



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ**  
**ΤΟΜΕΑΣ ΜΕΛΕΤΗΣ ΠΛΟΙΟΥ ΚΑΙ ΘΑΛΑΣΣΙΩΝ**  
**ΜΕΤΑΦΟΡΩΝ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη εφαρμογής Gaussian splatting σε περιβάλλον**  
**Unity**

Ρηγόπουλος Ρήγας Ρηγίνος

**Υπεύθυνος καθηγητής:** Αλέξανδρος Γκίνης, Αν. Καθηγητής Ε.Μ.Π.

**Εξεταστική Επιτροπή:** Αλέξανδρος Γκίνης, Αν. Καθηγητής Ε.Μ.Π.

Νικόλαος Βεντικός, Καθηγητής Ε.Μ.Π.

Νικόλαος Θέμελης, Επ. Καθηγητής Ε.Μ.Π.

Ρηγόπουλος Ρήγας Ρηγίνος

Ναυπηγός Μηχανολόγος Μηχανικός Ε.Μ.Π.

Copyright © Ρηγόπουλος Ρήγας Ρηγίνος, 2024

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

## Ευχαριστίες

Προτού ξεκινήσουμε να ασχοληθούμε με το θέμα της διπλωματικής εργασίας που ανέλαβα είναι απαραίτητο να ευχαριστήσω κάποιους ανθρώπους που συνέβαλαν καθοριστικά και καταλυτικά στην ολοκλήρωση αυτής της διπλωματικής εργασίας.

Πρώτο από όλους οφείλω να ευχαριστήσω τον κύριο Αλέξανδρο Γκίνη με τον οποίο αφού συζητήσαμε ενδελεχώς όλα τα υποψήφια θέματα μου έδωσε τη δυνατότητα να ασχοληθώ με αυτό το θέμα που αφορά τη καινοτόμο μέθοδο τρισδιάστατης αναπαράστασης με Gaussian splatting. Ακόμα, τον ευχαριστώ για το γεγονός ότι ασχολήθηκε και ο ίδιος πολύ με το να με καθοδηγεί και να μου λύνει ό,τι απορία είχα. Συνεχίζοντας, ο κύριος Λιάσκος Ορεστης, ο οποίος βοήθησε σημαντικά στην διπλωματική, αφού ενδιαφερόταν για κάθε πρόβλημα το οποίο συναντούσα και με βοηθούσε να το ξεπεράσω, αξίζει ένα μεγάλο ευχαριστώ για τη διαρκή βοήθεια του. Επιπλέον θα ήθελα να ευχαριστήσω τους κυρίους Bernhard Kerbl, Georgios Koranas, Thomas Leimkühler και George Drettakis, οι οποίοι ουσιαστικά φέρανε στην επιφάνεια το Gaussian splatting και ανέπτυξαν τον κώδικα στον οποίο θα βασιστούμε στη διπλωματική αυτή για τη τρισδιάστατη αναπαράσταση. Επίσης, ένα ευχαριστώ χρωστάω και στους ανθρώπους που ασχολήθηκαν με τη δημιουργία του COLMAP, του Unity καθώς και όλους όσους με βοήθησαν με αυτά που είχαν ανεβάσει στο GitHub.

Τέλος, θα ήθελα να ευχαριστήσω μέσα από τη καρδιά μου τους γονείς μου για τη διαρκή και ανιδιοτελή υποστήριξη που μου προσφέραν όλα αυτά τα χρόνια και συνεχίζουν να παρέχουν διαρκώς.

## Περίληψη

Η παρούσα διπλωματική εργασία με τίτλο «**Ανάπτυξη εφαρμογής Gaussian splatting σε περιβάλλον Unity**» έχει ως στόχο να εξετάσει τις βασικές αρχές , τις θεωρητικές βάσεις , τις πρακτικές εφαρμογές καθώς και τις προοπτικές αξιοποίησης της μεθόδου Gaussian splatting στον χώρο των τρισδιάστατων γραφικών και κατ' επέκταση στην χρήση του για αναπαράσταση χώρων του πλοίου. Η τεχνική Gaussian Splatting είναι μια μέθοδος απεικόνισης που στοχεύει στην αντιμετώπιση των προκλήσεων που σχετίζονται με την αναπαράσταση χώρων βασισμένη σε σημεία και είναι εξαιρετικά κατάλληλη για χώρους και σκηνές μεγάλης κλίμακας και πολυπλοκότητας. Η διπλωματική αυτή ερευνά τις θεωρητικές βάσεις του Gaussian Splatting, τα πλεονεκτήματά του έναντι των παραδοσιακών μεθόδων απεικόνισης και τις πρακτικές σκέψεις για την υλοποίησή του.

Πιο συγκεκριμένα, θα ασχοληθούμε και θα συζητήσουμε για τον τρόπο με τον οποίο προετοιμάζονται τα δεδομένα εισόδου για να γίνει η τρισδιάστατη αναπαράσταση με Gaussian Splatting. Αφού κατανοήσουμε τον τρόπο που λειτουργεί το COLMAP , το οποίο είναι το πρόγραμμα που προετοιμάζει τα δεδομένα , θα προσπαθήσουμε να βρούμε έναν διαφορετικό τρόπο ώστε μόνοι μας να δημιουργούμε τα δεδομένα που απαιτούνται για να γίνει η τρισδιάστατη αναπαράσταση, αξιοποιώντας μόνο τις φωτογραφίες και το point cloud του χώρου. Έπειτα ,θα επανέλθουμε στη διερεύνηση των δυνατοτήτων του COLMAP και θα χρησιμοποιήσουμε ένα προσθετό που έχει αναπτυχθεί για αυτό και ονομάζεται COLMAP-PCD. Με αυτό θα επιχειρήσουμε να κάνουμε image to point cloud registration στο χώρο του εργαστήριου της ναυτικής μηχανολογίας ούτως ώστε να έχουμε τα ακριβή δεδομένα εισόδου για να γίνει η τρισδιάστατη αναπαράσταση με Gaussian Splatting και να προκύψει τελικά ένα ρεαλιστικό και εντυπωσιακό αποτέλεσμα.

Κλείνοντας, θα προσπαθήσουμε να εισάγουμε με το βέλτιστο δυνατό τρόπο τη τρισδιάστατη αναπαράσταση στο Unity οπου εκεί θα αντιληφθούμε καλύτερα το αποτέλεσμα και θα μπορέσουμε να το επεξεργαστούμε. Ακόμα στο Unity θα προετοιμάσουμε τη τρισδιάστατη αναπαράσταση χρησιμοποιώντας το OpenXR ούτως ώστε να μπορούμε να περιηγηθούμε στο χώρο με ένα VR Headset και να ανακαλύψουμε τις δυνατότητες της εικονικής πραγματικότητας σε συνδυασμό με τη τρισδιάστατη αναπαράσταση με Gaussian Splatting.

# Abstract

This diploma thesis, titled "Development of a Gaussian Splatting Application in the Unity Environment" aims to examine the fundamental principles, theoretical foundations, practical applications, and potential utilization of the Gaussian Splatting method in the field of 3D graphics, specifically for representing ship spaces. Gaussian Splatting is a visualization technique designed to address the challenges associated with point-based space representation, making it particularly suitable for large-scale and complex spaces and scenes. This thesis explores the theoretical grounds of Gaussian Splatting, its advantages over traditional rendering methods, and the practical considerations for its implementation.

More specifically, the focus will be on how input data is prepared to enable 3D representation using Gaussian Splatting. After understanding how COLMAP works, the program that prepares the input data, we will attempt to find an alternative method to create the necessary data for 3D representation using only photographs and the point cloud of the space. Then, we will revisit the capabilities of COLMAP and utilize an add-on developed for it called COLMAP-PCD. With this, we will attempt to perform image-to-point cloud registration in the Maritime Engineering Laboratory so that we have the precise input data needed for 3D representation with Gaussian Splatting, ultimately leading to a realistic and impressive result.

Finally, we will try to optimally integrate the 3D representation into Unity, where we can better perceive and refine the result. Additionally, within Unity, we will prepare the 3D representation using OpenXR to enable navigation through the space with a VR headset, exploring the possibilities of virtual reality combined with 3D representation using Gaussian Splatting.

## Περιεχόμενα

1. Εισαγωγή.....	8
1.1 Υπόβαθρο.....	8
1.2 Στόχοι.....	8
2. Παραδοσιακές Μέθοδοι Απεικόνισης.....	9
2.1 Scanline Rendering.....	9
2.2 Ray Casting - Ray Tracing.....	10
2.3 Z-Buffer Rendering.....	12
2.4 Painter's Algorithm.....	13
3. Σημειοκεντρική Απεικόνιση (Point-Based Rendering).....	14
3.1 Βασικές Αρχές και Πλεονεκτήματα της Σημειοκεντρικής Απεικόνισης.....	15
3.2 Εφαρμογές της Σημειοκεντρικής Απεικόνισης.....	16
3.3 Point Cloud – «Σύννεφο σημείων».....	16
3.3.1 Θεμελιώδεις αρχές του Point Cloud – «Σύννεφου σημείων».....	17
3.3.2 Πλεονεκτήματα και Προκλήσεις της Αναπαράστασης με Point Clouds.....	17
3.3.3 Εφαρμογές των Point Clouds στα 3D-γραφικά.....	17
4. Gaussian Splatting.....	19
4.1 NeRF (Neural Radiance Field).....	20
4.2 Από το NeRF (Neural Radiance Field) στο 3D Gaussian Splatting.....	22
4.3 Τα βασικά για το 3D Gaussian Splatting.....	23
4.3.1 CUDA kernels.....	24
4.4 Gaussian Splatting για την αναπαράσταση τρισδιάστατων σκηνών.....	25
4.5 Μοντέλο σχηματισμού και αναπαράστασης εικόνας στο Gaussian Splatting.....	28
4.6 Βελτιστοποίηση στο Gaussian Splatting.....	32
4.7 Spherical Harmonics (SH).....	36
4.8 Περιορισμοί του Gaussian Splatting.....	38
5. Gaussian splatting στην πράξη.....	39
5.1 Τρισδιάστατη αναπαράσταση με τη μέθοδο Gaussian Splatting.....	40
5.1.1 Έτοιμα μοντέλα για τρισδιάστατη αναπαράσταση.....	40
5.1.2 Δημιουργία μοντέλων για τρισδιάστατη αναπαράσταση από δικά μας δεδομένα.....	45
5.2 Το COLMAP στη τρισδιάστατη αναπαράσταση με τη μέθοδο Gaussian Splatting.....	47
5.2.1 SfM (Structure-from-Motion) και MVS (Multi-View Stereo) στο COLMAP.....	48

5.2.2 Βάση Δεδομένων στο COLMAP.....	50
5.2.3 Αναγνώριση και Ταύτιση Χαρακτηριστικών στο COLMAP.....	52
5.2.4 Αποδεκτά Μοντέλα Κάμερας στο COLMAP.....	56
5.2.5 Δεδομένα εξόδου στο COLMAP.....	58
5.3 Τα Quaternions στην Τρισδιάστατη Αναπαράσταση.....	60
5.3 Χρήσιμες πληροφορίες για τον κώδικα εκπαίδευσης του μοντέλου με τη μέθοδο Gaussian Splatting.....	62
5.4 Gaussian Splatting χωρίς τη χρήση του COLMAP.....	64
5.4.1 CloudCompare.....	64
5.4.2 Δημιουργία Δεδομένων για 3D αναπαράσταση με Gaussian Splatting.....	66
5.5 Gaussian Splatting με τη χρήση του COLMAP-PCD.....	75
5.5.1 Τρόπος χρήση του COLMAP-PCD.....	75
5.5.2 Αποτελέσματα τρισδιάστατης αναπαράστασης μέσω Gaussian Splatting με δεδομένα εισόδου από το COLMAP-PCD.....	80
6. Gaussian splatting στο περιβάλλον του Unity.....	87
6.1 Λίγα λόγια για το περιβάλλον του Unity.....	87
6.2 Gaussian splatting στο περιβάλλον Unity.....	88
6.3 Gaussian splatting και Virtual Reality στο περιβάλλον Unity.....	90
7. Επίλογος.....	92
Βιβλιογραφία.....	93

# 1.Εισαγωγή

## 1.1 Υπόβαθρο

Ο διαρκώς εξελισσόμενος και αναπτυσσόμενος τομέας των τρισδιάστατων γραφικών χαρακτηρίζεται από μια συνεχή αναζήτηση και διερεύνηση νέων τεχνικών απεικόνισης που μπορούν να αντιμετωπίσουν και να αναπαραστήσουν επιτυχώς τις πολυπλοκότητες των σύγχρονων εικονικών περιβαλλόντων. Από τις αρχές της παραδοσιακής απεικόνισης έως την εμφάνιση της απεικόνισης βασισμένης σε σημεία, η επιδίωξη ρεαλισμού και αποτελεσματικότητας έχει οδηγήσει σε έξαρση της καινοτομίας και της έρευνας. Ωστόσο, καθώς οι σκηνές και οι χώροι προς αναπαράσταση εξελίσσονται ραγδαία σε πολυπλοκότητα και κλίμακα, οι περιορισμοί που εντοπίζονται τόσο στις παραδοσιακές όσο και στις μεθόδους απεικόνισης βασισμένες σε σημεία έχουν γίνει όλο και περισσότερο εμφανείς.

Αντιμετωπίζοντας αυτές τις προκλήσεις, το Gaussian Splatting εμφανίζεται ως μια ελπιδοφόρα, αισιόδοξη και δημιουργική λύση - ένα φωτεινό σημάδι καινοτομίας που στοχεύει να αναδιαμορφώσει την προσέγγιση στην απεικόνιση στα τρισδιάστατα γραφικά. Αυτό το κεφάλαιο αποτελεί μια αρχή για την εξερεύνηση του Gaussian Splatting, θέτοντας τις βάσεις με την ανάλυση του ιστορικού πλαισίου των τεχνικών απεικόνισης και των κινήτρων που μας καθοδηγούν ενεργά προς μια βαθύτερη κατανόηση αυτής της νέας προσέγγισης.

## 1.2 Στόχοι

Οι στόχοι αυτής της διπλωματικής εργασίας είναι πολλαπλοί. Καταρχάς, θα προσπαθήσουμε να ξετυλίξουμε τα θεωρητικά θεμέλια πάνω στα οποία στηρίζεται το Gaussian Splatting. Αυτό περιλαμβάνει μια λεπτομερή εξέταση των αρχών, των μαθηματικών τύπων και των εσωτερικών λειτουργιών που διακρίνουν αυτήν την τεχνική απεικόνισης. Στη συνέχεια, θα αναζητήσουμε και θα εξετάσουμε ποικίλες εφαρμογές του Gaussian Splatting, ξεπερνώντας τα παραδοσιακά όρια και βρίσκοντας χρησιμότητα στην επιστημονική οπτικοποίηση, τα γραφικά υπολογιστών και την εικονική πραγματικότητα. Τέλος, η εργασία αυτή μας οδηγεί σε ένα πρακτικό επίπεδο καθώς προχωράμε προς την υλοποίηση ενός λειτουργικού αλγορίθμου Gaussian Splatting, γεφυρώνοντας το χάσμα μεταξύ θεωρίας και πρακτικής εφαρμογής.

Καθώς αναλύουμε τα θεωρητικά πλαίσια, τους τομείς εφαρμογής και τις πρακτικές υλοποιήσεις, θα προσπαθήσουμε να συμβάλουμε όχι μόνο στον ακαδημαϊκό διάλογο περί των τρισδιάστατων γραφικών, αλλά και να παρέχουμε χειροπιαστές εισηγήσεις και



προτάσεις που μπορούν να αξιοποιηθούν τόσο από επαγγελματίες όσο και από ερευνητές. Αρχικά θα εξερευνήσουμε την ιστορική εξέλιξη των τεχνικών απεικόνισης, πριν την εμφάνιση του Gaussian Splatting ως μια λαμπρή καινοτομία στον χώρο αυτό και ένα δυναμικό μετασχηματισμό στον κόσμο της απεικόνισης τρισδιάστατων γραφικών.

## 2. Παραδοσιακές Μέθοδοι Απεικόνισης

Το πεδίο των γραφικών στους υπολογιστές έχει βιώσει μια δυναμική και έντονη εξέλιξη με την πάροδο των χρόνων, με μια συνεχή και εποικοδομητική αναζήτηση τεχνικών απεικόνισης που μπορούν να αναπαραστήσουν όσο το δυνατόν πιο πιστά και ρεαλιστικά τρισδιάστατες σκηνές. Οι παραδοσιακές μέθοδοι απεικόνισης τέθηκαν ως οι ακρογωνιαίοι λίθοι για την ανάπτυξη αυτού του τομέα, δημιουργώντας τη βάση πάνω στην οποία κατασκευάστηκαν οι επόμενες καινοτομίες όπως το Gaussian splatting . Αυτό το κεφάλαιο παρέχει μια σύντομη αλλά σφαιρική και σαφή επισκόπηση αυτών των παραδοσιακών μεθόδων απεικόνισης, εξερευνώντας την ιστορική τους σημασία και τις θεμελιώδεις αρχές τους.

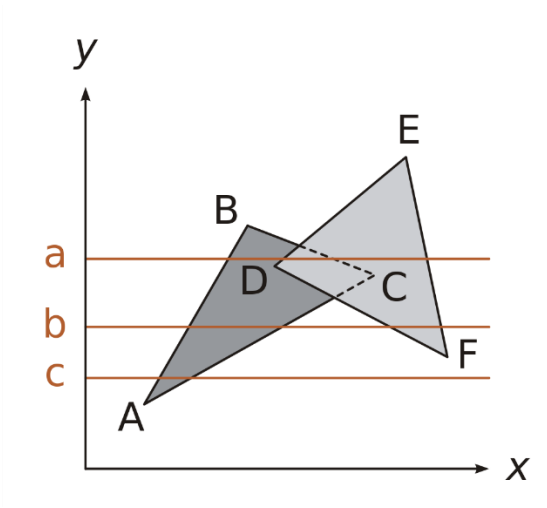
Για να κατανοήσουμε λοιπόν την πρόοδο που έχει επιτευχθεί στην απεικόνιση, είναι επιτακτικό να καταλάβουμε για αρχή το ιστορικό πλαίσιο στο οποίο εμφανίστηκαν οι παραδοσιακές μέθοδοι απεικόνισης. Όταν πρωτοξεκίνησε να γίνεται λόγος και να ασχολούμαστε με τα υπολογιστικά γραφικά, η απλότητα ,η σαφήνεια και η αποτελεσματικότητα ήταν καίριες λόγω της περιορισμένης και ανεπαρκούς για τα σημερινά δεδομένα υπολογιστικής ισχύς. Τεχνικές όπως το Scanline Rendering, το Ray Casting, το Ray Tracing, το Z-Buffer Rendering και το Painter's Algorithm ήταν από τις πιο πρωτοπόρες σε αυτήν την εποχή, ανοίγοντας το δρόμο για τους πιο πολύπλοκους αλγόριθμους απεικόνισης που έχουμε σήμερα.

### 2.1 Scanline Rendering

Το Scanline Rendering είναι μια θεμελιώδης τεχνική στα υπολογιστικά γραφικά που κατασκευάζει εικόνες γραμμή προς γραμμή και όχι πολύγωνο προς πολύγωνο ή pixel προς pixel , μέσω της διαδικασίας του «rasterization» ,δηλαδή της μετατροπής μιας ψηφιακής εικόνας από τη μορφή διανυσματικών γραφικών ,όπου τα χαρακτηριστικά από τις γραμμές ,τα σχήματα κλπ. χαρακτηρίζονται από μαθηματικούς τύπους ,σε ένα σύνολο από ανεξάρτητα pixel. Αυτή η μέθοδος διαιρεί τη σκηνή σε οριζόντιες γραμμές, καθορίζοντας τα χρώματα των pixel με τον διαδοχικό υπολογισμό τιμών σε αυτές τις γραμμές

Το κύριο πλεονέκτημα αυτής της μεθόδου έγκυται στο γεγονός ότι η ταξινόμηση των κορυφών κατά τη σάρωση του επιπέδου μειώνει τον αριθμό των συγκρίσεων μεταξύ των ακμών. Ένα ακόμα πλεονέκτημα είναι ότι δεν είναι απαραίτητο να μεταφραστούν οι

συντεταγμένες όλων των κορυφών από την κύρια μνήμη στην εργασιακή μνήμη — μόνο οι κορυφές που ορίζουν ακμές που τέμνουν την τρέχουσα γραμμή σάρωσης πρέπει να είναι στην ενεργό μνήμη, και κάθε κορυφή διαβάζεται μόνο μια φορά. Η κύρια μνήμη είναι συχνά πολύ αργή σε σύγκριση με τον σύνδεσμο μεταξύ της κεντρικής μονάδας επεξεργασίας και της μνήμης κρυφού διακομιστή, και έτσι η αποφυγή της επανεισόδου των κορυφών στην κύρια μνήμη μπορεί να παρέχει σημαντική επιτάχυνση. . Ενώ το Scanline Rendering είναι υπολογιστικά αποδοτική μέθοδος , αντιμετωπίζει προκλήσεις με ορισμένα γεωμετρικά περίπλοκα στοιχεία και δεν έχει τη δυνατότητα να χειριστεί εφέ γενικού και περιμετρικού φωτισμού. Παρ' όλα αυτά, η απλότητα και η ταχύτητά της την καθιστούν ιστορικά σημαντική στην απεικόνιση εικόνων.



Εικόνα 1: Παράδειγμα αλγορίθμου γραμμής σάρωσης<sup>[6]</sup>

## 2.2 Ray Casting - Ray Tracing

Το Ray casting είναι μια από τις αρχικές μεθόδους απεικόνισης που περιλαμβάνει τη σχεδίαση ακτίνων από το μάτι (κάμερα) προς τη σκηνή για να καθορίσει τα χρώματα των pixel. Αυτή η προσέγγιση παρέχει μια αρκετά ρεαλιστική απεικόνιση του φωτισμού και των σκιών σε σύγκριση με την απεικόνιση γραμμής σάρωσης (scanline rendering). Ωστόσο, το Ray casting μπορεί να απαιτεί αρκετή υπολογιστική ισχύ, ειδικά όταν οι σκηνές γίνονται πιο περίπλοκες. Η έμφαση της στην καταγραφή ρεαλιστικών εφέ φωτισμού την καθιστά ένα σημείο αναφοράς στην ανάπτυξη τεχνικών απεικόνισης

Το Ray tracing βασίζεται στις αρχές του Ray casting, εισάγοντας επιπλέον πολυπλοκότητα για να επιτύχει μια βέλτιστη φωτορεαλιστική αναπαράσταση. Αυτή η μέθοδος προσομοιάζει τη συμπεριφορά των ακτίνων φωτός καθώς αλληλεπιδρούν με επιφάνειες, δίνοντας έτσι τη δυνατότητα για την ακριβή και ρεαλιστική απεικόνιση αντανάκλασεων, διάθλασης ,διάχυσης του φωτός και περιμετρικού φωτισμού. Παρά τις

απαιτήσεις της σε υπολογιστική ισχύ, το Ray tracing θεωρείται η βάση για τη προσπάθεια βελτίωσης της φωτορεαλιστικής απεικόνιση.

Ποια είναι η διαφορά τους όμως; γιατί παραπάνω φαίνεται ότι είναι παρόμοιες , κι όμως όχι. Η αρχή του Ray casting είναι ότι οι ακτίνες εκπέμπονται και εντοπίζονται σε ομάδες βάσει ορισμένων γεωμετρικών περιορισμών , η ταχύτητα επεξεργασίας και απόδοσής αποτελέσματος είναι πολύ γρήγορη και αυτό μας καθιστά τη μέθοδο αυτή ικανή για επεξεργασία σε πραγματικό χρόνο. Όμως, η εικόνα που παράγεται δεν είναι και πολύ ρεαλιστική σε σχέση με τη μέθοδο του Ray tracing αφού δίνει εικόνες οι οποίες έχουν ένα στυλ παράθεσης «κουτιών» (Εικόνα 2) .Από την άλλη πλευρά στη μέθοδο του Ray tracing κάθε ακτίνα εκπέμπεται και χαράσσεται ξεχωριστά και έτσι κάθε σημείο, δηλαδή κάθε pixel , αντιστοιχεί σε μια ακτίνα .Αυτό όμως καθιστά τη μέθοδο αυτή πιο αργή και ακατάλληλη για επεξεργασία σε πραγματικό χρόνο λόγω της περιορισμένης υπολογιστικής ισχύς εκείνη την εποχή. Βέβαια η μέθοδος αυτή μπορούσε να αποδώσει πολύ ρεαλιστικές εικόνες ακόμα και πολύπλοκα σχήματα(Εικόνα 3) κάτι το οποίο αδυνατούσε η μέθοδος του Ray casting



Εικόνα 2: Σκηνή από το παιχνίδι Wolfenstein 3D (iD Software) (Ray casting).Είναι εμφανές πως τα αντικείμενα (όπλο) και ο εχθρός (σκύλος) δεν είναι τρισδιάστατα είναι σαν είναι κολλημένα στην εικόνα σε κατάλληλη κλίμακα<sup>[39]</sup>



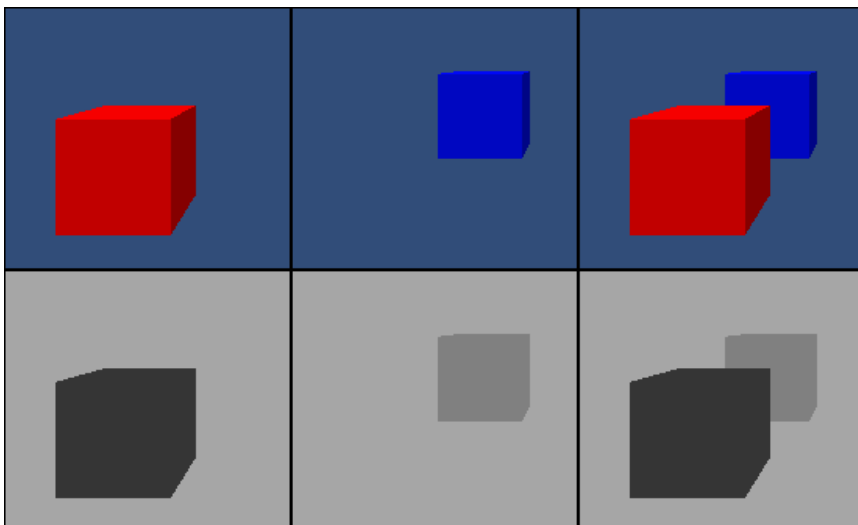
Εικόνα 3: Σκηνή από το παιχνίδι 7th Guest (Virgin Software/Trylobyte) (Ray tracing). Το αποτέλεσμα για την εποχή είναι πολύ ρεαλιστικό αλλά η κίνηση του παίκτη είναι περιορισμένη λόγω προκαθορισμένου «μονοπατιού» (εξαιτίας εικόνων που έχουν ήδη δημιουργηθεί) <sup>[39]</sup>

## 2.3 Z-Buffer Rendering

Η μέθοδος απεικόνισης Z-Buffer χρησιμοποιεί έναν ρυθμιστή βάθους για να καθορίσει την ορατότητα αντικειμένων σε μια σκηνή. Με τον συγκριτικό υπολογισμό των επιμέρους βαθών των αντικειμένων σε κάθε pixel, απεικονίζει μόνο αυτά που βρίσκονται πιο κοντά. Είναι μια μέθοδος που αξιοποιεί την εικόνα του χώρου (image space method). Οι μέθοδοι αυτές βασίζονται στο pixel που πρέπει να σχεδιαστεί σε 2D. Για αυτές τις μεθόδους, η πολυπλοκότητα του χρόνου εκτέλεσης έχει να κάνει με τον αριθμό των pixel πολλαπλασιασμένο με τον αριθμό των αντικειμένων. Και η πολυπλοκότητα χώρου είναι δύο φορές ο αριθμός των pixels επειδή απαιτούνται δύο πίνακες pixels, ένας για το ρυθμιστή του πλαισίου και ένας για τον ρυθμιστή του βάθους.

Πιο συγκεκριμένα, κατά την απεικόνιση σκηνών, κάθε pixel έχει συντεταγμένες X και Y (οριζόντιος και κατακόρυφος προσανατολισμός προς την κάμερα), καθώς και συντεταγμένη Z (απόσταση από την κάμερα). Το Z-Buffer είναι ένας διδιάστατος πίνακας (X και Y) που αποθηκεύει την τιμή Z για κάθε pixel στην οθόνη. Εάν ένα άλλο αντικείμενο πρέπει να απεικονιστεί στην ίδια θέση pixel, ο αλγόριθμος αντικαθιστά την προηγούμενη τιμή αν το νέο pixel είναι πιο κοντά στην κάμερα. Αυτός ο αλγόριθμος αυξάνει την ταχύτητα απεικόνισης για αδιαφανή αντικείμενα, αλλά τα διαφανή αντικείμενα δεν επωφελούνται, καθώς τα απομακρυσμένα αντικείμενα είναι μερικώς ορατά και πρέπει να απεικονιστούν πλήρως.

Συμπερασματικά, ενώ είναι αποτελεσματική για πολλά σενάρια, η Z-Buffer απεικόνιση έχει περιορισμούς στο χειρισμό και στην πιστή αναπαράσταση διαφανών ή ημιδιαφανών επιφανειών. Παρ' όλα αυτά, η απλότητα και η αποτελεσματικότητά της την καθιστούν δημοφιλή και ιδιαίτερα χρήσιμη, ιδίως σε εφαρμογές πραγματικού χρόνου.



Εικόνα 4: Αναπαράσταση βάθους με Z-Buffer αλγόριθμο<sup>[13]</sup>

## 2.4 Painter's Algorithm

Ο «αλγόριθμος του ζωγράφου» είναι μια τεχνική απεικόνισης που ταξινομεί τα αντικείμενα με βάση την απόστασή τους από τον θεατή και τα απεικονίζει με προτεραιότητα από τα πιο πίσω προς αυτά τα οποία βρίσκονται πιο μπροστά. Πιο συγκεκριμένα είναι ένας αλγόριθμος για τον προσδιορισμό ορατών επιφανειών σε τρισδιάστατα γραφικά που λειτουργεί σε μια βάση πολύγωνο προς πολύγωνο και όχι pixel προς pixel, σειρά προς σειρά ή περιοχή-προς-περιοχή. Όπως αναφέρθηκε ο αλγόριθμος του ζωγράφου δημιουργεί εικόνες ταξινομώντας τα πολύγωνα εντός της εικόνας με βάση το βάθος τους και τοποθετώντας κάθε πολύγωνο σε σειρά από το πιο μακρινό έως το πιο κοντινό αντικείμενο.

Αυτή η προσέγγιση προσομοιάζει τον τρόπο που ένας ζωγράφος θα ζωγράφιζε μια σκηνή, εξασφαλίζοντας τη σωστή θέση και θέαση των αντικειμένων καθώς και το πως κρύβεται το ένα από το άλλο. Ωστόσο, ο «αλγόριθμος του ζωγράφου» αντιμετωπίζει προκλήσεις με τον προσδιορισμό επιφανειών που τέμνονται και έχει περιορισμούς στο πως αντιμετωπίζει τη διαφάνεια. Παρά τα προβλήματα αυτά, η απλότητά του έχει συμβάλει στην ιστορική του σημασία για τα πρώιμα υπολογιστικά γραφικά.



Distant Background is painted first



Objects closer than the background are painted



Finally, Nearest Objects are painted



Εικόνα 5: *The Painter's Algorithm*<sup>[14]</sup>

### 3. Σημειοκεντρική Απεικόνιση (Point-Based Rendering)

Στον δυναμικό και ολοένα αναπτυσσόμενο κόσμο των τρισδιάστατων γραφικών, η εξέλιξη των τεχνικών απεικόνισης ήταν καθοριστικής σημασίας για την αντιμετώπιση των πολυπλοκοτήτων και των ιδιαίτερων απαιτήσεων των εικονικών περιβαλλόντων. Η Σημειοκεντρική Απεικόνιση (Point-Based Rendering, PBR) ξεχωρίζει ως ένα παράδειγμα που έχει επαναδιαμορφώσει και ουσιαστικά βελτιώσει δραστικά τον τρόπο με τον οποίο προσεγγίζουμε την αναπαράσταση και οπτικοποίηση τρισδιάστατων σκηνών. Τώρα λοιπόν θα εξερευνήσουμε και θα μελετήσουμε τα θεμέλια και τις προηγμένες εξελίξεις στην Σημειοκεντρική Απεικόνιση, θέτοντας με αυτόν τον τρόπο τις βάσεις για να καταλήξουμε στο θέμα που μας απασχολεί στη συγκεκριμένη εργασία δηλαδή το Gaussian Splatting.

Για να κατανοήσουμε τη σημασία της Σημειοκεντρικής Απεικόνισης, είναι καίριο και αναγκαίο να ανιχνεύσουμε και να εξετάσουμε την ιστορική πορεία της και πως τελικά καταλήξαμε σε αυτήν. Από την εποχή των πρώιμων γεωμετρικών μεθόδων αναπαράστασης μέχρι την εμφάνιση των νεότερων τεχνικών «ραστεροποίησης»

(μετατροπή μιας ψηφιακής εικόνας από τη μορφή διανυσματικών γραφικών ,όπου τα χαρακτηριστικά από τις γραμμές ,τα σχήματα κλπ. χαρακτηρίζονται από μαθηματικούς τύπους ,σε ένα σύνολο από ανεξάρτητα pixel), έγιναν εμφανείς οι περιορισμοί των παραδοσιακών μεθόδων απεικόνισης στην αντιμετώπιση πολύπλοκων και μεγάλης κλίμακας μεγέθους σκηνών. Η Σημειοκεντρική Απεικόνιση προέκυψε ως απάντηση και λύση σε αυτές τις προκλήσεις και σε αυτά τα προβλήματα, εισάγοντας μια νέα προσέγγιση που βασίζεται στην έννοια της χρήσης μεμονωμένων σημείων ως των βασικών στοιχείων για την αναπαράσταση της τρισδιάστατης σκηνής.

### 3.1 Βασικές Αρχές και Πλεονεκτήματα της Σημειοκεντρικής Απεικόνισης

Οι βασικές και πρωτεύουσες αρχές της Σημειοκεντρικής Απεικόνισης περιλαμβάνουν την αναπαράσταση μιας τρισδιάστατης σκηνής ως ένα σύνολο μεμονωμένων σημείων, καθένα από τα οποία φέρει βασικά χαρακτηριστικά, όπως θέση, χρώμα και ίσως επιπλέον πληροφορίες, όπως «surface normals», δηλαδή διανυσμάτων που βρίσκουμε στις επιφάνειες και μας βοηθούν να καταλάβουμε τη φύση των επιφανειών αυτών . Αντίθετα με τις παραδοσιακές μεθόδους που βασίζονται σε αναπαραστάσεις πλέγματος (mesh) ή στοιχείων όγκου (voxel) , η Σημειοκεντρική Απεικόνιση λειτουργεί απευθείας σε αυτά τα σημεία, προσφέροντας πλεονεκτήματα που έχουν να κάνουν με ευελιξία, προσαρμοστικότητα σε μεταβαλλόμενες πολυπλοκότητες σκηνών και αποτελεσματικότητα σε διάφορα σενάρια.

Η Σημειοκεντρική Απεικόνιση λοιπόν φέρνει στο προσκήνιο ένα σύνολο σημαντικών πλεονεκτημάτων που την διακρίνουν από τις παραδοσιακές μεθόδους απεικόνισης. Η ενσωματωμένη και ανεμπόδιστη προσαρμοστικότητα σε ακανόνιστες ,δυναμικές και πολύπλοκες σκηνές, η αποτελεσματική αναπαράσταση αραιών περιβαλλόντων και η δυνατότητα καταγραφής και απόδοσης λεπτομερειών είναι μερικά από τα θετικά της στοιχεία που την έχουν κάνει να ξεχωρίσει. Ωστόσο, προκλήσεις και προβλήματα όπως η επικάλυψη που οδηγεί σε αλλοίωση, η διαχείριση της πυκνότητας των σημείων καθώς και η διαχείριση των επιδράσεων της διάχυσης του φωτός και του περιμετρικού φωτισμού είναι μερικά από τα καίρια ζητήματα που αντιμετωπίζουν και απασχολούν τόσο τους ερευνητές όσο και τους επαγγελματίες.



## 3.2 Εφαρμογές της Σημειοκεντρικής Απεικόνισης

Η ευελιξία της Σημειοκεντρικής Απεικόνισης επεκτείνεται σε διάφορους τομείς, συμπεριλαμβανομένης της επιστημονικής οπτικοποίησης, των υπολογιστικών γραφικών και της εικονικής πραγματικότητας. Υπάρχουν ποικίλες πρακτικές εφαρμογές όπου η Σημειοκεντρική Απεικόνιση έχει επιδείξει και αποδείξει την αποτελεσματικότητά της. Από την αναπαράσταση πολύπλοκων επιστημονικών συνόλων δεδομένων έως τις υπερβολικά ρεαλιστικές και ταυτοχρόνως «τρομακτικές» εμπειρίες σε εικονικά περιβάλλοντα, η Σημειοκεντρική Απεικόνιση διαδραματίζει κρίσιμο ρόλο στην ενίσχυση και βελτίωση της οπτικής πιστότητας και ρεαλισμού των αναπαριστάμενων σκηνών και της αλληλεπίδρασης μεταξύ του θεατή και του χώρου που αναπαρίσταται.

Καθώς εξερευνούμε, εξετάζουμε και εμβαθύνουμε στον χώρο της Σημειοκεντρικής Απεικόνισης, γίνεται εμφανές ότι η συνεχής προσπάθεια για τελειοποίηση και καινοτομία είναι απαραίτητες για να καταφέρουμε κάποια στιγμή να αντιμετωπίσουμε και να εκμηδενίσουμε τους περιορισμούς της και να επιλύσουμε κάθε πρόβλημα που δημιουργείται. Το Gaussian Splatting εμφανίζεται ως μια πολλά υποσχόμενη εξέλιξη στον χώρο αυτό, προσφέροντας λύση σε ορισμένες από τις προκλήσεις που σχετίζονται με τη Σημειοκεντρική Απεικόνιση. Συνεχίζοντας ,σε αυτή την εργασία θα εμβαθύνουμε στις θεωρητικές βάσεις, τα πλεονεκτήματα και τις πρακτικές σκέψεις υλοποίησης του Gaussian Splatting, επισημαίνοντας και διασαφηνίζοντας τη δυνατότητά της μεθόδου αυτής να αντιμετωπίσει τις πολυπλοκότητες μεγάλης κλίμακας και ιδιαίτερων λεπτομερειών.

## 3.3 Point Cloud – «Σύννεφο σημείων»

Στον ευρύ χώρο των τρισδιάστατων γραφικών, η αναπαράσταση πολύπλοκων σκηνών έχει δει καινοτόμες και ριζοσπαστικές μεταβολές. Τα point cloud είναι ουσιαστικά μια συλλογή μεμονωμένων σημείων στο τρισδιάστατο χώρο και έχουν προκύψει ως ένα θεμελιώδες παράδειγμα αναπαράστασης. Θα εξετάσουμε, ευθύς αμέσως , τις αρχές και τις εφαρμογές της αναπαράστασης σε point cloud, δημιουργώντας έτσι τα θεμέλια και τις βάσεις για μια συνολική κατανόηση του ρόλου τους στις τεχνικές απεικόνισης και τη σχέση τους με το Gaussian Splatting.



### 3.3.1 Θεμελιώδεις αρχές του Point Cloud – «Σύννεφου σημείων»

Τα point clouds λειτουργούν ως ένα ευέλικτο και πλήρως επεκτάσιμο μέσο καταγραφής και μεταφοράς χωρικών πληροφοριών. Τα point clouds κατασκευάζονται από πραγματικά δεδομένα ή συνθετικά περιβάλλοντα. Κάθε σημείο εντός ενός σύννεφου καλύπτει και αποδίδει σημαντικά χαρακτηριστικά, όπως χωρικές συντεταγμένες, χρώμα, και πολλές φορές επιπλέον χαρακτηριστικά όπως ένταση ή αντανάκλαση.

Η κατανόηση των διαφόρων μεθόδων που χρησιμοποιούνται για την απόκτηση δεδομένων για τα point clouds είναι κρίσιμη για το να αντιληφθούμε τις διάφορες εφαρμογές τους. Τέτοιες τεχνικές απόκτησης δεδομένων είναι το LiDAR (Light Detection and Ranging), η φωτογραμμετρία, η σάρωση με δομημένο φως και οι αισθητήρες βάθους. Κάθε μέθοδος έχει τα πλεονεκτήματά της και τους περιορισμούς της, επηρεάζοντας την ποιότητα και την πυκνότητα του αποτελέσματος των point cloud .

### 3.3.2 Πλεονεκτήματα και Προκλήσεις της Αναπαράστασης με Point Clouds

Η χρήση point clouds στα γραφικά τρισδιάστατων αντικειμένων εισάγει αρκετά πλεονεκτήματα. Η ικανότητά τους να καταγράφουν πιστά και με ακρίβεια τη γεωμετρία και την εμφάνιση αντικειμένων του πραγματικού κόσμου, να χειρίζονται επιτυχώς μη ομοιόμορφα κατανεμημένα δεδομένα και να διευκολύνουν την αποθήκευση και τη μετάδοση τους είναι τα χαρακτηριστικά που καθιστούν τα point clouds μια σοφή επιλογή για διάφορες εφαρμογές. Η αυξανόμενη δημοφιλία της αναπαράστασης με point clouds έγγυται ακόμα και στο γεγονός ότι αποτελούν θεμελιώδες στοιχείο στις ροές μετεπεξεργασίας των τρισδιάστατων περιβαλλόντων αφού τα αρχικά δεδομένα μπορούν να επεξεργαστούν ευκολά και γρήγορα και να προκύψει μια πιο λεπτομερής, ακριβής και ρεαλιστική τρισδιάστατη αναπαράστασή.

Ενώ τα point clouds προσφέρουν ένα ισχυρό πλαίσιο αναπαράστασης, αντιμετωπίζουν αρκετές προκλήσεις. Η διαχείριση μεγάλου όγκου δεδομένων, η αντιμετώπιση του θορύβου και η εξασφάλιση της συνέπειας στην πυκνότητα των σημείων στις διάφορες σκηνές χρήζουν μεγάλης προσοχής.

### 3.3.3 Εφαρμογές των Point Clouds στα 3D-γραφικά

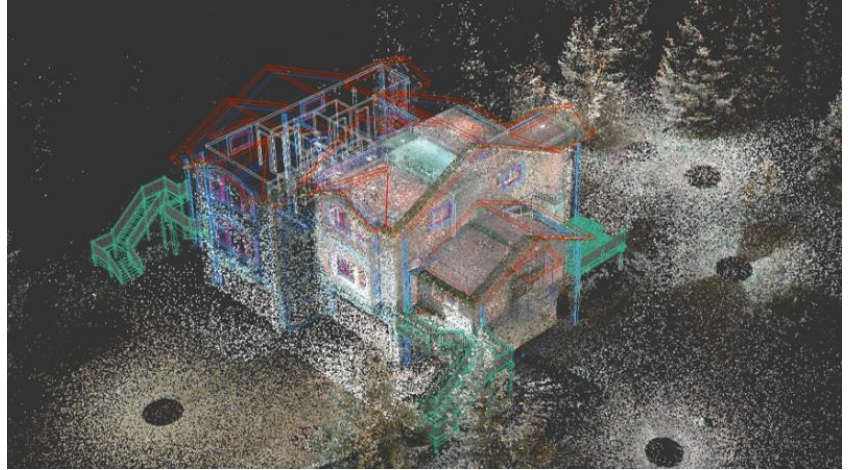
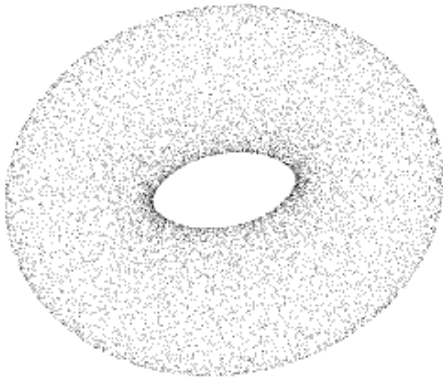
Η ενσωμάτωση των point clouds με τεχνικές απεικόνισης αποτελεί σημαντική προοδευτική κίνηση στην οπτικοποίηση πολύπλοκων σκηνών μεγάλης κλίμακας. Η αναπαράσταση με point clouds συνδυάζεται με προσεγγίσεις απεικόνισης, συμπεριλαμβανομένων του Point-Based Rendering (PBR) και κατά συνέπεια του Gaussian Splatting. Τα μοναδικά χαρακτηριστικά των point clouds, όπως η ανομοιόμορφη κατανομή

σημείων και οι μεταβαλλόμενες πυκνότητες, προσφέρουν τόσο ευκαιρίες όσο και προκλήσεις στην απεικόνιση και μετάδοση της πληροφορίας.

Τα point clouds βρίσκουν εφαρμογές σε διάφορους και ποικίλους τομείς των γραφικών 3D. Από εφαρμογές στον αστικό σχεδιασμό και στη μοντελοποίηση του φυσικού περιβάλλοντος, μέχρι την αποτύπωση της πολιτιστικής κληρονομιάς και την επαυξημένη και εικονική πραγματικότητα.

Το μέλλον της αναπαράστασης με point clouds στα γραφικά 3D έχει συναρπαστικές πιθανότητες εξέλιξης και βελτίωσης, καθοδηγούμενο από συνεχείς καινοτομίες. Οι εξελίξεις στις τεχνολογίες απόκτησης δεδομένων, όπως το LiDAR και οι σαρωτές με δομημένο φως, υπόσχονται υψηλότερης ποιότητας και πιο προσβάσιμα δεδομένα για τα point clouds. Οι προσπάθειες για αποτελεσματικές τεχνικές συμπίεσης δεδομένων θα καταφέρουν να αντιμετωπίσουν τον τεράστιο όγκο δεδομένων, επιτρέποντας μετάδοση της πληροφορίας και απεικόνιση σκηνών σε πραγματικό χρόνο. Η ενσωμάτωση τους με την τεχνητή νοημοσύνη, συμπεριλαμβανομένης της μηχανικής μάθησης και της «υπολογιστικής όρασης», οδηγεί σε προοπτικές για αυτοματοποιημένη αναγνώριση βασικών χαρακτηριστικών και βελτιωμένη σε σημαντικό βαθμό επεξεργασία των point clouds. Η εξέλιξη εργαλείων αλληλεπίδρασης με το θεατή στοχεύει στην ουσιαστική βελτίωση της εμπειρίας του χρήστη και τη διευκόλυνση της απροβλημάτιστης συνεργασίας σε διάφορες πλατφόρμες. Οι ενημερώσεις σε πραγματικό χρόνο για δυναμικές και πολύπλοκες σκηνές, οι λύσεις επεξεργασίας και αποθήκευσης στο cloud για εξοικονόμηση χώρου, και η αξιοποίηση των δυνατοτήτων των point clouds με συσκευές τελευταίας τεχνολογίας αιχμής δημιουργούν περισσότερο χώρο για έρευνα.

Τα point clouds λοιπόν παρέχουν τα αρχικά δεδομένα με χωρικές πληροφορίες για κάθε σημείο ενώ η τεχνική Gaussian Splatting είναι μια μέθοδος απεικόνισης που βοηθά στη μετατροπή αυτών των διακριτικών δεδομένων σημείων σε μια ομαλή, σαφή, αισθητικά ωραία και ρεαλιστική αναπαράσταση. Μαζί, διαδραματίζουν κρίσιμο ρόλο στην οπτικοποίηση και απεικόνιση πολύπλοκων 3D σκηνών που παρουσιάζουν ανομοιομορφίες, ιδιαιτερότητες και πολυπλοκότητες στην κατανομή των σημείων.



Εικόνα 6 : Σχεδιασμός με point clouds<sup>[20],[24]</sup>

## 4. Gaussian Splatting

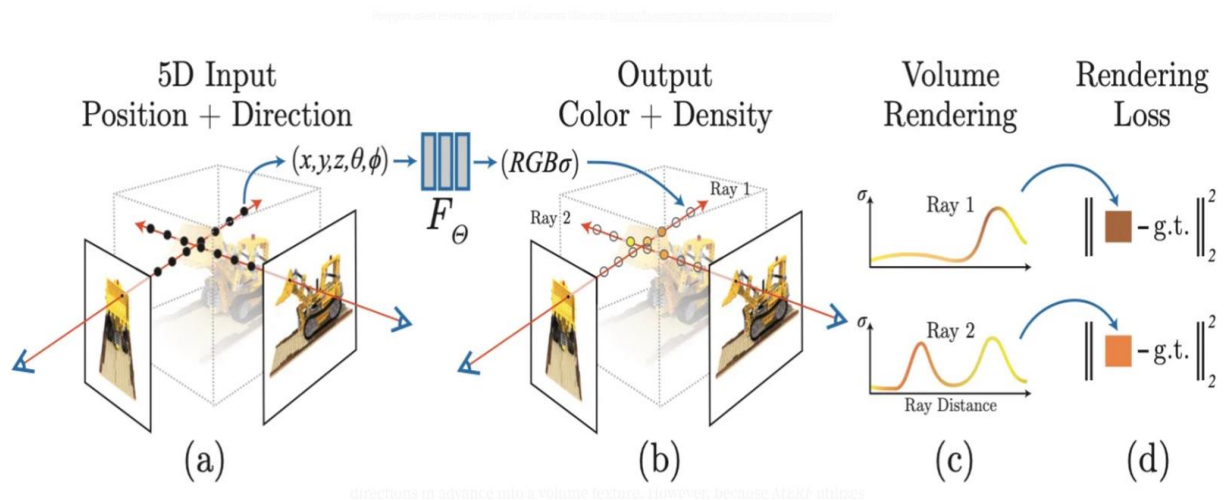
Το Gaussian splatting είναι μια τεχνική για την αναπαράσταση σκηνών στις τρεις διαστάσεις και τη δημιουργία νέων προβολών. Σε αντίθεση με τις παραδοσιακές μεθόδους αναπαράστασης που εξαρτώνται από δομημένα πλέγματα ή από νευρωνικούς όγκους, αυτή η τεχνική χρησιμοποιεί την ευελιξία και την προσαρμοστικότητα των τρισδιάστατων Γκαουσιανών. Αυτές οι μέθοδοι, που ανήκουν στην κατηγορία του Radiance Field όπως και το NeRF, προσφέρουν ταχύτερη εκπαίδευση και απεικόνιση, διατηρώντας παρόμοια ή και πολύ συχνά καλύτερη ποιότητα. Επιπλέον, είναι μέθοδοι πιο εύκολες στην κατανόηση, στη χρήση και τη μετεπεξεργασία. Το Gaussian splatting μπορεί να θεωρηθεί ως μια εναλλακτική προσέγγιση σε σχέση με μοντέλα όπως το NeRF, το οποίο υπήρξε πρωτοποριακό μοντέλο στο παρελθόν. Το Gaussian splatting θα οδηγήσει με τη σειρά του σε πολλές νέες έρευνες που θα πραγματευτούν την ρεαλιστική, πιστή και σε πραγματικό χρόνο αναπαράσταση μιας τρισδιάστατης πραγματικότητας για ποικίλες χρήσεις. Ας δούμε λοιπόν γιατί είναι τόσο ιδιαίτερο και ξεχωριστό και γιατί θεωρούμε πως είναι καλύτερο από το NeRF και έχει περισσότερες προοπτικές για εξέλιξη.

## 4.1 NeRF (Neural Radiance Field)

Για να ξεκινήσουμε να συζητάμε για τη μέθοδο τρισδιάστατης αναπαράστασης με NeRF πρέπει πρώτα να ερευνήσουμε γιατί οδηγηθήκαμε σε αυτή τη μέθοδο και δεν επαναπαυτήκαμε σε τεχνικές όπως η φωτογραμμετρία. Η φωτογραμμετρία είναι μια τεχνική αναπαράστασης, για την ακρίβεια τεχνική για τη δημιουργία τρισδιάστατων πολυγώνων αρχικά, που ανακατασκευάζει τρισδιάστατες δομές από δισδιάστατες εικόνες, αναλύοντας και επεξεργάζοντας τις γεωμετρικές τους σχέσεις. Για να δημιουργήσει τα τρισδιάστατα μοντέλα αναλύει πολλαπλές εικόνες από διάφορες οπτικές γωνίες, κάτι το οποίο επιτυγχάνεται με το να φωτογραφίζεται ένα αντικείμενο από διαφορετικές και πολλές γωνίες, παραδείγματος χάρη περιστρέφοντάς το σε έναν περιστροφικό δίσκο κατά τη διάρκεια της λήψης των εικόνων. Ωστόσο, αυτή η τεχνική έχει περιορισμούς, όπως η δυσκολία στην αποτύπωση με σαφήνεια διάφορων λεπτομερειών, η αντιμετώπιση μη-σταθερών και δυναμικών σκηνών και η ευαισθησία στις συνθήκες φωτισμού που μπορούν να αλλοιώσουν σημαντικά το τελικό αποτέλεσμα. Πιο συγκεκριμένα ενώ η φωτογραμμετρία είναι χρήσιμη για τη σάρωση αντικειμένων, δεν μπορεί να αποτυπώσει σκηνές χωρίς περίγραμμα, όπως ο ουρανός, ή λεπτομέρειες σε απομακρυσμένες σκηνές και ακόμα επειδή η βελτιστοποίηση πραγματοποιείται με την επίλυση συμβατικών και κλασσικών προβλημάτων τέτοιου είδους, δεν είναι δυνατή η ακριβής δημιουργία τρισδιάστατων πολυγώνων για ορισμένες περιπτώσεις χρήσης, όπως οι αντανakλάσεις και οι καθρέπτες.

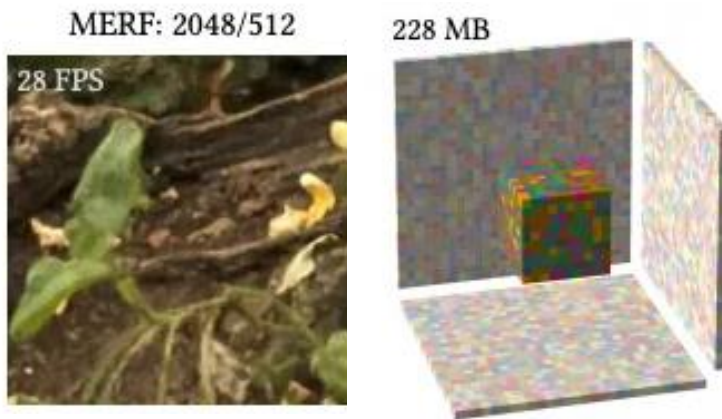
Τα «Νευρωνικά Πεδία Ακτινοβολίας» (NeRF) αντιμετωπίζουν αυτά τα μειονεκτήματα χρησιμοποιώντας το λεγόμενο «deep learning» για να μοντελοποιήσουν απευθείας πολύπλοκες και μεγάλης κλίμακας τρισδιάστατες σκηνές από διάφορες εικόνες. Αυτή η προσέγγιση ξεπερνά προκλήσεις που σχετίζονται με την αποτύπωση λεπτομερειών και τη διαχείριση δυναμικών ή ιδιαίτερων και παραμορφωμένων αντικειμένων. Τα NeRF παρέχουν μια πιο ευέλικτη, πιστή και ακριβή αναπαράσταση των τρισδιάστατων σκηνών, αποδεικνύοντας πως είναι ιδιαίτερα αποτελεσματικά σε περιπτώσεις όπου η παραδοσιακή φωτογραμμετρία αντιμετωπίζει προκλήσεις.

Η βασική και ουσιαστική ιδέα είναι να εκπαιδύσουμε ένα μοντέλο το οποίο θα αναπαριστά τη σκηνή με έμμεσο τρόπο, και συγκεκριμένα με την αναπαράσταση ενός «όγκου» που περιέχει πληροφορίες για την πυκνότητα και το χρώμα σε κάθε συγκεκριμένη θέση. Για να αποτυπώσουμε τη σκηνή από μια συγκεκριμένη οπτική γωνία, δειγματοληπούμε αυτόν τον όγκο για να υπολογίσουμε και να καταχωρήσουμε το χρώμα της τελικής εικόνας σε αυτό το δεδομένο σημείο. Αυτή η μέθοδος αποτύπωσης είναι σε θέση να αναπαράγει με ακρίβεια ανοιχτές σκηνές, καθώς και ανακλαστικά αντικείμενα. Ωστόσο, το μειονέκτημα είναι ότι το NeRF είναι αρκετά αργό στη διαδικασία αναπαράστασής και γι' αυτό το λόγο δε μπορεί να χρησιμοποιηθεί σε πραγματικό χρόνο .



Εικόνα 7: Μετάδοση πληροφορίας και αναπαράσταση στη μέθοδο NeRF<sup>[42]</sup>

Για να καταφέρουμε να επιταχύνουμε τη μέθοδο NeRF προτάθηκε μια νέα παραλλαγή του, το MERF (Memory-Efficient Radiance Fields) το οποίο επιτρέπει την επεξεργασία σε πραγματικό χρόνο ενσωματώνοντας εκ των προτέρων τα αποτελέσματα των ακτίνων που έχουν ήδη ιχνηλατηθεί σε πολλαπλές κατευθύνσεις σε ένα volume texture (=ένα ειδικό είδος πόρου που βοηθά στην αποθήκευση και χρήση πληροφοριών 3D όγκου σε μια 2D μορφή). Τα volume textures λειτουργούν παίρνοντας ένα τρισδιάστατο σχήμα και κόβοντάς το σε τομές που στη συνέχεια τοποθετούνται σε ένα πλέγμα σε μια 2D μορφή). Ωστόσο, επειδή το MERF χρησιμοποιεί volume textures με σταθερό μέγεθος, όπως το  $512 \times 512 \times 512$ , και υφές υψηλής ανάλυσης όπως το  $2048 \times 2048$ , το τελικό αποτέλεσμα υφίσταται μείωση στη λεπτομέρεια, την σαφήνεια και την πιστότητα του σχήματος.



Εικόνα 8: Αναπαράσταση με τη μέθοδο MERF<sup>[25]</sup>

## 4.2 Από το NeRF (Neural Radiance Field) στο 3D Gaussian Splatting

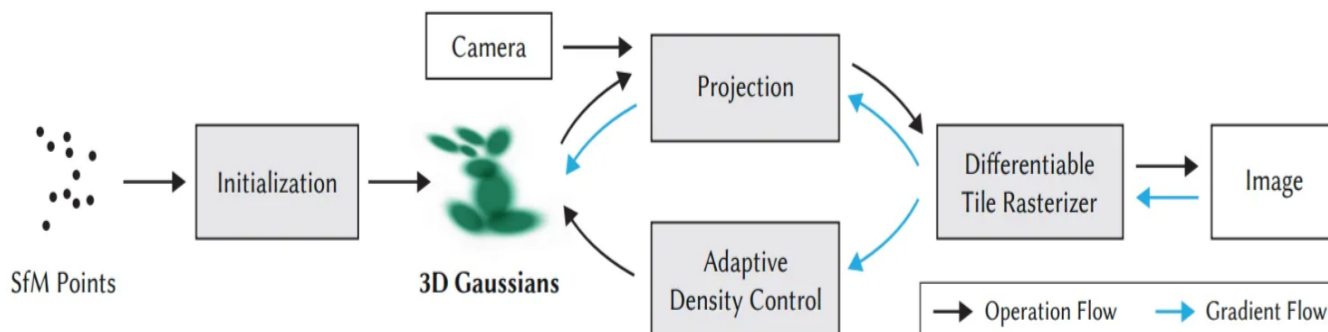
Όπως προαναφέραμε, το Neural Radiance Field (NeRF) είναι μια μέθοδος στα υπολογιστικά γραφικά που χρησιμοποιεί νευρωνικά δίκτυα για να μοντελοποιήσει και να απεικονίσει πολύπλοκες τρισδιάστατες σκηνές απευθείας από δισδιάστατες εικόνες. Με το να αποτυπώνει ταυτόχρονα και τη γεωμετρία και την εμφάνιση σε ένα ενοποιημένο μοντέλο, το NeRF επιτυγχάνει αναπαράσταση υψηλής ποιότητας και ρεαλιστικότητας των σκηνών. Ωστόσο, το μειονέκτημα του NeRF βρίσκεται στις υπολογιστικές του απαιτήσεις, καθιστώντας υπερβολικά βαρύ για εφαρμογές πραγματικού χρόνου ή σκηνές μεγάλης κλίμακας. Το 3D Gaussian Splatting αντιμετωπίζει αυτά τα μειονεκτήματα χρησιμοποιώντας 3D Gaussians, εκμεταλλευόμενο τα χαρακτηριστικά τους όπως οι ανισοτροπικές συνδιακυμάνσεις, οι θέσεις και οι «αδιαφάνειες», για την αποτελεσματική μοντελοποίηση πολύπλοκων σκηνών. Τα χαρακτηριστικά αυτά αποκτούνται μέσω μιας μεθόδου προσέγγισης μηχανικής μάθησης. Ωστόσο, η απεικόνιση δεν απαιτεί καμία βαριά επεξεργασία, παρέχοντας γρήγορη και πειστική απεικόνιση μέσω ενός «ραστεροποιητή βασισμένο σε πλακίδια»(tile-based rasterizer).



### 4.3 Τα βασικά για το 3D Gaussian Splatting

Κατά βάση και αρχικά η μεγαλύτερη συνεισφορά αυτής της μεθόδου είναι η υψηλή ταχύτητα απεικόνισης και αναπαράστασης τρισδιάστατων σκηνών, η οποία οφείλεται στην προσαρμοσμένη υλοποίηση ενός αλγορίθμου χρησιμοποιώντας προσαρμοσμένους CUDA kernels(4.3.1). Το Gaussian splatting δεν εμπλέκει καθόλου νευρωνικά δίκτυα αφού κάθε σκηνή που αναπαρίσταται είναι ένα σύνολο σημείων στον χώρο. Είναι αρκετά ενδιαφέρον ότι μια τέτοια μέθοδος γίνεται ολοένα και πιο δημοφιλής στον χώρο αυτό τον οποίο τον απασχολεί έντονα η τεχνητή νοημοσύνη και όλες οι έρευνες στρέφονται γύρω από μοντέλα που αποτελούνται από πάρα πολλές και διαφορετικές παραμέτρους. Η ιδέα αυτή προέρχεται από το «Surface splatting», για το οποίο είχε γίνει ήδη αναφορά από το μακρινό πλέον 2001, και για αυτό αποτελεί μια ενδιαφέρον παραλλαγή των κλασικών προσεγγίσεων της υπολογιστικής οπτικοποίησης που μπορεί να δώσει λύσεις σε διάφορα προβλήματα στο χώρο της τρισδιάστατης αναπαράστασης.

Η βασική «αρχιτεκτονική δομή» του Gaussian splatting αποτυπώνεται με σαφήνεια και συντομία στο παρακάτω σχήμα :



Εικόνα 9: Ξεκινώντας από ένα αραιό point cloud που προέκυψε από το SfM (Structure from Motion), η διαδικασία βελτιστοποίησης χρησιμοποιεί έναν γρήγορο και αποδοτικό « απεικονιστή βασισμένο σε πλακίδια»(tile-based renderer) και δημιουργεί ένα σύνολο από 3D Γκαουσιανές, των οποίων η πυκνότητά τους ελέγχεται με προσαρμοστικό τρόπο.<sup>[1]</sup>



Εικόνα 10: Σύγκριση διαφορετικών μεθόδων αναπαράστασης σε υψηλή ποιότητα όσον αφορά τη ταχύτητα απεικόνισης, το χρόνο εκπαίδευσης και τη ποιότητα της εικόνας ( Peak signal-to-noise ratio, όσο μεγαλύτερο τόσο το καλύτερο)<sup>[1]</sup>

### 4.3.1 CUDA kernels

Τα CUDA kernels είναι εξειδικευμένες συναρτήσεις ή διαδικασίες που εκτελούνται σε μια μονάδα επεξεργασίας γραφικών (GPU) χρησιμοποιώντας την πλατφόρμα παράλληλου υπολογισμού CUDA (Compute Unified Device Architecture) της NVIDIA. Η CUDA επιτρέπει στους προγραμματιστές να αξιοποιήσουν με τον καλύτερο δυνατό τρόπο τη δύναμη των GPU για παράλληλο προγραμματισμό γενικού σκοπού, πέρα από τις κλασικές τους ικανότητες στον τομέα της γραφικής απεικόνισης.

Στο πλαίσιο της Gaussian splatting ή άλλων υπολογισμών επιταχυνόμενων από GPU, τα CUDA kernels χρησιμοποιούνται για να πραγματοποιήσουν παράλληλη επεξεργασία στη GPU. Ένα kernel είναι μια συνάρτηση που σχεδιάζεται να εκτελείται από πολλά threads παράλληλα. Αυτά τα threads οργανώνονται σε blocks και τα blocks περαιτέρω οργανώνονται σε ένα πλέγμα. Το σύστημα εκτέλεσης της CUDA διαχειρίζεται την εκτέλεση αυτών των threads στη GPU.

Η παράλληλη φύση των CUDA kernels επιτρέπει την αποτελεσματική επεξεργασία μεγάλων συνόλων δεδομένων και πολύπλοκων υπολογισμών, καθιστώντας τα κατάλληλα για εργασίες όπως η απεικόνιση και η προσομοίωση σε πραγματικό χρόνο. Στην περίπτωση της Gaussian splatting, τα CUDA kernels μπορεί να χρησιμοποιηθούν για την επιτάχυνση του υπολογισμού των 3D Gaussian splats και άλλων σχετικών λειτουργιών, εκμεταλλευόμενα τις δυνατότητες παράλληλης επεξεργασίας της GPU για να επιτύχουν ταχύτερη απόδοση.



## 4.4 Gaussian Splatting για την αναπαράσταση τρισδιάστατων σκηνών

Όπως αναφέραμε και προηγουμένως, με το Gaussian Splatting ένας τρισδιάστατος κόσμος αναπαρίσταται με ένα σύνολο από τρισδιάστατα σημεία, για την ακρίβεια εκατομμύρια από αυτά, σε ένα εύρος περίπου 0,5-5 εκατομμύρια. Κάθε σημείο είναι ουσιαστικά μια τρισδιάστατη Gaussian με τις δικές της μοναδικές παραμέτρους που προσαρμόζονται για κάθε σκηνή, έτσι ώστε οι αποτυπώσεις αυτής της σκηνής να είναι πολύ παρόμοιες και σχεδόν πίστες με τις γνωστές εικόνες του συνόλου δεδομένων. Ο στόχος λοιπόν είναι να δημιουργηθεί μια ρεαλιστική τρισδιάστατη σκηνή από λίγες αραιές εικόνες. Οι παραδοσιακές μέθοδοι αντιμετωπίζουν δυσκολίες με τα αραιά δεδομένα, ιδίως όταν πρέπει να εκτιμηθούν τα surface normals. Αντί για αυτό, αυτή η προσέγγιση χρησιμοποιεί τρισδιάστατες Gaussians— μαθηματικές αναπαραστάσεις που καταγράφουν τη δομή της σκηνής χωρίς την ανάγκη για surface normals.

Κάθε 3D Gaussian παραμετροποιείται ως εξής:

- **Μέση τιμή (Mean)  $\mu$**  : Ερμηνεύεται ως η θέση στους άξονες  $x, y, z$ .
- **Πίνακας Συνδιακύμανσης (Covariance)  $\Sigma$**  : Περιέχει τις πληροφορίες σχετικά με την περιστροφή και την κλίμακα
- **Διαφάνεια (Opacity)  $\sigma(\alpha)$**  : Εφαρμόζεται ως μια σιγμοειδής συνάρτηση για να απεικονίσει την παράμετρο αυτή στο διάστημα  $[0,1]$
- **Παράμετροι Χρώματος** : Μπορεί να περιλαμβάνουν είτε 3 τιμές (R, G, B) είτε συντελεστές σφαιρικών αρμονικών (Spherical Harmonics (SH)) για την περιγραφή του χρώματος.

Όσον αφορά τον πίνακα συνδιακύμανσης, επιλέγεται να είναι ανισοτροπικός κατά σχεδιασμό και πρακτικά, αυτό σημαίνει ότι ένα τρισδιάστατο σημείο μπορεί να θεωρηθεί και ουσιαστικά είναι ένα ελλειψοειδές που μπορεί να περιστρέφεται και επεκτείνεται σε οποιαδήποτε κατεύθυνση στο χώρο. Πιο συγκεκριμένα, κάθε Gaussian είναι κεντραρισμένη σε ένα σημείο (τη μέση τιμή) και η θέση της στον χώρο καθορίζεται από έναν τρισδιάστατο πίνακα συνδιακύμανσης. Ωστόσο, για να αποδώσουμε τα 3D σχέδια αποτελεσματικά και με ακρίβεια, απαιτείται η προβολή τους στο 2D. Η πρόκληση λοιπόν περιορίζεται πλέον στην προβολή αυτών των τρισδιάστατων σχημάτων σε μια 2D επιφάνεια για την διαμόρφωση, επεξεργασία και τελική αναπαράσταση της σκηνής.



Εικόνα 11: Τα κέντρα (μέσες τιμές) των Gaussians<sup>[40]</sup>

Οι πίνακες συνδιακύμανσης όμως έχουν φυσική έννοια μόνο όταν είναι θετικά ημιορισμένοι. Επομένως, η μέθοδος του «gradient descent» [=είναι ένας αλγόριθμος βελτιστοποίησης για τον εντοπισμό ενός τοπικού ελάχιστου μιας διαφορίσιμης συνάρτησης. Στον χώρο της μηχανικής μάθησης, αυτή η μέθοδος χρησιμοποιείται απλώς για την εύρεση των τιμών των παραμέτρων μιας συνάρτησης (συντελεστές) που ελαχιστοποιούν μια συνάρτηση κόστους όσο το δυνατόν περισσότερο] δε μπορεί να αξιοποιηθεί, καθώς μπορεί να οδηγήσει σε μη έγκυρους πίνακες επειδή όπως γίνεται κατανοητό είναι δύσκολο να επιβληθούν και να εφαρμοστούν τέτοιοι περιορισμοί στον πίνακα αυτόν απευθείας. Αυτό επιτυγχάνεται με έναν επιδέξιο και έξυπνο τρόπο, χρησιμοποιώντας πίνακες για τη κλίμακα (S) και για τη περιστροφή (R) για να περιγράψουν αυτά τα ελλειψοειδή. Αυτή η μέθοδος επιτρέπει ανεξάρτητη επεξεργασία και βελτιστοποίηση, προσαρμόζοντας τα ελλειψοειδή σε διάφορα σχήματα στη σκηνή προς αναπαράσταση η οποία τελικά θα εξελιχθεί με αυτόν τον τρόπο σε μια ομοιόμορφη και ρεαλιστική τρισδιάστατη σκηνή που είναι οπτικά αρεστή και δε της λείπει τίποτα από

την πραγματική εικόνα. Η παραγοντοποίηση του πίνακα συνδιακύμανσης ( $\Sigma$ ) συμβαίνει δηλαδή ως εξής :

$$\Sigma = RSS^T R^T$$

Αυτή η παραγοντοποίηση μπορεί να γίνει κατανοητή ως μια διαμόρφωση ενός ελλειψοειδούς όπου :

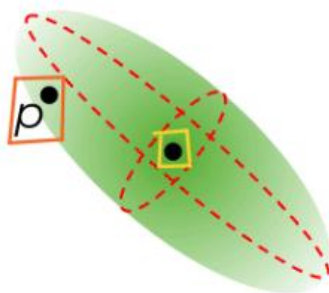
-Το S είναι ένας διαγώνιος πίνακας με 3 παραμέτρους για την κλίμακα.

-Το R είναι ένας πίνακας περιστροφής 3x3 που εκφράζεται αναλυτικά με 4 τετραδόνια (=«quaternions»=αποτελούν τη γενίκευση των δισδιάστατων συνθέτων αριθμών στις τρεις διαστάσεις)

Η ιδιαιτερότητα και βασικά κύριος λόγος της χρήσης των Gaussians βρίσκεται στο διπλό αντίκτυπο κάθε σημείου. Από τη μία πλευρά, κάθε σημείο αντιπροσωπεύει αποτελεσματικά μια περιορισμένη περιοχή στο χώρο κοντά στη μέση τιμή του, σύμφωνα με την συνδιακύμανση του. Από την άλλη πλευρά, έχει θεωρητικά άπειρη έκταση, πράγμα που σημαίνει ότι κάθε Gaussian καθορίζεται σε ολόκληρο τον τρισδιάστατο χώρο και μπορεί να αξιολογηθεί για οποιοδήποτε σημείο.

Η επίδραση μιας τρισδιάστατης Gaussian  $i$  σε ένα αυθαίρετο τρισδιάστατο σημείο  $p$  στον τρισδιάστατο χώρο καθορίζεται ως εξής:

$$f_i(p) = \sigma(\alpha_i) \exp\left(-\frac{1}{2}(p - \mu_i)^T \Sigma_i^{-1} (p - \mu_i)\right)$$



Εικόνα 12: Επίδραση μιας τρισδιάστατης Gaussian  $i$  σε ένα αυθαίρετο τρισδιάστατο σημείο  $p$  στον χώρο 3D<sup>[43]</sup>

Αυτή η εξίσωση μοιάζει αρκετά με μια συνάρτηση πυκνότητας πιθανότητας της κανονικής κατανομής πολλαπλών μεταβλητών , εκτός από τον όρο κανονικοποίησης με την ορίζουσα του πίνακα συνδιακύμανσης που αγνοείται, και αντί αυτού, δίνεται το βάρος στη διαφάνεια.

## 4.5 Μοντέλο σχηματισμού και αναπαράστασης εικόνας στο Gaussian Splatting

Έχοντας λοιπόν ένα σύνολο σημείων στον τρισδιάστατο χώρο είναι αρκετά ενδιαφέρον να δούμε πως μπορούμε να τα χρησιμοποιήσουμε αυτά για την απεικόνιση που θέλουμε να επιτύχουμε. Το μοντέλο σχηματισμού και δημιουργίας εικόνας που ακολουθείται στο Gaussian Splatting είναι παρόμοιο με αυτό που χρησιμοποιείται στο NeRF. Για να το διαπιστώσουμε αυτό αρκεί να μελετήσουμε τον τύπο που δίνεται στο NeRF για την απεικόνιση ενός «όγκου», τον οποίο έπειτα θα μετατρέψουμε και θα επαναδιατυπώσουμε κάνοντας ορισμένες απλές αλλαγές.

$$C(p) = \sum_{i=1}^N c_i (1 - \exp(-\sigma_i \delta_i)) T_i$$

Όπου

- $C(p)$  : το αναμενόμενο χρώμα του σημείου-pixel
- $\sigma_i$  : η πυκνότητα όγκου
- $\delta_i$  : διάκενο (απόσταση μεταξύ διαδοχικών δειγμάτων)
- $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$

Συνεπάγεται λοιπόν χρησιμοποιώντας τη σχέση για το  $T_i$  :

$$\begin{aligned} C(p) &= \sum_{i=1}^N c_i (1 - \exp(-\sigma_i \delta_i)) \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \\ &= \sum_{i=1}^N c_i (1 - \exp(-\sigma_i \delta_i)) \prod_{j=1}^{i-1} \exp(-\sigma_j \delta_j) \end{aligned}$$

Και θέτοντας  $a_i = 1 - \exp(-\sigma_i \delta_i)$  έχουμε :

$$C(p) = \sum_{i=1}^N c_i a_i \prod_{j=1}^{i-1} (1 - a_j)$$

Όπου ο  $\prod_{j=1}^{i-1}(1 - a_j)$  είναι ο όρος που αφορά τη διαφάνεια

Το τελικό χρώμα είναι ένα άθροισμα που εξαρτάται από τα χρώματα των τρισδιάστατων σημείων που παίρνονται ως δείγμα κατά μήκος της ακτίνας και από την διαφάνεια του καθενός από αυτά . Με αυτό υπόψη, ας κοιτάξουμε και ας επεξεργαστούμε επιτέλους το μοντέλο δημιουργίας εικόνας του Gaussian splatting:

$$C(p) = \sum_{i=1}^N c_i f_i^{2D}(p) \prod_{j=1}^{i-1} (1 - f_j^{2D}(p))$$

Όπου ο  $\prod_{j=1}^{i-1}(1 - f_j^{2D}(p))$  είναι ο όρος που αφορά τη διαφάνεια

Όπως παρατηρούμε λοιπόν οι δυο σχέσεις στις οποίες καταλήξαμε είναι σχεδόν ίδιες και η μόνη διαφορά έγκυται στον τρόπο υπολογισμού του  $a$ . Παρ' όλα αυτά αυτή η μικρή αλλαγή και ιδιαιτερότητα είναι εξαιρετικά σημαντική και καθοριστική στη πράξη αφού επιδρά δραστικά και άμεσα στις ταχύτητες αναπαράστασης. Δεν είναι καθόλου υπερβολή να πούμε πως αυτός είναι λόγος για τον οποίο μπορούμε να χρησιμοποιούμε το Gaussian Splatting σε πραγματικό χρόνο.

Για να κατανοήσουμε και να αντιληφθούμε εν τέλει γιατί συμβαίνει όμως αυτό, πρέπει να διερευνήσουμε και να καταλάβουμε τι σημαίνει το  $f^{2D}$  και ποιες είναι οι απαιτήσεις σε υπολογιστική ισχύ που αυτό θέτει. Αυτή η συνάρτηση όμως είναι απλώς η προβολή της  $f(p)$  που είδαμε στην προηγούμενη ενότητα σε 2D, δηλαδή σε ένα επίπεδο της εικόνας που απεικονίζεται. Τόσο ένα 3D σημείο όσο και η προβολή του είναι Gaussians με πολλές μεταβλητές και για αυτό το λόγο ο αντίκτυπος μιας προβεβλημένης 2D Gaussian σε ένα pixel μπορεί να υπολογιστεί χρησιμοποιώντας τον ίδιο τύπο με τον αντίκτυπο μιας 3D Gaussian σε άλλα σημεία στο 3D (δείτε την εικόνα 12). Η μόνη διαφορά είναι ότι η μέση τιμή  $\mu$  (mean) και η συνδιακύμανση  $\Sigma$ (Covariance) πρέπει να προβληθούν σε 2D, κάτι που γίνεται χρησιμοποιώντας παράγωγους από το EWA (elliptical weighted average) splatting.

Οι μέσες τιμές στον 2D μπορούν να αποκτηθούν εύκολα και γρήγορα προβάλλοντας ένα διάνυσμα  $\mu$  σε ομογενείς συντεταγμένες (με επιπλέον συντεταγμένη 1) σε ένα επίπεδο εικόνας χρησιμοποιώντας έναν ενδότερο πίνακα κάμερας  $K$  και έναν εξωτερικό πίνακα κάμερας  $E=[R|t]$ :

$$z \begin{bmatrix} u/z \\ v/z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KE \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \\ 1 \end{bmatrix}$$

$$\mu^{2D} = \begin{bmatrix} \mu_x^{2D} \\ \mu_y^{2D} \end{bmatrix} = \begin{bmatrix} u/z \\ v/z \end{bmatrix}$$

Το οποίο μπορεί να γραφεί και ως εξής :

$$\mu^{2D} = K \left( \frac{E\mu}{(E\mu)_z} \right)$$

Όπου εδώ το «z» στον δείκτη αναφέρεται στην z-κανονικοποίηση. Η συνδιακύμανση στο 2D ορίζεται χρησιμοποιώντας έναν Ιακωβιανό (Jacobian) της παραπάνω σχέσης:

$$J = \frac{\partial \mu^{2D}(\mu)}{\partial \mu}$$

$$\Sigma^{2D} = J \Sigma W^T J^T$$

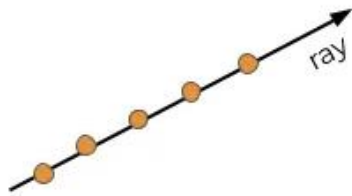
Η όλη διαδικασία συνεχίζει να παραμένει διαφορίσιμη κάτι που είναι απαραίτητο για την μετέπειτα βελτιστοποίηση.

Ο τύπος  $C(p) = \sum_{i=1}^N c_i f_i^{2D}(p) \prod_{j=1}^{i-1} (1 - f_j^{2D}(p))$  όπως αναφέραμε μας δίνει το χρώμα σε ένα συγκεκριμένο και μεμονωμένο pixel. Για να απεικονίσουμε μια ολόκληρη εικόνα, εξακολουθεί όμως να είναι απαραίτητο να διατρέξουμε μέσα από όλες τις ακτίνες  $H \times W$  (ύψος επί πλάτος), όπως και στο NeRF, ωστόσο η διαδικασία είναι πολύ πιο ελαφριά όσον αφορά την απαιτούμενη υπολογιστική ισχύ για αρκετούς λογούς.

Αρχικά για μια δεδομένη κάμερα, το  $f(p)$  κάθε 3D σημείου μπορεί να προβληθεί εκ των προτέρων σε 2D, πριν από την επανάληψη στα pixels. Με αυτόν τον τρόπο, όταν μια Gaussian αναμειγνύεται για λίγα κοντινά pixels, δεν θα χρειαστεί να το προβάλλουμε ξανά και ξανά. Ακόμα, δεν υπάρχει MLP (Multi-layer Perceptron) που θα χρειαστεί να εκτελεστεί  $H \cdot W \cdot P$  (ύψος επί πλάτος επί pixel) φορές για μια μόνο εικόνα, αφού οι 2D Gaussians αναμειγνύονται απευθείας σε μια εικόνα. Επιπλέον, δεν υπάρχει καμία ασάφεια για το ποιο 3D σημείο να αξιολογήσουμε κατά μήκος της ακτίνας, αφού δεν υπάρχει η ανάγκη να επιλέξουμε μια στρατηγική δειγματοληψίας ακτίνας. Ένα σύνολο 3D σημείων επικαλύπτουν την ακτίνα του κάθε pixel, το οποίο σύνολο είναι διακριτό και σταθερό μετά τη βελτιστοποίηση. Ένα στάδιο προ-επεξεργασίας ταξινόμησης γίνεται μία φορά ανά καρέ, σε μια μονάδα επεξεργασίας γραφικών (GPU), χρησιμοποιώντας μια προσαρμοσμένη υλοποίηση διαφοροποιησιμων CUDA kernels .



## NeRF



## Gaussian Splatting



Εικόνα 13 : Η ουσιαστική διαφορά στο concept του NeRF και του Gaussian Splatting<sup>[43]</sup>

Ο σκοπός του αλγόριθμου ταξινόμησης είναι να προετοιμάσει για την απεικόνιση χρώματος με τον τύπο που αναφέραμε παραπάνω αφού ταξινομήσει τα 3D σημεία ανά βάθος (εγγύτητα προς ένα επίπεδο εικόνας) και τα ομαδοποιήσει ανά πλακίδιο. Το πρώτο χρειάζεται για τον υπολογισμό της διαφάνειας, και το δεύτερο επιτρέπει τον περιορισμό του ζυγισμένου αθροίσματος για κάθε pixel στην ανάμιξη των τιμών  $\alpha$  (της διαφάνειας) μόνο των σχετικών 3D σημείων (ή των 2D προβολών τους, για να είμαστε πιο συγκεκριμένοι). Η ομαδοποίηση επιτυγχάνεται χρησιμοποιώντας απλά πλακίδια με 16x16 pixel και εφαρμόζεται έτσι ώστε μια Gaussian να μπορεί να «προσγειωθεί» σε λίγα πλακίδια αν επικαλύπτει περισσότερο από ένα μόνο πεδίο θέασης. Χάρη στην ταξινόμηση, η απεικόνιση του κάθε pixel μπορεί να μειωθεί στην ανάμιξη των τιμών  $\alpha$  (της διαφάνειας) προ-καθορισμένων σημείων από το πλακίδιο στο οποίο ανήκει το pixel.



Εικόνα 14: Αναπαριστώντας το πεδίο θέασης με Gaussians<sup>[29]</sup>

## 4.6 Βελτιστοποίηση στο Gaussian Splatting

Είναι αρκετά ενδιαφέρον το πως μπορούμε από ελλειψοειδή στο χώρο να παίρνουμε τελικά μια αξιόλογη και ωραία αισθητικά εικόνα χωρίς προβλήματα. Η αλήθεια μάλιστα είναι πως αν δεν υπάρξει η σωστή, μεθοδική και κατάλληλη επεξεργασία των Gaussians θα έχουμε πολλών ειδών θέματα στην αναπαράσταση όπως το λεγόμενο blur, δηλαδή η ύπαρξη σημείων στο χώρο που είναι θολά αλλά ακόμα και το να βλέπουμε ορισμένες ατέλειες οι οποίες μοιάζουν πολύ με τα ελλειψοειδή που παράγουμε (όπως φαίνεται και στην παρακάτω εικόνα). Ο μονός και σωστός τρόπος για να πάρουμε καλές και ρεαλιστικές αναπαραστάσεις είναι η ύπαρξη καλής αρχικοποίησης, διαφορίσιμης βελτιστοποίησης και προσαρμοστικής πυκνοποίησης.



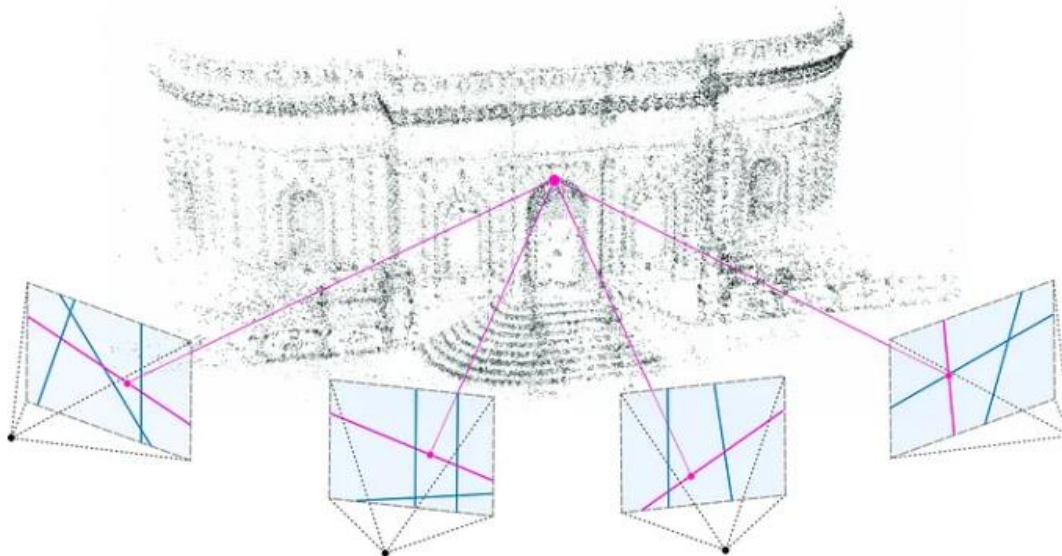
Εικόνα 15: Το αποτέλεσμα μιας αναπαράστασης που δεν έχει βελτιστοποιηθεί σωστά<sup>[43]</sup>

Η διαδικασία βελτιστοποίησης περιλαμβάνει κατά κύριο λόγο τη δημιουργία ενός πυκνού συνόλου από 3D Gaussians για να αντιπροσωπεύσει τελικά ρεαλιστικά και πιστά μια σκηνή όπου θα υπάρξει η λεγομένη σύνθεση ελεύθερης προβολής.

Για τη διαχείριση των ασαφειών που θα προκύψουν από την προβολή από το 3D στο 2D, η μέθοδος βελτιστοποίησης δημιουργεί δυναμικά, τροποποιεί και αλλάζει αλλά και πολλές φορές αφαιρεί στοιχεία γεωμετρίας. Οι ακριβείς παράμετροι της συνδιακύμανσης είναι απαραίτητες και καταλυτικές για μια συμπαγή αναπαράσταση, ιδίως όταν μας απασχολεί η αποτύπωση μεγάλων, ομοιόμορφων περιοχών όπου υπάρχει ο ελάχιστος αριθμός ανισοτροπικών Gaussians. Έτσι λοιπόν καταλήγουμε στα 3 βασικά βήματα που συμπεριλαμβάνονται στη σωστή βελτιστοποίηση όπως αυτά αναφέρθηκαν παραπάνω.



Η αρχικοποίηση αναφέρεται στις παραμέτρους του συνόλου σημείων 3D στην αρχή της εκπαίδευσης του μοντέλου. Για τις τοποθεσίες των σημείων (δηλαδή για τις μέσες τιμές), προτείνεται η χρήση ενός συνόλου σημείων που παράγεται από το Structure from Motion (SfM) (όπως φαίνεται στην Εικόνα 16). Η λογική είναι ότι για οποιαδήποτε ανακατασκευή και αναπαράσταση 3D, είτε με Gaussian Splatting, NeRF ή κάτι πιο κλασικό, πρέπει να γνωρίζουμε τις μήτρες κάμερας, οπότε θα πρέπει να τρέξουμε SfM για να τις αποκτήσουμε. Αφού λοιπόν το SfM παράγει ένα αραιό σύνολο σημείων αυτό επιλεγούμε να το χρησιμοποιήσουμε για αρχικοποίηση. Όταν ένα σύνολο σημείων δεν είναι διαθέσιμο για οποιοδήποτε λόγο, μπορεί να χρησιμοποιηθεί αντ' αυτού μια τυχαία αρχικοποίηση, με τον κίνδυνο μιας πιθανής απώλειας της τελικής ποιότητας της ανακατασκευής και αναπαράστασης της σκηνής. Οι πίνακες συνδιακύμανσης αρχικοποιούνται ως ιστροπικοί, με άλλα λόγια, τα σημεία 3D ξεκινούν ως σφαίρες. Οι ακτίνες ορίζονται με βάση τις μέσες αποστάσεις προς τα γειτονικά σημεία, έτσι ώστε ο τρισδιάστατος κόσμος να καλύπτεται ομαλά και να μην έχει "κενά" και άρα η αναπαράσταση να είναι ομαλή, συμπαγής και ομοιόμορφη.

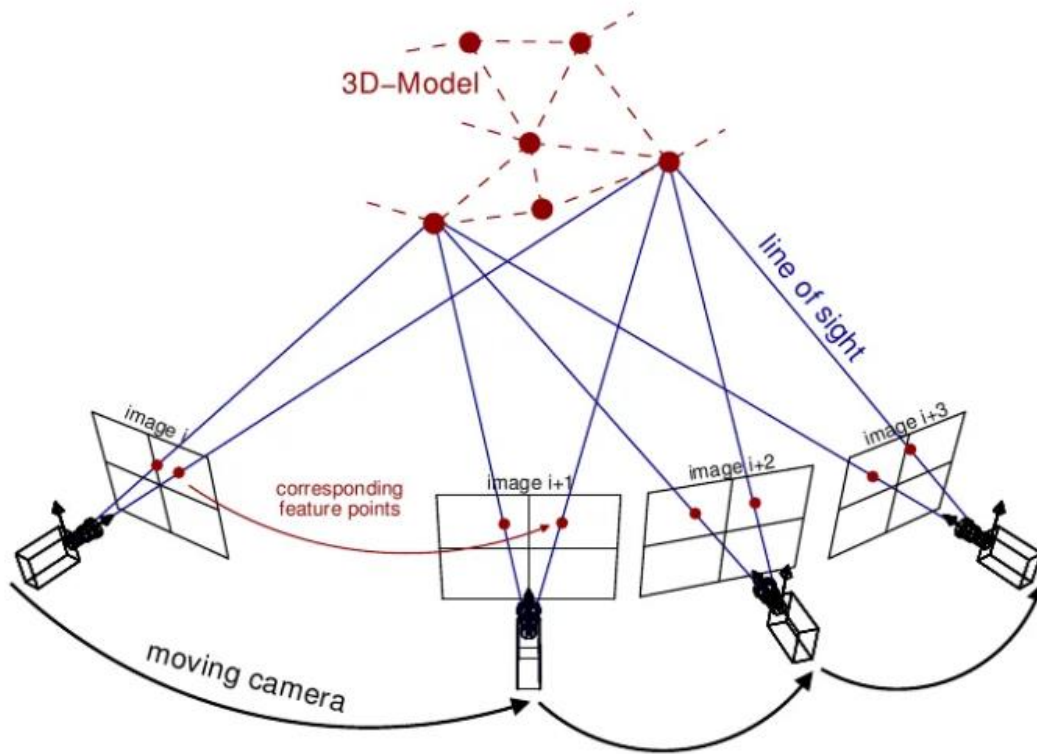


Εικόνα 16: Αραιό 3d point cloud που παράγεται με SfM αρχικοποίηση<sup>[30]</sup>

Το Structure from Motion (SfM) είναι μια τεχνική υπολογιστικής όρασης που ανακατασκευάζει και απεικονίζει μια τρισδιάστατη σκηνή από ένα σύνολο δισδιάστατων εικόνων ή καρέ βίντεο. Η διαδικασία περιλαμβάνει την εκτίμηση της κίνησης της κάμερας και την ανακατασκευή και τελική απεικόνιση της τρισδιάστατης δομής.

Για να το επιτύχει αυτό, πρώτα αναγνωρίζει και παρακολουθεί διακριτά χαρακτηριστικά σε διάφορες εικόνες, προσδιορίζοντας πώς κινούνται μεταξύ των καρέ. Χρησιμοποιώντας αυτές τις πληροφορίες, το SfM εκτιμά στη συνέχεια τις σχετικές θέσεις

των καμερών κατά τη διάρκεια του χρόνου. Στη συνέχεια, η τεχνική αυτή τριγωνομετρεί(με την έννοια του προσδιορισμού) τις τρισδιάστατες θέσεις αυτών των χαρακτηριστικών βρίσκοντας που οι αντίστοιχα ακτίνες από τις διάφορες θέσεις της κάμερας διασταυρώνονται και ταυτίζονται. Τέλος, μια διαδικασία βελτιστοποίησης, που ουσιαστικά κάνει μικροδιορθώσεις , γνωστή ως bundle adjustment, βελτιστοποιεί τις θέσεις των καμερών και τα τρισδιάστατα σημεία για να ελαχιστοποιήσει τις αποκλίσεις μεταξύ των πραγματικών και προβλεπόμενων θέσεων.



Εικόνα 17: Τρόπος λειτουργίας του Structure from Motion<sup>[30]</sup>

Συνεχίζοντας μετά την αρχικοποίηση, οι ακριβείς παράμετροι συνδιακύμανσης όπως προαναφέραμε είναι κρίσιμες για μια συμπαγή αναπαράσταση, ιδίως στην αποτύπωση μεγάλων, ομοιόμορφων περιοχών ,όταν διατίθεται δηλαδή ένας ελάχιστος αριθμός ανισοτροπικών Gaussians. Για αυτό το λόγο , χρησιμοποιείται το «stochastic gradient descent» για να προσαρμοστούν σωστά στη σκηνή , εκτιμώντας με αυτόν τον τρόπο τις παραμέτρους των Gaussians. Οι ενημερώσεις των θέσεων ακολουθούν εκθετική υποβάθμιση, καθοδηγούμενες από μια συνάρτηση απώλειας που συνδυάζει τους όρους L1 και D-SSIM (Structural Dissimilarity Index terms) ανάμεσα στην πραγματικότητα και την τρέχουσα αναπαράσταση.

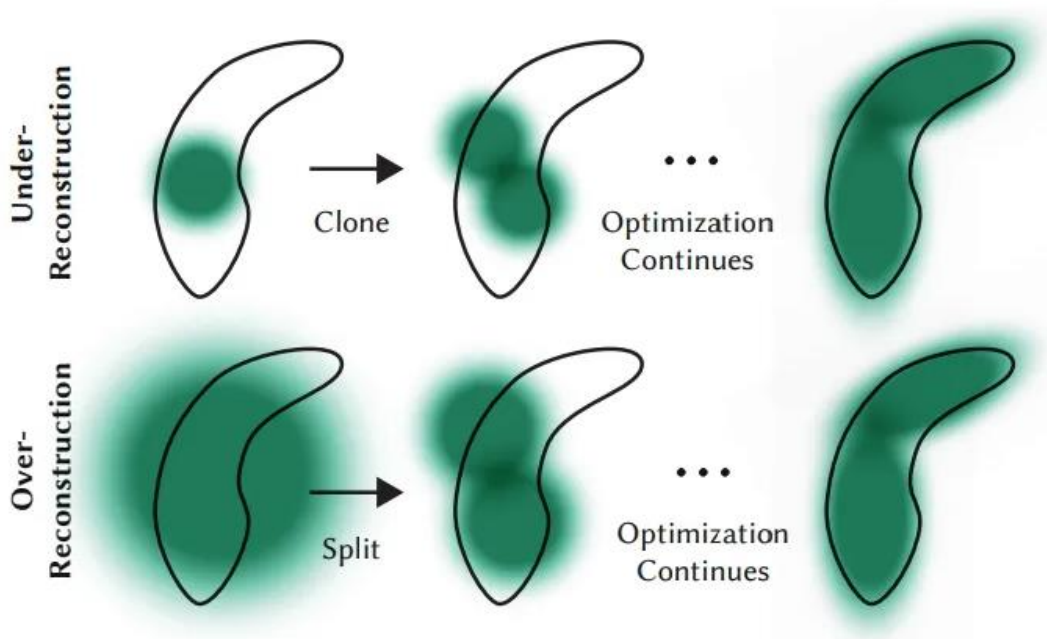
$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM}$$

Το Stochastic Gradient Descent (SGD) είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται συχνά στη μηχανική μάθηση και στο λεγόμενο deep learning για την εκπαίδευση μοντέλων. Πρόκειται για μια παραλλαγή του παραδοσιακού αλγορίθμου gradient descent και είναι ιδιαίτερα κατάλληλος για μεγάλα σύνολα δεδομένων. Στο κανονικό gradient descent, οι παράμετροι του μοντέλου ενημερώνονται βάσει της μέσης κλίσης της συνάρτησης απώλειας που υπολογίζεται για ολόκληρο το σύνολο δεδομένων. Αντίθετα, στο SGD, αντί να χρησιμοποιείται ολόκληρο το σύνολο δεδομένων, οι παράμετροι ενημερώνονται για κάθε μεμονωμένο σημείο δεδομένων ή για ένα μικρό σύνολο σημείων που επιλέγονται τυχαία από το σύνολο δεδομένων. Αυτό εισάγει τυχαιότητα στη διαδικασία βελτιστοποίησης. Τα πλεονεκτήματα του αφορούν την αποτελεσματικότητα αφού είναι πιο αποδοτικό καθώς επεξεργάζεται ένα υποσύνολο δεδομένων κάθε φορά καθώς και στη γενίκευση που κάνει αφού μπορεί να βοηθήσει το μοντέλο να μην κολλήσει σε ορισμένα τοπικά ελάχιστα και έτσι να δημιουργεί πρόβλημα.

Ας επανέλθουμε λοιπόν στα βήματα της σωστής βελτιστοποίησης αφού ήρθε η ώρα να φτάσουμε στο τρίτο και τελευταίο στάδιο την προσαρμοστική πυκνοποίηση. Μετά τη δημιουργία ενός αρχικού συνόλου αραιών σημείων χρησιμοποιώντας το Structure from Motion (SfM), η προσέγγιση της προσαρμοστικής πυκνοποίησης προσαρμόζει δυναμικά τον αριθμό και την πυκνότητα των 3D Gaussians για να βελτιώσει την αναπαράσταση της σκηνής κατά τη σύνθεση ελεύθερης προβολής.

Μετά από έναν αρχικό στάδιο βελτιστοποίησης, πραγματοποιείται πυκνοποίηση ανά κάποιο χρονικό διάστημα ας πούμε κάθε 100 επαναλήψεις, ούτως ώστε να αντιμετωπίσει τόσο τις περιοχές υπο-ανακατασκευής όσο και τις περιοχές υπερ-ανακατασκευής με μεγάλη κάλυψη από Gaussians (Εικόνα 18) ,εκμεταλλευόμενο τις κλίσεις των θέσεων στο χώρο προβολής.

Με την πυκνοποίηση ουσιαστικά κλωνοποιούνται οι μικρές Gaussians σε περιοχές υπο-ανακατασκευής και διαχωρίζονται οι μεγάλες Gaussians σε περιοχές υψηλής διακύμανσης. Για να ισορροπήσει τον συνολικό όγκο του συστήματος και τον αριθμό των Gaussians, οι τιμές  $\alpha$  επαναφέρονται περιοδικά, επιτρέποντας ελεγχόμενη αύξηση ή μείωση. Αυτή η έξυπνη και στρατηγική προσέγγιση, σε συνδυασμό με την περιοδική αφαίρεση υπερβολικά μεγάλων Gaussians, διατηρεί τον έλεγχο στον συνολικό αριθμό των Gaussians χωρίς την ανάγκη συμπίεσης ή παραμόρφωσης του χώρου. Αυτή η μέθοδος εξασφαλίζει ότι οι Gaussians διατηρούν την αναπαράστασή τους στον Ευκλείδιο χώρο κατά τη διαδικασία βελτιστοποίησης.



Εικόνα 18: Under and over reconstruction με κάλυψη από Gaussians<sup>[1]</sup>

## 4.7 Spherical Harmonics (SH)

Οι Spherical Harmonics (SH) είναι μαθηματικές συναρτήσεις που ορίζονται στην επιφάνεια μιας σφαίρας. Στα γραφικά των υπολογιστών, χρησιμοποιούνται συνήθως για την αναπαράσταση πληροφοριών φωτισμού με τρόπο συμπαγή, αποτελεσματικό και ευκρινή. Οι σφαιρικές αρμονικές συναρτήσεις χρησιμοποιούνται για την προσέγγιση πολύπλοκων συνθηκών φωτισμού σε μια σφαιρική περιοχή, επιτρέποντας ρεαλιστική απεικόνιση σε εικόνες που παράγονται στον υπολογιστή.

Πιο απλά, οι σφαιρικές αρμονικές αποσυνθέτουν το εισερχόμενο φως σε ένα σύνολο συντελεστών, κάθε ένας από τους οποίους συσχετίζεται με μια συγκεκριμένη SH συνάρτηση. Αυτοί οι συντελεστές αποτυπώνουν τα χαρακτηριστικά του φωτισμού, όπως η ένταση και το χρώμα του, σε διάφορες κατευθύνσεις σε μια σφαιρική επιφάνεια. Χρησιμοποιώντας ένα περιορισμένο αριθμό συντελεστών σφαιρικών αρμονικών, η αναπαράσταση παραμένει συνοπτική ενώ παρέχει μια αξιόλογη και σαφή προσέγγιση του περιβάλλοντος φωτισμού.

Στα γραφικά των υπολογιστών, οι σφαιρικές αρμονικές συναρτήσεις χρησιμοποιούνται συχνά σε τεχνικές όπως η προϋπολογισμένη μεταφορά ακτινοβολίας

(PRT) και η διάχυση του φωτός με τον περιμετρικό φωτισμό για να προσομοιάσουν ρεαλιστικά εφέ φωτισμού σε εικονικά σκηνικά και περιβάλλοντα.

Η σχέση είναι η εξής :

$$Y_l^m(\theta, \varphi) = (-1)^m \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{im\varphi}$$

Όπου:

- $l$  είναι ένας μη αρνητικός ακέραιος γνωστός ως βαθμός της σφαιρικής αρμονικής
- $m$  είναι ένας ακέραιος για τον οποίον ισχύει  $-\ell \leq m \leq \ell$  και είναι γνωστός ως τάξη
- $\theta$  είναι η πολική γωνία
- $\varphi$  είναι η αζιμουθιακή γωνία
- $P_l^m(\cos\theta)$  είναι το σχετικό πολώνυμο Legendre

l:		$P_l^m(\cos\theta) \cos(m\varphi)$	$P_l^{ m }(\cos\theta) \sin( m \varphi)$
0	s		
1	p		
2	d		
3	f		
4	g		
5	h		
6	i		
	m:	6 5 4 3 2 1 0	-1 -2 -3 -4 -5 -6

Εικόνα 19: Σφαιρικές αρμονικές συναρτήσεις για διαφορετικές τιμές των  $l$  και  $m$ <sup>[16]</sup>

## 4.8 Περιορισμοί του Gaussian Splatting

Παρά τα συνολικά εξαιρετικά αποτελέσματα, την εντυπωσιακή ταχύτητα απεικόνισης και τη γενικότερη τεράστια βελτίωση σε σχέση με τις άλλες μεθόδους πεδίου ακτινοβολίας, η απλότητα της αναπαράστασης με τη μέθοδο αυτή έρχεται με κόστος και οδηγεί σε περιορισμούς. Ο σημαντικότερος λόγος που οδηγούμαστε σε προβλήματα είναι οι διάφορες κανονικοποιήσεις που εισάγονται κατά την διάρκεια της βελτιστοποίησης για να προστατεύσουν το μοντέλο από "σπασμένες" Gaussians: σημεία που είναι πολύ μεγάλα, πολύ μακριά, περιττά, κλπ. Είναι πιθανό να προκύψουν «θολές» ή παραμορφωμένες Gaussians σε σκηνές με πολλές ιδιαιτερότητες και λεπτομέρειες ή σε πολύ μεγάλες σε μέγεθος. Ακόμα μπορεί οι Gaussians να φαίνονται σαν αν έχουν εκραγεί σαν να έχουν σκάσει σε ορισμένα περίεργα σημεία και ειδικά σε περιοχές όπου πολλές όψεις εξαρτώνται η μία από την άλλη.

Η επιλογή του Gaussian Splatting να εγκαταλείψει την συνεχή αναπαράσταση προς όφελος μιας διακριτής σημαίνει ότι το επαγωγικό συστηματικό σφάλμα των MLP (Multi-layer Perceptron) χάνεται. Στα NeRFs, ένα MLP εκτελεί μια έμμεση παρεμβολή και εξομαλύνει πιθανές αντιφάσεις μεταξύ των δεδομένων προβολών, ενώ οι 3D Gaussians είναι πιο ευαίσθητες, επαναφέροντας στο πρόβλημα που περιγράφεται παραπάνω.

Επιπλέον, η γκαουσιανή προβολή δεν είναι απαλλαγμένη από ορισμένα γνωστά σφάλματα που παρουσιάζουν και τα NeRFs, τα οποία και τα δύο κληρονομούν από το κοινό μοντέλο δημιουργίας εικόνας το οποίο αποδείξαμε προηγουμένως. Αυτό το κοινό μοντέλο οδηγεί σε χαμηλότερη ποιότητα στις λιγότερο εμφανείς ή αόρατες περιοχές, και δημιουργεί "floaters" κοντά σε ένα επίπεδο εικόνας τα οποία δε μπορούν να αφαιρεθούν κατά την βελτιστοποίηση. Επίσης, το Anti-Aliasing (δηλαδή κάποια τεχνική για να μειωθούν οι ακανόνιστες άκρες στις εικόνες που εμφανίζονται συχνά σε περιοχές υψηλής αντίθεσης μεταξύ των pixels ιδιαίτερα όταν υπάρχουν έντονες αλλαγές σε χρώμα και ένταση) δεν έχει υλοποιηθεί, αφήνοντας χώρο για βελτιώσεις όσον αφορά τις απότομες αλλαγές βάθους και σειράς ανάμειξης.

Τέλος, αξίζει να επισημανθεί πως η κατανάλωση μνήμης της κάρτας γραφικών, αν και είναι πιο συμμαζεμένη από τις προηγούμενες μεθόδους δε παύει να ξεπερνά τα 20 GB όταν εκπαιδεύεται σε μεγάλες και πολύπλοκες σκηνές. Σε αυτόν τον τομέα λοιπόν υπάρχουν πολλές προοπτικές εξέλιξης για τη μείωση της χρήσης της μνήμης ούτως ώστε να μπορεί να γίνει αυτή η δουλειά και σε υπολογιστές που δεν έχουν κάρτες γραφικών τελευταίας τεχνολογίας.



## 5. Gaussian splatting στην πράξη

Πριν μπούμε στον κώδικα αυτόν καθ' αυτόν ας κάνουμε μια σύνοψη για το πως και γιατί καταλήξαμε στη μέθοδο Gaussian Splatting για τρισδιάστατη αναπαράσταση. Η γενικότερη ενασχόληση και ανάπτυξη ,όπως προαναφέρθηκε, γύρω από τη σύνθεση προοπτικών προβολών, κυρίως με μεθόδους «Πεδίων Ακτινοβολίας» , έχουν προκαλέσει σημαντικές αλλαγές στον τρόπο με τον οποίο επεξεργάζονται σκηνές που καταγράφονται από πολλαπλές φωτογραφίες ή βίντεο. Ωστόσο, παρά την πρόοδο σε αυτό τον τομέα, η επίτευξη εξαιρετικής ποιοτικής οπτικής είναι ακόμα ακριβή από άποψη εκπαίδευσης και αναπαραγωγής. Επιπλέον, ενώ ορισμένες πρόσφατες μέθοδοι προσφέρουν μεγαλύτερες ταχύτητες επεξεργασίας, συχνά το κάνουν κατά αντάλλαγμα με την ποιότητα αναπαραγωγής. Αυτός ο συμβιβασμός είναι εμφανής ιδιαίτερα όταν πρόκειται για εκτενείς και περίπλοκες σκηνές σε ανάλυση 1080p, όπου καμία υπάρχουσα μέθοδος δεν μπορεί να παρέχει πραγματικού χρόνου ρυθμούς εμφάνισης. Για να αντιμετωπιστούν αυτές οι προκλήσεις, προτάθηκε μια νέα προσέγγιση που αποτελείται από τρία βασικά στοιχεία που όχι μόνο επιτυγχάνουν ασύγκριτη οπτική πιστότητα αλλά και διατηρούν ανταγωνιστικούς χρόνους εκπαίδευσης. Το καινοτόμο της πρότασης αυτής λοιπόν είναι ότι η μέθοδός μας επιτρέπει τη σύνθεση προοπτικών προβολών σε πραγματικό χρόνο υψηλής ποιότητας, επιτυγχάνοντας ρυθμούς καρέ της τάξης των 30 καρέ ανά δευτερόλεπτο ή περισσότερο σε ανάλυση 1080p. Καταρχάς, εκμεταλλευόμαστε τα λίγα σημεία που παράγονται κατά τη διάρκεια της βαθμονόμησης της κάμερας για να καθορίσουμε μια αναπαράσταση της σκηνής χρησιμοποιώντας 3D Gaussians , επιτρέποντας έτσι τη βελτιστοποίηση της σκηνής και αποφεύγοντας αποτελεσματικά τις περιττές υπολογιστικές διεργασίες σε κενούς χώρους. Ταυτόχρονα , εισάγουμε αλληλεπιδραστικούς μηχανισμούς βελτιστοποίησης και έλεγχο πυκνότητας για τις 3D Gaussians. Αυτή η διαδικασία διόρθωσης εξασφαλίζει μια πιστή αναπαράσταση ενώ ελαχιστοποιεί τον υπολογιστικό φόρτο. Η προσέγγισή αυτή επιδεικνύει ασύγκριτη οπτική ποιότητα και δυνατότητες αναπαραγωγής σε πραγματικό χρόνο, σημειώνοντας μια σημαντική πρόοδο στον τομέα της σύνθεσης νέων προοπτικών προβολών.

Ο κώδικας ο οποίος θα χρησιμοποιήσουμε για να δούμε επιτέλους πως λειτουργεί και τι αποτελέσματα παράγει η μέθοδος Gaussian Splatting για τρισδιάστατη αναπαράσταση αναπτύχθηκε από το «Inria» και το «Max Planck Institut for Informatik (MPII)»<sup>[1]</sup>

## 5.1 Τρισδιάστατη αναπαράσταση με τη μέθοδο Gaussian Splatting

Ο κώδικας καθώς και ο αναλυτικός οδηγός εγκατάστασης και χρήσης για το πρόγραμμα υπάρχει στο [GitHub](#) καθώς και βίντεο με αναλυτικές οδηγίες. Πριν προχωρήσουμε στην μελέτη των αποτελεσμάτων ας ασχοληθούμε λίγο συνοπτικά με τα βήματα που πρέπει να ακολουθήσουμε για να τρέξουμε τον κώδικα. Αρχικά θα πρέπει να κατεβάσουμε από το [GitHub](#) τα απαραίτητα αρχεία γνωρίζοντας ότι ο κώδικας βασίζεται σε 4 σημαντικά μέρη:

-Έναν βελτιστοποιητή βασισμένο σε PyTorch για την δημιουργία και ανάπτυξη ενός μοντέλου 3D Gaussian από δεδομένα εισόδου SfM (Structure from Motion). Ο βελτιστοποιητής αυτός έχει συγκεκριμένες και υψηλές απαιτήσεις.

-Ένα πρόγραμμα προβολής δικτύου που επιτρέπει τη σύνδεση και την οπτικοποίηση της διαδικασίας βελτιστοποίησης.

-Ένα πρόγραμμα προβολής πραγματικού χρόνου βασισμένο σε OpenGL για την απεικόνιση των εκπαιδευμένων μοντέλων σε πραγματικό χρόνο.

-Ένα κώδικα που μας βοηθά να μετατρέψουμε τις δικές μας εικόνες σε σύνολα δεδομένων SfM έτοιμα για βελτιστοποίηση χρησιμοποιώντας το COLMAP (θα γίνει εκτενής αναφορά αργότερα)

Εμείς τώρα θα δούμε ορισμένα από τα έτοιμα παραδείγματα καθώς και πως πολύ ευκολά δημιουργούμε, μόνο με φωτογραφίες ή βίντεο, τη βάση δεδομένων για να γίνει τελικά η τρισδιάστατη αναπαράσταση με Gaussian Splatting.

### 5.1.1 Έτοιμα μοντέλα για τρισδιάστατη αναπαράσταση

Στα αρχεία του κώδικα με τον οποίο πραγματοποιούμε τη τρισδιάστατη αναπαράσταση, υπάρχουν έτοιμα μοντέλα-παραδείγματα από χώρους από τους οποίους έχουν ληφθεί οι κατάλληλες φωτογραφίες από διάφορες και πολλές οπτικές γωνίες και έχουν επεξεργαστεί καταλληλά (θα αναφερθούμε αργότερα σε αυτό) ούτως ώστε να μπορεί να εκπαιδευτεί το μοντέλο και να γίνει η τρισδιάστατη αναπαράσταση της σκηνής με Gaussian Splatting. Τώρα λοιπόν θα παρουσιάσουμε δυο τέτοια παραδείγματα, το πρώτο είναι μια σκηνή από ένα πάρκο με ένα ποδήλατο να βρίσκεται σε ένα παγκάκι και το δεύτερο μια σκηνή από μια κουζίνα με ένα πάγκο γεμάτο υλικά για μαγειρική. Τα δεδομένα εισόδου για αυτά τα παραδείγματα είναι οι φωτογραφίες του χώρου που θα χρησιμοποιηθούν καθώς και τρία binary files που αφορούν τις κάμερες, τις φωτογραφίες και τα σημεία στον τρισδιάστατο χώρο. Για να εκπαιδύσουμε το μοντέλο είτε μπορούμε να εκτελέσουμε την εντολή στο command prompt «python train.py -s <path to dataset>» είτε να τρέξουμε τον κώδικα μέσω του Visual Studio Code δημιουργώντας το κατάλληλο launch.json για το που βρίσκονται τα δεδομένα. Σε κάθε περίπτωση θα πρέπει να έχουμε ενεργοποιήσει το περιβάλλον Conda για Gaussian Splatting. Το Conda είναι ένα σύστημα

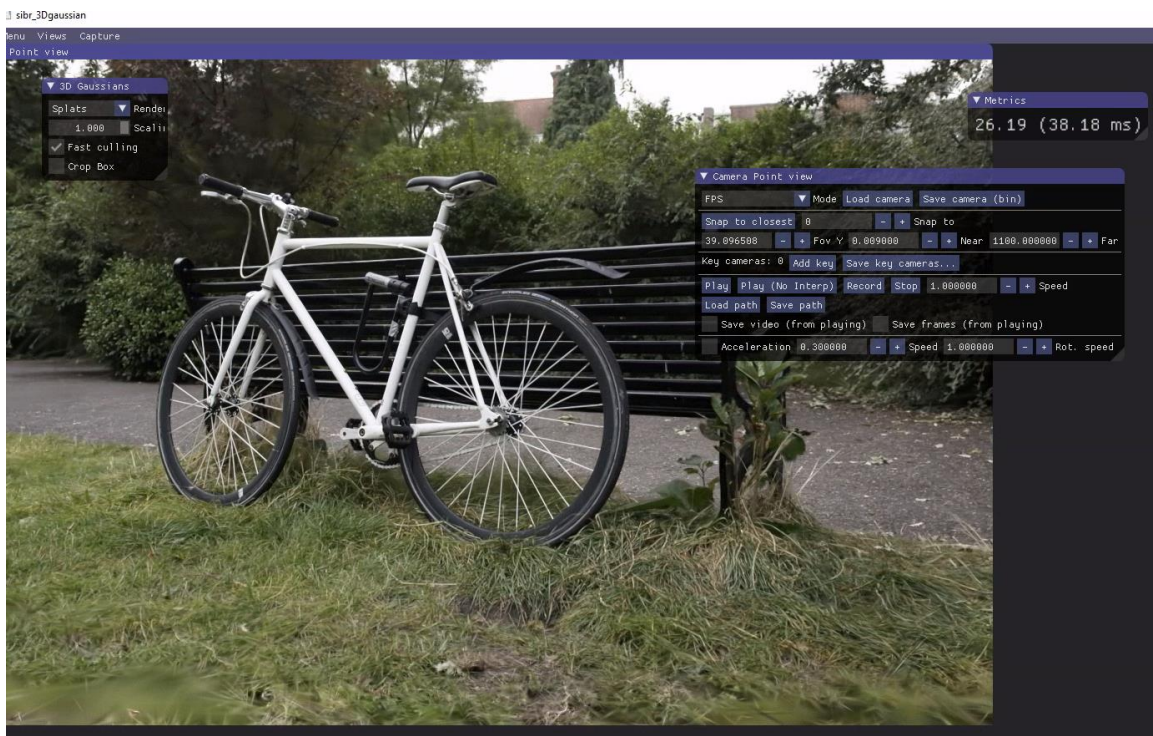
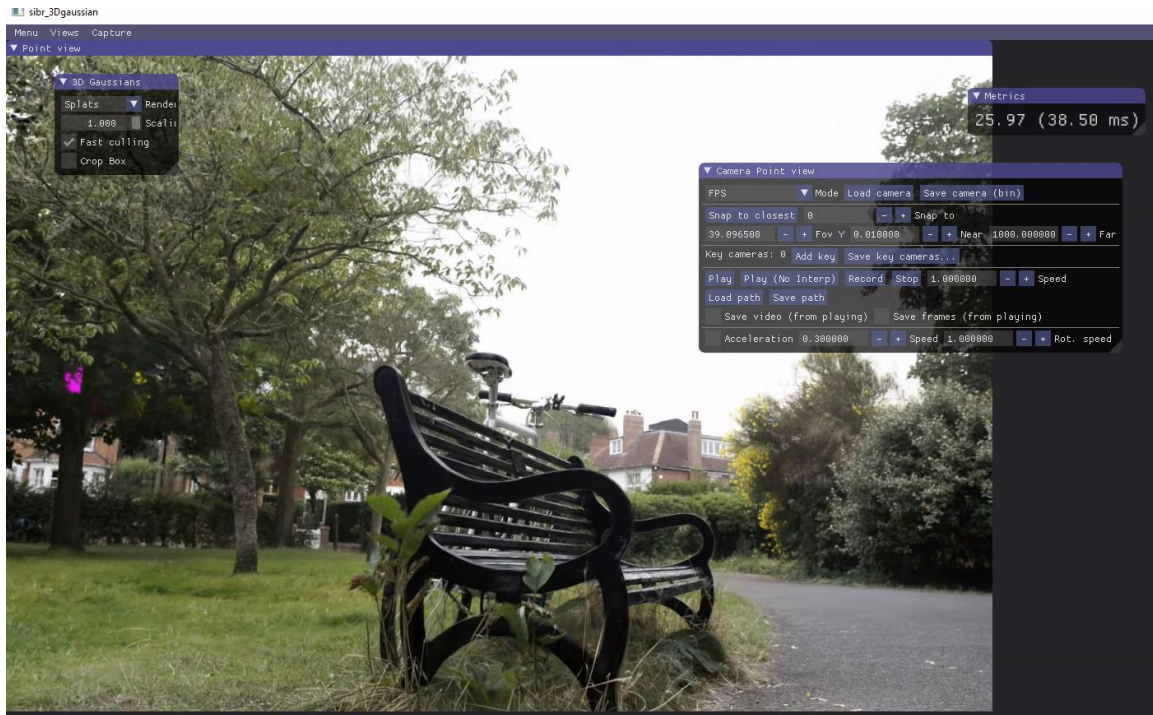


διαχείρισης πακέτων και περιβάλλοντος που μας επιτρέπει να δημιουργούμε, να διαχειριζόμαστε και να μεταβαίνουμε ανάμεσα σε διαφορετικά περιβάλλοντα λογισμικού στον υπολογιστή μας. Με αυτόν τον τρόπο επιτρέπουμε να χρησιμοποιούνται τα πακέτα και οι εξαρτήσεις που έχουν εγκατασταθεί εντός αυτού του περιβάλλοντος.

Το πρόγραμμα προβολής βασίζεται στο πλαίσιο SIBR (System for Image-Based Rendering), που αναπτύχθηκε από τον ομάδα GRAPHDECO για αρκετά έργα σύνθεσης νέων προοπτικών προβολών. Το περιβάλλον του SIBR παρέχει αρκετές μεθόδους πλοήγησης στη σκηνή. Από προεπιλογή, θα ξεκινήσουμε με έναν πλοηγό FPS, τον οποίο μπορούμε να ελέγξουμε με τα πλήκτρα W, A, S, D, Q, E για το διάλυμα μετάφρασης της κάμερας και I, K, J, L, U, O για περιστροφή. Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε έναν πλοηγό τύπου TrackBall (επιλέξιμο από το αιωρούμενο μενού). Μπορούμε επίσης να ευθυγραμμιστούμε σε μια κάμερα από το σύνολο δεδομένων με το κουμπί Snap to ή να βρούμε την πλησιέστερη κάμερα με το Snap to closest. Τα αιωρούμενα μενού μας επιτρέπουν επίσης να αλλάξουμε την ταχύτητα πλοήγησης. Μπορούμε ακόμα να χρησιμοποιήσουμε τον Πολλαπλασιαστή Κλιμάκωσης για να ελέγξουμε το μέγεθος των εμφανιζόμενων Gaussians ή να εμφανίσουμε τον αρχικό πυκνό point-cloud.

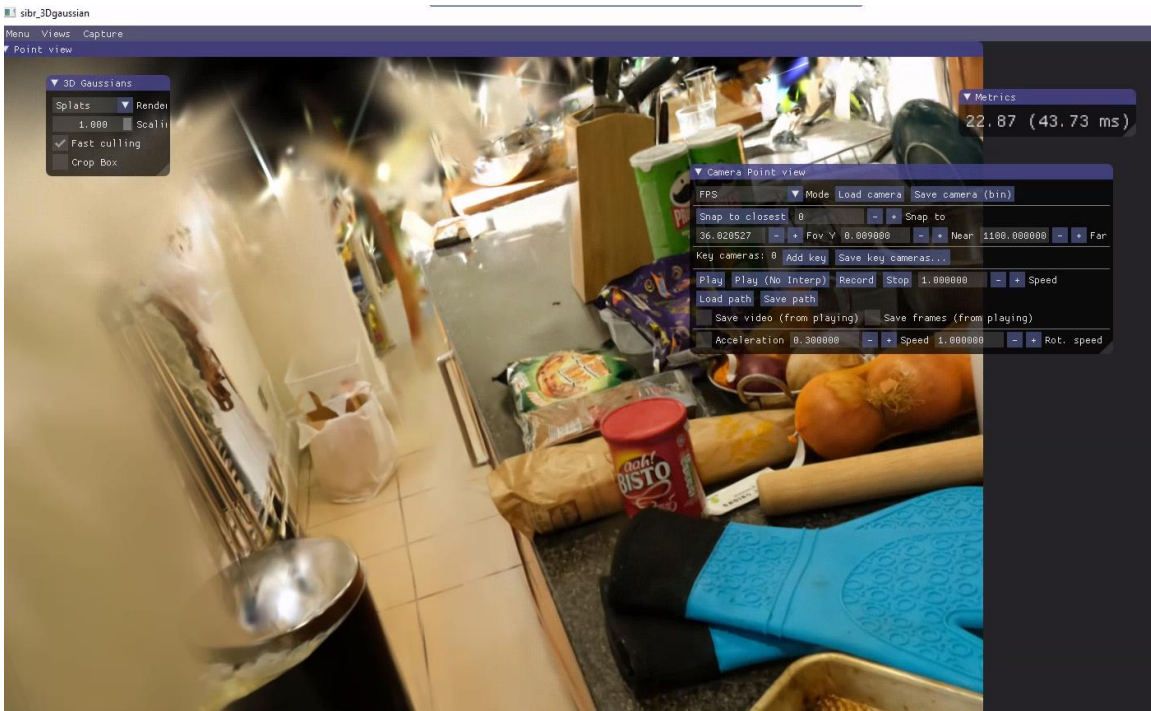
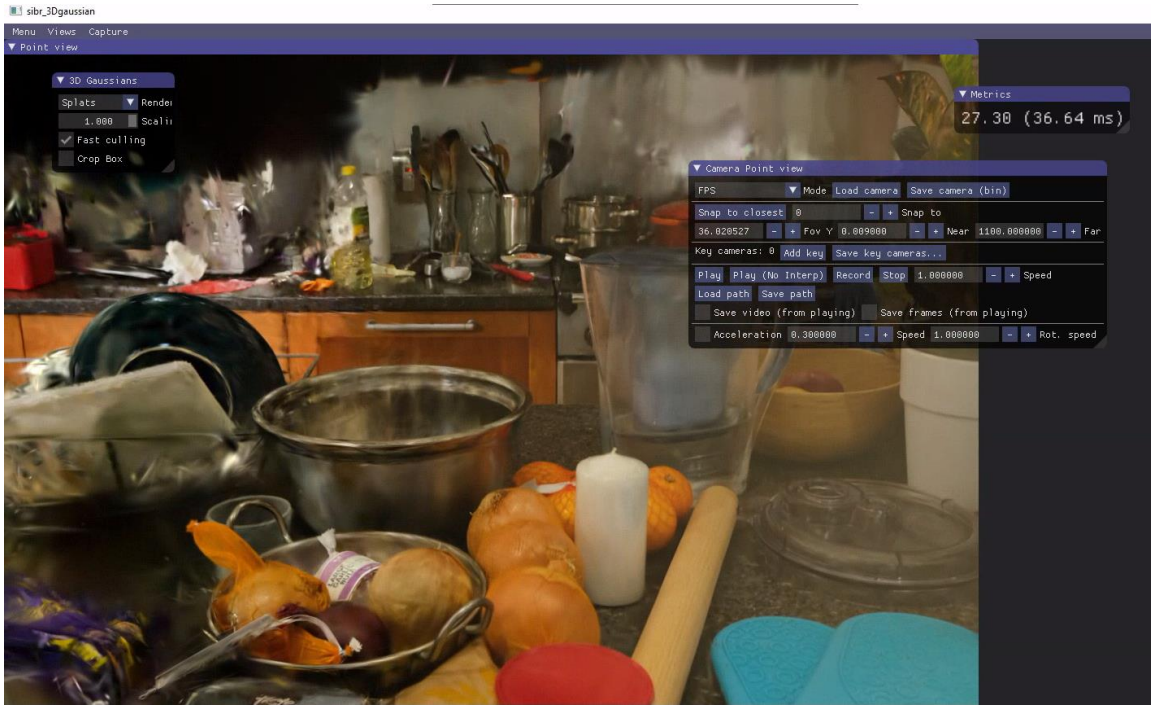
Ας δούμε λοιπόν τα παραδείγματα :

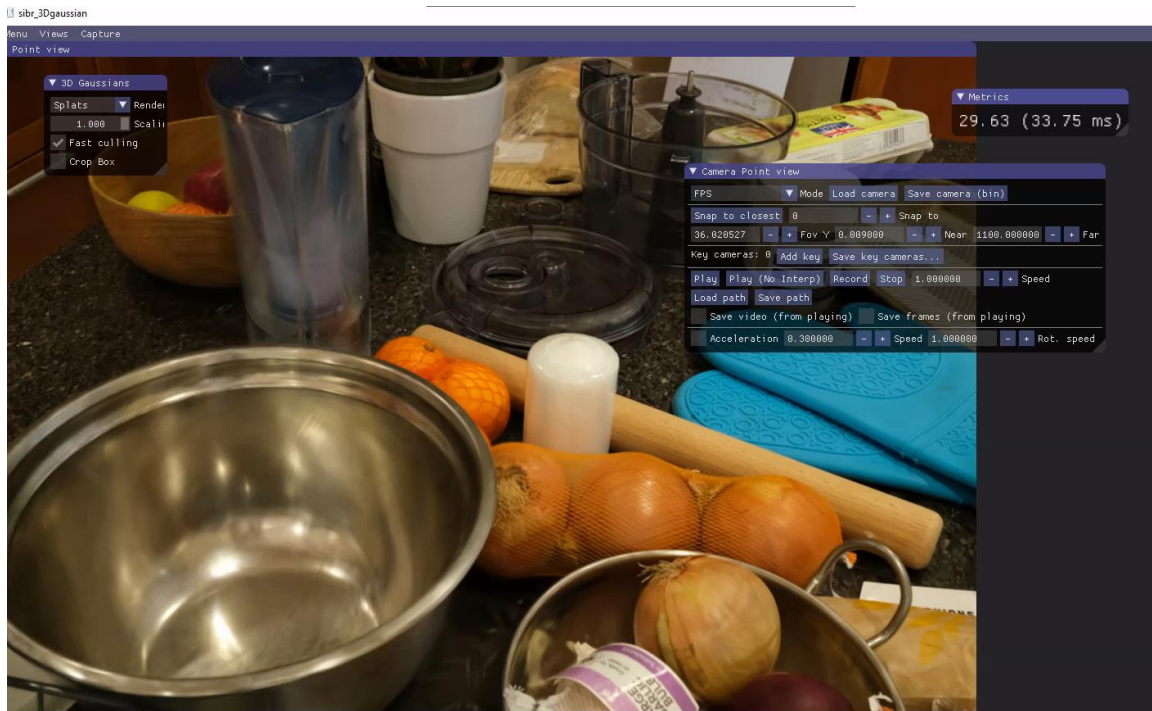




Εικόνες 20α,β,γ:Αποτελέσματα τρισδιάστατης αναπαράστασης ποδήλατου σε πάρκο







Εικόνες 21α,β,γ:Αποτελέσματα τρισδιάστατης αναπαράστασης κουζίνας

Βλέποντας και παρατηρώντας τα αποτελέσματα ενδελεχώς παρατηρούμε ότι στην αναπαράσταση του πάρκου με το ποδήλατο το αποτέλεσμα είναι εκπληκτικό ακόμα και για τη γύρω περιοχή με πολύ λίγα θολά σημεία σε αντίθεση με την αναπαράστασή της κουζίνας η οποία εκτός από σημεία που βλέπουμε ξεκάθαρα τις Gaussians υπάρχουν και σημεία του χώρου που είναι μαύρα κάτι που σημαίνει έλλειψη δεδομένων για τη δεδομένη περιοχή. Από τη μια δηλαδή έχουμε την αναπαράσταση του ποδηλάτου στο πάρκο για την οποία είναι σίγουρο ότι έχουν δοθεί πολύ περισσότερες φωτογραφίες σε περισσότερες και διαφορετικές οπτικές γωνίες για να έχουμε αυτό το τόσο ρεαλιστικό αποτέλεσμα ενώ στην αναπαράσταση της κουζίνας οι φωτογραφίες που δοθήκαν ήταν λιγότερες και εστιασμένες μόνο στον πάγκο της κουζίνας ο οποίος είναι ρεαλιστικός και χωρίς προβλήματα στην αναπαράσταση .

## 5.1.2 Δημιουργία μοντέλων για τρισδιάστατη αναπαράσταση από δικά μας δεδομένα

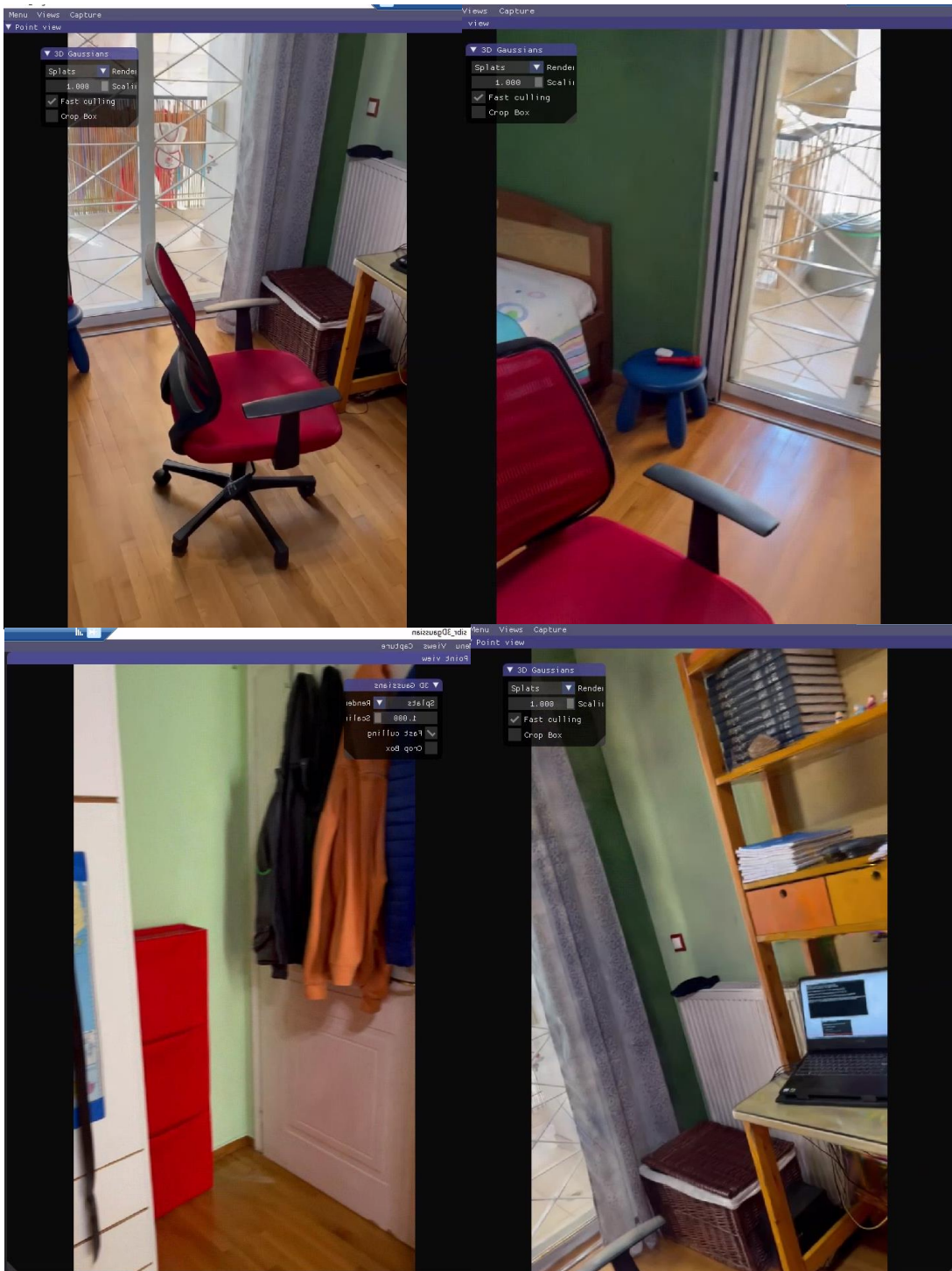
Ήρθε λοιπόν η ώρα να δείξουμε το ποσό εύκολο είναι εμείς χωρίς κανένα πρόβλημα και πολύ γρήγορα χωρίς να χρειαζόμαστε πολλά δεδομένα να αναπαραστήσουμε στον έναν 3D χώρο με Gaussian Splatting χωρίς δυσκολία και με ελάχιστα αρχικά δεδομένα. Τα μόνα λοιπόν δεδομένα εισόδου που χρειαζόμαστε είναι είτε ένα πλήθος φωτογραφιών του χώρου από διαφορετικές οπτικές γωνίες που να «σκάνάρουν» τον χώρο και να βλέπουν τα πάντα από τις περισσότερες δυνατές προοπτικές είτε ένα βίντεο από τον χώρο που να κάνει την ίδια δουλειά τραβηγμένο ακομα και από το κινητό τηλέφωνο μας. Το πιο σημαντικό είναι τραβώντας το βίντεο να κινούμαστε γύρω από τα αντικείμενα του χώρου και αργά για να μην υπάρχουν θολά σημεία και ταυτόχρονα σε διάφορες γωνίες.

Έστω λοιπόν ότι έχουμε αυτό το βίντεο το οποίο όπως εξηγήσαμε είναι πολύ εύκολο να τραβηχτεί από τον οποιονδήποτε αρκεί να υπάρχει μια ακόμα και πολύ απλή συσκευή καταγραφής βίντεο. Χρησιμοποιώντας τώρα το λογισμικό FFmpeg εξαγουμε από το βίντεο τις φωτογραφίες ανά όσα frames per second θέλουμε εμείς για να καλύπτουμε με τις φωτογραφίες σωστά όλο τον χώρο. Τι είναι όμως το FFmpeg;

Το FFmpeg είναι ένα ισχυρό και ευρέως χρησιμοποιούμενο λογισμικό ανοιχτού κώδικα για τον χειρισμό διάφορων πολυμέσων. Ουσιαστικά είναι μια συλλογή βιβλιοθηκών και εργαλείων για την κωδικοποίηση, αποκωδικοποίηση, μετακωδικοποίηση, σύμπλεξη, αποσύνδεση και αναπαραγωγή ήχου και βίντεο αρχείων. Με το FFmpeg, μπορούμε να εκτελέσουμε διάφορες εργασίες, όπως μετατροπή μεταξύ διαφορετικών μορφών ήχου και βίντεο, προσαρμογή bitrate, ανάλυσης και ποιότητας, εξαγωγή ήχου από αρχεία βίντεο, προσθήκη υποτίτλων και πολλά άλλα. Είναι ένα εργαλείο γραμμής εντολών, πράγμα που σημαίνει ότι αλληλεπιδρούμε με αυτό πληκτρολογώντας εντολές σε ένα τερματικό.

Αφού λοιπόν τώρα χρησιμοποιώντας το FFmpeg εξαγουμε τις φωτογραφίες και τις τοποθετήσουμε σε ένα φάκελο με όνομα input τότε οι φωτογραφίες είναι έτοιμες για τη πρώτη επεξεργασία τους ούτως ώστε να είναι έτοιμες έπειτα για την εκπαίδευση του μοντέλου. Η διαδικασία αυτή γίνεται πολύ απλά χρησιμοποιώντας τον κώδικα που υπάρχει στα αρχεία που κατεβάσαμε από το GitHub και συγκεκριμένα το `convert.py`. Όμως σε αυτό που πρέπει να εστιάσουμε τώρα είναι στο πρόγραμμα το οποίο προετοιμάζει τα δεδομένα μας ούτως ώστε να εκπαιδευθεί αργότερα σωστά το μοντέλο. Αυτό το πρόγραμμα είναι το Colmap. Προτού όμως περάσουμε στην επεξήγηση του τρόπου λειτουργία του Colmap ας δούμε τα αποτελέσματα της τρισδιάστατης αναπαράστασης ενός δωμάτιου για το οποίο δώσαμε εμείς ένα βίντεο και ακολουθήσαμε τη παραπάνω διαδικασία.





Εικόνες 22α,β,γ,δ:Αποτελέσματα τρισδιάστατης αναπαράστασης δωματίου με δικά μας δεδομένα

Τα μόνο δεδομένο εισόδου για τη συγκεκριμένη αναπαράσταση ήταν ένα βίντεο τραβηγμένο από ένα κινητό τηλέφωνο και το αποτέλεσμα που βλέπουμε είναι πιστό και ρεαλιστικό. Είναι φοβερό το πόσο εύκολα, γρήγορα και χωρίς επαγγελματικό εξοπλισμό μπορούμε να επιτύχουμε ένα τέτοιο αποτέλεσμα. Όλη η διαδικασία απαιτεί τέσσερα απλά βήματα την καταγραφή του βίντεο, τη δημιουργία φωτογραφιών από αυτό, την προεπεξεργασία των φωτογραφιών (χρήση COLMAP) και τέλος την εκπαίδευση του μοντέλου. Το μόνο ουσιαστικά χρονοβόρο βήμα είναι και το τελευταίο το οποίο απαιτεί ανάλογα με τα δεδομένα που θα δώσουμε γύρω στα 30 λεπτά, βέβαια ανάλογα και με τον αριθμό των φωτογραφιών που δίνονται στο COLMAP για προετοιμασία μπορεί να χρειαστούν από λίγα λεπτά μέχρι πολλές ώρες. Το μέγεθος των αρχείων που δημιουργούνται μπορεί να είναι από μερικές εκατοντάδες MB μέχρι αρκετά GB ανάλογα με τα δεδομένα εισόδου. Στη περίπτωση του δωμάτιου μας για παράδειγμα το τελικό αρχείο (.ply) δε ξεπερνάει τα 700 MB.

## 5.2 Το COLMAP στη τρισδιάστατη αναπαράσταση με τη μέθοδο Gaussian Splatting

Όπως αναφέραμε οι φωτογραφίες πρέπει να περάσουν από μια επεξεργασία πριν ξεκινήσει η εκπαίδευση του τρισδιάστατου μοντέλου. Αυτή την επεξεργασία και τη δημιουργία των κατάλληλων δεδομένων εισόδου αναλαμβάνει το COLMAP<sup>[2]</sup>.

Το COLMAP είναι ένα λογισμικό ανοιχτού κώδικα για την αναγνώριση, την αναδόμηση και την τρισδιάστατη χαρτογράφηση αντικειμένων από εικόνες. Χρησιμοποιείται ευρέως σε εφαρμογές όπως η ανάπτυξη δομών επαυξημένης πραγματικότητας, η ανάλυση εικόνων, η ρομποτική όραση, η υπολογιστική όραση, και η ανάπτυξη τρισδιάστατων μοντέλων. Το COLMAP είναι ένα λογισμικό που δημιουργεί δομή από κίνηση (το λεγόμενο Structure-from-Motion – SfM) και χρησιμοποιεί την τεχνική της πυκνής πολυεικονικής στερεοσκοπικής συνταύτισης (Multi-View Stereo (MVS)), έχοντας τόσο γραφικό περιβάλλον όσο και πρόσβαση στις εντολές του μέσα από τη γραμμή εντολών του υπολογιστή. Προσφέρει λοιπόν μια ευρεία γκάμα χαρακτηριστικών για την ανακατασκευή συλλογών εικόνων, είτε αυτές είναι ταξινομημένες είτε μη. Το λογισμικό έχει άδεια χρήσης BSD.

Οι βασικές λειτουργίες του COLMAP συνοπτικά περιλαμβάνουν:

- Εξαγωγή Χαρακτηριστικών: Ανίχνευση και περιγραφή των χαρακτηριστικών των εικόνων.
- Αντιστοίχιση Χαρακτηριστικών: Αντιστοίχιση των χαρακτηριστικών μεταξύ των εικόνων.
- Υπολογισμοί Δεδομένων Κάμερας: Υπολογισμός της θέσης και προσανατολισμού των καμερών από τις αντιστοιχίσεις χαρακτηριστικών.



- Τρισδιάστατη Αναπαράσταση: Δημιουργία τρισδιάστατων μοντέλων των αντικειμένων από τις αναδομημένες κάμερες και τις αντιστοιχίσεις χαρακτηριστικών.

Το COLMAP παρέχει ένα πλήθος αλγορίθμων και εργαλείων για την επεξεργασία εικόνων και την ανάλυση δεδομένων βίντεο, καθώς και ένα χρήσιμο περιβάλλον χρήστη για τη διευκόλυνση της διαδικασίας ανάπτυξης.

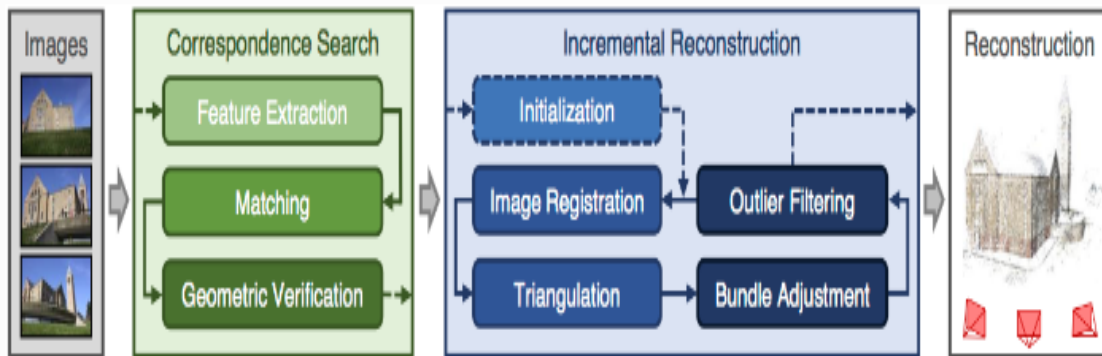
### 5.2.1 SfM (Structure-from-Motion) και MVS (Multi-View Stereo) στο COLMAP

Η τρισδιάστατη αναπαράσταση με βάση ενός πλήθους εικόνων, αρχικά, δημιουργεί μια αραιή αναπαράσταση της σκηνής και των θέσεων των καμερών των εικόνων (που δίνονται ως δεδομένα εισόδου) χρησιμοποιώντας τη Δομή από την Κίνηση (Structure-from-Motion). Αυτή η έξοδος δηλαδή αυτή η αραιή αναπαράσταση χρησιμεύει στη συνέχεια ως είσοδος στη πολυεικονική στερεοσκοπική συνταύτιση για την ανάκτηση μιας πυκνής αναπαράστασης της σκηνής.

Για να προχωρήσουμε σε αυτή τη διαδικασία ανοίγουμε το γραφικό περιβάλλον χρήστη του COLMAP το οποίο μας παρέχει ένα εργαλείο αυτόματης ανακατασκευής που απλά παίρνει ένα φάκελο με δεδομένα εισόδου εικόνες και παράγει μια αραιή και πυκνή ανακατασκευή σε έναν χώρο εργασίας. Επιλέγοντας πολύ απλά στις επιλογές Ανακατασκευή και έπειτα Αυτόματη Ανακατασκευή στο GUI η διαδικασία ξεκινάει και τα αποτελέσματα εγγράφονται στον φάκελο εργασίας.

Πως όμως αξιοποιείται η τεχνική του Structure-from-Motion στο COLMAP; Το Structure-from-Motion (SfM), όπως έχουμε ήδη εξηγήσει, είναι η διαδικασία ανακατασκευής της τρισδιάστατης δομής από τις προβολές της σε μια σειρά από εικόνες. Η είσοδος είναι ένα σύνολο από επικαλυπτόμενες εικόνες του ίδιου αντικειμένου, που έχουν ληφθεί από διαφορετικές θέσεις προβολής δηλαδή από διαφορετικές οπτικές γωνίες. Το αποτέλεσμα είναι η τρισδιάστατη ανακατασκευή του αντικειμένου, καθώς και οι ανακατασκευασμένες εσωτερικές και εξωτερικές παράμετροι των καμερών όλων των εικόνων. Συνήθως, τα συστήματα Structure-from-Motion διαμερίζουν αυτήν τη διαδικασία σε τρία στάδια:

- Ανίχνευση και εξαγωγή χαρακτηριστικών
- Αντιστοίχιση χαρακτηριστικών και γεωμετρική επαλήθευση
- Ανακατασκευή δομής και κίνησης



Εικόνα 23: Structure from Motion στο COLMAP<sup>[2]</sup>

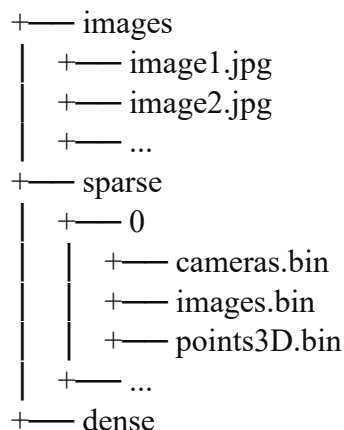
Το COLMAP αντικατοπτρίζει αυτά τα στάδια σε διάφορα βήματα της διαδικασίας δημιουργίας της τρισδιάστατης αναπαράστασης. Αν εμείς είμαστε υπεύθυνοι για τη διαδικασία λήψης φωτογραφιών, όπως είχαμε τονίσει και προηγουμένως είναι σημαντικό να ακολουθήσουμε ορισμένες επιπλέον πιο εξειδικευμένες οδηγίες για βέλτιστα αποτελέσματα τρισδιάστατης αναπαράστασης:

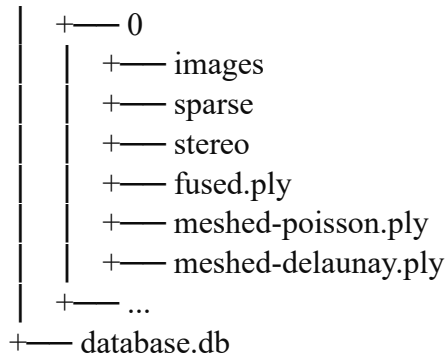
- Λήψη φωτογραφιών με περιεχόμενο στο σκηνικό. Ας αποφύγουμε τελείως εικόνες χωρίς αντικείμενα στη σκηνή(π.χ., ένα λευκό τοίχο ή ένα άδειο γραφείο). Αν στη σκηνή συμβαίνει αυτό μπορούμε να τοποθετήσουμε επιπλέον αντικείμενα στο παρασκήνιο, όπως αφίσες, καρτέκλες, μικροαντικείμενα κλπ.
- Λήψη φωτογραφιών σε παρόμοιες συνθήκες φωτισμού. Ας αποφύγουμε σκηνές με υψηλή δυναμική εύρους (π.χ., φωτογραφίες κατά την αντιμετώπιση του ήλιου με σκιές ή φωτογραφίες μέσω πορτών/παραθύρων) καθώς και τις αντανακλάσεις σε λαμπερές επιφάνειες.
- Λήψη φωτογραφιών με υψηλή οπτική επικάλυψη. Πρέπει να βεβαιωθούμε ότι κάθε αντικείμενο είναι ορατό σε τουλάχιστον 3 εικόνες - όσες περισσότερες εικόνες, τόσο το καλύτερο.
- Λήψη φωτογραφιών από διαφορετικές θέσεις προβολής. Ας μην πάρουμε μόνο εικόνες από την ίδια θέση με μόνο περιστροφή της κάμερας, είναι χρήσιμο να κάνουμε μερικά βήματα μετά από κάθε λήψη. Παράλληλα, είναι καλό να έχουμε αρκετές εικόνες από μια σχετικά παρόμοια θέση προβολής. Ας γνωρίζουμε ότι περισσότερες εικόνες δεν σημαίνει πάντα και απαραίτητα καλύτερο αποτέλεσμα αλλά πολλές φορές μπορεί να οδηγήσει σε αργή διαδικασία ανακατασκευής. Αν χρησιμοποιούμε ένα βίντεο ως είσοδο είναι κρίσιμη η επιλογή για την εξαγωγή των φωτογραφιών του ρυθμού των καρτέ ανά δευτερόλεπτο.

Όσον αφορά τη τεχνική της πυκνής πολυεικονικής στερεοσκοπικής συνταύτισης (Multi-View Stereo (MVS)) την χρησιμοποιεί για να ανακτήσει πληροφορίες βάθους κάθε pixel σε μια εικόνα. Το SfM χρησιμοποιείται για να εκτιμήσει την τρισδιάστατη δομή της σκηνής από μια σειρά από διαφορετικές λήψεις ενός αντικειμένου ή ενός περιβάλλοντος. Αφού πάρει τις πληροφορίες αυτές, μπορεί να ενώσει τις εικόνες για να παράγει έναν πυκνό νέφος σημείων (dense point cloud) της σκηνής. Αυτό το νέφος σημείων περιλαμβάνει πληροφορίες βάθους για κάθε σημείο που είναι ορατό στις διαφορετικές εικόνες. Στη συνέχεια, είναι δυνατόν να χρησιμοποιηθούν αλγόριθμοι όπως η (screened) Poisson surface reconstruction για να ανακτήσουμε την τρισδιάστατη επιφάνεια της σκηνής. Αυτοί οι αλγόριθμοι λαμβάνουν υπόψη τις πληροφορίες βάθους για κάθε σημείο στο πυκνό νέφος, προσπαθώντας να ανακατασκευάσουν την πραγματική γεωμετρία της σκηνής σε τρισδιάστατο χώρο.

## 5.2.2 Βάση Δεδομένων στο COLMAP

Το COLMAP είναι ,λοιπόν, ένα εργαλείο λογισμικού που διευκολύνει την ανάκτηση τρισδιάστατων δομών από συλλογές εικόνων. Για τον τυπικό και απλό χρήστη, οι διαδικασίες ανακατασκευής μπορούν να εκτελεστούν με λίγα βήματα χρησιμοποιώντας τις προεπιλεγμένες ρυθμίσεις, που παρέχει αυτό το λογισμικό χωρίς να γίνουν περαιτέρω αλλαγές. Εάν όμως έχουμε εξοικειωθεί αρκετά με το λογισμικό αυτό καθώς και με παρόμοια προγράμματα , το πρόγραμμα αυτό παρέχει πολλές επιλογές παραμετροποίησης, από τις οποίες μόνο κάποιες είναι εύκολα κατανοητές για αρχάριο που έχει ασχοληθεί λιγιστά με τη τρισδιάστατη αναπαράσταση και πιο συγκεκριμένα με αυτό το λογισμικό και τις δυνατότητες του. Αξίζει να σημειωθεί , όμως, ότι το πρόγραμμα λειτουργεί χωρίς κανένα πρόβλημα και πολύ αποτελεσματικά χωρίς την ανάγκη τροποποίησης οποιασδήποτε παραμέτρου. Οι προεπιλογές σε αυτή τη περίπτωση επιλέγονται αυτόματα ως μια ισορροπία μεταξύ ποιότητας ανακατασκευής και ταχύτητας. Το αποτέλεσμα του φακέλου όπου γίνονται οι παραπάνω εργασίες και θα αποθηκευτούν τελικά τα αποτελέσματα της αυτόματης ανακατασκευής έχει τη παρακάτω μορφή:





Για την επιλογή βέλτιστων επιλογών ανάκτησης για διαφορετικά σενάρια, μπορούμε να αλλάξουμε πολύ ευκολά τις ρυθμίσεις. Για να κατανοήσουμε όμως επακριβώς τι μπορούμε να αλλάξουμε και να ρυθμίσουμε ο πηγαίος κώδικας περιέχει περισσότερη τεκμηρίωση για όλες τις παραμέτρους.

Το COLMAP είναι ,όμως, λογισμικό έρευνας και όχι πρόγραμμα για επαγγελματική χρήση οπότε και σε σπάνιες περιπτώσεις μπορεί τελικά να οδηγήσει σε αποτελέσματα τα οποία είναι λανθασμένα και άκυρα εάν κάποιες προϋποθέσεις για τα δεδομένα εισόδου δεν τηρούνται. Σε αυτή την περίπτωση, το πρόγραμμα εκτυπώνει ένα αποσπασματικό μονοπάτι στην κονσόλα εξόδου για να μπορέσουμε να κατανοήσουμε ποιο είναι το πρόβλημα και να προσπαθήσουμε να το επιλύσουμε.

Ας αναφερθούμε λίγο στους όρους που χρησιμοποιούνται στο COLMAP για να κατανοήσουμε καλύτερα πως λειτουργεί το πρόγραμμα αυτό. Ο όρος "κάμερα" σχετίζεται λοιπόν με το φυσικό αντικείμενο της κάμερας, χρησιμοποιώντας τον ίδιο παράγοντα-ζουμ και φακό. Μια κάμερα ορίζει το εσωτερικό μοντέλο προβολής στο COLMAP. Μια μόνο κάμερα μπορεί να λάβει πολλαπλές εικόνες με την ίδια ανάλυση, εσωτερικές παραμέτρους και χαρακτηριστικά παραμόρφωσης. Ο όρος "εικόνα" σχετίζεται με ένα αρχείο bitmap, για παράδειγμα ένα αρχείο JPEG ή PNG στο δίσκο. Το COLMAP ανιχνεύει σημεία ενδιαφέροντος (keypoints) σε κάθε εικόνα, τα οποία περιγράφονται από αριθμητικές περιγραφές. Οι αντιστοιχίσεις μεταξύ των keypoints/περιγραφών, που βασίζονται αποκλειστικά στην εμφάνιση, ορίζονται από αντιστοιχίσεις (matches), ενώ οι αντιστοιχίσεις που είναι εσωτερικές επαληθεύονται γεωμετρικά και χρησιμοποιούνται για τη διαδικασία ανακατασκευής.

Το COLMAP καταρχάς υποθέτει ότι όλες οι εικόνες ,που δίνονται ως δεδομένα εισόδου, βρίσκονται σε έναν κοινό φάκελο εισόδου με πιθανώς ενσωματωμένους υπο-φακέλους μέσα στο βασικό φάκελο εργασιών. Αυτό που κάνει αρχικά λοιπόν είναι να

ελέγχει αναδρομικά όλες τις εικόνες που αποθηκεύονται σε αυτόν τον φάκελο ενώ αξίζει να σημειωθεί ότι υποστηρίζει διάφορες μορφές εικόνας. Άλλα αρχεία, που μπορεί να υπάρχουν στον φάκελο που δίνεται ως είσοδος, αγνοούνται αυτόματα. Εάν η υψηλή απόδοση και ταχύτητα είναι βασική απαίτηση μας και δε θέλουμε να κάνουμε καμία παραχώρηση σε αυτό το κομμάτι, τότε πρέπει να χωρίσουμε οποιαδήποτε αρχεία που δεν είναι εικόνες από τα βασικά δεδομένα εισόδου για να διευκολύνουμε τον διαχωρισμό που κάνει το λογισμικό. Οι εικόνες αναγνωρίζονται μοναδικά με τη σχετική διαδρομή τους στον φάκελο και τελικά με τη θέση τους στον δίσκο του υπολογιστή. Οπότε για μετέπειτα επεξεργασία, όπως η αντιστροφή παραμόρφωσης εικόνας ή η πυκνή ανακατασκευή, πρέπει να διατηρηθεί η σχετική δομή του φακέλου αλλιώς το λογισμικό δε θα μπορεί να αναγνωρίσει τι βρίσκεται που.

Ας δούμε λίγο πως θα χρησιμοποιήσουμε το COLMAP σε λίγα βήματα. Το πρώτο βήμα μας είναι να ξεκινήσουμε το γραφικό περιβάλλον χρήστη του COLMAP εκτελώντας τα προεγκατεστημένα δυαδικά αρχεία (Windows: COLMAP.bat, Mac: COLMAP.app) ή εκτελώντας `./src/colmap/exe/colmap_gui` από τον φάκελο CMake build. Στη συνέχεια, δημιουργούμε ένα νέο project επιλέγοντας File > New project. Σε αυτό το παράθυρο διαλόγου, πρέπει να επιλέξουμε πού θα αποθηκεύσουμε τη βάση δεδομένων και τον φάκελο που περιέχει τις εικόνες εισόδου. Για ευκολία, μπορούμε να αποθηκεύσουμε όλες τις ρυθμίσεις του έργου σε ένα αρχείο διαμόρφωσης επιλέγοντας File > Save project. Η διαμόρφωση του έργου αποθηκεύει τις πληροφορίες της απόλυτης διαδρομής της βάσης δεδομένων και του φακέλου εικόνων, εκτός από οποιεσδήποτε άλλες ρυθμίσεις παραμέτρων. Εάν αποφασίσουμε να μετακινήσουμε τη βάση δεδομένων ή τον φάκελο των εικόνων, πρέπει να αλλάξουμε τις διαδρομές αντίστοιχα δημιουργώντας ένα νέο project. Εναλλακτικά, το παραγόμενο αρχείο `.ini` διαμόρφωσης μπορεί να τροποποιηθεί απευθείας σε έναν επεξεργαστή κειμένου της επιλογής μας. Για να ανοίξουμε ένα υπάρχον project μπορούμε απλά να ανοίξουμε το αρχείο διαμόρφωσης επιλέγοντας File > Open project και όλες οι ρυθμίσεις παραμέτρων πρέπει να αποκατασταθούν.

### 5.2.3 Αναγνώριση και Ταύτιση Χαρακτηριστικών στο COLMAP

Πως λειτουργεί όμως η αναγνώριση και η ταύτιση χαρακτηριστικών στο COLMAP; Σε πρώτο στάδιο, με την ανίχνευση/εξαγωγή χαρακτηριστικών εντοπίζονται αραιά σημεία χαρακτηριστικών στην εικόνα και περιγράφεται η εμφάνιση τους χρησιμοποιώντας έναν αριθμητικό περιγραφέα. Το COLMAP εισάγει εικόνες και εκτελεί την ανίχνευση/εξαγωγή χαρακτηριστικών σε ένα βήμα, προκειμένου να φορτώνει τις εικόνες από τον δίσκο μόνο μία φορά. Προχωρώντας στην εξαγωγή των χαρακτηριστικών επιλέγουμε Processing > Extract features και πρέπει πρώτα να αποφασίσουμε για το μοντέλο εσωτερικής κάμερας που θα χρησιμοποιηθεί. Μπορούμε είτε να εξάγουμε αυτόματα πληροφορίες από τα ενσωματωμένα δεδομένα EXIF είτε να καθορίσουμε χειροκίνητα τις εσωτερικές παραμέτρους, για παράδειγμα όπως αποκτήθηκαν σε μια

εργαστηριακή βαθμονόμηση. Εάν μια εικόνα έχει μερικές πληροφορίες EXIF, το COLMAP προσπαθεί να βρει τις απουσιάζουσες προδιαγραφές της κάμερας σε μια μεγάλη βάση δεδομένων αυτόματα. Εάν όλες οι εικόνες μας ελήφθησαν από την ίδια φυσική κάμερα με ίδιο παράγοντα ζουμ, συνιστάται να μοιράζουμε τις εσωτερικές παραμέτρους μεταξύ όλων των εικόνων. Ας σημειώσουμε πως το πρόγραμμα θα τερματίσει ακατάλληλα εάν το ίδιο μοντέλο κάμερας μοιράζεται μεταξύ όλων των εικόνων, αλλά όλες οι εικόνες δεν έχουν το ίδιο μέγεθος ή την ίδια εστιακή απόσταση (focal length) EXIF. Εάν έχουμε αρκετές ομάδες εικόνων που μοιράζονται τις ίδιες εσωτερικές παραμέτρους της κάμερας, μπορούμε εύκολα να τροποποιήσουμε τα μοντέλα της κάμερας αργότερα. Εάν δεν είμαστε εξοικειωμένοι με αυτή τη διαδικασία, απλά κρατάμε τις προεπιλεγμένες παραμέτρους.

Σε δεύτερο στάδιο τώρα, η αντιστοίχιση και γεωμετρική επαλήθευση χαρακτηριστικών εντοπίζει τις αντιστοιχίες μεταξύ των χαρακτηριστικών σημείων σε διαφορετικές εικόνες. Για να προχωρήσουμε σε αυτό το στάδιο επιλέγουμε Processing > Match features και επιλέγουμε έναν από τους παρεχόμενους τρόπους αντιστοίχισης, που προορίζονται για διαφορετικά σενάρια εισόδου:

- **Exhaustive Matching**: Εάν ο αριθμός των εικόνων στο σύνολο δεδομένων σας είναι σχετικά μικρός (μέχρι κάποιες εκατοντάδες), αυτή η λειτουργία ταιριάσματος πρέπει να είναι αρκετά γρήγορη και οδηγεί στα καλύτερα αποτελέσματα ανακατασκευής. Εδώ, κάθε εικόνα ταιριάζεται με κάθε άλλη εικόνα, ενώ το μέγεθος του block (αναφέρεται σε μια σειρά από bytes ή bits, η οποία συνήθως περιέχει έναν ολόκληρο αριθμό εγγραφών και έχει μέγιστο μήκος) καθορίζει πόσες εικόνες φορτώνονται από τον δίσκο στη μνήμη ταυτόχρονα.
- **Sequential Matching**: Αυτή η λειτουργία είναι χρήσιμη εάν οι εικόνες αποκτώνται με διαδοχική σειρά, για παράδειγμα, από μια βιντεοκάμερα. Σε αυτήν την περίπτωση, συνεχόμενα καρέ έχουν οπτική επικάλυψη και δεν χρειάζεται να ταιριάξουμε όλα τα ζευγάρια εικόνων, εξοικονομώντας με αυτόν τον τρόπο πολύτιμο χρόνο. Αντ' αυτού, οι συνεχόμενες εικόνες που λήφθηκαν ταιριάζονται μεταξύ τους. Αυτή η λειτουργία ταιριάσματος διαθέτει ενσωματωμένο εντοπισμό βρόγχων βάσει ενός δέντρου λεξιλογίου, όπου κάθε νιοστή εικόνα ταιριάζεται με τις οπτικά παρόμοιες εικόνες της. Είναι σημαντικό να γνωρίζουμε ότι οι ονομασίες των αρχείων εικόνων πρέπει να είναι τακτοποιημένες διαδοχικά (π.χ. image0001.jpg, image0002.jpg, κλπ.). Η σειρά στη βάση δεδομένων δεν είναι σημαντική, δεδομένου ότι οι εικόνες ταξινομούνται ρητά σύμφωνα με τα ονόματα των αρχείων τους. Ο εντοπισμός βρόγχων απαιτεί ένα προ-εκπαιδευμένο δέντρο λεξιλογίου, το οποίο μπορεί να ληφθεί από τη διεύθυνση <https://demuc.de/colmap/>.
- **Vocabulary Tree Matching**: Σε αυτή τη λειτουργία ταιριάσματος κάθε εικόνα ταιριάζεται με τις οπτικά κοντινότερες γειτονικές της χρησιμοποιώντας ένα δέντρο λεξιλογίου με επανατακτοποίηση του χώρου. Αυτή είναι η προτεινόμενη λειτουργία ταιριάσματος για μεγάλες συλλογές εικόνων (αρκετές χιλιάδες). Αυτό

απαιτεί ένα προ-εκπαιδευμένο δέντρο λεξιλογίου, το οποίο μπορεί να ληφθεί από τη διεύθυνση <https://demuc.de/colmap/> (όπως αναφέρθηκε και παραπάνω)

- **Spatial Matching**: Η λειτουργία αυτή ταιριάζει κάθε εικόνα με τις κοντινότερες γειτονικές εικόνες στον χώρο. Οι χωρικές τοποθεσίες μπορούν να οριστούν χειροκίνητα στη διαχείριση της βάσης δεδομένων. Από προεπιλογή, το COLMAP εξάγει επίσης πληροφορίες GPS από τα EXIF και τις χρησιμοποιεί για την αναζήτηση των χωρικά πλησιέστερων γειτονικών. Εάν υπάρχουν ακριβείς πληροφορίες προηγούμενης τοποθεσίας, αυτή είναι η προτεινόμενη λειτουργία ταιριάσματος.
- **Transitive Matching**: Αυτή η λειτουργία χρησιμοποιεί τις μεταβατικές σχέσεις των ήδη υπάρχοντων αντιστοιχίσεων χαρακτηριστικών για την παραγωγή μιας πιο πλήρους αντιστοίχισης. Αν μια εικόνα A ταιριάζει με μια εικόνα B και η B ταιριάζει με την E, τότε αυτή η αντιστοίχιση επιχειρεί να ταιριάζει απευθείας την A με την E.
- **Custom Matching**: Η λειτουργία αυτή μας επιτρέπει να καθορίσουμε μεμονωμένα ζευγάρια εικόνων για την αντιστοίχιση ή να εισαγάγουμε μεμονωμένες αντιστοιχίσεις χαρακτηριστικών. Για να γίνει αυτή η διαδικασία χρειάζεται να ανατρέξουμε στις αναλυτικές οδηγίες του COLMAP για τον τρόπο με τον οποίο θα δώσουμε τα δεδομένα εισόδου.

Αφού τελειώσουμε ρυθμίζοντας τις παραπάνω παραμέτρους και επιλογές επιλέγουμε το Match και περιμένουμε να ολοκληρωθεί η διαδικασία ταιριάσματος ή μπορούμε να την ακυρώσουμε στη μέση γιατί αυτό το βήμα μπορεί να διαρκέσει αρκετό χρονικό διάστημα, ανάλογα με τον αριθμό των εικόνων, τον αριθμό των χαρακτηριστικών ανά εικόνα και την επιλεγμένη λειτουργία ταιριάσματος. Οι αναμενόμενοι χρόνοι για το αναλυτικό ταιρίασμα είναι από λίγα λεπτά για δεκάδες εικόνες έως λίγες ώρες για εκατοντάδες εικόνες και ημέρες ή εβδομάδες για χιλιάδες εικόνες. Εάν ακυρώσουμε τη διαδικασία ταιριάσματος ή εισαγάγουμε νέες εικόνες μετά το ταιρίασμα, το COLMAP θα ταιριάζει μόνο τα ζεύγη εικόνων που δεν έχουν ταιριαστεί προηγουμένως. Το περιττό κόστος της παράλειψης ήδη ταιριασμένων ζευγών εικόνων είναι χαμηλό. Αυτό επίσης επιτρέπει το ταιρίασμα πρόσθετων εικόνων που εισάγονται μετά από μια αρχική διαδικασία ταιριάσματος και επιτρέπει τον συνδυασμό διαφορετικών λειτουργιών ταιριάσματος για το ίδιο σύνολο δεδομένων. Όλα τα δεδομένα που εξάγονται θα αποθηκευτούν στο αρχείο βάσης δεδομένων και μπορούν να ελεγχθούν/διαχειριστούν στο εργαλείο διαχείρισης βάσεων δεδομένων. Αξίζει να σημειωθεί ότι η αντιστοίχιση χαρακτηριστικών απαιτεί μια ισχυρή μονάδα επεξεργασίας γραφικών (GPU) και ότι η απόδοση του υπολογιστή μας μπορεί να μειωθεί σημαντικά κατά τη διάρκεια της διαδικασίας ταιριάσματος.



Μετά την εκτέλεση των δύο προηγούμενων βημάτων, μπορούμε να ξεκινήσουμε τη διαδικασία ανακατασκευής (επιλέγοντας Reconstruction > Start). Το COLMAP φορτώνει αρχικά όλα τα εξαγόμενα δεδομένα από τη βάση δεδομένων στη μνήμη και εκκινεί την ανακατασκευή από ένα αρχικό ζευγάρι εικόνων. Στη συνέχεια, η σκηνή επεκτείνεται σταδιακά με την καταχώρηση νέων εικόνων και την τριγωνοποίηση νέων σημείων. Τα αποτελέσματα εμφανίζονται σε "πραγματικό χρόνο" κατά τη διάρκεια αυτής της διαδικασίας ανακατασκευής. Το COLMAP επιχειρεί να ανακατασκευάσει πολλαπλά μοντέλα εάν όχι όλες οι εικόνες καταχωρούνται στο ίδιο μοντέλο. Τα διαφορετικά μοντέλα μπορούν να επιλεγούν από το αναπτυσσόμενο μενού στη γραμμή εργαλείων. Εάν τα διαφορετικά μοντέλα έχουν κοινές καταχωρημένες εικόνες, μπορούμε να χρησιμοποιήσουμε το εκτελέσιμο `model_converter` για να τα συγχωνεύσουμε σε μία μοναδική ανακατασκευή. Ιδανικά, η ανακατασκευή λειτουργεί καλά και όλες οι εικόνες καταχωρούνται ορθά αλλιώς μπορούμε να επιχειρήσουμε επιπλέον και καλύτερη αντιστοίχιση χρησιμοποιώντας «Exhaustive Matching» ενεργοποιώντας την καθοδήγηση της αντιστοίχισης, αυξάνοντας τον αριθμό των πλησιέστερων γειτόνων στην «Vocabulary Tree Matching» ή αυξάνοντας την επικάλυψη στην «Sequential Matching». Εναλλακτικά επιλέγουμε χειροκίνητα ένα ζεύγος εικόνων, εάν το COLMAP αποτυγχάνει να αρχικοποιήσει σωστά. Επιλέγουμε Reconstruction > Reconstruction options > Init and set images και ορίζουμε εικόνες από το εργαλείο διαχείρισης βάσεων δεδομένων που έχουν αρκετές αντιστοιχίσεις από διαφορετικές γωνίες θέασης.

Μετά την ανακατασκευή μιας αραιής αναπαράστασης του σκηνικού και των θέσεων των καμερών των εικόνων εισόδου, το MVS (Multi-View Stereo) μπορεί τώρα να ανακτήσει και να δημιουργήσει μια πιο πυκνή γεωμετρία του σκηνικού. Το COLMAP διαθέτει μια ενσωματωμένη διαδικασία πυκνής ανακατασκευής για την παραγωγή «depth and normal maps» για όλες τις εγγεγραμμένες εικόνες, και για τη συγχώνευση τους σε έναν πυκνό νέφος σημείων, για να πέτυχει τελικά την εκτίμηση ενός πυκνού επιφανειακού μοντέλου από το συγχωνευμένο πυκνό νέφος σημείων χρησιμοποιώντας την ανακατασκευή Poisson ή Delaunay.

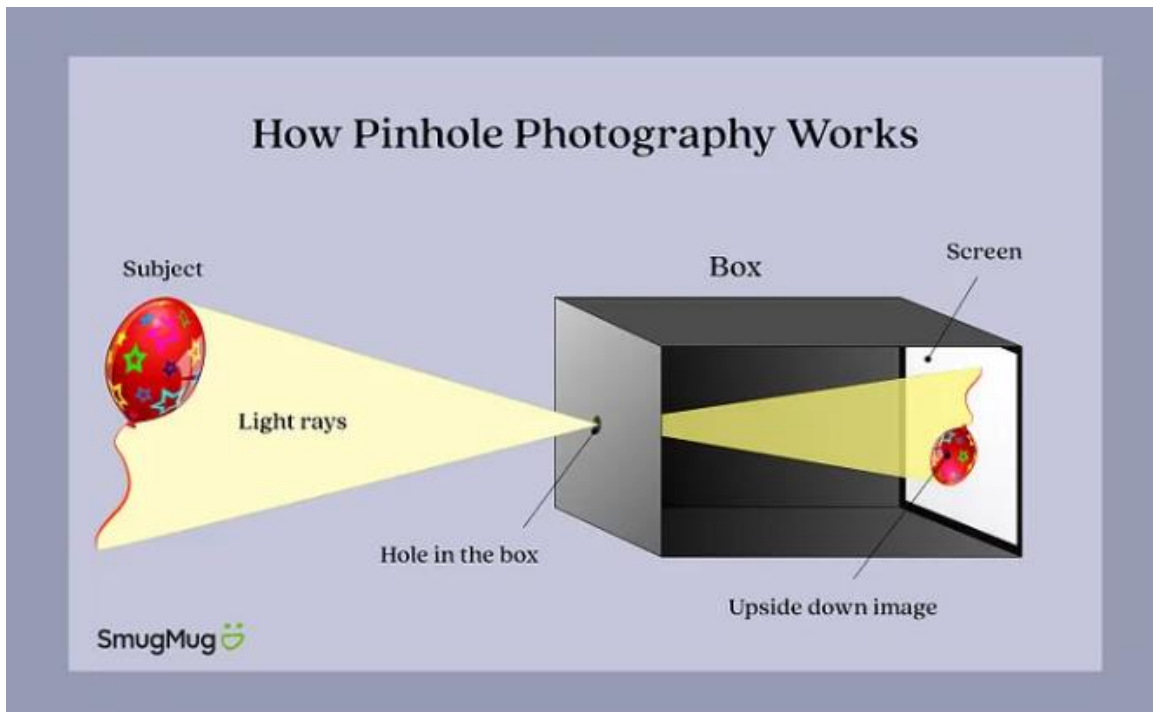
## 5.2.4 Αποδεκτά Μοντέλα Κάμερας στο COLMAP

Το COLMAP ενσωματώνει και χρησιμοποιεί διαφορετικά μοντέλα κάμερας με διαφορετική πολυπλοκότητα. Εάν δεν είναι γνωστές οι προαπαιτούμενες εσωτερικές παράμετροι, είναι γενικά καλύτερο να χρησιμοποιήσουμε το απλούστερο μοντέλο κάμερας που είναι όσο πολύπλοκο χρειάζεται για να μοντελοποιήσει αποτελεσματικά τα εφέ παραμόρφωσης:

- SIMPLE PINHOLE, PINHOLE: Χρησιμοποιούμε αυτά τα μοντέλα κάμερας εάν οι εικόνες μας δεν είναι παραμορφωμένες από πριν. Αυτά χρησιμοποιούν μια και δύο παραμέτρους εστίασης, αντίστοιχα. Αξίζει να αναφερθεί ότι ακόμα και στην περίπτωση μη παραμορφωμένων εικόνων, το COLMAP μπορεί να προσπαθήσει να βελτιώσει τις εσωτερικές παραμέτρους με ένα πιο πολύπλοκο μοντέλο κάμερας.
- SIMPLE RADIAL, RADIAL: Αυτό πρέπει να είναι το προτιμώμενο μοντέλο κάμερας εάν οι εσωτερικές παράμετροι είναι άγνωστες και κάθε εικόνα έχει διαφορετική βαθμονόμηση κάμερας, για παράδειγμα στην περίπτωση φωτογραφιών από το διαδίκτυο. Και τα δύο μοντέλα είναι απλοποιημένες εκδόσεις του μοντέλου OPENCV που μοντελοποιούν μόνο τα ακτινικά εφέ παραμόρφωσης με μια και δύο παραμέτρους, αντίστοιχα.
- OPENCV, FULL OPENCV: Θα χρησιμοποιήσουμε αυτά τα μοντέλα κάμερας εάν γνωρίζουμε τις παραμέτρους βαθμονόμησης από πριν. Μπορούμε επίσης να δοκιμάσουμε να αφήσουμε το COLMAP να εκτιμήσει τις παραμέτρους, εάν τις γνωστοποιήσουμε, αφού βέβαια γνωρίζουμε τις εσωτερικές παραμέτρους για πολλές εικόνες. Όμως, η αυτόματη αυτή εκτίμηση των παραμέτρων θα αποτύχει πιθανότατα, εάν κάθε εικόνα έχει ξεχωριστό σύνολο εσωτερικών παραμέτρων.
- SIMPLE RADIAL FISHEYE, RADIAL FISHEYE, OPENCV FISHEYE, FOV, THIN\_PRISM FISHEYE: Χρησιμοποιούμε αυτά τα μοντέλα κάμερας για φακούς υπερευρυγώνιου τύπου (fisheye) και ας τονίσουμε ότι όλα τα άλλα μοντέλα δεν είναι πραγματικά ικανά να μοντελοποιήσουν τα εφέ παραμόρφωσης των υπερευρυγώνιου φακών.

Μπορούμε να ελέγξουμε τις εκτιμώμενες εσωτερικές παραμέτρους κάνοντας διπλό κλικ σε συγκεκριμένες εικόνες στον προβολέα μοντέλου ή εξάγοντας το μοντέλο και

ανοίγοντας το αρχείο cameras.txt. Για να επιτύχουμε βέλτιστα αποτελέσματα ανακατασκευής, μπορεί να χρειαστεί να δοκιμάσουμε διαφορετικά μοντέλα κάμερας για το πρόβλημα τρισδιάστατης αναπαράστασης που αντιμετωπίζουμε . Γενικά, όταν η ανακατασκευή αποτυγχάνει και οι εκτιμηθείσες τιμές εστίασης και οι συντελεστές παραμόρφωσης είναι σημαντικά λανθασμένοι, αυτό είναι ένδειξη χρήσης ενός πολύ πολύπλοκου μοντέλου κάμερας για τη συγκεκριμένη αναπαράσταση. Αντίθετα, εάν το COLMAP χρησιμοποιεί πολλές επαναλαμβανόμενες τοπικές και γενικές προσαρμογές, αυτό είναι ένδειξη χρήσης ενός πολύ απλού μοντέλου κάμερας που δεν είναι σε θέση να μοντελοποιήσει πλήρως τα εφέ παραμόρφωσης .Κλείνοντας, αξίζει να αναφερθεί ότι μπορούμε να θέτουμε σε πολλές φωτογραφίες τις ίδιες εσωτερικές παραμέτρους για να πετύχουμε πιο αξιόπιστα αποτελέσματα ή μπορούμε ακόμα και να ρυθμίσουμε εμείς τις εσωτερικές παραμέτρους ακόμα και κατά τη διάρκεια της ανακατασκευής.



Εικόνα 24: Τρόπος λειτουργίας κάμερας PINHOLE<sup>[31]</sup>

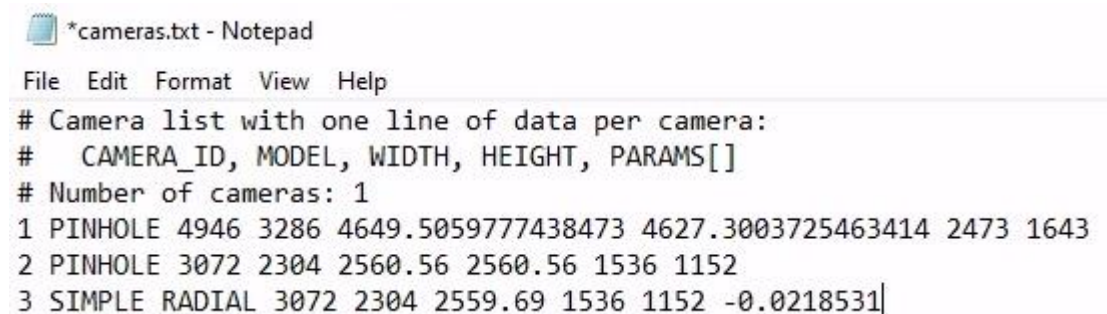
## 5.2.5 Δεδομένα εξόδου στο COLMAP

Ως προεπιλογή, το COLMAP χρησιμοποιεί ένα δυαδικής μορφής αρχείο (μηχανικά αναγνώσιμο και αρκετά γρήγορο) για την αποθήκευση αραιών μοντέλων. Επιπλέον, το COLMAP παρέχει την επιλογή να αποθηκεύει τα αραιά μοντέλα ως αρχεία κειμένου (εύκολα αναγνώσιμα και επεξεργάσιμα από εμάς αλλά πιο αργά). Και στις δύο περιπτώσεις, η πληροφορία διαιρείται σε τρία αρχεία για τις πληροφορίες σχετικά με τις κάμερες, τις εικόνες και τα σημεία. Οποιοσδήποτε φάκελος που περιέχει αυτά τα τρία αρχεία αποτελεί ένα αραιό μοντέλο. Τα δυαδικά αρχεία έχουν την κατάληξη αρχείου `.bin` και τα αρχεία κειμένου έχουν την κατάληξη αρχείου `.txt`. Ας αναφέρουμε ότι όταν φορτώνουμε ένα μοντέλο από μια βάση δεδομένων που περιέχει και δυαδικά και αρχεία κειμένου, το COLMAP προτιμά τη δυαδική μορφή.

Για να μετατρέψουμε ανάμεσα στη δυαδική και την μορφή κειμένου στο GUI (Graphical User Interface), μπορούμε να εισάγουμε και να φορτώσουμε το μοντέλο και στη συνέχεια να εξάγουμε το μοντέλο στην επιθυμητή μορφή εξόδου χρησιμοποιώντας τις αντίστοιχες εντολές. Επιπλέον, μπορούμε να εξάγουμε τα αραιά μοντέλα σε άλλες μορφές, όπως αρχεία NVM του VisualSfM, αρχεία Bundler, PLY, VRML, κ.λπ. Για να μετατρέψουμε τα μοντέλα ανάμεσα σε διάφορες μορφές αρχείων από τη γραμμή εντολών, χρησιμοποιούμε το εκτελέσιμο `model_converter`.

Όσον αφορά τα αρχεία με μορφή κειμένου, το COLMAP εξάγει τα παρακάτω τρία αρχεία κειμένου για κάθε ανακατασκευασμένο μοντέλο: `cameras.txt`, `images.txt` και `points3D.txt`. Τα σχόλια στο αρχείο κειμένου που ξεκινούν με τον χαρακτήρα `"#"` στην αρχή τους αγνοούνται. Οι πρώτες γραμμές σχολίων περιγράφουν επιγραμματικά τη μορφή των αρχείων κειμένου, όπως θα περιγράφεται πιο λεπτομερώς παρακάτω.

Το αρχείο `cameras.txt` περιέχει τις εσωτερικές παραμέτρους όλων των καμερών που δημιουργήθηκαν και ανακατασκευάστηκαν και κάθε μια σειρά του αρχείου περιέχει τα δεδομένα για μια κάμερα.



```
*cameras.txt - Notepad
File Edit Format View Help
# Camera list with one line of data per camera:
# CAMERA_ID, MODEL, WIDTH, HEIGHT, PARAMS[]
# Number of cameras: 1
1 PINHOLE 4946 3286 4649.5059777438473 4627.3003725463414 2473 1643
2 PINHOLE 3072 2304 2560.56 2560.56 1536 1152
3 SIMPLE_RADIAL 3072 2304 2559.69 1536 1152 -0.0218531|
```

Εικόνα 25: Μορφή του αρχείου `cameras.txt`

Όπως παρατηρούμε το αρχείο με τα δεδομένα για τις κάμερες περιλαμβάνει 3 διαφορετικές κάμερες με διαφορετικά μοντέλα παραμόρφωσης δύο εκ των οποίων έχουν ίδιες διαστάσεις φακού (3072x2304). Η αναλυτική περιγραφή των παραμέτρων αφορά το μοντέλο της εκάστοτε κάμερας αλλά έχουν να κάνουν και με την εστιακή απόσταση (focal

length ) , με το κύριο σημείο του φακού της κάμερας (principal point) και με άλλες διάφορες παραμέτρους που έχουν να κάνουν με τη παραμόρφωση.

Όσον αφορά τώρα το αρχείο images.txt αυτό περιέχει τη θέση και τα σημεία κλειδιά όλων των ανακατασκευασμένων εικόνων στο σύνολο δεδομένων και έχει μια τέτοια μορφή :

```
# Image list with two lines of data per image:
# IMAGE_ID, QW, QX, QY, QZ, TX, TY, TZ, CAMERA_ID, NAME
# POINTS2D[] as (X, Y, POINT3D_ID)
# Number of images: 2, mean observations per image: 2
1 0.851773 0.0165051 0.503764 -0.142941 -0.737434 1.02973 3.74354 1 P1180141.JPG
2362.39 248.498 58396 1784.7 268.254 59027 1784.7 268.254 -1
2 0.851773 0.0165051 0.503764 -0.142941 -0.737434 1.02973 3.74354 1 P1180142.JPG
1190.83 663.957 23056 1258.77 640.354 59070|
```

Εικόνα 26: Μορφή του αρχείου images.txt

Σε αυτό το αρχείο παρατηρούμε ότι οι πρώτες δύο γραμμές καθορίζουν τις πληροφορίες της πρώτης εικόνας, και έτσι συνεχίζεται. Η ανακατασκευασμένη θέση μιας εικόνας καθορίζεται ως η προβολή από τον κόσμο στο σύστημα συντεταγμένων της κάμερας μιας εικόνας χρησιμοποιώντας ένα quaternion (QW, QX, QY, QZ) και ένα διάνυσμα μετάφρασης (TX, TY, TZ). Το quaternion ορίζεται χρησιμοποιώντας τον κανόνα του Hamilton, ο οποίος χρησιμοποιείται, για παράδειγμα, και από τη βιβλιοθήκη Eigen. Οι συντεταγμένες της προβολής/κέντρου της κάμερας δίνονται από το  $-R^t * T$ , όπου  $R^t$  είναι ο αντίστροφος του 3x3 πίνακα περιστροφής που σχηματίζεται από το quaternion και το T είναι το διάνυσμα μετάφρασης. Το τοπικό σύστημα συντεταγμένων της κάμερας μιας εικόνας ορίζεται με τον τρόπο ότι ο άξονας X δείχνει προς τα δεξιά, ο άξονας Y προς τα κάτω και ο άξονας Z προς τα μπροστά όπως φαίνονται από την εικόνα.

Στο παράδειγμα μας και οι δύο εικόνες στο παραπάνω παράδειγμα χρησιμοποιούν το ίδιο μοντέλο κάμερας και μοιράζονται εσωτερικές παραμέτρους (CAMERA\_ID = 1). Το όνομα της εικόνας είναι σχετικό και ταυτίζεται με το όνομα της εικόνας στο βασικό φάκελο εικόνων του project. Η πρώτη εικόνα έχει 3 σημεία κλειδιά και η δεύτερη εικόνα έχει 2 σημεία κλειδιά, ενώ η τοποθεσία των σημείων κλειδιών καθορίζεται σε συντεταγμένες pixel. Και οι δύο εικόνες παρατηρούν 2 3D σημεία και να σημειώσουμε ότι το τελευταίο σημείο κλειδί της πρώτης εικόνας δεν παρατηρεί ένα 3D σημείο στην ανακατασκευή καθώς ο αναγνωριστής 3D σημείου είναι -1.

Κλείνοντας , το αρχείο points3D.txt περιέχει τις πληροφορίες όλων των ανακατασκευασμένων 3D σημείων στο σύνολο δεδομένων χρησιμοποιώντας μία γραμμή ανά σημείο, όπως φαίνεται παρακάτω :

```

points3D.txt - Notepad
File Edit Format View Help
# 3D point list with one line of data per point:
# POINT3D_ID, X, Y, Z, R, G, B, ERROR, TRACK[] as (IMAGE_ID, POINT2D_IDX)
# Number of points: 54275, mean track length: 4.6884569322892675
27590 4.7790641907029965 2.7006307887502494 -4.6488504020408525 91 79 67 1.3910532432768068 147 4755 148 3130 149 859
63081 0.27807352339462266 0.50790126975472483 -0.015576209773846036 0 0 0 1.24592195331602 58 12025 62 12180
62872 -7.3204958102017255 -3.424292839197705 -2.7832476714903107 63 65 62 1.2823662810936773 76 8545 26 9457
63145 -0.029486030809196786 0.31050968482027236 0.66357518750679034 72 72 57 1.4098770376595942 59 7666 110 8985 112 8590 114 8425 65 4526 63 7365 66 7191 115 5970
18418 -0.62326684820011025 1.2021453432215947 2.0188102419474037 101 102 50 1.7232456058276977 177 7295 139 7395 141 7573
62936 6.4718365833319838 -4.2807684546430043 -29.016674168855932 254 255 255 1.8435356859239658 18 7437 16 9966 14 10786
16451 3.426536634815766 1.156021873950009 1.796255100501098 103 93 84 1.0203813857427724 177 1229 193 11336 194 9861 179 853 139 878 181 4045 141 1077 146 7809
62270 13.287746513786617 -1.3866383866854179 2.3465080846481415 50 54 31 1.619271314923868 60 6991 9 6038 10 5056
60705 0.16618305975896271 0.82927244422884461 0.19309804641480841 173 181 190 1.589126580330807 59 10085 60 11485 9 10943 58 12087 98 10842
7274 -3.0122487525410606 0.62331454380778706 4.0557734431366699 75 86 26 0.8094608642488561 172 3881 171 734 173 4095
62403 -7.9063640891974583 -2.0802358865589943 3.5564980503673205 28 28 18 1.4672612378792618 27 7813 30 2479 33 4153 86 6228 35 4910 88 5679 90 6244 36 5535 166 82
61008 0.2629628187660471 0.55843282504801706 -0.038216972825725171 112 114 113 1.5850971122507882 29 10823 84 5113 82 4923 30 5113
63080 0.27076601889247165 0.50331316900619327 -0.0097759366260570125 63 68 44 2.3633971435982239 8 12625 62 12179 10 7495
63144 -0.03046868154027604 0.31027964850161022 0.66453462197312774 71 66 56 1.6282500572654632 59 7667 150 9753 110 8986 112 8591 114 8424 65 4525 63 7364 66 7190
62938 12.718585386391389 0.19429476124464462 -17.089722026007244 42 47 24 1.8212551966309449 17 10253 68 11132 16 10278
62252 -0.055517433580203591 0.45570393053033603 0.70535361130411822 221 221 219 2.3349193322659554 11 2470 58 4445 108 4368 9 2446
59854 15.90199430224438 -2.2666127286469147 -7.5091882376671037 37 47 44 1.8677601474961882 112 786 64 2242 14 3669

```

Εικόνα 27: Μορφή του αρχείου points3D.txt

Εδώ, υπάρχουν ανακατασκευασμένα 3D σημεία, όπου το POINT2D\_IDX ορίζει το δείκτη(με βάση το μηδέν) του σημείου κλειδιού στο αρχείο images.txt. Το σφάλμα δίνεται σε εικονοστοιχεία(pixel) του σφάλματος αναπροβολής.

### 5.3 Τα Quaternions στην Τρισδιάστατη Αναπαράσταση

Όπως αναφέραμε και προηγουμένως, η εκπαίδευσή του μοντέλου γίνεται χρησιμοποιώντας τα δεδομένα εξόδου του COLMAP. Σε αυτά τα δεδομένα εξόδου και συγκεκριμένα στο images.txt είδαμε να καταγράφονται τα quaternions για κάθε φωτογραφία χωρίς όμως να έχουμε εξηγήσει έστω και συνοπτικά τι είναι αυτά τα quaternions και γιατί είναι τόσο χρήσιμα στη τρισδιάστατη αναπαράσταση.

Τα quaternions είναι μαθηματικές οντότητες που επεκτείνουν τους σύνθετους αριθμούς και χρησιμοποιούνται εκτενώς στα γραφικά υπολογιστών τριών διαστάσεων. Ορίζονται ως μια τετραδιάστατη επέκταση των σύνθετων αριθμών και έχουν εφαρμογές σε διάφορους τομείς πέρα από τα γραφικά, συμπεριλαμβανομένης της φυσικής, της ρομποτικής και της υπολογιστικής όρασης και κατά συνέπεια της τρισδιάστατης αναπαράστασης.

Μερικοί βασικοί λόγοι για τους οποίους τα quaternions είναι σημαντικά για την αναπαράσταση στις τρεις διαστάσεις είναι οι εξής:

- Αποτελεσματική Αναπαράσταση Περιτροφών: Τα quaternions προσφέρουν ένα συμπαγές και αποδοτικό τρόπο αναπαράστασης περιστροφών στον τρισδιάστατο χώρο. Αντίθετα με άλλες μεθόδους όπως οι γωνίες Euler, τα quaternions αποφεύγουν προβλήματα όπως το “gimbal lock” (λόγω του οποίου κάποιες

προσανατολίσεις δεν μπορούν να αναπαρασταθούν) και παρέχουν έναν ομαλό, συνεχή τρόπο παρεμβολής μεταξύ των περιστροφών.

- Αριθμητική Ακρίβεια: Τα quaternions πλεονεκτούν έναντι των πινάκων περιστροφής (μια άλλη μέθοδος αναπαράστασης περιστροφών) όσον αφορά την αριθμητική ακρίβεια. Κατά την αντιμετώπιση πολλαπλών περιστροφών ή παρεμβολών μεταξύ συγκεκριμένων περιστροφών, τα quaternions είναι λιγότερο επιρρεπή σε σφάλματα στρογγυλοποίησης και ανακρίβειες υπολογισμών.
- Παρεμβολή και Ομαλή Κίνηση: Τα quaternions μπορούν εύκολα να παρεμβάλλονται για να δημιουργήσουν ομαλές κινήσεις μεταξύ διαφορετικών προσανατολισμών. Η γραμμική παρεμβολή μεταξύ των quaternions εξασφαλίζει ότι οι παρεμβολές παρακολουθούν τη συντομότερη διαδρομή στον τρισδιάστατο χώρο, η οποία είναι οπτικά ελκυστική και φυσικά ρεαλιστική.
- Πολυμορφία: Τα quaternions μπορούν να χρησιμοποιηθούν όχι μόνο για την αναπαράσταση περιστροφών αλλά και για τον προσανατολισμό στον τρισδιάστατο χώρο. Αυτή η πολυμορφία τους επιτρέπει να εφαρμόζονται σε διάφορες εργασίες, συμπεριλαμβανομένων της αναπαράστασης χαρακτήρων, του ελέγχου κάμερας (όπως στο COLMAP) και των προσομοιώσεων φυσικής.
- Μαθηματικές Ιδιότητες: Τα quaternions έχουν συγκεκριμένες μαθηματικές ιδιότητες που τα καθιστούν κατάλληλα για την αναπαράσταση περιστροφών. Για παράδειγμα, ο πολλαπλασιασμός quaternions αντιστοιχεί στη σύνθεση περιστροφών (αν και πρέπει να τονιστεί ότι έχει σημασία η σειρά με την οποία πολλαπλασιάζουμε τα quaternions), και τα quaternions με μέγεθος 1 αναπαριστούν μοναδικά περιστροφές χωρίς ασάφεια.

Λόγω αυτών των πλεονεκτημάτων, τα quaternions έχουν γίνει ένα βασικό εργαλείο στα γραφικά υπολογιστών τριών διαστάσεων και σε συναφείς τομείς, επιτρέποντας την αποτελεσματική και αξιόπιστη διαχείριση του προσανατολισμού και των περιστροφών σε εικονικά περιβάλλοντα.



## 5.3 Χρήσιμες πληροφορίες για τον κώδικα εκπαίδευσης του μοντέλου με τη μέθοδο Gaussian Splatting

Ας επανέλθουμε λοιπόν στον κώδικα που εκπαιδεύει το μοντέλο και αναπαριστά τελικά τη σκηνή με τη μέθοδο Gaussian Splatting. Προτού λοιπόν προχωρήσουμε στις προσαρμογές που μπορούμε να κάνουμε στον κώδικα με σκοπό να πέτυχουμε να αξιοποιήσουμε διαφορετικά αρχικά δεδομένα δηλαδή δεδομένα που δε προέρχονται μόνο από το COLMAP αξίζει να δούμε τι μπορούμε να αλλάξουμε στον κώδικα εύκολα και γρήγορα.

Οι πιο εύκολες αλλαγές που μπορούν να γίνουν είναι στις παραμέτρους βελτιστοποίησης και πιο συγκεκριμένα στον αριθμό των επαναλήψεων της βελτιστοποίησης, στην αρχική και τελική τιμή του ρυθμού εκμάθησης για τη προσαρμογή της θέσης των σημείων καθώς και τα βήματα μέχρι να φτάσει σε αυτή την τελική τιμή. Ακόμα μπορούν να γίνουν αλλαγές στο ρυθμό εκμάθησης χαρακτηριστικών, διαφάνειας, κλιμάκωσης και περιστροφής των σημείων. Τέλος, μπορούμε να προσαρμόσουμε τις παραμέτρους για τη πυκνοποίηση των σημείων ορίζοντας τις επαναλήψεις στις οποίες θα γίνεται αυτή η διαδικασία καθώς και η συχνότητα με την οποία θα γίνεται αυτή η πυκνοποίηση των σημείων.

Οι συγκεκριμένες αλλαγές μπορούν να γίνουν με μια απλή αλλαγή τιμής στο αρχείο `_init_.py` στο φάκελο `arguments` του αρχικού κώδικα.

Το πιο σημαντικό όμως που παρατηρήσαμε στον κώδικα είναι πως αν και διαβάζει τα δεδομένα εισόδου όπως τα εξάγει το COLMAP ουσιαστικά δε τα χρησιμοποιεί όλα αλλά κάποια πολύ βασικά. Πιο συγκεκριμένα, όσον αφορά το αρχείο `cameras.txt` όπου μοντέλο της κάμερας και να διαβάσει το μετατρέπει σε PINHOLE για να μπορέσει να αξιοποιήσει τα δεδομένα αυτά. Στο text file για τις φωτογραφίες (`images.txt`) παρατηρήσαμε ότι τα μόνα δεδομένα που χρειάζεται είναι το `IMAGE_ID`, το quaternion, το διάνυσμα μετάφρασης που όπως αναφέρθηκε προηγουμένως προκύπτει από το quaternion και τη θέση της κάμερας, το `CAMERA_ID` που έχουμε δηλώσει στο `cameras.txt` καθώς και το όνομα της κάθε φωτογραφίας.

```
*images.txt - Notepad
File Edit Format View Help
# IMAGE_ID, QW, QX, QY, QZ, TX, TY, TZ, CAMERA_ID, NAME
1 0.9015232185173202 -0.071399717394061007 -0.42328970786259285 -0.054624079377334551 0.0027671256602392891 -1.1024391297161953 3.8377925032777762 1 _DSC8679.JPG
2 0.8701255199311824 -0.21781308399726095 -0.44165694076729878 0.019241997825016153 -0.12973488530118726 -0.03753198764997201 4.126388580721929 1 _DSC8680.JPG
3 0.83941427137115043 -0.12259116463258216 -0.52511842115348006 -0.067865537187472832 0.04441280533781411 -0.83285473050610981 3.9107029657512999 1 _DSC8681.JPG
4 0.81484019443948685 -0.26668687065041125 -0.51280152481731744 0.044138041342408847 0.22061844554820231 0.48887626887526847 4.0806730794901201 1 _DSC8682.JPG
5 0.76030529725065887 -0.091748862129231182 -0.63801203404808193 -0.080365699651082631 0.18319551290555047 -1.0051175680695514 3.7261491072026143 1 _DSC8683.JPG
6 0.72859989271206105 -0.25145408189495366 -0.63410209523008576 0.061867389335209556 0.24929464273517743 0.56212055479017842 3.9061838646823266 1 _DSC8684.JPG
7 0.66650576548738993 -0.094318385646662062 -0.73699691920148525 -0.060906878007294697 0.45815658322998409 -0.75848777695457503 3.6695723888304999 1 _DSC8685.JPG
8 0.62919801171809886 -0.21193973507102654 -0.74204298492727483 0.092529825309077214 0.39878262337672599 0.58759547753469088 3.8438653635121818 1 _DSC8686.JPG
9 0.56511996638070539 -0.072452517106545627 -0.81728257836541629 -0.086251049055494952 0.649334961118962619 -0.80758381956748593 3.6905064908815404 1 _DSC8687.JPG
10 0.54724814792048404 -0.18814938639194329 -0.81182667990793045 0.07782489824986677 0.75703390360251355 0.5027914410126223 3.8400593674647268 1 _DSC8688.JPG
11 -0.4659041009452507 0.057600753238596841 0.87768659243420788 0.0963419296566803 0.77259771217514484 -0.87473001828732166 3.722990834424774 1 _DSC8689.JPG
12 -0.43035746776633649 0.16747155266064792 0.87558426505535503 -0.14176714631624784 0.72473681803040346 0.96487394823024386 3.8213525033360343 1 _DSC8690.JPG
13 -0.31530773250216787 0.030023029472659887 0.94160284094165869 0.11429672548337817 0.59459177708214428 -0.87253107761731552 3.9092048457793522 1 _DSC8691.JPG
14 -0.32630389359142914 0.11086853167230562 0.9351609004436412 -0.08190255181213002 0.79380264937153866 0.6319447345945316 4.0541605253439403 1 _DSC8692.JPG
15 -0.20703271372161469 0.026260296315075239 0.96727649902762702 0.14428709109307952 0.53068280601117612 -1.0955045298890089 3.9660780203139026 1 _DSC8693.JPG
16 -0.20822082984379731 0.072412435806533818 0.97154622560912995 -0.086586102577343199 0.65350098733534834 0.64410321129899306 4.2213051738359511 1 _DSC8694.JPG
17 -0.037084834107334402 0.03067514425523269 0.99058974299207758 0.12812367736919267 0.2842329770780071 -0.93633990050521509 4.299888706701928 1 _DSC8695.JPG
18 0.012376188725091681 -0.0019432091688079197 0.99095945487430603 -0.13357549433180307 -0.12516750614080369 1.1580714314804972 4.3773563764158219 1 _DSC8696.JPG
19 0.15885393159012176 0.010678618346698733 0.97589071716705256 0.14929401754840116 -0.06546814852342786 -1.0344293668330884 4.310925910694728 1 _DSC8697.JPG
20 0.15391339137931304 0.00363591113617062 0.97727907802499914 -0.13549708722971951 -0.049643460948086915 1.3684972205417514 4.3832156595878153 1 _DSC8698.JPG
21 0.3045396580263875 -0.08030524033453620009 0.94296683605730958 0.13442137398920498 -0.03883841773970957696 -0.92923776004734182 4.1797943353129199 1 _DSC8699.JPG
22 0.27679084376645678 -0.088353061202691666 0.94955921324856862 -0.11797400526355434 0.08884341716234001 1.144846852198679 4.2732008453009183 1 _DSC8700.JPG
23 0.4113525235336101 -0.0095645055854898 0.90203275651557224 0.13051639261310211 0.1703270686011446 -0.9986094371328922 4.0662256134452721 1 _DSC8701.JPG
24 0.40900638145524976 -0.13218962688151278 0.899281913893615 -0.080819068403752597 0.050159995253233447 0.7630282184374968 4.1372227520514624 1 _DSC8702.JPG
25 0.52989924408653721 -0.01535818822932159 0.83433400015392678 0.15118980049698992 0.070054641803974249 -1.1413320122135786 3.7705074178634832 1 _DSC8703.JPG
26 0.505994074914492 -0.1979497843278713 0.83045985466096639 -0.12297279712338429 -0.0070704263049110594 1.2501043034048585 3.8187531108344208 1 _DSC8704.JPG
27 0.6448544774451952 -0.034757294209661546 0.75185302721250835 0.13293479186990267 0.099931985554331249 -0.9314417487171871 3.6839065562539126 1 _DSC8705.JPG
28 0.63280832080643856 -0.15532536744243605 0.75799679267558662 0.029470691212316415 0.07122936252009926 -0.20744365280442256 3.9708505634417293 1 _DSC8706.JPG
29 0.73570953830891794 -0.097381687031693412 0.66381335791779628 0.1289980029451485 0.23470775657817233 -1.0601839071172239 3.76040045902498022 1 _DSC8707.JPG
30 0.71640205858030725 -0.1973457890013484 0.6690827379743598 -0.012289010133979547 0.18870627220337921 0.254198609156195 4.02658366043021034 1 _DSC8708.JPG
31 0.83077426191335058 -0.05448962978316322 0.54686737536195473 0.088210428823061376 0.17140223452582948 -0.88954638290957055 3.856943667950709 1 _DSC8709.JPG
32 0.80750826901120931 -0.02690007331114973 0.55223378271946943 0.01262797757850969 0.18282976974195586 0.029702983406468819 4.2245422684092651 1 _DSC8710.JPG
33 0.90163913426515385 -0.02949071244869106 0.4229480355733477 0.08539961412527614 0.12361639412341874 -1.2476783800809043 3.9243586143364286 1 _DSC8711.JPG
34 0.88177900036380141 -0.2330643647678274 0.40939476768023808 0.011063275840687075 -0.0078046779997050764 0.000467100066827785 4.3767881727121658 1 _DSC8712.JPG
```

Εικόνα 28:Απαραίτητα δεδομένα εισόδου στο images.txt

Τέλος, όσον αφορά το αρχείο με τα δεδομένα για τα σημεία στο τριδιάστατο χώρο (points3D.txt) τα δεδομένα που είναι απαραίτητα για το πρόγραμμα είναι το POINT\_ID , η θέση του κάθε σημείου και το χρώμα.

```
*points3D.txt - Notepad
File Edit Format View Help
# POINT3D_ID, X, Y, Z, R, G, B
27590 4.7790641907029965 2.7006307887502494 -4.6488504020408525 91 79 67
63081 0.27807352339462266 0.50790126975472483 -0.015576209773846036 0 0 0
62872 -7.32049581020917255 -3.424292839197705 -2.7832476714903107 63 65 62
63145 -0.029486038009196786 0.31050968482027236 0.66357518750679034 72 72 57
18418 -0.62326684820011025 1.2021453432215947 2.0188102419474037 101 102 50
62936 6.4718365033319838 -4.2807684546430043 -29.016674168855932 254 255 255
16451 3.4265366348155766 1.1556021873950009 1.7962551005011098 103 93 84
62270 13.287746513786617 -1.3866383866854179 2.3465088046481415 50 54 31
60705 0.16618305975890271 0.82927244422884461 0.19309864614180841 173 181 190
7274 -3.0122487525410606 0.62331454380778706 4.0557734431366699 75 86 26
62403 -7.9063640891974583 -2.0802358865589943 3.5564980503673205 28 28 18
61008 0.2629628187660471 0.55843282504801706 -0.038216972825725171 112 114 113
63080 0.27076601889247165 0.50331316900619327 -0.0097759366260570125 63 68 44
63144 -0.03046868154027604 0.31027964850161022 0.66453462197312774 71 66 56
62938 12.718585386391389 0.19429476124464462 -17.0897220260071442 42 47 24
62252 -0.055517433580203591 0.45570393053033603 0.7053536113041182 221 221 219
59854 15.901993430224438 -2.2666127286469147 -7.5091882376671037 37 47 44
63071 19.419954975222375 2.3092697470936132 -20.673913776856789 22 25 8
62316 6.8960043994662596 0.9814483714337271 1.9846796967315599 43 45 24
22725 5.6381535062161374 1.3291703019100041 -1.5472312703199644 64 66 54
63135 6.5164574083180398 -1.4031513520929417 -8.7417598303705102 27 30 21
61950 0.28012719731881386 0.87021522672622831 1.6764609312196161 147 140 130
27644 2.3868280883026531 1.5200117166065987 0.53354206111484781 202 196 174
63079 11.5095789273473675 -5.1658865801488902 3.0124186212551929 253 255 254
63143 0.0049088766574010992 0.146590043447842698 0.72227610460604907 252 254 253
18408 3.63721229292313324 1.1067664968314239 1.9353308637080038 58 52 48
17529 6.4786662921466149 -0.15841703115089401 5.1159025388643782 16 17 8
62260 5.550325461775317 0.49319137203138746 0.61899157424669349 50 53 35
```

Εικόνα 29:Απαραίτητα δεδομένα εισόδου στο points3D.txt

Αξίζει να σημειωθεί πως για να τρέξει ο κώδικας και να μην βγάλει σφάλμα πρέπει να γίνουν προσαρμογές στο `colmap_loader.py` στο φάκελο `scene` του αρχικού κώδικα ώστε να διαβάζονται μόνο τα απαραίτητα δεδομένα εισόδου στα προσαρμοσμένα αρχεία που θα δοθούν από εμάς. Οι προσαρμογές αυτές είναι πολύ απλές και αφορούν απλώς την αλλαγή στα πόσα δεδομένα διαβάζει κάθε φορά ο κώδικας. Για παράδειγμα, στα δεδομένα εισόδου για τις φωτογραφίες αφαιρούμε τις γραμμές του κώδικα που αφορούν την ανάγνωση δεδομένων πέρα από τα απαραίτητα που αναφέραμε.

## 5.4 Gaussian Splatting χωρίς τη χρήση του COLMAP

Αξιοποιώντας, λοιπόν, την παραπάνω δυνατότητα που μας προσφέρει ο κώδικας του Gaussian Splatting, δηλαδή το να μη χρειαζόμαστε τόσα πολλά δεδομένα όσα παράγει το COLMAP αλλά πολύ λιγότερα αφού αυτά τα λίγα είναι που πραγματικά χρειάζεται τελικά ο αλγόριθμος για να τρέξει, μπορούμε να μη χρησιμοποιούμε καθόλου το COLMAP σε ορισμένες περιπτώσεις. Μια από αυτές τις περιπτώσεις στην οποία θα εστιάσουμε σε αυτή τη διπλωματική εργασία είναι όταν ο χώρος ο οποίος θέλουμε να αναπαρασταθεί με τη μέθοδο Gaussian Splatting έχει σκαναριστεί τρισδιάστατα με τα κατάλληλα εργαλεία. Το αρχείο που προκύπτει από ένα 3D-SCAN μπορεί να έχει διάφορες μορφές αλλά κατά βάση περιέχει τα δεδομένα που χρειαζόμαστε για να φτιάξουμε τα τρία αρχεία (`points3D.txt`, `images.txt`, `cameras.txt`) που χρειάζεται ο αλγόριθμος για να τρέξει. Τα δεδομένα που περιλαμβάνει το αρχείο του 3D-SCAN μπορούμε να τα δούμε, να τα μελετήσουμε και να τα επεξεργαστούμε με διάφορες εφαρμογές. Εμείς επιλέγουμε το CloudCompare το οποίο είναι ένα δωρεάν λογισμικό που μας επιτρέπει να ανοίξουμε, να επεξεργασθούμε και να εξάγουμε δεδομένα από αρχεία που προέκυψαν από 3D-SCAN.

### 5.4.1 CloudCompare

Το CloudCompare είναι ένα ευέλικτο, ανοιχτού κώδικα λογισμικό σχεδιασμένο για την επεξεργασία και ανάλυση 3D νεφών σημείων (point cloud) και πλεγμάτων (meshes). Αρχικά αναπτύχθηκε για τη σύγκριση 3D νεφών σημείων που προέρχονται από διαφορετικές πηγές, αλλά έχει εξελιχθεί σε ένα ολοκληρωμένο εργαλείο που χρησιμοποιείται ευρέως σε διάφορους τομείς, όπως οι γεωεπιστήμες, η μηχανική και η αρχιτεκτονική.

Το CloudCompare υποστηρίζει διάφορες μορφές αρχείων νέφους σημείων, όπως LAS, LAZ, PLY, OBJ, ASC και E57, επιτρέποντας στους χρήστες να εισάγουν και να εξάγουν δεδομένα απρόσκοπτα. Οι χρήστες μπορούν να εφαρμόσουν διάφορα φίλτρα για τον καθαρισμό και τη βελτιστοποίηση των νεφών σημείων, συμπεριλαμβανομένης της μείωσης θορύβου, της αφαίρεσης ακραίων τιμών και της υποδειγματοληψίας. Το

λογισμικό ακόμα προσφέρει εργαλεία για την τμηματοποίηση των νεφών σημείων σε μικρότερα, διαχειρίσιμα μέρη για λεπτομερή ανάλυση ή συγκεκριμένες εργασίες. Το CloudCompare παρέχει ισχυρούς αλγόριθμους για την ευθυγράμμιση νεφών σημείων από διαφορετικά σύνολα δεδομένων ή πηγές, κάτι που είναι κρίσιμο για εφαρμογές όπως η 3D ανακατασκευή και η παρακολούθηση αλλαγών με την πάροδο του χρόνου.

Οι χρήστες μπορούν να δημιουργούν 3D πλέγματα από νέφη σημείων χρησιμοποιώντας διάφορους αλγόριθμους πλέξης. Αυτό είναι απαραίτητο για τη δημιουργία μοντέλων επιφάνειας και οπτικοποιήσεων. Οι χρήστες μπορούν να δημιουργούν 3D πλέγματα από νέφη σημείων χρησιμοποιώντας διάφορους αλγόριθμους πλέξης. Αυτό είναι απαραίτητο για τη δημιουργία μοντέλων επιφάνειας και οπτικοποιήσεων. Μία από τις βασικές λειτουργίες είναι η σύγκριση πλεγμάτων ή νεφών σημείων για την ανίχνευση διαφορών, παραμορφώσεων ή αλλαγών με την πάροδο του χρόνου. Αυτό είναι ιδιαίτερα χρήσιμο στον ποιοτικό έλεγχο και την παρακολούθηση δομικών αλλαγών.

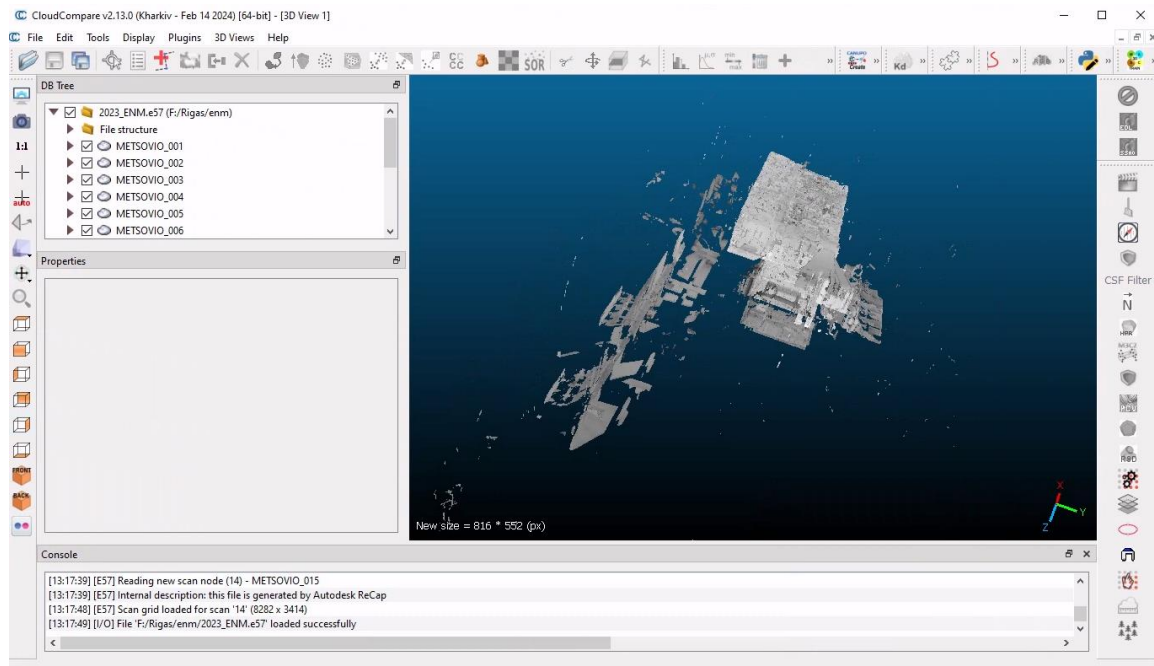
Επιπρόσθετα, το CloudCompare μπορεί να μετρά αποστάσεις μεταξύ σημείων, επιφανειών και όγκων με υψηλή ακρίβεια, κάτι που είναι τεράστιας σημασίας σε εφαρμογές που έχουν να κάνουν με τοπογραφία. Το λογισμικό περιλαμβάνει επιπλέον εργαλεία για τη διενέργεια στατιστικών αναλύσεων σε νέφη σημείων και πλέγματα, όπως η καταμέτρηση ιστογραμμάτων, η ανάλυση πυκνότητας και η εξαγωγή χαρακτηριστικών. Το CloudCompare διαπρέπει στην οπτικοποίηση μεγάλων νεφών σημείων και πλεγμάτων. Οι χρήστες μπορούν να χρωματίζουν δεδομένα βάσει πεδίων κλίμακας (π.χ., ένταση, ανύψωση), να εφαρμόζουν σκιάσεις και να δημιουργούν προσαρμοσμένες απόψεις για παρουσιάσεις και αναφορές. Τέλος, για προχωρημένους χρήστες, το CloudCompare προσφέρει μια διεπαφή γραμμής εντολών και δυνατότητες δημιουργίας σεναρίων για την αυτοματοποίηση επαναλαμβανόμενων εργασιών και την επεξεργασία μεγάλων συνόλων δεδομένων μαζικά.

Με λίγα λόγια τα βασικά πλεονεκτήματα του είναι ότι πρώτον, ως λογισμικό ανοιχτού κώδικα, το CloudCompare είναι διαθέσιμο δωρεάν και αναβαθμίζεται συνεχώς από μια κοινότητα προγραμματιστών και χρηστών, δεύτερον, υποστηρίζει μια ευρεία γκάμα μορφών αρχείων και παρέχει εργαλεία για διάφορους τύπους αναλύσεων, καθιστώντας το κατάλληλο για διαφορετικές εφαρμογές και τρίτον υπάρχει μια ισχυρή κοινότητα χρηστών, η οποία βοηθά τους νέους χρήστες να ξεκινήσουν και να βρουν λύσεις σε συγκεκριμένα προβλήματα.

Το CloudCompare ξεχωρίζει ως ένα ισχυρό και ευέλικτο εργαλείο για οποιονδήποτε εργάζεται με 3D νέφη σημείων και πλέγματα. Η ευρεία γκάμα λειτουργιών του, σε συνδυασμό με τον ανοιχτό του χαρακτήρα, το καθιστούν έναν απαραίτητο πόρο σε τομείς που απαιτούν ακριβή και λεπτομερή 3D ανάλυση. Είτε ασχολείστε με γεωχωρικές μελέτες, μηχανική, διατήρηση πολιτιστικής κληρονομιάς, είτε κάποια άλλη επιστήμη, το CloudCompare παρέχει τα εργαλεία που χρειάζεστε για να εργαστείτε αποτελεσματικά με δεδομένα 3D.



Ας δούμε λοιπόν πως είναι το interface του CloudCompare όταν με αυτό έχουμε ανοίξει ένα .e57 αρχείο που περιέχει σκαναρισμένο το χώρο του εργαστήριου ναυτικής μηχανολογίας του ΕΜΠ.



Εικόνα 30: Interface CloudCompare

## 5.4.2 Δημιουργία Δεδομένων για 3D αναπαράσταση με Gaussian Splatting

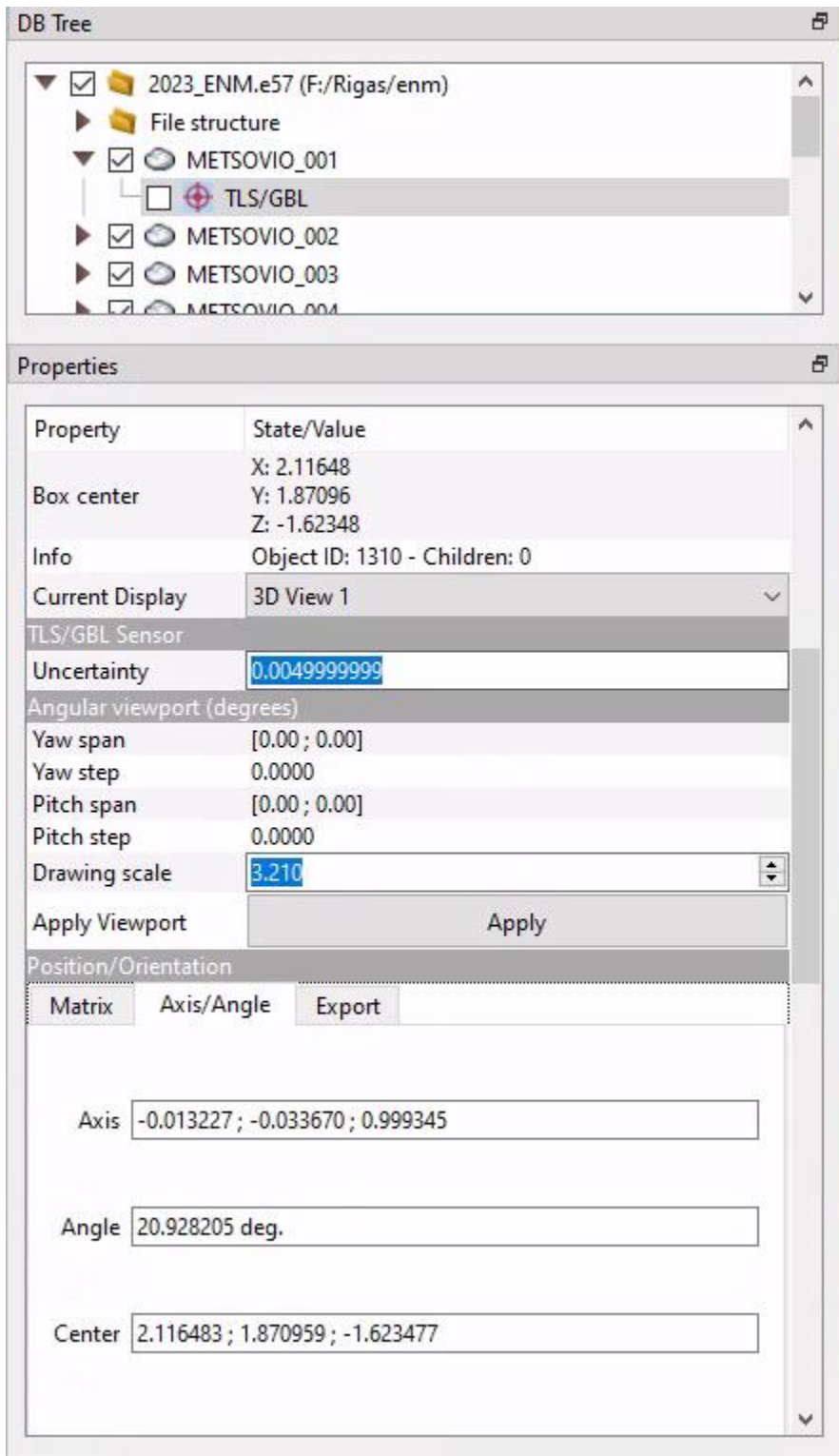
Ας δούμε λοιπόν τώρα για αρχή τα δεδομένα που χειριζόμαστε από το αρχείο που έχει προκύψει από το 3D-scan για να δημιουργήσουμε τα τρία αρχεία txt, που χρειάζονται εν τέλει για να τρέξει ο αλγόριθμος που θα κάνει τη τρισδιάστατη αναπαράσταση με Gaussian Splatting. Καταρχάς αξίζει να αναφερθεί ότι το τελικό αρχείο περιέχει δεδομένα από 3D-scans στο ίδιο χώρο σε διαφορετικά σημεία. Ξεκινώντας την άντληση των απαραίτητων δεδομένων για αρχή θα κάνουμε ένα subsample στο σύνολο των σημείων από όλα τα 3D-scans και έπειτα θα τα εξαγωγήμε σε ένα txt. Συνεχίζοντας, από κάθε 3D-scan θα εξάγουμε τις φωτογραφίες που τράβηξε η κάμερα στο συγκεκριμένο σημείο που θα είναι είτε 6 φωτογραφίες είτε μια φωτογραφία 360 μοιρών. Επιπλέον από κάθε 3D-scan θα πάρουμε την τοποθεσία της κάμερας καθώς και τους άξονες και τη γωνιά TLS. Τι είναι όμως οι άξονες και η γωνιά TLS όπως τα ονομάζει το CloudCompare;

Οι "TLS άξονες" αναφέρονται στους κύριους άξονες που προκύπτουν από ένα σύνολο δεδομένων από «Terrestrial Laser Scanning – TLS». Το TLS είναι μια μέθοδος καταγραφής 3D δεδομένων χρησιμοποιώντας τεχνολογία λέιζερ σάρωσης. Οι άξονες αυτοί

αντιπροσωπεύουν τον προσανατολισμό των δεδομένων σάρωσης στο τρισδιάστατο χώρο. Στο πλαίσιο του CloudCompare, οι άξονες αυτοί χρησιμοποιούνται για την κατανόηση και τον χειρισμό του χωρικού προσανατολισμού του νέφους σημείων. Οι άξονες X, Y, Z είναι οι τυπικοί άξονες καρτεσιανών συντεταγμένων που χρησιμοποιούνται για την αναπαράσταση του τρισδιάστατου χώρου. Στην περίπτωση ενός συνόλου δεδομένων TLS, αυτοί οι άξονες είναι ευθυγραμμισμένοι βάσει της θέσης και του προσανατολισμού του σαρωτή κατά τη διαδικασία συλλογής των δεδομένων. Οι "TLS γωνίες" αναφέρονται στις γωνίες περιστροφής γύρω από τους άξονες TLS που απαιτούνται για την ευθυγράμμιση ή τη μετατροπή του νέφους σημείων. Αυτές οι γωνίες χρησιμοποιούνται συχνά για την προσαρμογή του προσανατολισμού του νέφους σημείων ώστε να ταιριάζει σε ένα επιθυμητό σύστημα αναφοράς ή για τη διόρθωση αστοχιών στην ευθυγράμμιση των δεδομένων.

Στο CloudCompare, αυτές οι έννοιες χρησιμοποιούνται σε διάφορες λειτουργίες όπως στην προσαρμογή του προσανατολισμού ενός νέφους σημείων για να ταιριάζει με ένα άλλο ή να ενσωματωθεί σε ένα συγκεκριμένο σύστημα συντεταγμένων, στην εφαρμογή συγκεκριμένων περιστροφών και μετατοπίσεων σε νέφη σημείων για να επιτευχθεί η επιθυμητή χωρική διαμόρφωση αλλά και για την κατανόηση του προσανατολισμού και της θέσης των δεδομένων σάρωσης σε σχέση με τη θέση του σαρωτή και το περιβάλλον. Οι χρήστες δηλαδή μπορούν να εφαρμόσουν μετασχηματισμούς χρησιμοποιώντας TLS γωνίες για να διορθώσουν ή να προσαρμόσουν τον προσανατολισμό του νέφους σημείων. Αυτό είναι ιδιαίτερα χρήσιμο σε σενάρια όπου τα δεδομένα νέφους σημείων πρέπει να συγκριθούν ή να συγχωνευθούν με άλλα σύνολα δεδομένων. Εξετάζοντας τους TLS άξονες ,λοιπόν , οι χρήστες μπορούν να αποκτήσουν πληροφορίες για τη γεωμετρία της σάρωσης και τις χωρικές σχέσεις μέσα στα καταγεγραμμένα δεδομένα.

Στο CloudCompare, μπορούμε να χειριστούμε τους TLS άξονες και τις γωνίες μέσω των διάφορων εργαλείων μετασχηματισμού που παρέχονται στο λογισμικό. Αυτό περιλαμβάνει τη χρήση διαλόγων για την εισαγωγή χειροκίνητων τιμών περιστροφής και μετατόπισης ή τη χρήση αυτοματοποιημένων χαρακτηριστικών ευθυγράμμισης που υπολογίζουν τους απαραίτητους μετασχηματισμούς. Η κατανόηση αυτών των εννοιών είναι κρίσιμη για την αποτελεσματική επεξεργασία και ανάλυση των δεδομένων TLS στο CloudCompare, διασφαλίζοντας ακριβή αναπαράσταση και ευθυγράμμιση των δεδομένων νέφους σημείων.

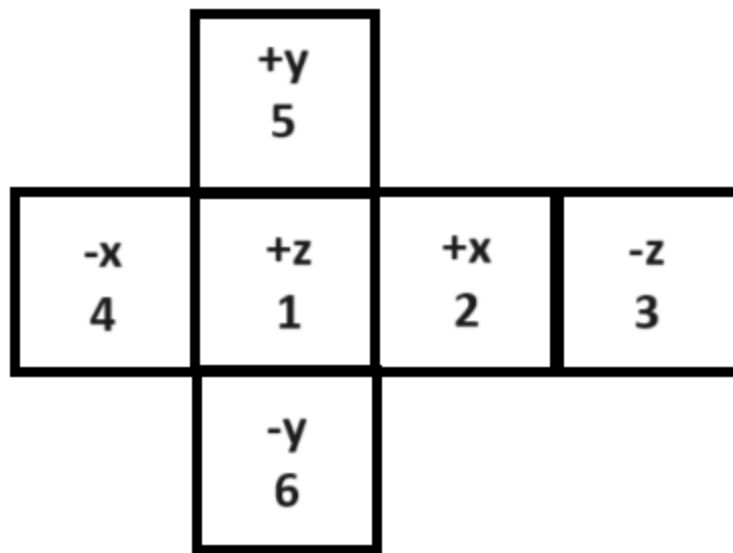


Εικόνα 31: TLS Άξονες και TLS Γωνία για ένα 3D-scan στο CloudCompare



Αφού λοιπόν εξηγήσαμε τι είναι οι λεγόμενοι άξονες TLS και γωνία TLS καθώς και ποια δεδομένα πρέπει να εξάγουμε από το αρχείο που προέκυψε από το 3D-scan θα πρέπει να δημιουργήσουμε έναν κώδικα που θα αξιοποιεί αυτά τα δεδομένα για να ετοιμάσει τελικά το αρχείο εισόδου για να πραγματοποιηθεί τρισδιάστατη αναπαράσταση με τη μέθοδο Gaussian Splatting. Το τελικό αρχείο εισόδου θα πρέπει να είναι ένας φάκελος έστω «input\_final» όπως εμείς τον ονομάσαμε με έναν υποφάκελο με όνομα «images» ,που θα περιέχει όλες τις φωτογραφίες με συγκεκριμένο όνομα η καθεμιά το οποίο θα καθορίσουμε εμείς , και έναν υποφάκελο με όνομα «sparse» που να περιέχει ένα φάκελο με όνομα «0» που θα περιέχει τα τρία αρχεία κειμένου cameras.txt , images.txt, points3D.txt τα οποία θα δημιουργήσουμε εμείς μέσα από τον κώδικα αυτόν.

Καταρχάς ας υποθέσουμε πως από κάθε επιμέρους 3D-scan έχουμε τις 6 φωτογραφίες στο χώρο (πάνω ,κάτω ,δεξιά ,αριστερά, μπροστά και πίσω) τις οποίες όταν τις εξάγουμε τους δίνουμε τον ανάλογο όνομα και τις τοποθετούμε σε ξεχωριστό φάκελο ανάλογα με το επιμέρους scan. Για παράδειγμα στη περίπτωση μας τις αποθηκεύουμε σε φάκελο με όνομα «input\_init\_er1» , 2 , 3 κ.λπ. και με ονόματα στην κάθε φωτογραφία από τον αριθμό 1 έως 6 ανάλογα τη θέση στο cubemap όπως φαίνεται παρακάτω :



Εικόνα 32: Αντιστοίχιση ονόματος φωτογραφίας με θέση στο cubemap

Ο κώδικας μας αξιοποιεί αυτές τις φωτογραφίες και αφού δημιουργήσει τη φωτογραφία cubemap με αυτήν δημιουργεί τη φωτογραφία 360 μοιρών. Οπότε πλέον είμαστε σχεδόν έτοιμοι να δημιουργήσουμε τον φάκελο images. Για να γίνει αυτό θα πρέπει από κάθε φωτογραφία 360 μοιρών να εξάγουμε πολλές φωτογραφίες θεωρώντας πως στεκόμαστε στη μέση και κοιτάμε δεξιόστροφα και από κάτω προς τα πάνω με σταθερό βήμα. Ορίζουμε την επιθυμητή ανάλυση της εικόνας στα 1024x1024, το επιθυμητό οπτικό πεδίο (field of view) 90 μοίρες επί 90 μοίρες και το βήμα από την δεξιόστροφη περιστροφή 30 μοιρών ενώ στην περιστροφή από κάτω προς τα πάνω βήμα

45 μοιρών. Τις φωτογραφίες που εξάγονται τις ονομάζουμε ανάλογα με τον αριθμό του 3D-scan και ανάλογα με τις γωνίες περιστροφής στους δύο άξονες. Οι φωτογραφίες αυτές αποθηκεύονται και ξεχωριστά ανάλογα με το εκάστοτε 3D-scan σε ξεχωριστό φάκελο, στο παράδειγμα μας ονομάζονται «input\_er1» ,2 ,3 κ.λ.π. και όλες μαζί δηλαδή από όλα τα επιμέρους 3D-scan στο φάκελο «images».

Συνεχίζοντας δημιουργούμε το αρχείο points3D.txt διατηρώντας από το αρχείο που έχουμε εξάγει με τις θέσεις των σημείων (αφού έχουμε κάνει μείωση του δείγματος των σημείων όπως είχαμε αναφέρει) από το CloudCompare ,μόνο τις έξι πρώτες στήλες που αφορούν τη θέση και το χρώμα των σημείων ,προσθέτουμε όμως μια πρώτη στήλη με αύξων αριθμό ξεκινώντας από το 1 που αποτελεί ταυτότητα του κάθε σημείου.

Όσον αφορά τη δημιουργία του αρχείου cameras.txt θεωρούμε ένα μοντέλο κάμερας για όλες τις φωτογραφίες αυτό της pinhole κάμερας και ορίζουμε τις παραμέτρους της με βάση την ανάλυση της φωτογραφίας που έχουμε δώσει προηγουμένως στην περίπτωση μας δηλαδή 1024x1024 οπότε οι παράμετροι είναι «height, width, focal length x, focal length y, principal point x, principal point y» .Πριν από όλα αυτά θέτουμε και το camera\_id που θεωρητικά είναι ένας αύξων αριθμός αλλά στη περίπτωση μας υπάρχει ένα μόνο μοντέλο κάμεράς οπότε τον θεωρούμε 1.

Η εστιακή απόσταση (focal length) είναι η απόσταση από το κέντρο του φακού μέχρι το σημείο όπου οι παράλληλες ακτίνες φωτός συγκλίνουν σε μια εστίαση. Μετριέται σε χιλιοστά (mm) και επηρεάζει το πόσο "κοντά" ή "μακριά" φαίνεται να είναι το αντικείμενο στη φωτογραφία. Μια μεγαλύτερη εστιακή απόσταση σημαίνει μεγαλύτερη μεγέθυνση και στενότερο οπτικό πεδίο, ενώ μια μικρότερη εστιακή απόσταση σημαίνει μικρότερη μεγέθυνση και ευρύτερο οπτικό πεδίο. Στη περίπτωση μας ,όμως, η εστιακή απόσταση μετριέται σε pixels. Η εστιακή απόσταση σε pixels είναι μια σημαντική παράμετρος για την ανάλυση εικόνας και την υπολογιστική όραση . Υπολογίζεται με βάση την πραγματική εστιακή απόσταση του φακού (σε χιλιοστά) και τις φυσικές διαστάσεις του αισθητήρα της κάμερας (επίσης σε χιλιοστά). Ο τύπος για τον υπολογισμό της εστιακής απόστασης σε pixels είναι:

$$f_{pixels} = \frac{f_{mm} * width_{pixels}}{width_{sensor\ mm}}$$

Εμείς όμως επειδή δε γνωρίζουμε το λόγο  $\frac{f_{mm}}{width_{sensor\ mm}}$  χρησιμοποιούμε έναν άλλο τύπο που αξιοποιεί την τιμή του οπτικού πεδίου(field of view,FOV) ,που στη περίπτωση μας είναι 90 μοίρες στο οριζόντιο και 90 μοίρες στο κατακόρυφο , ο οποίος είναι ο εξής :

$$f_{pixels} = \frac{width_{pixels} * 0.5}{\tan(FOV * 0.5)}$$

Οπότε, στη περίπτωση μας έχουμε  $f_x=f_y=512$

Το κύριο σημείο (principal point) είναι η θέση στο επίπεδο της εικόνας όπου ο οπτικός άξονας διασχίζει αυτό το επίπεδο. Συνήθως βρίσκεται κοντά στο κέντρο του αισθητήρα της κάμερας, αλλά μπορεί να αποκλίνει ανάλογα με το σχεδιασμό του φακού και της κάμερας. Για μια κάμερα με αισθητήρα 1024x1024 pixels, το κύριο σημείο είναι συνήθως κοντά στο κέντρο, δηλαδή γύρω από το σημείο (512, 512) σε pixels.

Πλέον, λοιπόν, το μόνο αρχείο που έμεινε να δημιουργήσουμε με τον κώδικα αυτό προετοιμασίας δεδομένων για να γίνει τρισδιάστατη αναπαράστασή με τη μέθοδο Gaussian Splatting είναι το images.txt. Για να δημιουργήσουμε αυτό το αρχείο χρειαζόμαστε σε κάθε σειρά δηλαδή για κάθε φωτογραφία το image\_id που είναι ένας αύξων αριθμός ξεκινώντας από το 1, το quaternion (Qw, Qx, Qy, Qz), το translation vector (Tx, Ty, Tz), το μοντέλο της κάμερας, το οποίο είναι σταθερό και ίδιο στη περίπτωση μας οπότε είναι το camera\_id που έχουμε δηλώσει στο cameras.txt, και τέλος το όνομα της φωτογραφίας. Το πρόβλημα στην δημιουργία του αρχείου αυτού έγκειται στον υπολογισμό του quaternion αρχικά και έπειτα του translation vector. Για να δημιουργήσουμε το quaternion για κάθε φωτογραφία θα χρειαστεί να δημιουργήσουμε πολλά quaternions για κάθε επιμέρους περιστροφή και έπειτα να τα πολλαπλασιάσουμε μεταξύ τους και με τη σωστή σειρά για το τελικό quaternion. Καταρχάς, επειδή όταν εμείς κοιτάμε ευθεία θεωρούμε πως οι μοίρες της εκάστοτε φωτογραφίας είναι 0 ενώ στο CloudCompare παρατηρούμε πως για 0 μοίρες κοιτάμε κάτω χρειαζόμαστε αρχικά ένα quaternion που θα μας περιστρέφει κατά 90 μοίρες ως προς τον x προς τα πάνω ( $q_x = \text{quaternion.from\_rotation\_vector}(\text{degrees\_to\_radians}(-90) * \text{np.array}([1, 0, 0]))$ ). Έπειτα, χρειαζόμαστε ένα quaternion που θα μας φέρει στην αρχική θέση της κάμερας και θα προκύπτει από τους άξονες και τη γωνιά TLS, τα οποία έχουμε εξάγει σε ξεχωριστά αρχεία κειμένου txt για να μπορεί ο κώδικας να τραβάει τα δεδομένα, το οποίο quaternion ονομάζουμε quaternion\_dir. Συνεχίζοντας, χρειαζόμαστε να εισάγουμε την περιστροφή κατά τον z άξονα ανάλογα με τη οριζόντια γωνιά της κάθε φωτογραφίας, όμως επειδή εμείς δίνουμε τις γωνίες θεωρώντας δεξιόστροφο σύστημα ενώ είναι αριστερόστροφο στην ουσία θα θέσουμε στους άξονες στον z -1. ( $q\_yaw = \text{quaternion.from\_rotation\_vector}(\text{horizontal\_angle} * \text{np.array}([0, 0, -1]))$ ). Όσον αφορά τη κατακόρυφη περιστροφή αυτή πρέπει να γίνεται κάθε φορά ως προς τους αρχικούς άξονες πριν δηλαδή εφαρμόσουμε τις υπόλοιπες περιστροφές. ( $q\_pitch = \text{quaternion.from\_rotation\_vector}(\text{vertical\_angle} * \text{np.array}([\text{math.cos}(\text{horizontal\_angle} + / - \text{tls\_angles}[\text{folder\_index}], \text{math.sin}(\text{horizontal\_angle} + / - \text{tls\_angles}[\text{folder\_index}]), 0]))$ ). Το τελικό quaternion λοιπόν είναι το adjusted\_quaternion\_dir =  $q_x * \text{quaternion\_dir} * q\_yaw * q\_pitch$ , πολλαπλασιασμός που σέβεται τη σύνθεση του τελικού quaternion.

Τέλος, για τον υπολογισμό του translation vector χρησιμοποιούμε τον τύπο από το COLMAP σύμφωνα με τον οποίο οι συντεταγμένες της προβολής/κέντρου της κάμερας, (τις οποίες όμως εμείς έχουμε από το αρχείο του 3D-scan και τις έχουμε εξάγει σε ένα αρχείο κειμένου για να τις τραβάει από εκεί ο κώδικας κατευθείαν) δίνονται από το  $-R^t$

\*  $T$ , όπου  $R^t$  είναι ο αντίστροφος του  $3 \times 3$  πίνακα περιστροφής που σχηματίζεται από το quaternion και το  $T$  είναι το διάνυσμα μετάφρασης.

Ο κώδικας αυτός για να τρέξει μπορεί να χρειαστεί από λίγα λεπτά μέχρι το πολύ καμία ώρα αν δοθούν υπερβολικά πολλά δεδομένα τα οποία θα έχει να προετοιμάσει. Πάντως κατά βάση είναι μια γρήγορη διαδικασία.

Στο συγκεκριμένο κώδικα σε γλώσσα προγραμματισμού Python που φτιάξαμε για να προετοιμάσουμε τα δεδομένα για να τρέξουμε τον προσαρμοσμένο κώδικα που θα πραγματοποιήσει τελικά την 3D αναπαράσταση με Gaussian Splatting χρησιμοποιούμε διάφορες βιβλιοθήκες της python τις οποίες θα αναλύσουμε συνοπτικά ευθύς αμέσως. Τι είναι όμως οι βιβλιοθήκες της Python;

Οι βιβλιοθήκες Python είναι συλλογές προγραμμάτων γραμμένων εκ των προτέρων που παρέχουν ένα ευρύ φάσμα λειτουργικότητας, επιτρέποντας στους προγραμματιστές να εκτελούν αποδοτικά διάφορες εργασίες χωρίς να χρειάζεται να γράφουν κώδικα από την αρχή. Αυτές οι βιβλιοθήκες καλύπτουν διάφορους τομείς όπως ανάλυση δεδομένων, μηχανική μάθηση, ανάπτυξη ιστοσελίδων, επιστημονική υπολογιστική, επεξεργασία εικόνας και πολλά άλλα. Δημοφιλείς βιβλιοθήκες περιλαμβάνουν τις «numpy» για αριθμητικούς υπολογισμούς, pandas για διαχείριση δεδομένων, «scikit-learn» για μηχανική μάθηση, «matplotlib» για οπτικοποίηση δεδομένων, «requests» για αιτήσεις HTTP και «beautifulsoup4» για εξαγωγή δεδομένων από ιστοσελίδες. Με την αξιοποίηση αυτών των βιβλιοθηκών, οι προγραμματιστές μπορούν να επιταχύνουν την ανάπτυξη, να εξασφαλίσουν την αξιοπιστία του κώδικα και να εστιάσουν σε πιο υψηλού επιπέδου πτυχές των έργων τους. Το εκτενές οικοσύστημα βιβλιοθηκών της Python είναι ένας από τους κύριους λόγους για την ευρεία αποδοχή και ευελιξία της σε διάφορες βιομηχανίες και εφαρμογές.

Στο κώδικα μας χρησιμοποιούμε τις εξής βιβλιοθήκες : «os», «math» , «numpy» , «cv2» , «shutil» , «py360convert» και «quaternion»

Καταρχάς, η «os» βιβλιοθήκη στην Python προσφέρει μια περιεκτική διεπαφή για την αλληλεπίδραση με το λειτουργικό σύστημα, επιτρέποντας στους προγραμματιστές να εκτελούν μια ποικιλία εργασιών που σχετίζονται με τη διαχείριση αρχείων και καταλόγων, τη διαχείριση διεργασιών και την αλλαγή μεταβλητών περιβάλλοντος. Επιτρέπει τη δημιουργία, τη διαγραφή και την πλοήγηση σε καταλόγους, καθώς και λειτουργίες σε αρχεία, όπως μετονομασία και διαγραφή. Επιπλέον, διευκολύνει την εκτέλεση εντολών συστήματος, την ανάκτηση πληροφοριών συστήματος και τη διαχείριση μεταβλητών περιβάλλοντος. Η βιβλιοθήκη αυτή παρέχει συναρτήσεις για την ανάκτηση του τρέχοντος καταλόγου εργασίας, την αλλαγή του και τη συλλογή πληροφοριών σχετικά με την τρέχουσα διεργασία. Με τις εκτεταμένες δυνατότητές του, η βιβλιοθήκη «os» είναι απαραίτητη για τη σύνταξη σεναρίων και εφαρμογών που χρειάζονται να αλληλεπιδρούν με το υποκείμενο λειτουργικό σύστημα με έναν γρήγορο και αξιόπιστο τρόπο.

Συνεχίζοντας, η «math» βιβλιοθήκη στην Python προσφέρει μια ευρεία γκάμα μαθηματικών συναρτήσεων και σταθερών, παρέχοντας βασικά εργαλεία για την εκτέλεση μαθηματικών υπολογισμών. Περιλαμβάνει βασικές αριθμητικές λειτουργίες όπως η sqrt για την τετραγωνική ρίζα, η factorial για τον υπολογισμό παραγοντικών, και η gcd για τον μεγαλύτερο κοινό διαιρέτη, καθώς και πιο προηγμένες συναρτήσεις όπως οι τριγωνομετρικές συναρτήσεις (π.χ., sin, cos, tan), οι λογαριθμικές συναρτήσεις (π.χ., log, log10), και οι εκθετικές συναρτήσεις (π.χ., exp). Επιπλέον, η βιβλιοθήκη αυτή παρέχει σταθερές όπως το  $\pi$  (pi) και το e. Προσφέροντας ένα εκτενές σύνολο μαθηματικών λειτουργιών, η βιβλιοθήκη «math» είναι ζωτικής σημασίας για επιστημονικούς υπολογισμούς, μηχανική, ανάλυση δεδομένων, και οποιαδήποτε εφαρμογή που απαιτεί ακριβείς και αποτελεσματικούς μαθηματικούς υπολογισμούς.

Η βιβλιοθήκη «numpy» στην Python είναι ένα θεμελιώδες πακέτο για επιστημονικούς υπολογισμούς, παρέχοντας υποστήριξη για μεγάλες, πολυδιάστατες σειρές και πίνακες, μαζί με μια συλλογή μαθηματικών συναρτήσεων οι οποίες λειτουργούν αποτελεσματικά σε αυτές τις σειρές και πίνακες. Προσφέρει αξιόπιστα εργαλεία για αριθμητικούς υπολογισμούς, συμπεριλαμβανομένων βασικών λειτουργιών όπως η πρόσθεση και ο πολλαπλασιασμός, καθώς και πιο πολύπλοκες λειτουργίες όπως η γραμμική άλγεβρα, οι μετασχηματισμοί Fourier, και η γεννήτρια τυχαίων αριθμών. Επιπλέον, το numpy περιλαμβάνει δυνατότητες για την ενσωμάτωση κώδικα C/C++ και Fortran, καθιστώντας το ιδιαίτερα ευέλικτο για εφαρμογές που απαιτούν μεγάλη απόδοση. Η εκτεταμένη λειτουργικότητα του καθιστά τη «numpy» ένα απαραίτητο εργαλείο για ανάλυση δεδομένων, μηχανική μάθηση και οποιοδήποτε επιστημονικό πεδίο που απαιτεί αποτελεσματικούς αριθμητικούς υπολογισμούς.

Ας περάσουμε τώρα στη βιβλιοθήκη «cv2» στην Python, που ανήκει ουσιαστικά στη μεγάλη βιβλιοθήκη OpenCV, είναι ένα απαραίτητο εργαλείο για εργασίες υπολογιστικής όρασης, παρέχοντας μια ευρεία γκάμα λειτουργιών για την επεξεργασία και ανάλυση εικόνων. Επιτρέπει τη φόρτωση, την επεξεργασία και την αποθήκευση εικόνων και βίντεο, υποστηρίζοντας διάφορες μορφές αρχείων. Με τη «cv2», μπορούμε να εκτελέσουμε λειτουργίες όπως φιλτράρισμα, μετασχηματισμούς και ανίχνευση ακμών, καθώς και πιο προηγμένες εργασίες όπως ανίχνευση αντικειμένων, αναγνώριση προσώπων και παρακολούθηση κίνησης. Η βιβλιοθήκη αυτή υποστηρίζει επίσης λειτουργίες σχεδίασης, επιτρέποντας την προσθήκη σχημάτων, κειμένου και γραμμών σε εικόνες. Η ισχυρή συλλογή των χαρακτηριστικών του καθιστά τη «cv2» απαραίτητη για εφαρμογές στη ρομποτική, την επαυξημένη πραγματικότητα και οποιοδήποτε πεδίο όπου η επεξεργασία οπτικών δεδομένων είναι κρίσιμη.

Μια ακόμα σημαντική βιβλιοθήκη της Python είναι η «shutil» η οποία παρέχει μια υψηλού επιπέδου διεπαφή για λειτουργίες αρχείων, καθιστώντας ευκολότερο τον αυτοματισμό εργασιών που αφορούν την αντιγραφή, την αφαίρεση και τη διαχείριση αρχείων. Περιλαμβάνει συναρτήσεις για την αντιγραφή αρχείων και ολόκληρων δέντρων καταλόγων, την αφαίρεση αρχείων και καταλόγων, και τη μετακίνηση αρχείων ή καταλόγων σε διαφορετικές τοποθεσίες. Η βιβλιοθήκη αυτή προσφέρει επίσης εργαλεία

για το χειρισμό της χρήσης του δίσκου, επιτρέποντάς μας να ελέγξουμε τον διαθέσιμο και τον χρησιμοποιούμενο χώρο στο σύστημα αρχείων. Με τις απλές και ισχυρές της λειτουργίες, η «shutil» είναι ένα απαραίτητο εργαλείο για εργασίες όπως οι λειτουργίες αντιγράφου ασφαλείας, η οργάνωση αρχείων και οι διαδικασίες καθαρισμού, διευκολύνοντας τη διαχείριση αρχείων και καταλόγων σε εφαρμογές Python.

Μια καίριας σημασίας, για τον κώδικα μας, βιβλιοθήκη στην Python είναι η «py360convert» η οποία σχεδιάστηκε ειδικά για την χειρισμό μετατροπών εικόνων και βίντεο 360 μοιρών, παρέχοντας εργαλεία για τη μετατροπή μεταξύ διαφορετικών μορφών πανοραμικών εικόνων. Υποστηρίζει διάφορες προβολές όπως η «equirectangular», ο κυβικός χάρτης (cubemap) και η «fisheye», επιτρέποντας στους χρήστες να μετακινούνται εύκολα μεταξύ αυτών των μορφών για ποικίλες εφαρμογές. Αυτή η βιβλιοθήκη είναι ιδιαίτερα χρήσιμη για τη δημιουργία περιεχομένου εικονικής πραγματικότητας (VR), όπου ο ακριβής και αποτελεσματικός χειρισμός πανοραμικών φωτογραφιών είναι κρίσιμος. Με τον απλοποιημένο τρόπο μετατροπής και χειρισμού πανοραμικών εικόνων και βίντεο, το «py360convert» ευνοεί την ανάπτυξη ρεαλιστικών εμπειριών VR και διαδραστικών περιβαλλόντων, καθιστώντας το ένα απαραίτητο εργαλείο για προγραμματιστές που εργάζονται στον τομέα της VR και των πανοραμικών μέσων.

Τέλος, η βιβλιοθήκη «quaternion» στην Python σχεδιάστηκε για την αναπαράσταση και την εκτέλεση υπολογισμών με quaternions, τα οποία χρησιμοποιούνται εκτενώς στα γραφικά 3D, τη ρομποτική, την αεροδιαστημική και τη φυσική για τον αποτελεσματικό χειρισμό περιστροφών και προσανατολισμών. Αυτή η βιβλιοθήκη παρέχει μια αξιόπιστη υλοποίηση των quaternions, επιτρέποντας την εύκολη δημιουργία, επεξεργασία και μετατροπή μεταξύ quaternions και άλλων περιστροφικών αναπαραστάσεων όπως οι γωνίες Euler και οι πίνακες περιστροφής. Υποστηρίζει αριθμητικές πράξεις, κανονικοποίηση, σύζευξη και παρεμβολή μεταξύ quaternions. Με τον τρόπο αυτό, διευκολύνοντας τους πολύπλοκους περιστροφικούς υπολογισμούς με υψηλή ακρίβεια και ελάχιστα αριθμητικά σφάλματα, η βιβλιοθήκη «quaternion» είναι απαραίτητη για εφαρμογές που απαιτούν ομαλές και ακριβείς περιστροφές στον τρισδιάστατο χώρο, όπως προσομοιώσεις και συστήματα ελέγχου στη ρομποτική και την αεροδιαστημική μηχανική.

## 5.5 Gaussian Splatting με τη χρήση του COLMAP-PCD

Οι σύγχρονες τεχνικές για ανακατασκευή με κάμερα με ένα φακό βασίζονται κυρίως στην μέθοδο Structure from Motion (SfM). Ωστόσο, αυτές οι μέθοδοι συχνά αποδίδουν αποτελέσματα ανακατασκευής που στερούνται κρίσιμων πληροφοριών κλίμακας και απόδοσης του μεγέθους σε ρεαλιστικό επίπεδο και με την πάροδο του χρόνου, η συσσώρευση εικόνων οδηγεί σε αναπόφευκτα ζητήματα παραμόρφωσης που δυσχεραίνουν την απόδοση του τελικού αποτελέσματος. Αντίθετα, οι μέθοδοι χαρτογράφησης που βασίζονται σε σαρώσεις LiDAR είναι δημοφιλείς στην ανακατασκευή σκηνών μεγάλης κλίμακας(π.χ. αστικά τοπία με μεγάλη πληροφορία) λόγω των ακριβών μετρήσεων απόστασης, μια ικανότητα που απουσιάζει θεμελιωδώς και με αισθητό τρόπο από τις οπτικές προσεγγίσεις. Οι ερευνητές έχουν επιχειρήσει να χρησιμοποιήσουν ταυτόχρονα μετρήσεις LiDAR και κάμερας στην αναζήτηση της ακριβούς κλίμακας και των ιδιαίτερων λεπτομερειών χρώματος στις χαρτογραφικές αποδόσεις των εξεταζόμενων τοπίων. Ωστόσο, τα αποτελέσματα εξαρτώνται από την εξωτερική βαθμονόμηση και την ακρίβεια συγχρονισμού κατά τη διάρκεια της διαδικασίας αυτής. Έτσι λοιπόν προτείνεται από ορισμένους ερευνητές που προσπαθούν να βελτιώσουν αυτά τα προβλήματα μια καινοτόμος, αποδοτική μέθοδος ανακατασκευής που χρησιμοποιεί έναν προϋπάρχοντα χάρτη LiDAR ως σταθερό περιορισμό για την αποτελεσματική αντιμετώπιση των ενσωματωμένων προβλημάτων κλίμακας και ρεαλισμού στην απόδοση των μεγεθών που υπάρχουν στην ανακατασκευή με κάμερες με ένα φακό. Η μέθοδος αυτή φαίνεται να είναι η πρώτη που καταχωρεί εικόνες στον χάρτη νέφους σημείων χωρίς να απαιτείται ταυτόχρονη λήψη δεδομένων από την κάμερα και τον LiDAR αισθητήρα, δίνοντάς μας την ευελιξία να διαχειριστούμε το επίπεδο λεπτομέρειας ανακατασκευής σε διάφορες περιοχές ενδιαφέροντος. Για τη διευκόλυνση και την ανάπτυξη περαιτέρω έρευνας πάνω σε αυτόν τον ολοένα και αναπτυσσόμενο τομέα, έχει κυκλοφορήσει το Colmap-PCD<sup>[44]</sup>, ένα εργαλείο ανοιχτού κώδικα που αξιοποιεί τον αλγόριθμο του Colmap, το οποίο επιτρέπει την ακριβή λεπτομερή καταχώρηση εικόνων σε ένα νέφος σημείων.

### 5.5.1 Τρόπος χρήση του COLMAP-PCD

Για να δοκιμάσουμε λοιπόν αυτό το νέο εργαλείο που θα μπορούσαμε εύκολα να το θεωρήσουμε και μια χρήσιμη και καινοτόμα προέκταση του Colmap έπρεπε να έχουμε ένα δικό μας παράδειγμα στο οποίο να μπορέσουμε να δοκιμάσουμε τον τρόπο χρήσης του αλλά και να ανακαλύψουμε τις νέες λειτουργίες που προσφέρει. Γι' αυτό το λόγο αποφασίσαμε πως ο χώρος στον οποίο θα δοκιμάσουμε αυτές τις νέες λειτουργίες θα είναι ο χώρος του εργαστήριου ναυτικής μηχανολογίας για τον οποίο είχαμε το νέφος σημείων(point cloud) αφού είχε πραγματοποιηθεί παλαιότερα 3D scanning αλλά οι φωτογραφίες ήταν λιγοστές και ασπρόμαυρες οπότε με αυτές δε θα μπορούσαμε σε καμία περίπτωση να πέτυχουμε μια ρεαλιστική και ζωντανή αναπαράσταση. Για να αποκτήσουμε όμως τις απαραίτητες φωτογραφίες από τον χώρο του εργαστήριου ναυτικής μηχανολογίας



χρησιμοποιήσαμε την 360 μοιρών κάμερα Insta360 ONE RS για την οποία αξίζει να κάνουμε μια μικρή αναφορά.

Η Insta360 ONE RS είναι μια ευέλικτη και ισχυρή κάμερα δράσης, σχεδιασμένη για την καταγραφή εντυπωσιακών και δυναμικών λήψεων. Διαθέτοντας μια μοντέρνα σχεδίαση με καινοτόμα χαρακτηριστικά, η Insta360 RS επιτρέπει στους χρήστες να αλλάζουν μεταξύ διαφορετικών επιλογών φακών, όπως ο φακός 4K Boost, ο φακός 360 και ο φακός ευρείας γωνίας 1 ίντσας, που έχει συν-αναπτυχθεί με την Leica. Αυτή η ευελιξία επιτρέπει στους χρήστες να προσαρμόζουν την εμπειρία λήψης τους σε διάφορα περιβάλλοντα και ανάγκες, είτε πρόκειται για την καταγραφή βίντεο υψηλής ανάλυσης 4K, είτε για την καταγραφή πανοραμικών λήψεων 360 μοιρών, είτε για εντυπωσιακές λήψεις σε συνθήκες χαμηλού φωτισμού με τον αισθητήρα 1 ίντσας. Η Insta360 ONE RS διαθέτει επίσης εντυπωσιακή τεχνολογία σταθεροποίησης, εξασφαλίζοντας ομαλές και σταθερές λήψεις ακόμη και υπό τις πιο απαιτητικές συνθήκες. Με χαρακτηριστικά όπως το FlowState Stabilization, το Active HDR και μια φιλική προς το χρήστη εφαρμογή για την επεξεργασία και κοινοποίηση, η Insta360 RS ξεχωρίζει ως ένα αξιόπιστο εργαλείο για δημιουργούς περιεχομένου, περιπετειώδεις και βιντεογράφους που επιθυμούν να ξεπεράσουν τα όρια της εκάστοτε δραστηριότητάς τους. Επιπλέον, η ανθεκτική κατασκευή και η αδιάβροχη σχεδίασή της την καθιστούν τον ιδανικό σύντροφο για εξωτερικές δραστηριότητες και ακραία σπορ. Αν και εμείς στην ουσία για τη χρήση την οποία θέλουμε τη κάμερα αξιοποιούμε πολύ λίγες από τις δυνατότητες της αποτελεί ένα καίριο εργαλείο για την απόκτηση των κατάλληλων φωτογραφιών από τον χώρο του εργαστήριου ναυτικής μηχανολογίας ,πάντα συνοδευμένη με ένα καλό, στιβαρό και σταθερό τρίποδο.

Χρησιμοποιώντας λοιπόν αυτή τη κάμερα εισερχόμαστε στον χώρο του εργαστήριου ναυτικής μηχανολογίας και τραβάμε φωτογραφίες σε 70 διαφορετικά σημεία του χώρου και σε δύο διαφορετικά ύψη. Είναι σημαντικό να καταγράψουμε τη σχετική θέση της πρώτης φωτογραφίας σε σχέση με τα υπόλοιπα αντικείμενα του χώρου για να μπορέσουμε έπειτα να βρούμε τη θέση της στο νέφος σημείων που έχουμε. Για να το πέτυχουμε αυτό θα ανοίξουμε το νέφος σημείων του εργαστήριου ναυτικής μηχανολογίας στο CloudCompare και θα βρούμε τη θέση του σημείου από το οποίο είχαμε μετρήσει τις αποστάσεις από τη πρώτη φωτογραφία ώστε τελικά να μας γίνει γνωστή η θέση της πρώτης φωτογραφίας μέσα στο νέφος σημείων. Αυτή η διαδικασία μπορεί να γίνει όχι αποκλειστικά για τη πρώτη φωτογραφία απλώς πρέπει να γίνει απαραίτητα για τουλάχιστον μία. Από τις 140 φωτογραφίες 360 μοιρών που τραβήξαμε σε 70 διαφορετικά σημεία εξαγουμε για καθεμία από αυτές 18 φωτογραφίες με βήμα στη στροφή γύρω από τον οριζόντιο άξονα 60 μοίρες και γύρω από τον κατακόρυφο 45 μοίρες και έτσι έχουμε συνολικά πλέον 2520 φωτογραφίες τις οποίες θα εισάγουμε στο Colmap-PCD.

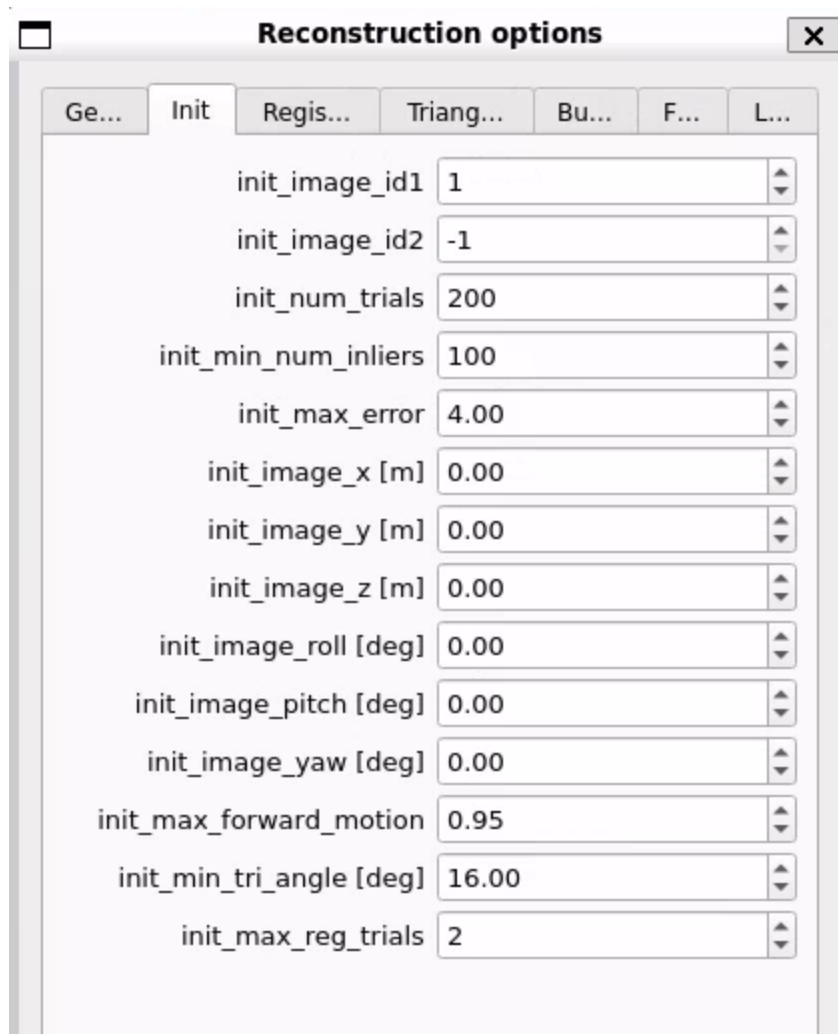
Πριν προχωρήσουμε λοιπόν στη χρήση του Colmap-PCD πρέπει να προετοιμάσουμε το νέφος σημείων με κατάλληλο τρόπο ώστε να μπορεί να αξιοποιηθεί καταλληλά. Για να γίνει αυτή η προετοιμασία χρησιμοποιούμε το CloudCompare και μειώνουμε την ανάλυση του σημείου νέφους σε 3-5cm ,δηλαδή διατηρούμε τα σημεία που απέχουν το λιγότερο 3-5cm και αγνοούμε τα υπόλοιπα. Στη δική μας περίπτωση για

μεγαλύτερη ακρίβεια θεωρούμε ελάχιστη απόσταση 1cm. Έπειτα υπολογίζουμε τις κανονικές χρησιμοποιώντας ακτίνα 10-20 cm και αποθηκεύουμε το σημειακό νέφος σε μορφή PLY (binary ή ASCII). Κανονικές (normals) είναι διευθύνσεις κάθετων γραμμών σε επιφάνειες σημείων στο τρισδιάστατο χώρο. Στον τομέα της υπολογιστικής γεωμετρίας και της επεξεργασίας σημειακών νεφών, οι κανονικές χρησιμοποιούνται για να προσδιορίσουν την κατεύθυνση της επιφάνειας σε κάθε σημείο του νέφους σημείων. Πως υπολογίζονται όμως;

Οι κανονικές υπολογίζονται συνήθως με την ανάλυση της γεωμετρίας των σημείων γύρω από κάθε σημείο στο σημειακό νέφος. Ένας κοινός τρόπος υπολογισμού τους είναι η χρήση της τοπικής εμβέλειας (radius) για να καθοριστούν τα κοντινότερα σημεία γύρω από το προς υπολογισμό σημείο. Μια κοινή τεχνική περιλαμβάνει αρχικά τη συγκέντρωση των γειτονικών σημείων δηλαδή τον εντοπισμό των σημείων που βρίσκονται σε μια συγκεκριμένη απόσταση (radius) από το σημείο ενδιαφέροντος και στη συνέχεια τον υπολογισμό της κανονικής με χρήση της μέσης κατεύθυνσης των εφαπτόμενων επιφανειών των γειτονικών σημείων για να υπολογιστεί η κανονική κατεύθυνση. Αυτό συνήθως γίνεται με την ανάλυση του επιπέδου που περνάει από τα γειτονικά σημεία. Ας πούμε ότι έχουμε ένα σημείο  $P$  με γειτονικά σημεία  $P1, P2, P3$ . Μπορούμε να χρησιμοποιήσουμε τις συντεταγμένες αυτών των σημείων για να βρούμε το επίπεδο που τα περιέχει και να προσδιορίσουμε την κανονική του επιπέδου. Σε μας όλη αυτή η διεργασία γίνεται αυτόματα και γρήγορα από το CloudCompare.

Τώρα λοιπόν ήρθε η ώρα να χρησιμοποιήσουμε το Colmap-PCD αφού έχουμε πλέον τα απαραίτητα δεδομένα εισόδου. Αρχικά λοιπόν αφού ανοίξουμε Colmap-PCD, το οποίο όπως είπαμε βασίζεται στον αλγόριθμο του Colmap, δημιουργούμε το database και ορίζουμε το φάκελο από όπου θα τραβάει τις φωτογραφίες. Στη συνέχεια, προχωράμε στην εξαγωγή των χαρακτηριστικών των φωτογραφιών αφού εισάγουμε το μοντέλο και τις παραμέτρους της κάμερας τα οποία εμείς έχουμε καθορίσει αφού εξαγάγαμε τις επιμέρους φωτογραφίες από τις φωτογραφίες των 360 μοιρών. Έπειτα προχωράμε στο ταίριασμα των χαρακτηριστικών των φωτογραφιών που εισαγάγαμε και αφού ολοκληρωθεί αυτή η διαδικασία εισάγουμε το νέφος σημείων (point cloud). Το πιο σημαντικό βήμα ούτως ώστε να δουλέψει το «image to point cloud registration» είναι να ορίσουμε τη θέση της πρώτης φωτογραφίας καθώς και ποια είναι αυτή αλλά και την αμέσως επόμενη στη διαδικασία αυτή ούτως ώστε να μπορέσουν να βρεθούν κοινά χαρακτηριστικά και να προχωρήσει ομαλά και αποτελεσματικά η διαδικασία αυτή. Όπως βλέπουμε στη παρακάτω

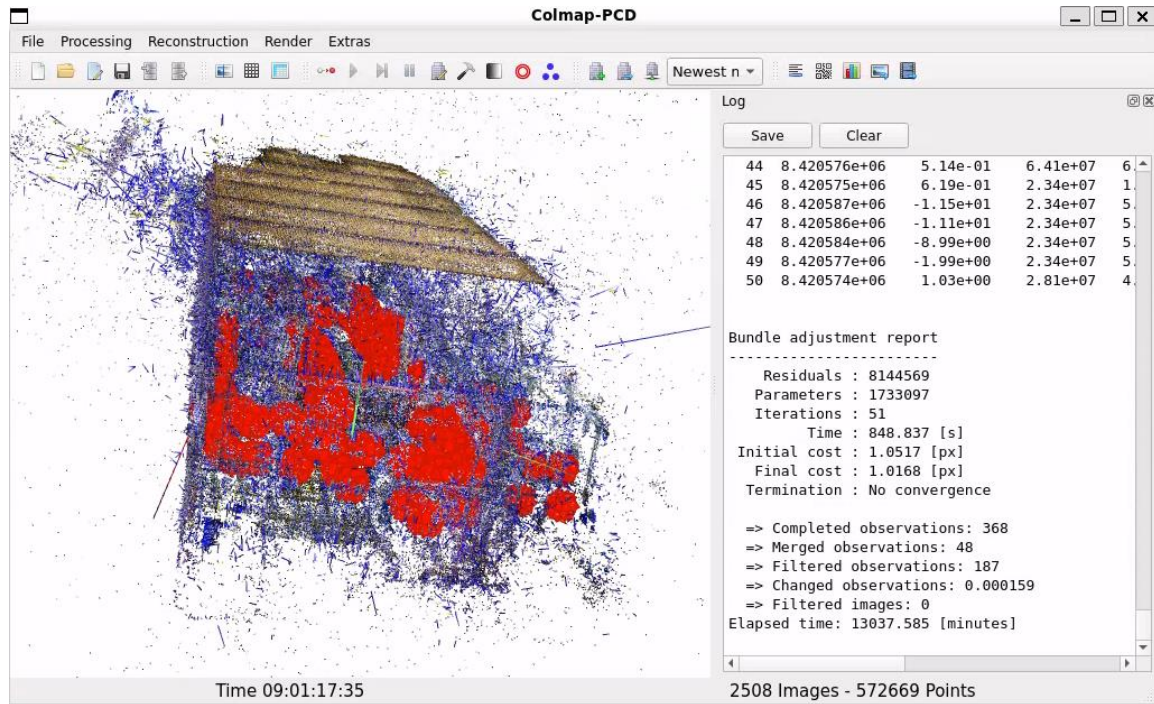
φωτογραφία ορίζουμε όχι μόνο τη θέση x,y,z της φωτογραφίας αλλά και τη περιστροφή της roll ,pitch ,yaw.



Εικόνα 33: Ρυθμίσεις ανακατασκευής Colmap-PCD

Είναι σημαντικό να σημειωθεί πως για να μπορέσουμε να γνωρίζουμε το id της κάθε φωτογραφίας πρέπει να «αναγκάσουμε» το Colmap-PCD να τις διαβάσει με συγκεκριμένη σειρά κάτι το οποίο πετυχαίνουμε προσθέτοντας στο όνομα της κάθε φωτογραφίας έναν αύξοντα τετραψήφιο αριθμό από το 0001 στο 2520. Αφού λοιπόν ολοκληρώσουμε όλα τα παραπάνω βήματα και ορίσουμε ως ζεύγος πρώτων φωτογραφιών τις φωτογραφίες που κοιτούν στην ίδια κατεύθυνση και προκύπτουν από διαφορετικές κοντινές φωτογραφίες 360 μοιρών ,ξεκινάμε την ανακατασκευή και περιμένουμε. Η ανακατασκευή αυτή μπορεί να διαρκέσει από λίγα λεπτά μέχρι πολλές μέρες ανάλογα με

το πόσες φωτογραφίες εισάγουμε αλλά και πόσο μεγάλο είναι το νέφος σημείων. Το τελικό αποτέλεσμα που προκύπτει φαίνεται στην παρακάτω φωτογραφία:



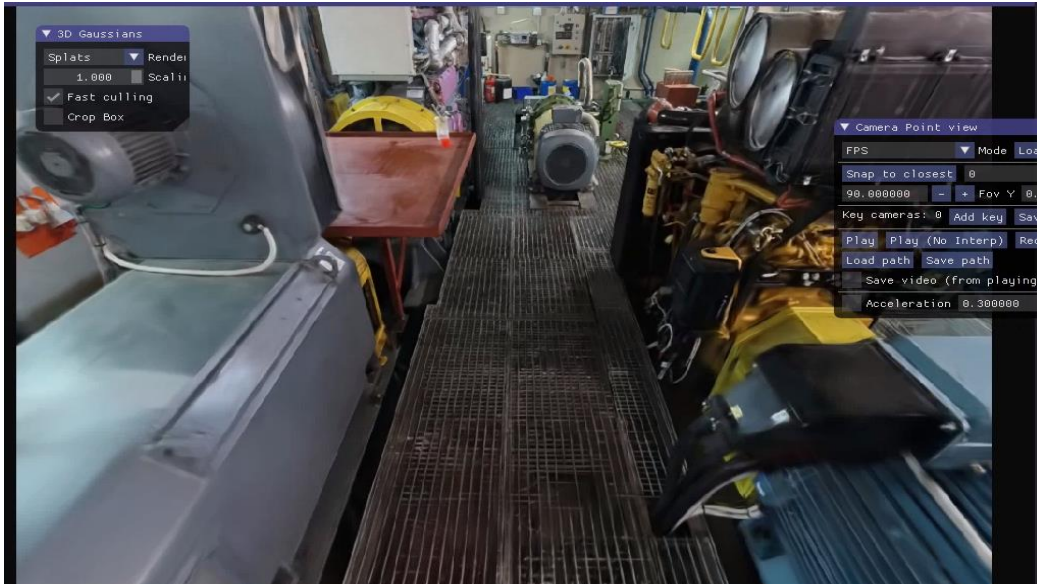
Εικόνα 34: Αποτέλεσμα ανακατασκευής στο Colmap-PCD

Παρατηρούμε λοιπόν ότι κατάφερε να εγγράψει στο νέφος σημείων 2508 από τις 2520 φωτογραφίες που του δώσαμε, ένα εξαιρετικό αποτέλεσμα το οποίο όμως για να παραχθεί χρειάστηκε σχεδόν 9 ημέρες. Αυτό βέβαια μπορεί να οφείλεται και σε παράλληλη χρήση του υπολογιστή και της κάρτας γραφικών που επιβράδυνε σημαντικά τη διαδικασία. Το αρχείο που δημιουργήθηκε δηλαδή για την ακρίβεια τα αρχεία που εξαγάγαμε για να τα χρησιμοποιήσουμε μετέπειτα ως δεδομένα εισόδου για τη τρισδιάστατη αναπαράσταση σε θέμα χώρου συνολικά (εξαιρώντας το χώρο των φωτογραφιών) δε ξεπερνούν το 1 GB. Πλέον λοιπόν έχουμε έτοιμα τα δεδομένα για να τα εξάγουμε από το Colmap και να τα εισάγουμε στον αλγόριθμο που πραγματοποιεί τρισδιάστατη αναπαράσταση με Gaussian splatting και να μελετήσουμε τα αποτελέσματα. Πάντως καταφέραμε με αυτή την προέκταση του Colmap να εγγράψουμε νέες φωτογραφίες σε ένα προϋπάρχον νέφος σημείων εύκολα και αποτελεσματικά.

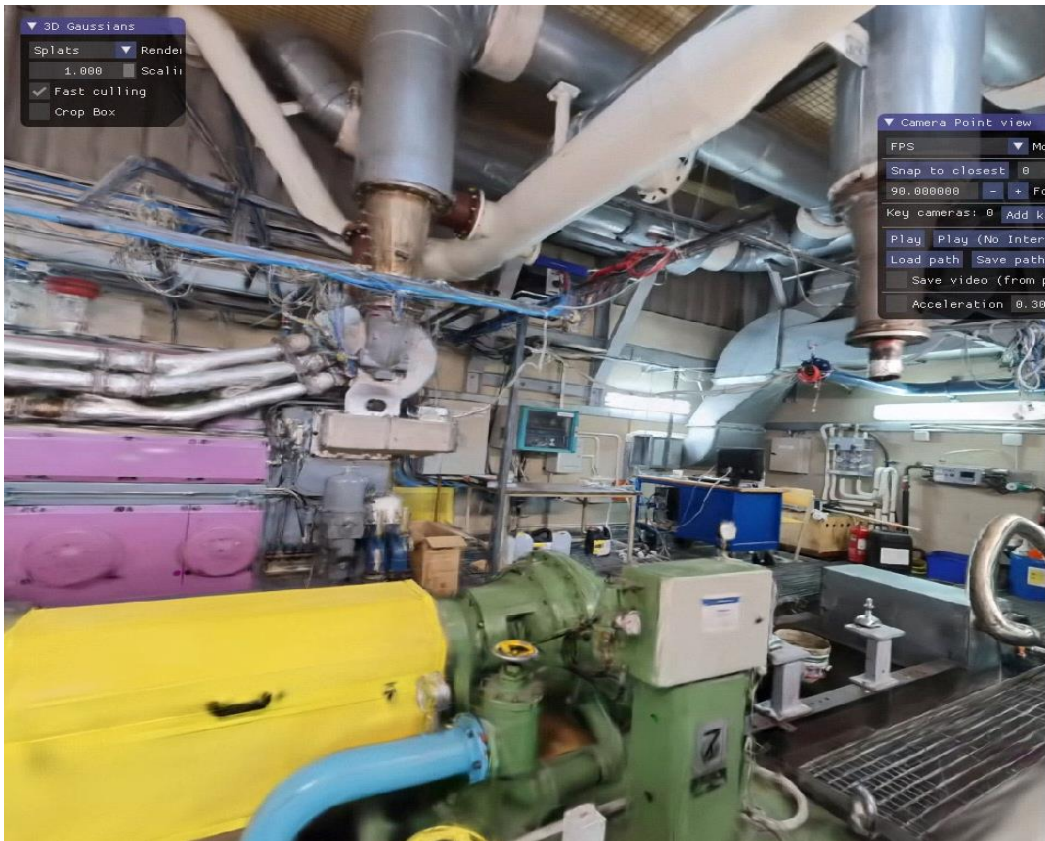
## 5.5.2 Αποτελέσματα τρισδιάστατης αναπαραστασης μέσω Gaussian Splatting με δεδομένα εισόδου από το COLMAP-PCD

Αφού λοιπόν εγγράψουμε τις φωτογραφίες στο νέφος σημείων που έχουμε μέσω του Colmap-PCD, εξάγουμε τα δεδομένα τα οποία παρήχθησαν είτε σε μορφή .bin είτε σε μορφή .txt αν θέλουμε να τα επεξεργαστούμε μετέπειτα ευκολότερα. Τα δεδομένα που παράγονται από το Colmap-PCD είναι τα τρία αρχεία που είχαμε αναφέρει προηγουμένως το cameras.txt, το images.txt και το points3D.txt με τα οποία δημιουργούμε το φάκελο που θα δοθεί ως είσοδο για να γίνει η τρισδιάστατη αναπαράσταση μέσω Gaussian Splatting. Το αποτέλεσμα όταν χρησιμοποιούμε επαρκή αριθμό φωτογραφιών από πολλές διαφορετικές γωνίες θέασης είναι εξαιρετικό, πολύ ρεαλιστικό και επιτρέπει στον θεατή να διακρίνει διάφορες λεπτομέρειες. Η κίνηση μέσα στον χώρο είναι ομαλή χωρίς πολλά κολλήματα κατά τη μετακίνηση της κάμερας θέασης και με ελάχιστα σημεία στο χώρο, όπου υπάρχουν θολά σημεία, κάτι που οφείλεται πιθανότατα στη μετακίνηση, προσθήκη ή αφαίρεση αντικειμένων στο χώρο του εργαστήριου ναυτικής μηχανολογίας μεταξύ της φοράς που έγινε το 3D scanning και της φοράς που πήραμε τις φωτογραφίες. Ήρθε λοιπόν η ώρα να παραθέσουμε εικόνες από το αποτέλεσμα της τρισδιάστατης αναπαραστασης που φανερώνουν το πόσο επιτυχημένη και τρομερά ρεαλιστική είναι η αναπαράσταση. Το παράδειγμα του οποίου θα παρουσιάσουμε το αποτέλεσμα είναι αυτό που χρησιμοποιεί όλες τις φωτογραφίες εκτός από αυτές που κοιτούν πάνω και κάτω (περιστροφή ως προς τον Y) στις 45 μοίρες αντίστοιχα και αφορούν αποκλειστικά τις γωνίες των 60, 120 και 240 μοιρών κατά την περιστροφή στον άξονα Z. Αυτό είναι ένα πολύ καλό παράδειγμα που αναδεικνύει την αποτελεσματικότητα αυτής της μεθόδου τρισδιάστατης αναπαραστασης. Αξίζει να σημειωθεί πως η ώρα που απαιτήθηκε για να τρέξει ο κώδικας για την τρισδιάστατη αναπαράσταση δε ξεπέρασε ξανά τα 30 λεπτά και ο όγκος του αρχείου δεν έφτασε ούτε το ένα 1GB. Βέβαια όταν χρησιμοποιούμε όλες τις φωτογραφίες ο κώδικας κολλάει γιατί δε φτάνει η μνήμη της κάρτας γραφικών ενώ αν βάλουμε πάνω από περίπου 1800 φωτογραφίες ο κώδικας μπορεί να χρειαστεί και μερικές ώρες. Στις φωτογραφίες που θα παραθέσουμε παρατηρούμε ότι από τις περισσότερες οπτικές και γωνίες θέασης το αποτέλεσμα είναι άκρως ρεαλιστικό και ζωντανό σαν να βρίσκεσαι την ίδια ώρα στο χώρο αυτό χωρίς να χάνεις καμία λεπτομέρεια. Σίγουρα βέβαια σε σημεία όπου οι λήψεις των φωτογραφιών είναι περιορισμένες έχει κάποια θολά σημεία όπως αναφέραμε παραπάνω. Αδιαμφισβήτητα όμως τα σημαντικά στοιχεία που παρατηρούμε σε αυτή την αναπαράσταση είναι η ακρίβεια στη λεπτομέρεια, στους χρωματισμούς και στις σκιές, στοιχεία που αυτή η μέθοδος καταφέρνει να τα παρουσιάσει τόσο επιτυχημένα και τόσο ζωντανά που κάνει άλλες μεθόδους τρισδιάστατης αναπαραστασης να φαίνονται πολύ λίγες μπροστά στις δυνατότητες που έχει το Gaussian Splatting. Μερικές εικόνες λοιπόν που αφορούν αυτό το παράδειγμα παρατίθενται παρακάτω:

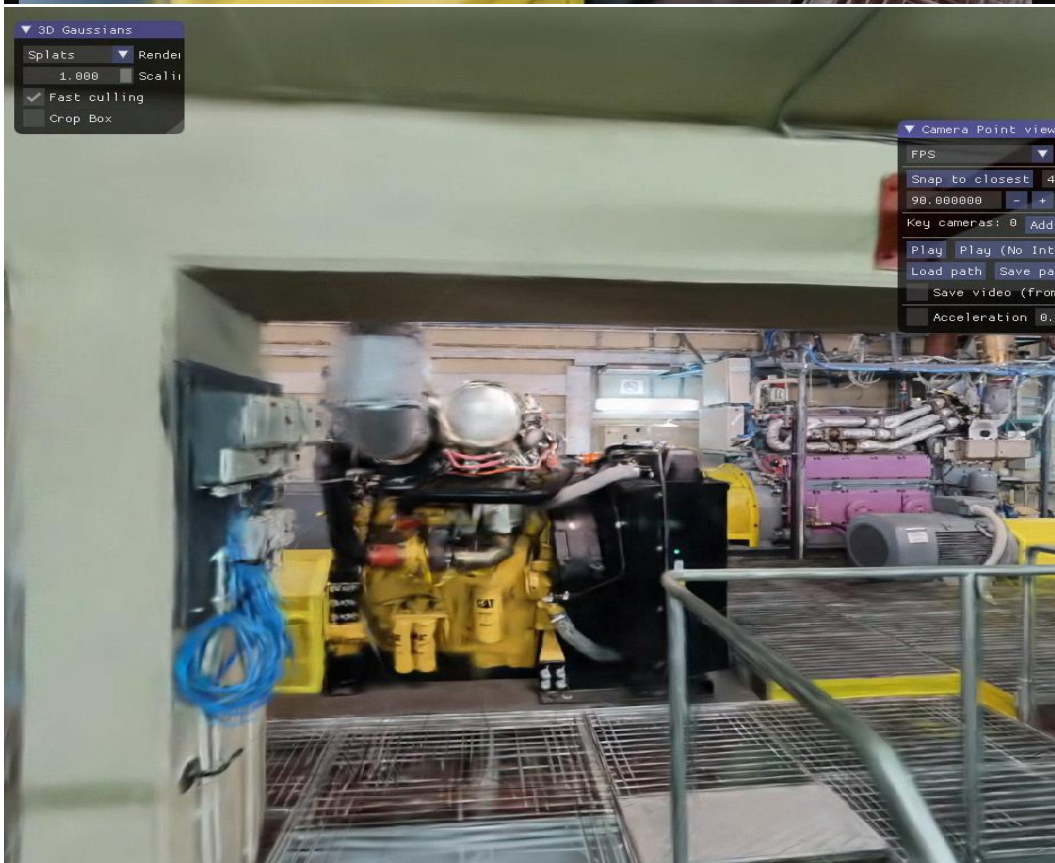
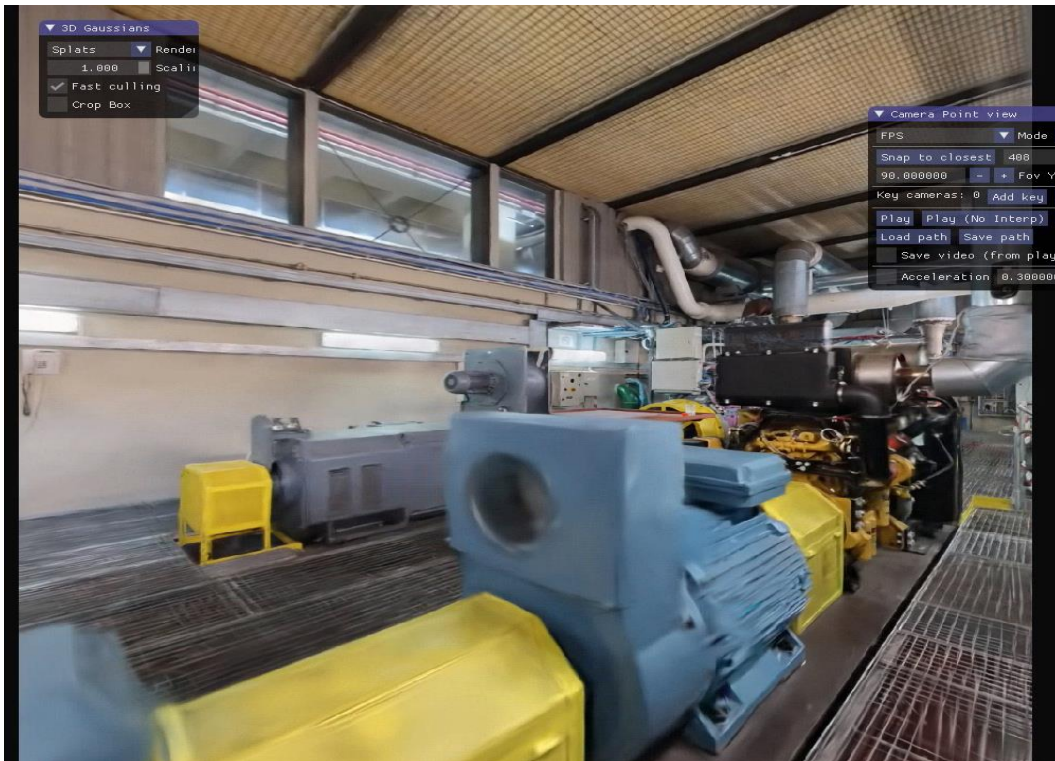










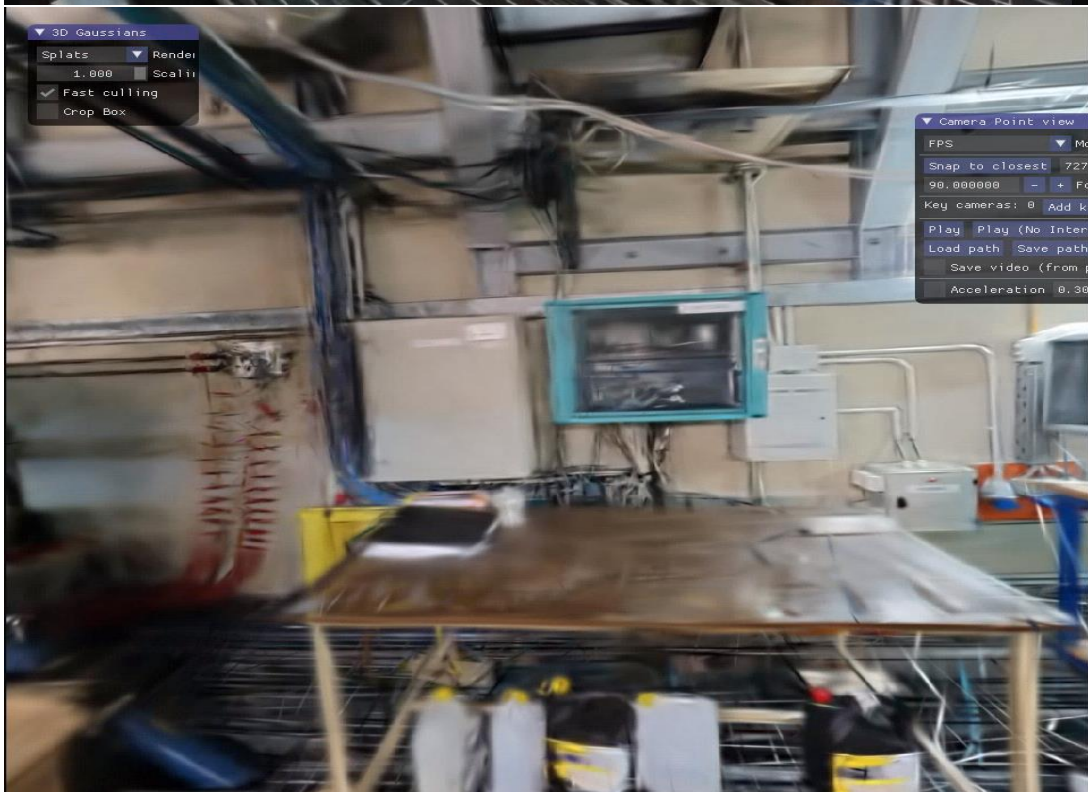


Εικόνες 35 α,β,γ,δ,ε,στ: Αποτελέσματα τρισδιάστατης αναπαράστασης με δεδομένα από το Colmap-PCD

Στη συνέχεια, στο συγκεκριμένο παράδειγμα το οποίο είδαμε ότι ήταν επιτυχημένο, επιχειρήσαμε να απενεργοποιήσουμε το “self-densify” δηλαδή να μη προστίθενται σημεία στο νέφος σημείων που του δίνουμε αρχικά αφού γνωρίζουμε πως σίγουρα αυτό είναι σωστό αφού έχει προέλθει από επιτυχημένο 3D scanning του χώρου του εργαστήριου ναυτικής μηχανολογίας. Όμως όπως αναφέραμε παραπάνω υπήρξε μετακίνηση ,προσθήκη ή αφαίρεση αντικειμένων στο χώρο μεταξύ της φοράς που έγινε το 3D scanning και της φοράς που πήραμε τις φωτογραφίες οπότε το αποτέλεσμα δεν είναι τόσο επιτυχημένο αφού υπάρχουν σημεία θολά και «σύννεφα» στο χώρο .Αυτό συμβαίνει γιατί είτε δεν υπάρχουν σημεία στο νέφος να αντιστοιχούν σε αντικείμενα που φαίνονται στις φωτογραφίες είτε γιατί υπάρχουν σημεία στο νέφος που αντιστοιχούν σε αντικείμενα που δεν υπάρχουν στις φωτογραφίες. Αυτή η αναντιστοιχία μεταξύ του δοθέντος νέφους σημείων και των φωτογραφιών δημιουργεί διάφορα προβλήματα στην τρισδιάστατη αναπαράστασή τα οποία μπορούμε να παρατηρήσουμε στις παρακάτω εικόνες :







Εικόνες 36 α,β,γ: Αποτελέσματα τρισδιάστατης αναπαράστασης με δεδομένα από το Colmap-PCD χωρίς “self-densify”

Εκτός από το συγκεκριμένο παράδειγμα στο οποίο αναφερθήκαμε εκτενώς παραπάνω εκτελέσαμε διάφορες δοκιμές χρησιμοποιώντας διαφορετικές φωτογραφίες από διαφορετικές γωνίες θέασης και ελέγχοντας το αποτέλεσμα προσπαθήσαμε να καταλάβουμε πόσες και ποιες φωτογραφίες χρειάζονται για να βγει ένα ρεαλιστικό και ευκρινές αποτέλεσμα. Καταρχάς, δοκιμάσαμε να τρέξουμε τον κώδικα αφαιρώντας κάθε δεύτερη φωτογραφία και έπειτα κάθε δεύτερη και τρίτη και ούτω καθεξής μέχρις ότου χαλάσει το αποτέλεσμα κάτι το οποίο έγινε στο επόμενο βήμα δηλαδή όταν κρατούσαμε μια και αφαιρούσαμε τις επόμενες τρεις φωτογραφίες. Συνεχίζοντας δοκιμάσαμε να αφαιρούμε όλες τις φωτογραφίες από μια θέση αρχικά αφαιρώντας κάθε δεύτερη θέση και προχωρώντας με την ίδια λογική όπως παραπάνω. Σε αυτή τη δοκιμή παρατηρήσαμε πως όταν κρατάμε τις φωτογραφίες από μια τοποθεσία και διώχνουμε τις φωτογραφίες από τις δυο επόμενες το αποτέλεσμα ξεκινάει να χαλάει σημαντικά ,όμως η δοκιμή από την οποία δεν έχει νόημα να συνεχίσουμε περαιτέρω λόγω του προβληματικού αποτελέσματος είναι όταν κρατάμε τις φωτογραφίες της πρώτης θέσης και αφαιρούμε αυτές από τις επόμενες τρεις θέσεις στο χώρο.

Δοκιμάσαμε και άλλα διάφορα παραδείγματα στα οποία κρατούσαμε φωτογραφίες διαφορετικές για κάθε γωνιά και για κάθε ύψος και αυτό που παρατηρήσαμε είναι πως αν καλύπτονται επαρκώς όλες οι γωνίες θέασης το αποτέλεσμα είναι αρκετά ρεαλιστικό και ζωντανό ενώ στα σημεία όπου οι φωτογραφίες είναι λιγοστές και δε βλέπουν το συγκεκριμένο χώρο από κάθε κατεύθυνση το αποτέλεσμα είναι θολό ,μη ακριβές και χάνει σημαντικά σε ζωντάνια και ευκρίνεια. Πιο συγκεκριμένα όταν αφαιρούμε εντελώς μια οπτική γωνία της κάμερας αν το συγκεκριμένο σημείο δε το βλέπει άλλη γωνιά άλλης κάμερας το αποτέλεσμα θα έχει σοβαρό θέμα.

## 6. Gaussian splatting στο περιβάλλον του Unity

### 6.1 Λίγα λόγια για το περιβάλλον του Unity

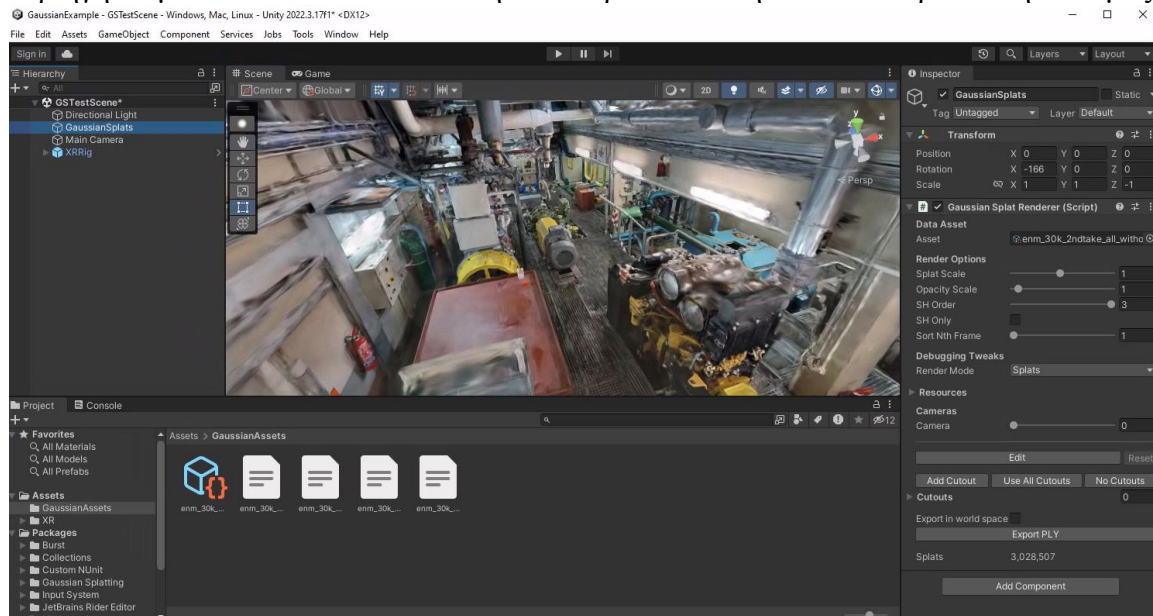
Το Unity<sup>[3]</sup> είναι μια ευέλικτη μηχανή ανάπτυξης παιχνιδιών που χρησιμοποιείται για τη δημιουργία τόσο 2D όσο και 3D παιχνιδιών και διαδραστικών εφαρμογών. Δίνει τη δυνατότητα στους προγραμματιστές να έχουν στη κατοχή τους ολοκληρωμένα εργαλεία και έτσι εκτεταμένες δυνατότητες για την ανάπτυξη παιχνιδιών για διάφορες πλατφόρμες, όπως υπολογιστές, κονσόλες, κινητές συσκευές μέχρι και εφαρμογές εικονικής πραγματικότητας. Το Unity προσφέρει ένα φιλικό προς τον χρήστη περιβάλλον και μια ισχυρή γλώσσα προγραμματισμού, τη C#, επιτρέποντας στους προγραμματιστές να δημιουργούν παιχνίδια και εφαρμογές εύκολα.

Το Unity είναι σχεδιασμένο για να απλοποιεί την ανάπτυξη παιχνιδιών, παρέχοντας προ-κατασκευασμένα εργαλεία και δυνατότητες που επιτρέπουν στους προγραμματιστές να αναπτύξουν και να τελειοποιήσουν τα έργα τους εύκολα. Ένα από τα πιο βασικά χαρακτηριστικά του Unity είναι ο ισχυρός editor του, που παρέχει μια πολυλειτουργική και εξελιγμένη οπτική διεπαφή για το σχεδιασμό και την κατασκευή παιχνιδιών. Οι χρήστες μπορούν να σύρουν και να αποθέτουν στοιχεία, όπως μοντέλα 3D και αρχεία ήχου, εύκολα και γρήγορα στον editor και να τα διαμορφώσουν για να δημιουργήσουν την επιθυμητή εμπειρία παιχνιδιού. Η υποστήριξη για multiplayer σε πραγματικό χρόνο και δικτύωση είναι ακόμα ένα σημαντικό χαρακτηριστικό του Unity. Η ασφάλεια αποτελεί επίσης προτεραιότητα για την πλατφόρμα. Διάφορα εργαλεία που έχουν αναπτυχθεί και όλο εξελίσσονται καθώς και όλο και νέα χαρακτηριστικά ασφαλείας προστατεύουν από κοινές απειλές που όμως δε πρέπει να αγνοούνται, όπως το hacking και το cheating. Είναι ακόμα σημαντικό να αναφερθεί πως υπάρχουν πολλά βίντεο εκμάθησης του Unity καθώς και οδηγοί χρήσης ελεύθεροι στο διαδίκτυο που εξηγούν ως επί το πλείστον τις λειτουργίες της εφαρμογής. Από ερασιτέχνες προγραμματιστές έως μεγάλες ομάδες επιχειρήσεων, όλοι επωφελούνται από την ευελιξία που προσφέρει η πλατφόρμα Unity για την ανάπτυξη παιχνιδιών.

## 6.2 Gaussian splatting στο περιβάλλον Unity

Αφού ολοκληρώσαμε το “image to point cloud registration” και τρέξαμε τον κώδικα για την τρισδιάστατη αναπαράσταση μέσω Gaussian Splatting πλέον μπορούμε να εισάγουμε και να επεξεργαστούμε αυτό το αρχείο της τρισδιάστατης αναπαράστασης στο Unity. Για να εισάγουμε όμως αυτό το αρχείο χρησιμοποιούμε ένα έτοιμο project για το Unity που έχει αναπτυχθεί από τον χρήστη “aras-p”<sup>[4]</sup> και μας καθιστά πολύ εύκολη την ενσωμάτωση του αρχείου στο περιβάλλον του Unity. Με αυτόν τον τρόπο εκτός από το γεγονός ότι μπορούμε να περιηγηθούμε μέσα στην τρισδιάστατη αναπαράσταση καλύτερα και πιο ομαλά ,μπορούμε και να δούμε από όλες τις δυνατές οπτικές γωνίες σε σχέση με τον default viewer που είχε αναπτύξει η ομάδα που ανέπτυξε τον κώδικα για το Gaussian Splatting. Ακόμα με το Unity ανοίγεται μπροστά μας μια τεράστια γκάμα από λειτουργίες τις οποίες μπορούμε να αξιοποιήσουμε για να βελτιώσουμε το οπτικό αποτέλεσμα ή και για να κατανοήσουμε καλύτερα τις δυνατότητες και τα όρια του Gaussian Splatting ως μέθοδο τρισδιάστατης αναπαράστασης.

Η διαδικασία εισαγωγής του αρχείου που έχει προκύψει από το Gaussian Splatting στο Unity είναι πολύ απλή και γρήγορη αφού το μόνο που χρειαζόμαστε ,αφού γίνει το clone από το GitHub και ανοίξουμε το project από τα έτοιμα που έχουν δημιουργηθεί , είναι να ακολουθήσουμε τα εξής βήματα: Από το μενού του Unity να επιλέξουμε «Tools -> Gaussian Splats -> Create GaussianSplatAsset» και αφού εισάγουμε το .ply αρχείο και επιλέξουμε την ποιότητα γραφικών , να δημιουργήσουμε το Gaussian Splat αρχείο. Αυτό το αρχείο το τοποθετούμε στο Gaussian Splat Renderer (script) και πλέον μπορούμε να περιηγηθούμε κανονικά στην τρισδιάστατη αναπαράσταση μας.



Εικόνα 37: Gaussian splatting στο περιβάλλον Unity

Αξίζει να σημειωθεί πως μέσα από το συγκεκριμένο Project για το Unity που έχει αναπτυχθεί από τον χρήστη “aras-p” μας δίνεται η δυνατότητα να εισάγουμε την τρισδιάστατη αναπαράσταση σε διαφορετικό project ανάλογα με τον τρόπο μετάδοσης των γραφικών δηλαδή ανάλογα με το “render pipeline” που επιθυμούμε να χρησιμοποιήσουμε. Τα “render pipelines” στο Unity είναι τρία το Universal Render Pipeline (URP) , το High-Definition Render Pipeline (HDRP) και το Built-In Render Pipeline.

Η Universal Render Pipeline έχει σχεδιαστεί για να παρέχει γραφικά υψηλής ποιότητας διατηρώντας ταυτόχρονα την βέλτιστη απόδοση. Η URP υποστηρίζει τις πιο πρόσφατες τεχνικές απόδοσης γραφικών. Υποστηρίζονται ο δυναμικός φωτισμός, η εκτέλεση σε GPU (GPU instancing) και τα εφέ μετα-επεξεργασίας (post-processing effects). Η URP είναι συμβατή με ένα ευρύ φάσμα πλατφορμών και λειτουργεί με μια διαφορετικών συσκευών. Είτε αναπτύσσουμε το παιχνίδι μας για κινητά τηλέφωνα είτε για υπολογιστές υψηλών επιδόσεων και κονσόλες, θα λειτουργεί άψογα και απροβλημάτιστα. Ακόμα είναι εύκολο να προσαρμοστεί ο τρόπος απόδοσης των γραφικών και να προστεθούν νέες λειτουργίες. Η URP μπορεί να είναι πολύ αποδοτική όμως για να το πέτυχει αυτό δεν έχει κάποιες ιδιαίτερες λειτουργίες και χαρακτηριστικά .

Η High Definition Render Pipeline είναι σχεδιασμένη για να παρέχει γραφικά υψηλής ποιότητας για προγράμματα και παιχνίδια υψηλών επιδόσεων σε κορυφαίες κονσόλες και στους υπολογιστές υψηλών προδιαγραφών. Η HDRP υποστηρίζει τις πιο σύγχρονες τεχνικές απόδοσης. Για παράδειγμα, η HDRP υποστηρίζει το ray tracing, τα ογκομετρικά σύννεφα, τον ογκομετρικό φωτισμό και τον φωτισμό HDR. Η HDRP έχει σχεδιαστεί για να παρέχει εξαιρετικά ρεαλιστικά γραφικά με ακριβείς αντανακλάσεις, σκιές και φωτισμούς. Πιο συγκεκριμένα, υποστηρίζει το ray tracing, το οποίο μπορεί να προσομοιώσει ρεαλιστικές αντανακλάσεις και σκιές. Αυτή η τεχνική επιτρέπει τη δημιουργία ακριβών αντανακλάσεων που βασίζονται στην πραγματική γεωμετρία της σκηνής, αντί για προκατασκευασμένες αντανακλάσεις. Η HDRP υποστηρίζει επίσης ογκομετρικό φωτισμό, ο οποίος μπορεί να προσομοιώσει τη διάχυση του φωτός στην ατμόσφαιρα. Αυτό μπορεί να βελτιώσει σημαντικά την οπτική ποιότητα και τη ρεαλιστικότητα μιας σκηνής, ιδιαίτερα σε εξωτερικά ή ατμοσφαιρικά περιβάλλοντα. Όσον αφορά το φωτισμό HDR που υποστηρίζει ,με αυτόν μπορεί παρέχει μεγαλύτερο εύρος φωτεινότητας και χρώματος.Βεβαια με την HDRP δε μπορούμε να απευθυνόμαστε σε συσκευές χαμηλών επιδόσεων όπως στη πλειοψηφία των κινητών τηλεφώνων.

Η Built-in Render Pipeline έχει σχεδιαστεί για να παρέχει μια ισορροπία μεταξύ απόδοσης και ποιότητας. Η Built-in υποστηρίζει ένα ευρύ φάσμα πλατφορμών και συσκευών, συμπεριλαμβανομένων των κινητών συσκευών και των υπολογιστών χαμηλών επιδόσεων. Αυτό συμβαίνει γιατί έχει σχεδιαστεί για να είναι ελαφριά και αποδοτική, ούτως ώστε να μπορεί να παρέχει γραφικά υψηλής ποιότητας ακόμα και σε συσκευές χαμηλών προδιαγραφών. Αυτό την καθιστά ιδανική επιλογή για προγραμματιστές που θέλουν τα παιχνίδια τους να είναι προσβάσιμα σε όσο το δυνατόν περισσότερους χρήστες. Η Built-in παρέχει μια σειρά από επιλογές προσαρμογής που επιτρέπουν στους προγραμματιστές να δημιουργήσουν μοναδικά οπτικά εφέ που δεν είναι διαθέσιμα αλλού.



Η Built-in προσφέρει επίσης μια σειρά από άλλα χαρακτηριστικά που την καθιστούν δημοφιλή επιλογή για προγραμματιστές. Για παράδειγμα, υποστηρίζει δυναμικό φωτισμό και εφέ μετα-επεξεργασίας (post-processing effects), τα οποία μπορούν να βελτιώσουν σημαντικά την οπτική ποιότητα ενός παιχνιδιού.

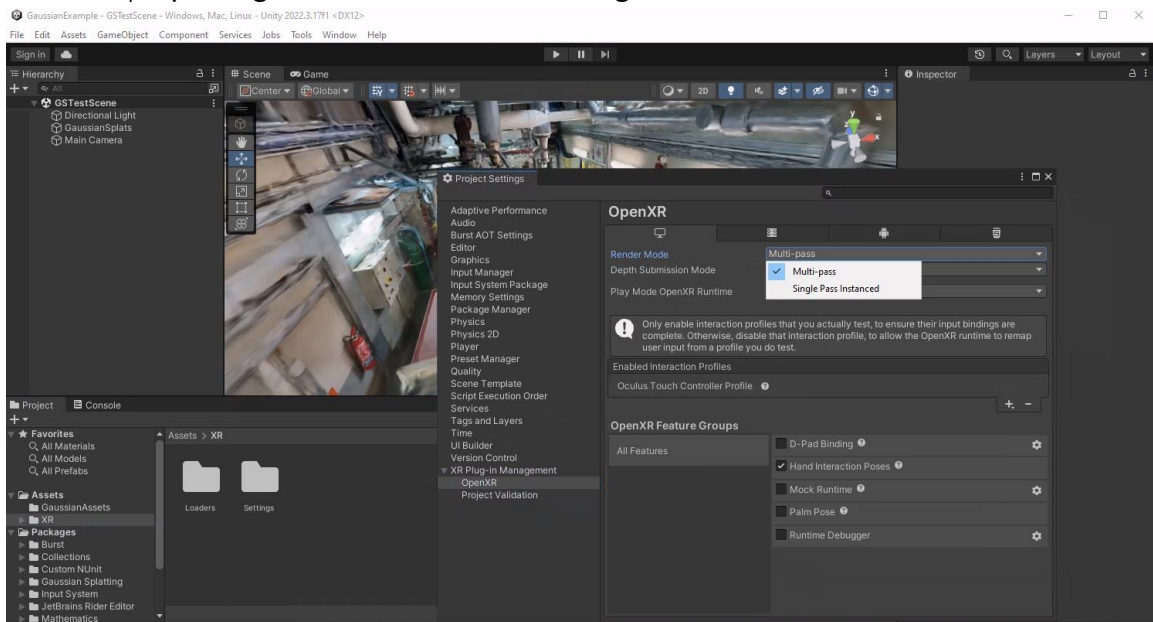
## 6.3 Gaussian splatting και Virtual Reality στο περιβάλλον Unity

Χρησιμοποιώντας λοιπόν το έτοιμο project για το Unity που έχει αναπτυχθεί από τον χρήστη “aras-p” θα προσπαθήσουμε να προετοιμάσουμε την τρισδιάστατη αναπαράσταση ούτως ώστε να μπορούμε να περιηγηθούμε στο χώρο που αναπαραστήσαμε με ένα VR Headset. Για να το πέτυχουμε αυτό πρώτα από όλα θα εισάγουμε το project μας στο Unity από το Package Manager το OpenXR Plugin και το XR Plugin Management. Τι είναι όμως το OpenXR ?

Το OpenXR είναι ένα ανοιχτού κώδικα, χωρίς δικαιώματα πρότυπο που αναπτύχθηκε από την Khronos Group και παρέχει μια ενοποιημένη διεπαφή για εφαρμογές εικονικής πραγματικότητας (VR) και επαυξημένης πραγματικότητας (AR), οι οποίες μαζί αναφέρονται ως εκτεταμένη πραγματικότητα ή XR (Extended Reality). Ο σκοπός του OpenXR είναι να απλοποιήσει την ανάπτυξη λογισμικού XR παρέχοντας ένα ενιαίο API που μπορεί να χρησιμοποιηθεί σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα.

Προχωρώντας λοιπόν στην προετοιμασία της τρισδιάστατης αναπαράστασης αφού ελέγξουμε και επιλύσουμε με αυτόματο τρόπο τα προβλήματα που βγάζει στο Project Validation το Unity, θα κάνουμε και ορισμένες επιλογές που αφορούν τόσο τη μέθοδο απόδοσης (αυτό θα το αναλύσουμε σε λίγο) όσο και άλλα χαρακτηριστικά που αφορούν κυρίως το VR Headset που θα χρησιμοποιήσουμε και τις λειτουργίες που θέλουμε να έχει. Στο Unity, κατά την ανάπτυξη εφαρμογών XR (Εκτεταμένης Πραγματικότητας), υπάρχουν διάφορες μέθοδοι απόδοσης (render modes) που μπορούμε να χρησιμοποιήσουμε ανάλογα με τις ανάγκες μας, τις δυνατότητες της εκάστοτε συσκευής και το είδος της εμπειρίας XR που θέλουμε να δημιουργήσουμε. Στη περίπτωση μας μας δίνεται η επιλογή ανάμεσα σε δύο στη Single-Pass Instanced Rendering και στη Multi-Pass Rendering. Η Single-Pass Instanced Rendering, η οποία είναι η πιο συνηθισμένη μέθοδος απόδοσης για εφαρμογές XR, επιτρέπει στο Unity να αποδώσει τη σκηνή και για τα δύο μάτια σε μία μόνο διαδικασία βελτιώνοντας την απόδοση. Αυτό συμβαίνει καθώς απαιτεί μόνο ένα σετ εντολών για να αποδοθεί η σκηνή και για τα δύο μάτια, και λειτουργεί καλά με τα περισσότερα σύγχρονα VR headsets. Αντίθετα με τη Multi-Pass Rendering το Unity αποδίδει τη σκηνή δύο φορές, μία για κάθε μάτι. Αυτή είναι η απλούστερη μέθοδος απόδοσης και είναι συμβατή με όλες τις συσκευές που υποστηρίζουν XR αφού είναι πιο εύκολη στην υλοποίηση αλλά είναι λιγότερο αποδοτική από τις μεθόδους single-pass.

Στη περίπτωση μας δοκιμάσαμε και τις δύο μεθόδους όμως αυτό που παρατηρήσαμε είναι πως στη Single-Pass Instanced Rendering όταν δοκιμάζαμε να δούμε τη τρισδιάστατη αναπαράσταση με το VR Headset είναι πως το δεξί μάτι στο VR Headset δε λειτουργούσε οπότε αλλάξαμε σε Multi-Pass Rendering και όλα λειτουργούσαν άψογα. Το αποτέλεσμα ήταν άκρως ρεαλιστικό και ζωντανό αλλά αξίζει να σημειωθεί πως για να μπορείς να κινείσαι ομαλά, να μη κολλάς και να τα βλέπεις όλα σαν να βρίσκεσαι εκείνη τη στιγμή στον πραγματικό χώρο χρειάζεσαι και μια πολύ καλή κάρτα γραφικών για να μπορούν να αυξηθούν όσο το δυνατόν περισσότερο τα frames per second(fps).Βέβαια οι απαιτήσεις για καλή κάρτα γραφικών θα μπορούσαν να μειωθούν σημαντικά αν μπορούσε να δουλέψει με Single-Pass Instanced Rendering.



Εικόνα 38: Επιλογές OpenXR στο Unity

Αυτό που αξίζει να σημειωθεί ακόμα είναι πως όταν ανοίγουμε το project που έχει δημιουργηθεί σε Universal Render Pipeline και προσπαθούμε να περιηγηθούμε στο χώρο με το VR Headset παρατηρούμε πως τα δυο «μάτια» είναι τοποθετημένα ανάποδα δηλαδή ότι θα πρέπει να βλέπουμε στο αριστερό το βλέπουμε στο δεξί και το ανάποδο. Ενώ, όταν χρησιμοποιούμε το project που έχει δημιουργηθεί σε Built-in Render Pipeline η χρήση του VR Headset και η περιήγηση στην τρισδιάστατη αναπαράσταση γίνεται απροβλημάτιστα.

Το VR Headset που χρησιμοποιήσαμε για τις δοκιμές είναι το Meta Quest 3, το τελευταίο μοντέλο της σειράς αυτόνομων VR headsets από την Meta, το οποίο φέρνει αρκετές βελτιώσεις σε σχέση με τον προκάτοχό του. Γενικά το Meta Quest 3 προσφέρει μια πιο ολοκληρωμένη και αναβαθμισμένη εμπειρία VR, ιδανική τόσο για τους νέους χρήστες όσο και για τους έμπειρους χρήστες. Με την αυξημένη ανάλυση, τις βελτιώσεις στην άνεση και τις δυνατότητες Mixed Reality, το Quest 3 προωθεί τα όρια του αυτόνομου VR και καθιστά την εμπειρία πιο προσιτή και καθηλωτική από ποτέ

## 7. Επίλογος

Με λίγα λόγια λοιπόν σε αυτή τη διπλωματική εργασία προσπαθήσαμε και θέλουμε να ελπίζουμε πως καταφέραμε να εξηγήσουμε με σαφήνεια πως λειτουργεί η τρισδιάστατη αναπαράσταση με Gaussian splatting και ποια είναι τα πλεονεκτήματα της έναντι των άλλων μεθόδων τρισδιάστατης αναπαράστασης. Ακόμα, αναλύσαμε τον τρόπο λειτουργίας του COLMAP που προετοιμάζει τα δεδομένα εισόδου για την τρισδιάστατη αναπαράσταση αλλά και του plug-in του COLMAP-PCD με το οποίο πέτυχαμε να κάνουμε image to point cloud registration στο χώρο του εργαστηρίου ναυτικής μηχανολογίας. Επιπλέον, δημιουργήσαμε ένα κώδικα με τον οποίο μπορέσαμε να δημιουργούμε μονοί μας τα δεδομένα εισόδου παραλείποντας το COLMAP. Τέλος, εισαγάγαμε την τρισδιάστατη αναπαράσταση στο Unity και την προετοιμάσαμε με κατάλληλο τρόπο ώστε κάποιος να μπορεί να περιηγηθεί μέσα σε αυτή με ένα VR Headset.

Είναι αδιαμφισβήτητο πως η τρισδιάστατη αναπαράσταση με Gaussian splatting είναι μια καινοτόμα λύση στον τομέα αυτό και έχει πολύ χώρο για βελτίωση. Αρχικά, το πιο σημαντικό είναι να μειωθούν οι απαιτήσεις σε υπολογιστική ισχύ και ιδιαίτερα η ανάγκη για μια τόσο ισχυρή και ταυτόχρονα ακριβή κάρτα γραφικών. Κάτι τέτοιο θα μπορούσε να συμβεί με σημαντικές αλλαγές στη βάση του κώδικα ούτως ώστε να απαιτείται λιγότερη δέσμευση μνήμης. Ακόμα στη περίπτωση που είχαμε ένα πιο πρόσφατο point cloud που ανταποκρινόταν επακριβώς στο χώρο που αναπαριστούσαμε θα μπορούσαμε να απενεργοποιήσουμε και το self-densify και να κερδίσουμε και έτσι σε απόδοση. Επιπλέον, θα ήταν ενδιαφέρον να γίνει επιπλέον μελέτη στον τρόπο που γίνεται το image to point cloud registration με το COLMAP-PCD και στο πως αυτό μπορεί να γίνει πιο σύντομο και πιο αποτελεσματικό, αν για παράδειγμα του δοθεί από πριν μια λογική σύνδεσης μεταξύ των εικόνων. Όσον αφορά το Unity εκεί το εύρος των δυνατοτήτων που μας δίνεται για ερευνά είναι τεράστιο. Από το να βελτιώσουμε τις σκιάς, τις αντανακλάσεις και το φωτισμό μέχρι να επεξεργαστούμε την ίδια την αναπαράσταση αφαιρώντας τα θολά σημεία που δημιουργούνται ή ακόμα και να προσθέσουμε σε αυτήν την αναπαράσταση επιπλέον πληροφορία. Τέλος, θα ήταν χρήσιμο να βρούμε τρόπους με τους οποίους θα μπορέσει να γίνει πιο συμβατό στο χώρο της εικονικής πραγματικότητας και να λειτουργεί και εκεί με λιγότερες απαιτήσεις σε υπολογιστική ισχύ, για παράδειγμα ξεκινώντας με το να βρούμε έναν τρόπο να γίνει με Single-Pass Instanced Rendering.

Γενικά, το θέμα της διπλωματικής ήταν πολύ ευρύ και όπως καταλάβατε προσπαθήσαμε να εξηγήσουμε και να αναλύσουμε τα πιο βασικά από διάφορες πτυχές του θέματος εμβαθύνοντας λίγο περισσότερο σε ότι μας κέντριζε το ενδιαφέρον όπως στο image to point cloud registration το οποίο είναι χρήσιμο για χώρους πλοίων που θέλουμε να αναπαραστήσουμε τρισδιάστατα και με ρεαλιστικό τρόπο. Σίγουρα θα ήταν ενδιαφέρον στα επόμενα χρονιά να δούμε ορισμένα από τα ζητήματα που αντιμετωπίσαμε να βελτιώνονται και να επιλύονται με ευφάνταστους τρόπους και να πετυχαίνουμε με ελάχιστες απαιτήσεις ένα τόσο ρεαλιστικό αποτέλεσμα που θα το μπερδεύουμε με το πραγματικό.

## Βιβλιογραφία

- 1) 3D Gaussian Splatting <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- 2) COLMAP <https://colmap.github.io/index.html>
- 3) Unity <https://unity.com/>
- 4) Gaussian Splatting playground in Unity  
<https://github.com/aras-p/UnityGaussianSplatting>
- 5) Guide for beginners: What is Scanline Rendering?  
<https://viscircle.de/guide-for-beginners-what-is-scanline-rendering/?lang=en>
- 6) Scanline Rendering  
<https://academic-accelerator.com/encyclopedia/scanline-rendering>
- 7) Raycasting <https://lodev.org/cgtutor/raycasting.html>
- 8) Raycasting  
<https://people.computing.clemson.edu/~dhouse/courses/405/notes/raycast.pdf>
- 9) What's the Difference Between Ray Tracing, Ray Casting, and Ray Charles?  
<https://www.electronicdesign.com/technologies/industrial/displays/article/21801219/whats-the-difference-between-ray-tracing-ray-casting-and-ray-charles>
- 10) Z-Buffer or Depth-Buffer method  
<https://www.geeksforgeeks.org/z-buffer-depth-buffer-method/>
- 11) Z-buffering <https://www.computerhope.com/jargon/z/zbuffering.htm>
- 12) Fast, Low Memory Z-buffering when performing Medium-Quality Rendering  
[https://www.realtimerendering.com/erich/haines\\_Fast\\_Low\\_Memory\\_Z-Buffer\\_jgt\\_v1\\_n3.pdf](https://www.realtimerendering.com/erich/haines_Fast_Low_Memory_Z-Buffer_jgt_v1_n3.pdf)
- 13) Rasterization: Z-buffer Algorithm  
<https://eccarrilloe.github.io/2017/09/24/Rasterization-Z-buffer-Algorithm/>
- 14) Painter's Algorithm in Computer Graphics  
<https://www.geeksforgeeks.org/painters-algorithm-in-computer-graphics/>
- 15) Painter's Algorithm [https://en.wikipedia.org/wiki/Painter%27s\\_algorithm](https://en.wikipedia.org/wiki/Painter%27s_algorithm)
- 16) Spherical harmonics [https://en.wikipedia.org/wiki/Spherical\\_harmonics](https://en.wikipedia.org/wiki/Spherical_harmonics)
- 17) Point-based rendering techniques  
<https://www.sciencedirect.com/science/article/pii/S0097849304001530>
- 18) Point-Based graphics  
<https://ls7-gv.cs.tu-dortmund.de/research/points/points.html>
- 19) Efficient Point-Based Rendering Using Image Reconstruction  
<https://graphics.tudelft.nl/~marroquim/publications/pdfs/marroquim-pbg2007.pdf>
- 20) Point cloud [https://en.wikipedia.org/wiki/Point\\_cloud](https://en.wikipedia.org/wiki/Point_cloud)
- 21) Point cloud <https://www.gigabyte.com/Glossary/point-cloud>
- 22) Point Cloud <https://www.sciencedirect.com/topics/computer-science/point-cloud>
- 23) What are point clouds? 5 easy facts that explain point clouds  
<https://info.vercator.com/blog/what-are-point-clouds-5-easy-facts-that-explain-point-clouds>
- 24) Point Clouds 6 - Scan-to-BIM  
<https://www.bricsys.com/es-pe/blog/point-clouds-6-scan-to-bim>

- 25) MERF: Memory-Efficient Radiance Fields for Real-time View Synthesis in Unbounded Scenes  
<https://arxiv.org/pdf/2302.12249.pdf>
- 26) Volume Texture <https://docs.unrealengine.com/4.26/en-US/RenderingAndGraphics/Textures/VolumeTextures/>
- 27) CUDA Refresher: The CUDA Programming Model  
<https://developer.nvidia.com/blog/cuda-refresher-cuda-programming-model/>
- 28) CUDA <https://hpc.it.auth.gr/cuda-intro/>
- 29) Sample Representation  
[https://docs.nerf.studio/nerfology/model\\_components/visualize\\_samples.html#d-frustum](https://docs.nerf.studio/nerfology/model_components/visualize_samples.html#d-frustum)
- 30) Exploring the use of 3D GIS as an analytical tool in archaeological excavation practice  
[https://www.researchgate.net/publication/303824023\\_Exploring\\_the\\_use\\_of\\_3D\\_GIS\\_as\\_an\\_analytical\\_tool\\_in\\_archaeological\\_excavation\\_practice](https://www.researchgate.net/publication/303824023_Exploring_the_use_of_3D_GIS_as_an_analytical_tool_in_archaeological_excavation_practice)
- 31) A beginner's guide to pinhole photography  
<https://news.smugmug.com/a-beginners-guide-to-pinhole-photography-ce7cf6b06fcb>
- 32) Understanding Quaternions <https://www.3dgep.com/understanding-quaternions/>
- 33) Quaternion <https://en.wikipedia.org/wiki/Quaternion>
- 34) Focal Length and Intrinsic Camera Parameters  
<https://www.baeldung.com/cs/focal-length-intrinsic-camera-parameters>
- 35) Unity: Understanding URP, HDRP, and Built-In Render Pipeline  
<https://www.occasoftware.com/blog/unity-understanding-urp-hdrp-built-in>
- 36) Open XR <https://www.khronos.org/openxr/>
- 37) Meta Quest 3 <https://www.meta.com/quest/quest-3/>
- 38) The Single-Pass VS Multi-Pass rendering in Unity for VR  
<https://themaxscriptguy.medium.com/the-single-pass-vs-multi-pass-rendering-in-unity-for-vr-4962ac30ceb9>
- 39) RAY-CASTING AND RAY-TRACING  
<https://www.permadi.com/tutorial/raycast/rayc2.html>
- 40) Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis [2308.09713] [Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis \(arxiv.org\)](https://arxiv.org/abs/2308.09713)
- 41) EWA Volume Splatting  
<https://www.cs.umd.edu/~zwicker/publications/EWAVolumeSplatting-VIS01.pdf>
- 42) NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis  
<https://arxiv.org/abs/2003.08934>
- 43) A Comprehensive Overview of Gaussian Splatting  
<https://towardsdatascience.com/a-comprehensive-overview-of-gaussian-splatting-e7d570081362>
- 44) COLMAP\_PCD <https://github.com/XiaoBaiiiii/colmap-pcd>