

Διπλωματική Εργασία

Υλοποίηση Αλγορίθμων Κίνησης Αρθρωτών Χαρακτήρων

Γκόλες Βασίλης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ



Πολυτεχνική Σχολή

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Ιωάννινα, Οκτώβριος 2017

Λέξεις κλειδιά: προσδιορισμός περιβλήματος, σκελετική παραμόρφωση SSD, linear blend skinning LBS, απλοποίηση μοντέλου, τετραγωνικό σφάλμα.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ.....	1
1.1 Σχετική Εργασία.....	1
ΚΕΦΑΛΑΙΟ 2 ΜΑΘΗΜΑΤΙΚΟ ΥΠΟΒΑΘΡΟ.....	2
2.1 Μετασχηματισμοί Πινάκων.....	2
2.1.1 Γραμμικοί Μετασχηματισμοί.....	2
2.1.2 Συγγενής Μετασχηματισμοί.....	10
2.1.3 Αντίστροφοι Μετασχηματισμοί.....	11
2.2 Quaternions (Τετραδόνια).....	12
2.2.1 Επισκόπηση.....	12
2.2.2 Τι είναι τα Quaternions.....	12
2.2.3 Ιδιότητες Quaternions.....	12
2.3 Στοιχεία Θεωρίας Επιπέδου.....	16
2.3.1 Καθορισμός ενός Σημείου και ενός Διανύσματος.....	16
2.3.2 Εξίσωση Επιπέδου και Πίνακας Τετραγωνικής Απόστασης.....	16
2.3.3 Ποσότητες Παραμόρφωσης Επιπέδου.....	17
2.4 Σύστημα Βαρυκεντρικών Συντεταγμένων.....	18
2.4.1 Τι είναι Βαρυκεντρικές Συντεταγμένες.....	18
2.4.2 Βαρυκεντρικές Συντεταγμένες Τριγώνου.....	19
ΚΕΦΑΛΑΙΟ 3 ΣΧΗΜΑΤΙΚΗ ΠΑΡΑΜΟΡΦΩΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ (SSD).....	20
3.1 Μοντέλα Με Σκελετική Ιεραρχία.....	20
3.1.1 Ιεραρχική Μοντελοποίηση.....	20
3.1.2 Ιεραρχικός Μετασχηματισμός Αντικειμένων.....	21
3.2 Μέθοδος Προσδιορισμού Περιβλήματος.....	22
3.3 Υπολογισμός Βαρών Κορυφών.....	24
3.4 Linear Blend Skinning.....	25
3.4.1 Υπολογισμός Θέσεων των Κορυφών και Εφαρμογή της LBS.....	26
3.5 Αποτελέσματα.....	27

ΚΕΦΑΛΑΙΟ 4 ΒΕΛΤΙΩΣΗ ΑΠΟΔΟΣΗΣ ΜΕΣΩ ΑΠΛΟΠΟΙΗΣΗΣ.....	31
4.1 Εισαγωγή.....	31
4.2 Απλοποίηση Επιφάνειας.....	32
4.3 Απλοποίηση Μέσω Συνένωσης Κορυφών.....	35
4.3.1 Επιλογή Ζεύγους.....	36
4.4 Προσέγγιση Με Χρήση Τετραγωνικού Σφάλματος	36
4.4.1 Περίληψη Αλγορίθμου.....	39
4.4.2 Χρονική Πολυπλοκότητα Αλγορίθμου.....	39
ΚΕΦΑΛΑΙΟ 5 ΥΛΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	41
5.1 Λεπτομέρειες Υλοποίησης.....	41
5.2 Αποτελέσματα.....	41
5.3 Μελλοντικές Επεκτάσεις.....	43
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	44

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 2.1: Περιστροφή ως προς τον άξονα z κατά θ radians.....	4
Εικόνα 2.2: Κλιμάκωση σύμφωνα τα a,b,c.....	6
Εικόνα 2.3: Κλιμάκωση κύβου.....	7
Εικόνα 2.4: Διάτμηση ως προς τον άξονα x.....	8
Εικόνα 2.5: Διάτμηση ως προς τον άξονα y.....	9
Εικόνα 2.6: Διάτμηση ως προς τον άξονα z.....	10
Εικόνα 2.7: Μετατόπιση σημείου κατά a,b,c.....	11
Εικόνα 2.8: Επίπεδο στο χώρο.....	16
Εικόνα 2.9: Παραμόρφωση όψης σε μία ακολουθία κινήσεων ενός τριγωνισμένου μοντέλου.....	18
Εικόνα 2.10: Βαρυκεντρικές συντεταγμένες σε σχέση με το τρίγωνο.....	19
Εικόνα 3.1: Ιεραρχικό μοντέλο αναπαράστασης.....	20
Εικόνα 3.2: Ιεραρχικό μοντέλο αναπαράστασης μετασχηματισμού.....	21
Εικόνα 3.3: Αρθρωτό μοντέλο ενσωματωμένο σε εξωτερικό περίβλημα.....	23
Εικόνα 3.4: Αρθρωτά μοντέλα με αναπαράσταση χρωματικής κωδικοποίησης.....	24
Εικόνα 3.5: Απεικόνιση θερμικής αντιστοίχισης μεταξύ δύο άκρων.....	25
Εικόνα 3.6: Απεικόνιση χρωματικής κωδικοποίησης άκρων πάνω σένα ζώο.....	25
Εικόνα 3.7: Μοντελοποίηση δύο άκρων.....	26
Εικόνα 3.8: Απεικόνιση νέα θέσης κορυφής μετά τον μετασχηματισμό.....	27
Εικόνα 3.9: Απεικόνιση παραμόρφωσης μετά τον μετασχηματισμό.....	28
Εικόνα 3.10: Αναπαράσταση bind pose.....	28
Εικόνα 3.11: Αναπαράσταση μετασχηματισμένου μοντέλου.....	28
Εικόνα 3.12: Απεικόνιση διαφόρων στιγμιότυπων της εφαρμογής μας.....	29
Εικόνα 3.13. Αγκώνας που παραμορφώνεται.	30
Εικόνα 3.14. Ο βραχίονας στον καρπό καθώς υπολογίζεται από τον LBS. Καθώς ο καρπός πλησιάζει τις 180° ο βραχίονας παραμορφώνεται.	30
Εικόνα 4.1: Μία κορυφή αφαιρείται και η προκύπτουσα οπή τριγωνοποιείται.....	33
Εικόνα 4.2: Ομοιόμορφη ομαδοποίηση στις δύο διαστάσεις.....	34
Εικόνα 4.3: Απαλοιφή ή Συνένωση Ακμής. Η υπογραμμισμένη ακμή συμπτύσσεται σε ένα μοναδικό σημείο.	35

Εικόνα 4.4: Συνένωση δύο μη συνεκτικών περιοχών του μοντέλου. Οι διακεκομμένες γραμμές δείχνουν ότι οι δύο κόμβοι έχουν ενωθεί σε έναν.	35
Εικόνα 4.5: Οπτική απεικόνιση τετραγωνικού σφάλματος.....	38
Εικόνα 5.1: Screenshot της γραφικής διεπαφής της εφαρμογής μας.....	41
Εικόνα 5.2 Ακολουθία κίνησης ανθρωπόμορφου χαρακτήρα.	41
Εικόνα 5.3 Ο χρόνος που απαιτείται για την απλοποίηση του μοντέλου που φαίνεται στην εικόνα 5.1.....	42
Εικόνα 5.4 Γράφημα το οποίο απεικονίζει το χρόνο που απαιτείται από τον LBS για να υπολογίζει τις θέσεις των κορυφών στην CPU και στην GPU αντίστοιχα σε κάθε frame.....	42

ΕΚΤΕΝΗΣ ΠΕΡΙΛΗΨΗ

Γκόλες Βασίλης του Γιώργου και της Γεωργίας

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Οκτώβριος, 2017.

Υλοποίηση Αλγορίθμων Κίνησης Αρθρωτών Χαρακτήρων

Επιβλέπων Καθηγητής: Ιωάννης Φούντος

Η υλοποίηση κίνησης ιεραρχικών και μη μοντέλων τα οποία μπορεί να περιλαμβάνουν χαρακτήρες όπως (ανθρώπους, ρομπότ, η ανθρωπόμορφους χαρακτήρες) ζώα όπως (σκυλιά, γάτες η αράχνες) μηχανικές συσκευές όπως (ρολόγια, τρίκυκλα) και ούτω καθεξής στα γραφικά υπολογιστών απαιτούν μεθόδους οι οποίες αποδίδουν αποτελέσματα υψηλής ποιότητας σε πραγματικό χρόνο.

Στην εργασία αυτή παρουσιάζουμε μία μέθοδο προσδιορισμού περιβλήματος (skinning) σε αρθρωτούς χαρακτήρες για την απόδοση κίνησης με την μικρότερη δυνατή παραμόρφωση και τον προσδιορισμό βαρών των αρθρώσεων μετά από σημαντική απλοποίηση του αρχικού μοντέλου. Ο προσδιορισμός των βαρών στα παραμορφωμένα μοντέλα που παράγονται γίνεται με τέτοιο τρόπο ώστε να τα νέα βάρη να αποδίδουν μια όσο πιο φυσιολογική κίνηση στα σημεία του μοντέλου με την ελάχιστη δυνατή παραμόρφωση.

Αρχικά παραθέτουμε το μαθηματικό υπόβαθρο που απαιτείται για την κατανόηση των μεθόδων που παρουσιάζουμε. Στην συνέχεια παρουσιάζουμε την επικρατούσα μέθοδο προσδιορισμού περιβλήματος για αρθρωτούς χαρακτήρες την οποία μπορούμε να κάνουμε και αναγωγή σε μη αρθρωτούς χαρακτήρες και στην συνέχεια παραθέτουμε μία μέθοδο απλοποίησης ταυτόχρονα και συμπίεσης τόσο αρθρωτών αλλά και μη μοντέλων η οποία παρέχει όσο το δυνατό καλύτερη προσέγγιση. Υποδεικνύουμε αρχές για αποδοτική διάδοση βαρών επιρροής, στα απλοποιημένα μοντέλα για επίτευξη κίνησης με το μικρότερο δυνατό σφάλμα. Τέλος υπάρχει μια σύντομη περιγραφή της εφαρμογής μας, αποτελέσματα των μετρήσεών μας, συμπεράσματα και επεκτάσεις.

EXTENSIVE SUMMARY IN ENGLISH

Goles, Vasilis, G

BSc, Computer Science and Engineering Department, University of Ioannina, Greece October 2017

Implementing Algorithms For Skinning Articulated Characters

Thesis Supervisor: Ioannis Fudos

Implementing animation based on hierarchical models which may include humanoid characters (such as people, robots, or aliens), animals (such as dogs, cats, or spiders), mechanical devices (watches, tricycles), and so on in computer graphics require methods that deliver high-quality results in real time.

In this paper we implement a skeletal subspace deformation method, a method for attaching skin to a hierarchical skeleton which naturally deforms when we manipulate the skeletons joints angles, or during animation, also we implement a mesh simplification method based on M.Garland and P.S.Heckbert[14] method where someone can apply a significant simplification to the original model and determine new weights for the distorted model. We determine the new weights values in such a way that they provide minimal distortion when we manipulate or animate the deformed model.

First, we present the mathematical background needed to understand the methods we present. Then we present the dominant method, for skeletal subspace deformation also known as Linear Blend Skinning(LBS), a method which can be converted to apply also to no non-skeletal characters. After that, we present a method for both simplifying and compressing articulated and not-articulated models which provides quite good results. Finally, there is a brief description of our application, results of our measurements, conclusions and possible future extensions.

1.1 Σχετική Εργασία

1.1 Σχετική Εργασία

Στην παρούσα εργασία σε πρώτη φάση υλοποιήσαμε τεχνικές και αλγορίθμους σχετικά με τους ιεραρχικούς μετασχηματισμούς και τον προσδιορισμό περιβλήματος. Το τελικό αποτέλεσμα αυτού είναι ένας αρθρωτός χαρακτήρας με τον οποίο μπορούμε να αλληλεπιδράσουμε είτε με κουμπιά, θέτοντας τα άκρα του σε κάποια συγκεκριμένη θέση ανά καρτέ και αυτό γίνεται χρησιμοποιώντας πραγματικό περιβλήμα, είτε αυτόματα παράγοντας προκαθορισμένη ακολουθία κινήσεων. Θα χρησιμοποιούμε τους όρους άκρο και άρθρωση εναλλακτικά. Πιο συγκεκριμένα έγινε υλοποίηση του αλγορίθμου LBS τόσο στην CPU όσο και στην GPU. Γι' αυτό χρησιμοποιήσαμε τρία αρχεία τα οποία είναι το αρχείο με την γεωμετρία του μοντέλου μας το οποίο καταλήγει σε .obj το αρχείο με τα βάρη το οποίο καταλήγει σε .attach και το αρχείο που περιγράφει τις θέσεις των άκρων και καταλήγει σε .skel. Επίσης όταν πρόκειται για αυτόματα παραγωγή κίνησης χρησιμοποιούμε το αρχείο που καταλήγει σε .bvh το οποίο περιγράφει την ακολουθία κίνησης των αρθρώσεων. Για καλύτερη διαχείριση κάποιων λειτουργιών των γραφικών και την γραφική διεπαφή χρησιμοποιήσαμε ένα σύνολο έτοιμων μεθόδων (framework) από την Nvidia Research το οποίο ενσωματώνουμε στον κώδικα μας με τη χρήση του "using namespace FW". Από αυτό το framework χρησιμοποιήσαμε κυρίως το μέρος εκείνο που περιλαμβάνει την βιβλιοθήκη Math.hpp κα έχει μεθόδους όπως Vec3f, Mat4f για πράξεις με διανύσματα και πίνακες μεγέθους 4x4. Επίσης στη φάση αυτή η ροή της εργασίας και οι συμβολισμοί που ακολουθήθηκαν βασίστηκε κατά κύριο λόγο στις διαφάνειες του Jaakko Lehtinen από το μάθημα CS-C3100 – Computer Graphics 2016. Η πλατφόρμα υλοποίησης που χρησιμοποιήσαμε είναι η Microsoft Visual Studio 2015 και ως γλώσσα προγραμματισμού χρησιμοποιήσαμε την C++, επίσης έγινε χρήση της γλώσσας GL Shading Language για την υλοποίηση του αλγορίθμου LBS.

Σε δεύτερη φάση υλοποιήσαμε αλγόριθμους για την απλοποίηση πολυγωνικών επιφανειών. Και συγκεκριμένα υλοποιήσαμε τον αλγόριθμο απλοποίησης των Garland και Heckbert[14] για να απλοποιήσουμε τα μοντέλα μας. Εδώ ο χρήστης μπορεί να φορτώσει ένα πολυγωνικό μοντέλο, να κάνει την απλοποίηση πατώντας κάποιο κουμπί και να αποθηκεύσει το νέο απλοποιημένο μοντέλο το οποίο μπορεί να επεξεργαστεί ξανά. Εδώ χρησιμοποιήσαμε ξανά το framework της Nvidia Research για υπολογισμούς με πίνακες. Ο κώδικας για την φόρτωση και απεικόνιση στην οθόνη των μοντέλων μας είναι ίδιος με αυτόν της πρώτης φάσης. Η ροή και οι συμβολισμοί που ακολουθήθηκαν βασίζονται σύμφωνα με τους [14] [15]. Σε αυτή τη φάση ο χρήστης διαβάζει μόνον τα αρχεία που περιγράφουν την γεωμετρία του μοντέλου και καταλήγουν σε .obj και το αρχείο που περιέχει τα βάρη και μπορεί να δημιουργήσει πάλι τέτοια αρχεία για μετέπειτα επεξεργασία. Παρόμοια και εδώ χρησιμοποιήσαμε ως πλατφόρμα ανάπτυξης την Visual Studio 2015 και ως γλώσσα προγραμματισμού την C++.

Τέλος στο τελευταίο κεφάλαιο έγιναν κάποια πειράματα σχετικά με την υλοποίηση του LBS στην CPU και την GPU για το πόσο γρήγορα υπολογίζει τη νέα θέση της κορυφής, και του αλγορίθμου απλοποίησης με βάση διάφορες πολιτικές.

ΚΕΦΑΛΑΙΟ 2

ΜΑΘΗΜΑΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 Μετασχηματισμοί Πινάκων

2.2 Quaternions (Τετραδόνια)

2.3 Στοιχεία Θεωρίας Επιπέδου

2.4 Σύστημα Βαρυκεντρικών Συντεταγμένων

2.1 Μετασχηματισμοί Πινάκων

Σε αυτή την ενότητα θα περιγράψουμε συγκεκριμένες πτυχές της θεωρίας πινάκων και πώς οι πίνακες σχετίζονται με τους τρισδιάστατους μετασχηματισμούς.

2.1.1 Γραμμικοί Μετασχηματισμοί

Στην γραμμική άλγεβρα ως γραμμικό μετασχηματισμό ορίζουμε μια συνάρτηση η οποία απεικονίζει κάθε διάνυσμα ενός διανυσματικού χώρου \mathbb{R}^n σε ένα άλλο διάνυσμα του \mathbb{R}^n διατηρώντας τις γραμμικές ιδιότητες του χώρου \mathbb{R}^n . Με άλλα λόγια,

$$T: \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (2.1)$$

Ένας μετασχηματισμός για να είναι γραμμικός πρέπει να ισχύει ότι:

$$T(v + w) = T(v) + T(w)$$

και

$$T(cv) = cT(v), \quad (2.2)$$

για κάθε διάνυσμα v και w και για κάθε σταθερά c . Ισοδύναμα,

$$T(cv + dw) = cT(v) + dT(w) \quad (2.3)$$

για κάθε διάνυσμα v και w και για κάθε σταθερά c και d . Αξίζει να σημειωθεί ότι $T(0) = 0$, γιατί ένα δεν ίσχυε αυτό τότε δεν θα ήταν αλήθεια ότι $T(c0) = cT(0)$;

Δοθέντος ενός γραμμικού μετασχηματισμού T , πώς μπορούμε να κατασκευάσουμε έναν πίνακα A που αναπαριστά αυτό τον μετασχηματισμό;

Πρώτα, επιλέγουμε δύο διανύσματα βάσης, έστω v_1, v_2, \dots, v_n του \mathbb{R}^n έτσι ώστε να αναθέσουμε συντεταγμένες στα διανύσματα εισόδου και w_1, w_2, \dots, w_m του \mathbb{R}^m για να αναθέσουμε συντεταγμένες στα διανύσματα εξόδου. Θέλουμε να βρούμε έναν πίνακα A τέτοιο ώστε $T(v) = Av$, όπου τα v και Av παίρνουν συντεταγμένες από αυτά τα διανύσματα βάσης.

Η πρώτη στήλη του A αποτελείται από τους συντελεστές $a_{11}, a_{21}, \dots, a_{m1}$ του $T(v_1) = a_{11} w_1 + a_{21} w_2 + \dots + a_{m1} w_m$. Τα στοιχεία της στήλης i του πίνακα A προσδιορίζονται από τη σχέση $T(v_i) = a_{1i} w_1 + a_{2i} w_2 + \dots + a_{mi} w_m$. Επειδή έχουμε εξασφαλίσει ότι $T(v_i) = Av_i$ για κάθε διάνυσμα βάσης v_i και επειδή

ο T είναι γραμμικός μετασχηματισμός, ισχύει ότι $T(\mathbf{v}) = A\mathbf{v}$ για όλα τα διανύσματα \mathbf{v} του διανυσματικού χώρου εισόδου.

Στα γραφικά υπολογιστών, όλοι οι μετασχηματισμοί λαμβάνουν χώρα στις 3 διαστάσεις οπότε και εμείς θα εστιάσουμε την προσοχή μας σε αυτή τη διάσταση. Όταν οι γραμμικοί μετασχηματισμοί εφαρμόζονται πάνω σε αντικείμενα έχουν την ιδιότητα να διατηρούν τις αρχικές τους ιδιότητες όπως π.χ. διατηρούνται οι αποστάσεις, διατηρούνται οι γωνίες, οι παράλληλες γραμμές παραμένουν παράλληλες, τα επίπεδα παραμένουν επίπεδα κ.α.. Σύμφωνα το αποτέλεσμα που έχει ο κάθε μετασχηματισμός πάνω στο αντικείμενο που εφαρμόζετε οι γραμμικοί μετασχηματισμοί μπορούν να αναπαριστούν περιστροφή, κλιμάκωση, διάτμηση κ.α. με τα τελευταία να είναι τα πιο συχνά χρησιμοποιούμενα.

Περιστροφή

Επισκόπηση

Οι περιστροφές είναι πολύπλοκοι μετασχηματισμοί. Η κύρια πολυπλοκότητα είναι ότι στις τρεις διαστάσεις, οι περιστροφές εκτελούνται γύρω από έναν άξονα τον οποίο συνήθως καθορίζουμε από ένα σημείο και την κατεύθυνση ενός διανύσματος. Η γενική ιδέα είναι να αναπτυχθεί αυτός ο μετασχηματισμός σχετικά με τους τρεις άξονες συντεταγμένων και στη συνέχεια να γενικευθεί αυτό ώστε να μπορεί να εφαρμοστεί η περιστροφή γύρω από έναν αυθαίρετο άξονα. Η γενική ιδέα για τον μετασχηματισμό αυτό προέρχεται από την περιστροφή γύρω από ένα σημείο στις δύο διαστάσεις.

Μετασχηματισμός περιστροφής ως προς τον άξονα z

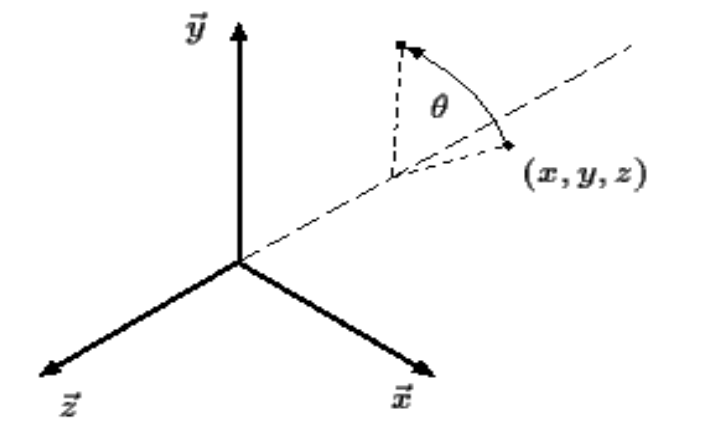
Ο μετασχηματισμός για περιστροφή κατά θ radians ως προς τον άξονα z δίνεται από τον παρακάτω 4 x 4 πίνακα

$$R_{z,\theta} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Αν εφαρμόσουμε τον μετασχηματισμό (2.4) σε ένα σημείο (x, y, z) , παίρνουμε:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x \cos \theta - y \sin \theta & x \sin \theta + y \cos \theta & z & 1 \end{bmatrix} \quad (2.5)$$

Μία αναπαράσταση αυτού απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 2.1: Περιστροφή ως προς τον άξονα z κατά θ radians [UC Davis Ken Joy]

Για έναν αυθαίρετο άξονα ο οποίος περιγράφεται από το διάνυσμα $n = [n_x n_y n_z]^T$, ο πίνακας που υλοποιεί την περιστροφή κατά θ radians γνωστός και ως τύπος Rodriguez δίνεται από τη σχέση:

$$R_{n;\theta} = \begin{bmatrix} n_x n_x (1 - c) + c & n_x n_y (1 - c) + n_z s & n_x n_z (1 - c) + n_y s & 0 \\ n_y n_x (1 - c) + n_z s & n_y n_y (1 - c) + c & n_y n_z (1 - c) + n_x s & 0 \\ n_z n_x (1 - c) + n_y s & n_z n_y (1 - c) + n_x s & n_z n_z (1 - c) + c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

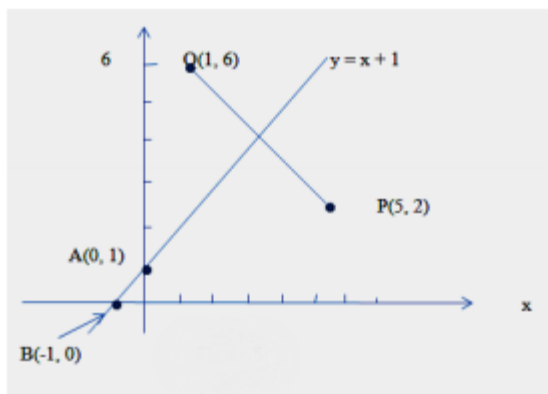
όπου $c = \cos \theta$ και $s = \sin \theta$

Αντανάκλαση

Επισκόπηση

Η αντανάκλαση είναι ένας απλό μετασχηματισμός. Οι απλές αντανακλάσεις αντικατοπτρίζουν το αντικείμενο ως προς τον άξονα x ως προς τον άξονα z και ως προς την αρχή των αξόνων. Η αντανάκλαση σε σχέση με μία αυθαίρετη γραμμή η οποία δίνεται από την σχέση ($y = ax + b$) είναι πιο πολύπλοκη.

Μία αναπαράσταση της αντανάκλασης φαίνεται στο παρακάτω σχήμα.



Το σημείο Q είναι αντανάκλαση του σημείου P ως προς την ευθεία $y = ax + b$ όπου στην περίπτωση μας το $a = 1$ και $b = 1$. [Wikipedia]

Ο πίνακας που μας δίνει την αντανάκλαση στις τρεις διαστάσεις δίνεται παρακάτω και προκύπτει από τον νόμο αντανάκλασης διανυσμάτων.

$$M := \begin{bmatrix} 1 - 2n_x^2 & -2n_x n_y & -2n_x n_z & 0 \\ -2n_x n_y & 1 - 2n_y^2 & -2n_y n_z & 0 \\ -2n_x n_z & -2n_y n_z & 1 - 2n_z^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Κλιμάκωση

Επισκόπηση

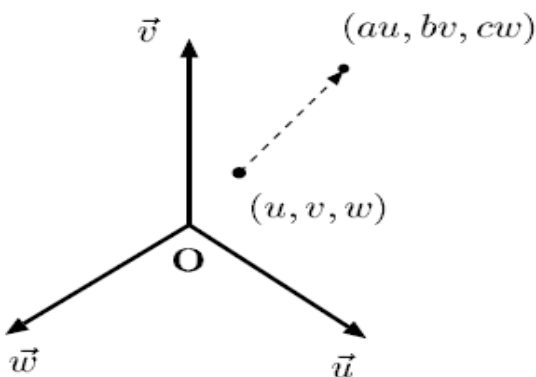
Η κλιμάκωση, είναι ένας γραμμικός μετασχηματισμός που απλά κλιμακώνει τις συντεταγμένες ενός αντικειμένου.

Εφαρμογή του μετασχηματισμού κλιμάκωσης στις τοπικές συντεταγμένες του αντικειμένου.

Δοθέντος ενός συστήματος συντεταγμένων και ενός σημείου \mathbf{P} το οποίο έχει τοπικές συντεταγμένες (u, v, w) , εάν εφαρμόσουμε τον μετασχηματισμό $S_{a,b,c}$ στις τοπικές συντεταγμένες αυτού του σημείου, παίρνουμε

$$[u \quad v \quad w \quad 1] \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [au \quad bv \quad cw \quad 1] \quad (2.7)$$

Μία αναπαράσταση αυτού απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 2.2: Κλιμάκωση σύμφωνα τα a, b, c [UC Davis Ken Joy]

Είναι δύσκολο να δούμε το σημείο εκκίνησης της κλιμάκωσης όταν εργαζόμαστε μόνον με συντεταγμένες, οπότε για παράδειγμα ας θεωρήσουμε τις 8 κορυφές ενός κύβου ο οποίος βρίσκεται τοποθετημένος στην αρχή των καρτεσιανών συντεταγμένων.

$$(-1, -1, 1)$$

$$(-1, 1, 1)$$

$$(1, 1, 1)$$

$$(1, -1, 1)$$

$$(-1, -1, -1)$$

$$(-1, 1, -1)$$

$$(1, 1, -1)$$

$$(1, -1, -1)$$

Ας θεωρήσουμε τον μετασχηματισμό κλιμάκωσης που δίνεται από τον παρακάτω πίνακα

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

και εφαρμόσουμε αυτό τον πίνακα σε κάθε συντεταγμένη των σημείων σε σχέση με το σύστημα συντεταγμένων το νέο σύνολο σημείων που παίρνουμε ισούται με:

$$(-2, -2, 2)$$

$(-2, 2, 2)$

$(2, 2, 2)$

$(2, -2, 2)$

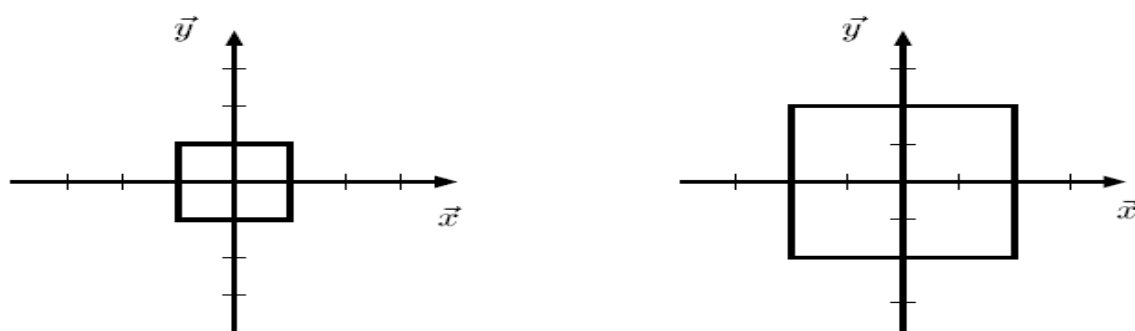
$(-2, -2, -2)$

$(-2, 2, -2)$

$(2, 2, -2)$

$(2, -2, -2)$

Το οποίο είναι αποτέλεσμα της κλιμάκωσης στον κύβο. (Ο παραγόμενος κύβος έχει όγκο 8 φορές τον όγκο του αρχικού κύβου). Μία αναπαράσταση αυτού δίνεται στην παρακάτω εικόνα (Σημειώνουμε ότι σε αυτή την εικόνα ο κύβος παρατηρείται κάθετα ως προς το άξονα z).



Εικόνα 2.3: Κλιμάκωση κύβου [UC Davis Ken Joy]

Διάτμηση

Επισκόπηση

Ο μετασχηματισμός διάτμησης στις τρεις διαστάσεις μεταβάλλει δύο από τις τρεις τιμές των συντεταγμένων αναλογικά ως προς την τιμή της τρίτης συντεταγμένης.

Μετασχηματισμός διάτμησης ως προς τον άξονα x .

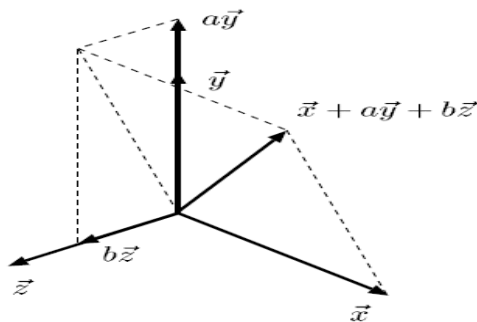
Δοθέντος ενός συστήματος συντεταγμένων εμείς εφαρμόζουμε τον μετασχηματισμό διάτμησης ως προς x στο πρώτο διάνυσμα του συστήματος συντεταγμένων προσθέτοντας σε αυτό ένα γραμμικό συνδυασμό των δύο άλλων διανυσμάτων. Οπότε ορίζουμε τον μετασχηματισμό αυτό με έναν 4×4 πίνακα ο οποίος έχει την εξής μορφή:

$$H_{x;a,b} = \begin{bmatrix} 1 & a & b & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Αν εφαρμόσουμε τον μετασχηματισμό (2.8) σε ένα σημείο (x, y, z) , παίρνουμε:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & a & b & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & ax + y & bx + z & 1 \end{bmatrix} \quad (2.9)$$

Μία αναπαράσταση αυτού απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 2.4: Διάτμηση ως προς τον άξονα x [UC Davis Ken Joy]

Μετασχηματισμός διάτμησης ως προς τον άξονα y

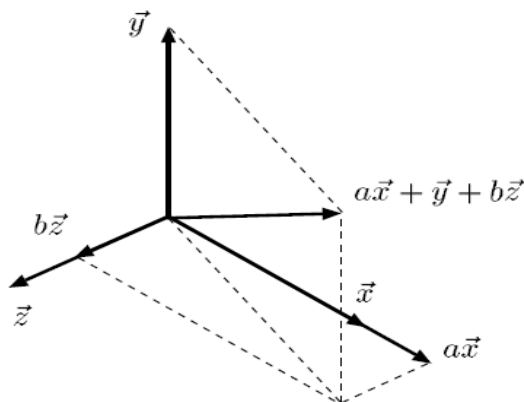
Δοθέντος ενός συστήματος συντεταγμένων εμείς εφαρμόζουμε τον μετασχηματισμό διάτμησης ως προς y στο δεύτερο διάνυσμα του συστήματος συντεταγμένων προσθέτοντας σε αυτό ένα γραμμικό συνδυασμό των δύο άλλων διανυσμάτων. Οπότε ορίζουμε τον μετασχηματισμό αυτό με έναν 4×4 πίνακα ο οποίος έχει την εξής μορφή:

$$H_{y;a,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Αν εφαρμόσουμε τον μετασχηματισμό (2.10) σε ένα σημείο (x, y, z) , παίρνουμε:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x + ay \quad y \quad z + by \quad 1] \quad (2.11)$$

Μία αναπαράσταση αυτού απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 2.5: Διάτμηση ως προς τον άξονα y [UC Davis Ken Joy]

Μετασχηματισμός διάτμησης ως προς τον άξονα z.

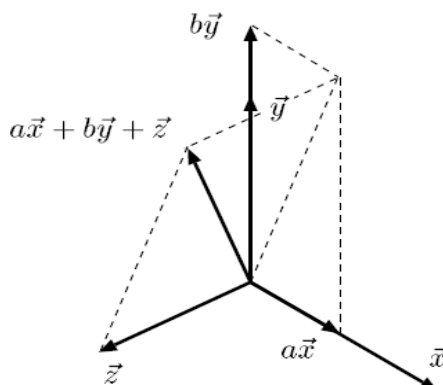
Δοθέντος ενός συστήματος συντεταγμένων εμείς εφαρμόζουμε τον μετασχηματισμό διάτμησης ως προς z στο τρίτο διάνυσμα του συστήματος συντεταγμένων προσθέτοντας σε αυτό ένα γραμμικό συνδυασμό των δύο άλλων διανυσμάτων. Οπότε ορίζουμε τον μετασχηματισμό αυτό με έναν 4 x 4 πίνακα ο οποίος έχει την εξής μορφή:

$$H_{z;a,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

Αν εφαρμόσουμε τον μετασχηματισμό (2.12) σε ένα σημείο (x, y, z), παίρνουμε:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x + az \quad y + bz \quad z \quad 1] \quad (2.13)$$

Μία αναπαράσταση αυτού απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 2.6: Διάτμηση ως προς τον άξονα z [UC Davis Ken Joy]

2.1.2 Συγγενής Μετασχηματισμοί

Τα θεμελιώδη τρισδιάστατα αντικείμενα που αποτελούν τη βάση για όλες τις λειτουργίες των γραφικών είναι το σημείο και το διάνυσμα (μερικές φορές ονομάζεται και ελεύθερο διάνυσμα). Αυτά τα αντικείμενα έχουν έναν εντελώς διαφορετικό ρόλο στις λειτουργίες του τρισδιάστατου χώρου. Και συγκεκριμένα:

- Ένα σημείο κατέχει μια θέση στο χώρο. Το μόνο χαρακτηριστικό που διακρίνει ένα σημείο από ένα άλλο είναι η θέση του.
- Ένας διάνυσμα έχει τόσο μέγεθος όσο και κατεύθυνση, αλλά δεν έχει σταθερή θέση στο χώρο.

Αυτά τα σημεία και τα διανύσματα παράγουν ένα συγγενή χώρο.

Ένας συγγενής μετασχηματισμός L από τον χώρο \mathbb{R}^n στον χώρο \mathbb{R}^n ορίζεται από έναν μη μοναδιαίο, γραμμικό πίνακα $A \in \mathbb{R}^{n \times n}$ και ένα διάνυσμα $b \in \mathbb{R}^n$ τέτοιο ώστε:

$$L: \mathbb{R}^n \rightarrow \mathbb{R}^n; x \mapsto Ax + b \quad (2.14)$$

Θα πρέπει να σημειωθεί ότι ένας τέτοιος μετασχηματισμός δεν είναι γραμμικός εκτός αν το $b = 0$. Για να τροποποιήσουμε κάπως αυτό και να επιτραπεί στους συγγενείς μετασχηματισμούς να εκτελούνται ως πολλαπλασιασμοί πινάκων ο L εγείρεται κατά μία διάσταση. Εκφράζοντας το $x \in \mathbb{R}^n$ ως $\begin{bmatrix} x \\ 1 \end{bmatrix} \in \mathbb{R}^{n+1}$, ο L γίνεται μία απεικόνιση από τον \mathbb{R}^{n+1} στον \mathbb{R}^{n+1} όποτε γράφουμε:

$$L: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}; \begin{bmatrix} x \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \quad (2.15)$$

Αυτή η αναπαράσταση του x ονομάζεται ομογενής αναπαράσταση και χρησιμοποιείται εκτεταμένα στα γραφικά υπολογιστών αφού επιτρέπει στα A, b να εκφράζονται με ένα μοναδικό πίνακα διατηρώντας παράλληλα τη γραμμικότητα των μετασχηματισμών.

Μετατόπιση

Επισκόπηση

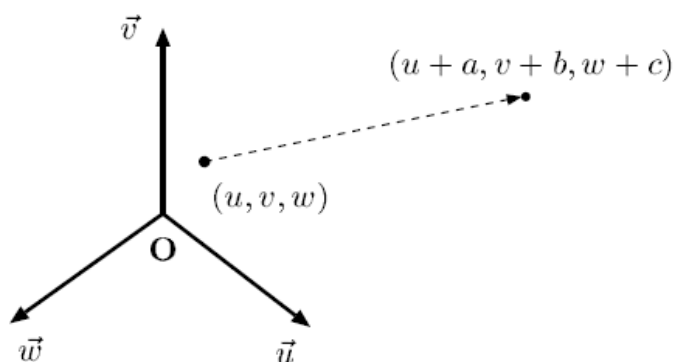
Η μετατόπιση, είναι ένας απλός μετασχηματισμός που μετατοπίζει τα σημεία ενός αντικειμένου κατά κάποια σταθερή απόσταση προς μια προκαθορισμένη κατεύθυνση.

Εφαρμογή του μετασχηματισμού στις τοπικές συντεταγμένες του αντικειμένου.

Δοθέντος ενός συστήματος συντεταγμένων και ενός σημείου \mathbf{P} το οποίο έχει τοπικές συντεταγμένες (u, v, w) , σε ομογενείς συντεταγμένες εάν εφαρμόσουμε τον μετασχηματισμό μετατόπισης στις τοπικές συντεταγμένες αυτού του σημείου, παίρνουμε

$$\begin{bmatrix} u & v & w & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & 1 \end{bmatrix} = \begin{bmatrix} u+a & v+b & w+c & 1 \end{bmatrix} \quad (2.16)$$

Μία αναπαράσταση αυτού απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 2.7: Μετατόπιση σημείου κατά a, b, c [UC Davis Ken Joy]

2.1.3 Αντίστροφοι Μετασχηματισμοί

Ένας μετασχηματισμός είναι αντιστρέψιμος εάν υπάρχει ένας αντίθετος μετασχηματισμός που μπορεί να επαναφέρει το μετασχηματισμένο αντικείμενο στην προηγούμενη του κατάσταση. Δεδομένου ότι οι μετασχηματισμοί εκφράζονται ως πίνακες, η εύρεση του αντιστρόφου μετασχηματισμού είναι ισοδύναμη με την εύρεση του αντίστροφου πίνακα του μετασχηματισμού. Όσον αφορά τους συγγενείς μετασχηματισμούς, η αντιστροφή του πίνακα που υλοποιεί την μετατόπιση γίνεται θέτοντας την άρνηση του πίνακα, δηλαδή η μετατόπιση του αντικειμένου γίνεται στην αντίθετη κατεύθυνση. Ωστόσο, δεν είναι όλοι οι γραμμικοί μετασχηματισμοί αντιστρέψιμοι, γι' αυτό πρέπει να ελέγχουμε αν ένας πίνακας είναι μοναδιαίος. Αυτό δεν συμβαίνει με τους Ευκλείδειους μετασχηματισμούς οι οποίοι είναι πάντα αναστρέψιμοι δεδομένου ότι για κάθε πίνακα R όπου R είναι ορθογώνιος ισχύει ότι: $R^{-1} = R^T$.

Γνωρίζοντας ότι οι μετασχηματισμοί σύνθεσης των $M = M_1 M_2 \dots M_n =$ είναι αντιστρέψιμοι, ο αντίστροφος μετασχηματισμός του M δίνεται από τη σχέση:

$$M^{-1} = M_1^{-1} M_2^{-1} \dots M_n^{-1}$$

2.2 Quaternions (Τετραδόνια)

2.2.1 Επισκόπηση

Τα quaternions ανακαλύφθηκαν από τον Hamilton [13], έναν φυσικό ο οποίος ερευνούσε για μία επέκταση του συστήματος των μιγαδικών αριθμών για να το χρησιμοποιήσει στην γεωμετρική οπτική. Τα quaternions από τότε έχουν εξελιχθεί και χρησιμοποιούνται τόσο στα γραφικά υπολογιστών όσο και στην ρομποτική επειδή μπορούν να ελέγχουν τις περιστροφές στον τρισδιάστατο χώρο. Παρακάτω ορίζουμε και αναλύουμε τις ιδιότητες των quaternions.

2.2.2 Τι είναι τα Quaternions?

Οι μιγαδικοί αριθμοί είναι μια επέκταση του συστήματος των πραγματικών αριθμών και μπορούν να γραφούν στην μορφή $a + bi$, όπου τα a και b είναι πραγματικοί αριθμοί και το $i^2 = -1$. Τα quaternions είναι απλά μία επέκταση των μιγαδικών αριθμών.

Ένα quaternion συνήθως γράφεται ως:

$$q = a + bi + cj + dk \quad (2.17)$$

όπου τα a , b , και c είναι πραγματικές τιμές, και τα i , j και k έχουν τις ιδιότητες ότι:

$$i^2 = -1, j^2 = -1, k^2 = -1$$

$$ij = k,$$

$$jk = i,$$

$$ki = j,$$

$$ji = -k,$$

$$kj = -i,$$

$$ik = -j$$

Αυτό είναι καθαρά μία επέκταση των μιγαδικών αριθμών όπου οι μιγαδικοί αριθμοί είναι αυτοί οι οποίοι έχουν τα $c = d = 0$ και οι πραγματικοί είναι αυτοί που έχουν τα $b = c = d = 0$.

Πρόσθεση Και Πολλαπλασιασμός Quaternions

Η πρόσθεση μεταξύ quaternions είναι μια απλή διαδικασία όπου απλά προσθέτουμε κατά συντελεστές. Αυτό σημαίνει ότι αν $q_1 = a_1 + b_1 i + c_1 j + d_1 k$ και $q_2 = a_2 + b_2 i + c_2 j + d_2 k$, τότε το άθροισμα δύο quaternions ισούται με:

$$q_1 + q_2 = (a_1 + a_2) + (b_1 + b_2)i + (c_1 + c_2)j + (d_1 + d_2)k \quad (2.18)$$

Ο πολλαπλασιασμός είναι κάπως πιο πολύπλοκος, αφού πρώτα πρέπει να πολλαπλασιάσουμε τους συντελεστές και μετά να χρησιμοποιήσουμε τους τύπους του γινομένου για τα i , j , και k και να απλοποιήσαμε την έκφραση. Οπότε το γινόμενο των q_1 και q_2 ισούται με:

$$\begin{aligned}
 q_1 q_2 &= (a_1 + b_1 i + c_1 j + d_1 k) (a_2 + b_2 i + c_2 j + d_2 k) \\
 &= a_1 a_2 + a_1 b_2 i + a_1 c_2 j + a_1 d_2 k + b_1 a_2 i + b_1 b_2 i i + b_1 c_2 i j + b_1 d_2 i k \\
 &\quad + c_1 a_2 j + c_1 b_2 j i + c_1 c_2 j j + c_1 d_2 j k + d_1 a_2 k + d_1 b_2 k i + d_1 c_2 k j + d_1 d_2 k k \\
 &= (a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) + (a_1 b_2 + a_2 b_1 + c_1 d_2 - d_1 c_2) i \\
 &\quad + (a_1 c_2 + a_2 c_1 + d_1 b_2 - b_1 d_2) j + (a_1 d_2 + a_2 d_1 + b_1 c_2 - c_1 b_2) k
 \end{aligned} \tag{2.19}$$

Εναλλακτική αναπαράσταση των quaternions.

Η αναπαράσταση για τον πολλαπλασιασμό των quaternions που δίνεται από την σχέση (2.19) είναι αρκετά πολύπλοκη και καταλήγει σε μεγαλύτερη πολυπλοκότητα όταν πρόκειται για διαίρεση και αντίστροφους τύπους. Τα quaternions μπορούν να γραφτούν με μια διαφορετική μορφή, μία μορφή η οποία εμπεριέχει διανύσματα τα οποία απλοποιούν δραστηρικά την αρχική αναπαράσταση των quaternions. Αυτή η έκφραση έχει γίνει και η προτιμότερη μορφή για την αναπαράσταση των quaternions.

Σε αυτή την μορφή το quaternion $q = a + bi + cj + dk$ γράφεται ως

$$(a, \vec{v}) \tag{2.20}$$

όπου \vec{v} είναι το διάνυσμα $\langle b, c, d \rangle$.

Μπορούμε να ξαναγράψουμε την έκφραση του αθροίσματος δύο quaternions $q_1 = (a_1, \vec{v}_1)$ και $q_2 = (a_2, \vec{v}_2)$ ως εξής:

$$q_1 + q_2 = (a_1 + a_2, \vec{v}_1 + \vec{v}_2) \tag{2.21}$$

και την αναπαράσταση του γινομένου ως

$$q_1 q_2 = (a_1 a_2 - \vec{v}_1 \cdot \vec{v}_2, a_1 \vec{v}_2 + a_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \tag{2.22}$$

Με κάποιες αλγεβρικές μετατροπές αυτοί οι τύποι μπορεί να αποδειχτεί ότι είναι ισοδύναμοι με αυτούς που αναπαριστούμε με τα i, j, k . Σημειώνουμε ότι τα quaternions της μορφής $(a, \langle 0, 0, 0 \rangle)$ μπορούν να συσχετιστούν με τους πραγματικούς αριθμούς, και τα quaternions της μορφής $(a, \langle b, 0, 0 \rangle)$ μπορούν να συσχετιστούν με μιγαδικούς αριθμούς.

2.2.3 Ιδιότητες Quaternions

Με αυτή τη νέα αναπαράσταση είναι εύκολο να αναπτυχθεί ένα πλήρες σύνολο ιδιοτήτων των quaternions. Δοθέντος των quaternions $q = (a, \vec{v})$, $q_1 = (a_1, \vec{v}_1)$, και $q_2 = (a_2, \vec{v}_2)$, μπορούμε να επαληθεύσουμε τις ακόλουθες ιδιότητες.

Πρόσθεση – Το άθροισμα των q_1 και q_2 ισούται με

$$q_1 + q_2 = (a_1 + a_2, \vec{v}_1 + \vec{v}_2) \tag{2.23}$$

Άρνηση – Το αντίστροφο $-q$ του q ισούται με

$$-q = (-a, -\vec{v}) \quad (2.24)$$

Αφαίρεση – Η διαφορά των q_1 και q_2 ισούται με

$$q_1 - q_2 = q_1 + (-q_2) = (a_1 - a_2, \vec{v}_1 - \vec{v}_2) \quad (2.25)$$

Πολλαπλασιασμός – Το γινόμενο των q_1 και q_2 ισούται με

$$q_1 q_2 = (a_1 a_2 - \vec{v}_1 \cdot \vec{v}_2, a_1 \vec{v}_2 + a_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \quad (2.26)$$

Ταυτόσημα – Η πολλαπλασιαστική ταυτότητα είναι $(1, \vec{0})$. Αυτό μπορεί να ελεγχθεί εύκολα

$$(a, \vec{v})(1, \vec{0}) = (a - \vec{v} \cdot \vec{0}, a\vec{0} + 1\vec{v} + \vec{v} \times \vec{0}) = (a, \vec{v}) \quad (2.27)$$

$$(1, \vec{0})(a, \vec{v}) = (a - \vec{0} \cdot \vec{v}, 1\vec{v} + a\vec{0} + \vec{0} \times \vec{v}) = (a, \vec{v}) \quad (2.28)$$

Αντίστροφος Πολλαπλασιασμός – Το αντίστροφο q^{-1} του $q = (a, \vec{v})$ δίνεται από τη σχέση

$$q^{-1} = \left(\frac{a}{a^2 + |\vec{v}|^2}, \frac{-\vec{v}}{a^2 + |\vec{v}|^2} \right) \quad (2.29)$$

Αυτό μπορεί να ελεγχθεί εύκολα αν υπολογίσουμε ότι

$$\begin{aligned} (a, \vec{v})(a, -\vec{v}) &= (a^2 + \vec{v} \cdot \vec{v}, -a\vec{v} + a\vec{v} + \vec{v} \times (-\vec{v})) \\ &= (a^2 + |\vec{v}|^2, \vec{0}) \end{aligned} \quad (2.30)$$

και έτσι $qq^{-1} = q^{-1}q = (1, \vec{0})$.

Διαίρεση – Το πηλίκο του q_1 με το q_2 είναι

$$\frac{q_1}{q_2} = q_1 q_2^{-1} \quad (2.31)$$

Σημείωση

Τα quaternions της μορφής $(a, \vec{0})$ συνήθως υποδηλώνονται στην πραγματική τους μορφή ως a , αυτό επιτρέπει τον πολλαπλασιασμό με μία σταθερά να δίνεται από τη σχέση:

Πολλαπλασιασμός με σταθερά – Αν το c είναι μια σταθερά τότε

$$\begin{aligned} cq &= (c, \vec{0})q \\ &= (c, \vec{0})(a, \vec{v}) \\ &= (ca - \vec{0} \cdot \vec{v}, c\vec{v} + a\vec{0} + \vec{0} \times \vec{v}) \\ &= (ca, c\vec{v}) \end{aligned} \quad (2.32)$$

Αυτό μας επιτρέπει να απλοποιήσουμε κάποιες εκφράσεις όπως την πολλαπλασιαστική ταυτότητα.

Το αντίστροφο q^{-1} του $q = (a, \vec{v})$ δίνεται από την παρακάτω σχέση

$$q^{-1} = \frac{(a, -\vec{v})}{a^2 + |\vec{v}|^2} \quad (2.33)$$

Αυτό μας επιτρέπει επίσης να γράψουμε την πολλαπλασιαστική ταυτότητα των quaternions ως 1 αντί για $(1, \vec{0})$, και την αθροιστική ταυτότητα ως 0. Τα quaternions δεν είναι αντιμεταθετικά ως προς τον πολλαπλασιασμό. Ενώ μπορούμε να προσθέσουμε, να αφαιρέσουμε, να πολλαπλασιάσουμε και να διαιρέσουμε τα quaternions, πρέπει πάντα να γνωρίζουμε τη σειρά με την οποία γίνονται αυτές οι πράξεις. Αυτό συμβαίνει γιατί τα quaternions δεν είναι αντιμεταθετικά ως προς τον πολλαπλασιασμό γενικά ισχύει ότι $q_1 q_2 \neq q_2 q_1$.

Ένα αντιπροσωπευτικό παράδειγμα γι' αυτό είναι αν θεωρήσουμε τα $q_1 = (1, \langle 1, 0, 0 \rangle)$ και $q_2 = (2, \langle 0, 1, 0 \rangle)$.

Πολλαπλασιάζοντας αυτά παίρνουμε

$$q_1 q_2 = (2 - 0, \langle 0, 1, 0 \rangle + 2 \langle 1, 0, 0 \rangle + \langle 0, 0, 1 \rangle) = (2, \langle 2, 1, 1 \rangle)$$

και

$$q_2 q_1 = (2 - 0, 2 \langle 1, 0, 0 \rangle + \langle 0, 1, 0 \rangle + \langle 0, 0, -1 \rangle) = (2, \langle 2, 1, -1 \rangle)$$

τα οποία δεν είναι ίδια. Αυτό συμβαίνει επειδή το εξωτερικό γινόμενο δίνει διαφορετικά αποτελέσματα σε σχέση με την σειρά των διανυσμάτων τα οποία πολλαπλασιάζονται γενικά ισχύει ότι, $\vec{v}_1 \times \vec{v}_2 \neq \vec{v}_2 \times \vec{v}_1$.

Μήκος ενός quaternion, Μοναδιαία quaternions

Ορίζουμε το μήκος ενός quaternion $q = (a, \vec{v})$ να ισούται με

$$|q| = \sqrt{a^2 + |\vec{v}|^2} \quad (2.34)$$

όπου $|\vec{v}|$ είναι το μέγεθος του διανύσματος \vec{v} . Μοναδιαία quaternions είναι αυτά που έχουν το μήκος ίσο με 1.

2.3 Στοιχεία Θεωρίας Επιπέδου

Επισκόπηση

Ένα επίπεδο στον τρισδιάστατο χώρο είναι ένας χώρος σημείων τα οποία είναι κάθετα σε ένα διάνυσμα \vec{n} (συνήθως αναφέρετε ως κανονικό διάνυσμα) και περνούν από ένα σημείο P. Αυτά αποτελούν θεμελιώδης γεωμετρικές δομές στα γραφικά υπολογιστών και την γεωμετρική αναπαράσταση αντικειμένων.

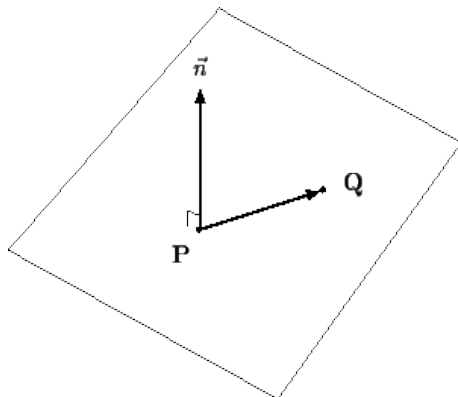
2.3.1 Καθορισμός ενός Σημείου και ενός Διανύσματος

Ένα επίπεδο στον τρισδιάστατο χώρο είναι ένας χώρος σημείων τα οποία είναι κάθετα σε ένα διάνυσμα \vec{n} και περνούν από ένα σημείο P. Το σημείο και το διάνυσμα καθορίζουν με μοναδικό

τρόπο ένα επίπεδο. Έστω \mathcal{P} το επίπεδο που ορίζεται από το διάνυσμα \vec{n} και το σημείο \mathbf{P} . Τότε για κάθε σημείο \mathbf{Q} στο επίπεδο, πρέπει να ισχύει ότι:

$$\vec{n} \cdot \langle \mathbf{Q} - \mathbf{P} \rangle = 0 \quad (2.35)$$

αφού το διάνυσμα $\mathbf{Q} - \mathbf{P}$ θα βρίσκεται στο επίπεδο. Αυτή η σχέση απεικονίζεται στο ακόλουθο σχήμα.



Εικόνα 2.8: Ορισμός επιπέδου [UC Davis Ken Joy]

2.3.2 Εξίσωση Επιπέδου και Πίνακας Τετραγωνικής Απόστασης

Υποθέτουμε ότι μας δίνεται ένα επίπεδο που ορίζεται από ένα σημείο \mathbf{P} και ένα διάνυσμα \vec{n} . Αν γράψουμε το διάνυσμα το \vec{n} ως $\vec{n} = \langle x_n, y_n, z_n \rangle$ και το σημείο \mathbf{P} ως $\mathbf{P} = (x_p, y_p, z_p)$, και ένα αυθαίρετο σημείο \mathbf{Q} ως $\mathbf{Q} = (x_q, y_q, z_q)$ πάνω στο επίπεδο τότε από τα παραπάνω έχουμε ότι:

$$\begin{aligned} 0 &= \vec{n} \cdot \langle \mathbf{Q} - \mathbf{P} \rangle \\ &= \langle x_n, y_n, z_n \rangle \cdot \langle x_q - x_p, y_q - y_p, z_q - z_p \rangle \\ &= x_n(x_q - x_p) + y_n(y_q - y_p) + z_n(z_q - z_p) \end{aligned}$$

απ' όπου μπορούμε να γράψουμε,

$$x_n x_q + y_n y_q + z_n z_q - (x_n x_p + y_n y_p + z_n z_p) = 0 \quad (2.36)$$

το οποίο είναι της μορφής

$$ax + by + cz + d = 0 \quad (2.37)$$

και αποτελεί σύνηθες μορφή της εξίσωσης ενός επιπέδου.

Στην παρούσα εργασία τα επίπεδα εξειδικεύονται στο πλαίσιο των τριγώνων, τα οποία αποτελούν δομικά στοιχεία για την κατασκευή τριγωνικών πολυεδρικών επιφανειών. Στις επόμενες ενότητες όλα τα επίπεδα θα αναφέρονται σε σχέση με κάποιο τρίγωνο του οποίου οι κορυφές είναι $V = [v_1, v_2, v_3]$, $v_i \in \mathbb{R}^3$. Στο πλαίσιο αυτό στην εξίσωση του επιπέδου η παράμετρος d είναι το τριπλό γινόμενο των συντεταγμένων των ακμών του τριγώνου.

$$d = [t_1, t_2, t_3] = ((t_1 \times t_2) \cdot t_3) \quad (2.38)$$

Η απόσταση ενός διανύσματος από ένα επίπεδο δίνεται από τη σχέση:

$$D = \frac{|ax+by+cz+d|}{\sqrt{a^2+b^2+c^2}} \quad (2.39)$$

Αν γράψουμε την σχέση $\langle Q - P \rangle$ ως $(Q - P) = v$ όπου v είναι διάνυσμα, η απόσταση οποιουδήποτε σημείου x από το επίπεδο του τριγώνου δίνεται από τη σχέση $(x-v)^T \cdot n$. Διαισθητικά ένα σημείο βρίσκεται εντός του επιπέδου του τριγώνου εάν η απόσταση του από αυτό είναι 0.

Συνδυάζοντας τις (2,36) και (2.38) μπορούμε να αποφανθούμε ότι ένα επίπεδο μπορεί να αναπαριστάτε από ένα διάνυσμα $p = [a \ b \ c \ d]^T$ οπότε:

$$(p^T \cdot x) = [a \ b \ c \ d] \cdot [a \ b \ c \ d]^T \quad (2.40)$$

είναι η απόσταση κάθε διανύσματος x από το επίπεδο p .

Η τετραγωνική απόσταση $Q_p(x)$ συνεπώς ορίζεται ως:

$$\begin{aligned} Q_p(x) &= (p^T x)^2 \\ &= (p^T x)(p^T x) \\ &= (x^T p)(p^T x) \\ &= x^T (pp^T) x \\ &= x^T K_p x \end{aligned} \quad (2.41)$$

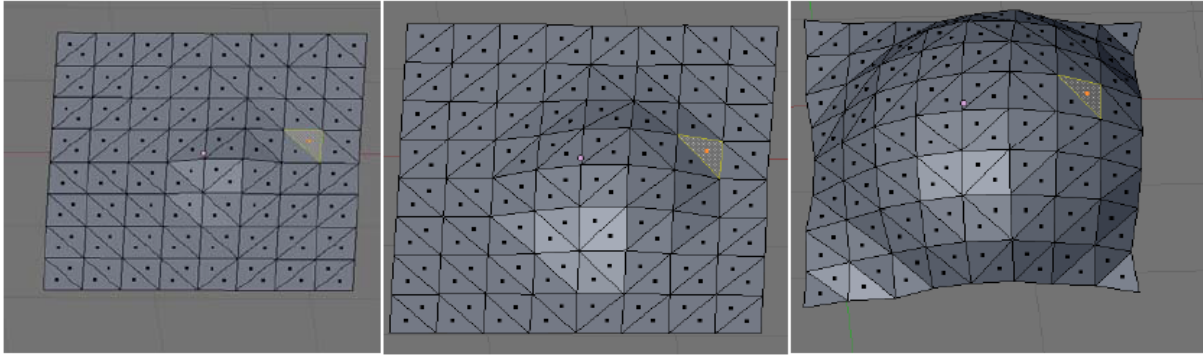
όπου K_p είναι ο συμμετρικός πίνακας:

$$K_p = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (2.42)$$

Ο πίνακας K_p λέγεται πίνακας τετραγωνικού σφάλματος και χρησιμεύει στον υπολογισμό της απόστασης κάθε σημείου από ένα επίπεδο $p = [a \ b \ c \ d]$. Ο πίνακας K_p χρησιμοποιείται στην εργασία μας για την απλοποίηση του μοντέλου.

2.3.3 Ποσότητες Παραμόρφωσης Επιπέδου.

Όταν η τριγωνοποίηση χρησιμοποιείται για την δημιουργία πολυεδρικών επιφανειών, το τρίγωνο γίνεται δομικό στοιχείο. Αν ένα τρισδιάστατο μοντέλο αλλάζει την κατάστασή του σε κάθε καρέ σχεδόν όλα τα τρίγωνα που το αποτελούν υπόκεινται σε μια σειρά μετασχηματισμών. Η εικόνα 2.9 δείχνει πώς ή όψη ενός τριγωνισμένου μοντέλου μπορεί να παραμορφωθεί μέσα σε μια ακολουθία κινούμενων εικόνων.



Εικόνα 2.9: Παραμόρφωση όψης σε μία ακολουθία κινήσεων ενός τριγωνισμένου μοντέλου.[16]

Υπάρχουν διάφορα μέσα για την ποσοτικοποίηση αυτής της παραμόρφωσης. Παρακάτω αναφέρουμε μια μέθοδο. Σε αυτή τη μέθοδο αναφερόμαστε σε μία ακολουθία κινήσεων ενός τριγωνισμένου μοντέλου η οποία αποτελείται από n πόζες P_0, \dots, P_n μία από αυτές τις ακολουθίες επιλέγεται ως σημείο αναφοράς έστω (P_0) .

Περιοχή παραμόρφωσης τριγώνου

Ένας τρόπος είναι να μετρήσουμε πώς η επιφάνεια του τριγώνου μεταβάλλεται μεταξύ της θέσης αναφοράς P_0 και της τρέχουσας θέσης P_i . Η επιφάνεια του τριγώνου $T = [t_1 \ t_2 \ t_3]$, $t_i \in \mathbb{R}^3$ στις τρεις διαστάσεις δίνεται από:

$$\text{Επιφάνεια} = \frac{1}{2} \sqrt{\left(\det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \right)^2 + \left(\det \begin{pmatrix} y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \\ 1 & 1 & 1 \end{pmatrix} \right)^2 + \left(\det \begin{pmatrix} z_1 & z_2 & z_3 \\ x_1 & x_2 & x_3 \\ 1 & 1 & 1 \end{pmatrix} \right)^2}$$

2.4 Σύστημα Βαρυκεντρικών Συντεταγμένων

Δοθέντος ενός συνόλου σημείων στον τρισδιάστατο χώρο, μπορούμε να ορίσουμε ένα τοπικό σύστημα συντεταγμένων σε σχέση με αυτά τα σημεία. Αυτό το σύστημα συντεταγμένων ονομάζεται βαρυκεντρικές συντεταγμένες και αναλύεται παρακάτω.

2.4.1 Τι είναι Βαρυκεντρικές Συντεταγμένες;

Ας θεωρήσουμε ένα σύνολο σημείων $\mathbf{P}_0 \ \mathbf{P}_1 \dots \mathbf{P}_n$ και ας θεωρήσουμε το σύνολο όλων των συγγενών μετασχηματισμών που παράγονται από αυτά τα σημεία. Αυτό σημαίνει ότι όλα τα σημεία μπορούν να γραφτούν ως

$$a_0 \mathbf{P}_0 + a_1 \mathbf{P}_1 + \dots + a_n \mathbf{P}_n$$

για κάποια $a_0 + a_1 + \dots + a_n = 1$

Αυτό το σύνολο σημείων αποτελεί ένα συγγενή χώρο και οι συντεταγμένες

$$(a_0, a_1, \dots, a_n)$$

καλούνται βαρυκεντρικές συντεταγμένες των σημείων του χώρου. Τέτοια συστήματα συντεταγμένων χρησιμοποιούνται αρκετά συχνά όπως για παράδειγμα όταν δουλεύουμε με τρίγωνα.

2.4.2 Βαρυκεντρικές Συντεταγμένες Τριγώνου.

Ας θεωρήσουμε τα σημεία $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$ στο επίπεδο. Αν τα a_1, a_2, a_3 είναι σταθερές τέτοιες ώστε $a_1 + a_2 + a_3 = 1$, τότε το σημείο \mathbf{P} το οποίο ορίζεται ως

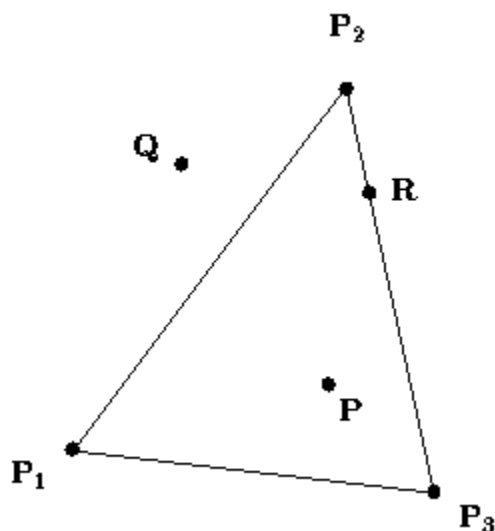
$$\mathbf{P} = a_1 \mathbf{P}_1 + a_2 \mathbf{P}_2 + a_3 \mathbf{P}_3 \quad (2.43)$$

είναι ένα σημείο πάνω στο επίπεδο του τριγώνου που ορίζεται από τα $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$. Το σημείο είναι εντός του τριγώνου $\mathbf{P}_1 \mathbf{P}_2 \mathbf{P}_3$ αν $0 \leq a_1, a_2, a_3 \leq 1$. Αν κάποιο από τα a είναι μικρότερο του μηδενός ή μεγαλύτερο του 1, το σημείο \mathbf{P} είναι εκτός του τριγώνου. Αν κάποιο από τα a είναι 0, τότε το σημείο \mathbf{P} βρίσκεται πάνω σε μία από τις ακμές του τριγώνου. Το παρακάτω σχήμα απεικονίζει ένα τέτοιο τρίγωνο καθώς και τα σημεία \mathbf{P} , \mathbf{Q} και \mathbf{R} , αυτά τα σημεία έχουν υπολογιστεί χρησιμοποιώντας τις παρακάτω τιμές για κάθε a :

$$\mathbf{P} \quad a_1 = a_2 = \frac{1}{4}, \quad a_3 = \frac{1}{2}$$

$$\mathbf{Q} \quad a_1 = \frac{1}{2}, \quad a_2 = \frac{1}{4}, \quad a_3 = \frac{1}{2}$$

$$\mathbf{R} \quad a_1 = 0, \quad a_2 = 1/4, \quad a_3 = \frac{1}{2}$$



Εικόνα 2.10 Βαρυκεντρικές συντεταγμένες τριγώνου [UC Davis Ken Joy]

Επιπλέον αν οι βαρυκεντρικές συντεταγμένες αθροίζουν στο 1 τότε αναπαριστούν ένα σημείο, αν οι συντεταγμένες αυτές αθροίζουν στο μηδέν τότε αναπαριστούν ένα διάνυσμα.

ΚΕΦΑΛΑΙΟ 3

ΣΧΗΜΑΤΙΚΗ ΠΑΡΑΜΟΡΦΩΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ (SSD)

3.1 Μοντέλα Με Σκελετική Ιεραρχία

3.2 Μέθοδος Προσδιορισμού Περιβλήματος

3.3 Υπολογισμός Βαρών Κορυφών

3.4 Linear Blend Skinning

3.5 Αποτελέσματα

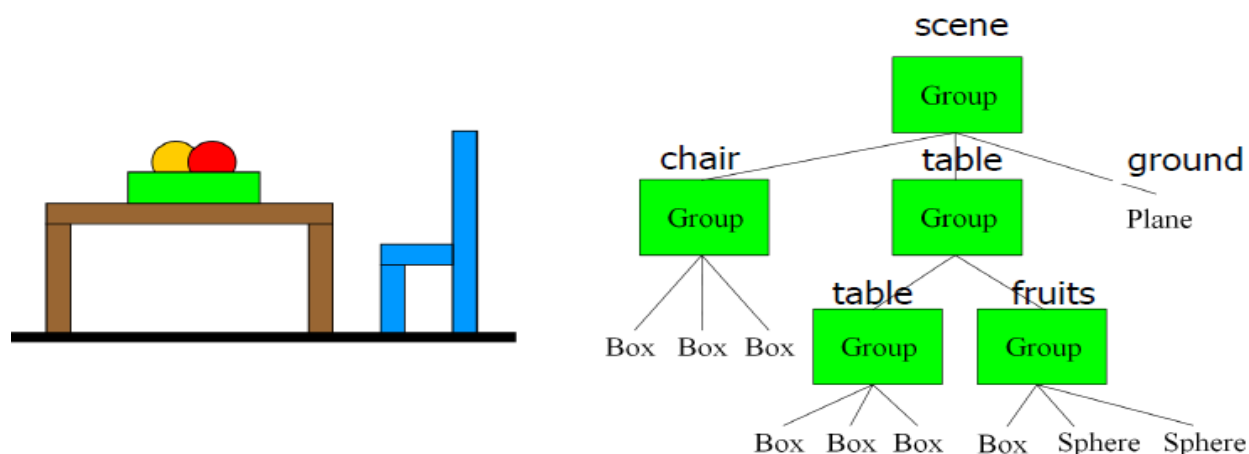
Στην παρακάτω ενότητα θα παρουσιάσουμε μεθόδους για τον προσδιορισμό περιβλήματος σε αρθρωτούς χαρακτήρες και πώς τα βάρη επιδρούν σε αυτό.

3.1 Μοντέλα Με Σκελετική Ιεραρχία

3.1.1 Ιεραρχική Μοντελοποίηση

Τα τρίγωνα, οι παραμετρικές καμπύλες και οι επιφάνειες είναι τα δομικά στοιχεία από τα οποία διαμορφώνονται τα περισσότερα σύνθετα αντικείμενα του πραγματικού κόσμου. Η ιεραρχική μοντελοποίηση δημιουργεί περίπλοκα αντικείμενα του πραγματικού κόσμου συνδυάζοντας απλά πρωτόγονα σχήματα σε πιο περίπλοκα συγκεντρωτικά αντικείμενα.

Στην παρακάτω εικόνα το γράφημα “scene” αποδίδει την λογική οργάνωση των αντικειμένων που βρίσκονται αριστερά στη σκηνή.



Εικόνα 3.1: Ιεραρχικό μοντέλο αναπαράστασης [Aalto University Jaakko Lehtinen]

Το ιεραρχικό αυτό μοντέλο ή αλλιώς ιεραρχικό δένδρο αποτελεί χρήσιμη δομή δεδομένων για την αναπαράσταση της σκηνής μας η οποία συνήθως αποτελείται από:

- Γεωμετρία (πολυεδρικές επιφάνειες)

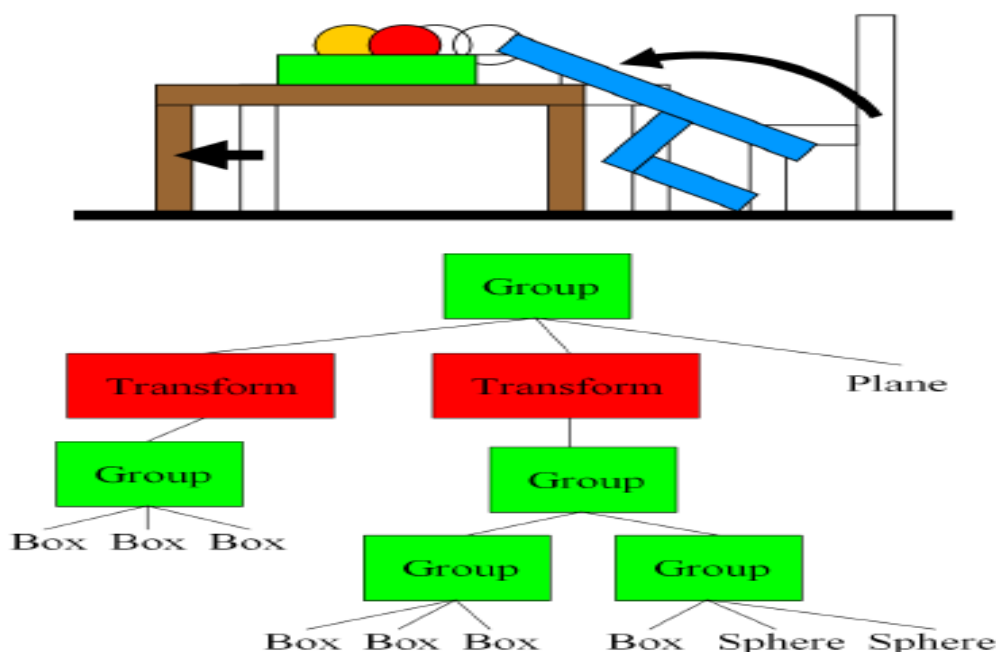
- Μετασχηματισμούς
- Υλικά, χρώματα
- Πολλαπλά στιγμιότυπα μοντέλων (animation)

Με χρήση αυτής της δομής ο χειρισμός των αρθρωτών κινούμενων χαρακτήρων (animated) γίνεται ιδιαίτερα εύκολος. Είναι χρήσιμο να σημειώσουμε ότι αντικείμενα τα οποία βρίσκονται κοντά το ένα με το άλλο στο ιεραρχικό αυτό δένδρο δεν είναι απαραίτητο να βρίσκονται και χωρικά κοντά το ένα με το άλλο.

3.1.2 Ιεραρχικός Μετασχηματισμός Αντικειμένων

Ένας μετασχηματισμός σε έναν κόμβο επηρεάζει ολόκληρο το υπό-δένδρο αυτού του κόμβου. Κάθε κόμβος έχει το δικό του τοπικό σύστημα συντεταγμένων. Οι μετασχηματισμοί ορίζονται πάντα σε σχέση με τον γονικό κόμβο. Η μετάβαση από το τοπικό σύστημα συντεταγμένων σε ένα καθολικό σύστημα συντεταγμένων και η απόδοση στην σκηνή μας είναι απόρροια συνένωσης όλων των μετασχηματισμών από τον τρέχων κόμβο προς τη ρίζα.

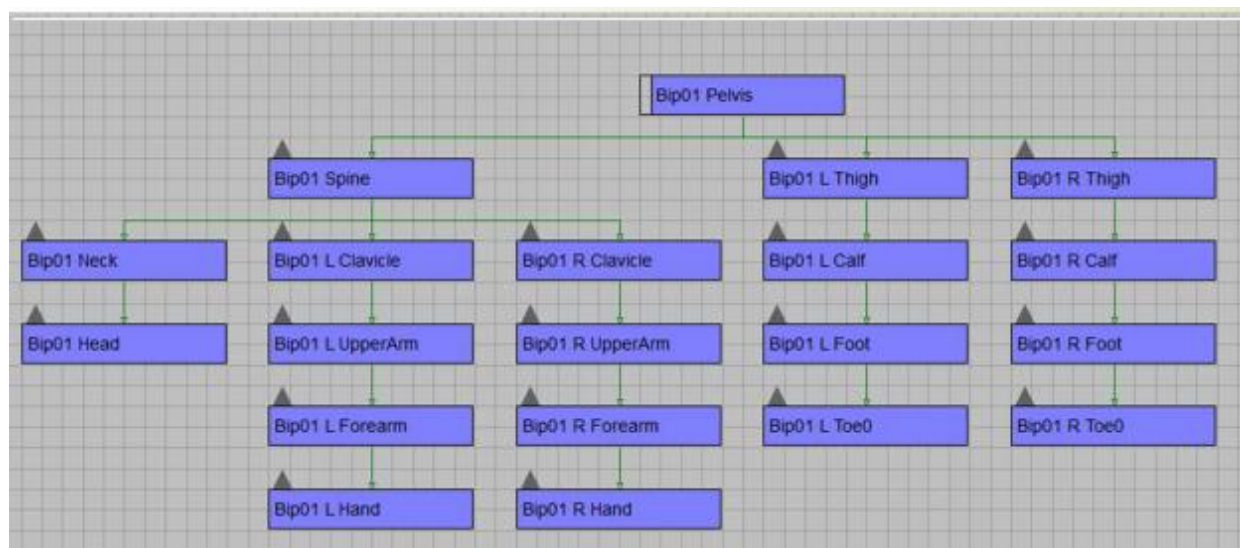
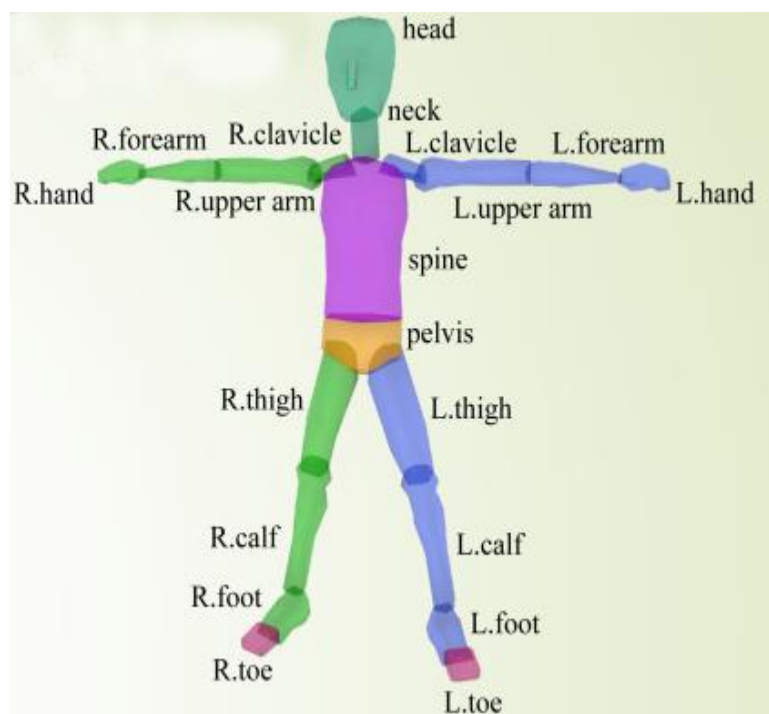
Ένα στιγμιότυπο ενός μετασχηματισμού της σκηνής απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 3.2: Ιεραρχικό μοντέλο αναπαράστασης μετασχηματισμού [Aalto University Jaakko Lehtinen]

Η διάσχιση που χρησιμοποιούμε για να επισκεφθούμε τους κόμβους του δένδρου είναι συνήθως η DFS (αναζήτηση κατά βάθος) ξεκινώντας από την κορυφή και πηγαίνοντας προς τα κάτω από αριστερά προς δεξιά η οποία μπορεί να υλοποιηθεί τόσο με αναδρομικούς αλλά και με ακολουθιακούς αλγορίθμους. Όταν συναντάμε έναν κόμβο που έχει κάποια γεωμετρία την αποδίδουμε στη σκηνή. Κατά την διάσχιση του δένδρου η κατάσταση ενημερώνεται τόσο για τους μετασχηματισμούς αλλά και για άλλα χαρακτηριστικά όπως το χρώμα το υλικό κ.α. Μία κατάλληλη δομή για αυτό είναι η στοίβα στην οποία εισάγουμε την τρέχουσα κατάσταση όταν επισκεπτόμαστε έναν κόμβο και διαγράφουμε κάποια κατάσταση όταν αλλάζουμε κόμβο.

Μοντελοποίηση ενός χαρακτήρα με χρήση σκελετικής ιεραρχίας.[wikipedia]



3.2 Μέθοδος προσδιορισμού περιβλήματος

Μια δημοφιλής μέθοδος για την υλοποίηση κίνησης ενός χαρακτήρα είναι με χρήση ενός σκελετού, ο οποίος αποτελείται από νοητά οστά προσαρμοσμένα στο σχήμα του μοντέλου μας[19]. Ο σκελετός είναι ενσωματωμένος σε ένα πολυγωνικό πλέγμα. Όταν ο σκελετός κινείται, οι κορυφές του πλέγματος του πολυγώνου κινούνται αντίστοιχα.

Η πιο δημοφιλής τεχνική: για τον προσδιορισμό περιβλήματος για υλοποίηση κίνησης ονομάζεται Σχηματική Παραμόρφωση Χώρου.(Skeletal Space Deformation)(SSD)

-Άλλες συγγενικές ονομασίες είναι:

- ανάμειξη κορυφών
- προσδιορισμός περιβλήματος με χρήση πίνακα

• γραμμική ανάμειξη περιβλήματος (LBS)
Linear blend skinning.

Η μέθοδος αυτή δεν έχει δημοσιευτεί αλλά αναφέρεται εκτεταμένα σε μελέτες η οποίες τείνουν να την βελτιώνουν ή να την επεκτείνουν.

Παρακάτω περιγράφουμε κάποιες γενικές αρχές για καλύτερη κατανόηση της μεθόδου.

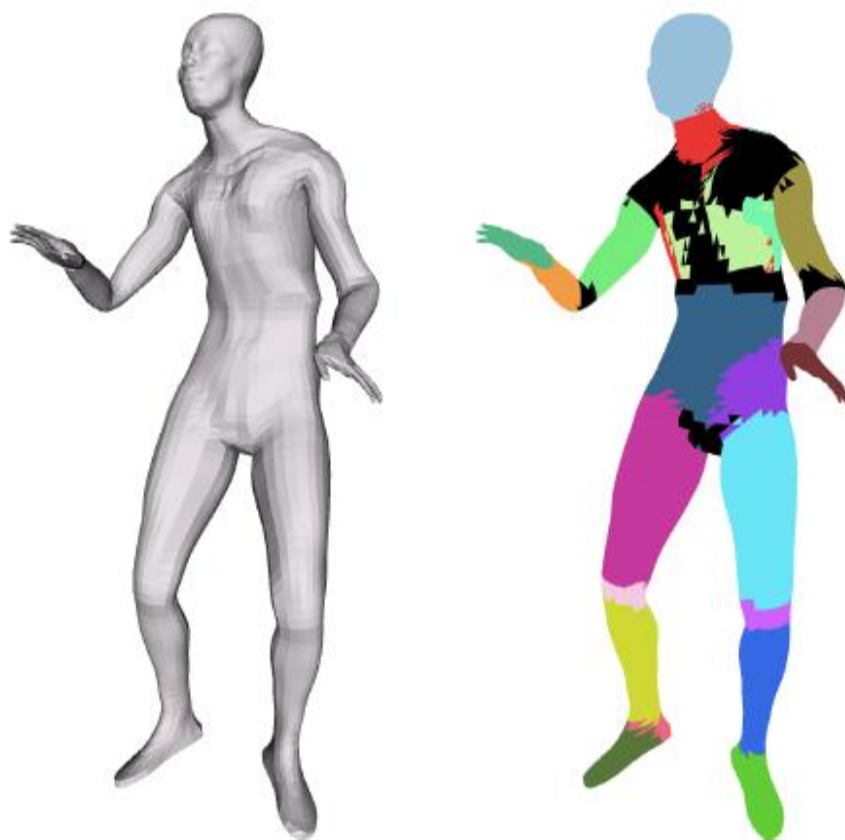
Κάθε οστό "παραμορφώνει το χώρο γύρω του "σε κάθε μετασχηματισμό όπως (περιστροφή, μετατόπιση κ.α.).

Τι γίνεται αν επισυνάψουμε κάθε κορυφή του περιβλήματος σε ένα μόνο οστό;

Το περίβλημα θα είναι άκαμπτο, και οι αρθρώσεις θα παραμορφώνονται κατά έναν αφύσικο τρόπο. Ας επισυνάψουμε τότε μια κορυφή σε περισσότερα από ένα οστά. Αν γίνει αυτό τότε στη μέση του οστού, τα σημεία του περιβλήματος ακολουθούν την περιστροφή του οστού (σχεδόν) άκαμπτα. Σε μια άρθρωση, το περίβλημα παραμορφώνεται σύμφωνα με έναν "σταθμισμένο συνδυασμό βαρών" των οστών.



Εικόνα 3.3: Αρθρωτό μοντέλο ενσωματωμένο σε εξωτερικό περίβλημα. [EpicGames.com]



Τα χρωματιστά τρίγωνα έχουν επισυναπτεί σε ένα μόνο άκρο.

Τα μαύρα τρίγωνα έχουν επισυναπτεί σε περισσότερα από ένα άκρα.

Εικόνα 3.4: Αρθρωτά μοντέλα με αναπαράσταση χρωματικής κωδικοποίησης. [James & Twigg 2005]

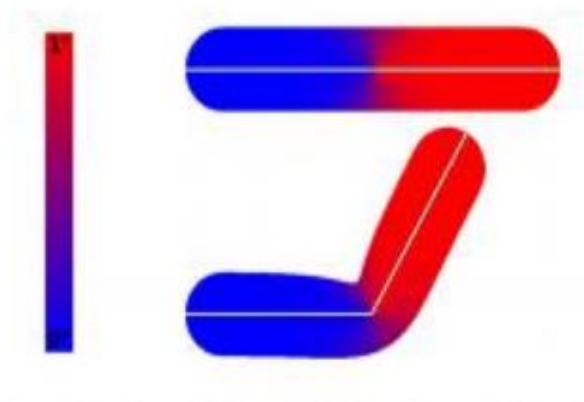
3.3 Υπολογισμός Βαρών Κορυφών

Ο προσδιορισμός των βαρών για την LBS αποτελεί χρονοβόρα και κουραστική διαδικασία. Η αλλαγή στη συμπεριφορά μιας κορυφής καθώς τα βάρη αλλάζουν είναι συχνά αντιφατική και μπορεί μην είναι σαφές αν υπάρχει μια τιμή που να δίνει την επιθυμητή θέση. Οι Mohr, Tokheim και Gleicher [21] κατασκεύασαν ένα εργαλείο το οποίο επιτρέπει τη αλλαγή των βαρών δια δραστικά αν και αυτή η προσέγγιση είναι αρκετά χρονοβόρα. Ωστόσο μία πρόσφατη προσέγγιση που υιοθετήθηκε από τους [22] [25] είναι να εκπαιδεύσουν το μοντέλο που πρόκειται να διαχειριστούμε θέτοντας τα βάρη έτσι ώστε να παρέχουν την μέγιστη δυνατή γεωμετρική προσαρμογή σε ένα σύνολο από εκπαιδευτικές πόζες αυτού του μοντέλου. Έτσι κατασκευάζουμε ένα σύστημα από εξισώσεις χρησιμοποιώντας τις θέσεις των κορυφών ως λύσεις για ένα πλήθος από εκπαιδευτικές πόζες. Αυτό το σύστημα μετά λύνεται για να προσδιορίσουμε τα βάρη. Καθώς αυτό το σύστημα ενδέχεται να έχει περισσότερες εξισώσεις απ' ότι αγνώστους λύνεται με την μορφή ελαχίστων τετραγώνων για να παρέχει όσο καλύτερη προσέγγιση.

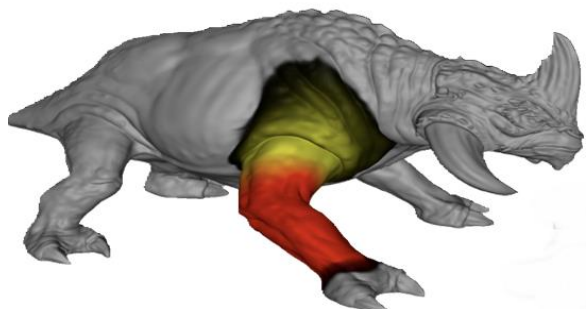
Παρακάτω παραθέτουμε τους συμβολισμούς και την ροή που ακολουθήσαμε σε αυτή την εργασία για τον προσδιορισμό των βαρών.

Θα αναθέσουμε ένα βάρος w_{ij} σε κάθε κορυφή \mathbf{p}_i για κάθε οστό \mathbf{B}_j .

Πόσο θα πρέπει να κινηθεί η κορυφή i με βάση το οστό j ; Για $w_{ij} = 1$ σημαίνει ότι η κορυφή p_i είναι άκαμπτα συνδεδεμένη με το B_j .



Εικόνα 3.5: Πάνω: θερμική ισορροπία για τα δύο άκρα. Κάτω: αποτέλεσμα περιστροφής δεξιού άκρου σύμφωνα την αντιστοίχιση θερμικής ισορροπίας.



Εικόνα 3.6: Απεικόνιση χρωματικής κωδικοποίησης άκρων πάνω σ' ένα ζώο. Όσο πιο σκούρα είναι η περιοχή τόσο μικρότερη είναι η επιρροή. [Wang and Phillips 2002]

Θα περιορίσουμε τον αριθμό των βαρών N τα οποία μπορούν να επηρεάσουν μία κορυφή σε $N = 8$ (οστά/κορυφή) κάτι το οποίο αποτελεί μία συνηθισμένη επιλογή επίσης επιλέγονται και 4 (οστά/κορυφή).

Γιατί όμως να κάνουμε κάτι τέτοιο;

Συνήθως δεν χρειαζόμαστε πάρα πολλά. Επίσης, ο αποθηκευτικός χώρος αποτελεί ένα ζήτημα.

Ιδιότητες Βαρών.

1. Συνήθως απαιτούμε τα βάρη να παίρνουν θετικές τιμές
2. Επιπλέον, θέλουμε το συνολικό άθροισμα όλων των βαρών για κάθε κορυφή να ισούται με 1, κάτι το οποίο σημαίνει ανεξαρτησία μετατόπισης.

3.4 Linear Blend Skinning

Στη συνέχεια θα αναλύσουμε τον αλγόριθμο προσδιορισμού περιβλήματος LBS και πώς πετυχαίνει αυτό που θέλουμε.

3.4.1 Υπολογισμός Θέσεων των Κορυφών και Εφαρμογή της LBS

Πώς γίνεται ο υπολογισμός των θέσεων των κορυφών.

Βήμα 1: Μετασχηματίζουμε κάθε κορυφή \mathbf{p}_i με βάση το οστό στο οποίο έχει αντιστοιχιστεί σαν να ήταν άκαμπτα συνδεδεμένη σε αυτό.

Βήμα 2: Συνδυάζουμε τα αποτελέσματα χρησιμοποιώντας τις τιμές των βαρών.

Οι παρακάτω σχέσεις δίνουν την νέα θέση των μετασχηματισμένων κορυφών σε συνδυασμό με τα βάρη.

$$\mathbf{p}'_{ij} = \mathbf{T}_j \mathbf{p}_i \quad (3.1)$$

$$\mathbf{p}'_i = \sum_j w_{ij} \mathbf{p}'_{ij} \quad (3.2)$$

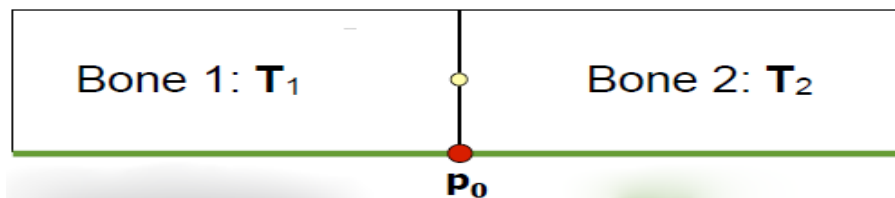
Το \mathbf{p}'_{ij} στη σχέση (3.1) είναι η κορυφή i μετασχηματισμένη με βάση το οστό j .

Το \mathbf{T}_j είναι ο τρέχων μετασχηματισμός του οστού j .

Το \mathbf{p}'_i στη σχέση (3.2) είναι η νέα θέση στο περίβλημα, της κορυφής \mathbf{p}_i .

Εφαρμογή της LBS σε μια κορυφή.

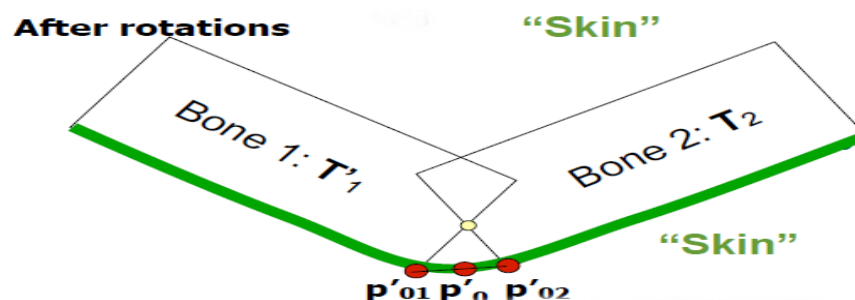
Στο παρακάτω σχήμα η κορυφή \mathbf{p}_0 έχει βάρη $w_{01} = 0.5$ και $w_{02} = 0.5$.



Εικόνα 3.7: Μοντελοποίηση δύο άκρων. [Aalto University Jaakko Lehtinen]

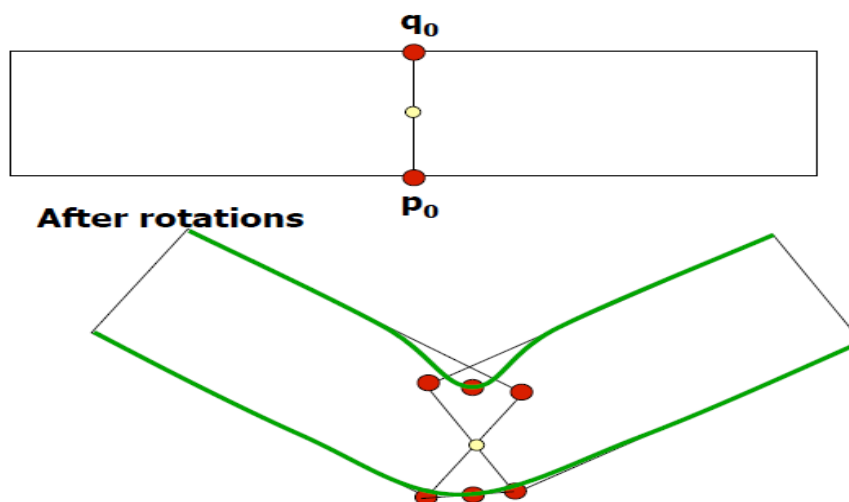
Εφαρμόζοντας τους μετασχηματισμούς με βάση τα \mathbf{T}'_1 και \mathbf{T}'_2 έχει ως αποτέλεσμα να πάρουμε τα \mathbf{p}'_{01} και \mathbf{p}'_{02}

Συνδυάζοντας τα \mathbf{p}'_{01} και \mathbf{p}'_{02} με τα αντίστοιχα βάρη η νέα θέση της \mathbf{p}_0 είναι $\mathbf{p}'_0 = 0.5 * \mathbf{p}'_{01} + 0.5 * \mathbf{p}'_{02}$ κάτι το οποίο βλέπουμε στο παρακάτω σχήμα.



Εικόνα 3.8: Απεικόνιση νέα θέσης κορυφής μετά τον μετασχηματισμό. [Aalto University Jaakko Lehtinen]

Η LBS παρόλα αυτά δεν είναι τέλεια. Η παρακάτω εικόνα δείχνει τις παραμορφώσεις που υπόκειται το περίβλημα μετά τον μετασχηματισμό.



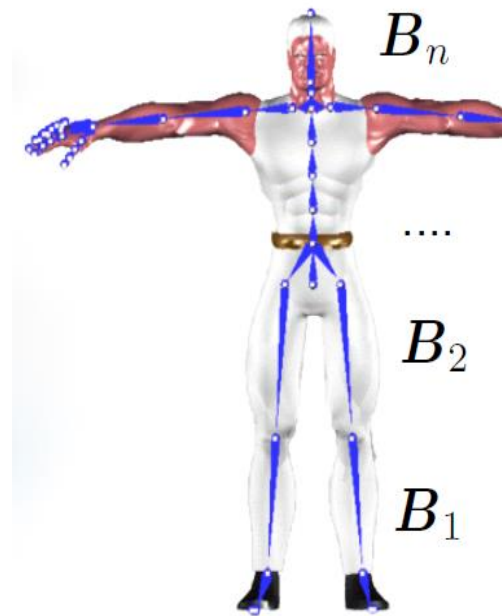
Εικόνα 3.9: Απεικόνιση παραμόρφωσης μετά τον μετασχηματισμό. [Aalto University Jaakko Lehtinen]

Περιγραφή και αναπαράσταση της Bind Pose.

Αρχικά μας δίνεται ένας σκελετός και ένα περίβλημα το οποίο βρίσκεται σε μία θέση η οποία λέγεται προεπιλεγμένη θέση (bind pose η T pose) επειδή τα χέρια του μοντέλου μας είναι σε σχήμα T. Σε αυτή τη θέση οι συντεταγμένες των κορυφών δίνονται σε σχέση με το χώρο του αντικειμένου του περιβλήματος. Προηγουμένως δεν αναφέραμε ότι για να έχει νόημα η σχέση $\mathbf{p}'_{ij} = \mathbf{T}_j \mathbf{p}_i$ τα συστήματα συντεταγμένων πρέπει να ταιριάζουν.

Στην φάση του ταιριάσματος του σκελετού με το περίβλημα, εμείς στρέφουμε τον σκελετό προς τα πάνω σύμφωνα το μη παραμορφωμένο περίβλημα.

Αυτό μας δίνει μια κατάσταση στάσης των μετασχηματισμών των οστών \mathbf{B}_j . Στη συνέχεια προσδιορίζουμε τα βάρη των κορυφών w_{ij} . Πώς γίνεται όμως αυτό; Συνήθως από καλλιτέχνες.



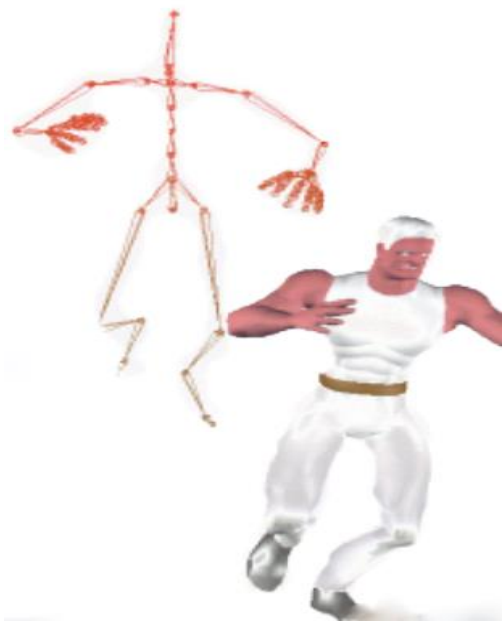
Εικόνα 3.10: Αναπαράσταση bind pose. [okino.com]

Όταν υλοποιούμε κίνηση του μοντέλου τα \mathbf{T}_j αλλάζουν. Ο μετασχηματισμός \mathbf{T}_j υλοποιεί την απεικόνιση από το τοπικό σύστημα συντεταγμένων του οστού j στο σύστημα συντεταγμένων του αντικειμένου που χειριζόμαστε.

Για να μπορέσουμε να παραμορφώσουμε το \mathbf{p}_i σύμφωνα με το \mathbf{T}_j , πρέπει πρώτα να εκφράσουμε το \mathbf{p}_i με βάση το τοπικό σύστημα συντεταγμένων του οστού j. Εδώ είναι που ο μετασχηματισμός \mathbf{B}_j της bind pose λαμβάνει χώρα.

$$\mathbf{p}'_{ij} = \mathbf{T}_j \mathbf{B}_j^{-1} \mathbf{p}_i \quad (3.3)$$

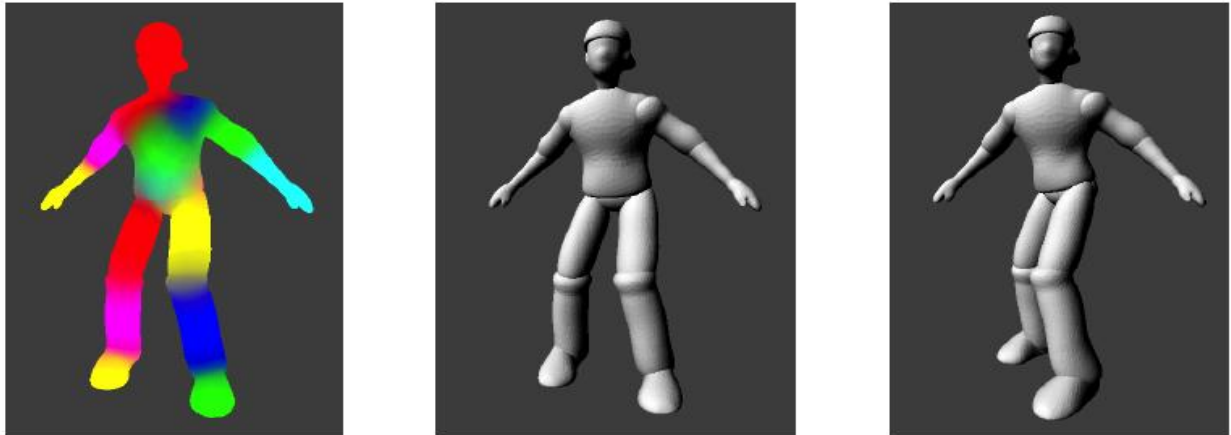
Η παραπάνω σχέση απεικονίζει το \mathbf{p}_i από την bind pose του τοπικού συστήματος συντεταγμένων του οστού j με χρήση του \mathbf{B}_j στον παραμορφωμένο χώρο του αντικειμένου με χρήση του \mathbf{T}_j .



Εικόνα 3.11: Αναπαράσταση μετασχηματισμένου μοντέλου.[12]

Τι είναι το $\mathbf{T}_j \mathbf{B}_j^{-1}$; Είναι η σχετική θέση των μετασχηματισμών των οστών μεταξύ της τρέχουσας θέσης και της bind θέσης.

Στιγμιότυπα εκτέλεσης της εφαρμογής μας πάνω σ' ένα ανθρωπόμορφο χαρακτήρα



Εικόνα 3.12: Αριστερά: Απεικόνιση βαρών των οστών πάνω στις κορυφές με χρωματική απεικόνιση, Μέση: απεικόνιση bind pose με κατευθυντικό φωτισμό. Δεξιά: παραμορφωμένο δέρμα, έχει γίνει περιστροφή της μέσης ως προς τον άξονα y κατά 0.7.

Ψευδοκώδικας περιβλήματος

Για κάθε κορυφή του περιβλήματος \mathbf{p}_i εφαρμόζουμε τη σχέση

$$\mathbf{p}'_i = \sum_j w_{ij} \mathbf{T}_j(t) \mathbf{B}_j^{-1} \mathbf{p}_i \quad (3.4)$$

Αντίστοιχα για τον υπολογισμό των κάθετων διανυσμάτων πάνω στην επιφάνεια του μοντέλου κάτι το οποίο χρησιμεύει στον υπολογισμό φωτισμού του μοντέλου μας χρησιμοποιούμε την παρακάτω σχέση

$$\mathbf{n}'_i = \left(\sum_j w_{ij} \mathbf{T}_j(t) \mathbf{B}_j^{-1} \right)^{-T} \mathbf{n}_i \quad (3.5)$$

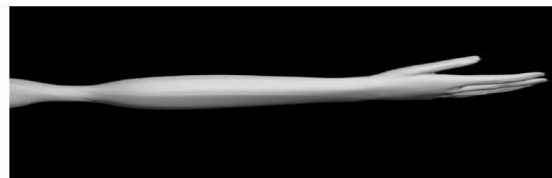
Ουσιαστικά η παραπάνω σχέση είναι το αντίστροφο του ανάστροφου πίνακα της σχέσης 3.4. Πρέπει να σημειώσουμε ότι τα βάρη είναι σταθερά κατά την διάρκεια του χρόνου εκτέλεση του αλγορίθμου. Έτσι μόνον ένας μικρός αριθμός πινάκων αλλάζει.

Δεδομένου ότι ελάχιστη πληροφορία ανανεώνεται σε κάθε χρονική στιγμή αυτό επιτρέπει την υλοποίηση στην GPU “vertex shaders”.

Στις παρακάτω εικόνες φαίνονται στιγμιότυπα κατά την εκτέλεση του LBS πάνω σε ένα χέρι



Εικόνα 3.13. Αγκώνας που παραμορφώνεται.[19]



Εικόνα 3.14. Ο βραχίονας στον καρπό καθώς υπολογίζεται από τον LBS. Καθώς ο καρπός πλησιάζει τις 180° ο βραχίονας παραμορφώνεται.[19]

3.5 Αποτελέσματα

Το πιο σημαντικό αποτέλεσμα της μεθόδου αυτής είναι ότι τα παραγόμενα μοντέλα παρουσιάζουν απώλεια όγκου όταν περιστρέφονται σε ακραίες γωνίες. Αυτό παρατηρείται στις παραμορφώσεις των αρθρώσεων και στο περίβλημα π.χ. Εικόνα 3.14. Αυτά τα ανεπιθύμητα αποτελέσματα προκύπτουν επειδή κατά την εύρεση της νέας θέσης της κορυφής σε μία συγκεκριμένη πόζα οι πίνακες μετασχηματισμού των αντίστοιχων οστών παρεμβάλλονται γραμμικά και η γραμμική παρεμβολή αυτή δεν είναι ισοδύναμη με την γραμμική παρεμβολή των περιστροφών τους. Μια προσέγγιση για να αφαιρέσουμε τη γραμμικότητα που είναι εγγενής στους συνδυασμούς των μετασχηματισμών των οστών που χρησιμοποιούνται από την LBS είναι αυτή των Mohr και Gleicher [21] οι οποίοι εισάγουν επιπλέον κόκαλα στις αρθρώσεις που περιστρέφονται κατά το ήμισυ μεταξύ των δύο συνδεδεμένων οστών ενώ οι Magnenat-Thalmann et al. [24] χρησιμοποιούν έναν συντελεστή ανάμειξης πινάκων που αναπτύχθηκε από τον Alexa [23] [24], και οι Kavan και Sloan [19] [20] χρησιμοποιούν τη γραμμική παρεμβολή των quaternions.

Επιπλέον η μέθοδος αυτή παρουσιάζει έλλειψη επεκτασιμότητας. Μία αύξηση του αριθμού των πολυγωνικών μοντέλων απαιτεί παραγωγή υψηλότερων ποιοτικών αποτελεσμάτων αλλά αυτή η μέθοδος αυξάνει το κόστος αποθήκευσης και υπολογισμού.

Παρόλα αυτά η μέθοδος αυτή παραμένει αρκετά δημοφιλής εξαιτίας της απλότητας της και της υπολογιστικής της απόδοσης.

Μία μέτρηση σχετικά με την υπολογιστική ταχύτητα του αλγορίθμου διεξάγεται από τους [25] οι οποίοι δημιουργούν έναν renderer ο οποίος υλοποιεί κίνηση μοντέλων στο υλικό και μετράει τον αριθμό τον καρέ όταν αποδίδουμε διαφορετικά μοντέλα στη σκηνή. Στη συνέχεια μετράει την ταχύτητα με την οποία μπορεί να βρεθεί η νέα θέση της κορυφής μιας δεδομένης σκελετική απεικόνισης.

ΚΕΦΑΛΑΙΟ 4

ΒΕΛΤΙΩΣΗ ΑΠΟΔΟΣΗΣ ΜΕΣΩ ΑΠΛΟΠΟΙΗΣΗΣ

4.1 Εισαγωγή

4.2 Απλοποίηση Επιφανειας

4.3 Απλοποίηση Μέσω Συνένωσης Κορυφών

4.4 Προσέγγιση Με Χρήση Τετραγωνικού Σφάλματος

Περίληψη

Πολλές εφαρμογές στα γραφικά υπολογιστών απαιτούν σύνθετα, πολύ λεπτομερή μοντέλα. Ωστόσο το πραγματικό επίπεδο λεπτομέρειας που είναι πραγματικά απαραίτητο σε εφαρμογές διαφορετικού σκοπού μπορεί να διαφέρει σημαντικά. Για καλύτερη διαχείριση του χρόνου επεξεργασίας της εφαρμογής μας, είναι συχνά επιθυμητό να χρησιμοποιούμε προσεγγίσεις μοντέλων αντί για υπερβολικά λεπτομερή μοντέλα.

Στην παρακάτω ενότητα θα αναλύσουμε τον αλγόριθμο απλούστευσης πολυγωνικών επιφανειών των Garland και Heckbert [14] [15] που χρησιμοποιήσαμε στην εργασία και πως αυτός υπολογίζει νέα βάρη τα οποία ενσωματώνουμε στο απλοποιημένο μας μοντέλο για την εφαρμογή της LBS και την επίτευξη κίνησης.

Ο αλγόριθμος αυτός παράγει γρήγορα, υψηλής ποιότητας προσεγγίσεις των αρχικών μοντέλων, χρησιμοποιώντας επαναληπτικές συνενώσεις ζευγών κορυφών για την απλοποίηση του μοντέλου και την διατήρηση του επιφανειακού σφάλματος χρησιμοποιώντας τετραγωνικούς πίνακες. Με την συνένωση αυθαίρετων ζευγών κορυφών ο αλγόριθμος μπορεί να ενώσει ασύνδετες περιοχές του μοντέλου πάνω στο οποίο εφαρμόζεται.

Αυτό μπορεί να πετύχει πολύ καλύτερες προσεγγίσεις τόσο οπτικά αλλά και σε σχέση με το γεωμετρικό σφάλμα. Για να επιτραπεί η τοπολογική σύνδεση μεταξύ περιοχών ο αλγόριθμος υποστηρίζει επιπλέον non-manifold¹ (μη πολλαπλότητας) γεωμετρικά μοντέλα επιφανειών.

4.1 Εισαγωγή

Συνήθως οι εφαρμογές των γραφικών υπολογιστών απαιτούν σύνθετα μοντέλα για να διατηρήσουν ένα πειστικό επίπεδο ρεαλισμού. Συνεπώς τα μοντέλα δημιουργούνται με πολύ υψηλή ανάλυση για να ικανοποιήσουν τις απαιτήσεις για λεπτομέρεια. Ωστόσο η πολυπλοκότητα των εν λόγω μοντέλων δεν απαιτείται πάντα και δεδομένου ότι το υπολογιστικό κόστος τέτοιων μοντέλων είναι ευθύς ανάλογο της πολυπλοκότητας τους, είναι χρήσιμο να έχουμε απλούστερες εκδόσεις σύνθετων μοντέλων. Φυσικά αυτό θα θέλαμε να γίνεται αυτόματα, κάτι το οποίο πραγματεύεται ο αλγόριθμος που θα αναλύσουμε σε επόμενες ενότητες αυτού του κεφαλαίου.

Θα επικεντρώσουμε το ενδιαφέρον μας στη απλούστευση μοντέλων πολυγωνικών επιφανειών. Θα θεωρήσουμε ότι τα μοντέλα δομούνται μόνον από τρίγωνα [9]. Αυτό δεν συνεπάγεται βλάβη της

γενικότητας δεδομένου ότι κάθε μοντέλο μπορεί να τριγωνιστεί ως μέρος μιας προ επεξεργασίας. Για να επιτευχθούν πιο αξιόπιστα αποτελέσματα, όταν οι κορυφές δύο όψεων τέμνονται σε ένα σημείο, θα θεωρήσουμε ότι οι όψεις μοιράζονται το ίδιο σημείο και όχι δύο ξεχωριστά σημεία τα οποία τα οποία τυχαίνει να συμπίπτουν στο χώρο. Ο αλγόριθμος που θα αναλύσουμε παρακάτω βασίζεται στην επαναληπτική συνένωση ζευγών κορυφών (το οποίο αποτελεί μια γενίκευση της αφαίρεσης ακμών). Καθώς ο αλγόριθμος προχωράει διατηρείται ένα γεωμετρικό σφάλμα του τρέχοντος μοντέλου. Αυτό το γεωμετρικό σφάλμα αποδίδεται με τη βοήθεια τετραγωνικών πινάκων. Τα κύρια πλεονεκτήματα του αλγορίθμου είναι:

- **Αποδοτικότητα:** Ο αλγόριθμος είναι σε θέση να απλοποιήσει πολύπλοκα μοντέλα αρκετά γρήγορα. Για παράδειγμα μπορεί να απλοποιήσει ένα μοντέλο με 13.000 όψεις σε 100 σε λιγότερο από 5 δευτερόλεπτα.
- **Ποιότητα:** Οι προσεγγίσεις που παράγονται από τον αλγόριθμο διατηρούν υψηλή πιστότητα στο αρχικό μοντέλο. Τα κύρια χαρακτηριστικά του μοντέλου διατηρούνται ακόμη και μετά από σημαντική απλούστευση.
- **Γενίκευση:** Σε αντίθεση με τους περισσότερους αλγόριθμους απλοποίησης ο αλγόριθμος αυτός είναι δυνατόν να συνενώσει ασύνδετες περιοχές του μοντέλου. Υπό την προϋπόθεση ότι η τοπολογία του αντικειμένου δεν αποτελεί σημαντικό παράγοντα αυτό μπορεί να διευκολύνει τις καλύτερες προσεγγίσεις αντικειμένων με πολλές ασύνδετες περιοχές. Αυτό απαιτεί από τον αλγόριθμο να υποστηρίζει non-manifold¹ μοντέλα.

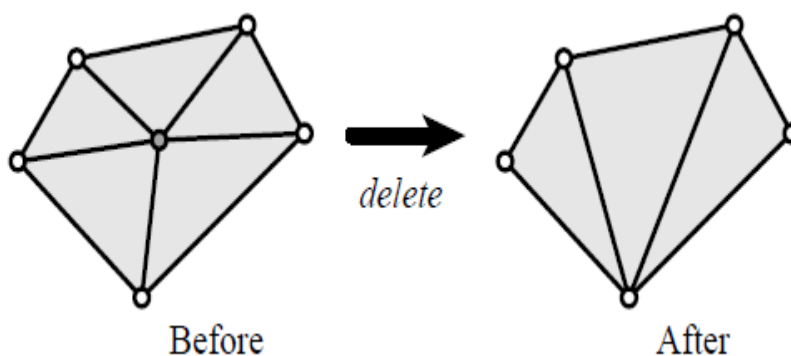
Στόχος του αλγορίθμου αυτού είναι να δέχεται ως είσοδο ένα τριγωνισμένο μοντέλο και να παράγει μία απλοποιημένη προσέγγιση του αρχικού μοντέλου. Το παραγόμενο μοντέλο πρέπει να πληροί κάποια κριτήρια όπως ένα επιθυμητό πλήθος όψεων και ένα ανεκτό σφάλμα. Δεν υποθέτουμε ότι η τοπολογία του μοντέλου πρέπει να διατηρηθεί. Ωστόσο σε ορισμένες επιστημονικές περιοχές όπως η ιατρική η διατήρηση της τοπολογίας του αντικειμένου μπορεί να είναι απαραίτητη. Ωστόσο στην περίπτωση της απόδοσης αντικειμένων στη σκηνή, η τοπολογία του αντικειμένου είναι λιγότερο σημαντική από την συνολική εμφάνιση του μοντέλου.

4.2 Απλοποίηση Επιφάνειας

Το πρόβλημα της απλοποίησης επιφανειών, έχει λάβει αυξημένη προσοχή τα τελευταία χρόνια. Αρκετοί αλγόριθμοι έχουν προταθεί για απλοποίηση επιφανειών. Τέτοιοι αλγόριθμοι μπορούν να κατηγοριοποιηθούν σε τρεις κλάσεις:

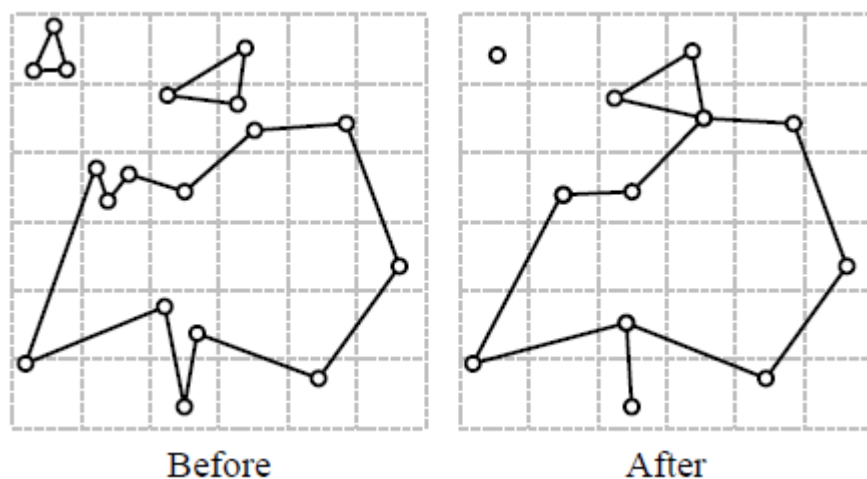
- **Απαλοιφή Κορυφών.** Η μέθοδος αυτή περιγράφει έναν αλγόριθμο τον οποίο θα ονομάσουμε απαλοιφή κορυφών. Αυτή η μέθοδος επαναληπτικά επιλέγει μία κορυφή για αφαίρεση, αφαιρεί όλες τις όψεις που προσπίπτουν σε αυτή και τριγωνοποιεί τις ενδεχόμενες οπές που μπορεί να δημιουργήθηκαν. Οι Soucy και Laurendeau [10] περιέγραψαν μία πιο εκλεπτυσμένη μορφή αλλά ουσιαστικά έναν παρόμοιο αλγόριθμο. Αυτοί οι αλγόριθμοι παρόλο που παρέχουν υψηλή απόδοση και ποιοτικά μοντέλα, διατηρούν προσεχτικά την τοπολογία του μοντέλου, και έτσι δεν είναι πραγματικά κατάλληλη για την απόδοση μοντέλων όπως στην περίπτωση μας που η τοπολογία δεν παίζει και τόσο σημαντικό ρόλο.

¹Στα μαθηματικά με τον όρο *manifold* (πολλαπλότητα) ορίζουμε τον τοπολογικό χώρο ο οποίος τοπικά μοιάζει με Ευκλείδειο χώρο [11] σε κάθε σημείο, ωστόσο αυτό σε παγκόσμιο επίπεδο μπορεί να μην ισχύει.



Εικόνα 4.1: Μία κορυφή έχει διαγραφεί και η προκύπτουσα οπή τριγωνοποιείται. [18]

- Ομαδοποίηση Κορυφών.** Ο αλγόριθμος ο οποίος περιγράφεται από τους Rossignac και Borrel [8] είναι ένας από τους λίγους ικανούς να επεξεργάζεται αυθαίρετα πολύγωνα. Ένας νοητός κύβος τοποθετείται γύρω από το αρχικό μοντέλο και χωρίζεται σε ένα πλέγμα. Μέσα σε κάθε κελί του πλέγματος οι κορυφές που βρίσκονται εκεί ομαδοποιούνται σε μία μοναδική κορυφή και οι όψεις του μοντέλου ανανεώνονται αντίστοιχα. Αυτή η διαδικασία εκτελείται πολύ γρήγορα και μπορεί να επιφέρει δραστικές αλλαγές στην τοπολογία του μοντέλου. Ωστόσο ενώ το μέγεθος των κελιών του πλέγματος παρέχουν ένα γεωμετρικό σφάλμα οι ποιότητα του παραγόμενου μοντέλου είναι αρκετά χαμηλή. Επιπλέον είναι δύσκολο να παράγουμε έναν προκαθορισμένο αριθμό όψεων δεδομένου ότι το πλήθος των όψεων προσδιορίζετε έμμεσα από το μέγεθος των κελιών του πλέγματος. Η ακριβής προσέγγιση που παράγεται εξαρτάται επίσης και από την θέση και τον προσανατολισμό του μοντέλου σε σχέση με το πλέγμα. Αυτή η ομοιόμορφη μέθοδος μπορεί να γενικευτεί με τη χρήση ενός προσαρμοστικού πλέγματος όπως π.χ. ένα octree οχτάριο [6].



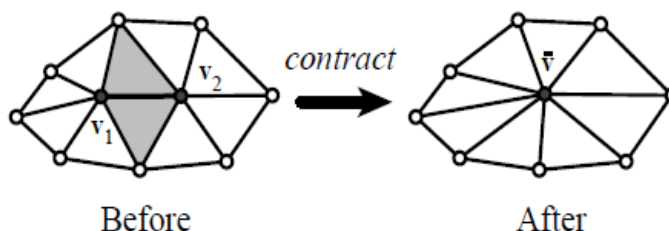
Εικόνα 4.2: Ομοιόμορφη ομαδοποίηση στις δύο διαστάσεις. [18]

- Επαναληπτική Σύμπτυξη Ακμών:** Πολύ αλγόριθμοι έχουν προταθεί οι οποίοι απλοποιούν τα μοντέλα με επαναληπτική σύμπτυξη ακμών (βλέπε Εικόνα 4.1). Η ουσιαστική διαφορά αυτών

των αλγορίθμων έγκειται στο πώς επιλέγουν τις ακμές για σύμπτυξη. Τέτοιους αλγόριθμους μπορούμε να αναφέρουμε όπως αυτούς των Hoppe [4,3] Ronfard και Rossignac [7], Gueziec [2]. Αυτοί οι αλγόριθμοι φαίνεται να έχουν σχεδιαστεί για να εφαρμόζονται πάνω σε manifold επιφάνειες αν και μπορούν να χρησιμοποιηθούν και σε non-manifold επιφάνειες. Με επαναληπτική εφαρμογή της σύμπτυξης ακμής [1] μπορούν να κλείσουν οπές αλλά δεν μπορούν να συνδέσουν ασύνδετες τοπολογικά επιφάνειες. Είναι κρίσιμο το προσεγγιστικό μοντέλο να βρίσκεται εντός μιας απόστασης από το πραγματικό μοντέλο και η τοπολογία του να παραμένει αναλλοίωτη.

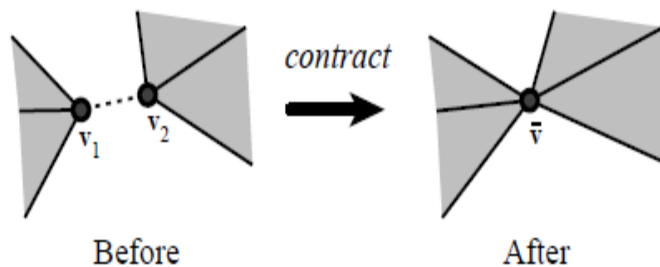
Ωστόσο κανένας από τους αλγορίθμους που αναφέραμε παραπάνω δεν συνδυάζει την αποδοτικότητα, την ποιότητα και την γενικευτική ιδιότητα που θα θέλαμε για την περίπτωση μας. Οι αλγόριθμοι απαλοιφής κορυφών είναι ακατάλληλη για την περίπτωσή μας διότι διατηρούν με πολύ προσοχή την αρχική τοπολογία του μοντέλου και υποθέτουν manifold γεωμετρικές επιφάνειες. Οι αλγόριθμοι ομαδοποίησης κορυφών αν και είναι πολύ γρήγοροι και έχουν καλή γενικευτική ιδιότητα, ωστόσο παρέχουν ελάχιστο έλεγχο όσο αφορά το τελικό αποτέλεσμα και παράγουν χαμηλής ποιότητας αποτελέσματα. Οι αλγόριθμοι απαλοιφής ακμών δεν υποστηρίζουν τη συνένωση ασύνδετων επιφανειών.

Ο αλγόριθμος των Garland και Heckbert που χρησιμοποιήσαμε υποστηρίζει τόσο την συνένωση τοπολογικά ασύνδετων επιφανειών αλλά και αποδίδει υψηλής ποιότητας αποτελέσματα. Ο αλγόριθμος αυτός κατέχει τόσο την γενικότητα της ομαδοποίησης κορυφών όσο και την ποιότητα και τον έλεγχο της επαναληπτικής απαλοιφής κορυφών και επιπλέον είναι πιο γρήγορος σε σχέση με άλλες μεθόδους όπως την [3].



Εικόνα 4.3: Απαλοιφή ή Συνένωση Ακμής. Η υπογραμμισμένη ακμή συμπτύσσεται σε ένα μοναδικό σημείο. [14]

Τα γειτονικά τρίγωνα η αλλιώς όψεις εκφυλίζονται και διαγράφονται κατά την συνένωση.



Εικόνα 4.4: Συνένωση δύο μη συνεκτικών περιοχών του μοντέλου. Οι διακεκομμένες γραμμές δείχνουν ότι οι δύο κόμβοι έχουν ενωθεί σε έναν. [14]

4.3 Απλοποίηση Μέσω Συνένωσης Κορυφών

Ο αλγόριθμος βασίζεται στην επαναληπτική συνένωση ζευγών κορυφών. Θα χρησιμοποιούμε εναλλακτικά τους όρους κορυφή και κόμβος. Όλα τα διανύσματα που χρησιμοποιούμε θεωρούμε ότι είναι διανύσματα στήλης και συμβολίζονται με πεζά σύμβολα. Οπότε, $u^T v = u \cdot v$ συμβολίζει το εσωτερικό γινόμενο δύο διανυσμάτων στήλης u και v . Οι πίνακες συμβολίζονται με κεφαλαία κι έντονους χαρακτήρες οπότε το $A = uv^T$ συμβολίζει το εξωτερικό γινόμενο $a_{ij} = u_i v_j$. Συμβολίζουμε την συνένωση, δύο κόμβων με τη σχέση $(v_1, v_2) \rightarrow \bar{v}$, η οποία μετακινεί τους κόμβους v_1, v_2 στην νέα θέση \bar{v} , συνδέει όλες τις προσπίπτουσες ακμές τους στον v_1 και διαγράφει τον κόμβο v_2 .

Στη συνέχεια όλες οι ακμές και όψεις που έχουν εκφυλιστεί αφαιρούνται. Αν τα (v_1, v_2) , είναι ακμή τότε μία ή περισσότερες όψεις θα αφαιρεθούν (βλέπε Εικόνα 4.3). Διαφορετικά δύο ξεχωριστά προηγούμενα τμήματα του μοντέλου θα ενωθούν σε ένα \bar{v} (βλέπε Εικόνα 4.4). Αυτή η έννοια της συνένωσης είναι στην πραγματικότητα αρκετά γενική διότι μπορούμε να συρρικνώσουμε ένα σύνολο κόμβων σε μια μοναδική κορυφή: $(v_1, v_2, \dots, v_k) \rightarrow \bar{v}$. Ξεκινώντας με το αρχικό μοντέλο M_n , μια ακολουθία συνενώσεων ζευγών κόμβων εφαρμόζεται έως ότου εκπληρωθούν οι στόχοι απλούστευσης και παραχθεί μια τελική προσέγγιση M_g . Επειδή κάθε συρρίκνωση αντιστοιχεί σε μια τοπική τροποποίηση του τρέχοντος μοντέλου, ο αλγόριθμος στην πραγματικότητα δημιουργεί μια ακολουθία μοντέλων M_n, M_{n-1}, \dots, M_g .

Έτσι, μια μοναδική εκτέλεση μπορεί να παράγει ένα μεγάλο αριθμό προσεγγιστικών μοντέλων. Το κύριο όφελος που κερδίζουμε όταν επιτρέπουμε τη συνένωση μη συνεκτικών περιοχών προηγουμένως είναι να υποστηρίξουμε non-manifold επιφάνειες. Τη στιγμή που δύο ξεχωριστές περιοχές ενώνονται είναι πολύ πιθανό να δημιουργηθεί μία non-manifold επιφάνεια. Σε αντίθετη περίπτωση θα απαιτούσε ιδιαίτερη αντιμετώπιση να διασφαλίσουμε ότι μια συνένωση δεν δημιουργήσει μία non-manifold περιοχή.

4.3.1 Επιλογή Ζεύγους.

Επιλέγουμε το έγκυρο σύνολο των ζευγών που θα συνενώσουμε κατά το χρόνο αρχικοποίησης του αλγορίθμου, και αυτό θα είναι το σύνολο που θα εξετάσουμε και κατά την διάρκεια εκτέλεσης του

αλγορίθμου. Η επιλογή αυτή βασίζεται στο γεγονός ότι μια σε μια καλή αρχική προσέγγιση οι κορυφές δεν κινούνται μακριά από την αρχική θέση τους.

Θεωρούμε ότι ένα ζεύγος (v_1, v_2) είναι έγκυρο για συνένωση αν:

1. (v_1, v_2) είναι ακμή ή
2. $\|v_1 - v_2\| < t$, όπου t είναι μία παράμετρος κατωφλίου.

Χρησιμοποιώντας για κατώφλι το $t = 0$ αυτό δίνει έναν απλό αλγόριθμο συνένωσης κορυφών. Υψηλότερες τιμές κατωφλίου επιτρέπουν την συνένωση μη συνδεδεμένων κορυφών. Φυσικά, το όριο αυτό πρέπει να επιλεγεί με κάποια προσοχή αν είναι πάρα πολύ μεγάλο θα μπορούσε να γίνει συνένωση διαφόρων περιοχών του μοντέλου που πιθανώς να είναι ανεπιθύμητο. Γι' αυτό και ακολουθούμε το σύνολο των αρχικών ζευγών κορυφών. Σε κάθε κορυφή συσχετίζουμε ένα σύνολο κορυφών στο οποίο αποτελεί μέλος. Όταν εκτελούμε τη διαδικασία $(v_1, v_2) \rightarrow \bar{v}$, το v_1 όχι μόνο αποκτά όλες τις ακμές που ήταν συνδεδεμένες στον v_2 , αλλά επίσης συγχωνεύει το σύνολο των ζευγών του v_2 στο δικό του σετ. Κάθε εμφάνιση του v_2 σε ένα έγκυρο ζεύγος αντικαθίσταται από το v_1 και αφαιρούνται τα διπλότυπα ζεύγη.

4.4 Προσέγγιση Με Χρήση Τετραγωνικού Σφάλματος.

Για να επιλέξουμε μια απαλοιφή να εκτελεστεί κατά τη διάρκεια μιας δεδομένης επανάληψης, χρειαζόμαστε κάποια ιδέα για το κόστος της απαλοιφής. Για να καθορίσουμε αυτό το κόστος, προσπαθούμε να χαρακτηρίσουμε το σφάλμα σε κάθε κορυφή. Για να το κάνουμε αυτό, συσχετίζουμε ένα συμμετρικό 4×4 πίνακα \mathbf{Q} σε κάθε κορυφή και καθορίζουμε το σφάλμα στην κορυφή

$$v = [v_x \ v_y \ v_z \ 1]^T \text{ να είναι τετραγωνικής μορφής } \Delta(v) = v^T \mathbf{Q} v.$$

Για την κατασκευή των τετραγωνικών πινάκων πρέπει να επιλέξουμε μία ευρετική συνάρτηση η οποία χαρακτηρίζει το σφάλμα μας. Η ευρετική που χρησιμοποιούμε είναι παρόμοια με αυτή των Ronfard και Rossignac [7]. Σύμφωνα την [7] παρατηρούμε ότι στο αρχικό μοντέλο, κάθε κορυφή είναι λύση της τομής ενός συνόλου επιπέδων, και πιο συγκεκριμένα των επιπέδων των τριγώνων που συναντώνται σε εκείνο το σημείο. Μπορούμε να συσχετίσουμε ένα σύνολο επιπέδων σε σχέση με κάθε κορυφή και να ορίσουμε το σφάλμα της κορυφής σε σχέση με αυτό το σύνολο ως το άθροισμα των τετραγωνικών αποστάσεων από τα επίπεδα.

$$\Delta(v) = ([v_x \ v_y \ v_z]^T) = \sum_{p \in \text{planes}(v)} (p^T v)^2 \quad (4.1)$$

Όπου $\mathbf{p} = [a \ b \ c \ d]^T$ αντιπροσωπεύει το επίπεδο που ορίζεται από την εξίσωση $ax + by + cz + d = 0$ όπου $a^2 + b^2 + c^2 = 1$.

Αυτή η προσέγγιση μέτρησης του σφάλματος είναι παρόμοια με την [7], αν και στην περίπτωσή μας έχουμε χρησιμοποιήσει άθροιση αντί για το μέγιστο ως προς το σύνολο των επιπέδων. Το σύνολο των επιπέδων σε μία κορυφή αρχικοποιείται έτσι ώστε να είναι τα επίπεδα των τριγώνων που συναντώνται σε αυτή την κορυφή. Σημειώνουμε ότι αν ήταν σκόπιμο να αποδώσουμε ρητά αυτά τα σύνολα επιπέδων, όπως κάνει η [7], θα μπορούσαμε να αποδώσουμε τα επίπεδα μετά τη συνένωση $(v_1, v_2) \rightarrow \bar{v}$ με χρήση του κανόνα: τα επίπεδα $(\bar{v}) = \text{τα επίπεδα } (v_1) \cup \text{τα επίπεδα } (v_2)$. Ωστόσο αυτό μπορεί να χρειαστεί ένα σημαντικό μέγεθος αποθηκευτικού χώρου που δεν μειώνεται καθώς η απλοποίηση εξελίσσεται.

Η μετρική σφάλματος που δίνεται στη (4.1) μπορεί να γραφεί σε τετραγωνική μορφή ως εξής:

$$\begin{aligned}
 \Delta(v) &= \sum_{p \in \text{planes}(v)} (v^T p) (p^T v) \\
 &= \sum_{p \in \text{planes}(v)} v^T (pp^T) v \\
 &= v^T \left(\sum_{p \in \text{planes}(v)} K_p \right) v
 \end{aligned} \tag{4.2}$$

όπου K_p είναι ο τετραγωνικός πίνακας:

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

Αυτός ο πίνακας χρησιμεύει για να βρούμε την τετραγωνική απόσταση οποιουδήποτε σημείου στο χώρο ως προς το επίπεδο \mathbf{p} . Μπορούμε να αθροίσουμε αυτούς τους πίνακες και να αναπαραστήσουμε ένα ολόκληρο σύνολο επιπέδων μέσω ενός μοναδικού πίνακα \mathbf{Q} .

Χειριζόμαστε το σύνολο των επιπέδων χρησιμοποιώντας έναν ενιαίο πίνακα, δηλαδή αντί να υπολογίσουμε μια συνένωση της μορφής (επίπεδα (v_1) \cup επίπεδα (v_2)) απλά προσθέτουμε δύο πίνακες \mathbf{Q} ($\mathbf{Q}_1 + \mathbf{Q}_2$). Αν τα σύνολα που αντιπροσωπεύονται από τα \mathbf{Q}_1 και \mathbf{Q}_2 στο αρχικό μοντέλο είναι ξένα, τότε η άθροιση ισοδυναμεί με το σύνολο της ένωσης. Αν υπάρχει κάποια επικάλυψη, τότε μπορεί να υπάρχει ένα μόνο επίπεδο το οποίο υπολογίζονται πολλές φορές. Ωστόσο, κάθε επίπεδο μπορεί να μετρηθεί το πολύ 3 φορές. Αυτό μπορεί να προκαλέσει κάποια ανακρίβεια στη μέτρηση σφαλμάτων, αλλά έχει σημαντικά οφέλη όσον αφορά τον χώρο που απαιτείται για την διατήρηση ενός επιπέδου και το κόστος ενημέρωσης.

Έτσι, για να υπολογίσουμε τους αρχικούς πίνακες \mathbf{Q} που απαιτούνται για τη συνένωση του ζεύγους μας, κάθε κορυφή πρέπει να διατηρεί τα επίπεδα των τριγώνων που συναντώνται σε αυτή την κορυφή. Για κάθε κορυφή, αυτό το σύνολο επιπέδων καθορίζει διάφορα θεμελιώδη τετραγωνικά σφάλματα K_p . Ο \mathbf{Q} για αυτή την κορυφή είναι το άθροισμα των θεμελιωδών τετραγωνικών σφαλμάτων. Σημειώνουμε ότι η αρχική εκτίμηση του σφάλματος για κάθε κορυφή είναι 0, δεδομένου ότι κάθε κορυφή βρίσκεται εντός των επιπέδων όλων των προσπίπτων τριγώνων.

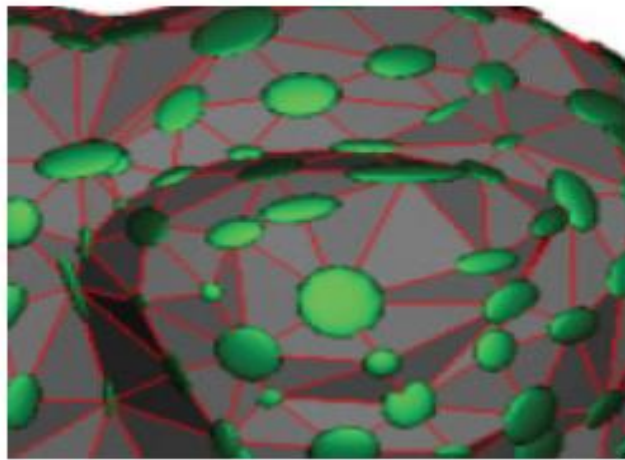
Προκειμένου να εκτελέσουμε την συνένωση (v_1, v_2) $\rightarrow \bar{v}$ πρέπει επίσης να επιλέξουμε μία θέση για το \bar{v} . Μία επιλογή θα ήταν να είναι είτε το v_1, v_2 ή το $(v_1 + v_2) / 2$ ανάλογα πιο από αυτά παράγει το μικρότερο $\Delta(\bar{v})$. Ωστόσο καλό θα ήταν να βρούμε μία θέση του \bar{v} η οποία ελαχιστοποιεί το σφάλμα. Δεδομένου ότι η συνάρτηση σφάλματος Δ είναι τετραγωνική, η εύρεση ελαχίστου είναι ένα γραμμικό πρόβλημα. Οπότε βρίσκουμε το \bar{v} λύνοντας το $\partial\Delta/\partial x = \partial\Delta/\partial y = \partial\Delta/\partial z = 0$. Αυτό είναι ισοδύναμο με την επίλυση του:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.3}$$

ως προς \bar{v} . Η τελευταία γραμμή του πίνακα είναι κενή επειδή το \bar{v} ομογενές διάνυσμα και ο συντελεστής w είναι πάντα 1. Υποθέτοντας ότι αυτός ο πίνακας είναι αντιστρέψιμος, παίρνουμε ότι:

$$\bar{v} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.4)$$

Αν αυτός ο πίνακας δεν είναι αντιστρέψιμος υπολογίζουμε την θέση του \bar{v} επιλέγοντας μεταξύ του μέσου η των άκρων.



Εικόνα 4.5: Οπτική αναπαράσταση τετραγωνικού σφάλματος. Όσο μικρότερα είναι τα ελλειψοειδές τόσο μεγαλύτερο είναι το σφάλμα. Όπως παρατηρούμε το σφάλμα είναι μικρότερο σε επίπεδες περιοχές και μεγαλύτερο σε γωνίες. [18]

4.4.1 Περίληψη Αλγορίθμου

Ο αλγόριθμος απλοποίησης βασίζεται στις συνενώσεις ζευγών κορυφών και το τετραγωνικό σφάλμα. Η τρέχουσα εφαρμογή χρησιμοποιεί μία δομή γράφου για να διατηρεί τις κορυφές, τις ακμές και τις όψεις κατά την εκτέλεση του αλγορίθμου. Για να ενημερώνουμε ένα σύνολο έγκυρων ζευγών, κάθε κορυφή διατηρεί μια λίστα με τα ζεύγη των οποίων είναι μέλος.

Ο αλγόριθμος μπορεί σύντομα να συνοψιστεί ως εξής:

1. Υπολόγισε τους πίνακες Q για όλες τις κορυφές.
2. Επέλεξε όλα τα έγκυρα ζεύγη.
3. Υπολόγισε τον βέλτιστο στόχο συνένωσης \bar{v} για κάθε έγκυρο ζεύγος (v_1, v_2) . Το σφάλμα $\bar{v}^T(Q_1, Q_2)$ \bar{v} αυτής της κορυφής αποτελεί το κόστος συνένωσης του ζεύγους.
4. Τοποθέτησε όλα τα ζεύγη σε ένα σωρό με βάση το κόστος με το ελάχιστο ζεύγος κόστους στην κορυφή.
5. Αφαίρεσε διαδοχικά το ζεύγος (v_1, v_2) του ελάχιστου κόστους από τον σωρό, συνένωσε αυτό

το ζεύγους και ενημέρωσε τα κόστη όλων των έγκυρων ζευγών που αφορούν το v_1 .

4.4.2 Χρονική Πολυπλοκότητα Αλγορίθμου.

Ο αλγόριθμος ξεκινάει με την φάση της αρχικοποίησης. Ο υπολογισμός όλων των quadrics παίρνει $O(n)$ χρόνο όπου $O(n)$ είναι το σύνολο των υποψήφιων ζευγών που έχουν επιλεγεί. Η επιλογή και αξιολόγηση όλων των υποψήφιων ζευγών απαιτεί επίσης $O(n)$ χρόνο.

Η τοποθέτηση όλων των υποψήφιων ζευγών στο σωρό απαιτεί $O(n \log n)$ χρόνο. Οπότε ο συνολικός χρόνος αρχικοποίησης είναι $O(n \log n)$. Τώρα, εξετάζουμε την επανάληψη της i -οστής φάσης απλοποίησης. Πρέπει να επιλέξουμε το ζευγάρι με το ελάχιστο κόστος, να το συνενώσουμε και να ενημερώσουμε τα γειτονικά ζεύγη. Οπότε το κόστος μίας απλής επανάληψης είναι $O(\log(n - ki))$ όπου k είναι το σύνολο των όψεων που αφαιρέθηκαν. Προσθέτοντας όλες τις επαναλήψεις το συνολικό κόστος ισούται με:

$$\log n + \log(n - k) + \log(n - 2k) + \dots + \log m$$

Δεδομένου ότι το k είναι σταθερά, μπορούμε να το αντικαταστήσουμε με το επάνω όριο

$$\log n + \log(n - 1) + \log(n - 2) + \dots + \log m$$

το οποίο ισούται με:

$$\log \frac{n!}{m!} = \log n! - \log m! = O(n \log n - m \log m)$$

Οπότε, η συνολική χρονική πολυπλοκότητα του αλγορίθμου είναι $O(n \log n)$

Εμπειρικά αυτή η πολυπλοκότητα δεν αυξάνεται όταν γίνεται επεξεργασία ιδιαίτερα μεγάλων μοντέλων.

Οι Kalvin και Taylor [5] περιγράφουν μια κάπως παρόμοια χρήση του τετραγωνικού σφάλματος για να αναπαραστήσουν τα σύνολα επιπέδων. Ακολουθούν σύνολα επιπέδων που ταιριάζουν σε ένα σύνολο σημείων εντός κάποιου σφάλματος.

ΚΕΦΑΛΑΙΟ 5

ΥΛΟΠΟΙΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

5.1 Λεπτομέρειες Υλοποίησης

5.2 Αποτελέσματα

5.3 Μελλοντικές Επεκτάσεις

5.1 Λεπτομέρειες Υλοποίησης

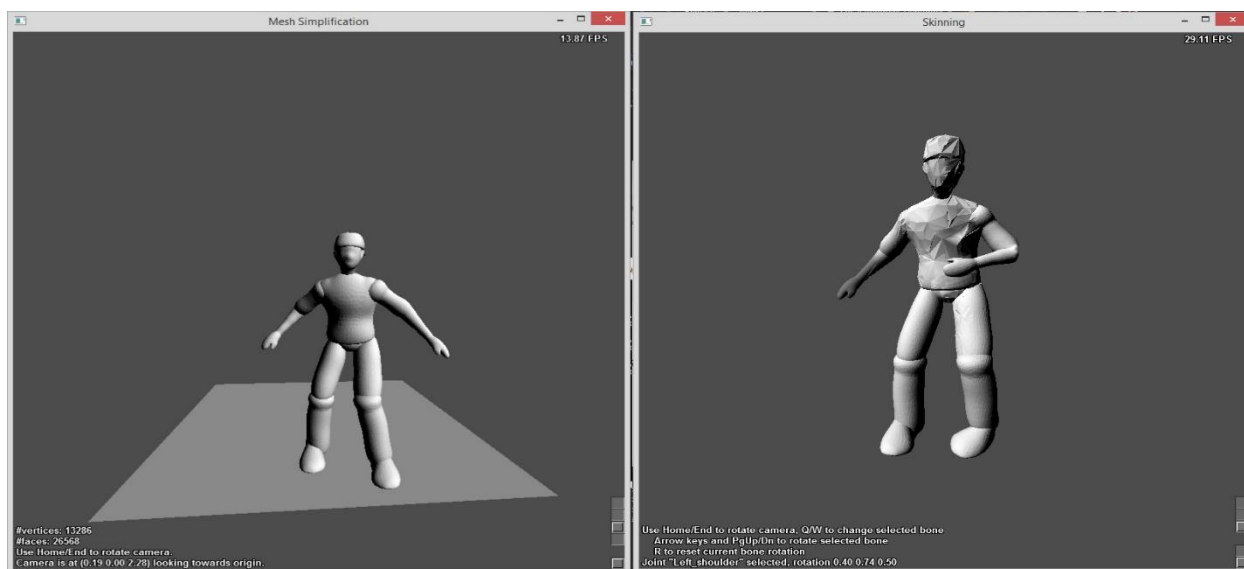
Η εφαρμογή μας υλοποιήθηκε με χρήση της γλώσσας προγραμματισμού C++. Έγινε ενσωμάτωση της βιβλιοθήκης Math από το περιβάλλον της Nvidia η οποία προσφέρει ένα πλήθος συναρτήσεων ιδανικές για πράξεις με διανύσματα και πίνακες. Όλη η εφαρμογή γράφτηκε με χρήση του προγράμματος Microsoft Visual Studio Professional 2015.

Η εφαρμογή έχει τη δυνατότητα να φορτώνει αρχεία σε μορφή Wavefront.obj, και να τα αποδίδει στη σκηνή μας. Ο χρήστης μπορεί να επεξεργαστεί τα μοντέλα που φορτώνει μέσω ενός γραφικού περιβάλλοντος, να δει ανά πάσα στιγμή χρήσιμες πληροφορίες για τα μοντέλα, να αλληλεπιδράσει με αυτά και να δει τα αποτελέσματα σε πραγματικό χρόνο.

Έγινε χρήση επίσης της GLSL γλώσσα σκίασης για την υλοποίηση της LBS. Ο χρήστης μπορεί να επιλέξει να εκτελέσει περιστροφές στα αντίστοιχα μοντέλα με τη χρήση κάποιων κουμπιών, επιλέγοντας κάθε φορά διαφορετικά άκρα όταν πρόκειται για ανθρώπινους χαρακτήρες. Επιπλέον μπορεί να εφαρμόσει την τεχνική απλοποίησης μοντέλου που περιγράψαμε σε προηγούμενες ενότητες και να πάρει αποτελέσματα σε πραγματικό χρόνο. Μπορεί επίσης να αποθηκεύσει αυτά τα μοντέλα και να τα φορτώσει ξανά βλέποντας έτσι το αποτέλεσμα σε νέα απλοποιημένα μοντέλα. Ένα στιγμιότυπο της εφαρμογής μας φαίνεται στην εικόνα 5.1 Όλα τα πειράματα εκτελέστηκαν σε λειτουργικό σύστημα, Windows8 64 bit, με χαρακτηριστικά επεξεργαστή AMD E1-6010 APU, κάρτα γραφικών AMD Radeon R2, 1350Mhz και 4GB Ram. Όλοι οι χαρακτήρες και αντικείμενα που χρησιμοποιήθηκαν δημιουργήθηκαν χρησιμοποιώντας το Blender.

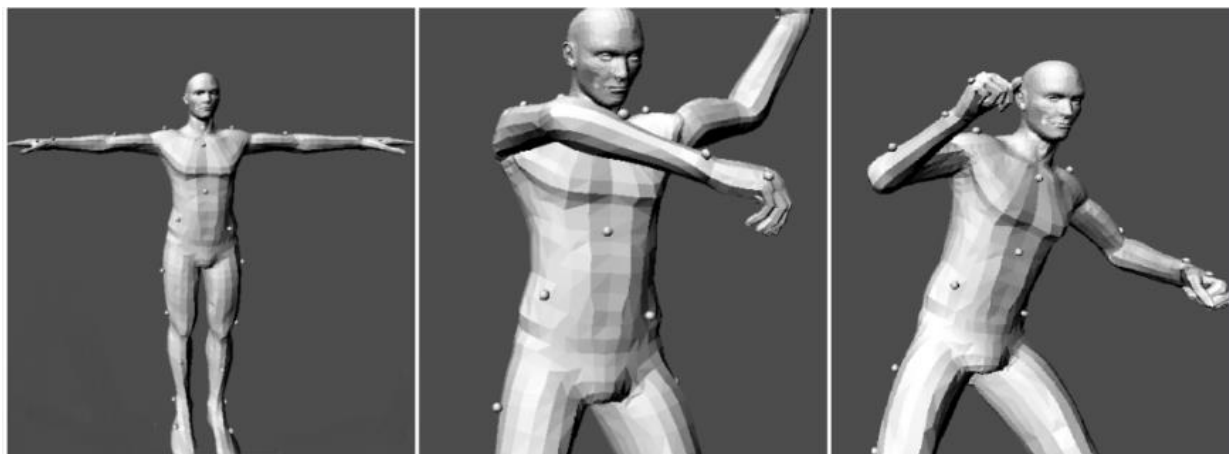
5.2 Αποτελέσματα

Παρακάτω παραθέτουμε στιγμιότυπα κατά την εκτέλεση της εφαρμογής μας πάνω σε διάφορα μοντέλα και τα αποτελέσματα τα οποία παίρνουμε.

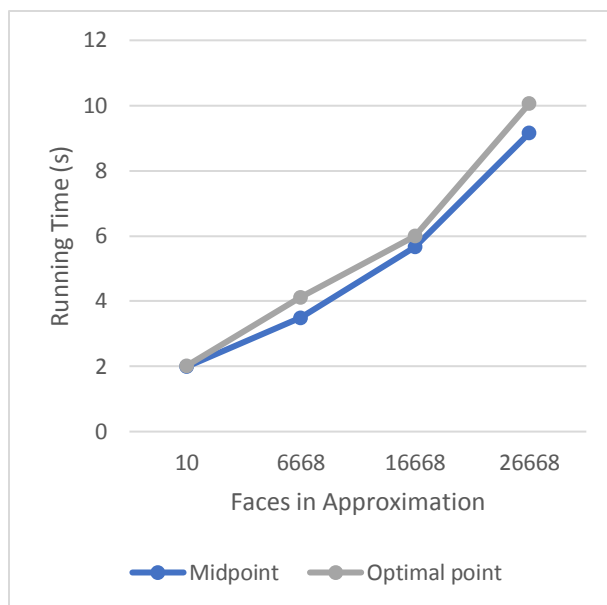


Εικόνα 5.1: Screenshot της γραφικής διεπαφής της εφαρμογής μας

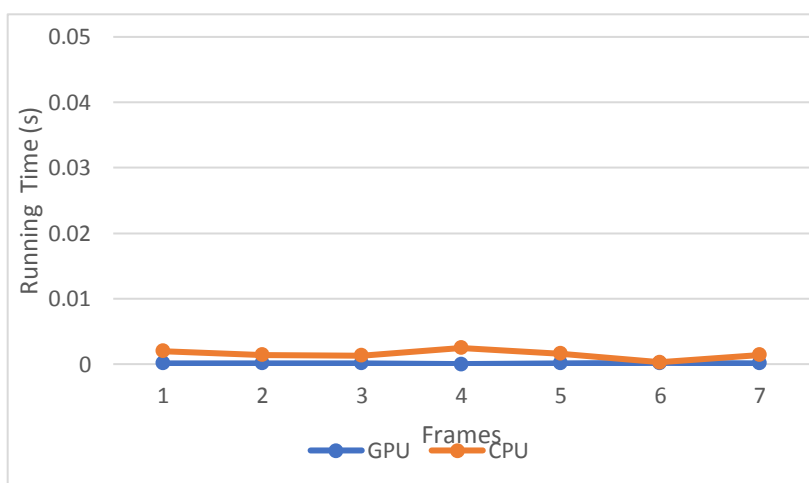
Στα δεξιά της εικόνας 5.1 φαίνεται ένας χαρακτήρας ο οποίος έχει φορτωθεί στην εφαρμογή μας και έχει αποδοθεί στην σκηνή ο χαρακτήρας αυτός βρίσκεται σε προκαθορισμένη στάση, και δεν έχει υποστεί καμιά επεξεργασία. Κάτω αριστερά μπορούμε να δούμε επίσης πληροφορίες σχετικά με τις κορυφές και τις όψεις του μοντέλου. Αριστερά της εικόνας 5.1 φαίνεται ο ίδιος χαρακτήρας ο οποίος έχει υποστεί απλοποίηση κατά 30% σε σχέση με το αρχικό μοντέλο και έχει υποστεί μία περιστροφή στον αριστερό αγκώνα. Παρατηρούμε ότι ακόμη και μετά από αυτή την απλοποίηση τα αποτελέσματα της LBS πάνω στο μοντέλο μας είναι αρκετά ικανοποιητικά.



Εικόνα 5.2 Ακολουθία κίνησης ανθρωπόμορφου χαρακτήρα.



Εικόνα 5.3 Ο χρόνος που απαιτείται για την απλοποίηση του μοντέλου που φαίνεται στην εικόνα 5.1. Ακολουθήθηκαν δύο διαφορετικές πολιτικές η πρώτη θέτει ως νέα θέση κορυφής το μέσο των δύο κορυφών που αφαιρέθηκαν ενώ η δεύτερη υπολογίζει τη νέα θέση με βάση το τετραγωνικό σφάλμα.



Εικόνα 5.4 Γράφημα το οποίο απεικονίζει το χρόνο που απαιτείται από τον LBS για να υπολογίζει τις θέσεις των κορυφών στην CPU και στην GPU αντίστοιχα σε κάθε frame. Γενικά παρατηρούμε ότι σε αυτή την περίπτωση ο υπολογισμός στην GPU είναι κατά 10% πιο γρήγορος απ' ότι στην CPU.

5.3 Μελλοντικές Επεκτάσεις

Στην παρούσα εργασία παρουσιάσαμε την μέθοδο προσδιορισμού περιβλήματος LBS την οποία εφαρμόσαμε πάνω σε ακολουθίες κινούμενων και μη σκελετικών χαρακτήρων. Εφαρμόσαμε την μέθοδο τετραγωνικού σφάλματος για την απλοποίηση πολυγωνικών επιφανειών και προσδιορίσαμε νέα βάρη έτσι ώστε να εφαρμόσουμε την LBS στα νέα απλοποιημένα μοντέλα. Για πολύ λεπτομερή μοντέλα η απλοποίηση και εφαρμογή της LBS στα απλοποιημένα μοντέλα έχει αρκετά ικανοποιητικά αποτελέσματα. Ωστόσο από ένα σημείο και μετά η LBS δεν δίνει καλά αποτελέσματα στα υπεραπλουστευμένα μοντέλα. Πιθανές επεκτάσεις μεθόδων αυτής της κατηγορίας είναι π.χ. εφαρμογή της LBS κάνοντας χρήση των quaternions για να εξαλείψουμε τις παραμορφώσεις στις αρθρώσεις, εφαρμογής μεθόδου η οποία εφαρμόζεται πάνω σε αρθρωτούς χαρακτήρες με βάση χωρικούς περιορισμούς. Για παράδειγμα, να μπορούμε να σύρουμε ένα σημείο του μοντέλου μας ας πούμε το χέρι και ο αγκώνας να επεκτείνεται με φυσικό τρόπο. Μία άλλη μέθοδο θα ήταν να εφαρμόσουμε αντίστροφη κινηματική, η οποία επιτρέπει την απόδοση ενός μοντέλου στην σκηνή μας χωρίς τον υποκείμενο σκελετό. Τέλος μία άλλη επέκταση θα ήταν να συνδυάσουμε την εφαρμογή μας με την εφαρμογή Kinect της Microsoft όπου η εφαρμογή αυτή είναι μία συσκευή η οποία αναγνωρίζει της στάση των ανθρώπων που στέκονται μπροστά από αυτή και τους αναθέτει ένα σκελετό στην στάση που βρίσκονται σε πραγματικό χρόνο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks, and William Wright. Simplification envelopes. In SIGGRAPH '96 Proc., pages 119–128, Aug. 1996. <http://www.cs.unc.edu/~geom/envelope.html>.
- [2] Andre' Gue'ziec. Surface simplification with variable tolerance. In Second Annual Intl. Symp. on Medical Robotics and Computer Assisted Surgery (MRCAS '95), pages 132–139, November 1995.
- [3] Hugues Hoppe. Progressive meshes. In SIGGRAPH '96 Proc., pages 99–108, Aug. 1996. <http://www.research.microsoft.com/research/graphics/hoppe/>.
- [4] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In SIGGRAPH '93 Proc., pages 19–26, Aug. 1993. <http://www.research.microsoft.com/research/graphics/hoppe/>.
- [5] Alan D. Kalvin and Russell H. Taylor. Superfaces: polygonal mesh simplification with bounded error. IEEE Computer Graphics and Appl., 16(3), May 1996. <http://www.computer.org/pubs/cg&a/articles/g30064.pdf>.
- [6] David Luebke and Carl Erikson. View-dependent simplification of arbitrary polygonal environments. In SIGGRAPH 97 Proc., August 1997.
- [7] Re'mi Ronfard and Jarek Rossignac. Full-range approximation of triangulated polyhedra. Computer Graphics Forum, 15(3), Aug. 1996. Proc. Eurographics '96.
- [8] Jarek Rossignac and Paul Borrel. Multi-resolution 3D approximations for rendering complex scenes. In B. Falcidieno and T. Kunii, editors, Modeling in Computer Graphics: Methods and Applications, pages 455–465, 1993.
- [9] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. Computer Graphics (SIGGRAPH '92 Proc.), 26(2):65–70, July 1992.
- [10] Marc Soucy and Denis Laurendeau. Multiresolution surface modeling based on hierarchical triangulation. Computer Vision and Image Understanding, 63(1):1–14, 1996.
- [11] Deroose. T. Coordinate-free geometric programming. Technical Report 89-09-16, Department of Computer Science and Engineering, University of Wisconsin Seattle, Washington 1994.
- [12] Okino.com
- [13] W. R. Hamilton. "On quaternions, or on a new system of imaginaries in algebra", Philosophical Magazine. Vol. 25, n 3. p. 489-495. 1844.
- [14] M. Garland and P. S. Heckbert. Surface simplification using quadratic error metrics. Comput. Graphics, Proc. SIGGRAPH 1997, 209-216.
- [15] M. Garland and P. S. Heckbert. Simplifying surfaces with color and texture using quadratic error metrics. In "Proc. Visualization, 1998", IEEE Computer Society Press, Los Alamitos, California, 279-286.
- [16] Fast Realistic Skinning For Animating Deformable Objects Γεώργιος Αντωνόπουλος November 2010
- [17] 11. K.G.Der, R.W. Sumner, and J. Popovic. Inverse kinematics for 'reduced deformable models. ACM Trans. Graph., 25(3):1174– 1179, July 2006
- [18] Quadric-Based Polygonal Surface Simplification Michael Garland May 9, 1999 CMU-CS-99-105
- [19] 21. L. Kavan, P.-P. Sloan, and C. O'Sullivan. Fast and efficient skinning of animated meshes. Computer Graphics Forum, 29(2):327– 336, 2010
- [20] L. Kavan, P. Sloan, C. O'Sullivan. "Fast and Efficient Skinning of Animated Meshes". Eurographics 2010

- [21] P. G. Kry, D. L. James, D. K. Pai. “EigenSkin: Real Time Large Deformation Character Skinning in Hardware”. In the Proceedings of the 2002 ACM SIGGRAPH /Eurographics Symposium on Computer Animation, ACM Press, p.153-159
 - a. Mohr and M. Gleicher. Building efficient, accurate character skins from examples. ACM Trans. Graphics, 22(3):562–568, 2003.
- [22] P. G. Kry, D. L. James, and D. K. Pai. Eigenskin: real time large deformation character skinning in hardware. In Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 153–159. ACM Press, 2002.
- [23] M. Alexa. Linear combination of transformations. In SIGGRAPH ’02: Proceedings of the 29th annual manipulation of interactive character skins. In Proceedings of the 2003 symposium on Interactive 3D graphics, pages 27–30. ACM Press, 2003.
- [24] N. Magnenat-Thalmann, F. Cordier, H. Seo, and G. Papagianakis. Modeling of bodies and clothes for virtual environments. In Third International Conference on Cyberworlds (CW’04), pages 201–208, 2004.
- [25] B. Merry, P. Marais, and J. Gain. Animation space: A truly linear framework for character animation. ACM Trans. Graph., 25(4):1400–1423, 2006.