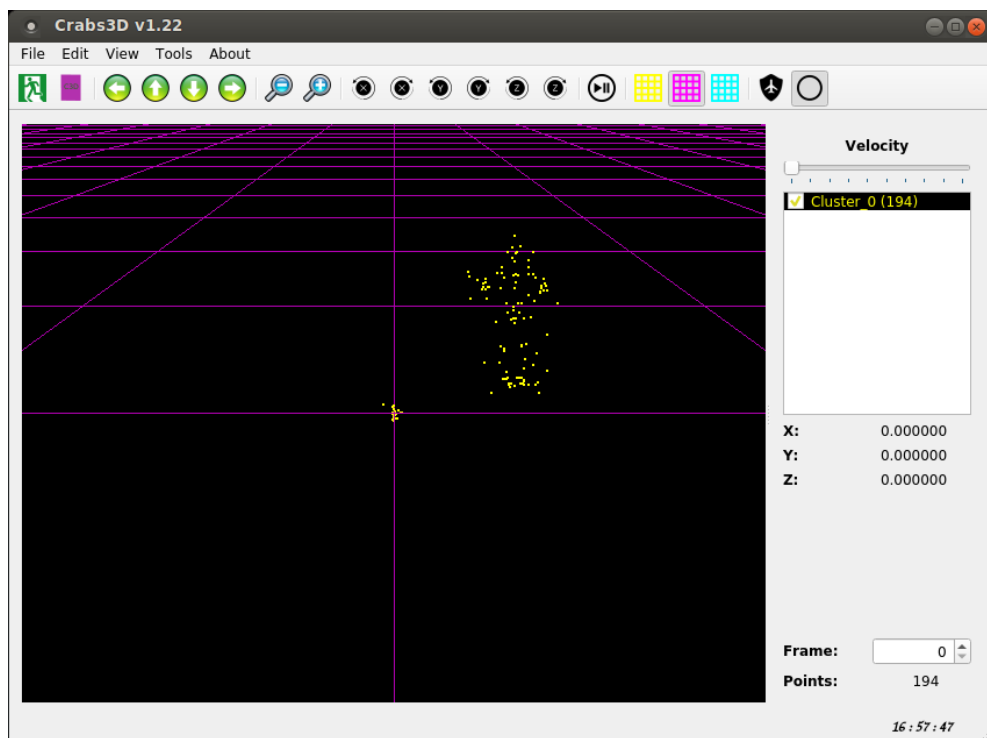

Ανάπτυξη Πλατφόρμας Οπτικοποίησης 3D Σημείων Αρθρώσεων Χορευτικών Κινήσεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΙΩΑΝΝΗΣ ΚΑΒΟΥΡΑΣ



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2017-2018



National
Technical
University of
Athens

Περιεχόμενα

1	Εισαγωγή	5
1.1	Άυλη Πολιτιστική Κληρονομιά	5
1.2	Ελληνικοί Παραδοσιακοί Χοροί	6
1.3	Περίληψη της Διπλωματικής	7
2	Περιγραφή των Αρχείων C3D	9
2.1	Γενικές Πληροφορίες	9
2.2	Η Βασική Δομή	9
2.3	Επισκόπηση	10
2.3.1	Γενική Εφαρμογή	10
2.4	Το αρχείο C3D	11
2.4.1	Κεφαλίδα	11
2.4.2	Ενότητα Παραμέτρων	11
2.4.2.α'	Κεφαλίδα Παραμέτρων:	13
2.4.2.β'	Ομάδες Παραμέτρων:	14
2.4.2.γ'	Παράμετροι:	15
2.4.2.δ'	Παραδείγματα Δομών Παραμέτρων:	16
2.4.3	Ενότητα Δεδομένων	17
2.4.3.α'	Τρισδιάστατα Δεδομένα - Ακέραη Μορφοποίηση:	18
2.4.3.β'	Τρισδιάστατα Δεδομένα - Δεκαδική Μορφοποίηση:	19
3	Σύντομη Περιγραφή της OpenGL	21
3.1	Τι είναι η OpenGL	21
3.2	Σύνταξη των συναρτήσεων της OpenGL	22
3.3	Χρήση της OpenGL στον κώδικα	22
3.3.1	glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)	22
3.3.2	glClear(GLbitfield mask)	23
3.3.3	glPolygonMode(GLenum face, GLenum mode)	23
3.3.4	glMatrixMode(GLenum mode)	23
3.3.5	glEnable(GLenum cap)	23
3.3.6	glViewport(GLint x, GLint y, GLsizei width, GLsizei height)	23
3.3.7	glLoadIdentity()	23
3.3.8	glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)	24
3.3.9	glPointSize(GLfloat size)	24
3.3.10	glLineWidth(GLfloat width)	24
3.3.11	glBegin(GLenum mode) και glEnd()	24
3.3.12	glColor3f(GLfloat red, GLfloat green, GLfloat blue)	24
3.3.13	glVertex3f(GLfloat x, GLfloat y, GLfloat z)	24
3.3.14	glFlush()	24
4	Ανάπτυξη των Κλάσεων - Συναρτήσεων	25
4.1	Ανάγνωση των Δεδομένων	25
4.1.1	Class C3D	28
4.1.2	Class Header_c3d	30
4.1.3	Class Parameter_c3d	33
4.1.4	Class Data_c3d	34
4.2	Επεξεργασία Δεδομένων	35
4.2.1	Συναρτήσεις Διαχείρισης του Νέφους Σημείων	36
4.2.2	Συναρτήσεις Επεξεργασίας του Νέφους Σημείων	37
4.2.3	Συναρτήσεις Ταξινόμησης και Διαχείρισης των Ομάδων του Νέφους Σημείων	38

5	Επίδειξη της Διεπαφής	39
5.1	Εισαγωγή των Δεδομένων στην Εφαρμογή	39
5.2	Αφαίρεση του Θορύβου από το Νέφος Σημείων	42
5.2.1	Αφαίρεση του Συσσωρευμένου Θορύβου	42
5.2.2	Αφαίρεση του Συγχεόμενου Θορύβου	44
5.2.3	Διόρθωση του Νέφους Σημείων	46
5.3	Ταξινόμηση - Μέθοδος K-Means	48
5.4	Αναγνώριση των Κλάσεων	51
5.5	Εξαγωγή των Αποτελεσμάτων	53
5.6	Crabs Editor	57
6	Εκπαίδευση Νευρωνικού Δικτύου	59
6.1	Δημιουργία Δεδομένων	59
6.2	Περιγραφή της Μεθόδου Convolution Neural Network	59
6.3	Περιγραφή των Μεγεθών Αξιολόγησης	60
6.4	Πειραματικά Αποτελέσματα	60
7	Εναλλακτικές Λύσεις και Προτάσεις	63
7.1	Αφαίρεση Θορύβου με τη Μέθοδο Ταξινόμησης DBScan	63
7.2	Διόρθωση του Νέφους Σημείων με Χρήση Γειτόνων	64
7.3	Ταξινόμηση με Χρήση της Μεθόδου Bounding Volume	65
8	Επίλογος - Συμπεράσματα - Μελλοντικές Επεκτάσεις	67
8.1	Γνώση που Έλαβα	67
8.2	Προσφορά της Διπλωματικής	67
8.3	Μελλοντικά Πειράματα	68
8.4	Επίλογος	68
9	Βιβλιογραφία	69

Κεφάλαιο 1

Εισαγωγή

1.1 Άυλη Πολιτιστική Κληρονομιά

Η ιδέα της Άυλης Πολιτιστικής Κληρονομιάς – Intangible Culture Inheritance (ICH) – θεωρείται από τα μέλη της UNESCO ως η κληρονομιά που ασχολείται με τις άυλες πτυχές του πολιτισμού. Το 2001 η UNESCO πραγματοποίησε έρευνα μεταξύ των κρατών και των μη κυβερνητικών οργανώσεων προσαπλώντας να συμφωνήσει για έναν ορισμό σχετικά με την άυλη πολιτιστική κληρονομιά, ενώ το 2003 καταρτίστηκε η σύμβαση για την ανάδειξη και τη διαφύλαξη της [20], [4], [8].

Η σύμβαση για τη διαφύλαξη της Άυλης Πολιτιστικής Κληρονομιάς ορίζει ως άυλη πολιτιστική κληρονομιά τις πρακτικές, τις αναπαραστάσεις, τις εκφράσεις, καθώς και τις γνώσεις και δεξιότητες (περιλαμβάνοντας μουσικά όργανα, αντικείμενα, τεχνουργήματα, πολιτιστικούς χώρους), τα οποία κοινότητες ή ομάδες αναγνωρίζουν ως κομμάτι της πολιτιστικής τους κληρονομιάς. Σε ορισμένες περιπτώσεις καλείται και ζωντανή πολιτιστική κληρονομιά και εκδηλώνεται μεταξύ άλλων στους ακόλουθους τομείς[18]:

- Προφορικές παραδόσεις και εκφράσεις, συμπεριλαμβανομένης και της γλώσσας ως μέσου της άυλης πολιτιστικής κληρονομιάς.
- Τέχνες του θεάματος.
- Κοινωνικές πρακτικές, τελετουργίες και εορταστικά γεγονότα.
- Γνώση και πρακτικές σχετικά με τη φύση και το σύμπαν.
- Παραδοσιακή τεχνική.

Η πολιτιστική κληρονομιά γενικά αποτελείται από προϊόντα και διαδικασίες μιας κουλτούρας, οι οποίες διατηρούνται και μεταφέρονται μέσω των γενεών. Κάποια από αυτή την κληρονομιά παίρνει τη μορφή πολιτιστικής περιουσίας, που αποτελείται από απτά έργα τέχνης (χτίρια) ή άλλες μορφές τέχνης (πίνακες ζωγραφικής, γλυπτά, μάσκες κλπ). Πολλά μέρη του πολιτισμού, ωστόσο, είναι άυλα, συμπεριλαμβανομένου του τραγουδιού, της μουσικής, του χορού, του δράματος, των δεξιοτήτων, της κουζίνας, των βιοτεχνιών και των φεστιβάλ. Είναι μορφές κουλτούρας που μπορούν να καταγραφούν, αλλά δεν μπορούν να αγγιχτούν ή να αποθηκευτούν σε φυσική μορφή, όπως σε ένα μουσείο, αλλά μόνο μέσα σε ένα «όχημα» που μπορεί να το εκφράζει, αναπαριστά ή αναπαράγει. Ο ΟΗΕ ονόμασε αυτά τα πολιτιστικά οχήματα «Ανθρώπινους Θησαυρούς» (Human Treasures) [27], [21].

Σύμφωνα με τη σύμβαση του 2003 για τη διαφύλαξη της άυλης πολιτιστικής κληρονομιάς, η άυλη πολιτιστική κληρονομιά (ζωντανή κληρονομιά) είναι το κύριο στοιχείο της πολιτιστικής ποικιλομορφίας της ανθρωπότητας και η διατήρησή της αποτελεί εγγύηση για τη συνεχή δημιουργικότητα. Η άυλη πολιτιστική κληρονομιά ορίστηκε συγκεκριμένα ως εξής:

«Ως άυλη πολιτιστική κληρονομιά νοούνται οι πρακτικές, οι παραστάσεις, οι εκφράσεις, οι γνώσεις, οι δεξιότητες – καθώς και τα όργανα, τα αντικείμενα, τα τεχνουργήματα και τα πολιτιστικά περιβαλλοντικά που συνδέονται με αυτά – που οι κοινότητες, οι ομάδες και σε ορισμένες περιπτώσεις, τα άτομα αναγνωρίζουν ως μέρος της πολιτιστικής τους κληρονομιάς. Αυτή η άυλη πολιτιστική κληρονομιά που μεταδίδεται από γενιά σε γενιά, αναπαράγεται διαρκώς από κοινότητες και ομάδες ως απάντηση στο περιβάλλον τους, την αλληλεπίδρασή τους με τη φύση και την ιστορία τους και τους παρέχει μία αίσθηση ταυτότητας και συνέχειας, προάγοντας έτσι το σεβασμό της πολιτιστικής πολυμορφίας της ανθρώπινης δημιουργικότητας. Για τους σκοπούς της παρούσας σύμβασης, θα ληφθεί υπόψη μόνο η άυλη πολιτιστική κληρονομιά που είναι συμβατή με τα υφιστάμενα διεθνή μέσα για τα ανθρώπινα δικαιώματα, καθώς και τις απαιτήσεις αμοιβαίου σεβασμού μεταξύ των κοινοτήτων, των ομάδων, των ατόμων και της αειφόρου ανάπτυξης[...]

Κάποιες σημαντικές μορφές έκφρασης της άυλης πολιτιστικής κληρονομιάς παρατίθενται παρακάτω:

1. **Προφορική Ιστορία:** Η άυλη πολιτιστική κληρονομιά είναι ελαφρώς διαφορετική από την πειθαρχία της προφορικής ιστορίας, την καταγραφή, τη διατήρηση και την ερμηνεία των ιστορικών πληροφοριών (συγκεκριμένα, τη προφορική παράδοση), με βάση τις προσωπικές εμπειρίες και απόψεις του

ομιλητή. Η άυλη πολιτιστική κληρονομιά επιχειρεί να διατηρήσει την πολιτιστική κληρονομιά με τον λαό ή την κοινότητα, προστατεύοντας τις διαδικασίες που επιτρέπουν τη μετάδοση των παραδόσεων και της κοινής γνώσης, ενώ η προφορική ιστορία επιδιώκει να συλλέξει και να διαφυλάξει ιστορικές πληροφορίες που προέρχονται από άτομα και ομάδες.

2. **Διατροφική Κληρονομιά:** Δεδομένου ότι η αειφόρος ανάπτυξη κερδίζει τη δυναμική ως προτεραιότητα των πολιτικών της UNESCO για την κληρονομιά, ένας αυξανόμενος αριθμός σχετικών με τα τρόφιμα υποψηφιοτήτων υποβάλλονται για εγγραφή στους καταλόγους της σύμβασης για τη διαφύλαξη της άυλης πολιτιστικής κληρονομιάς. Η μεσογειακή διατροφή, η παραδοσιακή μεξικάνικη κουζίνα και η ιαπωνική διατροφική κουλτούρα του washoku είναι μόνο μερικά παραδείγματα αυτού του ανερχόμενου φαινομένου.
3. **Χορευτική Κληρονομιά:** Οι κατάλογοι της UNESCO για την άυλη πολιτιστική κληρονομιά περιλαμβάνουν επίσης ποικίλα είδη χορού, συχνά συνδεδεμένα με το τραγούδι, τη μουσική και τις γιορτές, από όλο τον κόσμο. Οι κατάλογοι περιλαμβάνουν: εορταστικούς και τελετουργικούς χορούς, όπως τη μουσική και το χορό Ma'di Bowre Lyre από την Ουγκάντα και τα παραδοσιακά τραγούδια και χορούς της Καλβέλλας από το Ινδικό Ρατζαστάν και κοινωνικούς χορούς, όπως η ρούμπα από την Κούβα. Επίσης ορισμένοι χοροί εντοπίζονται και ασκούνται κυρίως στη χώρα προέλευσής τους, όπως η Sankirtana, ένα καλλιτεχνικό δρώμενο που περιλαμβάνει τυμπανοκρουσίες και τραγούδι, από την Ινδία.

Άλλες μορφές χορού, παρόλο που είναι επίσημα αναγνωρισμένες ως κληρονομιά από τη χώρα καταγωγής τους, ασκούνται και απολαμβάνονται σε όλο τον κόσμο. Για παράδειγμα, το φλαμένκο, από την Ισπανία, και το τάνγκο, από την Αργεντινή (Σχήμα 1.1) και την Ουρουγουάη, έχουν μία διεθνή διάσταση. Ο χορός είναι ένα πολύ περίπλοκο φαινόμενο, το οποίο περιλαμβάνει τον πολιτισμό, τις παραδόσεις, τη χρήση ανθρώπινων σωμάτων, αντικειμένων (όπως κοστούμια και σκηνικά), καθώς και μία συγκεκριμένη χρήση της μουσικής, του χώρου και μερικές φορές του φωτός. Ως αποτέλεσμα, πολλά ενσώματα και άυλα στοιχεία συνδυάζονται στο χορό, καθιστώντας το ένα προκλητικό, αλλά εξαιρετικά ενδιαφέρον είδος κληρονομιάς που πρέπει να προστατευθεί.



Σχήμα 1.1. Αργεντινικό Τάνγκο

1.2 Ελληνικοί Παραδοσιακοί Χοροί

Στους αρχαίους χρόνους **χορός** λεγόταν ο χώρος, στον οποίο χόρευαν και τραγουδούσαν οι άνθρωποι εκείνης της εποχής. Σήμερα ως χορός, ορίζεται το σύνολο των ρυθμικών κινήσεων και συσπάσεων του σώματος, αυτό που στην αρχαιότητα καλούνταν ότσηση ή χορεία. Ο χορός αποτελεί ένα από τα αρχαιότερα εκφραστικά μέσα και χρονολογικά έπεται του τραγουδιού. Οι ρίζες του χορού στην Ελλάδα εντοπίζονται περίπου το 1000 π.Χ.

Από τους σύγχρονους ελληνικούς χορούς, δεν απουσιάζουν τα αρχαία ρυθμικά και μουσικά μοτίβα. Συγκεκριμένα, οι νεοελληνικοί χοροί διακρίνονται ουσιαστικά σε δύο βασικές κατηγορίες: στους «συρτούς», στους «πηδηχτούς» και στις παραλλαγές αυτών. Η ονομασία του κάθε χορού συνήθως σχετίζεται με τον τόπο καταγωγής του (ο συρτός-καλαματιανός, ο μακεδονικός κλπ) ή έχει την ονομασία κυρίων προσώπων (ο Μενούσης, ο Μανέτας) ή μπορεί να πάρει και την ονομασία του από διάφορες εποχές (ο πασχαλινός χορός) ή να προέρχεται από τις ονομασίες επαγγελματιών (ο χορός των σφουγγαράδων). Οι ελληνικοί χοροί παρουσιάζουν αξιοσημείωτη ποικιλία παραλλαγών και επηρεάζονται μορφολογικά από τον τόπο καταγωγής τους. Έτσι συναντώνται χοροί οι οποίοι έχουν «βαρύ» ύφος (χοροί της ηπειρωτικής Ελλάδας) και αποκαλούνται συχνά «λεβέντικοι», ενώ υπάρχουν και χοροί που είναι περισσότερο «ανάλαφροι» και λυρικοί (χοροί της νησιωτικής Ελλάδας).

Γενικότερα, οι παραδοσιακοί χοροί της Ελλάδας παρουσιάζουν μεγάλη ποικιλία. Κάθε περιοχή ή χωριό της Ελλάδας έχει τους δικούς του χορούς, οι οποίοι διαφέρουν από περιοχή σε περιοχή ή ακόμη και από χωριό σε χωριό. Αυτή η διαφορά των χορών οφείλεται σε λόγους, όπως το κλίμα, ο τρόπος ζωής των κατοίκων, οι πόλεμοι, οι καταστροφές και διάφορων άλλων λόγων. Οι παραδοσιακοί ελληνικοί χοροί, λόγω της ποικιλίας που παρουσιάζουν, χωρίζονται σε επιμέρους κατηγορίες, όπως φαίνεται παρακάτω[35]:

1. Ανάλογα με το θέμα τους χωρίζονται σε:

- | | |
|------------------------------|------------------------------|
| (α') Θρησκευτικούς | (γ') Ερωτικούς |
| (β') Πολεμικούς ή Ψυρρίχιους | (δ') Πόλεμο-ερωτικούς χορούς |

2. Ανάλογα με το σχήμα τους χωρίζονται σε:

- | | |
|----------------|--------------------------|
| (α') Κυκλικούς | (β') Αντικριστούς χορούς |
|----------------|--------------------------|

3. Ανάλογα με το φύλλο χωρίζονται σε:

- | | |
|------------------|---------------|
| (α') Αντρικούς | (γ') Μεικτούς |
| (β') Γυναικείους | |

4. Ανάλογα με τον τόπο χωρίζονται σε:

- | | |
|------------------------------|---------------|
| (α') Πανελλήνιους ή Εθνικούς | (β') Τοπικούς |
|------------------------------|---------------|

Οι κυκλικοί χοροί συναντώνται κυρίως σε ανοικτό κύκλο και πιο σπάνια σε κλειστό. Τους αντικριστούς χορούς (καρσιλαμάδες) αποτελούν οι χοροί, στους οποίους οι χορευτές στέκονται σε δύο σειρές απέναντι (αντικριστά). Υπάρχουν βέβαια και χοροί, όπως το ζειμπέκικο, που χορεύονται από ένα άτομο, και άλλοι, όπως ο χορός των μαχαριών, που χορεύονται από δύο άτομα.

Οι πανελλήνιοι χοροί είναι ο συρτός-καλαματιανός και ο τσάμικος ή κλέφτικος. Τοπικοί χοροί είναι (i)οι Ηπειρωτικοί, (ii)οι Θρακιώτικοι, (iii)οι Νησιωτικοί, (iv)οι Κρητικοί, (v)οι χοροί της Μακεδονίας, (vi)οι ποντιακοί κλπ.

Ο κάθε χορός χορεύεται διαφορετικά, έχει τη δική του μουσική και το όνομά του έχει κάποια σημασία. Υπάρχουν βέβαια και κάποιοι χοροί, όπως της βόρειας Ελλάδος, οι οποίοι φέρουν αλληλεπιδράσεις με χορούς των γειτονικών λαών(βαλκανικών) και χορεύονται με διάφορες κατά τόπους παραλλαγές σε όλη τη βαλκανική περιοχή. Σε άλλες περιπτώσεις, όπως σε μερικά νησιά, η αλληλεπίδραση προέρχεται από την ύπαρξη της εκάστοτε κατεχόμενης δύναμης (Ενετοκρατία, Φραγκοκρατία, Τουρκοκρατία). [34]

1.3 Περίληψη της Διπλωματικής

Η παρούσα διπλωματική στοχεύει στη δημιουργία ενός μέσου οπτικοποίησης των ελληνικών παραδοσιακών χορών, μέσα στο οποίο θα μπορεί να αποθηκεύεται, να επεξεργάζεται και να αναπαράγεται, οποιοσδήποτε χορός. Η συλλογή των πρωτογενών δεδομένων είχε προηγηθεί και πραγματοποιήθηκε με χρήση Nikon κάμερας και το αρχείο αποθήκευσης είναι μορφοποίησης C3D (τα δεδομένα δόθηκαν από το εργαστήριο φωτογραμμετρίας). Πιο συγκεκριμένα, τα δεδομένα αποτελούνται από τους εξής παραδοσιακούς χορούς:

- Συρτό Γραμμικό στα 2
- Συρτό Κυκλικό στα 2
- Συρτό Γραμμικό στα 3
- Συρτό Κυκλικό στα 3
- Συρτό Μακεδονικό Γραμμικό
- Συρτό Μακεδονικό Κυκλικό
- Συρτό Καλαματιανό Γραμμικό
- Συρτό Καλαματιανό Κυκλικό
- Τρεχάτο Γραμμικό
- Τρεχάτο Κυκλικό
- Έντεκα

Για την ανάγνωση, επεξεργασία και οπτικοποίηση των δεδομένων, έχει συνταχθεί κώδικας, σε γλώσσα προγραμματισμού C/C++, ενώ χρησιμοποιήθηκαν οι βιβλιοθήκες Qt (για τη δημιουργία του παραθύρου απόδοσης και της φιλικής αλληλεπίδρασης μεταξύ λογισμικού και χρήστη) και OpenGL (για την οπτικοποίηση των δεδομένων), ενώ για την ανάγνωση των δεδομένων δημιουργήθηκε ξεχωριστή βιβλιοθήκη, η οποία περιορίζεται στα πλαίσια του σκοπού της διπλωματικής εργασίας.

Στο επόμενο στάδιο χρησιμοποιούνται αλγόριθμοι αφαίρεσης θορύβου και αλγορίθμων ταξινόμησης για τον προσδιορισμό των βασικών μελών του ανθρώπινου σκελετού. Τέλος τα επεξεργασμένα δεδομένα αποτελούν στοιχεία εισόδου σε νευρωνικό δίκτυο (αλγόριθμο deep learning) για την εκπαίδευση του υπολογιστή στην αναγνώριση του ανθρώπινου σκελετού. Όλα τα πειράματα πραγματοποιήθηκαν στο χορό Συρτό Γραμμικό στα 2 (Σχήμα 1.2), ενώ η εκπαίδευση του υπολογιστή έγινε σε περιβάλλον MATLAB.



Σχήμα 1.2. Συρτός

Κεφάλαιο 2

Περιγραφή των Αρχείων C3D

2.1 Γενικές Πληροφορίες

Η μορφοποίηση C3D (coordinate 3D) δεδομένων αποτελεί μία από τις πολλές μορφοποιήσεις που αναπτύχθηκαν για το φωτογραμμετρικό λογισμικό AMASS. Το λογισμικό AMASS αναπτύχθηκε από τον Andrew Dainis, Ph.D. ως εμπορικό προϊόν κατά τα έτη 1986-1987 προκειμένου να αντικαταστήσει τα σχετικά αναποτελεσματικά και ανακριβή βιομηχανικά φωτογραμμετρικά λογισμικά που ήταν διαθέσιμα εκείνη την περίοδο. Η πρώτη χρήση του λογισμικού αυτού ήταν για ιατρικούς ερευνητικούς σκοπούς στο Βιομηχανικό Εργαστήριο του Εθνικού Ινστιτούτου Υγείας των Ηνωμένων Πολιτειών

Η μορφοποίηση C3D παρέχει ένα άνετο και αποτελεσματικό τρόπο αποθήκευσης τρισδιάστατων συνηταγμένων και αναλογικών δεδομένων, μαζί με όλες τις σχετικές παραμέτρους που συνοδεύουν κάθε μία μέτρηση. Η χρήση της μορφοποίησης C3D ξεκίνησε από το 1987 και καθόρισε τις προδιαγραφές για ένα κοινό πρότυπο αρχείων μορφοποίησης C3D [14].

Η βασική ιδέα για την ανάπτυξη της μορφοποίησης C3D ήταν η ανάγκη να αποθηκεύονται όλα τα τρισδιάστατα σημεία και αριθμητικά δεδομένα κάθε μέτρησης σε ένα αρχείο μαζί με τις διάφορες παραμέτρους που τα περιγράφουν. Πριν την υλοποίηση αυτής της μορφοποίησης τα τρισδιάστατα σημεία και αριθμητικά δεδομένα, καθώς και οι παράμετροι αυτών, αποθηκεύονταν σε ξεχωριστά αρχεία καθένα απ' αυτά με τη δική του μορφοποίηση κι έτσι δημιουργούνταν τα εξής προβλήματα:

1. Κάθε κατασκευαστής κατέβαλε μεγάλη προσπάθεια προκειμένου να αρχειοθετήσει και να διατηρήσει το μεγάλο αριθμό των αρχείων αυτών.
2. Οι ενημερώσεις και οι αλλαγές στις εφαρμογές απαιτούσαν προσεχτικό σχεδιασμό, ώστε να διατηρήσουν τη συμβατότητα τους με τα υποστηριζόμενα αρχεία.
3. Υπήρχε η αναγκαία απαίτηση από τους χρήστες να κατανοούν τι κάνει η κάθε μία μορφοποίηση ώστε να αξιοποιήσουν τα δεδομένα που είχαν συλλέξει.
4. Η σύγκριση μεταξύ ίδιων δεδομένων, αλλά διαφορετικών κατασκευαστών ήταν αδύνατη, αφότου υπήρχε ασυμβατότητα μεταξύ των μεθόδων αποθήκευσης των δεδομένων και παραμέτρων αυτών.
5. Συχνά οι ενημερώσεις των συστημάτων εισήγαγαν αλλαγές στις μορφοποιήσεις, οι οποίες καθιστούσαν τις παλιές μη αναγνώσιμες στις νεότερες εκδόσεις του προγράμματος.

Συνεπώς η ανάπτυξη της μορφοποίησης C3D αποτέλεσε έναν ενδεδειγμένο τρόπο επίλυσης των παραπάνω προβλημάτων. Η δημιουργία ενός μοναδικού αρχείου, καλά αρχειοθετημένου, αποθηκευμένου σε δυαδική μορφή απλούστεψε τόσο τη συντήρηση των λογισμικών πακέτων όσο και την αρχειοθέτηση των δεδομένων. Οι χρήστες είχαν πρόσβαση στα δεδομένα τους διαβάζοντας μόνο ένα αρχείο και η ευρεία χρήση του έκανε εφικτή κι εύκολη τη σύγκριση δεδομένων από διάφορα ερευνητικά εργαστήρια ανά τον κόσμο, τα οποία χρησιμοποιούσαν διαφορετικά συστήματα καταγραφής της κίνησης (motion capture systems). Τέλος η προτυποποίηση της μορφοποίησης C3D έδωσε τη δυνατότητα στους κατασκευαστές να ενημερώνουν τα συστήματά τους ελεύθερα, χωρίς να υπάρχει ο κίνδυνος να καταστούν μη αναγνώσιμα τα δεδομένα που έχουν παραχθεί από προηγούμενα συστήματα.

2.2 Η Βασική Δομή

Το αρχείο μορφοποίησης C3D χαρακτηρίζεται από τρία τμήματα:

1. Δεδομένα: σε αυτό το επίπεδο το C3D αρχείο είναι απλώς ένα αρχείο δυαδικής μορφής στο οποίο βρίσκεται αποθηκευμένη καθαρή(raw) τρισδιάστατη και αναλογική πληροφορία.
2. Πρότυπες Παράμετροι: προκαθορισμένη πληροφορία σχετικά με τα καθαρά τρισδιάστατα και αναλογικά δεδομένα που απαιτείται για την ανάγνωσή τους.
3. Προσαρμοσμένες Παράμετροι: πληροφορίες προσδιορισμένες για λογισμικό από συγκεκριμένο κατασκευαστή.

Ένας κατασκευαστικός στόχος της μορφοποίησης C3D ήταν να καταστήσει εύκολο στο χρήστη την καταγραφή, την εξέταση, και αν είναι αναγκαίο την τροποποίηση, κάθε παραμέτρου που περιέχεται σε ένα C3D αρχείο.

Άλλοι στόχοι της μορφοποίησης C3D ήταν η ελαχιστοποίηση του απαραίτητου αποθηκευτικού χώρου στο δίσκο, η μείωση του αριθμού των απαραίτητων αρχείων περιγραφής των δεδομένων, η επαρκής ταχύτητα κι η εύκολη πρόσβαση στο περιεχόμενο των αρχείων. Επιπλέον προκειμένου να επιτραπεί στο συνήθη χρήστη να προβάλλει και να τροποποιεί όλες τις παραμέτρους, η μορφοποίηση C3D επιτρέπει στις παραμέτρους της να περιέχουν μοναδικές περιγραφές για κάθε παραμετρικό αντικείμενο, ώστε οι διάφορες συναρτήσεις κάθε παραμέτρου μπορούν να αρχειοθετηθούν μέσα στο ίδιο το C3D αρχείο. Κατά αυτό τον τρόπο δίνεται η ευελιξία στους χρήστες να αποθηκεύουν πολλά διαφορετικά είδη δεδομένων σχετιζόμενα με τις παραμέτρους εντός του C3D αρχείου, καθώς διατηρείται σε ένα βαθμό μία εσωτερική αρχειοθέτηση η οποία λείπει από άλλες μορφοποιήσεις αρχείων.

Τα C3D αρχεία δεν έχουν κανένα συγκεκριμένο όριο όσον αφορά τα δεδομένα ή τον τύπο και αριθμό των παραμέτρων που μπορούν να αποθηκεύσουν. Η μορφοποίηση είναι εύκολα επεκτάσιμη ώστε να αποθηκεύονται επιπλέον παράμετροι και δεδομένα. Χάρης σε αυτό, η συμβατότητα των C3D αρχείων είναι διαχρονική. Αυτό σημαίνει ότι αρχεία που δημιουργήθηκαν στα μέσα της δεκαετίας του 1980 να μπορούν να διαβαστούν από σημερινά προγράμματα, καθώς επίσης και προγράμματα που κατασκευάστηκαν εκείνη την περίοδο, μπορούν να διαβάσουν αρχεία του σήμερα.

Όπως ήδη έχει αναφερθεί, τα C3D αρχεία είναι δυαδικής μορφοποίησης(binary format), αν και στα αρχεία ASCII μπορεί εύκολα να περιέχεται περιγραφή σχετιζόμενη με παραμέτρους και είναι πιο εύκολη η πρόσβαση σε αυτά από πολλές εφαρμογές, η επιλογή της μορφοποίησης ASCII κρίνεται αναποτελεσματική τόσο ως προς την αποθήκευση όσο και ως προς την πρόσβαση. Στα αρχεία ASCII η πρόσβαση πρέπει να γίνεται διαδοχικά, ενώ παρατηρείται αναποτελεσματικότητα, όταν απαιτείται τυχαία πρόσβαση στα δεδομένα. Από την άλλη μεριά, η μορφοποίηση δυαδικών αρχείων αποτελεί ένα αποτελεσματικό υποκατάστατο των αρχείων ASCII, αφότου διαχειρίζονται αποτελεσματικότερα την αποθήκευση και πρόσβαση των δεδομένων και μπορούν να αποθηκεύσουν πολλές διαφορετικές παραμέτρους και σχετικές περιγραφές αυτών. Ωστόσο, η πρόσβαση στα δυαδικά αρχεία απαιτεί συγκεκριμένες εφαρμογές. Στα δυαδικά αρχεία, τα δεδομένα αποθηκεύονται με ένα συγκεκριμένο τρόπο και οι εφαρμογές πρέπει να γνωρίζουν πλήρως της δομή του αρχείου, ώστε να έχουν πρόσβαση σε αυτό. Πέραν όμως από αυτή την πολυπλοκότητα, η αποτελεσματικότητα και η ταχύτητα στην πρόσβαση των δυαδικών αρχείων τους δίνει μεγάλο προβάδισμα και σαν αποτέλεσμα, η μορφοποίηση C3D επιλέχθηκε για την αποθήκευση βιομηχανικών δεδομένων και παραμέτρων.

Τέλος αξίζει να αναφερθεί πως από την αρχή της δημιουργίας των αρχείων μορφοποίησης C3D αναγνωρίστηκε η χρησιμότητα να δημιουργηθεί ένα κοινό πρότυπο για κάθε βιομηχανική χρήση. Ως αποτέλεσμα δημιουργήθηκε ένας μικρός αριθμός παραμέτρων, οι οποίες περιγράφουν θεμελιώδη στοιχεία για τα τρισδιάστατα δεδομένα και δίνετε επιπλέον η δυνατότητα στους χρήστες να καθορίζουν, παράγουν και αποθηκεύουν μέσα στο αρχείο, όσες παραμέτρους και περιγραφές θέλουν σχετικά με τα καθορισμένα εργαστηριακά αντικείμενα, ώστε καθένας που διαβάζει το αρχείο C3D με κατάλληλο λογισμικό να έχει πρόσβαση σε αυτά.

2.3 Επισκόπηση

Με την ευρύτερη έννοια το αρχείο μορφοποίησης C3D είναι μία συγκεκριμένη εφαρμογή μιας πιο γενικής μορφοποίησης (μορφοποίηση ADTech), η οποία έχει τα εξής χαρακτηριστικά:

- Το πρώτο byte του αρχείου δείχνει την τοποθεσία της αρχής των παραμέτρων. Το δεύτερο byte είναι πάντα 0x50h (στο δεκαδικό 80) και επιβεβαιώνει πως το αρχείο έχει κατασκευαστεί χρησιμοποιώντας τη μορφοποίηση ADTech.
- Οι παράμετροι, στην ενότητα των παραμέτρων, είναι αποθηκευμένοι σε ομάδες σύμφωνα με μία καλά οργανωμένη παραμετρική μορφοποίηση.
- Οι παράμετροι υποδεικνύουν την τοποθεσία των διαφόρων ενότητων των δεδομένων, στο αρχείο, καθώς επίσης μπορεί να περιγράφουν και το περιεχόμενο κάθε πρόσθετης ενότητας.

2.3.1 Γενική Εφαρμογή

Ένα C3D αρχείο είναι μία εφαρμογή της παραπάνω γενικής μορφοποίησης, όπου η πρώτη δομική ενότητα του αρχείου, αποτελείται από ένα πρότυπο δείκτη (ID word) ακολουθούμενη από μία εγγραφή κεφαλίδας

η οποία περιλαμβάνει έναν αριθμό παραμέτρων, οι οποίες στην πραγματικότητα είναι αντίγραφα των παραμέτρων της παραμετρικής ενότητας του C3D αρχείου. Τα C3D αρχεία επιπλέον περιλαμβάνουν μία ενότητα δεδομένων, στην οποία αποθηκεύεται τρισδιάστατη και αναλογική πληροφορία – η τοποθεσία της ενότητας δεδομένων περιγράφεται στην ενότητα των παραμέτρων.

2.4 Το αρχείο C3D

Οι τιμές των δεδομένων σε ένα C3D αρχείο αποθηκεύονται σε λέξεις των 16-bit προσημασμένης ακέραιας μορφής (signed integer format) ή εναλλακτικά σε δεκαδική μορφή (floating-point format). Η μορφή των δεδομένων μπορεί να προσδιοριστεί από την ανάγνωση της κεφαλίδας του αρχείου C3D, σε δυαδικό επίπεδο, χωρίς να χρειαστεί να γίνουν υποθέσεις για τη μορφή των δεδομένων. Για λόγους συμβατότητας με τη Fortran και για τα διάφορα λειτουργικά συστήματα, όλα τα αρχεία C3D πρέπει να είναι οργανωμένα σε δομές (blocks) των 512-bytes ή αλλιώς 256x16-bit λέξεις.

Όλα τα αρχεία C3D πρέπει να περιέχουν κατά ελάχιστον τις τρεις βασικές ενότητες πληροφορίας, οι οποίες είναι:

- Μία **κεφαλίδα** αποτελούμενη από 512-bytes.
- Μία **ενότητα παραμέτρων** αποτελούμενη από μία ή περισσότερες δομές των 512-bytes.
- Μία **ενότητα τρισδιάστατης και αναλογικής πληροφορίας (3d point and analog data section)** αποτελούμενη από μία ή περισσότερες δομές των 512-bytes.

Σημείωση: Κάθε **δομή** αποτελείται από **512-bytes**, συνεπώς κάθε φορά που θα χρησιμοποιείται η λέξη **δομή**, για το υπόλοιπο του κεφαλαίου, εννοείται **512-bytes**.

2.4.1 Κεφαλίδα

Η πρώτη ενότητα ενός C3D αρχείου είναι η κεφαλίδα, η οποία αρχίζει πάντα στο πρώτο byte της πρώτης δομής του αρχείου, ενώ αποτελείται αυστηρά από 512-bytes. Η πρώτη λέξη του C3D αρχείου προσδιορίζει την αρχή της ενότητας των παραμέτρων του αρχείου C3D.

Εκτός από το δείκτη που προσδιορίζει την αρχή της ενότητας των παραμέτρων, στην κεφαλίδα περιέχονται και αντίγραφα από διάφορες παραμέτρους της ενότητας των παραμέτρων.

Γενικά η τοποθεσία των τρισδιάστατων δεδομένων, καθώς και άλλες σημαντικές παράμετροι, συνιστάται να διαβάζονται πάντα από την ενότητα των παραμέτρων. Ο λόγος, όπου υπάρχει αντίγραφο αυτών των παραμέτρων στην κεφαλίδα, είναι για να μπορούν να δημιουργηθούν μικρά προγράμματα, τα οποία διαβάζουν τα τρισδιάστατα δεδομένα (3D points), χωρίς να χρειάζεται να αποκωδικοποιήσουν όλη την πληροφορία που υπάρχει στο αρχείο C3D.

Στον Πίνακα 2.1 περιγράφεται συνοπτικά η κεφαλίδα ενός C3D αρχείου.

2.4.2 Ενότητα Παραμέτρων

Η ενότητα των παραμέτρων συνήθως αρχίζει στη δεύτερη δομή του C3D αρχείου, αν και αυτό δεν είναι σταθερό και δε θα πρέπει να θεωρείται δεδομένο για κάθε αρχείο. Το αρχείο C3D προϋποθέτει πως η ενότητα των παραμέτρων αρχίζει στο όριο του 512 byte, το οποίο υποδεικνύεται από το δείκτη της κεφαλίδας των C3D. Η ενότητα των παραμέτρων είναι μεταβλητή σε μήκος, ωστόσο συνηθίζεται να αποτελείται από 8 έως 10 δομές.

Σε όλα τα C3D αρχεία υπάρχει η ενότητα των παραμέτρων, στην οποία αποθηκεύεται πληροφορία σχετικά με τα τρισδιάστατα και/ή τα αναλογικά δεδομένα που είναι αποθηκευμένα στο C3D αρχείο. Οι παράμετροι πρέπει να παρέχουν όλη την πληροφορία που χρειάζεται μία εφαρμογή, προκειμένου να έχει πρόσβαση στα δεδομένα του C3D αρχείου και να μπορεί να τα επεξεργαστεί.

Αν και στην κεφαλίδα των C3D αρχείων παρέχεται η πρόσβαση σε κάποιες βασικές πληροφορίες σχετικά με το περιεχόμενο του C3D αρχείου, όπως είναι ο αριθμός των τρισδιάστατων σημείων ή των αναλογικών καναλιών, οι παράμετροι, εντός της ενότητας των παραμέτρων, είναι αυτές που κάνουν το αρχείο καταληπτό. Για παράδειγμα, στην κεφαλίδα αναφέρεται πως το αρχείο περιέχει 50 καρτέ δεδομένων, καθένα από τα οποία περιέχει 20 τρισδιάστατα σημεία – ωστόσο στις παραμέτρους αναφέρεται πως το σημείο 10 σε κάθε καρτέ έχει την ετικέτα "IFLH" που μεταφράζεται ως "Index Finger at Left Hand".

Πίνακας 2.1. Κεφαλίδα C3D αρχείου.

Λέξη (16-bit)	Τυπική Τιμή	Περιγραφή
1	0x5002 hex	Byte 1: Δείχνει την τοποθεσία της αρχής της ενότητας των παραμέτρων. Byte 2: Τιμή κλειδί 0x50h (80 στο δεκαδικό) υποδεικνύοντας το C3D αρχείο.
2	nn	Ακέραιος αριθμός που περιγράφει τον αριθμό των τριδιάστατων δεδομένων (C3D points)
3	nn	Ο συνολικός αριθμός αναλογικών μετρήσεων ανά καρτέ, δηλαδή ο αριθμός των καναλιών πολλαπλασιαζόμενος με τα δείγματα ανά κανάλι.
4	1	Ο αριθμός του πρώτου καρτέ των τριδιάστατων δεδομένων.
5	nn	Ο αριθμός του τελευταίου καρτέ των τριδιάστατων δεδομένων.
6	10	Μέγιστο κενό παρεμβολής μεταξύ τριδιάστατων δεδομένων.
7-8	nnnn	Τριδιάστατος συντελεστής κλίμακας, ο οποίος μετατρέπει τα προσημασμένα ακέραια τριδιάστατα σημεία στο σύστημα αναφοράς της μετρημένης μονάδας. Εάν ο αριθμός αυτός είναι αρνητικός, τότε το αρχείο είναι υπό κλίμακα και τα δεδομένα είναι αποθηκευμένα σε δεκαδική μορφή (floating-point)
9	nn	DATA_START – ο αριθμός της αρχής της ενότητας των τριδιάστατων και αναλογικών δεδομένων.
10	nn	Ο αριθμός των αναλογικών δειγμάτων ανά καρτέ.
11-12	60,000	Η συχνότητα εναλλαγής των καρτέ σε Hz (floating-point)
13-147	0x00 hex	Κατοχυρωμένο για μελλοντική χρήση.
148	0x3039 hex	Μια τιμή κλειδί (12345 ακέραια) εάν υπάρχουν δεδομένα «Label and Range», αλλιώς η τιμή είναι 0x00 hex.
149	nn	Η αρχή της ενότητας «Label and Range» (εάν υπάρχει)
150	0x3039 hex	Μία τιμή κλειδί (12345 ακέραιος), εάν το αρχείο υποστηρίζει ετικέτες 4 χαρακτήρων. Μία παλαιότερη μορφοποίηση υποστήριζε μόνο ετικέτες 2 χαρακτήρων
151	0	Αριθμός των καθορισμένων χρονικών συμβάντων (0 έως 18)
152	0x00 hex	Κατοχυρωμένο για μελλοντική χρήση.
153-188	-	Χρονικά συμβάντα (floating-point) σε δευτερόλεπτα (το πολύ 18 συμβάντα)
189-197	-	18-bytes εμφάνιση συμβάντων, flags 0x00 = ON, 0x01 = OFF
198	0x00 hex	Κατοχυρωμένο για μελλοντική χρήση.
199-234	-	Ετικέτες συμβάντων. Κάθε ετικέτα είναι μήκους 4 χαρακτήρων.
235-256	0x00 hex	Κατοχυρωμένο για μελλοντική χρήση.

Χωρίς την ενότητα των παραμέτρων, το C3D αρχείο είναι απλά μία συλλογή δειγμάτων δεδομένων αποθηκευμένα σε μία μορφοποίηση δεδομένων. Η ενότητα των παραμέτρων είναι αυτή, η οποία κάνει το αρχείο C3D να είναι τόσο προσαρμόσιμο και λειτουργικό, ανεξάρτητα από την πηγή των δεδομένων.

Κάθε παράμετρος, στην ενότητα των παραμέτρων, προσδιορίζεται από ένα όνομα και ένα τύπο δεδομένων. Μία παράμετρος μπορεί να έχει διαστάσεις, οι οποίες περιγράφουν πόσα στοιχεία μπορεί να περιλαμβάνει κάποιο δεδομένο. Κάθε παράμετρος μπορεί επίσης να έχει μία περιγραφή σχετιζόμενη με την ιδιότητά της. Αν και το αρχείο C3D προαπαιτεί κάποιες συγκεκριμένες πληροφορίες ως παράμετρος, κάθε χρήστης μπορεί να δημιουργήσει τις δικές του παραμέτρους και κάθε εφαρμογή συμβατή με τη μορφοποίηση C3D μπορεί να διαβάσει αυτή την πληροφορία.

Σχετιζόμενες παράμετροι (για παράδειγμα μία συλλογή παραμέτρων που περιέχουν πληροφορία σχετικά με τα αναλογικά δεδομένα) μπορούν να οργανωθούν σε "ομάδες" ("groups") – κάθε παράμετρος μέσα σε ένα C3D αρχείο ανήκει σε μία συγκεκριμένη ομάδα. Κάθε ομάδα έχει ένα μοναδικό όνομα και μπορεί να έχει και μία περιγραφή σχετική με το περιεχόμενό του.

Όταν μία παράμετρος αναφέρεται σε λίστες και σε εντολές, το όνομα των ομάδων και το όνομα των παραμέτρων χωρίζονται με άνω-κάτω τελεία (:) έτσι ώστε μία παράμετρος, π.χ. "SCALE", η οποία ανήκει στην ομάδα "ANALOG" γράφεται ως ANALOG:SCALE – το όνομα της ομάδας γράφεται πριν το όνομα της παραμέτρου. Η απόδοση των παραμέτρων εντός των ομάδων με αυτό τον τρόπο, δίνει τη δυνατότητα να υπάρχουν ίδιες παράμετροι, σε διαφορετικές ομάδες, όπως για παράδειγμα η παράμετρος κλίμακα μπορεί να υπάρχει στις ομάδες της αναλογικής και τρισδιάστατης πληροφορίας ως ANALOG:SCALE και POINT:SCALE.

Το όνομα μίας παραμέτρου μπορεί να αποτελείται από συνδυασμούς χαρακτήρων αγγλικών γραμμάτων Α έως Ζ, αριθμών 0 έως 9 και την κάτω παύλα "_", ενώ τα άλλα σημεία στίξης ή εκτυπώσιμοι χαρακτήρες δεν μπορούν να χρησιμοποιηθούν. Τα ονόματα των ομάδων και των παραμέτρων, δεν πρέπει να ξεκινούν με αριθμό ή κάτω παύλα. Τέλος οι μικροί χαρακτήρες αν και μπορούν να χρησιμοποιηθούν για να ονοματίσουν μία ομάδα ή παράμετρο, δε συνιστάται, διότι το χρησιμοποιούμενο πρότυπο είναι η ονοματοδοσία να γίνεται με κεφαλαίους χαρακτήρες.

Για λόγους συμβατότητας μεταξύ των εφαρμογών, στις προδιαγραφές της μορφοποίησης C3D αναφέρεται ότι όταν το όνομα μίας ομάδας ή παραμέτρου ερμηνεύεται, μόνο οι πρώτοι έξι χαρακτήρες του ονόματος της ομάδας και οι πρώτοι έξι χαρακτήρες του ονόματος των παραμέτρων χρησιμοποιούνται. Επομένως είναι σημαντικό όλα τα ονόματα των ομάδων, και όλα τα ονόματα των παραμέτρων εντός των ομάδων, να εμφανίζουν τουλάχιστον μία διαφορά στους πρώτους έξι χαρακτήρες. Το ίδιο όνομα μπορεί να χρησιμοποιηθεί σε δύο παραμέτρους, όταν αυτές βρίσκονται σε διαφορετικές ομάδες.

Ο τύπος της παραμέτρου προσδιορίζει τον τύπο των δεδομένων που μπορούν να αποθηκευτούν. Υπάρχουν τέσσερις διαφορετικοί τύποι: ακέραιοι, δεκαδικοί, χαρακτήρες και ένα byte (integer, floating-point, character and byte). Ένας ακέραιος τύπος είναι ένας 16-bit προσημασμένος αριθμός μεταξύ -32768 και +32767 και ένας δεκαδικός τύπος είναι ένας αριθμός, ο οποίος περιέχει δεκαδικά ψηφία ή γράφεται σε επιστημονική μορφή (scientific exponential representation). Ένας χαρακτήρας αποτελεί ένα πραγματικό σύμβολο, όπως κάποιο γράμμα πληκτρολογίου, ενώ το ένα byte είναι ένας 8-bit προσημασμένος ακέραιος αριθμός μεταξύ -128 έως +127.

Ένας μη προσημασμένος 16-bit ακέραιος μπορεί να αποθηκεύσει θετικούς αριθμούς στο εύρος 0 έως +65535 καθώς επίσης τα μη προσημασμένα bytes έχουν ένα εύρος από 0 έως +255. Οι μη προσημασμένοι ακέραιοι εξορισμού υστερούν από κάθε δυνατότητα καταγραφής προσήμου κι έτσι είναι πάντα θετικοί. Η ιδιότητα αυτή είναι χρήσιμη για την καταγραφή μεγάλων αριθμών στην ενότητα των παραμέτρων εντός του C3D αρχείου, με την προϋπόθεση ότι είναι γνωστό ότι ο αριθμός θα είναι πάντα θετικός. Ως αποτέλεσμα κάποιες εφαρμογές διαχειρίζονται συγκεκριμένες παραμέτρους ως μη προσημασμένες - κάτι το οποίο τις δίνει την ικανότητα να καταγράφουν μεγαλύτερο αριθμό καρέ ή μεγαλύτερες παραμέτρους. Τα C3D αρχεία τα οποία περιλαμβάνουν μη προσημασμένους ακέραιους ονομάζονται «Μη Προσημασμένα C3D Αρχεία (Unsigned C3D Files)».

Συνοψίζοντας η ενότητα των παραμέτρων χωρίζεται σε τρία τμήματα:

1. Μία **κεφαλίδα** αποτελούμενη από 4-byte, στην οποία αναγράφονται κάποια στοιχεία σχετικά με τον τρόπο που πρέπει να διαβαστεί το αρχείο.
2. Μία **ομάδα παραμέτρων**, άγνωστου αριθμού bytes.
3. Οι **παράμετροι** εντός μίας ομάδας παραμέτρων, άγνωστου αριθμού bytes.

2.4.2.α' Κεφαλίδα Παραμέτρων: Στην κεφαλίδα του αρχείου C3D περιέχεται ένας δείκτης, ο οποίος προσδιορίζει την αρχή της ενότητας των παραμέτρων στο αρχείο. Ο δείκτης αυτός αντιστοιχεί

σε τιμή ενός byte και προσδιορίζει τον αριθμό των δομών των 512-bytes που αρχίζει η ενότητα των παραμέτρων θεωρώντας ως πρώτη δομή την κεφαλίδα του αρχείου.

Στην ενότητα των παραμέτρων, τα πρώτα τέσσερα bytes, αποτελούν μία κεφαλίδα με κάποια στοιχεία σχετιζόμενα με την προέλευση του αρχείου και τη δομή της ενότητας των παραμέτρων. Τα στοιχεία αυτά φαίνονται στον Πίνακα 2.2.

Πίνακας 2.2. Πίνακας Κεφαλίδας Παραμέτρων.

Byte	Τυπική Τιμή	Περιγραφή
1	0x00 hex	Κατοχυρωμένη για παραμετρική χρήση.
2	0x00 hex	Κατοχυρωμένη για παραμετρική χρήση.
3	Nn	Ο αριθμός των παραμετρικών δομών που θα ακολουθήσουν.
4	85	83 decimal + processor type Processor type 1 = Intel Processor type 2 = DEC (VAX, PDP-11) Processor type 3 = MIPS(SGIS/MIPS)

Τα πρώτα δύο byte έχουν νόημα, μόνον όταν η ενότητα των παραμέτρων βρίσκεται στην αρχή του C3D αρχείου και αποτελούν την πρώτη λέξη του αρχείου. Αυτό εξηγείται από τη γενική προτυποποίηση των ADTech αρχείων, η οποία απαιτεί το πρώτο byte να δείχνει την αρχή της ενότητας των παραμέτρων ενώ το δεύτερο να περιέχει τον ακέραιο αριθμό 80. Συνεπώς σε αυτή την περίπτωση αναμένεται τα πρώτα δύο bytes να έχουν την τιμή 0x5001 στο δεκαεξαδικό σύστημα, δηλαδή byte-1 = 1 και byte-2 = 80.

Το τρίτο byte στην κεφαλίδα των παραμέτρων περιέχει τον αριθμό των δομών, από τις οποίες αποτελείται η ενότητα των παραμέτρων, θεωρώντας ως πρώτη δομή αυτή που περιέχει το πρώτο byte της κεφαλίδας των παραμέτρων.

Το τέταρτο byte της κεφαλίδας των παραμέτρων αντιστοιχεί στον **τύπο του επεξεργαστή** από τον οποίο προήλθαν τα δεδομένα. Η αιτία που υπάρχει αυτή η πληροφορία στην κεφαλίδα των παραμέτρων, είναι για να δώσει την ευελιξία στην εφαρμογή να αναγνωρίσει τον τρόπο με τον οποίο διαβάζονται οι δεκαδικοί και οι προσημασμένοι αριθμοί στο C3D αρχείο.

2.4.2.β' Ομάδες Παραμέτρων: Το πρώτο byte στις ομάδες παραμέτρων αντιστοιχεί στον αριθμό των χαρακτήρων του ονόματος της ομάδας. Το όνομα μίας ομάδας μπορεί να έχει από 1 έως 127 χαρακτήρες (χρησιμοποιώντας αγγλικούς χαρακτήρες – κεφαλαίους ή μικρούς – και αριθμούς), αν και πρέπει να λαμβάνεται υπόψη πως το συνολικό είναι οι 4 χαρακτήρες να χρησιμοποιούνται ως ελάχιστο. Επιπλέον συνιστάται οι πρώτοι έξι χαρακτήρες να είναι μοναδικοί, ώστε να υπάρχει συμβατότητα ανάγνωσης του αρχείου από παλαιότερα λογισμικά. Τέλος ο αριθμός των χαρακτήρων, διαβάζεται πάντα ως θετικός ανεξάρτητα από το πρόσημο με το οποίο έχει αποθηκευτεί. Στην περίπτωση που ο αριθμός είναι αρνητικός, τότε η ομάδα θεωρείται κλειδωμένη.

Το δεύτερο byte περιέχει τον κωδικό αριθμό της ομάδας – ο αριθμός αυτός είναι πάντα μεταξύ -1 και -127 (πρέπει να διαβαστεί ως ένα byte) και χρησιμοποιείται για να συνδέει τις παραμέτρους με τις ομάδες τους. Οι παράμετροι έχουν θετικούς κωδικούς αριθμούς και ανήκουν στην ομάδα που έχει τον ίδιο κατά απόλυτη τιμή αριθμό, αλλά με αντίθετο πρόσημο.

Η ροή, η οποία περιέχει το όνομα της ομάδας αρχίζει στο τρίτο byte. Το όνομα κάθε ομάδας, όπως αναφέρθηκε ήδη, μπορεί να έχει από 1 έως 127 χαρακτήρες, αν και οι τέσσερις χαρακτήρες θα πρέπει να θεωρούνται ως ελάχιστοι. Τα ονόματα των ομάδων δεν πρέπει να αρχίζουν με αριθμό.

Τα επόμενα δύο byte δείχνουν την αρχή της επόμενης παραμέτρου ή ομάδας. Ενώ το επόμενο από αυτό το byte, αντιστοιχεί στον αριθμό των χαρακτήρων της περιγραφής της ομάδας ή παραμέτρου. Η περιγραφή μπορεί να περιέχει οποιονδήποτε χαρακτήρα, καθώς επίσης μπορεί να έχει μέγεθος 0 (κενή περιγραφή).

Στον Πίνακα 2.3 φαίνεται η δομή μιας ομάδας παραμέτρων.

Πίνακας 2.3. Δομή Ομάδας Παραμέτρων.

Byte	Μήκος (bytes)	Περιγραφή
1	1	Ο αριθμός των χαρακτήρων του ονόματος της Ομάδας (1-127) – η τιμή αυτή μπορεί να είναι και αρνητικός αριθμός, το οποίο υποδεικνύει πως η ομάδα είναι «κλειδωμένη».
2	1	Κωδικός αριθμός της ομάδας (-1 έως -127 ...πάντα αρνητικός).
3	N	Όνομα ομάδας (χαρακτήρες ASCII – κεφαλαία αγγλικά A-Z, 0-9 και κάτω παύλα _ μόνο).
3+N	2	Ένας προσημασμένος αριθμός που αντιστοιχεί στην αρχή της επόμενης ομάδας/παραμέτρου.
3+N+2	1	Ο αριθμός των χαρακτήρων της περιγραφής της Ομάδας.
3+N+3	M	Περιγραφή Ομάδας (χαρακτήρες ASCII – επιτρέπονται όλοι)

2.4.2.γ’ Παράμετροι: Το πρώτο byte των παραμέτρων αντιστοιχεί στον αριθμό των χαρακτήρων του ονόματος της παραμέτρου. Τα ονόματα των παραμέτρων ακολουθούν τους ίδιους κανόνες με τα ονόματα των ομάδων. Συνεπώς, όταν ο αριθμός των χαρακτήρων είναι αρνητικός η παράμετρος είναι κλειδωμένη.

Ο μηχανισμός κλειδώματος δημιουργήθηκε, ώστε να αποτραπεί από τους συνηθισμένους χρήστες η δυνατότητα αλλαγών στις παραμέτρους, χρησιμοποιώντας ειδικά λογισμικά (όπως το PRM και το C3Deditor). Το γεγονός ότι ένα πρόγραμμα αντιλαμβάνεται μία παράμετρο ως κλειδωμένη, δε σημαίνει πως όλα τα λογισμικά αντιλαμβάνονται την παράμετρο ως κλειδωμένη. Ωστόσο συνιστάται τα λογισμικά να μην αφήνουν τους απλούς χρήστες να πειράζουν παραμέτρους που χρησιμοποιούν το σήμα της κλειδωμένης παραμέτρου, διότι μπορεί να καταστρέψουν το αρχείο C3D.

Το επόμενο byte περιέχει ένα κωδικό αριθμό, ο οποίος είναι πάντα θετικός και μεταξύ +1 έως +127 και συνδέει την παράμετρο με την αντίστοιχη ομάδα. Η παράμετρος με μία θετική κωδική τιμή αντιστοιχεί στην ομάδα με ίδια κατά απόλυτη τιμή.

Η ροή που περιέχει το όνομα της παραμέτρου αρχίζει στο τρίτο byte και τελειώνει τόσα N-bytes μετά – όσα ορίζει ο αριθμός των χαρακτήρων. Το όνομα της παραμέτρου υπακούει στους ίδιους κανόνες με το όνομα της ομάδας.

Στη συνέχεια ακολουθεί μία λέξη που δείχνει την αρχή της επόμενης παραμέτρου ή ομάδας και αμέσως μετά ακολουθεί ένα byte το οποίο περιγράφει τον τύπο της παραμέτρου. Μία παράμετρος μπορεί να είναι τύπου χαρακτήρων, ενός byte, ακέραιος αριθμός και δεκαδικός αριθμός (character, byte, integer or floating-point). Εδώ πρέπει να σημειωθεί πως οι δεκαδικοί αριθμοί μπορούν να αποθηκευτούν στο αρχείο με έναν από τους τρεις τύπους επεξεργαστών που αναφέρθηκαν στην κεφαλίδα των παραμέτρων (Intel, DEC and MIPS). Το ίδιο συμβαίνει και για τους προσημασμένους ακέραιους, που μπορούν να αποθηκευτούν με δύο διαφορετικούς τρόπους:

1. **Μικροί ενδιάνοι** (little endian) για επεξεργαστές τύπου DEC/Intel.
2. **Μεγάλοι ενδιάνοι** (big endian) για επεξεργαστές τύπου MIPS.

Παραδοσιακά όλοι οι ακέραιοι στην ενότητα των παραμέτρων είναι προσημασμένοι μεταξύ των τιμών -32768 έως +32767 και όλα τα bytes είναι προσημασμένοι ακέραιοι μεταξύ -128 έως +127.

Το επόμενο byte στις παραμέτρους είναι ο αριθμός των διαστάσεων, ο οποίος μπορεί να λάβει τιμές από 0 (καμία διάσταση) έως το μέγιστο 7 διαστάσεις. Η παράμετρος με μηδέν διαστάσεις είναι κλιμακωτή. Εάν η παράμετρος έχει διαστάσεις (είναι πίνακας), τότε η πραγματική διάσταση αποθηκεύεται στα επόμενα δύο bytes και αμέσως μετά ακολουθούν οι παράμετροι.

Μετά τις τιμές των παραμέτρων ακολουθεί ένα byte που δείχνει το μέγεθος της περιγραφής κι έπειτα ακολουθεί η περιγραφή, η οποία υπακούει στους ίδιους κανόνες με την περιγραφή των ομάδων.

Στον πίνακα του σχήματος 2.4 φαίνεται συνοπτικά η δομή μιας παραμέτρου.

Πίνακας 2.4. Δομή Παραμέτρου.

Byte	Μήκος (bytes)	Περιγραφή
1	1	Ο αριθμός των χαρακτήρων του ονόματος της Παραμέτρου (1-127) – η τιμή αυτή μπορεί να είναι και αρνητικός αριθμός, το οποίο υποδεικνύει πως η παράμετρος είναι «κλειδωμένη».
2	1	Κωδικός αριθμός της ομάδας (+1 έως +127 ...πάντα θετικός).
3	N	Όνομα παραμέτρου (χαρακτήρες ASCII – κεφαλαία αγγλικά A-Z, 0-9 και κάτω παύλα _ μόνο).
3+N	2	Ένας προσημασμένος αριθμός που αντιστοιχεί στην αρχή της επόμενης ομάδας/παραμέτρου.
3+N+2	1	Μήκος σε bytes για κάθε στοιχείο: -1 για δεδομένα χαρακτήρων, 1 για δεδομένα byte, 2 για δεδομένα ακέραιων (16bit integer) 4 για δεδομένα δεκαδικών (floating-point – REAL data)
3+N+3	1	Αριθμός των διαστάσεων (0-7) των παραμέτρων – 0 εάν η παράμετρος είναι κλιμακωτή.
3+N+4	D	Διαστάσεις παραμέτρων.
3+N+4+D	T	Δεδομένα παραμέτρων.
3+N+4+D+T	1	Ο αριθμός των χαρακτήρων της περιγραφής της Παραμέτρου.
3+N+4+D+T+1	M	Περιγραφή Παραμέτρου.

2.4.2.δ’ Παραδείγματα Δομών Παραμέτρων: Όπως αναφέρθηκε παραπάνω, οι παράμετροι χωρίζονται σε δομές, οι οποίες αποτελούνται από ομάδες, μέσα στις οποίες υπάρχουν διάφορες παράμετροι. Δύο από τις απαραίτητες και σημαντικότερες δομές - οι οποίες χρησιμοποιήθηκαν και κατά την εκπόνηση του λογισμικού της παρούσας διπλωματικής - είναι οι εξής:

- **TRIAL:** Στην κεφαλίδα του C3D αρχείου περιέχονται το πρώτο και το τελευταίο καρέ, από τα οποία υπολογίζεται το πλήθος των καρέ:

$$\text{Frame}_{\text{size}} = \text{Last}_{\text{frame}} - \text{First}_{\text{frame}}$$

Το αποτέλεσμα της παραπάνω πράξης, που αντιστοιχεί στο συνολικό αριθμό των καρέ, βρίσκεται αποθηκευμένο στην παράμετρο POINT::FRAME. Επιπλέον η μορφοποίηση C3D προσδιορίζει αυτές τις τιμές ως μη προσημασμένες ακέραιες τιμές, κάτι το οποίο μεταβάλλει το εύρος τιμών σε 0 έως 65635.

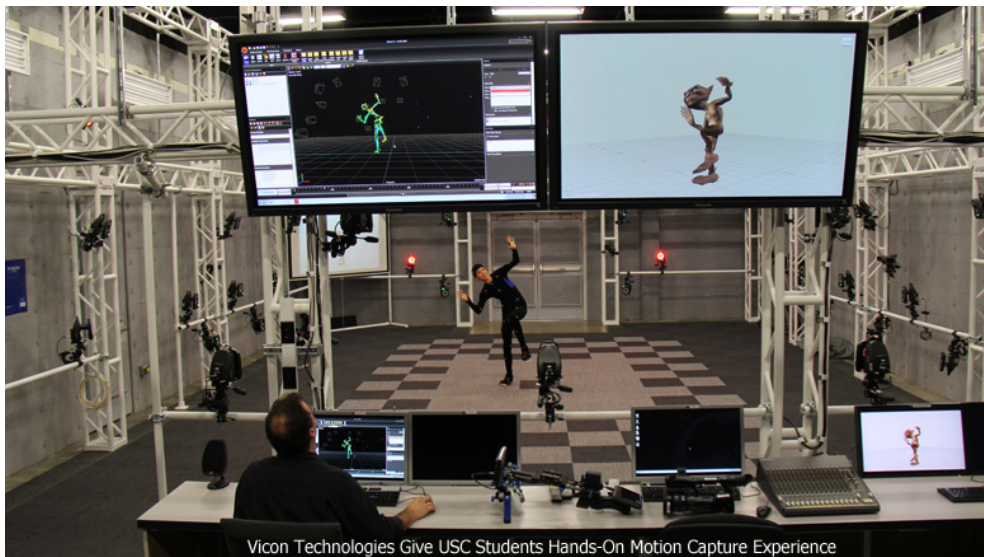
Όταν χρησιμοποιούνται υψηλής ταχύτητας κάμερες βίντεο, αυτό μεταφράζεται σε μία δοκιμή η οποία ξεπερνά τα δύο λεπτά στα 240Hz. Για να ξεπεραστεί το όριο των 16-bit ακέραιας τιμής και να επιτραπεί η χρήση περισσότερων από 65535 καρέ, έχουν εισαχθεί δύο ειδικές παράμετροι, οι οποίες ονομάζονται ACTUAL_START_FIELD και ACTUAL_END_FIELD και μπορούν να αποθηκεύσουν τιμές 32-bit. Για το λόγο αυτό οι εφαρμογές που διαβάζουν C3D αρχεία, πρέπει να χρησιμοποιούν αυτές τις τιμές για την ανάγνωση των δεδομένων και όταν αυτές οι τιμές δεν είναι διαθέσιμες, τότε να χρησιμοποιούν τις αντίστοιχες τιμές της κεφαλίδας ή την τιμή POINT::FRAMES.

- **POINT:** Η ομάδα αυτή παρέχει πληροφορίες σχετικά με τα τριδιάστατα δείγματα δεδομένων, τα οποία περιέχονται μέσα στο αρχείο C3D, καθώς επίσης και βασικές πληροφορίες σχετικά με το περιβάλλον των δεδομένων. Ως αποτέλεσμα κάποιες παράμετροι της ομάδας POINT, όπως είναι

οι POINT::DATA_START, POINT::FRAMES και POINT::USED απαιτούνται ακόμα και όταν το αρχείο C3D περιέχει μόνο αναλογική πληροφορία χωρίς τριδιάστατη πληροφορία. Η παράμετρος POINT::DATA_START χρειάζεται ώστε να παρέχεται πληροφορία σχετικά με την αρχή των τριδιάστατων δεδομένων και της αναλογικής πληροφορίας που είναι αποθηκευμένη μέσα στο αρχείο C3D. Η παράμετρος POINT::USED ενεργοποιεί την εφαρμογή ώστε να καθορίσει τον αριθμό των τριδιάστατων σημείων που έχουν καταγραφεί στην περιοχή των δεδομένων – η τιμή αυτή πρέπει να ορίζεται μηδενική, όταν ο αποθηκευτικός χώρος των τριδιάστατων σημείων και της αναλογικής πληροφορίας, δεν περιέχει κανένα δεδομένο τριδιάστατων σημείων.

2.4.3 Ενότητα Δεδομένων

Το αρχείο μορφοποίησης C3D είναι σχεδιασμένο να αποθηκεύει τριδιάστατη και αναλογική πληροφορία, έτσι ώστε οι τριδιάστατες τοποθεσίες ενός πλήθους μαρκαρισμένων σημείων (Σχήμα 2.1), να μπορούν να συγχρονιστούν με αναλογικές μετρήσεις που πραγματοποιήθηκαν την ίδια στιγμή. Για αυτό το λόγο τα τριδιάστατα και αναλογικά δείγματα αλληλοσυνδέονται καρέ-καρέ, σε όλα τα δεδομένα. Ως αποτέλεσμα η καταγραφή των C3D δεδομένων είναι ευέλικτη και μπορεί να χρησιμοποιηθεί για να δημιουργηθούν αρχεία που θα περιέχουν είτε μόνο τριδιάστατα δεδομένα, είτε διδιάστατα δεδομένα, είτε αναλογικά δεδομένα, είτε κάθε συνδυασμό των παραπάνω. Επιπλέον είναι δυνατό (αν και όχι τόσο λειτουργικό) να αποθηκεύονται αποτελέσματα κινηματικών υπολογισμών (γωνίες, στιγμές, επιταχύνσεις κλπ) στην μορφοποίηση καταγραφής των τριδιάστατων δεδομένων.



Σχήμα 2.1. Καταγραφή στόχων-μαρκών με σύστημα Vicon.

Αν και η μορφοποίηση C3D έχει σχεδιαστεί για την καταγραφή τριδιάστατης χωρικής πληροφορίας, μπορεί να καταγραφεί και διδιάστατη πληροφορία, ορίζοντας κάποια από τις συντεταγμένες σταθερή και υπολογίζοντας τις άλλες δύο από τη θέση του παρατηρητή. Αυτό επιτρέπει στα αρχεία C3D να χρησιμοποιούνται από συστήματα που υποστηρίζουν μετρήσεις από μία κάμερα - επομένως ένας παρατηρητής (κάμερα) μπορεί να παρέχει χωρική πληροφορία για τα επίπεδα Y και Z, περιορίζοντας την κίνηση κατά του άξονα X πάνω σε ένα σταθερό επίπεδο.

Τα τριδιάστατα και αναλογικά δείγματα δεδομένων καταγράφονται σε διαδοχικά καρέ, ξεκινώντας από την αρχή της πρώτης δομής των 512-byte που καθορίζεται από την παράμετρο POINT::DATA_START. Εάν στο αρχείο C3D περιέχεται τριδιάστατη και αναλογική πληροφορία, τότε το καρέ των τριδιάστατων σημείων αποθηκεύεται πάντα πρώτο, αρχίζοντας από το πρώτο καρέ των δεδομένων. Εάν υπάρχει μόνο ένας τύπος δεδομένων, τότε η ενότητα των δεδομένων θα περιέχει διαδοχικά καρέ από δείγματα δεδομένων.

Η πρότυπη περιγραφή της μορφοποίησης των αρχείων C3D αναφέρει πως η παράμετρος που δείχνει την αρχή των δεδομένων – POINT::DATA_START – αποθηκεύεται ως προσημασμένος αχέραιος των

16-bit. Αυτό περιορίζει την αποθηκευτική χωρητικότητα της τρισδιάστατης και αναλογικής πληροφορίας σε μέγιστο όριο χωρητικότητας των 16Mb (32767*512/1024). Αντιμετωπίζοντας αυτή την παράμετρο ως μη προσημασμένο ακέραιο (το οποίο μπορεί εύκολα να εντοπιστεί καθώς θα έχει αρνητική τιμή, στην περίπτωση που η τιμή της είναι μεγαλύτερη από 32767 όταν διαβάζεται ως προσημασμένος ακέραιος), το μέγιστο όριο χωρητικότητας της τρισδιάστατης και αναλογικής πληροφορίας μπορεί να επεκταθεί στα πρώτα 32Mb της ενότητας των δεδομένων.

Τα τρισδιάστατα και αναλογικά δείγματα δεδομένων, μπορούν να αποθηκευτούν είτε ως προσημασμένοι ακέραιοι, είτε ως δεκαδικοί αριθμοί. Ο τρόπος αποθήκευσης που θα επιλεγεί, αναφέρεται και στα τρισδιάστατα και στα αναλογικά δεδομένα που υπάρχουν αποθηκευμένα στο ίδιο C3D αρχείο. Αυτό σημαίνει ότι δεν υπάρχει τρόπος να υπάρξει πάνω από ένας τύπος αποθήκευσης των δεδομένων, μιας και υπάρχει μόνο ένα όρισμα, το οποίο αναφέρεται στον τύπο αποθήκευσης των δεδομένων.

Το εύρος της ενότητας των δεδομένων είναι γνωστό από την ενότητα των παραμέτρων (αριθμός καρτέ, αριθμός τρισδιάστατων σημείων και αριθμός αναλογικών σημείων) κι έτσι δε χρειάζεται να υπάρξει κάποιο όρισμα που να αναφέρεται στο τέλος των δεδομένων - τα δεδομένα απλώς διαβάζονται από το πρώτο καρτέ έως το τελευταίο καρτέ. Κάθε περισευούμενο byte έως ότου καλυφθεί η τελευταία δομή των 512-bytes, θα πρέπει να καλύπτεται με 0x00 για συμβατότητα με παλαιότερα λογισμικά που βασίζονται πάνω στις αρχές της FORTRAN. Τόσο τα τρισδιάστατα, όσο και τα αναλογικά δείγματα δεδομένων μπορούν να ξεπεράσουν τα όρια της δομής των 512-byte εντός του C3D αρχείου.

Στο Σχήμα 2.2 φαίνεται η γενική οργάνωση της ενότητας των δεδομένων, όπως αυτή περιγράφεται παραπάνω.

3D δεδομένα του πρώτου καρτέ
Αναλογικά δεδομένα του πρώτου καρτέ
3D δεδομένα του δεύτερου καρτέ
Αναλογικά δεδομένα του δεύτερου καρτέ
...
...
3D δεδομένα του νιοστού καρτέ
Αναλογικά δεδομένα του νιοστού καρτέ

Σχήμα 2.2. Δομή οργάνωσης των δεδομένων.

Σε αυτό το σημείο πρέπει να διευκρινιστεί ότι η σειρά με την οποία αποθηκεύονται οι τιμές των τρισδιάστατων δεδομένων, σε ένα αρχείο C3D, δεν είναι προκαθορισμένη, αλλά είναι γνωστό ότι αποθηκεύονται στην ενότητα των παραμέτρων, με τη σειρά περιγραφής τους. Έτσι οι εφαρμογές, οι οποίες έχουν πρόσβαση στα τρισδιάστατα δεδομένα, προκειμένου να καθορίσουν τη σειρά με την οποία έχουν αποθηκευτεί τα δεδομένα, πρέπει να είναι σε θέση να διαβάζουν και να κατανοούν τις ετικέτες των σημείων στην ενότητα των παραμέτρων. Εφαρμογές οι οποίες θεωρούν πως τα τρισδιάστατα σημεία θα αποθηκεύονται πάντα με την ίδια σειρά, θα εμφανίζουν προβλήματα όταν θα προβάλλουν τα ίδια σημεία (από άλλο αρχείο), τα οποία έχουν αποθηκευτεί με άλλη σειρά. Τα αναλογικά δείγματα προβάλλονται για κάθε τρισδιάστατο καρτέ και καταγράφονται διαδοχικά - κάθε τρισδιάστατο καρτέ ακολουθείται από αναλογικά δεδομένα τα οποία σχετίζονται με το συγκεκριμένο τρισδιάστατο καρτέ.

Παρακάτω περιγράφονται συνοπτικά οι τρόποι αποθήκευσης και ανάγνωσης, των τρισδιάστατων δειγμάτων που βρίσκονται αποθηκευμένα στην ενότητα των δεδομένων. Για τα αναλογικά δείγματα δε θα δοθεί περισσότερη έμφαση, εφόσον το λογισμικό που αναπτύχθηκε για την εκπόνηση της διπλωματικής δε διαχειρίζεται την αναλογική πληροφορία.

2.4.3.α' Τρισδιάστατα Δεδομένα - Ακέραιη Μορφοποίηση: Εάν η τιμή της παραμέτρου POINT::SCALE είναι θετική, τότε η μορφοποίηση αποθήκευσης των τρισδιάστατων δεδομένων θα είναι προσημασμένη ακέραιη (16-bit signed integer). Σημειώνεται ότι η τιμή της παραμέτρου POINT::SCALE μπορεί εκτός από την ενότητα των παραμέτρων να εντοπιστεί και στην κεφαλίδα του αρχείου (λέξεις 7-8). Στον Πίνακα 2.5 φαίνεται η δομή οργάνωσης με την οποία αποθηκεύονται τα τρισδιάστατα δεδομένα ακέραιας μορφής:

Οι πρώτες τρεις προσημασμένες ακέραιες λέξεις, αντιστοιχούν στις συντεταγμένες των τρισδιάστατων δεδομένων X, Y και Z , οι οποίες έχουν διαιρεθεί με την παράμετρο POINT::SCALE.

Πίνακας 2.5. Δομή δεδομένων ακέραιας μορφής.

Λέξη (16-bit)	Περιεχόμενο (προσημασμένος ακέραιος)
1	Συντεταγμένη X του σημείου διαιρεμένη με το συντελεστή POINT:SCALE.
2	Συντεταγμένη Y του σημείου διαιρεμένη με το συντελεστή POINT:SCALE.
3	Συντεταγμένη Z του σημείου διαιρεμένη με το συντελεστή POINT:SCALE.
4	Byte 1: Οι κάμερες που μετρούν το σημείο (1-bit για κάθε κάμερα) Byte 2: Μέσο σφάλμα διαιρεμένο με το συντελεστή POINT:SCALE.

Η τέταρτη λέξη των τρισδιάστατων ακέραιων δεδομένων αντιστοιχεί σε δύο bytes. Το πρώτο byte καταγράφει ποιοι παρατηρητές (συνήθως κάμερες) παρέχουν πληροφορίες, οι οποίες έχουν χρησιμοποιηθεί για τον υπολογισμό του τρισδιάστατου σημείου, καθώς το δεύτερο byte περιέχει το μέσο σφάλμα της καταγραφής του τρισδιάστατου σημείου. Το μέσο σφάλμα στην πραγματικότητα είναι η σχετική ακρίβεια της τρισδιάστατης μέτρησης και η διαβαζόμενη τιμή πρέπει να πολλαπλασιάζεται με την παράμετρο POINT:SCALE, ώστε να αναχθεί στην πραγματική κλίμακα.

2.4.3.β' Τρισδιάστατα Δεδομένα - Δεκαδική Μορφοποίηση: Εάν η παράμετρος της κλίμακας (POINT:SCALE) είναι αρνητική, τότε τα τρισδιάστατα δεδομένα έχουν αποθηκευτεί υπό κλίμακα σε δεκαδική μορφή. Αυτού του είδους η μορφοποίηση είναι μεγαλύτερης ακρίβειας και από τη στιγμή που τα δεδομένα είναι ήδη υπό κλίμακα, δε χρειάζεται να πολλαπλασιαστούν με τον συντελεστή κλίμακας (παράμετρος POINT:SCALE). Ωστόσο το μειονέκτημα αυτής της αποθηκευτικής μορφής είναι ότι απαιτούνται 8 λέξεις των 16-bit, συνεπώς απαιτείται διπλάσια χωρητικότητα από ότι εάν αποθηκεύονταν σε ακέραιη μορφή. Τέλος η τιμή POINT:SCALE χρησιμοποιείται μόνο για τον υπολογισμό της τιμής του μέσου σφάλματος της κάθε τρισδιάστατης μέτρησης.

Ο Πίνακας 2.6 περιγράφει τη δομή οργάνωσης με την οποία αποθηκεύονται τα τρισδιάστατα δεδομένα δεκαδικής μορφής:

Πίνακας 2.6. Δομή δεδομένων δεκαδικής μορφής.

Λέξη (16-bit)	Περιεχόμενο (δεκαδικός αριθμός)
1-2	Πραγματική συντεταγμένη X του σημείου.
3-4	Πραγματική συντεταγμένη Z του σημείου.
5-6	Πραγματική συντεταγμένη Z του σημείου.
7-8	Αφότου μετατραπεί πρώτα σε προσημασμένο ακέραιο: Bytes 1-2: Οι κάμερες που μετρούν το σημείο (1-bit για κάθε κάμερα) Bytes 3-4: Μέσο σφάλμα διαιρεμένο με το συντελεστή POINT:SCALE.

Οι πρώτες τρεις δεκαδικές λέξεις (32-bit η κάθε δεκαδική λέξη) καταγράφουν τις υπό κλίμακα συντεταγμένες X, Y και Z των τρισδιάστατων σημείων. Η τέταρτη δεκαδική τιμή πρέπει πρώτα να μετατραπεί σε προσημασμένο ακέραιο και στη συνέχεια να ερμηνευτεί ως δύο ξεχωριστά bytes. Το πρώτο byte καταγράφει τους παρατηρητές (συνήθως κάμερες) οι οποίοι παρέχουν πληροφορίες σχετικά με τις κάμερες που χρησιμοποιήθηκαν για τον υπολογισμό του τρισδιάστατου σημείου, ενώ το δεύτερο byte περιέχει το μέσο σφάλμα της τρισδιάστατης μέτρησης. Το σφάλμα του τρισδιάστατου σημείου είναι μία εκτίμηση, η οποία παρέχει πληροφορία για τη σχετική ακρίβεια της τρισδιάστατης μέτρησης.

Κεφάλαιο 3

Σύντομη Περιγραφή της OpenGL

3.1 Τι είναι η OpenGL

Η OpenGL είναι μία διεπαφή προγραμματισμού εφαρμογών (application programming interface - API) ή πιο απλά είναι μία βιβλιοθήκη εφαρμογών για πρόσβαση σε λειτουργίες που σχετίζονται με γραφικά υπολογιστών. Η έκδοση 4.3 της OpenGL περιέχει πάνω από 500 διακριτές εντολές, οι οποίες χρησιμοποιούνται για τον καθορισμό αντικειμένων, εικόνων και διαδικασιών που απαιτούνται για την παραγωγή διαδραστικών τριδιάστατων γραφικών εφαρμογών. Η ΟΠΕΓΛ χρησιμοποιείται στην παραγωγή 4Δ μοντέλων όπως σε [13], [19], [6], [7].

Η OpenGL έχει σχεδιαστεί ως μία βελτιωμένη διασύνδεση, ανεξάρτητα από το υλικό (hardware independent), η οποία μπορεί να εφαρμοστεί σε μεγάλο αριθμό διαφορετικών τύπων υλικών συστημάτων γραφικών ή εξ ολοκλήρου σε λογισμικό (εάν δεν υπάρχει κάποιο υλικό γραφικών - graphic hardware - στο σύστημα) ανεξάρτητα από το λειτουργικό σύστημα του υπολογιστή ή το παράθυρο του λογισμικού. Συνεπώς, η OpenGL δεν περιλαμβάνει λειτουργίες εκτέλεσης εργασιών παραθύρου ή επεξεργασίας εισόδου χρήστη, αλλά αντιθέτως, η εφαρμογή θα πρέπει να είναι σε θέση να χρησιμοποιεί τις λειτουργίες που παρέχονται κάθε φορά από το σύστημα παραθύρων, στο οποίο θα εκτελεστεί. Ομοίως, η OpenGL δεν παρέχει καμία λειτουργία για την περιγραφή μοντέλων τριδιάστατων αντικειμένων ή λειτουργιών για την ανάγνωση αρχείων εικόνας (όπως αρχείων JPEG). Η δημιουργία ενός τριδιάστατου αντικειμένου προϋποθέτει την καλή περιγραφή του, χρησιμοποιώντας απλές γεωμετρικές δομές, όπως σημεία, γραμμές, τρίγωνα και κηλίδες.

Η βιβλιοθήκη OpenGL αναπτύχθηκε αρχικά από τη Silicon Graphics Computer Systems και η πρώτη έκδοσή της δημοσιεύθηκε τον Ιούλιο του 1994. Από τότε αναπτύχθηκαν πολλές εκδοχές της OpenGL, καθώς επίσης και πολλές βιβλιοθήκες εφαρμογών δημιουργήθηκαν βασισμένες στην OpenGL απλοποιώντας την ανάπτυξη εφαρμογών, είτε για τη δημιουργία βίντεο παιχνιδιών, είτε για τη δημιουργία οπτικοποίησης για επιστημονικούς ή ιατρικούς σκοπούς, είτε απλά για την προβολή εικόνων. Ωστόσο οι πιο μοντέρνες εκδοχές της OpenGL διαφέρουν σημαντικά από την αρχική έκδοσή.

Στην παρακάτω λίστα περιγράφονται οι σημαντικότερες λειτουργίες, τις οποίες μία εφαρμογή που χρησιμοποιεί τη βιβλιοθήκη OpenGL πραγματοποιεί για την απόδοση (προβολή) μίας εικόνας:

1. Καθορισμός των δεδομένων για την κατασκευή σχημάτων από τα γεωμετρικά θεμελιώδη στοιχεία της OpenGL.
2. Εκτέλεση διαφόρων shaders για την πραγματοποίηση υπολογισμών της θέσης, του χρώματος και άλλων στοιχείων οπτικοποίησης, κατά την εισαγωγή των θεμελιωδών στοιχείων.
3. Μετατροπή των μαθηματικών περιγραφών που εισήχθησαν κατά την εισαγωγή των θεμελιωδών στοιχείων σε στοιχεία (fragments) τα οποία σχετίζονται με την τοποθεσία τους στην οθόνη. Η διαδικασία αυτή καλείται rasterization.
4. Τελικώς, εκτελείται ένας fragment shader για κάθε ένα από τα στοιχεία (fragments) που δημιουργήθηκαν κατά τη διαδικασία rasterization και προσδιορισμό του τελικού χρώματος και θέσης του κάθε στοιχείου.
5. Πιθανώς να πραγματοποιούνται και επιπλέον διαδικασίες πριν τη δημιουργία των στοιχείων, όπως ο καθορισμός για το αν ένα στοιχείο είναι ορατό ή όχι.

Η βιβλιοθήκη OpenGL υλοποιήθηκε ως σύστημα διακομιστή - πελάτη (client - server system), με την εφαρμογή να έχει το ρόλο του πελάτη (client) και την υλοποίηση της OpenGL που παρέχεται από τον κατασκευαστή του γραφικού υλικού (computer graphic hardware) να έχει το ρόλο του διακομιστή (server). Σε ορισμένες περιπτώσεις η υλοποίηση της παραπάνω σχέσης, της βιβλιοθήκης OpenGL, εκτελείται σε διαφορετικά μηχανήματα, τα οποία είναι συνδεδεμένα σε ένα δίκτυο. Σε αυτές τις περιπτώσεις ο πελάτης είναι αυτός που ζητά την εκτέλεση μίας εντολής της OpenGL, η οποία θα μετατραπεί σε ειδικό πρωτόκολλο συστήματος, το οποίο θα μεταδοθεί στον διακομιστή - εξυπηρετητή (server), μέσω του κοινόχρηστου δικτύου τους, όπου και θα εκτελεστεί για να παραχθεί η τελική εικόνα [3].

3.2 Σύνταξη των συναρτήσεων της OpenGL

Οι εντολές της βιβλιοθήκης OpenGL έχουν συνταχθεί με μία συγκεκριμένη τυποποίηση. Για παράδειγμα η συνάρτηση με την οποία δημιουργείται ένα σημείο έχει όνομα `glVertex3f(...)` ή `glVertex2i(...)`. Παρατηρώντας τις παραπάνω συναρτήσεις, μπορεί εύκολα κάποιος να διακρίνει τα εξής χαρακτηριστικά:

1. Κάθε εντολή της βιβλιοθήκης OpenGL αρχίζει με την προσφώνηση `gl`. Αυτό συμβαίνει, επειδή υπάρχουν και άλλες βοηθητικές βιβλιοθήκες συμβατές με την OpenGL, οι οποίες έχουν άλλες προσφωνήσεις, όπως π.χ. `glut` για τη βιβλιοθήκη OpenGL Utility Toolkit.
2. Το δεύτερο τμήμα της εντολής υποδεικνύει το τι κάνει η εντολή. Στα παραπάνω παραδείγματα το τμήμα αυτό αποτελεί η λέξη `Vertex` και δημιουργεί ένα κόμβο – σημείο. Αντίστοιχα η εντολή προσδιορισμού χρώματος, χρησιμοποιεί τη λέξη `Color`.
3. Το τρίτο τμήμα κάθε εντολής, εάν υπάρχει, είναι το πλήθος των ορισμάτων της εντολής. Για παράδειγμα η εντολή `glVertex3f(...)` δέχεται τρία ορίσματα, ενώ η εντολή `glVertex2i(...)` δέχεται δύο ορίσματα.
4. Τέλος ο αριθμός ακολουθείται από τουλάχιστον ένα γράμμα, το οποίο προσδιορίζει τον τύπο των δεδομένων που δέχεται (π.χ. `f` για floating point μορφοποίηση)

Στον Πίνακα 3.1 φαίνονται οι διάφοροι τύποι μορφοποιήσεων που μπορεί να δέχεται μία εντολή (συνάρτηση) και η εισαγωγή τους στις συναρτήσεις.

Πίνακας 3.1. Τύποι μορφοποιήσεων της OpenGL.

Κατάληξη	Τύπος Δεδομένων	Μορφοποίηση στη C	Ορισμός μορφοποίησης στην OpenGL
b	8-bit ακέραιος	signed char	GLbyte
s	16-bit ακέραιος	short	GLshort
i	32-bit ακέραιος	int ή long	GLint, GLsize
f	32-bit δεκαδικός	float	GLfloat, GLclampf
d	64-bit δεκαδικός	double	GLdouble, GLclampd
ub	8-bit μη προσημασμένος ακέραιος	unsigned char	GLubyte, GLboolean
us	16-bit μη προσημασμένος ακέραιος	unsigned short	GLushort
ui	32-bit μη προσημασμένος ακέραιος	unsigned int ή unsigned long	GLuint, GLenum, GLbitfield

Τέλος κάποιες από τις εντολές μπορεί να τελειώνουν με το γράμμα `v`, το οποίο υποδεικνύει πως αντί για μια σειρά ορισμάτων, απαιτείται ένας μόνο πίνακας μεγέθους όσο ο αριθμός των ορισμάτων και αντίστοιχης μορφοποίησης. Για παράδειγμα η εντολή `glColor3fv(...)`, δέχεται σαν όρισμα έναν πίνακα τριών διαστάσεων δεκαδικής μορφοποίησης (`GLfloat array`) [1].

3.3 Χρήση της OpenGL στον κώδικα

Για την οπτικοποίηση του νέφους σημείων των δεδομένων, χρησιμοποιήθηκαν συναρτήσεις τόσο της βιβλιοθήκης OpenGL, όσο και της βοηθητικής βιβλιοθήκης GLU. Παρακάτω περιγράφονται εν συντομία οι συναρτήσεις που χρησιμοποιήθηκαν:

3.3.1 `glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)`

Καθορίζει το χρώμα καθαρισμού της οθόνης.

Οι παράμετροι της συνάρτησης είναι το ποσοστό των τιμών συμμετοχής του κόκκινου, πράσινου, μπλέ και αδιαφάνειας στο τελικό χρώμα καθαρισμού. Το εύρος τιμών της συνάρτησης είναι [0,1], ενώ το αποτέλεσμα της συνάρτησης γίνεται αντιληπτό κάθε φορά που καλείται η συνάρτηση `glClear(...)` [17][15][16].

3.3.2 `glClear(GLbitfield mask)`

Καθαρισμός των buffers χρησιμοποιώντας τις οριζόμενες τιμές.

Τα έγκυρα ορίσματα που δέχεται αυτή η συνάρτηση είναι:

- Bitwise OR
- `GL_COLOR_BUFFER_BIT`: Υποδηλώνει το color buffer.
- `GL_DEPTH_BUFFER_BIT`: Υποδηλώνει το depth buffer.
- `GL_STENCIL_BUFFER_BIT`: Υποδηλώνει το stencil buffer.

Η συνάρτηση `glClear` θέτει την περιοχή bitplane του παραθύρου με τις τιμές που έχουν επιλεγεί προηγουμένως χρησιμοποιώντας τις εντολές `glClearColor`, `glClearDepth` και `glClearStencil` [17][15][16].

3.3.3 `glPolygonMode(GLenum face, GLenum mode)`

Επιλογή μεθόδου ραστεροποίησης πολυγώνου [15][16].

Οι παράμετροι της συνάρτησης είναι:

- `face` – Καθορίζει τα πολύγωνα στα οποία αναφέρεται η μέθοδος. Όταν αναφέρεται στο μπροστά και πίσω μέρος των πολυγώνων, το όρισμα είναι `GL_FRONT_AND_BACK`.
- `mode` – Καθορίζει τη μέθοδο ραστεροποίησης των πολυγώνων. Αποδεκτές τιμές είναι οι `GL_POINT`, `GL_LINE` και `GL_FILL`. Η προκαθορισμένη τιμή είναι `GL_FILL` για το μπροστά και πίσω μέρος των πολυγώνων.

3.3.4 `glMatrixMode(GLenum mode)`

Ορίζει ποιος πίνακας θα θεωρείται ως προκαθορισμένος από εκείνη τη στιγμή κι έπειτα.

Η μοναδική παράμετρος που δέχεται είναι η `mode` - ορίζει ποια στοίβα πίνακα αποτελεί στόχο για τις επόμενες λειτουργίες πινάκων. Οι αποδεκτές τιμές είναι τρεις: `GL_MODELVIEW`, `GL_PROJECTION` και `GL_TEXTURE`. Η προκαθορισμένη τιμή είναι η `GL_MODELVIEW`. Επιπλέον εάν υποστηρίζεται η επέκταση `ARB_imaging`, τότε είναι αποδεκτή και μία τέταρτη τιμή με όνομα `GL_COLOR` [17][15][16].

3.3.5 `glEnable(GLenum cap)`

Ενεργοποιεί ή απενεργοποιεί λειτουργίες της GL από τη μεριά του εξυπηρετητή (server)

Δέχεται την παράμετρο `cap` - Ορίζει μία συμβολική σταθερά η οποία υποδεικνύει κάποια λειτουργία της GL [17][15][16].

3.3.6 `glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`

Ορίζει το παράθυρο θέασης [17][15][16].

Οι παράμετροι της συνάρτησης θέασης είναι οι εξής:

- `x,y` – Καθορίζει την ελάχιστη αριστερή γωνία του τετραγώνου θέασης, σε pixels. Η προκαθορισμένη τιμή είναι (0,0).
- `width, height` – Καθορίζει το πλάτος (`width`) και ύψος (`height`) του παραθύρου θέασης. Όταν ένα περιεχόμενο της GL συνδέεται σε ένα παράθυρο, τότε το πλάτος και το ύψος ορίζονται όσο είναι οι διαστάσεις του παραθύρου.

3.3.7 `glLoadIdentity()`

Αντικαθιστά τον υφιστάμενο πίνακα με τον πίνακα που έχει οριστεί μέσω κάποιας άλλης συνάρτησης (π.χ. `glMatrixMode(...)`) [17][15][16].

3.3.8 glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)

Πολλαπλασιάζει τον υφιστάμενο πίνακα με τον πίνακα στροφών [17][15][16].

Οι παράμετροι της συνάρτησης είναι οι εξής:

- angle – Ορίζει τη γωνία στροφής, σε μοίρες.
- x, y, z – Ορίζει τις συντεταγμένες x, y και z ενός διανύσματος, αντίστοιχα.

3.3.9 glPointSize(GLfloat size)

Καθορίζει τη διάμετρο του ραστεροποιημένου σημείου [17][15][16].

Η μοναδική παράμετρος είναι η size – Ορίζει τη διάμετρο του ραστεροποιημένου σημείου. Η προκαθορισμένη τιμή είναι 1.

3.3.10 glLineWidth(GLfloat width)

Καθορίζει το πλάτος της ραστεροποιημένης γραμμής [17][15][16].

Η μοναδική παράμετρος είναι η width – Ορίζει το πλάτος της ραστεροποιημένης γραμμής. Η προκαθορισμένη τιμή είναι 1.

3.3.11 glBegin(GLenum mode) και glEnd()

Οριοθετεί τις κορυφές ενός θεμελειακού στοιχείου ή μιας ομάδας αντίστοιχων θεμελειακών στοιχείων [17][15][16].

Η μοναδική παράμετρος είναι η mode – Ορίζει το(α) θεμελειακό(ά) στοιχείο(α) με τα οποία θα δημιουργηθούν οι κόμβοι που παρουσιάζονται μεταξύ των συναρτήσεων glBegin και του μεταγενέστερου glEnd. Δέκα συμβολικές σταθερές είναι αποδεκτές:

1. GL_POINTS
2. GL_LINES
3. GL_LINE_STRIP
4. GL_LINE_LOOP
5. GL_TRIANGLES
6. GL_TRIANGLE_STRIP
7. GL_TRIANGLE_FAN
8. GL_QUADS
9. GL_QUAD_STRIP
10. GL_POLYGON

3.3.12 glColor3f(GLfloat red, GLfloat green, GLfloat blue)

Η συνάρτηση αυτή δέχεται τρεις παραμέτρους (red, green, blue), οι οποίες καθορίζουν το νέο ποσοστό των τιμών του κόκκινου, του πράσινου και του μπλε για το νέο υφιστάμενο χρώμα με το οποίο θα σχεδιαστούν τα θεμελειακά στοιχεία [17][15][16].

3.3.13 glVertex3f(GLfloat x, GLfloat y, GLfloat z)

Η συνάρτηση αυτή δέχεται τρεις παραμέτρους (x, y, z), οι οποίες καθορίζουν τις συντεταγμένες x, y, z ενός κόμβου (κορυφής) [17][15][16].

3.3.14 glFlush()

Επιβάλλει την εκτέλεση των εντολών της GL σε τετελεσμένο χρόνο [17][15][16].

Κεφάλαιο 4

Ανάπτυξη των Κλάσεων - Συναρτήσεων

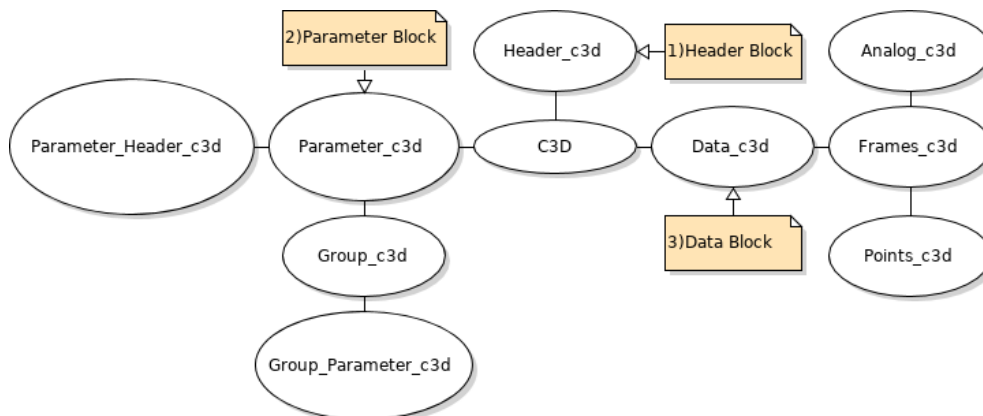
Για την οπτικοποίηση και επεξεργασία των δεδομένων κρίθηκε αναγκαία η δημιουργία κλάσεων και συναρτήσεων. Τόσο για την ανάπτυξη της πλατφόρμας οπτικοποίησης, όσο και για την υλοποίηση των αλγορίθμων επεξεργασίας των δεδομένων χρησιμοποιήθηκε η γλώσσα προγραμματισμού C/C++, ενώ για τη δημιουργία του γραφικού περιβάλλοντος χρησιμοποιήθηκε η πλατφόρμα QtCreator.

Για την καλύτερη περιγραφή των αλγορίθμων κρίνεται σκόπιμη η κατηγοριοποίηση των κλάσεων και συναρτήσεων σε δύο ομάδες, ανάλογα με τις λειτουργίες που επιτελούν.

1. **Ανάγνωση των δεδομένων:** Στην κατηγορία αυτή ανήκουν οι κλάσεις οι οποίες διαβάζουν ένα αρχείο C3D. Οι κλάσεις αυτές έχουν προσαρμοστεί στους σκοπούς της παρούσας διπλωματικής. Αυτό σημαίνει ότι μπορούν να διαβαστούν αρχεία C3D, τα οποία έχουν ληφθεί από σύστημα καμερών Vicon. Η ανάγνωση ενός αρχείου που έχει ληφθεί από άλλο σύστημα ή είναι παλαιότερης εποχής, ενδέχεται να έχει μη επιθυμητά αποτελέσματα.
2. **Επεξεργασία των δεδομένων:** Εάν κι εφόσον η ανάγνωση των δεδομένων ολοκληρωθεί με επιτυχία, τότε τα δεδομένα αντιγράφονται σε μία κλάση περιγραφής ενός νέφους σημείων. Συνεπώς η κατηγορία αυτή περιέχει αυτή την κλάση και τις επιμέρους βοηθητικές κλάσεις, καθώς επίσης και όλες τις συναρτήσεις – εντολές – με τις οποίες πραγματοποιείται η επεξεργασία των δεδομένων.

4.1 Ανάγνωση των Δεδομένων

Η ανάγνωση των δεδομένων, αποτελεί τη θεμελιώδη λίθο για την υλοποίηση της επεξεργασίας των δεδομένων. Στο Κεφάλαιο 2 έγινε περιγραφή της δομής του αρχείου C3D. Όπως αναφέρθηκε, η σωστή ανάγνωση και καταχώρηση των δεδομένων στην εφαρμογή, απαιτεί τη σωστή χρήση συγκεκριμένων παραμέτρων, από την ενότητα των παραμέτρων.



Σχήμα 4.1. Δομή οργάνωσης για την ανάγνωση του αρχείου C3D.

Στο Σχήμα 4.1 απεικονίζεται η δομή οργάνωσης για την ανάπτυξη των κλάσεων ανάγνωσης του αρχείου C3D. Η οργάνωση χωρίζεται σε τέσσερις βασικές κλάσεις (κάθε μία σε ξεχωριστά αρχεία κεφαλίδας και πηγαίου κώδικα), ενώ υπάρχουν και δευτερεύουσες κλάσεις, οι οποίες χρησιμοποιούνται για την περιγραφή συγκεκριμένων οντοτήτων, όπως για παράδειγμα είναι οι ομάδες παραμέτρων και οι παράμετροι της ενότητας των παραμέτρων.

Πιο συγκεκριμένα οι κλάσεις χωρίζονται ως εξής:

1. Κλάση **Header_c3d**: Περιέχει τις παραμέτρους της κεφαλίδας ενός αρχείου C3D και συναρτήσεις για την ανάγνωση τους από το αρχείο C3D και την εκτύπωσή τους σε αρχείο κειμένου. Επιπλέον περιέχει συναρτήσεις για την ανάγνωση των τιμών τους από την εφαρμογή.
2. Κλάση **Parameter_c3d**: Χρησιμοποιεί δευτερεύουσες κλάσεις προκειμένου να περιγράψει πλήρως την οργάνωση της ενότητας των παραμέτρων ενός αρχείου C3D. Επιπλέον περιέχει συναρτήσεις για την ανάγνωση των παραμέτρων, την εκτύπωσή τους σε αρχείο κειμένου, καθώς και τη ανάγνωση των παραμέτρων από την εφαρμογή.
3. Κλάση **Data_c3d**: Περιέχει δευτερεύουσες κλάσεις προκειμένου να περιγράψει την οργάνωση της ενότητας των δεδομένων. Επιπλέον περιέχει συναρτήσεις για την ανάγνωση των δεδομένων, την εκτύπωσή τους σε αρχείο μορφοποίησης CSV, καθώς και ανάγνωσή τους από την εφαρμογή.
4. Κλάση **C3D**: Αποτελεί το συνδυαστικό κρίκο των παραπάνω κλάσεων. Σε αυτή την κλάση περιγράφεται πλήρως η δομή του αρχείου C3D, ενώ περιέχει και συναρτήσεις με τις οποίες διαχειρίζεται τις μεθόδους ανάγνωσης του αρχείου C3D, αλλά και την επικοινωνία των παραπάνω κλάσεων με το ευρύτερο πρόγραμμα.

Επιπλέον η κλάση αυτή περιέχει μία μεταβλητή κλάση με όνομα **PointCloud**, η οποία αποτελεί τη βάση της επεξεργασίας του νέφους σημείου. Αυτό σημαίνει πως όταν το αρχείο C3D διαβαστεί επιτυχώς, η τριδιάστατη πληροφορία μεταβιβάζεται και αντιγράφεται στις παραμέτρους (μεταβλητές) της κλάσης **PointCloud** και από εκείνη τη στιγμή και έπειτα τα δεδομένα είναι έτοιμα να προβληθούν στην οθόνη και να επεξεργαστούν.

Στο Σχήμα 4.2 περιγράφεται η πλήρης οργάνωση των αρχείων των κλάσεων, με χρήση σχημάτων UML. Κάθε βέλος δείχνει ποια κλάση χρησιμοποιείται από ποιά. Το πέρας του βέλους δηλώνει ότι εκείνη η κλάση χρησιμοποιείται από την κλάση της αρχής του βέλους. Η αρίθμηση δηλώνει τη σειρά με την οποία καλείται κάθε κλάση κατά τη διάρκεια της ανάγνωσης του αρχείου. Για παράδειγμα, πρώτα ο κώδικας εισέρχεται στην κλάση C3D κι αμέσως μετά στην κλάση Header_c3d (Σύνδεση 1-1). Μόλις ολοκληρωθεί η ανάγνωση της κεφαλίδας του αρχείου C3D, ο κώδικας εισέρχεται στην κλάση Parameter_c3d (Σύνδεση 1-2) και αρχίζει η ανάγνωση της κεφαλίδας των παραμέτρων (Σύνδεση 2-A). Η διαδικασία αυτή συνεχίζεται μέχρι να ολοκληρωθεί η ανάγνωση του αρχείου C3D.

Στα σημεία όπου απαιτείται βρόγχος για την ανάγνωση του αρχείου (π.χ. ανάγνωση των ομάδων παραμέτρων ή των παραμέτρων), ο κώδικας επιστρέφει στην απαραίτητη σύνδεση και η διαδικασία επαναλαμβάνεται μέχρι την ολοκλήρωση του αντίστοιχου τμήματος του αρχείου.

Επιπλέον σημειώνεται ότι στο Σχήμα 4.2 απεικονίζονται μόνο οι κυριότερες κλάσεις, συναρτήσεις και μεταβλητές. Στα αντίστοιχα αρχεία κεφαλίδας και πηγαίου κώδικα υπάρχουν και άλλες βοηθητικές κλάσεις. Ένα παράδειγμα βοηθητικών κλάσεων, που περιέχονται στην κεφαλίδα του αρχείου περιγραφής των παραμέτρων, είναι κλάσεις οι οποίες περιγράφουν τις βασικές ομάδες παραμέτρων, όπως αυτές αναπτύχθηκαν στο κεφάλαιο περιγραφής των αρχείων C3D.

Μερικά παραδείγματα βοηθητικών κλάσεων είναι τα παρακάτω:

- | | |
|-------------------|------------------|
| 1. Trial | 6. Event_Context |
| 2. Subject | 7. Event |
| 3. Point | 8. Manufacturer |
| 4. Analog | 9. Analysis |
| 5. Force_Platform | 10. Processing |

Στις ενότητες που ακολουθούν παρατίθενται αποσπάσματα και περιγραφές από τον κώδικα που υλοποιεί το Σχήμα 4.2.

Σχήμα 4.2. Αναλυτική δομή οργάνωσης των κλάσεων.



4.1.1 Class C3D

```
14 class C3D {
15 public:
16     C3D();
17     inline Header_c3d Header(void) {return header;}
18     inline Parameter_c3d Parameter(void) {return parameter;}
19     inline Data_c3d Data(void) {return data;}
20     inline Trial getTrial(void) {return trial;}
21     inline Subjects getSubjects(void) {return subjects;}
22     inline Point getPoint(void) {return point;}
23     inline Manufacturer getManufacturer(void) {return manufacturer;}
24     inline PointCloud Cloud(void) {return cloud;}
25     inline bool IsOpen(void) {return isOpen;}
26     inline int SysEndian(void) {return sysEndian;}
27     inline float Multiplier(void) {return multiplier;}
28     bool Read_C3D_File(std::string filePath, C3D &c3d_f);
29
30     void CleanMemory(void) {
31         isOpen = false;
32         parameter.CleanMemory();
33         data.CleanMemory();
34         cloud.CleanMemory();
35     }
36     int VisiblePoints(C3D& c3d_f, int frame) {
37         return c3d_f.cloud.VisiblePoints(c3d_f.cloud, frame);}
38     //Noise Remove Cloud Calling Methods
39     void AccumulatedNoiseRemoval(C3D& c3d_f, my_float_ thresh,
40         my_float_ tolerance) {
41         c3d_f.cloud.AccumulatedNoiseRemoval(c3d_f.cloud, thresh,
42         tolerance);
43     }
44     void NearestNeighborRemoval(C3D& c3d_f, my_float_ thresh,
45         int tolerance) {
46         c3d_f.cloud.NearestNeighborRemoval(c3d_f.cloud, thresh,
47         tolerance);
48     }
49     void ResetVisibility(C3D& c3d_f) {
50         c3d_f.cloud.ResetVisibility(c3d_f.cloud); }
51
52     void FixCloudLinear(C3D& c3d_f) {
53         c3d_f.cloud.FixCloudLinear(c3d_f.cloud); }
54
55     void FixCloudLinearWithWeights(C3D& c3d_f) {
56         c3d_f.cloud.FixCloudLinearWithWeights(c3d_f.cloud); }
57
58     void K_Means(C3D& c3d_f, int frameIndex, int clusterSize) {
59         c3d_f.cloud.K_means(c3d_f.cloud, frameIndex, clusterSize);}
60
61     void SetClusters(ClusterWindow &clusterWin, C3D* c3d_f,
62         bool* okState, int index) {
63         clusterWin.SetClusters(clusterWin, &c3d_f->cloud,
64         okState, index);
65     }
66     void SetPointList(PointList &list, C3D* c3d_f) {
67         list.SetList(list, &c3d_f->cloud);
68     }
69     void SetClusterVisibility(C3D& c3d_f, int clusterIndex,
70         bool state) {
71         c3d_f.cloud.SetClusterVisibility(c3d_f.cloud, clusterIndex,
72         state);
73     }
}
```

```

74 private:
75     Header_c3d header;
76     Parameter_c3d parameter;
77     Data_c3d data;
78     Trial trial;
79     Subjects subjects;
80     Point point;
81     Manufacturer manufacturer;
82     PointCloud cloud;
83     bool isOpen = false;
84     int sysEndian;
85     int fileEndian;
86     float multiplier;
87 };

```

Εντός της κλάσης C3D συναντώνται αρκετές συναρτήσεις, οι οποίες επεξεργάζονται το νέφος σημείων και οι οποίες θα καλυφθούν στην Ενότητα 4.2 «Επεξεργασία Δεδομένων».

Η μοναδική συνάρτηση ανάγνωσης των δεδομένων είναι η

1. **Read_C3D_File(...)**: Ανοίγει το αρχείο C3D και χρησιμοποιώντας συναρτήσεις των κλάσεων Header_c3d, Parameter_c3d και Data_c3d, διαβάζει τα δεδομένα του αρχείου.

```

10 bool C3D::Read_C3D_File(std::string filePath, C3D &c3d_f) {
11     FILE* file;
12     file = fopen(filePath.c_str(), "r"); // Open C3D file
13     // Read C3D Header Block
14     c3d_f.header = c3d_f.header.ReadHeaderBlock(file);
15     // Read C3D Parameter Block
16     c3d_f.parameter = c3d_f.parameter.ReadParameterHeaderBlock(file);
17     if(system_endian::little_endian())
18         sysEndian = LITTLE_ENDIAN_PROCESSOR;
19     if(c3d_f.parameter.Header().Processor() == PROCESSOR_MIPS)
20         fileEndian = BIG_ENDIAN_PROCESSOR;
21     int endianFlag = SAME_ENDIAN;
22     if(sysEndian != fileEndian) {
23         endianFlag = DIFF_ENDIAN;
24         c3d_f.header = c3d_f.header.SwapHeaderEndianSystem(c3d_f.header);
25     }
26     c3d_f.parameter = c3d_f.parameter.ReadGroupParameterBlock(file,
27         c3d_f.parameter, endianFlag);
28     int pSize = c3d_f.header.NumberOfPoints();
29     int aSize = c3d_f.header.NumberOfAnalog();
30     int fSize = c3d_f.header.LastFrame() - c3d_f.header.FirstFrame() + 1;
31     //Read C3D Data block
32     c3d_f.data.SetSizes(c3d_f.data, fSize, pSize, aSize);
33     c3d_f.data = c3d_f.data.ReadData(c3d_f.data, file,
34         c3d_f.header.ScaleFactor(), endianFlag);
35     c3d_f.trial = c3d_f.trial.SetTrial(c3d_f.parameter);
36     c3d_f.subjects = c3d_f.subjects.SetSubjects(c3d_f.parameter);
37     c3d_f.point = c3d_f.point.SetPoint(c3d_f.parameter);
38     c3d_f.manufacturer = c3d_f.manufacturer.SetManufacturer(c3d_f.parameter);
39     multiplier = c3d_f.point.MultiplierForMeters(c3d_f.point);
40     //Set Cloud
41     c3d_f.cloud.SetCloud(c3d_f.cloud, c3d_f.data,
42         c3d_f.point, multiplier);
43     return true;
44 }

```

4.1.2 Class Header_c3d

```
9 class Header_c3d
10 {
11 public:
12     const static int futureBlock_1_size = 135;
13     const static int eventTime_size = 18;
14     const static int displayFlag_size = 9;
15     const static int eventLabel_size = 18;
16     const static int futureBlock_4_size = 22;
17     Header_c3d(); /* Constructor */
18     inline byte ParameterBlock(void) {return parameterBlock;}
19     inline byte NumberID(void) {return idNumber;}
20     inline byte NumberOfPoints(void) {return pointsNumber;}
21     inline byte NumberOfAnalog(void) {return analogNumber;}
22     inline byte FirstFrame(void) {return firstFrame;}
23     inline byte LastFrame(void) {return lastFrame;}
24     inline byte MaxGap(void) {return maximumInterpolationGap;}
25     inline real ScaleFactor(void) {return scaleFactor;}
26     inline byte DataStart(void) {return dataStart;}
27     inline byte AnalogPerFrame(void) {return analogPerFrame;}
28     inline real FrameRate(void) {return frameRate;}
29     inline byte FutureBlock_1(const int index) {
30         if(index < 0 || index >= futureBlock_1_size)
31             return (byte)NULL;
32         else
33             return futureUseBlock_1[index];
34     }
35     inline byte KeyValue_1(void) {return keyValue_1;}
36     inline byte FirstLabelRange(void) {return firstLabelRange;}
37     inline byte KeyValue_2(void) {return keyValue_2;}
38     inline byte EventTimeSize(void) {return eventTimeSize;}
39     inline byte FutureBlock_2(void) {return futureUseBlock_2;}
40     inline real EventTimeInSeconds(const int index) {
41         if(index < 0 || index >= eventTime_size)
42             return (real)NULL;
43         else
44             return eventTimeInSeconds[index];
45     }
46     inline byte EventDisplayFlags(const int index) {
47         if(index < 0 || index >= displayFlag_size)
48             return (byte)NULL;
49         else
50             return eventDisplayFlags[index];
51     }
52     inline byte FutureBlock_3(void) {return futureUseBlock_3;}
53     inline std::string EventLabel(const int index) {
54         if(index < 0 || index >= eventLabel_size)
55             return (byte)NULL;
56         else {
57             std::string str(eventLabels[index]);
58             return str;
59         }
60     }
61     inline byte FutureBlock_4(const int index) {
62         if(index < 0 || index >= displayFlag_size)
63             return (byte)NULL;
64         else
65             return futureUseBlock_4[index];
66     }
67 }
```

```

68     /* Read the 512-bytes Header block */
69     Header_c3d ReadHeaderBlock(FILE* file);
70     /* Swap the Endian Reading System */
71     Header_c3d SwapHeaderEndianSystem(Header_c3d header);
72     /* Print Header to File */
73     bool PrintToFile(std::string path);
74
75 private:
76     /* byte 1: Points to the first block of the parameter section. */
77     byte parameterBlock;
78     /* byte 2: Key value 0x50hex indicating a C3D file */
79     byte idNumber;
80     /* Number of 3D points in the C3D file
81      * (i.e. the number of the stored trajectories). */
82     byte pointsNumber;
83     /* Total number of analog measurements per 3D frame, i.e. number of
84      * channels multiplied by the samples per channel. */
85     byte analogNumber;
86     /* Number of the first frame of the 3D data. */
87     byte firstFrame;
88     /* Number of the last frame of 3D data. */
89     byte lastFrame;
90     /* Maximum interpolation gap in 3D frames. */
91     byte maximumInterpolationGap;
92     /* The 3D scale factor (realing-point) that converts signed integer
93      * 3D data to reference system measurement unit. If this is negative
94      * then the file is scaled in realing-point. */
95     real scaleFactor;
96     /* DATA_START - the number of the first block of the 3D and
97      * Analog data section. */
98     byte dataStart;
99     /* The number of Analog samples per 3D frame. */
100    byte analogPerFrame;
101    /* The 3D frame rate in Hz (realing-point) */
102    real frameRate;
103    /* Reserved for future use. */
104    std::vector<byte> futureUseBlock_1;
105    /* A key value (12345 decimal) is written here if Label and Range
106     * data is present, otherwise write 0x00. */
107    byte keyValue_1;
108    /* The first block of the Label and Range section (if present). */
109    byte firstLabelRange;
110    /* A key value (12345 decimal) present if this file supports 4 char
111     * event labels. An older format supported only 2 character labels.*/
112    byte keyValue_2;
113    /* Number of defined time event (0 to 18). */
114    byte eventTimeSize;
115    /* Reserved for future use. */
116    byte futureUseBlock_2;
117    /* Event time (realing-point) measured in seconds (up to 18 events). */
118    std::vector<real> eventTimeInSeconds;
119    /* 18-byte Event display Flags 0x00 = ON or 0x01 = OFF. */
120    std::vector<byte> eventDisplayFlags;
121    /* Reserved for future use. */
122    byte futureUseBlock_3;
123    /* Event labels. Each label is 4 character long. */
124    std::vector<std::string> eventLabels;
125    /* Reserved for future use. */
126    std::vector<byte> futureUseBlock_4;
127 };

```

Η κλάση `Header_c3d` περιέχει μεταβλητές, οι οποίες περιγράφουν την κεφαλίδα του αρχείου C3D. Οι τιμές των μεταβλητών αυτών επικοινωνούν με την εφαρμογή μέσω συναρτήσεων που επιστρέφουν τις τιμές τους.

Ωστόσο υπάρχουν και οι συναρτήσεις, οι οποίες διαβάζουν την κεφαλίδα του C3D αρχείου και οι οποίες είναι:

1. **ReadHeaderBlock(...):** Διαβάζει την κεφαλίδα του C3D αρχείου και αποθηκεύει τις τιμές στις αντίστοιχες μεταβλητές.
2. **SwapHeaderEndianSystem(...):** Στην περίπτωση όπου ο επεξεργαστής του υπολογιστή και ο επεξεργαστής του δέκτη καταγραφής είναι διαφορετικοί, καλείται η συνάρτηση αυτή και «μεταφράζει» τα δεδομένα από τον επεξεργαστή του δέκτη στον επεξεργαστή του συστήματος. Αυτό επιτυγχάνεται με την αλλαγή των ενδιανών.
3. **PrintToFile(...):** Εκτυπώνει την κεφαλίδα σε αρχείο κειμένου.

```
38 /* Read the 512-bytes Header block */
39 Header_c3d Header_c3d::ReadHeaderBlock(FILE* file) {
40     Header_c3d header;
41     readFromFile(&header.parameterBlock, SIZE_8_BIT, file);
42     if(header.parameterBlock == 1) {
43         fseek(file, 0, SEEK_SET);
44         return header;
45     }
46     readFromFile(&header.idNumber, SIZE_8_BIT, file);
47     readFromFile(&header.pointsNumber, SIZE_16_BIT, file);
48     readFromFile(&header.analogNumber, SIZE_16_BIT, file);
49     readFromFile(&header.firstFrame, SIZE_16_BIT, file);
50     readFromFile(&header.lastFrame, SIZE_16_BIT, file);
51     readFromFile(&header.maximumInterpolationGap, SIZE_16_BIT, file);
52     readFromFile(&header.scaleFactor, SIZE_32_BIT, file);
53     readFromFile(&header.dataStart, SIZE_16_BIT, file);
54     readFromFile(&header.analogPerFrame, SIZE_16_BIT, file);
55     readFromFile(&header.frameRate, SIZE_32_BIT, file);
56     for(auto iter : header.futureUseBlock_1)
57         readFromFile(&iter, SIZE_16_BIT, file);
58     readFromFile(&header.keyValue_1, SIZE_16_BIT, file);
59     readFromFile(&header.firstLabelRange, SIZE_16_BIT, file);
60     readFromFile(&header.keyValue_2, SIZE_16_BIT, file);
61     readFromFile(&header.eventTimeSize, SIZE_16_BIT, file);
62     readFromFile(&header.futureUseBlock_2, SIZE_16_BIT, file);
63     for(auto iter : header.eventTimeInSeconds)
64         readFromFile(&iter, SIZE_32_BIT, file);
65     for(auto iter : header.eventDisplayFlags)
66         readFromFile(&iter, SIZE_16_BIT, file);
67     readFromFile(&header.futureUseBlock_3, SIZE_16_BIT, file);
68     for(auto iter : header.eventLabels)
69         for(auto siter : iter)
70             readFromFile(&siter, SIZE_8_BIT, file);
71     for(auto iter : header.futureUseBlock_4)
72         readFromFile(&iter, SIZE_16_BIT, file);
73     return header;
74 }
```


4.1.3 Class Parameter_c3d

```
194 class Parameter_c3d
195 {
196 public:
197     Parameter_c3d();
198     inline Parameter_Header_c3d Header(void) {return header;}
199     inline short int GroupSize(void) {return groupSize;}
200     inline Group_c3d Group(const int index) {
201         if(index > 127 || index < 0)
202             return group[0];
203         else
204             return group[index];
205     }
206     Parameter_c3d ReadParameterHeaderBlock(FILE* file);
207     Parameter_c3d ReadGroupParameterBlock(FILE* file, Parameter_c3d parameter,
208         const int flag);
209     bool PrintToFile(std::string path);
210     void CleanMemory(void) {
211         for(int i = 0; i != groupSize; ++i)
212             group->CleanMemory(group[i]);
213         free(group);
214     }
215 private:
216     Parameter_Header_c3d header;
217     short int groupSize;
218     Group_c3d* group;
219 };
```

Η κλάση `Parameter_c3d` περιέχει μεταβλητές κλάσεων, οι οποίες περιγράφουν την ενότητα των παραμέτρων του αρχείου C3D, και συναρτήσεις με τις οποίες δίνει τη δυνατότητα στην εφαρμογή να επικοινωνήσει με τις τιμές των μεταβλητών αυτών. Επιπλέον περιέχει συναρτήσεις, οι οποίες διαβάζουν την ενότητα των παραμέτρων του αρχείου C3D, οι οποίες είναι:

1. **ReadParameterHeaderBlock(...):** Διαβάζει την κεφαλίδα των παραμέτρων του C3D αρχείου και αποθηκεύει τις τιμές στις αντίστοιχες μεταβλητές.
2. **ReadGroupParameterBlock(...):** Διαβάζει και αποθηκεύει στις αντίστοιχες μεταβλητές κλάσεων τις παραμέτρους και τις ομάδες παραμέτρων ενός αρχείου C3D.
3. **PrintToFile(...):** Εκτυπώνει τις παραμέτρους σε αρχείο κειμένου.

Η φιλοσοφία σχεδιασμού της κλάσης είναι αρχικά η ανάγνωση της κεφαλίδας των παραμέτρων και στη συνέχεια το πρόγραμμα εισέρχεται σε ένα βρόγχο προκειμένου να διαβάσει τις ομάδες και τις παραμέτρους κάθε ομάδας. Αυτό επιτυγχάνεται με τη δημιουργία μίας δευτερεύουσας κλάσης με όνομα `Group_c3d`, η οποία περιγράφει τη μορφή μίας ομάδας. Αντίστοιχα η κλάση `Group_c3d` χρησιμοποιεί την κλάση `Parameter_c3d`, όπως φαίνεται στο Σχήμα 4.2, για να καταγράψει τις παραμέτρους. Η κλάση `Parameter_c3d` περιγράφει τη δομή μιας παραμέτρου.

4.1.4 Class Data_c3d

```
88 class Data_c3d {
89 public:
90     Data_c3d();
91     inline Frame_c3d Frame(const int index) {return frame[index];}
92     inline int PointSize(void) {return pointSize;}
93     inline int FrameSize(void) {return frameSize;}
94     inline int AnalogDataSize(void) {return analogDataSize;}
95     /* Read Data */
96     Data_c3d ReadData(Data_c3d data, FILE* file, const float scale,
97                     const int flag);
98     void SetSizes(Data_c3d& data, const int fSize, const int pSize,
99                     const int aSize){
100         data.frameSize = fSize;
101         data.pointSize = pSize;
102         data.analogDataSize = aSize;
103     }
104     /* Print Points to CSV File */
105     bool PrintPointsToFile_CSV(std::string path);
106     /* Print Analog Data to CSV File */
107     bool PrintAnalogToFile_CSV(std::string path);
108     void CleanMemory() {
109         for(int i = 0; i != frameSize; ++i)
110             frame->CleanMemory(frame[i]);
111         free(frame);
112     }
113 private:
114     Frame_c3d* frame;
115     int pointSize;
116     int frameSize;
117     int analogDataSize;
118 };
```

Η κλάση Data_c3d περιέχει μεταβλητές κλάσεων, οι οποίες περιγράφουν την ενότητα των δεδομένων του αρχείου C3D, και συναρτήσεις επικοινωνίας της εφαρμογής με τις τιμές των μεταβλητών αυτών. Τέλος οι βασικές συναρτήσεις με τις οποίες διαβάζονται τα δεδομένα είναι οι εξής:

1. **ReadData(...)**: Διαβάζει τα χωρικά και αναλογικά δεδομένα ενός αρχείου C3D και τα αποθηκεύει στις αντίστοιχες μεταβλητές κλάσεων.
2. **SetSizes(...)**: Αποθηκεύει το μέγεθος των καρτέ, των σημείων ανά καρτέ και τα αναλογικά δεδομένα ανά καρτέ.
3. **PrintPointsToFile_CSV(...)**: Εκτυπώνει τα τριδιάστατα χωρικά δεδομένα σε αρχείο CSV.
4. **PrintAnalogToFile_CSV(...)**: Εκτυπώνει τα αναλογικά δεδομένα σε αρχείο CSV.

Προκειμένου να γίνει καλύτερη διαχείριση της πληροφορίας η κλάση χρησιμοποιεί δημιουργεί μία μεταβλητή δείκτη τύπου Frame_c3d (χρήση δευτερεύουσας) κλάσης. Η δευτερεύουσα κλάση Frame_c3d περιέχει μεταβλητές περιγραφής ενός καρτέ. Αντίστοιχα η κλάση Frame_c3d καλεί την κλάση Point_c3d. Η κλάση Point_c3d περιέχει τις μεταβλητές που περιγράφουν μία μέτρηση ενός σημείου, όπως αυτό αναφέρθηκε στο κεφάλαιο περιγραφής των αρχείων C3D.

4.2 Επεξεργασία Δεδομένων

Η δεύτερη ομάδα κλάσεων και συναρτήσεων αποτελείται από τις κλάσεις και συναρτήσεις που επεξεργάζονται το νέφος σημείων. Η κύρια κλάση ονομάζεται **PointCloud** και η ανάγκη να δημιουργηθεί μία τέτοια κλάση, είναι για λόγους συμβατότητας της εφαρμογής και με άλλους τύπους αρχείων πέραν των αρχείων C3D (ακόμα η εφαρμογή δεν υποστηρίζει την ανάγνωση άλλων τύπων αρχείων).

Στην κλάση αυτή συναντώνται τρεις ομάδες συναρτήσεων:

1. Συναρτήσεις διαχείρισης του νέφους σημείων (οπτικοποίηση, εισαγωγή κι εξαγωγή των δεδομένων).
2. Συναρτήσεις επεξεργασίας του νέφους σημείων (αφαίρεση θορύβου, διόρθωση νέφους σημείων).
3. Συναρτήσεις ταξινόμησης και διαχείρισης των ομάδων του νέφους σημείων.

Class PointCloud:

```
206 class PointCloud {
207 public:
208     PointCloud();
209     inline Frame Frame_(const int index) { return frame[index]; }
210     inline int FrameSize(void) { return frameSize; }
211     inline int PointSize(void) { return pointSize; }
212     inline Cluster Cluster_(const int index) { return cluster[index]; }
213     inline int ClusterSize(void) { return clusterSize; }
214     void SetCloud(PointCloud& cloud, Data_c3d data, Point point,
215                 const float multiplier);
216     inline void SetPointVisibility(PointCloud& cloud, int frameIndex,
217     int pointIndex, bool state) {
218         cloud.frame->SetPointVisibility(cloud.frame[frameIndex],
219         pointIndex, state); }
220
221
222     inline void ResetVisibility(PointCloud& cloud) {
223         for(int i = 0; i != cloud.frameSize; ++i)
224             for(int j = 0; j != cloud.pointSize; ++j)
225                 cloud.frame->SetPointVisibility(cloud.frame[i], j, true);}
226
227     inline int VisiblePoints(PointCloud& cloud, int frame) {
228         int cnt = 0;
229         for(int i = 0; i < cloud.PointSize(); i++)
230             if(cloud.Frame_(frame).Point_(i).Visible())
231                 ++cnt;
232         return cnt; }
233
234     inline void CleanMemory(void) {
235         for(int i = 0; i != frameSize; ++i)
236             frame->CleanMemory(frame[i]);
237         free(frame);
238         free(cluster); }
239
240     void PrintCloud(void);
241     void PrintCloud(int fromFrame, int toFrame, char seperator,
242                 std::string path);
243     void PrintClusters(int fromFrame, int toFrame, char seperator,
244                 std::string path);
245     void PrintDist(void);
```

```

246
247 void AccumulatedNoiseRemoval(PointCloud& cloud, my_float_ thresh,
248                               my_float_ tolerance);
249 void NearestNeighborRemoval(PointCloud& cloud, my_float_ thresh,
250                               int tolerance);
251 void FixCloudLinear(PointCloud& cloud);
252 void FixCloudLinearWithWeights(PointCloud& cloud);
253
254 void K_means(PointCloud& cloud, int frameIndex, int clusterSize);
255 void SetClusterColor(PointCloud& cloud, int clusterIndex, int red,
256                       int green, int blue) {
257     cloud.cluster->SetColor(cloud.cluster[clusterIndex], red,
258                             green, blue); }
259
260 void SetPointSize(PointCloud& cloud, int clusterIndex, my_float_ sz)
261 { cloud.cluster->SetPointSize(cloud.cluster[clusterIndex], sz); }
262
263 void SetClusterVisibility(PointCloud& cloud, int clusterIndex,
264                             bool state) {
265     cloud.cluster->SetVisibility(cloud.cluster[clusterIndex],
266                                 state); }
267
268 void SetName(PointCloud& cloud, int clusterIndex,
269              const std::string& c_name) {
270     cloud.cluster->SetName(cloud.cluster[clusterIndex], c_name); }
271
272 private:
273     Frame* frame;
274     int frameSize;
275     int pointSize;
276     Cluster* cluster;
277     int clusterSize;
278 };

```

4.2.1 Συναρτήσεις Διαχείρισης του Νέφους Σημείων

Η πρώτη ομάδα συναρτήσεων, είναι αυτές της διαχείρισης του νέφους σημείων.

Ως **νέφος σημείων** ο κώδικας αντιλαμβάνεται ένα πλήθος σημείων, τα οποία αποτελούνται από **συντεταγμένες** (x, y, z, t) , όπου t είναι η διάσταση του χρόνου, η οποία εκφράζεται μέσω της έννοιας του ενός **καρέ**.

Ως **καρέ** η εφαρμογή αντιλαμβάνεται ένα σύνολο σημείων, τα οποία αποτελούνται από **συντεταγμένες** (x, y, z) . Αυτό το σύνολο σημείων, στην πραγματικότητα εκφράζει το στιγμιότυπο μίας χρονικής στιγμής t .

Ως **σημείο** η εφαρμογή αντιλαμβάνεται μία τριάδα μεταβλητών (x, y, z) . Επιπλέον στην κλάση περιγραφής των σημείων περιλαμβάνονται και κάποιες επιπλέον μεταβλητές, οι οποίες βοηθούν την εφαρμογή να το προβάλει στην οθόνη. Οι μεταβλητές αυτές είναι **ένας κωδικός αριθμός (id)**, που ορίζει τη θέση του σημείου στο καρέ, **ένας κωδικός αριθμός (c_id)**, που ορίζει την **κλάση (ομάδα)** στην οποία ανήκει το σημείο και **μία λογική μεταβλητή**, η οποία εκφράζει το αν πρέπει ή όχι το σημείο να προβληθεί στην οθόνη και να συμπεριληφθεί στην επεξεργασία των δεδομένων.

Ως **κλάση (ομάδα)** η εφαρμογή αντιλαμβάνεται ένα σύνολο σημείων, τα οποία μοιράζονται τον ίδιο κωδικό κλάσης. Ο κωδικός κλάσης **μηδέν** όταν το νέφος σημείων αντιστοιχεί σε **μία** κλάση εκφράζει όλο το νέφος σημείων, ενώ όταν το νέφος σημείων μοιράζεται σε **περισσότερες από μία** κλάσεις εκφράζει τα σημεία που δεν εμφανίζονται στην οθόνη και δε συμμετάσχουν στην επεξεργασία, διότι θεωρούνται **προβληματικά (θόρυβος)**. Για αυτό και σε πολλές περιπτώσεις παρακάτω, η όποια αναφορά γίνει στην κλάση αυτή, θα χρησιμοποιεί το όνομα **«κλάση θορύβου»**. Επιπλέον η κλάση θορύβου, ανεξάρτητα από το πλήθος των σημείων που έχει καταχωρημένα, έχει κέντρο μάζας το σημείο $(0, 0, 0)$.

Οι συναρτήσεις διαχείρισης του νέφους που περιλαμβάνονται μέσα στην κλάση PointCloud, είναι οι εξής:

1. **SetCloud(...)**: Δημιουργεί το νέφος σημείων, από τα αρχικά C3D δεδομένα. Στη συνάρτηση αυτή δεσμεύεται η απαραίτητη μνήμη του νέφους σημείων και του μέγιστου αριθμού των κλάσεων ταξινόμησης που υποστηρίζει η εφαρμογή. Επιπλέον σε αυτή τη συνάρτηση πραγματοποιείται μία προεπεξεργασία και για κάθε σημείο σε κάθε καρέ, υπολογίζονται οι ευκλείδειες αποστάσεις του κάθε σημείου από τα υπόλοιπα σημεία και η τιμή της απόστασης καταχωρείται σε ένα πίνακα διαστάσεων:

$$\text{FrameSize} \times \text{PointSize} \times \text{PointSize}$$

2. **SetPointVisibility(...)**: Ορίζει τη λογική τιμή ορατότητας ενός σημείου.
3. **ResetVisibility(...)**: Ορίζει τις λογικές τιμές όλων των σημείων ως αληθείς. Αυτό σημαίνει πως όλα τα σημεία θα προβληθούν στην οθόνη.
4. **CleanMemory(...)**: Αποδεσμεύει την μνήμη που δεσμεύτηκε στη συνάρτηση SetCloud().
5. **PrintCloud(...)**: Εκτυπώνει το νέφος σημείων σε αρχείο μορφοποίησης CSV. Η κλάση περιέχει δύο συναρτήσεις με αυτό το όνομα που εκτελούν την ίδια διαδικασία, αλλά με ελαφρώς διαφορετικό αποτέλεσμα.
6. **PrintDist(...)**: Εκτυπώνει τον πίνακα των αποστάσεων σε αρχείο CSV.

4.2.2 Συναρτήσεις Επεξεργασίας του Νέφους Σημείων

Η δεύτερη ομάδα συναρτήσεων που συναντώνται στην κλάση PointCloud είναι αυτή της επεξεργασίας του νέφους σημείων. Οι συναρτήσεις αυτές κυρίως αφαιρούν κάποια σημεία που θεωρούνται θόρυβος, ενώ υπάρχουν και συναρτήσεις που διαβάζουν το υφιστάμενο νέφος σημείων και προσπαθούν να υποθέσουν τη θέση ενός σημείου στο χώρο, χρησιμοποιώντας την τάση του σημείου από τα προηγούμενα δύο καρέ.

Η εφαρμογή με την έννοια **αφαίρεση θορύβου** αντιλαμβάνεται την αλλαγή ορατότητας ενός σημείου. Αυτό δίνει τη δυνατότητα στην εφαρμογή να μπορεί να επεξεργαστεί το νέφος σημείου πολλές φορές, χωρίς να απαιτείται από την εφαρμογή να ξαναδιαβάσει τα δεδομένα εισαγωγής, για να ξαναδημιουργήσει το νέφος σημείων. Ωστόσο υπάρχουν κι εντολές, οι οποίες τροποποιούν τις συντεταγμένες των σημείων. Οι εντολές αυτές πειράζουν όλο το νέφος σημείων και για αυτό πρέπει να εκτελούνται όταν κρίνεται απαραίτητο. Μετά την εκτέλεση των εντολών αυτών, προκειμένου το νέφος σημείου να επιστρέψει στην αρχική του δομή, απαιτείται να διαβαστεί ξανά το αρχείο C3D.

Οι συναρτήσεις επεξεργασίας του νέφους σημείων που περιέχονται μέσα στην κλάση PointCloud, είναι οι εξής:

1. **AccumulatedNoiseRemoval(...)**: Αφαιρεί το συσσωρευμένο θόρυβο. Ως συσσωρευμένος θόρυβος θεωρούνται σημεία που δε μεταβάλλονται πάνω από μία δεδομένη τιμή απόστασης κατωφλίσωσης σε δεδομένο ποσοστό των καρέ του νέφους σημείων.
2. **NearestNeighborRemoval(...)**: Διαβάζει για κάθε σημείο σε κάθε καρέ, τον πίνακα των αποστάσεων κι αν το σημείο αυτό έχει ένα δεδομένο πλήθος γειτόνων που απέχουν από αυτό απόσταση μικρότερη ή ίση από μία δεδομένη τιμή απόστασης, τότε το σημείο αυτό θεωρείται αξιόπιστο. Η συνθήκη αυτή πρέπει να ικανοποιείται για όλα τα καρέ του νέφους σημείων. Εάν η συνθήκη αυτή δεν ικανοποιείται σε τουλάχιστον ένα καρέ, τότε το σημείο αυτό απομακρύνεται από όλα τα καρέ.

Η συνάρτηση αυτή απομακρύνει τα σημεία εκείνα που ενώ σε κάποια καρέ αποτελούν τμήμα του αντικειμένου, σε κάποια άλλα μετατρέπονται σε σημεία θορύβου. Η συνάρτηση αυτή συναντάει πρόβλημα, όταν σημεία αντικειμένου, μετατρέπονται σε θόρυβο με κοντινές ή ίδιες συντεταγμένες. Σε αυτή την περίπτωση απαιτείται διόρθωση του νέφους σημείων.

3. **FixCloudLinear(...)**: Διορθώνει το νέφος σημείων χρησιμοποιώντας την τάση του σημείου, την οποία υπολογίζει χρησιμοποιώντας τα προηγούμενα δύο καρέ. Ως τάση σημείου ορίζεται η διαφορά των συντεταγμένων του σημείου από το προηγούμενο και το προ-προηγούμενο καρέ. Δηλαδή:

$$\tau = \text{frame}_{n-1} - \text{frame}_{n-2}$$

Όπου:

$\tau \rightarrow$ η τάση του σημείου.

$n \rightarrow$ το υφιστάμενο καρέ.

Από την παραπάνω σχέση, προκύπτει πως δεν υφίσταται τάση για τα πρώτα δύο καρέ του νέφους σημείων, ενώ η τάση για διορθωμένα σημεία, είναι σταθερή και ισούται με την τάση των δύο τελευταίων έγκυρων καρέ.

Το αποτέλεσμα της συνάρτησης αυτής είναι μη αναστρέψιμο.

4. **FixCloudLinearWithWeights(...)**: Η συνάρτηση αυτή εκτελεί την ίδια διαδικασία με την `FixCloudLinear(...)`, με τη διαφορά ότι σε κάθε συνεχόμενο καρέ θορύβου μειώνει την τάση. Το αποτέλεσμα της συνάρτησης δεν είναι αποδεχτό στις περιπτώσεις που σημεία του αντικειμένου συμπεριφέρονται ως θόρυβος σε μεγάλο πλήθος διαδοχικών καρέ.

Το αποτέλεσμα της συνάρτησης αυτής είναι μη αναστρέψιμο.

4.2.3 Συναρτήσεις Ταξινόμησης και Διαχείρισης των Ομάδων του Νέφους Σημείων

Η τρίτη ομάδα συναρτήσεων που συναντάται στην κλάση `PointCloud`, διαχειρίζεται τις κλάσεις των σημείων. Οι συναρτήσεις της ομάδας είναι οι εξής:

1. **K_means(...)**: Ταξινομεί τα σημεία του νέφους σημείων ενός συγκεκριμένου καρέ σε ομάδες. Η ταξινόμηση αυτή εάν και εκτελείται σε ένα μόνο καρέ, ισχύει για όλα τα καρέ του νέφους σημείων. Ο αλγόριθμος που χρησιμοποιείται είναι ο K-Means και ο μέγιστος αριθμός των κλάσεων που μπορεί να ταξινομήσει είναι εννιά κλάσεις. Ο περιορισμός αυτός οφείλεται στην αδυναμία του αλγορίθμου να εμφανίσει σωστό αποτέλεσμα, δίχως να έχει καλές αρχικές τιμές των κεντροειδών. Για τις ανάγκες τις διπλωματικής, χρειάζονται έντεκα σημεία σκελετού (ιδανικά έντεκα κλάσεις). Ωστόσο όπως θα αναφερθεί παρακάτω, υπάρχει τρόπος να προσδιοριστούν και οι έντεκα τιμές του σκελετού χρησιμοποιώντας ταξινόμηση K-Means για οχτώ κλάσεις.

Εάν χρειαστεί για οποιαδήποτε εφαρμογή να ταξινομήσει ένα νέφος σημείου σε περισσότερες από εννιά κλάσεις, τότε το μόνο που χρειάζεται είναι η προσθήκη υπολογισμού των επιπλέον αρχικών κεντροειδών στην αρχή του αλγορίθμου.

2. **PrintClusters(...)**: Εκτυπώνει τις κλάσεις (ομάδες) του νέφους σημείων και τα κέντρα μάζας τους σε ξεχωριστά αρχεία μορφοποίησης CSV. Από την εκτύπωση εξαιρείται η κλάση θορύβου.
3. **SetClusterColor(...)**: Ορίζει το χρώμα σχεδίασης των αντικειμένων της κλάσης σημείων στην οθόνη.
4. **SetPointSize(...)**: Ορίζει το μέγεθος απεικόνισης των σημείων μιας κλάσης σημείων.
5. **SetClusterVisibility(...)**: Ορίζει την ορατότητα μιας κλάσης σημείων. Αποσαφηνίζεται, πως ένα σημείο που ανήκει σε μία κλάση με τιμή ορατότητα `false` και ή τιμή ορατότητας του σημείου είναι `true`, τότε το σημείο αυτό θα συμμετάσχει στις συναρτήσεις που διαχειρίζονται ή επεξεργάζονται τα σημεία.
6. **SetName(...)**: Ορίζει το όνομα μία κλάσης σημείων.

Κεφάλαιο 5

Επίδειξη της Διεπαφής

Στην εισαγωγή έγινε αναφορά στη δημιουργία εφαρμογής¹ με στόχο την οπτικοποίηση και επεξεργασία των τρισδιάστατων δεδομένων. Σε αυτό το κεφάλαιο θα πραγματοποιηθεί ανάλυση της εφαρμογής οπτικοποίησης και επεξεργασίας των τρισδιάστατων σημείων. Συγκεκριμένα η διαδικασία αυτή αποτελείται από τα εξής στάδια:

1. Εισαγωγή των δεδομένων στην εφαρμογή.
2. Αφαίρεση του θορύβου από το νέφος σημείου.
3. Ταξινόμηση των σημείων.
4. Αναγνώριση των κλάσεων.
5. Εξαγωγή των αποτελεσμάτων.

Επιπλέον στο προηγούμενο κεφάλαιο έγινε περιγραφή των κλάσεων και των συναρτήσεων με τον οποίο διαβάζονται και αποθηκεύονται τα δεδομένα σε μεταβλητές εντός της εφαρμογής, καθώς επίσης και των διαφόρων μεθόδων επεξεργασίας αυτών. Σε αυτό το κεφάλαιο θα γίνει παρουσίαση της διεπαφής και πιο συγκεκριμένα με τη χρήση σχημάτων, θα παρουσιαστεί αναλυτική η αλληλεπίδραση του χρήστη με το λογισμικό, ώστε να πραγματοποιηθούν οι αλγόριθμοι του προηγούμενου κεφαλαίου.

5.1 Εισαγωγή των Δεδομένων στην Εφαρμογή

Πριν πραγματοποιηθεί η οποιαδήποτε επεξεργασία, πρέπει πρώτα η εφαρμογή να διαβάσει τα δεδομένα. Η διαδικασία αυτή παρουσιάζεται παρακάτω.

Στο Σχήμα 5.2 απεικονίζεται το αρχικό παράθυρο, όταν ανοίγει η εφαρμογή. Με μία γρήγορη ματιά παρατηρούνται τα εξής χαρακτηριστικά:

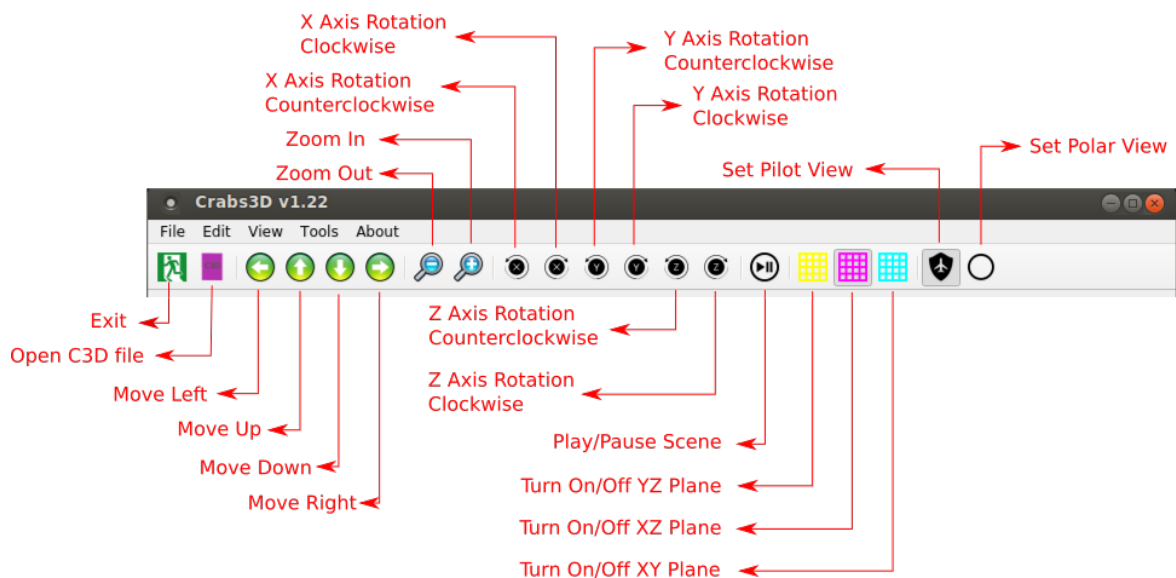
1. Στο πάνω μέρος του παραθύρου αναγράφεται το όνομα του λογισμικού «Crabs3D v1.22».
2. Ακριβώς κάτω από το όνομα του παραθύρου, βρίσκεται η γραμμή εντολών με βασικές λειτουργίες διαχείρισης της απεικόνισης της προβαλλόμενης σκηνής.

Συγκεκριμένα η γραμμή εντολών αποτελείται από τις εξής βασικές κατηγορίες:

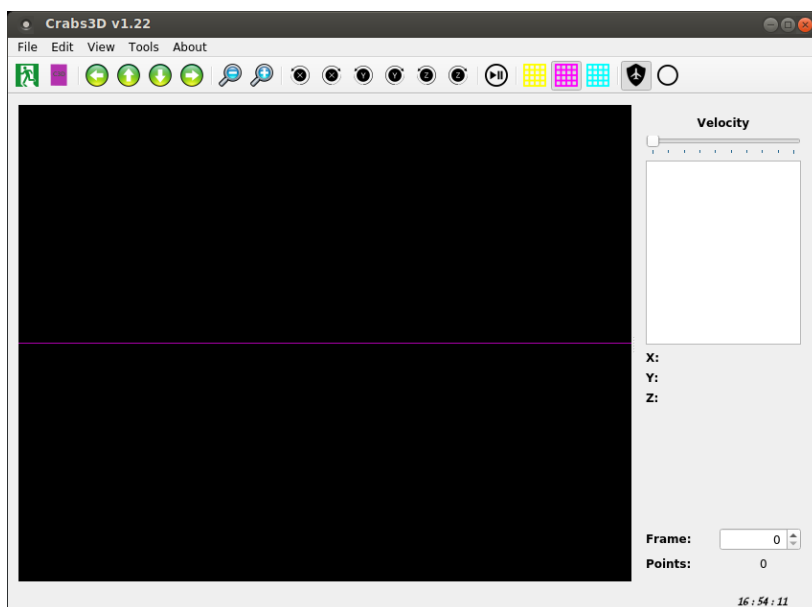
- (α) **File:** Περιέχει εντολές για την εισαγωγή και εξαγωγή των δεδομένων. Η εφαρμογή μπορεί να διαβάσει μόνο αρχεία C3D και να εξάγει πληροφορία σε μορφή TXT και CSV.
- (β) **Edit:** Περιέχει απλές εντολές επεξεργασίας και οπτικοποίησης της σκηνής, όπως είναι η αλλαγή της κλίμακας απεικόνισης και η επεξεργασία του τρόπου απεικόνισης των ομάδων.
- (γ) **View:** Στην κατηγορία αυτή περιλαμβάνονται εντολές περιήγησης της σκηνής. Χρησιμοποιώντας αυτές τις εντολές, ο χρήστης μπορεί να αλλάξει την οπτική γωνία, καθώς επίσης και τον τρόπο προβολής της σκηνής.
- (δ) **Tools:** Περιέχει τις εντολές επεξεργασίας του νέφους σημείων, όπως είναι η αφαίρεση θορύβου και η ταξινόμηση του νέφους σημείων με χρήση του αλγορίθμου K-Means.
- (ε) **About:** Περιέχει πληροφορίες σχετικά με το λογισμικό.

Για μερικές από τις εντολές που υπάρχουν στις παραπάνω ομάδες υπάρχουν και συντομεύσεις στη δευτερεύουσα γραμμή εντολών. Στο Σχήμα 5.1 περιγράφεται η δευτερεύουσα γραμμή εντολών.

3. Κάτω από τη δευτερεύουσα γραμμή εντολών βρίσκεται το παράθυρο απεικόνισης της σκηνής και η ματζέντα γραμμή αντιστοιχεί στο επίπεδο XZ . Σημειώνεται ότι η αρχική προβολή είναι η ορθογραφική.
4. Δεξιά από τη σκηνή βρίσκονται η μπάρα της ταχύτητας αλλαγής των καρτέ, από κάτω η λίστα των κλάσεων (ομάδων) των σημείων και από κάτω απεικονίζονται οι συντεταγμένες (X, Y, Z) των κεντροειδών των κλάσεων. Στην κάτω δεξιά γωνία απεικονίζονται το υφιστάμενο καρτέ, το πλήθος των σημείων που είναι ορατά στην σκηνή και η υφιστάμενη ώρα.



Σχήμα 5.1. Δευτερευουσα Γραμμή Εντολών



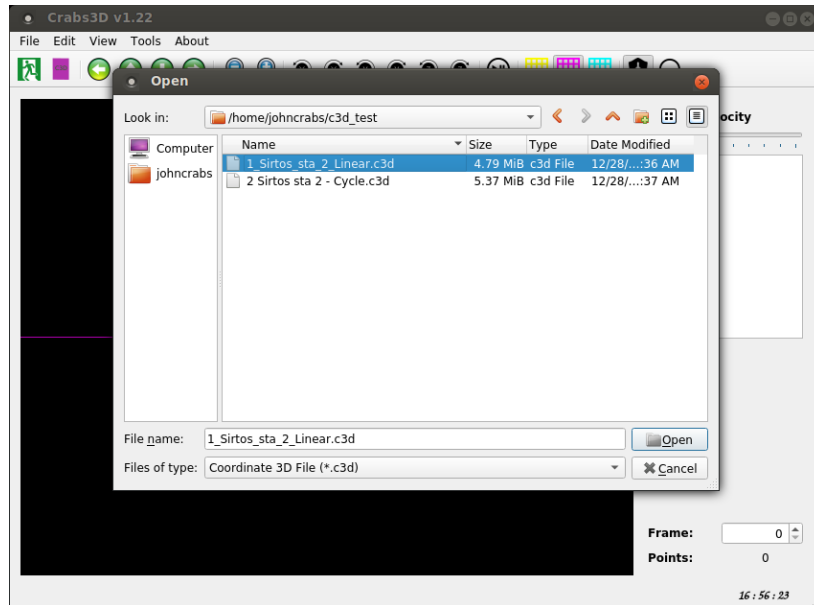
Σχήμα 5.2. Αρχικό παράθυρο.

Στο Σχήμα 5.3 περιγράφεται η διαδικασία εισαγωγής ενός αρχείου C3D στην εφαρμογή. Αυτό υλοποιείται είτε κάνοντας κλικ πάνω στο μωβ εικονίδιο της γραμμής εντολών, είτε ακολουθώντας το μονοπάτι File → Import → C3D.

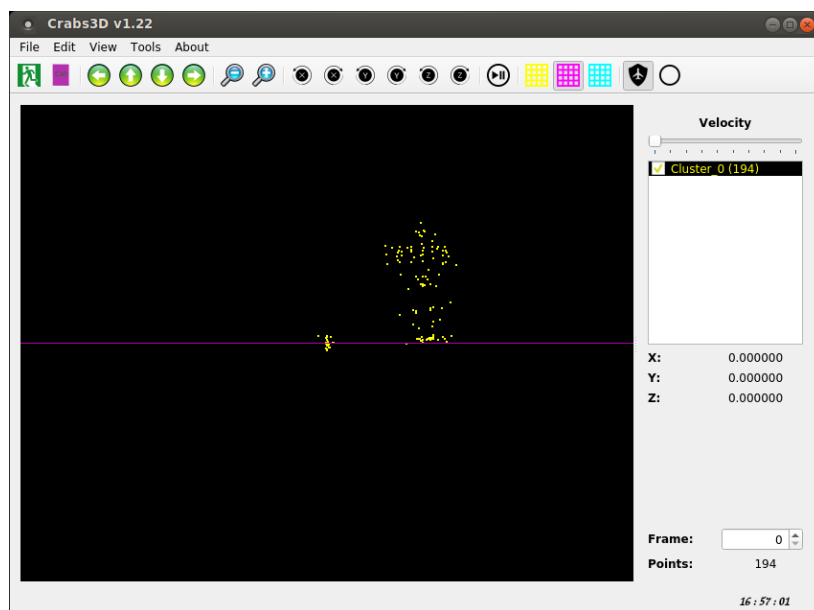
Στα Σχήματα 5.4 και 5.5 απεικονίζεται το νέφος σημείων σε ορθογραφική προβολή και σε προβολή υπό γωνία, μετά την επιτυχή ανάγνωση των παραμέτρων και των συντεταγμένων του αρχείου.

Η πρώτη παρατήρηση είναι πως το νέφος σημείων έχει αρκετό θόρυβο.

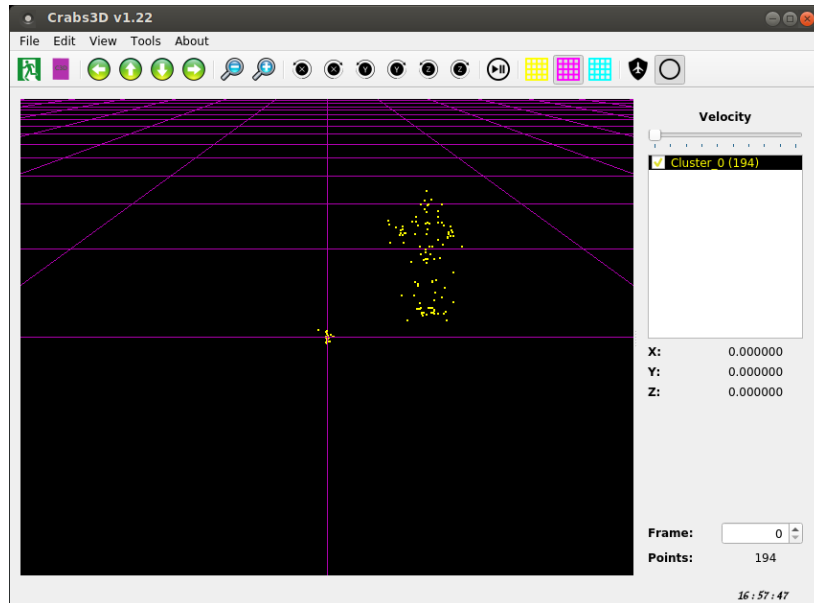
¹ Το πλήρες λογισμικό πακέτο βρίσκεται αναρτημένο στο Github: <https://github.com/JohnCrabs/Crabs3Dv122>



Σχήμα 5.3. Άνοιγμα αρχείου C3D.



Σχήμα 5.4. Απεικόνιση σκηνής δεδομένων σε ορθογραφική προβολή.



Σχήμα 5.5. Απεικόνιση σκηνής δεδομένων υπό στροφή.

5.2 Αφαίρεση του Θόρυβου από το Νέφος Σημείων

Μετά την ανάγνωση και οπτικοποίηση του νέφος σημείων, ακολουθεί η διαδικασία της αφαίρεσης του θορύβου και της διόρθωσης του. Έπειτα από παρατηρήσεις που έγιναν κατά την εναλλαγή των καρτέ, εντοπίστηκαν τριών ειδών προβλήματα στο νέφος σημείων [23].

1. Συσσωρευμένος θόρυβος γύρω από την αρχή των αξόνων σε όλα τα καρτέ του νέφους σημείων (συσσωρευμένος θόρυβος).
2. Στα τελευταία καρτέ εντοπίζονται σημεία τα οποία προηγουμένως αποτελούσαν σημεία του αντικειμένου, να κινούνται ελεύθερα στο χώρο δημιουργώντας σύγχυση στο νέφος σημείων (συγχεόμενος θόρυβος).
3. Η τρίτη κατηγορία προβληματικών σημείων, η οποία εμφανίζεται στα ίδια καρτέ με το συγχεόμενο θόρυβο, είναι σημεία τα οποία αποτελούσαν σε προηγούμενα καρτέ σημεία αντικειμένου, ενώ στα τελευταία καρτέ να μεταφέρονται στην αρχή των αξόνων με τιμή $(0, 0, 0)$. Ο θόρυβος αυτός αποδείχθηκε ως ο πιο δύσκολος να διορθωθεί, διότι τα σημεία αυτά δε θεωρούνται ούτε σημεία συσσώρευσης, αλλά ούτε και απομακρυσμένα σημεία. Επιπλέον εάν αφαιρεθούν από το νέφος σημείων, τότε αρχίζει να χάνεται η περιγραφή του αντικειμένου στα αρχικά καρτέ, κάτι το οποίο δεν είναι επιθυμητό.

Στις παρακάτω υποενότητες περιγράφονται οι τρόποι απομάκρυνσης και διόρθωσης των παραπάνω προβλημάτων.

5.2.1 Αφαίρεση του Συσσωρευμένου Θορύβου

Η απομάκρυνση του συσσωρευμένου θορύβου υλοποιείται από το αντίστροφο του ορισμού του.

Ως **συσσωρευμένο θόρυβο** η εφαρμογή αντιλαμβάνεται τα σημεία των οποίων οι συντεταγμένες δεν μετακινούνται σε απόσταση μεγαλύτερη από ένα δεδομένο κατώφλι, από το αρχικό καρτέ. Η συνθήκη αυτή πρέπει να ικανοποιείται για ένα δεδομένο ποσοστό καρτέ, ώστε ο θόρυβος να θεωρηθεί συσσωρευμένος.

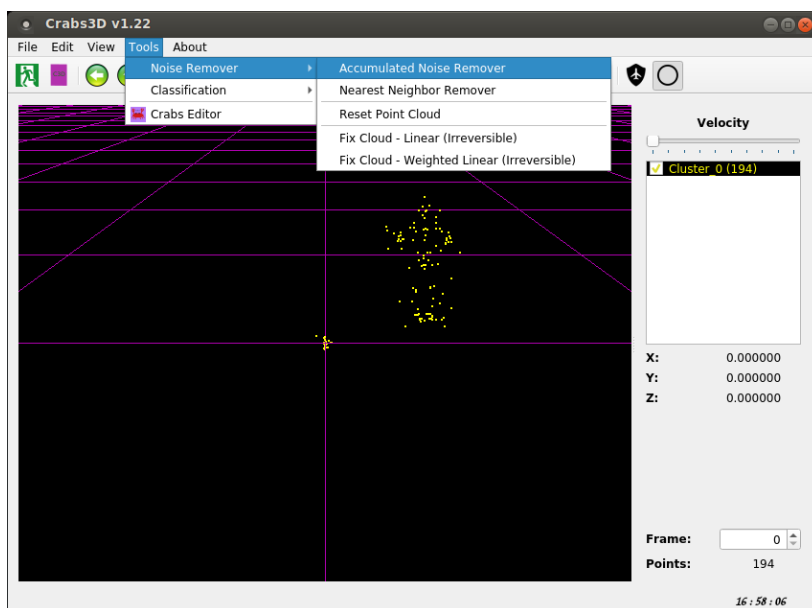
Συνεπώς, σημεία τα οποία πληρούν τις παραπάνω συνθήκες δεν προβάλλονται στην οθόνη έπειτα από την εκτέλεση του αντίστοιχου αλγορίθμου (Σχήμα 5.8). Στο Σχήμα 5.6 φαίνεται η τοποθεσία της εντολής, ενώ στο Σχήμα 5.7 φαίνεται το παράθυρο επιλογής των παραμέτρων του αλγορίθμου. Η πρώτη παράμετρος εισαγωγής είναι ένα κατώφλι που εκφράζει τη μέγιστη μετατόπιση – σε μέτρα – ενός σημείου από το πρώτο καρτέ, ώστε αυτό να θεωρηθεί σταθερό, ενώ η δεύτερη παράμετρος εκφράζει το ελάχιστο

ποσοστό των καρέ που πρέπει να ξεπερνά ένα σημείο για να θεωρηθεί σταθερό. Για το συγκεκριμένο νέφος σημείων, οι προκαθορισμένες τιμές αρκούν για την αφαίρεση του συσσωρευμένου θορύβου.

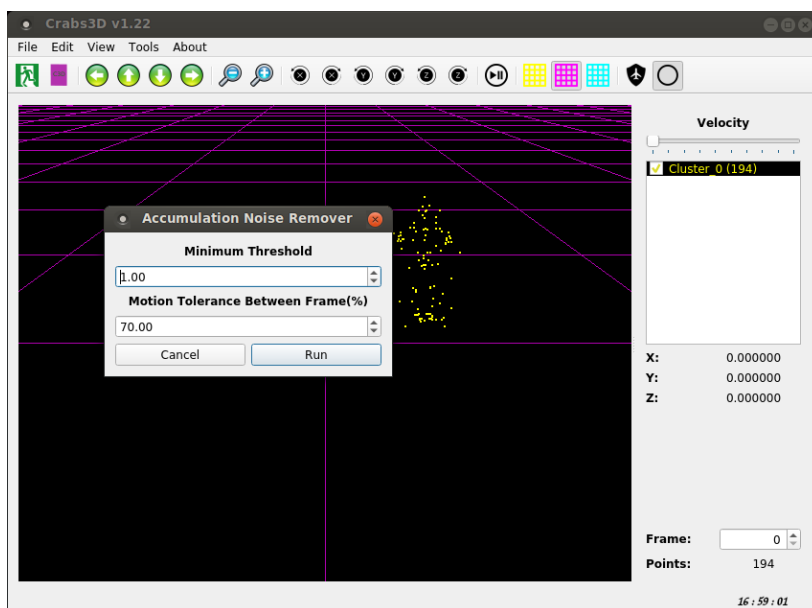
Μία δεύτερη παρατήρηση στο Σχήμα 5.8, είναι ότι ο αριθμός των σημείων που εμφανίζονται στην οθόνη, από 194 που ήταν στην αρχή, μετά την αφαίρεση του συσσωρευμένου θορύβου είναι 116. Αυτό σημαίνει ότι τα

$$194 - 116 = 78$$

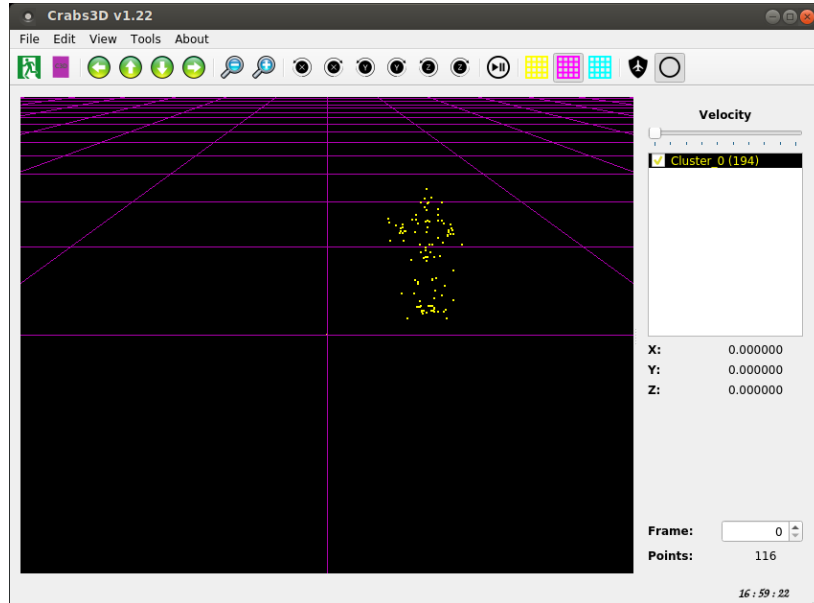
σημεία, είναι σημεία συσσωρευμένου θορύβου.



Σχήμα 5.6. Εντολή αφαίρεσης συσσωρευμένου θορύβου.



Σχήμα 5.7. Παράθυρο εισαγωγής παραμέτρων για την αφαίρεση του συσσωρευμένου θορύβου.



Σχήμα 5.8. Αποτέλεσμα του αλγορίθμου.

5.2.2 Αφαίρεση του Συγχεόμενου Θορύβου

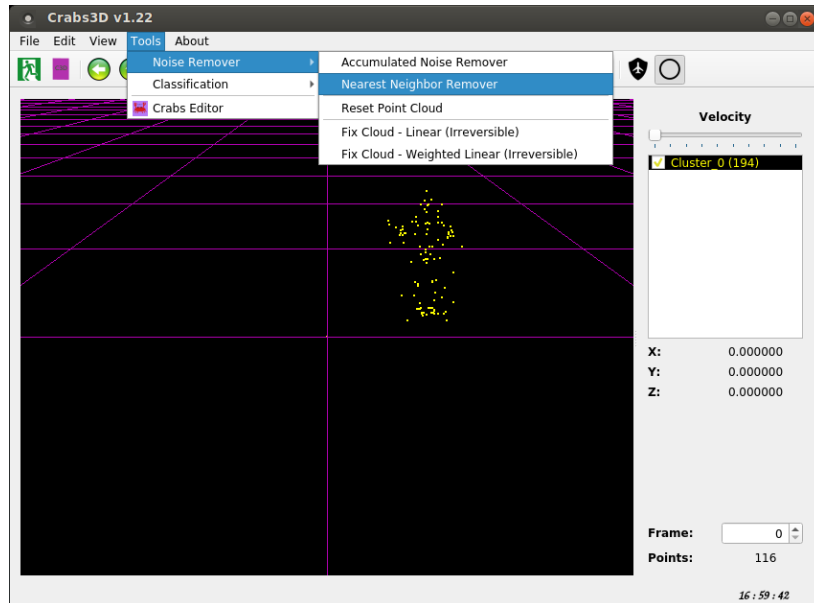
Ως **συγχεόμενος θόρυβος** θεωρούνται από την εφαρμογή τα σημεία εκείνα, τα οποία σε κάποια καρέ συμπεριφέρονται ως σημεία του αντικείμενου και σε άλλα συμπεριφέρονται ως θόρυβος. Στο Σχήμα 5.8 παρατηρείται επίσης ότι το αντικείμενο είναι υπερφορτωμένο με σημεία. Ο αλγόριθμος αυτός, όπως φαίνεται στο Σχήμα 5.11, έχει επίσης την ικανότητα να αποφορτίζει το αντικείμενο από επιπλέον σημεία.

Στο Σχήμα 5.9 απεικονίζεται η τοποθεσία της εντολής του αλγορίθμου, ενώ στο Σχήμα 5.7 απεικονίζεται το παράθυρο εισαγωγής των παραμέτρων. Ο συγκεκριμένος αλγόριθμος δέχεται δύο παραμέτρους, η πρώτη είναι η μέγιστη απόσταση που χρειάζεται να έχουν δύο σημεία, ώστε να θεωρηθεί πως ανήκουν στην ίδια γειτονιά, ενώ η δεύτερη παράμετρος είναι το ελάχιστο πλήθος των σημείων που αποτελούν μία γειτονιά (ανοχή). Εάν η γειτονιά ενός σημείου δεν πληρεί τον ελάχιστο αυτό αριθμό, σε ένα καρέ, τότε το σημείο αυτό θεωρείται θόρυβος σε όλα τα καρέ και απομακρύνεται από το νέφος σημείων (παύει να φαίνεται στην οθόνη).

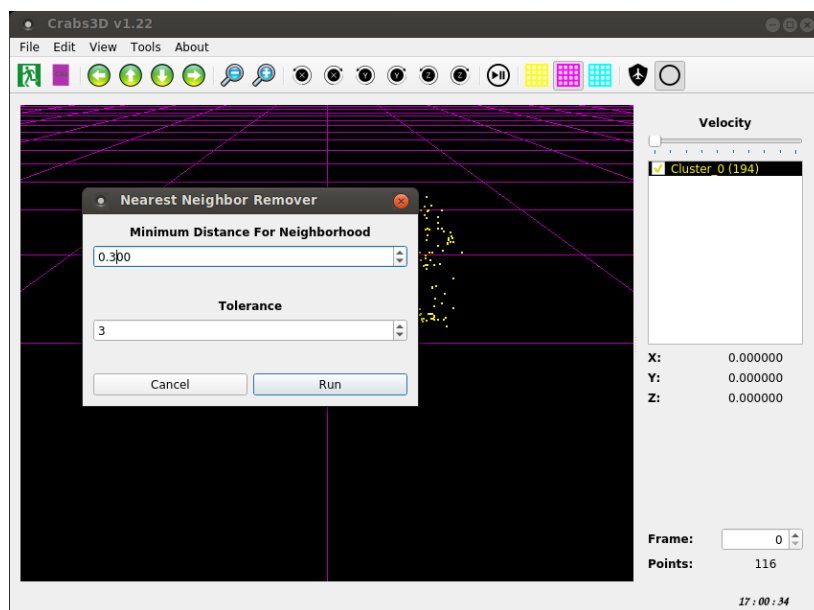
Το αποτέλεσμα του αλγορίθμου για τιμή απόστασης 0.3 και ελάχιστη γειτονιά τα 3 σημεία, φαίνεται στο Σχήμα 5.11. Παρατηρείται ότι μετά την εκτέλεση του αλγορίθμου απομακρύνθηκαν από το νέφος σημείων

$$116 - 74 = 42$$

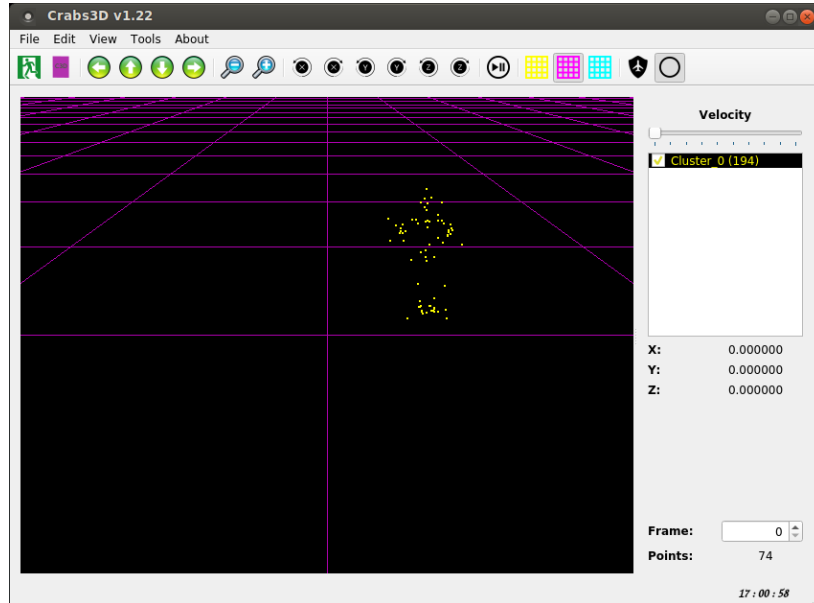
σημεία, των οποίων η συμπεριφορά θεωρήθηκε συγχεόμενος θόρυβος. Παρατηρείται επίσης ότι ο αλγόριθμος αδυνατεί να απομακρύνει σημεία θορύβου, όταν αυτά βρίσκονται πολύ κοντά μεταξύ τους ή έχουν ίδιες συντεταγμένες. Αναφέρεται επίσης ότι το αποτέλεσμα των παραπάνω δύο αλγορίθμων μπορεί να ανακληθεί με την εντολή Tools → Noise Remover → Reset Point Cloud.



Σχήμα 5.9. Εντολή αφαίρεσης συγχρόνου θορύβου.



Σχήμα 5.10. Παράθυρο εισαγωγής παραμέτρων για την αφαίρεση του συγχρόνου θορύβου.



Σχήμα 5.11. Αποτέλεσμα του αλγορίθμου.

5.2.3 Διόρθωση του Νέφους Σημείων

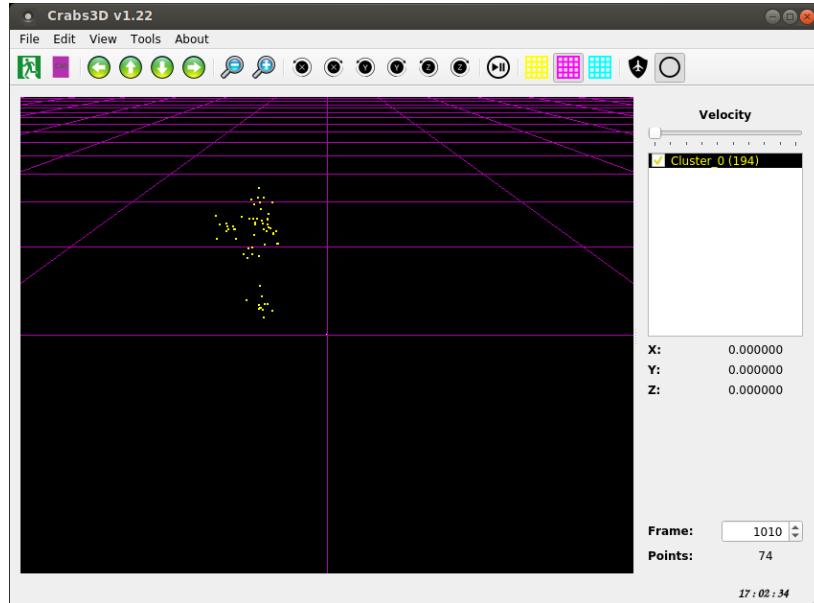
Την αδυναμία της πλήρης απομάκρυνσης του συγγεόμενου θορύβου, αναλαμβάνει να διορθώσει η εντολή Fix Cloud. Η εντολή αυτή αλλάζει τις συντεταγμένες των σημείων σε όλο το νέφος σημείων, συμπεριλαμβανομένων και των σημείων που δεν εμφανίζονται στην οθόνη. Επιπλέον επειδή αλλάζει τις συντεταγμένες των σημείων, θα πρέπει να εκτελείται μόνο όταν είναι απαραίτητη η ανάγκη.

Η εφαρμογή ως διόρθωση του νέφους σημείων αντιλαμβάνεται τη διόρθωση της τάσης στα σημεία όπου αρχικά κινούνται προς μία κατεύθυνση με σχεδόν σταθερό βήμα (τάση), ξαφνικά η τιμή μετατόπισής, μεταβάλλεται απότομα. Επιπλέον ο συγκεκριμένος αλγόριθμος κάνει την παραδοχή ότι η τάση είναι γραμμική και ακολουθεί την πορεία των δύο τελευταίων καρτέ, όπου το σημείο συμπεριφέρεται ως σημείο αντικειμένου.

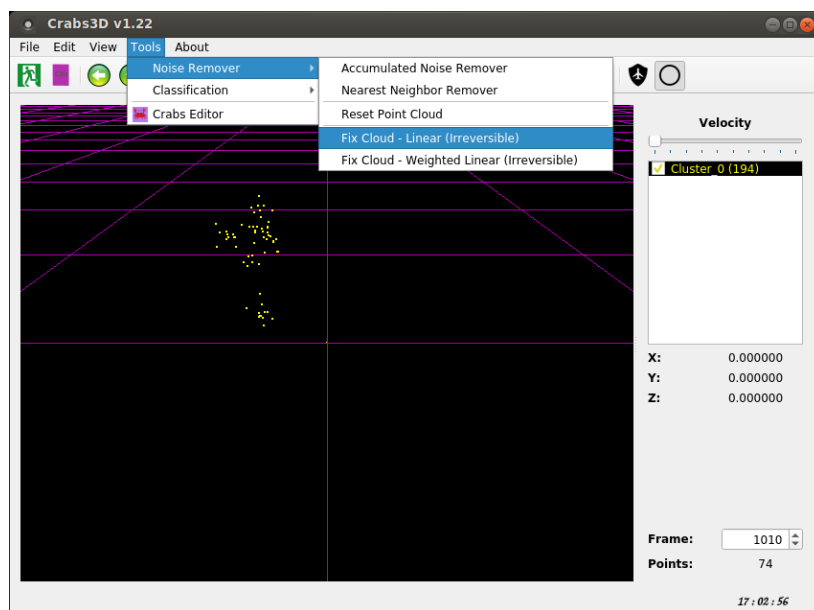
Στο Σχήμα 5.12 παρατηρείται ότι στο καρτέ 1010 το δεξί πόδι της φιγούρας εξαφανίζεται, ενώ στην αρχή των αξόνων σχεδιάζεται ένα σημείο. Ακολουθώντας το μηχανισμό του προηγούμενου αλγορίθμου, το σημείο αυτό θα έπρεπε να είχε θεωρηθεί σαν σημείο σύγχυσης από την εφαρμογή και δεν θα έπρεπε να απεικονίζεται. Η απεικόνισή του, σημαίνει ότι ένα πλήθος σημείων του αντικειμένου κατά την καταγραφή των δεδομένων, πιθανόν να μην ήταν ορατά στο δέκτη και γι' αυτό το λόγο ο δέκτης θεώρησε πως έχουν συντεταγμένες (0, 0, 0).

Εκτελώντας τη διαδικασία της διόρθωσης του νέφους σημείων, όπως φαίνεται στο Σχήμα 5.13, τα σημεία αυτά λαμβάνουν συντεταγμένες. Οι συντεταγμένες αυτές δεν είναι οι πραγματικές, αλλά μπορεί να θεωρηθεί πως προσεγγίζουν το πραγματικό κεντροειδές της κίνησης, του οποίου η εύρεση είναι το ζητούμενο.

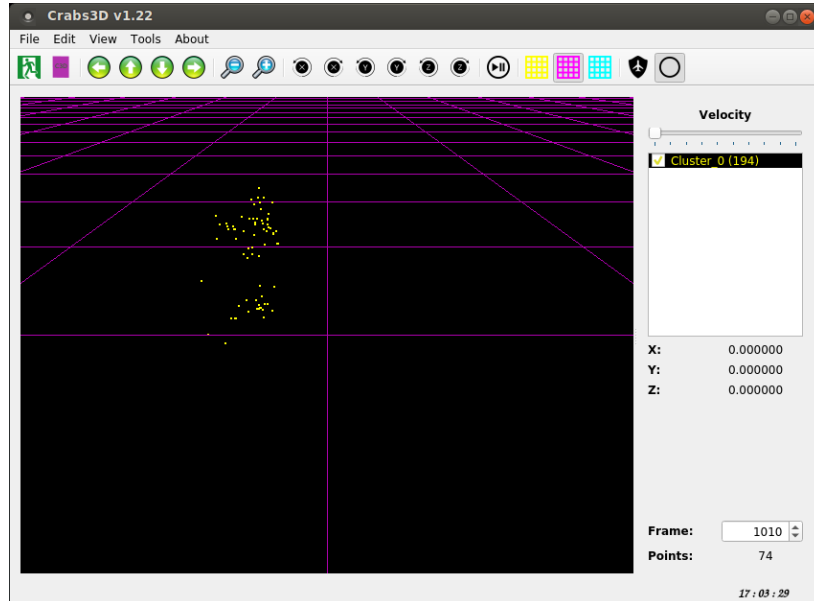
Στο Σχήμα 5.14 απεικονίζεται το αποτέλεσμα του αλγορίθμου. Σημειώνεται πως επειδή ο αλγόριθμος θεωρεί την τάση γραμμική με βήμα την τελευταία έγκυρη τάση, αν τα σημεία συμπεριφέρονται για πολλά συνεχόμενα καρτέ ως θόρυβος, τότε από ένα σημείο και μετά τα σημεία αυτά θα αποκλίνουν από το αντικείμενο με σταθερό βήμα.



Σχήμα 5.12. Στο καρέ 1010 στην αρχή των αξόνων παρατηρείται ένα κίτρινο σημείο.



Σχήμα 5.13. Εντολή Απλής Γραμμικής Διόρθωσης Σημείων.



Σχήμα 5.14. Αποτέλεσμα του αλγορίθμου.

5.3 Ταξινόμηση - Μέθοδος K-Means

Στο Σχήμα 5.15, απεικονίζεται το καρέ 242, στο οποίο διακρίνονται με ευκρίνεια τα σημεία που περιγράφουν τα χαρακτηριστικά του ανθρώπινου σκελετού, ενώ στα Σχήματα 5.16 και 5.17 απεικονίζεται η επιλογή των παραμέτρων για την εκτέλεση του αλγορίθμου ταξινόμησης K-Means [12].

Οι τιμές των παραμέτρων του K-Means που καλείται να επιλέξει ο χρήστης είναι το καρέ της ταξινόμησης (καρέ 242) και ο αριθμός των κλάσεων ταξινόμησης (8 κλάσεις στο συγκεκριμένο παράδειγμα). Όπως είναι γνωστό, ο αλγόριθμος K-Means προκειμένου να ταξινομήσει σωστά τις κλάσεις, πρέπει να έχει καλές αρχικές τιμές, οι οποίες στον κώδικα που προγραμματίστηκε δίνονται ως εξής:

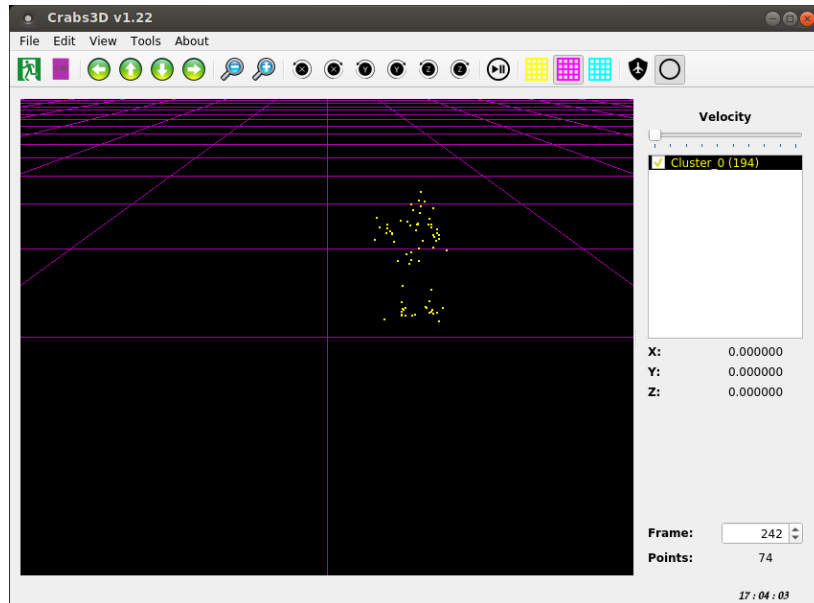
- Για την κλάση 1 το σημείο με την ελάχιστη συντεταγμένη X.
- Για την κλάση 2 το σημείο με την μέγιστη συντεταγμένη X.
- Για την κλάση 3 το σημείο με την ελάχιστη συντεταγμένη Y.
- Για την κλάση 4 το σημείο με την μέγιστη συντεταγμένη Y.
- Για την κλάση 5 το σημείο με την ελάχιστη συντεταγμένη Z.
- Για την κλάση 6 το σημείο με την μέγιστη συντεταγμένη Z.
- Για την κλάση 7 το μέσο των σημείων των κλάσεων 1 και 2.
- Για την κλάση 8 το μέσο των σημείων των κλάσεων 3 και 4.
- Για την κλάση 9 το μέσο των σημείων των κλάσεων 5 και 6.

Μετά την εισαγωγή των παραμέτρων, το επόμενο βήμα είναι η εκτέλεση του αλγορίθμου K-Means (Σχήμα 5.18). Το αποτέλεσμα του αλγορίθμου απεικονίζεται στο Σχήμα 5.19.

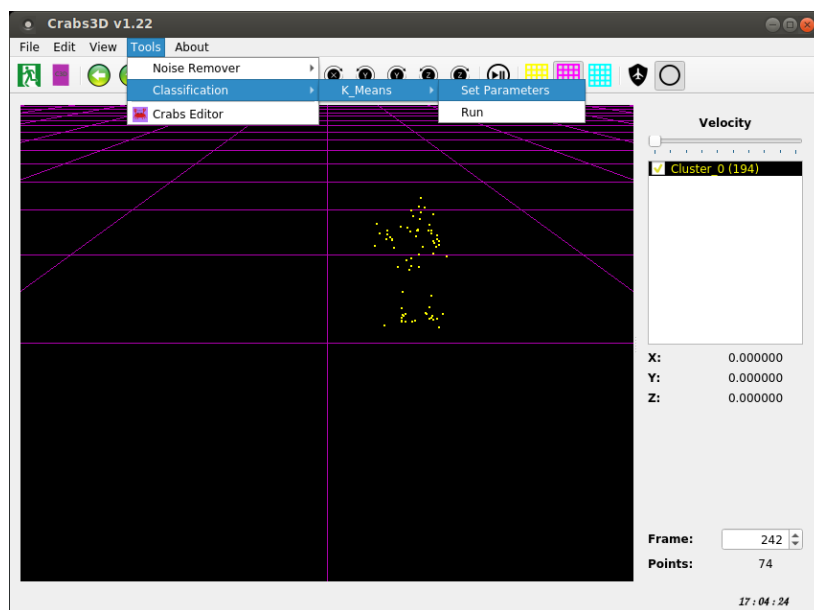
Από την εκτέλεση του αλγορίθμου παρατηρούνται τα εξής:

- Η Κλάση 0 (κλάση θορύβου) περιέχει όσα σημεία έχουν αφαιρεθεί από τους προηγούμενους αλγορίθμους.
- Τα χρώματα των κλάσεων είναι τυχαία.
- Η ταξινόμηση των σημείων σε κλάσεις είναι ικανοποιητική.

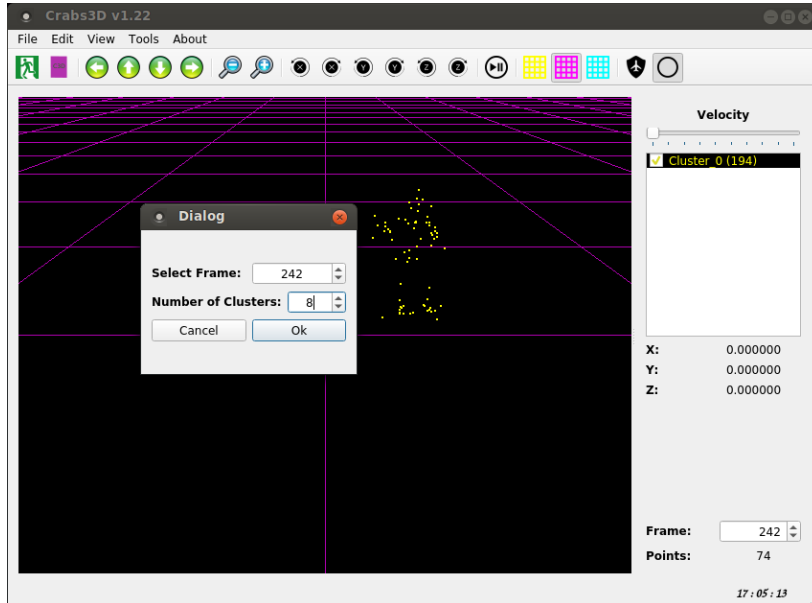
Η ουσιαστική επεξεργασία των δεδομένων ολοκληρώνεται σε αυτό το σημείο. Το επόμενο βήμα είναι η αναγνώριση των σημείων του σκελετού που περιγράφουν οι κλάσεις. Αυτό επιτυγχάνεται με τη φωτοερμηνεία.



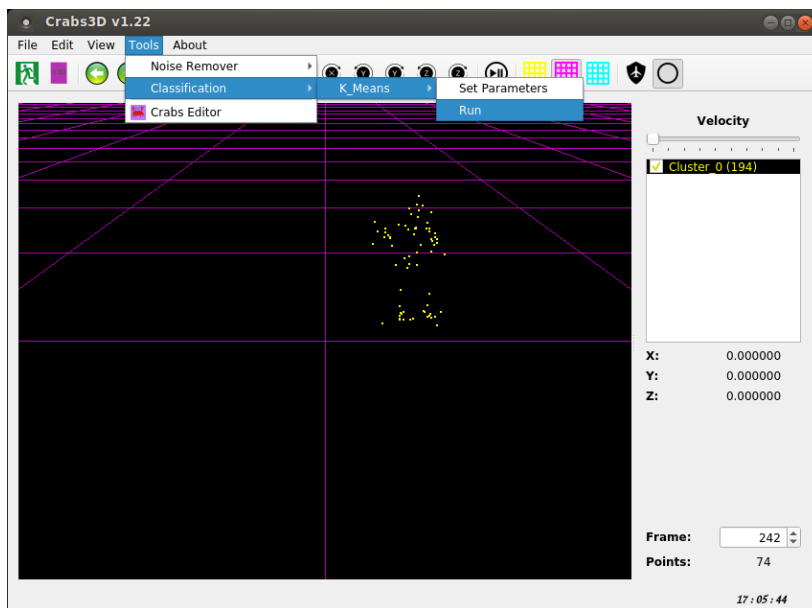
Σχήμα 5.15. Στο καρέ 242 διακρίνονται τα χαρακτηριστικά του σκελετού.



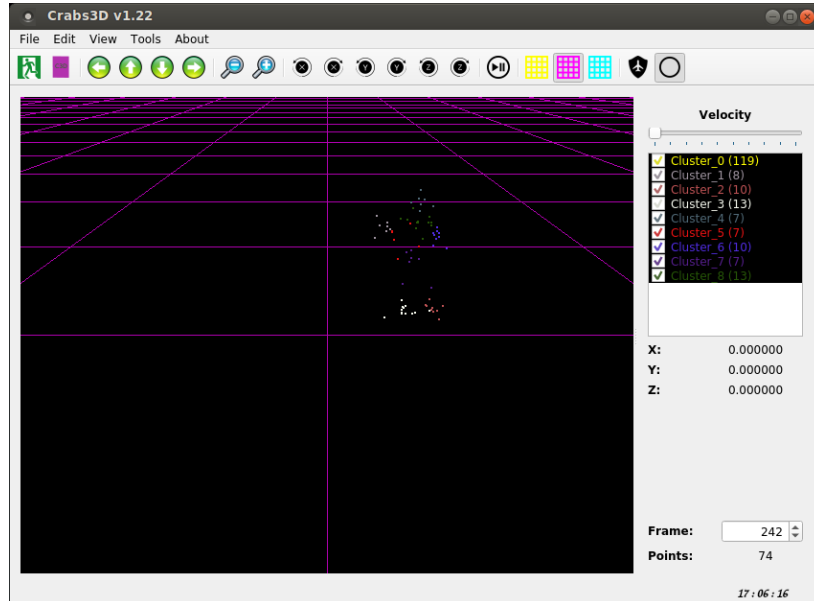
Σχήμα 5.16. Εισαγωγή των παραμέτρων του αλγορίθμου K-Means.



Σχήμα 5.17. Παράθυρο εισαγωγής παραμέτρων.



Σχήμα 5.18. Εκτέλεση αλγορίθμου.



Σχήμα 5.19. Αποτέλεσμα του αλγορίθμου.

5.4 Αναγνώριση των Κλάσεων

Για τη διαχείριση των κλάσεων, δημιουργήθηκε ειδικό παράθυρο, στο οποίο ο χρήστης έχει τη δυνατότητα να τροποποιεί το όνομα μίας κλάσης, το χρώμα της και το μέγεθος απεικόνισης των σημείων της κλάσης. Το παράθυρο αυτό απεικονίζεται στο Σχήμα 5.20. Για να ανοίξει ο χρήστης το παράθυρο μπορεί είτε να κάνει διπλό κλικ πάνω στην κλάση που θέλει να τροποποιήσει, είτε να επιλέξει την εντολή του παραθύρου από τη γραμμή εντολών Edit → Cluster.

Η προτεινόμενη διαδικασία αναγνώρισης των κλάσεων χωρίζεται σε δύο στάδια:

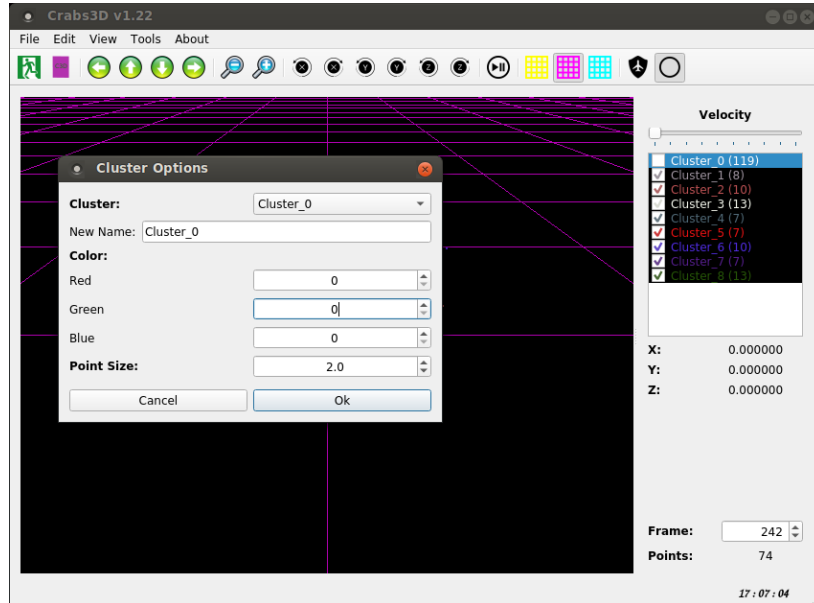
1. Αλλαγή χρώματος των κλάσεων.
2. Αναγνώριση της κλάσης.

Πραγματοποιώντας τα παραπάνω βήματα και όπως παρατηρείται και από τα Σχήματα 5.21 και 5.22, αναγνωρίζονται τα εξής μέλη του σκελετού.

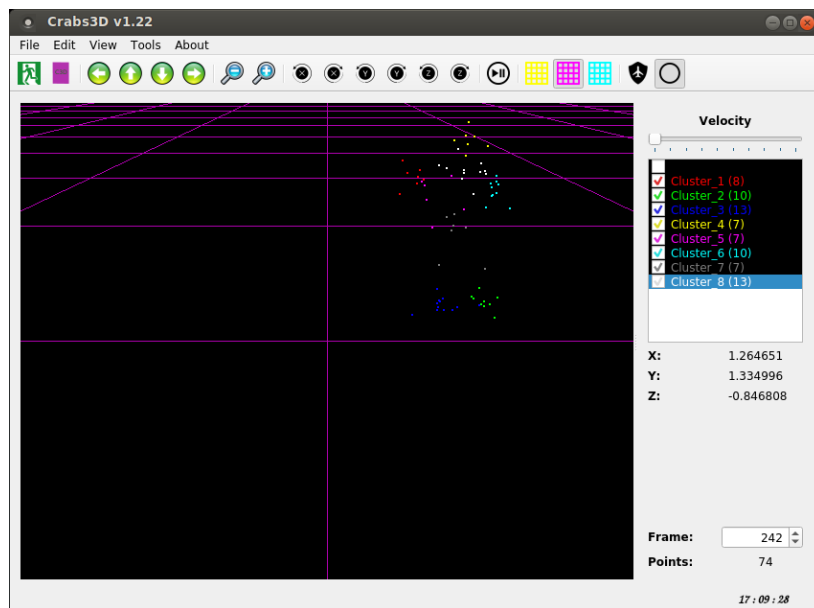
1. Κλάση 1 → Κόκκινο Χρώμα → Απεικονίζει το δεξί χέρι.
2. Κλάση 2 → Πράσινο Χρώμα → Απεικονίζει το αριστερό πόδι.
3. Κλάση 3 → Μπλε Χρώμα → Απεικονίζει το δεξί πόδι.
4. Κλάση 4 → Κίτρινο Χρώμα → Απεικονίζει το κεφάλι.
5. Κλάση 5 → Ματζέντα Χρώμα → Θεωρείται θόρυβος.
6. Κλάση 6 → Κυανό Χρώμα → Απεικονίζει το αριστερό χέρι.
7. Κλάση 7 → Γκρι Χρώμα → Απεικονίζει τη λεκάνη.
8. Κλάση 8 → Άσπρο Χρώμα → Απεικονίζει το στήθος.

Από την ταξινόμηση με 8 κλάσεις αναγνωρίστηκαν τα 7 από τα 11 σημεία του σκελετού. Τα υπόλοιπα 4 σημεία είναι σημεία τα οποία περιγράφουν τα γόνατα και τους ώμους. Τα σημεία αυτά μπορούν να ληφθούν από τις κλάσεις «Λεκάνη» και «Στήθος», επιλέγοντας σημεία του αρχικού νέφους – π.χ. για τα γόνατα θα ληφθούν τα σημεία εκείνα που έχουν τη μικρότερη συντεταγμένη Y στην κλάση «Λεκάνη».

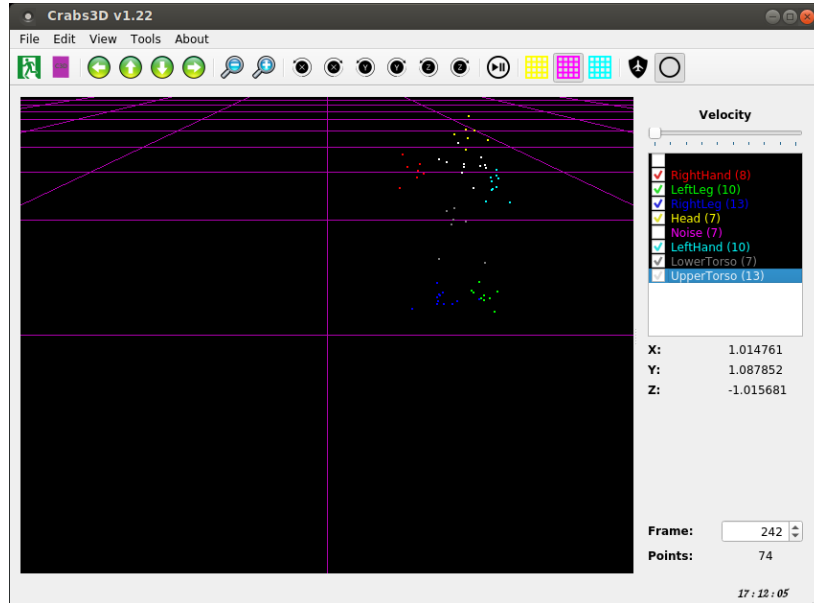
Τέλος αναφέρεται, ότι κατά τα πειράματα που έγιναν στο νευρωνικό δίκτυο, οι ονομασίες των σημείων του σκελετού είναι διαφορετικές από αυτές που φαίνονται στο Σχήμα 5.8.



Σχήμα 5.20. Παράθυρο κλάσεων.



Σχήμα 5.21. Αλλαγή χρώματος των κλάσεων.



Σχήμα 5.22. Αναγνώριση των κλάσεων.

5.5 Εξαγωγή των Αποτελεσμάτων

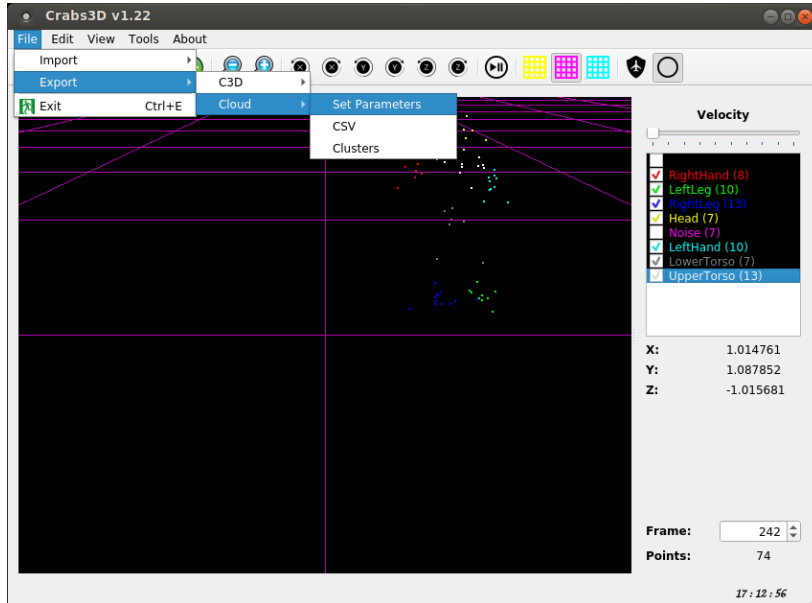
Το τελευταίο στάδιο για να ολοκληρωθεί η διαδικασία είναι η εξαγωγή των αποτελεσμάτων. Η εφαρμογή δίνει την επιλογή στον χρήστη να αποφασίσει πόσα και ποια καρτέκια θέλει να κρατήσει για περαιτέρω επεξεργασία ή μελλοντική χρήση. Το αρχείο αποθήκευσης των αποτελεσμάτων έχει μορφοποίηση CSV και ο χρήστης επιλέγει τον τρόπο διαχωρισμού των δεδομένων.

Στο Σχήμα 5.23 απεικονίζεται η τοποθεσία της εντολής για την επιλογή των παραμέτρων, ενώ στο Σχήμα 5.24 το παράθυρο επιλογής των παραμέτρων. Μόλις ο χρήστης πατήσει Ok, οτιδήποτε αποθηκεύσει θα περιλαμβάνει τα πρώτα 1000 καρτέκια και οι τιμές θα διαχωρίζονται με το ελληνικό ερωτηματικό (;).

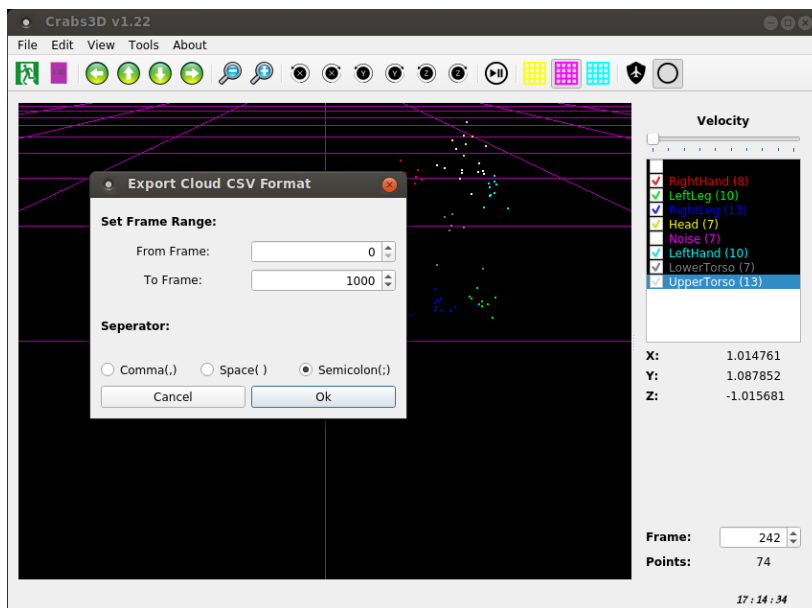
Στα Σχήματα 5.25 και 5.26 απεικονίζεται η εξαγωγή του νέφους σημείων σε αρχείο μορφοποίησης CSV. Το αρχείο CSV θα περιέχει τις συντεταγμένες των σημείων που εμφανίζονται στην οθόνη, για τα πρώτα 1000 καρτέκια. Πρέπει να σημειωθεί εδώ, ότι στο μονοπάτι εξαγωγής πρέπει να δοθεί η προέκταση .csv.

Στα Σχήματα 5.27 και 5.28 απεικονίζεται η εξαγωγή των αρχείων κλάσεων του νέφους σημείων. Η εφαρμογή για την εξαγωγή των κλάσεων χρειάζεται το φάκελο αποθήκευσης (μονοπάτι) και ένα όνομα δίχως την προέκταση .csv. Για κάθε μία κλάση θα δημιουργήσει ένα ξεχωριστό αρχείο και θα προτείνει στο όνομα που δόθηκε από το χρήστη το όνομα της κλάσης και της προέκτασης .csv, ενώ θα αποθηκευτούν επίσης σε ένα ξεχωριστό αρχείο CSV τα κεντροειδή των κλάσεων.

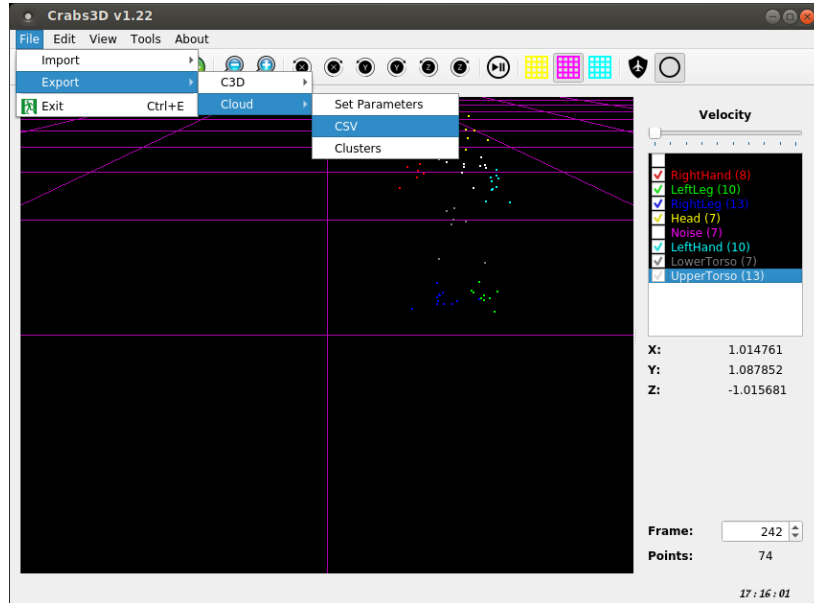
Μετά την εξαγωγή των αποτελεσμάτων, όπως περιγράφεται παραπάνω, ο τελικός φάκελος πρέπει να περιλαμβάνει τα αρχεία που φαίνονται στο Σχήμα 5.29.



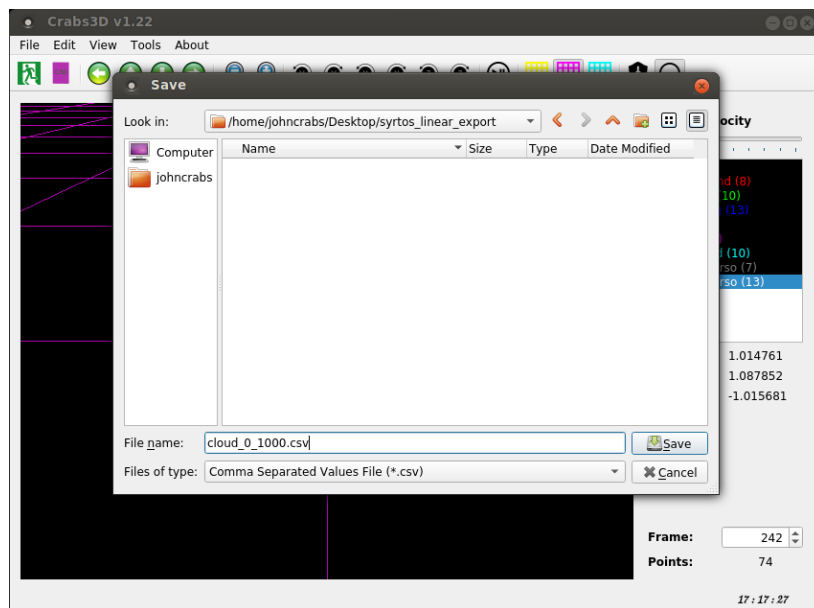
Σχήμα 5.23. Επιλογή παραμέτρων εξαγωγής των αποτελεσμάτων.



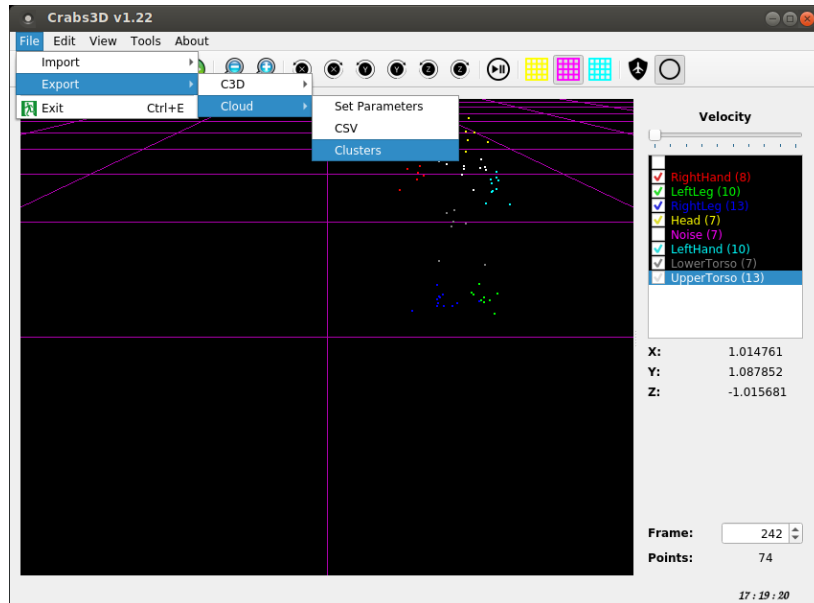
Σχήμα 5.24. Παράθυρο επιλογής των παραμέτρων.



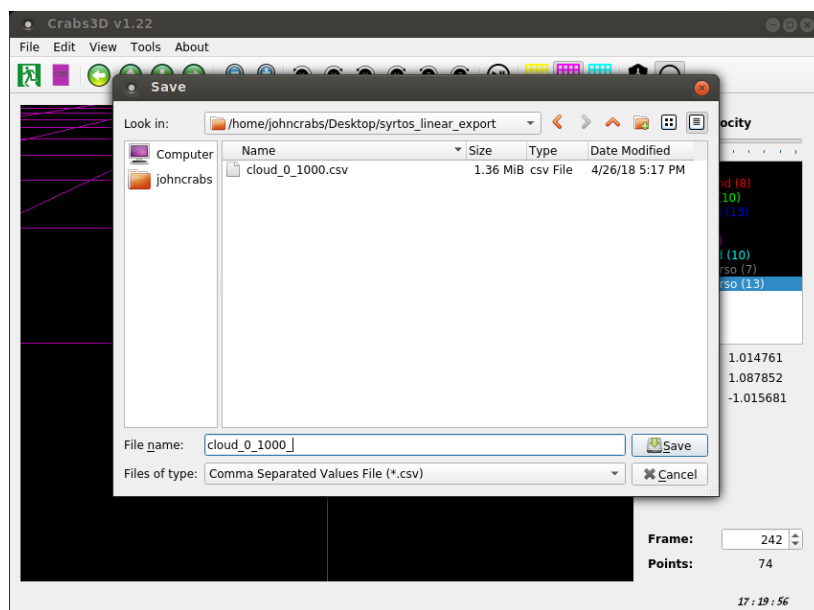
Σχήμα 5.25. Εξαγωγή των σημείων του νέφους σημείων που εμφανίζονται στην οθόνη.



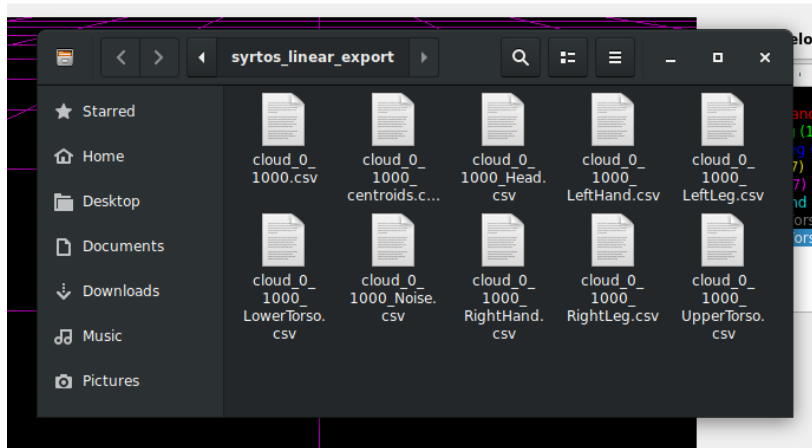
Σχήμα 5.26. Επιλογή τοποθεσίας αποθήκευσης.



Σχήμα 5.27. Εξαγωγή των αρχείων κλάσεων.



Σχήμα 5.28. Επιλογή τοποθεσίας αποθήκευσης.



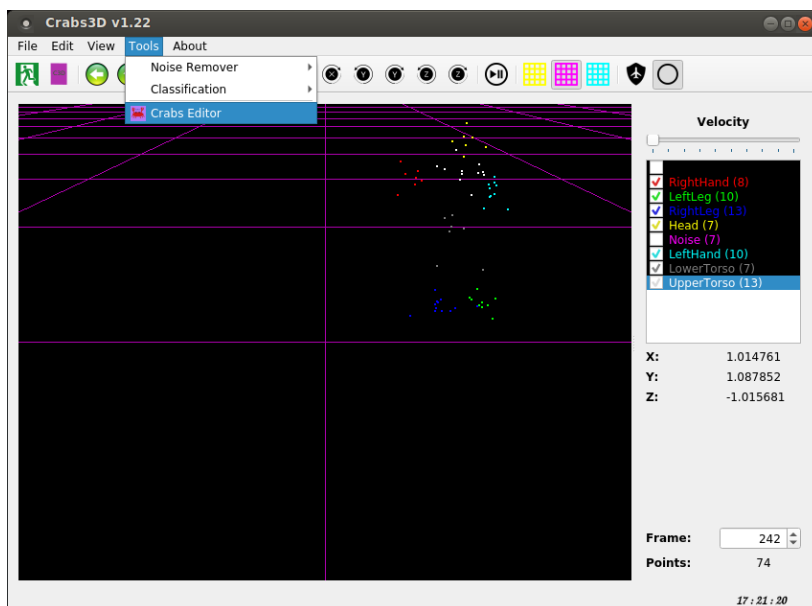
Σχήμα 5.29. Ο φάκελος μετά την αποθήκευση των αρχείων.

5.6 Crabs Editor

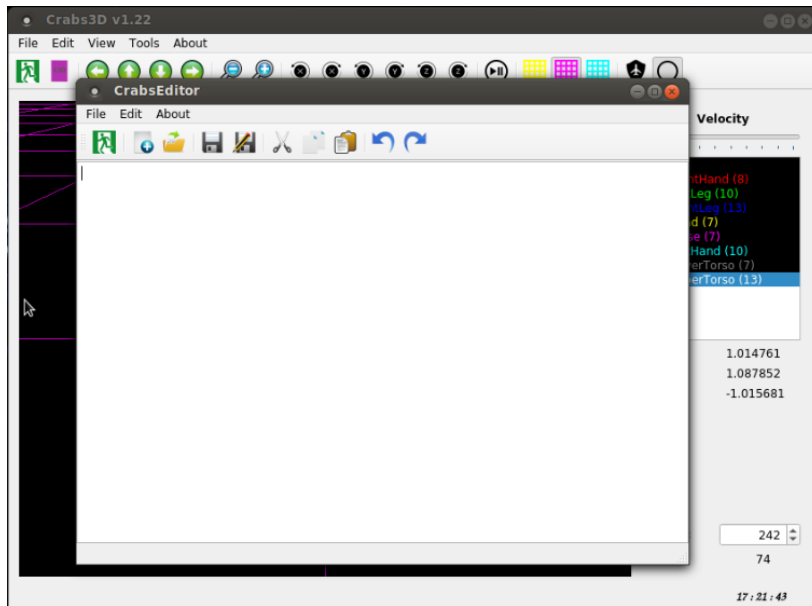
Η εφαρμογή εκτός από τις εντολές διαχείρισης κι επεξεργασίας του νέφους σημείων, περιλαμβάνει κι έναν ελαφρύ κειμενογράφο για να μπορεί ο χρήστης να ανοίγει τα αρχεία που αποθήκευσε. Η εφαρμογή δίνει και τη δυνατότητα αποθήκευσης κάθε δομής του αρχείου C3D και ο κειμενογράφος είναι ιδιαίτερα χρήσιμος για την ανάγνωση των παραμέτρων.

Επειδή τόσο η περιγραφή του κειμενογράφου, όσο και η παρατήρηση των δομών του αρχείου C3D, ξεπερνούν τα όρια της διπλωματικής, δε θα δοθούν περισσότερες λεπτομέρειες. Η αναφορά αυτή γίνεται μόνο για να δηλωθεί η ύπαρξη του κειμενογράφου εντός της εφαρμογής.

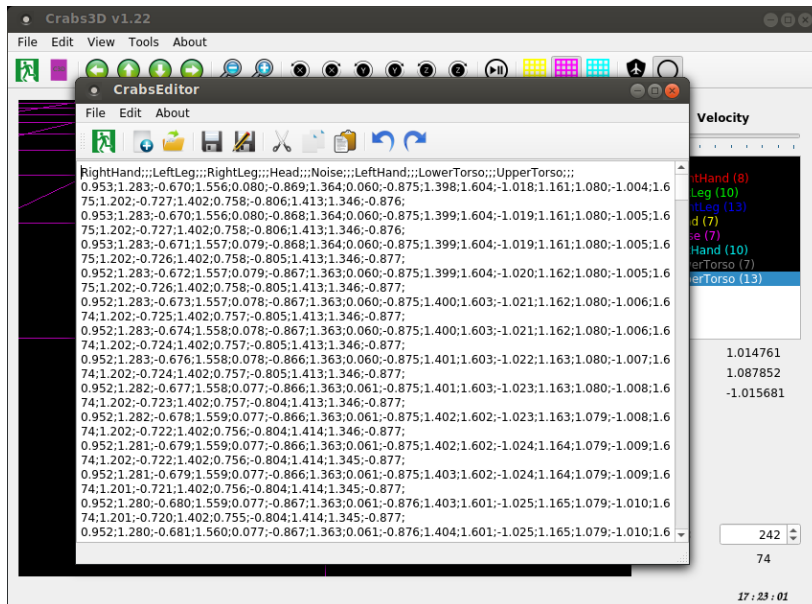
Στο Σχήμα 5.30 απεικονίζεται ο τρόπος με τον οποίο ανοίγει ο κειμενογράφος με όνομα CrabsEditor. Στο Σχήμα 5.31 απεικονίζεται ο κειμενογράφος, μόλις ανοίξει και στο Σχήμα 5.32 φαίνεται ο κειμενογράφος, μετά το άνοιγμα του αρχείου CSV με τα κεντροειδή των κλάσεων ταξινόμησης, όπως αυτό δημιουργήθηκε και αποθηκεύτηκε με τις διαδικασίες που περιγράφησαν παραπάνω.



Σχήμα 5.30. Άνοιγμα του κειμενογράφου CrabsEditor.



Σχήμα 5.31. Κειμενογράφος CrabsEditor.



Σχήμα 5.32. Ανάγνωση του αρχείου CSV των κεντροειδών.

Κεφάλαιο 6

Εκπαίδευση Νευρωνικού Δικτύου

Το επόμενο βήμα αφότου δημιουργηθεί ο σκελετός είναι η εκπαίδευση ενός νευρωνικού δικτύου. Η διαδικασία αυτή πραγματοποιήθηκε σε περιβάλλον MATLAB και η μέθοδος εκπαίδευσης που επιλέχθηκε είναι η Convolution Neural Network (CNN). Στην παρούσα εργασία επιλέχθηκε Συνελεκτικό Νευρωνικό Δίκτυο (δνολυτιοναλ Νευραλ Νετωρκς) που ανήκει στην βαθιά μάθηση [22], [32].

6.1 Δημιουργία Δεδομένων

Στο Κεφάλαιο 5 έγινε περιγραφή του λογισμικού οπτικοποίησης και επεξεργασίας ενός νέφους σημείων. Χρησιμοποιώντας τις παραπάνω μεθόδους, δημιουργήθηκαν τα εξής αρχεία για την εκπαίδευση του νευρωνικού δικτύου.

1. Αρχεία εισαγωγής και εξαγωγής με τα 1000 πρώτα καρέ:
 - (α') Αρχείο εισαγωγής (input) για εκπαίδευση αποτελούμενο από 50 σημεία.
 - (β') Αρχείο εξαγωγής (output) για εκπαίδευση αποτελούμενο από τα 11 σημεία του σκελετού.
 - (γ') Αρχείο εισαγωγής (input) ελέγχου αποτελούμενο από 50 σημεία.
 - (δ') Αρχείο εξαγωγής (output) ελέγχου αποτελούμενο από τα 11 σημεία του σκελετού.
2. Αρχεία εισαγωγής και εξαγωγής με τα 500 πρώτα καρέ:
 - (α') Αρχείο εισαγωγής (input) για εκπαίδευση αποτελούμενο από 50 σημεία.
 - (β') Αρχείο εξαγωγής (output) για εκπαίδευση αποτελούμενο από τα 11 σημεία του σκελετού.
 - (γ') Αρχείο εισαγωγής (input) ελέγχου αποτελούμενο από 50 σημεία.
 - (δ') Αρχείο εξαγωγής (output) ελέγχου αποτελούμενο από τα 11 σημεία του σκελετού.

Τα παραπάνω αρχεία προέκυψαν από την επεξεργασία δύο διαφορετικών δειγμάτων περιγραφής του Συρτού χορού.

6.2 Περιγραφή της Μεθόδου Convolution Neural Network

Ένα νευρωνικό δίκτυο CNN αποτελείται από τύπους νευρωνικών επιπέδων [22]:

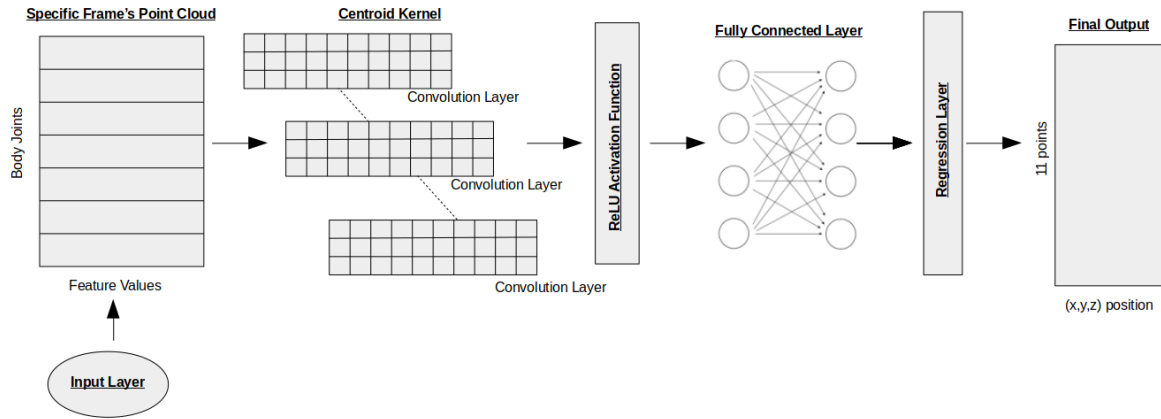
1. **Convolutional Layers:** Στα επίπεδα συνέλιξης (convolutional layers), η μέθοδος CNN [31] χρησιμοποιεί διάφορους πυρήνες για να περιστρέψει ολόκληρη την εικόνα καθώς και τους ενδιάμεσους χάρτες χαρακτηριστικών, δημιουργώντας διάφορους χάρτες χαρακτηριστικών. Λόγω των πλεονεκτημάτων της λειτουργίας συνέλιξης, αρκετά πειράματα την πρότειναν ως υποκατάστατο πλήρως συνδεδεμένων επιπέδων με στόχο τη μείωση του χρόνου εκμάθησης του νευρωνικού δικτύου [24], [25], [26]. Προσαρμοστικές τεχνικές εκπαίδευσης έχουν επίσης προταθεί [9], [11], [10].
2. **ReLU Layers:** Οι διορθωμένες γραμμικές μονάδες (rectified linear units – ReLU) εφαρμόζονται στη συνάρτηση μη κορεσμού ενεργοποίησης, όπως φαίνεται στην παρακάτω εξίσωση.

$$f(x) = \max(0, x) \quad (6.1)$$

Η εξίσωση αυτή αυξάνει τις μη γραμμικές ιδιότητες όλου του δικτύου, χωρίς να επηρεάζει τα ευαίσθητα πεδία του επιπέδου συνέλιξης.

3. **Fully Connected Layers:** Ακολουθώντας αρκετά επίπεδα συνέλιξης και διορθωμένων γραμμικών μονάδων, οι υψηλού επιπέδου συχνότητες του νευρωνικού δικτύου εκτελούνται μέσω των πλήρη συνδεδεμένων επιπέδων. Οι νευρώνες σε ένα πλήρες συνδεδεμένο επίπεδο ενεργοποιούνται με όλα τα αποτελέσματα των προηγούμενων επιπέδων. Η ενεργοποίηση τους μπορεί επομένως να υπολογιστεί με πολλαπλασιασμό πινάκων που έπεται τη μετατόπιση μεροληψίας. Τελικά οι 2D πίνακες χαρακτηριστικών μετατρέπονται σε ένα διάνυσμα χαρακτηριστικών 1D. Ο παραγόμενος φορέας είτε μπορεί να τροφοδοτηθεί σε ένα συγκεκριμένο αριθμό κατηγοριών για να ταξινομηθεί, είτε μπορεί να θεωρηθεί ως ένα διάνυσμα χαρακτηριστικών για περαιτέρω επεξεργασία.

Το Σχήμα 6.1 απεικονίζει την τοπολογία της μεθόδου CNN, όπως αυτή περιγραφτήκε παραπάνω.



Σχήμα 6.1. Τοπολογία της μεθόδου Convolutional Neural Network.

6.3 Περιγραφή των Μεγεθών Αξιολόγησης

Για την αξιολόγηση των αποτελεσμάτων χρησιμοποιήθηκαν οι παρακάτω δείκτες:

1. **Minimum Deviation (MIN):** Εφράζεται από την ελάχιστη τιμή σφάλματος, που προκύπτει από την απόλυτη τιμή της αφαίρεσης της πραγματικής τιμής συντεταγμένης από την εκτιμώμενη τιμή συντεταγμένης.
2. **Maximum Deviation (MAX):** Εφράζεται από τη μέγιστη τιμή σφάλματος, που προκύπτει από την απόλυτη τιμή της αφαίρεσης της πραγματικής τιμής συντεταγμένης από την εκτιμώμενη τιμή συντεταγμένης.
3. **Mean Absolute Estimation (MAE):** Ορίζεται ως:

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - m(X)| \quad (6.2)$$

Όπου x_i είναι η εκτιμώμενη τιμή και $m(X)$ είναι η κεντρική τάση.

4. **Root Mean Square Estimation (RMSE):** Ορίζεται ως:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_i - m(X))^2}{n}} \quad (6.3)$$

Όπου x_i είναι η εκτιμώμενη τιμή και $m(X)$ είναι η κεντρική τάση.

6.4 Πειραματικά Αποτελέσματα

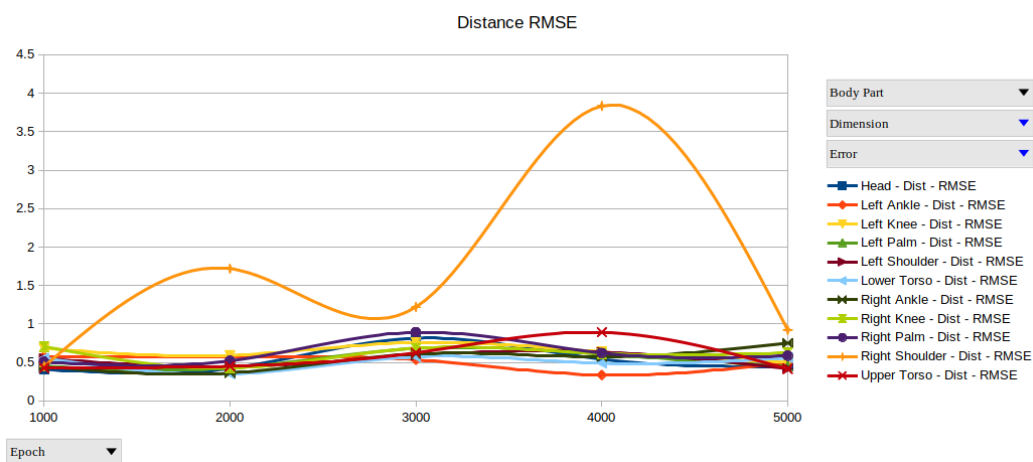
Προκειμένου να γίνει εκτίμηση του σκελετού, πραγματοποιήθηκαν πέντε πειράματα με τα δεδομένα των 1000 καρτέ και ένα πείραμα με τα δεδομένα των 500 καρτέ στο καλύτερο εκτιμώμενο αποτέλεσμα – που προέκυψε από τα πειράματα των 1000 καρτέ. Στον Πίνακα 6.1 απεικονίζονται τα αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν.

Πίνακας 6.1. Τα αποτελέσματα των πειραμάτων.

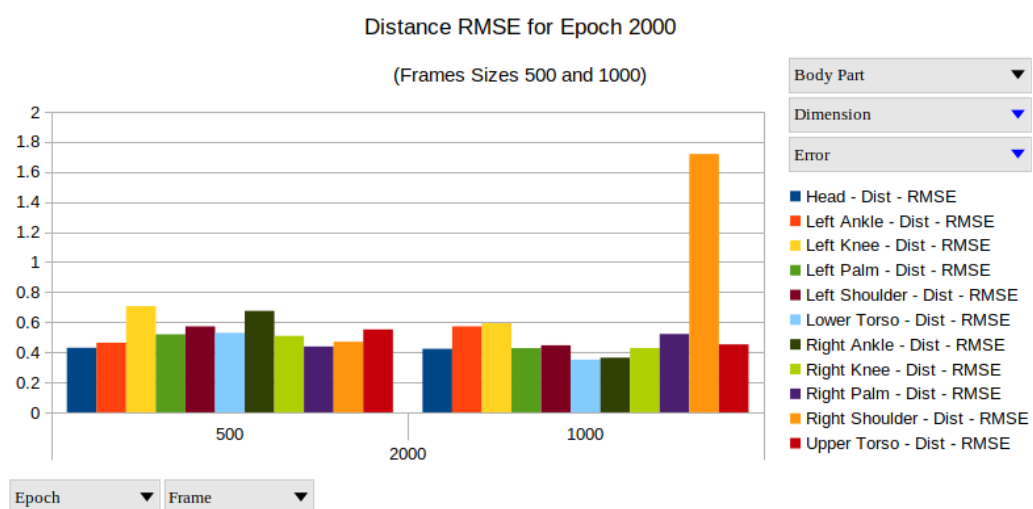
Training Test for 1000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0903	0.4682	0.3189	0.3325	0.2612	0.4391	0.3477	0.3500	0.0003	0.2938	0.1346	0.1576	0.4906	0.5078
Left Palm	0.1120	0.5133	0.3088	0.3216	0.1170	0.4458	0.2655	0.2742	0.0004	0.2866	0.1372	0.1557	0.4297	0.4504
Head	0.1944	0.5034	0.3793	0.3876	0.0002	0.1252	0.0404	0.0486	0.0016	0.2589	0.1110	0.1264	0.3973	0.4106
Upper Torso	0.1873	0.4737	0.3643	0.3706	0.0001	0.2882	0.0525	0.0601	0.0004	0.3687	0.1807	0.2064	0.4100	0.4284
Lower Torso	0.3648	0.6569	0.5204	0.5261	0.0364	0.2058	0.1256	0.1304	0.0002	0.2137	0.1127	0.1247	0.5471	0.5562
Right Shoulder	0.0008	0.5644	0.2305	0.2668	0.0001	0.2544	0.0689	0.0840	0.0003	0.10740	0.2645	0.3513	0.3575	0.4491
Left Shoulder	0.1680	0.8349	0.4355	0.4666	0.0001	0.2746	0.0653	0.1032	0.0002	0.4657	0.2459	0.2746	0.5044	0.5512
Right Knee	0.3059	0.8789	0.6361	0.6554	0.0557	0.2345	0.1435	0.1486	0.0016	0.4408	0.1913	0.2189	0.6796	0.7068
Left Knee	0.3693	0.7366	0.5597	0.5675	0.2087	0.5211	0.3584	0.3657	0.0001	0.1639	0.0768	0.0894	0.6690	0.6810
Right Ankle	0.0001	0.1639	0.0768	0.0894	0.1292	0.6807	0.4112	0.4343	0.0001	0.0900	0.0283	0.0361	0.4193	0.4449
Left Ankle	0.2071	0.6947	0.4906	0.5016	0.0682	0.2191	0.1330	0.1371	0.1329	0.2071	0.2346	0.2439	0.5598	0.5744
Training Test for 2000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0002	0.4353	0.1787	0.2280	0.2268	0.5457	0.3758	0.3834	0.0019	0.4412	0.2509	0.2694	0.4859	0.5211
Left Palm	0.0001	0.3698	0.1916	0.2167	0.2224	0.5088	0.3353	0.3420	0.0002	0.2595	0.1159	0.1354	0.4032	0.4269
Head	0.0005	0.5309	0.3000	0.3253	0.1518	0.4070	0.2521	0.2596	0.0001	0.1751	0.0551	0.0732	0.3957	0.4226
Upper Torso	0.0002	0.4093	0.1831	0.2205	0.0065	0.3475	0.2300	0.2359	0.0006	0.6280	0.2621	0.3166	0.3939	0.4522
Lower Torso	0.0003	0.3279	0.1812	0.1955	0.1466	0.3421	0.2185	0.2218	0.0003	0.4234	0.1498	0.1879	0.3210	0.3503
Right Shoulder	0.0015	2.6338	0.8822	1.1857	0.0032	0.6290	0.4075	0.4212	0.0012	2.6413	0.8618	1.1712	1.2989	1.7190
Left Shoulder	0.0002	0.4799	0.1630	0.2280	0.0224	0.3284	0.1561	0.1745	0.0007	0.6828	0.3144	0.3411	0.3870	0.4459
Right Knee	0.0004	0.5411	0.2341	0.2721	0.1595	0.3374	0.2190	0.2218	0.0001	0.4506	0.2057	0.2436	0.3809	0.4273
Left Knee	0.0001	0.3307	0.1266	0.1612	0.3838	0.7579	0.5526	0.5600	0.0001	0.2641	0.0861	0.1086	0.5734	0.5928
Right Ankle	0.0001	0.2641	0.0861	0.1086	0.0339	0.6287	0.3047	0.3432	0.0001	0.1351	0.0354	0.0506	0.3186	0.3635
Left Ankle	0.0912	0.6240	0.3557	0.3827	0.0573	0.2639	0.1565	0.1643	0.1409	0.5429	0.3781	0.3918	0.5422	0.5718
Training Test for 3000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0515	0.6427	0.2697	0.3175	0.4703	1.0074	0.7712	0.7843	0.0025	0.4389	0.2624	0.2816	0.8581	0.8918
Left Palm	0.0170	0.3021	0.1112	0.1283	0.3348	0.8011	0.6197	0.6324	0.0623	0.3231	0.2046	0.2167	0.6620	0.6807
Head	0.2290	0.7120	0.4461	0.4724	0.3330	0.7028	0.5570	0.5662	0.2460	0.5291	0.3363	0.3450	0.7889	0.8141
Upper Torso	0.0466	0.5526	0.2991	0.3150	0.0156	0.5784	0.4380	0.4494	0.0128	0.5129	0.2759	0.3008	0.5979	0.6258
Lower Torso	0.2496	0.6172	0.4612	0.4673	0.1730	0.4548	0.3216	0.3288	0.0001	0.1664	0.0350	0.0469	0.5633	0.5733
Right Shoulder	0.0018	1.5806	0.6349	0.6859	0.0003	0.9963	0.6150	0.6936	0.0043	1.5896	0.5176	0.7348	1.0243	1.2213
Left Shoulder	0.0003	0.4796	0.1775	0.2427	0.1434	0.5412	0.3963	0.4149	0.0038	0.3205	0.3205	0.3429	0.5397	0.5904
Right Knee	0.3267	0.9705	0.6378	0.6474	0.0740	0.2828	0.1690	0.1745	0.0004	0.3357	0.1085	0.1432	0.6687	0.6856
Left Knee	0.0001	0.3472	0.1487	0.1744	0.4003	0.8814	0.6918	0.7022	0.0561	0.3591	0.2192	0.2346	0.7408	0.7606
Right Ankle	0.0561	0.3591	0.2192	0.2346	0.3417	0.7271	0.5357	0.5465	0.0417	0.2033	0.1320	0.1380	0.5937	0.6105
Left Ankle	0.0001	0.2673	0.1080	0.1248	0.1728	0.3148	0.2335	0.2362	0.0758	0.6393	0.4517	0.4640	0.5198	0.5354
Training Test for 4000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0010	0.4986	0.2828	0.3073	0.2629	0.6454	0.4464	0.4551	0.0002	0.5030	0.2540	0.2937	0.5863	0.6227
Left Palm	0.0003	0.5658	0.2140	0.2137	0.4038	0.6819	0.5455	0.5494	0.0003	0.2927	0.1136	0.1352	0.5969	0.6048
Head	0.0002	0.5493	0.3645	0.3909	0.2055	0.4214	0.2989	0.3026	0.0023	0.4572	0.1871	0.2220	0.5072	0.5419
Upper Torso	0.0005	1.1522	0.5110	0.6128	0.1976	0.5850	0.3579	0.3691	0.0007	1.1237	0.4745	0.5305	0.7838	0.8906
Lower Torso	0.0019	0.4606	0.2819	0.3071	0.1560	0.2496	0.2440	0.2481	0.0010	0.5580	0.2403	0.2915	0.4436	0.4908
Right Shoulder	0.0006	5.8187	1.7217	2.6620	0.1786	2.2193	0.8004	1.0930	0.0003	5.5726	1.6070	2.5340	2.4874	3.8343
Left Shoulder	0.0001	0.4768	0.3057	0.3286	0.1968	0.4898	0.3460	0.3528	0.0015	0.6745	0.3614	0.4114	0.5863	0.6338
Right Knee	0.0024	0.6906	0.3877	0.4458	0.2336	0.4173	0.3282	0.3310	0.0005	0.5148	0.2212	0.2641	0.5540	0.6149
Left Knee	0.0029	0.4495	0.2284	0.2589	0.0984	0.6692	0.3715	0.4082	0.0031	0.8519	0.3435	0.4144	0.5551	0.6367
Right Ankle	0.0031	0.8519	0.3435	0.4144	0.0004	0.5885	0.3441	0.3943	0.0001	0.1210	0.0492	0.0581	0.4887	0.5750
Left Ankle	0.0056	0.3608	0.2369	0.2467	0.0001	0.1577	0.0623	0.0765	0.0001	0.3914	0.1841	0.2147	0.3064	0.3359
Training Test for 5000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0532	0.5253	0.2693	0.2995	0.1912	0.6608	0.3824	0.3959	0.0007	0.4666	0.2908	0.3141	0.5507	0.5874
Left Palm	0.0003	0.3306	0.1316	0.1622	0.0003	0.4748	0.2175	0.2540	0.1097	0.4872	0.3174	0.3357	0.4067	0.4511
Head	0.0006	0.5617	0.3068	0.3451	0.0037	0.4176	0.2500	0.2688	0.0001	0.1962	0.0866	0.1026	0.4051	0.4493
Upper Torso	0.0640	0.5373	0.3141	0.3520	0.0001	0.2765	0.1252	0.1537	0.0001	0.3749	0.1293	0.1587	0.3620	0.4156
Lower Torso	0.0007	0.6104	0.3843	0.4031	0.0001	0.2654	0.1126	0.1397	0.0006	0.5303	0.3511	0.3630	0.5326	0.5602
Right Shoulder	0.0069	1.6632	0.6372	0.7003	0.0001	0.5389	0.1091	0.1628	0.0001	1.4517	0.5098	0.5770	0.8233	0.9219
Left Shoulder	0.0235	0.4891	0.2197	0.2633	0.0001	0.3267	0.1540	0.1812	0.0001	0.4595	0.2356	0.2715	0.3571	0.4194
Right Knee	0.0036	0.8403	0.4710	0.5172	0.0001	0.2285	0.1105	0.1253	0.0027	0.4644	0.3158	0.3275	0.5777	0.6249
Left Knee	0.0004	0.4307	0.1821	0.2147	0.0002	0.2383	0.0715	0.0917	0.0658	0.6808	0.4157	0.4411	0.4594	0.4991
Right Ankle	0.0658	0.6808	0.4157	0.4411	0.0585	0.8037	0.5383	0.5940	0.0027	0.2006	0.1240	0.1304	0.6913	0.7513
Left Ankle	0.0001	0.1659	0.0795	0.0890	0.0006	0.2574	0.1064	0.1206	0.1384	0.7087	0.4468	0.4754	0.4661	0.4985
Training Test for 2000 Epochs with 500 frames														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0005	0.6301	0.3165	0.3665	0.0827	0.1969	0.1326	0.1358	0.0155	0.2825	0.1784	0.1977	0.3868	0.4380
Left Palm	0.0008	0.7446	0.3310	0.4119	0.1791	0.3081	0.2412	0.2433	0.0785	0.3187	0.1874	0.2004	0.4504	0.5187
Head	0.0003	0.5994	0.3360	0.3821	0.0497	0.1373	0.0855	0.0873	0.0892	0.2604	0.1686	0.1750	0.3855	0.4292
Upper Torso	0.0009	0.5671	0.3374	0.3816	0.0039	0.1041	0.0473	0.0516	0.3032	0.5148	0.3892	0.3946	0.5173	0.5514
Lower Torso	0.0005	0.6655	0.3384	0.3942	0.0664	0.1474	0.0958	0.0972	0.2230	0.4908	0.3287	0.3393	0.4814	0.5291
Right Shoulder	0.0006	0.6393	0.3420	0.3930	0.0223	0.1106	0.0581	0.0608	0.1213	0.3555	0.2383	0.2495	0.4209	0.4695
Left Shoulder	0.0011	0.6299	0.3369	0.3862	0.0254	0.1277	0.0695	0.0725	0.3141	0.5410	0.4099	0.4153	0.5351	0.5717
Right Knee	0.0003	0.6497	0.3588	0.4078	0.1580	0.2771	0.1985	0.1998	0.0290	0.3530	0.2049	0.2288	0.4584	0.5085
Left Knee	0.0005	0.7429	0.3560	0.4303	0.1265	0.2615	0.1757	0.1788	0.4147	0.7033	0.5248	0.5310	0.6580	0.7065
Right Ankle	0.4147	0.7033	0.5248	0.5310	0.0022	0.5565	0.3746	0.4142	0.0079	0.0951	0.0381	0.0438	0.6459	0.6749
Left Ankle	0.0032	0.6513	0.3033	0.3544	0.0198	0.1398	0.0649	0.0737	0.1112	0				

Στο Σχήμα 6.2 απεικονίζεται το διάγραμμα των σφαλμάτων RMSE των αποστάσεων για τα πειραματικά δεδομένα των 1000 καρέ. Το σφάλμα αυτό προέκυψε χρησιμοποιώντας την ευκλείδεια απόσταση και υπολογίζοντας την τετραγωνική ρίζα του αθροίσματος των τριών συνιστώσων σφαλμάτων RMSE. Από αυτό το διάγραμμα προκύπτει ότι το καλύτερο εκτιμώμενο αποτέλεσμα είναι αυτό των 2000 εποχών. Εξαιρώντας το μέλος «Δεξί Ώμος», το οποίο έχει μεγάλο σφάλμα, όλα τα άλλα μέλη του σκελετού έχουν ελάχιστο σφάλμα σε αυτή την εποχή. Γενικά η συνολική εικόνα του μέλους «Δεξί Ώμος», δείχνει ότι υπάρχουν χονδροειδή σφάλματα και σε αυτά οφείλεται η μεγάλη αυτή απόκλιση – ενδεχομένως να μην έχει γίνει καλή διάρθρωση στα τελευταία καρέ ή το ίδιο το σημείο του σκελετού να είναι λάθος.

Στο Σχήμα 6.3 απεικονίζεται το διάγραμμα σύγκρισης μεταξύ των αποτελεσμάτων των δεδομένων 500 και 1000 καρέ για το πείραμα των 2000 εποχών. Στα δεδομένα των 500 καρέ όλες οι τιμές σφάλματος είναι μικρότερες των 70 εκατοστών και το μέλος «Δεξί Ώμος» που εμφανίζει μεγάλη τιμή στο πείραμα των 1000 καρέ, τώρα εμφανίζει τιμή σφάλματος 42 εκατοστών – συμπεραίνετε ότι τα δεδομένα των 1000 καρέ έχουν αρκετό θόρυβο στα τελευταία καρέ.



Σχήμα 6.2. Σφάλματα αποστάσεων για όλες τις εποχές – πειράματα των δεδομένων των 1000 καρέ.



Σχήμα 6.3. Σύγκριση σφαλμάτων αποστάσεων για τα πειράματα των εποχών 2000 μεταξύ των δεδομένων των 500 και 1000 καρέ.

Κεφάλαιο 7

Εναλλακτικές Λύσεις και Προτάσεις

Κατά τη διάρκεια της ανάπτυξης της εφαρμογής οπτικοποίησης, συζητήθηκαν διάφορες εναλλακτικές λύσεις και προτάσεις σχετικά με την αντιμετώπιση των προβλημάτων. Στο κεφάλαιο αυτό θα γίνει μία σύντομη αναφορά στις εναλλακτικές αυτές προτάσεις.

7.1 Αφαίρεση Θορύβου με τη Μέθοδο Ταξινόμησης DBScan

Πριν την ανάπτυξη των μεθόδων αφαίρεσης του συσσωρευμένου και του συγχεόμενου θορύβου, πραγματοποιήθηκαν δοκιμές για την αφαίρεση του θορύβου, με χρήση της μεθόδου ταξινόμησης DBScan [23], [5].

Οι δοκιμές έγιναν χρησιμοποιώντας την έτοιμη εντολή DBSCAN, που περιέχεται στο πακέτρο scikit-learn της python. Στη μέθοδο αυτή δόθηκαν δύο τιμές παραμέτρων. Η πρώτη τιμή αντιστοιχεί στη μέγιστη απόσταση που μπορούν να έχουν δύο σημεία, ώστε να θεωρηθεί πως ανήκουν στην ίδια γειτονιά, ενώ η δεύτερη τιμή αντιστοιχεί στον ελάχιστο αριθμό γειτόνων που μπορεί να έχει ένα σημείο, ώστε να θεωρηθεί κέντρο της γειτονιάς (συμπεριλαμβανομένου και του ίδιου του σημείου).

Η μέθοδος αυτή δεν έδωσε ικανοποιητικά αποτελέσματα, διότι τα δεδομένα περιέχουν αρκετό θόρυβο. Επιπλέον ο προτεινόμενος αλγόριθμος στηριζόταν στην εκτέλεση της μεθόδου DBScan σε κάθε καρέ των δεδομένων και με χρήση ειδικών κανόνων έπρεπε να αξιολογηθούν τα αποτελέσματα και να γίνει επιλογή των n καλύτερων σημείων από κάθε κλάση περιγραφής του αντικείμενου.

Στο Σχήμα 7.2 απεικονίζεται σε σύστημα καρτεσιανών συντεταγμένων το αποτέλεσμα της ταξινόμησης. Με μαύρο χρώμα απεικονίζονται τα σημεία που θεωρεί ο αλγόριθμος λανθασμένα (θόρυβος). Από την απεικόνιση επίσης φαίνεται πως αυτά τα σημεία, αποτελούν σημεία που περιγράφουν τα χέρια και τα γόνατα του χορευτή.

Στο Σχήμα 7.1 απεικονίζεται ο πίνακας ταξινόμησης των σημείων.

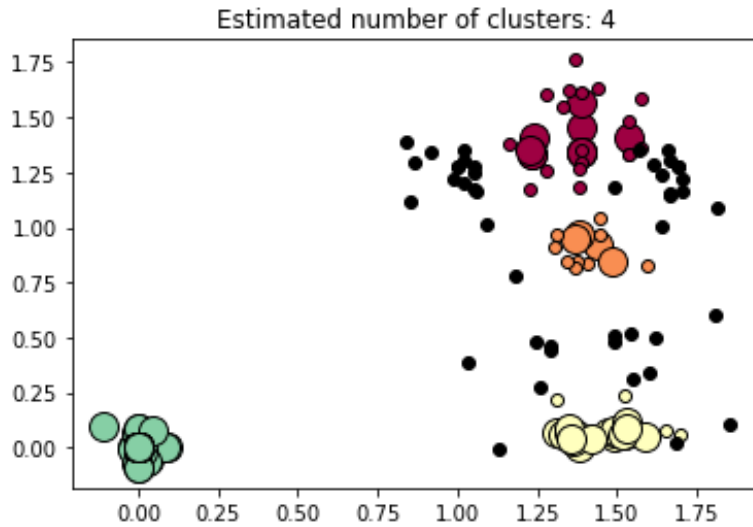
Η μέθοδος αυτή απορρίφθηκε για δύο βασικούς λόγους.

1. Η μέθοδος αυτή έπρεπε να χρησιμοποιηθεί για την αφαίρεση του θορύβου και την ταξινόμηση των κλάσεων, ώστε τελικά να προκύψουν ένα αρχείο με τα σημεία περιγραφής του αντικείμενου σε κάθε καρέ και ένα αρχείο με τα σημεία του σκελετού (κεντροειδή των κλάσεων), που περιγράφουν το σκελετό, κάτι το οποίο είναι πολύ δύσκολο, αν όχι αδύνατο, να γίνει σε αυτά τα δεδομένα.
2. Η μέθοδος αυτή εκτελείται με χρήση γλώσσας προγραμματισμού python, ενώ το γραφικό περιβάλλον έχει σχεδιαστεί σε γλώσσα προγραμματισμού C++. Αυτό πρακτικά σημαίνει ότι η επεξεργασία των δεδομένων θα γινόταν εκτός του βασικού λογισμικού πακέτου της εφαρμογής, κάτι το οποίο καθιστά τις διαδικασίες επεξεργασίας, οπτικοποίησης και αξιολόγησης των αποτελεσμάτων χρονοβόρα.

```
In [8]: print(labels)
```

```
[ 0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1  0 -1 -1 -1 -1  1  1  1  1  1
-1 -1  2  2  2 -1 -1 -1  2  2  2  1  2  0  0  0  0  0  0  0  0 -1 -1 -1
 1  2  2  2  2 -1 -1 -1 -1 -1 -1  1  0 -1 -1 -1 -1  2 -1 -1 -1  2  2  2
 2  1  1  1  1  0  0  0  0 -1 -1  1  2  2  2  2 -1 -1 -1 -1 -1 -1 -1
 0 -1 -1 -1 -1  2 -1  2 -1  2  2  2  2  0  0  0  0  3  3  3  3  3  3  3
 3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
 3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
 3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
 3  3]
```

Σχήμα 7.1. Πίνακας των κλάσεων για το καρέ 0



Σχήμα 7.2. Απεικόνιση των κλάσεων για το καρέ 0

7.2 Διόρθωση του Νέφους Σημείων με Χρήση Γειτόνων

Οι αλγόριθμοι που αναπτύχθηκαν στην εφαρμογή σχετικά με τη διόρθωση του θορύβου, βασίζονται στην τάση του σημείου που λαμβάνεται από τα δύο τελευταία έγγραφα καρέ. Ωστόσο κατά τη διάρκεια της επίλυσης του προβλήματος, σαν εναλλακτική των αλγορίθμων, προτάθηκε η χρήση των εγχύτερων γειτόνων.

Κάθε σημείο απεικονίζει μία συγκεκριμένη μάρκα πάνω στο σώμα του αντικειμένου (χορευτή). Αυτό σημαίνει ότι υπάρχουν μάρκες, οι οποίες σε κάθε καρέ πρέπει να απέχουν σταθερή απόσταση. Αυτό πρακτικά είναι αδύνατο, διότι όπως είναι γνωστό, σε κάθε μέτρηση υπάρχουν σφάλματα [30], [29].

Λαμβάνοντας υπόψιν τα παραπάνω, ο προτεινόμενος αλγόριθμος ακολουθεί τα εξής βήματα:

1. Εισαγωγή τιμής κατωφλίσωσης που ορίζει την ανοχή της μεταβολής, λόγω σφάλματος, της «σταθερής» απόστασης μεταξύ των γειτονικών σημείων και δημιουργία μετρητή με τιμή μηδέν.
2. Για κάθε καρέ και για κάθε σημείο να πραγματοποιηθεί έλεγχος αφαιρώντας την απόσταση ενός σημείου με το γειτονικό του, με στόχο των υπολογισμών των διαφορών των αποστάσεων από καρέ σε καρέ.
3. Εάν η διαφορά των αποστάσεων είναι μικρότερη από τη δοθείσα ανοχή, τότε οι αποστάσεις να προστεθούν μεταξύ τους και να αυξηθεί ο μετρητής κατά ένα. Αυτή η διαδικασία θα πραγματοποιείται μέχρι να βρεθεί η πρώτη απόσταση εκτός ανοχής.
4. Μόλις εντοπιστεί μία απόσταση εκτός των ορίων ανοχής, τότε θα υπολογιστεί ο μέσος όρος των αποστάσεων και η τιμή του μέσου όρου θα θεωρείται ως η σταθερή απόσταση των σημείων.
5. Καταχώρηση της τιμής αυτής σε πίνακα και επανάληψη των παραπάνω βημάτων έως ότου καλυφθούν όλοι οι γείτονες του σημείου.
6. Αφότου δημιουργηθεί ο πίνακας για όλα τα σημεία, τότε για κάθε ένα σημείο του αντικειμένου υπολογίζεται η διαφορά της θέσης του από καρέ σε καρέ.
7. Εάν η μετατόπιση του σημείου υπερβαίνει το μέσο όρο των μετατοπίσεων κι αν η διαφορά των αποστάσεων των γειτονικών σημείων σε εκείνα τα καρέ είναι μεγάλη, τότε να γίνει διόρθωση του σημείου, προσθέτοντας στις συντεταγμένες των γειτονικών του σημείων τις αποστάσεις που έχουν αποθηκευτεί στον πίνακα.
8. Η τελική τιμή της θέσης του σημείου θα υπολογιστεί από τον μέσο όρο των τιμών των θέσεων που θα προκύψουν.

Για την πλήρη κατανόηση του παραπάνω αλγορίθμου, επισημαίνεται ότι με την έννοια απόσταση εννοείται η διαφορά μεταξύ των συντεταγμένων του ίδιου σημείου, ενώ με τον όρο θέση εννοείται η τριάδα συντεταγμένων. Επιπλέον ο αλγόριθμος θα επαναληφθεί όσες φορές χρειαστεί, ώστε όλα τα σημεία στο τέλος να βρίσκονται στην καλύτερη θέση, η οποία ορίζεται από τους εγχύτερους γείτονες.

Ο αλγόριθμος στηρίζεται στη λογική των κοκάλων. Δύο μάρκες οι οποίες έχουν τοποθετηθεί η μία στον ώμο και η άλλη στον αγκώνα ενός χορευτή, θα πρέπει να έχουν ίδια απόσταση σε όλα τα καρέ ή τουλάχιστον στο μεγαλύτερο μέρος των καρέ. Αυτό σημαίνει ότι τα σημεία αυτά είναι γείτονες και η σταθερή τους απόσταση μπορεί να ονομαστεί ορθώς «κόκαλο».

Ο αλγόριθμος μειωνεκτεί σε δύο πράγματα. Το πρώτο είναι η προγραμματιστική του πολυπλοκότητα και το δεύτερο είναι πως ο χρόνος της διαδικασίας εξαρτάται από το πλήθος των σημείων, αλλά και από το μέγεθος των προβληματικών σημείων (αυτό σημαίνει ότι ο χρήστης θα περιμένει μερικά λεπτά μέχρι να δει το αποτέλεσμα της εντολής).

7.3 Ταξινόμηση με Χρήση της Μεθόδου Bounding Volume

Η μέθοδος Bounding Volume είναι μία μέθοδος, η οποία ορίζει έναν κύβο μέσα στο οποίο ότι σημείο υπάρχει, ανήκει στην ίδια κλάση. Ο κύβος ορίζεται από δύο σημεία με συντεταγμένες $(X_{min}, Y_{min}, Z_{min})$ και $(X_{max}, Y_{max}, Z_{max})$ και οποιοδήποτε σημείο βρίσκεται μέσα στα όρια που ορίζουν οι συντεταγμένες αυτές ανήκει σε αυτή την κλάση.

Ο αλγόριθμος αυτός πλεονεκτεί στο ότι ο χρήστης έχει τον πλήρη έλεγχο για την ταξινόμηση των σημείων σε κάθε κλάση, αφού οι τιμές των σημείων που ορίζουν τα όρια του ορθογωνίου ή κύβου εισάγονται από το χρήστη. Για αυτό το λόγο και η μέθοδος αυτή μπορεί να λειτουργήσει και σε νέφη σημείων με θόρυβο.

Αν και η μέθοδος αυτή πλεονεκτεί στο σύνολό της, απαιτείται η δημιουργία του κατάλληλου γραφικού περιβάλλοντος και αυτό αυξάνει την προγραμματιστική πολυπλοκότητα του αλγορίθμου. Για αυτό το λόγο και επιλέχθηκε ως εναλλακτική η ταξινόμηση με τη μέθοδο K-Means αντί της μεθόδου Bounding Volume [28].

Κεφάλαιο 8

Επίλογος - Συμπεράσματα - Μελλοντικές Επεκτάσεις

Κατά τη διάρκεια εκπόνησης αυτής της διπλωματικής αντιμετωπίστηκαν αρκετές δυσκολίες, οι οποίες οδήγησαν στον επανασχεδιασμό αντιμετώπισης του προβλήματος, καθώς επίσης τα αποτελέσματα από τα πειράματα του νευρωνικού δικτύου, οδήγησαν στη συγγραφή μίας δημοσίευσης (paper).

Τα συμπεράσματα που προέκυψαν από την εκπόνηση και υλοποίηση αυτής της διπλωματικής, παρατίθενται στις επόμενες ενότητες.

8.1 Γνώση που Έλαβα

Κατά τη διάρκεια εκπόνησης της διπλωματικής αντιμετωπίστηκαν αρκετές δυσκολίες. Το πρώτο και μεγαλύτερο εμπόδιο, ήταν η ανάγνωση των δεδομένων με τρόπο τέτοιο ώστε να μπορεί να γίνει όσο το δυνατόν πιο εύκολη η επεξεργασία τους.

Το αρχείο C3D είναι ένα αρχείο binary, το οποίο σημαίνει ότι ακολουθεί μία συγκεκριμένη δομή ανάγνωσης. Συνεπώς κατά τη διάρκεια ανάγνωσης των δεδομένων αναγκάστηκα να μάθω τη δομή οργάνωσης ενός δυαδικού αρχείου, χρησιμοποιώντας ως πειραματικά δεδομένα τα αρχεία C3D. Επιπλέον κατά τη δημιουργία του κώδικα χρειάστηκε να αναζητήσω πληροφορίες σχετικά με τον τρόπο που καταχωρούν και διαβάζουν οι επεξεργαστές ένα δυαδικό αριθμό (από δεξιά προς τα αριστερά – μικροί ενδιάνοι – ή από αριστερά προς τα δεξιά – μεγάλοι ενδιάνοι).

Η επόμενο φάση της διπλωματικής απαιτούσε τη δημιουργία του γραφικού περιβάλλοντος. Στο στάδιο αυτό χρησιμοποιήθηκαν δύο διεπαφές για την υλοποίηση του, η διεπαφή QtCreator για τη δημιουργία και διαχείριση των παραθύρων και η διεπαφή της OpenGL για την οπτικοποίηση των αντικειμένων στην οθόνη. Το στάδιο αυτό αποτελεί την εισαγωγή μου στον κόσμο των τριδιάστατων γραφικών υπολογιστών, τα οποία χρησιμοποιούν το μοντέλο της κεντρικής προβολής. Η διαφορά που έχει το μοντέλο αυτό με την κλασική φωτογραμμετρία, είναι ότι στη φωτογραμμετρία ο χώρος υπάρχει και ο στόχος εκεί είναι ο υπολογισμός του μοντέλου, ενώ στα γραφικά υπολογιστών πρέπει πρώτα να δημιουργηθεί το μοντέλο της κεντρικής προβολής και στη συνέχεια να δημιουργηθεί και να προβληθεί ο χώρος (το πρόβλημα αντιστρέφεται). Επιπλέον κατά τη διάρκεια εκπόνησης αυτής της φάσης κατανόησα καλύτερα τη χρήση των δεικτών στη γλώσσα προγραμματισμού C/C++.

Τέλος έμαθα να χρησιμοποιώ απλές μεθόδους επεξεργασίας νέφους σημείων, για την αφαίρεση του θορύβου και την ταξινόμηση των σημείων σε ομάδες (εύρεση των μελών που αποτελούν το σκελετό), ενώ σε δεύτερη φάση ήρθα σε επαφή με μεθόδους νευρωνικών δικτύων και διαδικασίες εκμάθησης του υπολογιστή, που αποτελεί τη σύγχρονη εξέλιξη της επιστήμης της φωτογραμμετρίας (όραση υπολογιστών).

8.2 Προσφορά της Διπλωματικής

Η εκπόνηση της διπλωματικής έγινε στα πλαίσια ενός ερευνητικού προγράμματος για την διαφύλαξη της άυλης πολιτιστικής κληρονομιάς. Στην εισαγωγή έγινε αναφορά πως τμήμα της άυλης πολιτιστικής κληρονομιάς είναι ο χορός. Η παρούσα διπλωματική εστιάζει στην οπτικοποίηση και επεξεργασία δεδομένων που σχετίζονται με την απεικόνιση των βασικών κινήσεων ενός παραδοσιακού χορού (στα πλαίσια της διπλωματικής έγινε επεξεργασία σε δεδομένα που περιγράφουν το Συρτό).

Σε πρώτη φάση η διπλωματική απομονώνει το αντικείμενο από το θόρυβο και προσδιορίζει τα σημεία περιγραφής του σκελετού του αντικειμένου (χορευτής). Στην πραγματικότητα δεν υπάρχει ανάγκη απεικόνισης του χορευτή, χρησιμοποιώντας ένα νέφος σημείων. Ο ιδανικός αριθμός για την απεικόνιση των χορευτικών κινήσεων είναι τα 11 σημεία του σκελετού.

- | | |
|--------------------------|----------------------------|
| 1. Κεφάλι | 7. Άκρο αριστερού χεριού |
| 2. Στήθος | 8. Γόνατο δεξιού ποδιού |
| 3. Λεκάνη | 9. Γόνατο αριστερού ποδιού |
| 4. Ώμος δεξιού χεριού | 10. Άκρο δεξιού ποδιού |
| 5. Ώμος αριστερού χεριού | 11. Άκρο αριστερού ποδιού |
| 6. Άκρο δεξιού χεριού | |

Με τις μεθόδους που αναπτύχθηκαν στην παρούσα διπλωματική μπορούν να αναγνωρίζονται με αυτοματοποιημένες διαδικασίες (αφαίρεση θορύβου και ταξινόμηση) μέχρι 7 σημεία σκελετού (για το παράδειγμα του Κεφαλαίου 6 αυτά είναι: **Κεφάλι** → **Στήθος** → **Άκρο δεξιού χεριού** → **Άκρο αριστερού χεριού** → **Λεκάνη** → **Άκρο δεξιού ποδιού** → **Άκρο αριστερού ποδιού**). Τα υπόλοιπα τέσσερα σημεία προσδιορίζονται χειροκίνητα λαμβάνοντας συντεταγμένες από σημεία του νέφους.

Τέλος η σημαντικότερη προσφορά της διπλωματικής – η οποία δημοσιεύτηκε – είναι τα πειραματικά αποτελέσματα που πραγματοποιήθηκαν για την εύρεση των 11 σημείων του σκελετού με τη χρήση νευρωνικών δικτύων. Τα αποτελέσματα, αν και απέχουν πολύ από τα ιδανικά, αποτελούν ένα σημαντικό πρώτο βήμα, για την ανάπτυξη μεθόδων όρασης υπολογιστών με δεδομένα εισαγωγής νέφη σημείων, αντί εικόνων ή βίντεο.

8.3 Μελλοντικά Πειράματα

Μερικά πειράματα που μπορούν να πραγματοποιηθούν στο μέλλον, είναι τα εξής:

1. Εύρεση των σημείων του σκελετού με τη χρήση διαφορετικής μεθόδου και όχι τη μέθοδο CNN.
2. Εισαγωγή καλύτερων και πληρέστερων δεδομένων, προκειμένου να γίνει καλύτερη εκπαίδευση του δικτύου. Τα δεδομένα που χρησιμοποιήθηκαν σε αυτή τη διπλωματική ήταν γεμάτα θόρυβο, ενώ όπως αποδεικνύεται από το Σχήμα 6.3 συνεχίζει να υπάρχει θόρυβος και στα δεδομένα εκπαίδευσης και ελέγχου.
3. Τα πειράματα μπορούν να επεκταθούν και σε άλλους τομείς εκτός του χορού. Παραδείγματος χάρη, μπορεί ληφθούν δείγματα και να γίνει μελέτη σε διάφορες ασκήσεις της ρυθμικής γυμναστικής.

8.4 Επίλογος

Εν κατακλείδι η διπλωματική αυτή αποτελεί τη βάση για τη δημιουργία ενός ευρύτερου φάσματος εφαρμογών, εκμάθησης και αξιολόγησης μοντέλων. Επιπλέον η εφαρμογή που προγραμματίστηκε κατά την εκπόνηση της διπλωματικής οπτικοποιεί ένα μεγάλο μέρος των σύγχρονων αρχείων C3D και αυτό δίνει τη δυνατότητα στους χρήστες να επεξεργαστούν τα δεδομένα τους σε ένα φιλικό κι εύχρηστο περιβάλλον. Τέλος στο περιβάλλον αυτό μπορούν να προστεθούν διάφοροι αλγόριθμοι, οι οποίοι θα καλύπτουν τις ανάγκες του χρήστη για την επίλυση του υφιστάμενου κάθε φορά προβλήματος. Με αυτό τον τρόπο μπορούν να λυθούν περισσότερα προβλήματα από όσα έχουν ήδη προταθεί.

Κεφάλαιο 9

Βιβλιογραφία

- [1] Addison-Wesley. *OpenGL Programming Guide, The Official Guide to Learning OpenGL, Version 1.1*. United States of America, second edition, January 1997.
- [2] J. Blake. unesco's 2003 convention on intangible cultural heritage. *The implications of community involvement*, pages 45–50, 2008.
- [3] J. K. Dave Shreiner, Graham Sellers and B. Licea-Kane. *OpenGL Programming Guide, The Official Guide to Learning OpenGL, Version 4.3*. United States of America, eighth edition, March 2013.
- [4] K. Dimitropoulos, S. Manitsaris, F. Tsalakanidou, S. Nikolopoulos, B. Denby, S. Kork, L. Crevier-Buchman, C. Pillot-Loiseau, M. Adda-Decker, S. Dupont, J. Tilmanne, M. Ott, M. Alivizatos, E. Yilmaz, L. Hadjileontiadis, V. Charisis, O. Deroo, A. Manitsaris, I. Kompatsiaris, and N. Grammalidis. capturing the intangible: An introduction to the i-treasures project. volume 2, pages 773–781, 2014.
- [5] A. Doulamis, N. Doulamis, E. Protopapadakis, A. Voulodimos, and M. Ioannides. 4d modelling in cultural heritage. In *Advances in Digital Cultural Heritage*, pages 174–196. Springer, 2018.
- [6] A. Doulamis, M. Ioannides, N. Doulamis, A. Hadjiprocopis, D. Fritsch, O. Balet, M. Julien, E. Protopapadakis, K. Makantasis, G. Weinlinger, et al. 4d reconstruction of the past. In *First International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2013)*, volume 8795, page 87950J. International Society for Optics and Photonics, 2013.
- [7] A. Doulamis, S. Soile, N. Doulamis, C. Chrisouli, N. Grammalidis, K. Dimitropoulos, C. Manesis, C. Potsiou, and C. Ioannidis. selective 4d modelling framework for spatial-temporal land information management system. In *Third International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2015)*, volume 9535, page 953506. International Society for Optics and Photonics, 2015.
- [8] A. Doulamis, A. Voulodimos, N. Doulamis, S. Soile, and A. Lampropoulos. transforming intangible folkloric performing arts into tangible choreographic digital objects: The terpsichore approach. volume 5, pages 451–460, 2017.
- [9] N. Doulamis. adaptable deep learning structures for object labeling/tracking under dynamic visual environments. *Multimedia Tools and Applications*, 77(8):9651–9689, 2018.
- [10] N. Doulamis and A. Doulamis. semi-supervised deep learning for object tracking and classification. pages 848–852, 2014.
- [11] N. Doulamis and A. Voulodimos. fast-mdl: Fast adaptive supervised training of multi-layered deep learning models for consistent object tracking and classification. pages 318–323, 2016.
- [12] N. D. Doulamis, P. Kokkinos, and E. Varvarigos. resource selection for tasks with time requirements using spectral clustering. *IEEE Transactions on Computers*, 63(2):461–474, 2014.
- [13] M. Ioannides, A. Hadjiprocopi, N. Doulamis, A. Doulamis, E. Protopapadakis, K. Makantasis, P. Santos, D. Fellner, A. Stork, O. Balet, et al. online 4d reconstruction using multi-images available under open access. *ISPRS Annals of the Photogrammetry, Remote Sensing and Saptial Information Sciences, II-5 W*, 1:169–174, 2013.
- [14] I. Kavouras, E. Protopapadakis, A. Doulamis, and N. Doulamis. skeleton extraction of dance sequences from 3d points using convolutional neural networks based on a new developed 3cd visualization interface. 2018.
- [15] Khronos. opengl 2.1 reference pages. "<https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/>".
- [16] Khronos. opengl 4 reference pages. "<https://www.khronos.org/registry/OpenGL-Refpages/gl4/>".
- [17] Khronos. opengl es 1.1 reference pages. "<https://www.khronos.org/registry/OpenGL-Refpages/es1.1/xhtml>".
- [18] R. Kurin et al. safeguarding intangible cultural heritage: Key factors in implementing the 2003 convention. *International Journal of Intangible Heritage*, 2(8):9–20, 2007.

- [19] G. Kyriakaki, A. Doulamis, N. Doulamis, M. Ioannides, K. Makantasis, E. Protopapadakis, A. Hadjiprocopis, K. Wenzel, D. Fritsch, M. Klein, et al. 4d reconstruction of tangible cultural heritage objects from web-retrieved images. *International Journal of Heritage in the Digital Era*, 3(2):431–451, 2014.
- [20] S. Labadi. *UNESCO, cultural heritage, and outstanding universal value: Value-based analyses of the World Heritage and Intangible Cultural Heritage Conventions*. Rowman & Littlefield, 2013.
- [21] A. Laggis, N. Doulamis, E. Protopapadakis, and A. Georgopoulos. a low-cost markerless tracking system for trajectory interpretation. volume 42, pages 413–418, 2017.
- [22] Y. LeCun, Y. Bengio, and G. Hinton. deep learning. *nature*, 521(7553):436, 2015.
- [23] K. Makantasis, A. Doulamis, N. Doulamis, and M. Ioannides. in the wild image retrieval and clustering for 3d cultural heritage landmarks reconstruction. *Tools and Applications*, 75(7):3593–3629, 2016.
- [24] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis. deep supervised learning for hyperspectral data classification through convolutional neural networks. volume 2015-November, pages 4959–4962, 2015.
- [25] K. Makantasis, K. Karantzalos, A. Doulamis, and K. Loupos. deep learning-based man-made object detection from hyperspectral data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9474:717–727, 2015.
- [26] E. Maltezos, N. Doulamis, A. Doulamis, and C. Ioannidis. deep convolutional neural networks for building extraction from orthoimages and dense image matching point clouds. *Journal of Applied Remote Sensing*, 11(4), 2017.
- [27] E. Protopapadakis, A. Grammatikopoulou, A. Doulamis, and N. Grammalidis. folk dance pattern recognition over depth images acquired via kinect sensor. volume 42, pages 587–593, 2017.
- [28] E. Protopapadakis, A. Voulodimos, A. Doulamis, S. Camarinopoulos, N. Doulamis, and G. Miaoulis. dance pose identification from motion capture data: A comparison of classifiers. *Technologies*, 6(1):31, 2018.
- [29] I. Rallis, N. Doulamis, A. Doulamis, A. Voulodimos, and V. Vescoukis. spatio-temporal summarization of dance choreographies. *Computers and Graphics (Pergamon)*, 73:88–101, 2018.
- [30] I. Rallis, I. Georgoulas, N. Doulamis, A. Voulodimos, and P. Terzopoulos. extraction of key postures from 3d human motion data for choreography summarization. pages 94–101, 2017.
- [31] A. Vedaldi and K. Lenc. matconvnet: Convolutional neural networks for matlab. pages 689–692, 2015.
- [32] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018, 2018.
- [33] Wikipedia. intangible cultural heritage. "https://en.wikipedia.org/wiki/Intangible_cultural_heritage".
- [34] Γιάννης Δελόπουλος. Πάρε Δώσε, Παραδοσιακοί χοροί της Ελλάδας. "<https://www.pare-dose.net/85>", Μάρτιος 2008.
- [35] Ξ. Παπακώστας. Ι. Πραντσιδής. ανδ Ε. Παλλάτου. ο παραδοσιακός χορός στα χωριά του νομού δράμας: Εθνογραφικά στοιχεία και ρυθμοκινητική ανάλυση. *Inquiries in Sport & Physical Education*, 4(3):430–441, 2006.