλογής γλώσσας του χρήστη στο σύστημα.

Το κατάλληλο μήνυμα ανακτάται από τα resources της εφαρμογής (αρχείο uimessages.props από τον κατάλογο assets/ της εφαρμογής).

Κεφαλαιώδους σημασίας είναι η κλάση FirstScreenActivity που επεκτείνει την κλάση Activity του Android API και αποτελεί το σημείο εισόδου της εφαρμογής. Βασικός ρόλος της κλάσης αυτής είναι ο σχεδιασμός της γραφικής διεπαφής. Τα στοιχεία (components) της γραφικής διεπαφής δηλώνονται στο δομημένο XML αρχείο res/layout/main.xml. Η κλάση αυτή ανήκει στο πακέτο gr.unipi.cs.zavi.android.sound.gui, το οποίο, όπως έχει ήδη αναφερθεί περιέχει όλες τις κλάσεις που αφορούν τη γραφική διεπαφή.

Στο ίδιο πακέτο περιέχονται οι κλάσεις WaveformGraphicView και ShortTimeEnergyGraphicView. Και οι δύο αυτές κλάσεις επεκτείνουν την κλάση View του πακέτου android.view. Κάθε κλάση που επεκτείνει την κλάση View αποτελεί ένα γραφικό στοιχείο που δύναται να εμφανιστεί στη γραφική διεπαφή. Σε συνέχεια αυτής της υλοποίησης, οι δύο παραπάνω κλάσεις αναφέρονται στο αρχείο res/layout/main.xml που -όπως ήδη αναφέρθηκε- περιέχει το γραφικό layout της εφαρμογής. Κάθε κλάση που επεκτείνει την κλάση View του Android API, δύναται να κάνει overide τη μέθοδο onDraw(Canvas) ώστε να εξειδικεύσει τον τρόπο με τον οποίο εμφανίζεται το εν λόγω στοιχείο. Εν προκειμένω, η κλάση WaveformGraphicView χρησιμοποιεί τη μέθοδο onDraw προκειμένου να εμφανιστεί με κατάλληλο τρόπο η κυματομορφή του ηχογραφημένου ήχου. Ομοίως, η κλάση ShortTimeEnergyGraphicView κάνει overide τη μέθοδο onDraw προκειμένου να εμφανιστεί με κατάλληλο και φιλικό προς το χρήστη τρόπο η περιβάλλουσα ενέργειας του ήχου, καθώς και τα εντοπισμένα τμήματα του ήχου (βλ. Εικόνα 16).

Οι κλάσεις WaveformGraphicView και ShortTimeEnergyGraphicView υλοποιούν τη διεπαφή (interface) RecordingEventListener (που βρίσκεται στο πακέτο gr.unipi.cs.zavi.android.sound.event) και αποτελεί το interface που πρέπει να υλοποιήσει κάθε κλάση που θέλει να αποκρίνεται σε συμβάντα ηχογράφησης. Τα συμβάντα αυτά είναι η έναρξη και η λήξη της ηχογράφησης και αντιπροσωπεύονται από τις κλάσεις RecordingStartedEvent και RecordingFinishedEvent του ίδιου πακέτου. Οι κλάσεις WaveformGraphicView και ShortTimeEnergyGraphicView πρέπει να ενημερώνονται για τέτοιου είδους συμβάντα της εφαρμογής προκειμένου να προσαρμόζουν την εμφάνισή τους αναλόγως. Πιο συγκεκριμένα, όταν αρχίζει η ηχογράφηση, τα δύο αυτά γραφικά στοιχεία εμφανίζουν μήνυμα αναμονής, ενώ, όταν η ηχογράφηση ολοκληρωθεί το στοιχείο της κλάσης WaveformGraphicView εμφανίζει την κυματομορφή ενώ το στοιχείο της κλάσης ShortTimeEnergyGraphicView πραγματοποιεί τις παρακάτω ενέργειες:

- Επικοινωνεί με ένα instance της κλάσης SoundFeatureExtractor, προκειμένου να υπολογισθούν οι τιμές της ενέργειας βραχέος χρόνου και zero crossing rate.
- Εμφανίζει την περιβάλλουσα ενέργειας σύμφωνα με τις υπολογισθείσες τιμές.
- Επικοινωνεί με το instance της κλάσης SoundFeatureExtractor προκειμένου να εντοπισθούν τα κατάλληλα τμήματα.
- Εμφανίζει τα τμήματα αυτά στην οθόνη.
- Επικοινωνεί με instances κλάσεων του Weka API προκειμένου να πραγματοποιηθεί η ομαδοποίηση.

Η κλάση SoundFeatureExtractor του πακέτου gr.unipi.cs.zavi.android.sound.process, όπως ήδη αναφέρθηκε περιέχει μεθόδους επεξεργασίας σήματος για τον υπολογισμό της ενέργειας βραχέος χρόνου και το ρυθμού διέλευσης από το μηδέν. Η κλάση αυτή περιέχει και βοηθητική private μέθοδο για την αφαίρεση του DC offset. Επιπλέον, αυτή η κλάση αναλαμβάνει και την κατάτμηση του ήχου σε ηχητικά

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

συμβάντα. Συνεπώς, η συγκεκριμένη κλάση αποτελεί τον πυρήνα της επεξεργασίας του σήματος ήχου από την εφαρμογή.

Στο πακέτο gr.unipi.cs.zavi.android.sound.process περιέχει τρεις κλάσεις που αντιπροσωπεύουν πιθανές δυσλειτουργίες (Exceptions) κατά την εκτέλεση της εφαρμογής:

- InvalidEnergyValueException – Ακατάλληλη τιμή ενέργειας
- InvalidZCRValueException – Ακατάλληλη τιμή ρυθμού διέλευσης από το μηδέν
- NotProperlyInitializedSoundFeaturesException – Αδυναμία υπολογισμού τμημάτων εξαιτίας απουσίας κάποιων εκ των διανυσμάτων των τιμών ενέργειας ή ZCR.

Τέλος, όπως αναφέρθηκε το πακέτο gr.unipi.cs.zavi.android.sound περιέχει κλάσεις που αφορούν την είσοδο/έξοδο του ήχου. Πιο συγκεκριμένα, η κλάση RecorderThread αναλαμβάνει την ηχογράφηση του ήχου με χρήση του συστήματος εισόδου της συσκευής. Η κλάση RecorderThread επεκτείνει την κλάση AsyncTask του πακέτου android.os του Android API. Η κλάση αυτή αντιπροσωπεύει ένα ξεχωριστό νήμα (thread) του περιβάλλοντος εκτέλεσης του Android.

Η κλάση RecorderThread κάνει overide τις μεθόδους doInBackground (κύριο μέρος της εκτέλεσης του νήματος) και onPostExecute (όπου ειδοποιούνται όλους οι RecordingEventListeners με το συμβάν RecordingFinishedEvent).

Στο πακέτο gr.unipi.cs.zavi.android.sound περιέχεται επίσης η κλάση SoundPlayer η οποία αναλαμβάνει την φόρτωση των δειγμάτων ήχου (beat01.wav και beat02.wav) και την αναπαραγωγή της τελικής σύνθεσης.

Η κλάση SoundPlayer εκκινεί ένα νέο νήμα με υψηλή προτεραιότητα στο περιβάλλον εκτέλεσης του Android, με το οποίο γίνεται η φόρτωση των ηχητικών δειγμάτων των κρουστικών ήχων και στη συνέχεια γίνεται η αναπαραγωγή του τελικού ήχου. Για την αναπαραγωγή του ήχου γίνεται χρήση της κλάσης SoundPool του πακέτου android.media εντός ενός synchronized block.

Στο παραδοτέο της εργασίας περιλαμβάνεται ολόκληρος ο πηγαίος κώδικας της τελικής εφαρμογής.

3.4 Προβλήματα κατά την Υλοποίηση

Κατά τη φάση της υλοποίησης της τελικής εφαρμογής, σημειώθηκαν δύο βασικά προβλήματα που εμπόδισαν την υλοποίηση σύμφωνα με τον αρχικό σχεδιασμό. Τα προβλήματα αυτά μπορούν να συνοψισθούν ως εξής:

- **Αδυναμία λήψης ηχογραφημένου υλικού μέσω της εικονικής συσκευής του Android**. Λόγω έλλειψης πραγματικής συσκευής με λειτουργικό σύστημα Android, ήταν αναγκαστική η χρήση εικονικής συσκευής του Android Virtual Device Manager. Ωστόσο, η παραπάνω επιλογή οδήγησε σε αδυναμία ελέγχου του thread ηχογράφησης που είχε υλοποιηθεί, καθώς ήταν αδύνατον να χρησιμοποιηθεί η είσοδος της κάρτας ήχου του PC όπου έγινε η εγκατάσταση του virtual device και του SDK ως είσοδος ήχου της εικονικής συσκευής.

  Προκειμένου να αντιμετωπιστεί το παραπάνω ζήτημα, το κομμάτι κώδικα που πραγματοποιούσε την δειγματοληψία από τη συσκευή ήχου της συσκευής Android αντικαταστάθηκε από κώδικα που πραγματοποιεί ανάγνωση αποθηκευμένων ηχητικών δειγμάτων. Ο κώδικας αυτός κάνει χρήση έτοιμης βιβλιοθήκης για την ανάγνωση αρχείων ήχου WAV. Η βιβλιοθήκη αυτή βρέθηκε από τον ιστότοπο http://www.labbookpages.co.uk.

- **Αδυναμία χρήσης της βιβλιοθήκης σύνθεσης ήχου jSyn**. Όπως αρχικά σχεδιάστηκε, κατά την αναπαραγωγή του τελικού ήχου, επρόκειτο να γίνει χρήση της πολύ δημοφιλούς βιβλιοθήκης σύνθεσης ήχου για Java ονόματι jSyn. Τα τελευταία χρόνια, η βιβλιοθήκη jSyn έχει κάνει βήματα προς την πλήρη μεταφερτότητα (υπήρχαν κομμάτια του πυρήνα του jSyn γραμμένα σε

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

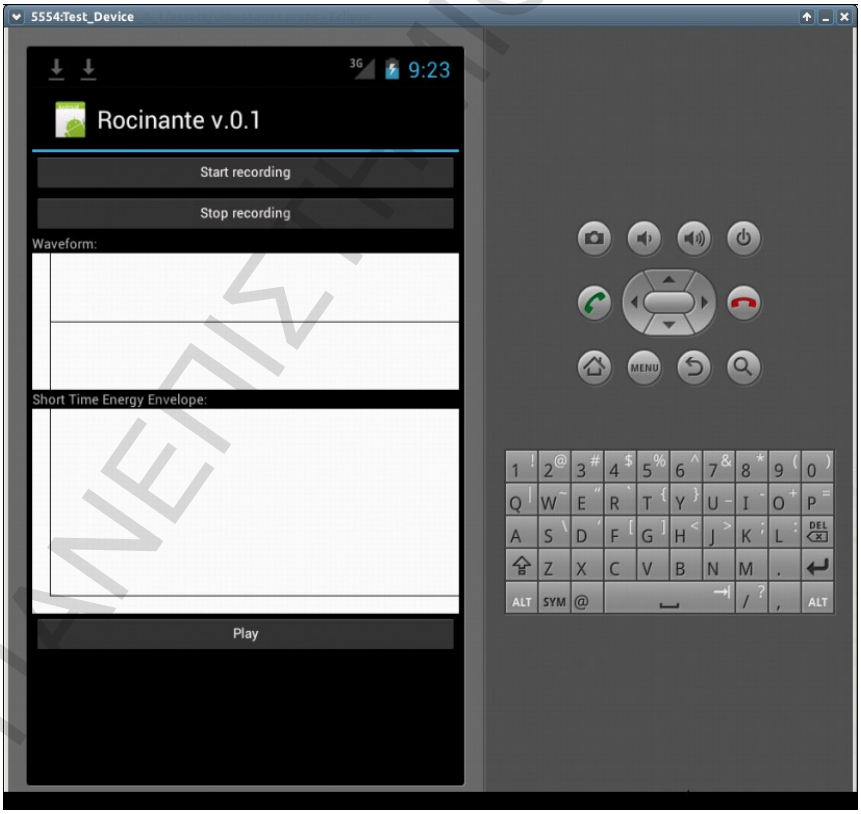
γλώσσα C). Ωστόσο, η βιβλιοθήκη jSyn βασίζεται στο πακέτο javax.sound του API της Java, το οποίο δεν υποστηρίζεται από το API του Android. Συνεπώς, η υλοποίηση στράφηκε προς τη χρήση κλάσεων του πακέτου android.media για την επίτευξη της αναπαραγωγής.

Προς πληροφόρηση παρατίθεται το απόσπασμα κώδικα που επρόκειτο να χρησιμοποιηθεί κατά την ηχογράφηση:

```
AudioRecord recorder;
short audioData[];
int bufferSize;
ArrayList<Short> samples = new ArrayList<Short>();
bufferSize = AudioRecord.getMinBufferSize(8000,
            AudioFormat.CHANNEL_CONFIGURATION_MONO,
            AudioFormat.ENCODING_PCM_16BIT);
System.out.println("" + bufferSize);
recorder = new AudioRecord(AudioSource.MIC, 8000,
            AudioFormat.CHANNEL_CONFIGURATION_MONO,
            AudioFormat.ENCODING_PCM_16BIT, bufferSize);
recording = true;
audioData = new short[bufferSize];

// loop forever
while (recording) {
      if (recorder.getState() == AudioRecord.STATE_INITIALIZED) {
            if (recorder.getRecordingState() ==
                        AudioRecord.RECORDSTATE_STOPPED) {
                  recorder.startRecording();
            } else {
                  recorder.read(audioData, 0, bufferSize);
                  for (int j = 0; j < bufferSize; j++) {
                        samples.add(audioData[j]);
                  }
                  System.out.println("Read " + bufferSize
                              + " samples from mic");
            }
      }
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

```
}
if (recorder.getState() == AudioRecord.RECORDSTATE_RECORDING) {
      recorder.stop();
}
recorder.release();
recorder = null;
System.out.println("Resources released!");

System.out.println(samples.size() + " samples read.");

waveform = new short[samples.size()];
for (int i = 0; i < waveform.length; i++) {
      waveform[i] = samples.get(i);
}
```



Εικόνα 14: Η αρχική οθόνη της εφαρμογής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Το παραπάνω απόσπασμα κώδικα προορίζεται για τη μέθοδο doInBackground της κλάσης RecorderThread.

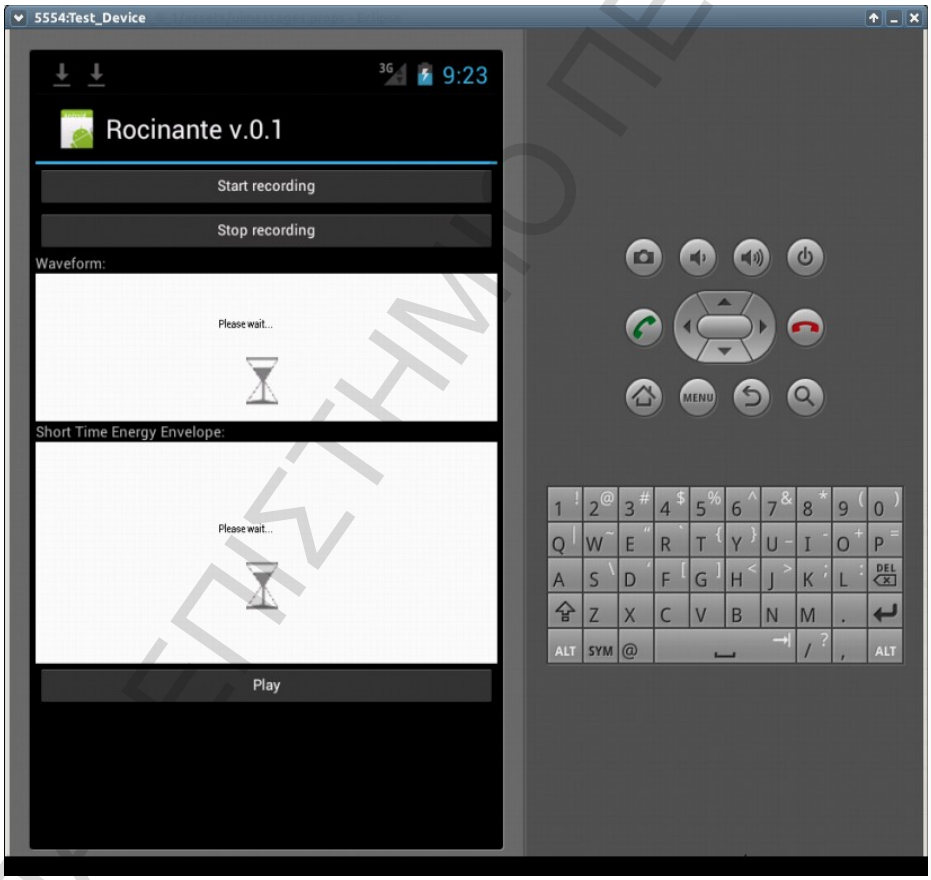
3.5 Παρουσίαση της Τελικής Εφαρμογής

Όπως έχει ήδη αναφερθεί, η τελική εφαρμογή περιέχει μια οθόνη (βλ. Εικόνα 11). Η αρχική κατάσταση της οθόνης κατά την εκτέλεση της εφαρμογής στην εικονική συσκευή φαίνεται στην Εικόνα 14.

Η εφαρμογή έχει πάρει το όνομα “Rocinante” προς τιμήν του αγέρωχου και περήφανου αλόγου του Δον Κιχώτη.

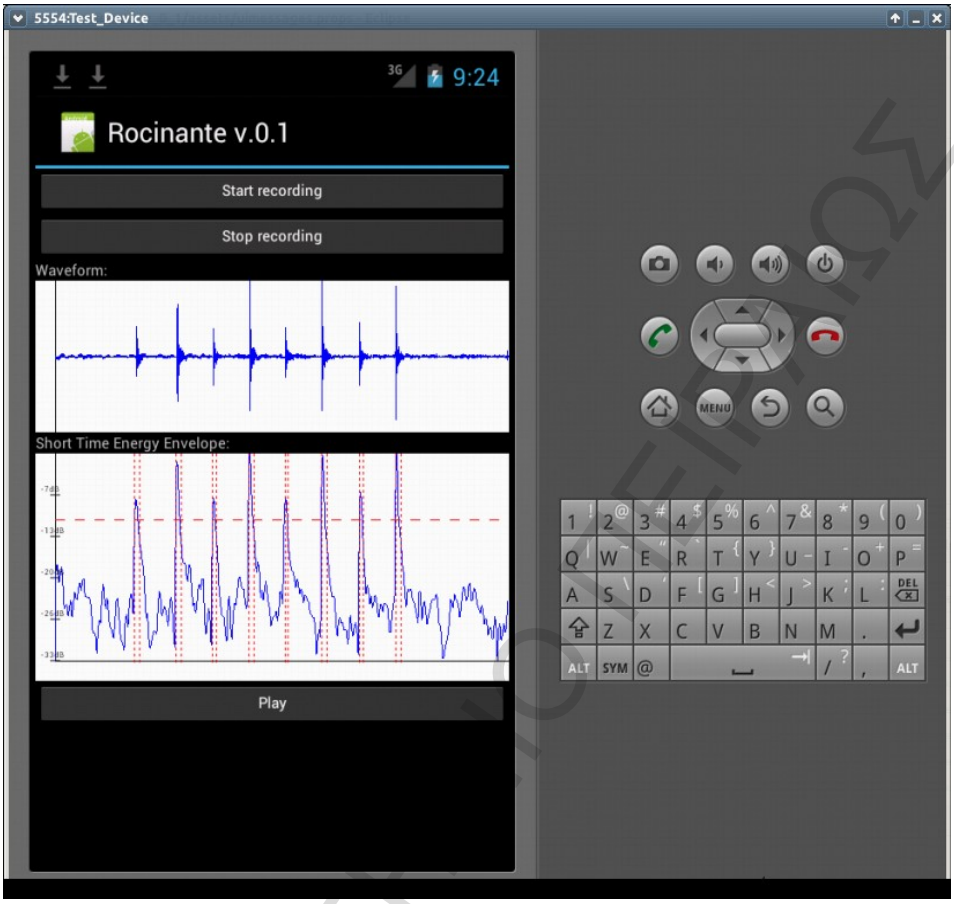
Ο χρήστης πρέπει να επιλέξει το πλήκτρο “Start recording” προκειμένου να αρχίσει η ηχογράφηση (λόγω του προβλήματος που σημειώθηκε στην Ενότητα 3.4, με την επιλογή του πλήκτρου “Start recording” δεν γίνεται ηχογράφηση αλλά ανάγνωση ενός προηχογραφημένου ηχητικού δοκιμίου από το σύστημα αρχείων της κάρτας αποθήκευσης της εικονικής συσκευής).

Κατά τη διάρκεια της ηχογράφησης (ανάγνωσης του αρχείου), εμφανίζεται η Εικόνα 15.



Εικόνα 15: Μήνυμα αναμονής από τα γραφικά στοιχεία της εφαρμογής.

Τα αποτελέσματα της ηχογράφησης, του υπολογισμού της περιβάλλουσας ενέργειας βραχέος χρόνου και της κατάτμησης εμφανίζονται στην Εικόνα 16.



Εικόνα 16: Τα γραφικά στοιχεία της εφαρμογής εμφανίζουν την κυματομορφή, την ενέργεια βραχέος χρόνου και τα αποτελέσματα της κατάτμησης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

4. Συμπεράσματα

Η εν λόγω εργασία δεν εμβαθύνει, ούτε προτείνει κάποια πρωτοπόρα τεχνική μεθοδολογία για την ομαδοποίηση ή την εξαγωγή των χαρακτηριστικών του ήχου. Ωστόσο, η πρόκληση που παρουσιάστηκε στην παρούσα εργασία δεν ήταν άλλη από την ανάγκη ελαχιστοποίησης των παραπάνω αλγόριθμων σε ένα ελάχιστο σύνολο ενεργειών, για τη μεταφορά τους στην πλατφόρμα Android, δηλαδή σε μικροσυσκευές με χαμηλή υπολογιστική ισχύ και ελάχιστη μνήμη. Φυσικά, η επιτυχία του εγχειρήματος καθορίζεται σε μεγάλο βαθμό και από το μέγεθος του ηχητικού δείγματος, συνεπώς, επαφίεται και στον τελικό χρήστη η “ευθύνη” χρήσης μικρών σε διάρκεια ηχογραφήσεων.

Το δίχως άλλο, αρκετά ενδιαφέρον ήταν και το εγχείρημα μεταφοράς των προαναφερθέντων αλγόριθμων σε μια γλώσσα προγραμματισμού κατ' εξοχήν αντικειμενοστρεφής, όπως η Java. Η μέθοδος προσέγγισης αναλύθηκε σε αρκετά μεγάλο βαθμό στην Ενότητα 3.3, όπου έγινε εκτενής περιγραφή της προσέγγισης όπου ακολουθήθηκε κατά την υλοποίηση και το “δέσιμο” της λογικής επεξεργασίας ηχητικού σήματος με την αντικειμενοστρεφή λογική, αλλά και τα ειδικά τεχνικά ζητήματα που εγείρονταν κατά τόπους εξαιτίας της πλατφόρμας Android.

Σαφέστατα, ο τελικός αλγόριθμος επιδέχεται μεγάλων βελτιώσεων, στο βαθμό πάντα του δυνατού. Επίσης, η ομαδοποίηση θα μπορούσε να ερευνηθεί περισσότερο και να εισαχθεί η δυνατότητα προσδιορισμού του πλήθους των ομάδων.

Αντίστοιχα, ο αλγόριθμος σύνθεσης κινείται στην απλή αναπαραγωγή έτοιμων ηχητικών δοκιμίων, συνεπώς επιδέχεται αρκετών βελτιώσεων, στο βαθμό πάντα που δεν θα εξαντλήσει την υπολογιστική ισχύ της συσκευής Android.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

5. Βιβλιογραφία – Αναφορές

Eckel B., “Thinking in Java, Fourth Edition”, Prentice Hall, 2006.

Murphy M. L., “Android Programming Tutorials”, CommonsWare LLC, 2011.

Theodoridis S., Pikrakis A., Koutroumbas K., Cavouras D., “Introduction to Pattern Recognition, A MATLAB Approach”, Academic Press (imprint of Elsevier), 2010.

http://www.music-ir.org/mirex/wiki/Query_by_Tapping, “MIREX – Query by Tapping”.

http://www.labbookpages.co.uk, “The Lab Book Pages”.

http://www.softsynth.com/jsyn/, “Jsyn Audio Synthesis API for Java”.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ