

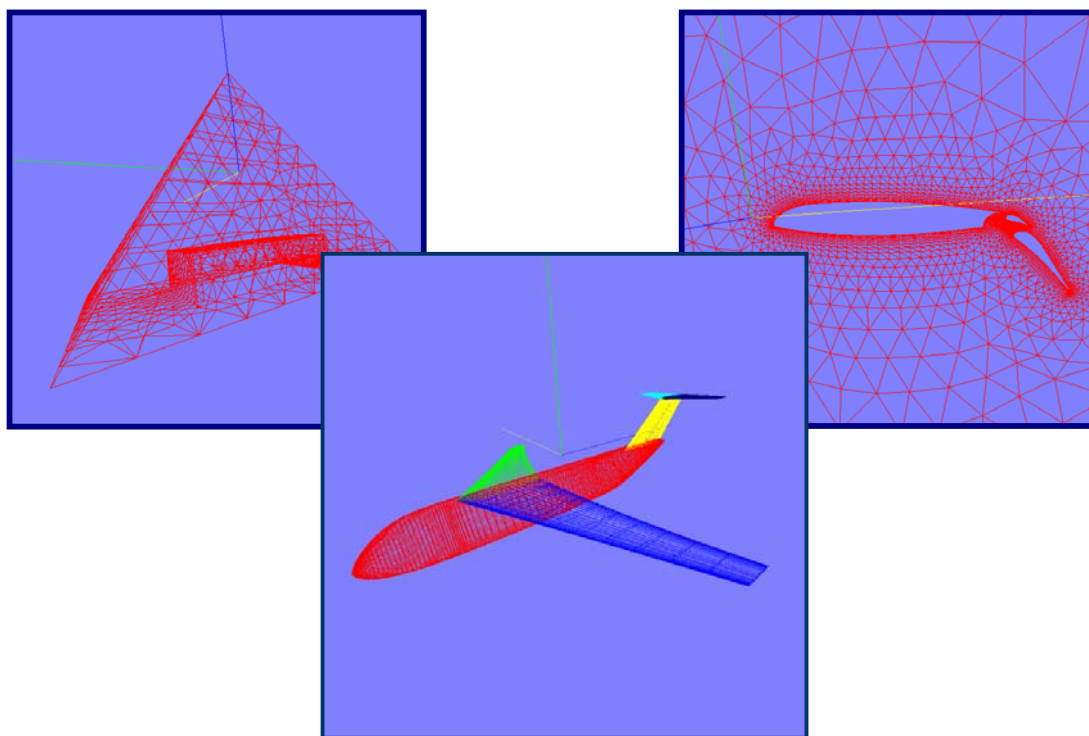
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Τμήμα Μηχανικών Παραγωγής και Διοίκησης

Διπλωματική Εργασία

**Ανάπτυξη Γραφικού Περιβάλλοντος για την
απεικόνιση και διαχείριση τρισδιάστατων πλεγμάτων
και επιφανειών**

Ιωάννης Γ. Κοντολάτης



Εξεταστική Επιτροπή:
Τσουρβελούδης Νικόλαος
Ιωαννίδης Στράτος

Επιβλέπων Καθηγητής:
Ιωάννης Κ. Νικολός

Χανιά 2004

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ευκαιρία της ολοκλήρωσης της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω τον καθηγητή μου Ιωάννη Κ. Νικολό για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα, καθώς και για την αμέριστη βοήθεια και γνώση που μου παρείχε κατά τη διάρκεια της συνεργασίας μας.

Θα ήθελα επίσης να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής, Τσουρβελούδη Νικόλαο και Ιωαννίδη Ευστράτιο, για τις παρατηρήσεις τους και την καθοδήγησή τους.

Ακόμη ευχαριστώ τη συνάδελφο και φίλη, Ντιπτένη Μαρίνα, για τη βοήθειά της στην υλοποίηση της παρούσας εργασίας, όπως και για τις συμβουλές της. Επίσης, θα ήθελα να ευχαριστήσω τους συναδέλφους μου για την άψογη συνεργασία και ανταλλαγή απόψεων που είχαμε.

Ένα μεγάλο ευχαριστώ απευθύνεται ξεχωριστά στην οικογένειά μου που με στήριξε ηθικά και υλικά καθ' όλη τη διάρκεια των σπουδών μου, και βοήθησε ώστε να γίνει πραγματικότητα ένα παιδικό μου όνειρο.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	2
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ	6
1.1 Ψηφιακά Γραφικά (Computer Graphics)	6
1.1.1 Εφαρμογές	7
1.2 Λογισμικό Γραφικών	10
1.2.1 Συναρτήσεις Γραφικών	10
1.2.2 Τυποποιήσεις Λογισμικού	11
1.2.3 Άλλα Πακέτα Γραφικών	12
1.2.4 Microsoft DirectX	13
1.2.5 Σύγκριση OpenGL και DirectX	14
ΚΕΦΑΛΑΙΟ 2: ΕΙΣΑΓΩΓΗ ΣΤΗΝ OpenGL	16
2.1 Η Βασική Σύνταξη της OpenGL	16
2.2 Σχετικές Βιβλιοθήκες	17
2.3 Αρχεία Επικεφαλίδων (Header Files)	18
2.4 Διαχείριση Παραθύρου Απεικόνισης Χρησιμοποιώντας το GLUT	18
ΚΕΦΑΛΑΙΟ 3: ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΥΝΤΕΤΑΓΜΕΝΩΝ	21
3.1 Γενικά	21
3.2 Συστήματα Αναφοράς Συντεταγμένων	23
3.2.1 Δισδιάστατες Καρτεσιανές Συντεταγμένες Οθόνης	23
3.2.2 Πρότυπα Διδιάστατα Καρτεσιανά Συστήματα Αναφοράς	24
3.2.3 Πολικές Συντεταγμένες στο xy -Επίπεδο	24
3.2.4 Πρότυπα Τρισδιάστατα Καρτεσιανά Συστήματα Αναφοράς	25
3.2.5 Τρισδιάστατες Καρτεσιανές Συντεταγμένες Οθόνης	26
3.2.6 Τρισδιάστατα Καμπυλόγραμμα Συστήματα Συντεταγμένων	27
3.2.7 Συμπαγής Γωνία (Solid Angle)	28
ΚΕΦΑΛΑΙΟ 4: ΓΕΩΜΕΤΡΙΚΟΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ	29
4.1 Γενικά	29
4.2 Βασικοί Δισδιάστατοι Γεωμετρικοί Μετασχηματισμοί	29
4.2.1 Δισδιάστατη Μεταφορά	30
4.2.2 Δισδιάστατη Περιστροφή	31
4.2.3 Δισδιάστατη Αλλαγή Κλίμακας	33
4.3 Αναπαραστάσεις Πινάκων και Ομογενείς Συντεταγμένες	35
4.3.1 Ομογενείς Συντεταγμένες	36
4.3.2 Διδιάστατος Πίνακας Μεταφοράς	37
4.3.3 Δισδιάστατος Πίνακας Περιστροφής	37
4.3.4 Δισδιάστατος Πίνακας Αλλαγής Κλίμακας	38
4.4 Αντίστροφοι Μετασχηματισμοί	38
4.5 Σύνθετοι Δισδιάστατοι Μετασχηματισμοί	39
4.5.1 Σύνθετες Δισδιάστατες Μεταφορές	40
4.5.2 Σύνθετες Δισδιάστατες Περιστροφές	40
4.5.3 Σύνθετες Δισδιάστατες Αλλαγές Κλίμακας	41
4.5.4 Γενική Δισδιάστατη Περιστροφή Περί Σημείου (Pivot-Point Rotation)	41

4.5.5 Γενική Δισδιάστατη Αλλαγή Κλίμακας Βάση Σημείου (Fixed-Point Scaling).....	42
4.6 Γεωμετρικοί Μετασχηματισμοί Σε Τρισδιάστατο Χώρο.....	43
4.6.1 Τρισδιάστατη Μεταφορά.....	43
4.6.2 Τρισδιάστατη Περιστροφή.....	44
4.6.2.1 Τρισδιάστατες Περιστροφές Περί Αξόνων Συστήματος Συντεταγμένων.....	45
4.6.2.2 Γενικές Τρισδιάστατες Περιστροφές.....	47
4.6.3 Τρισδιάστατη Αλλαγή Κλίμακας.....	54
4.6.4 Σύνθετοι Τρισδιάστατοι Μετασχηματισμοί.....	56
ΚΕΦΑΛΑΙΟ 5: ΟΠΤΙΚΗ ΚΑΙ ΠΡΟΒΟΛΕΣ.....	58
5.1 Εισαγωγή.....	58
5.2 Έλεγχος της κάμερας.....	58
5.3 Προβολές.....	60
5.3.1 Όγκος οπτικής (view volume).....	60
5.3.2 Ορθοκανονική προβολή.....	60
5.3.3 Προοπτική προβολή.....	61
5.4 Ορισμός της οπτικής οδού (viewport).....	61
5.5 Αντιστροφή του καναλιού μεταφοράς της οπτικής οδού.....	62
ΚΕΦΑΛΑΙΟ 6: ΛΟΓΙΣΜΙΚΟ MeshViewer v1.0.....	63
6.1 Εισαγωγή.....	63
6.2 Διάρθρωση του κώδικα.....	63
6.2.1 Ανάγνωση αρχείων.....	64
6.2.1.1 openDPLFile().....	64
6.2.1.2 open3D_DPLFile().....	66
6.2.1.3 openDATFile().....	67
6.2.2 Σχεδιασμός πλεγμάτων και επιφανειών – Συνάρτηση designMesh()...68	
6.2.3 Σχεδιασμός αξόνων συστήματος συντεταγμένων – Συνάρτηση designAxis().....	70
6.2.4 Σχεδιασμός των 5 βασικών όψεων.....	70
6.2.5 Συναρτήσεις περιστροφής.....	72
6.2.6 Συναρτήσεις μεταφοράς.....	73
6.2.7 Συναρτήσεις αλλαγής κλίμακας.....	74
6.2.8 Συνάρτηση αλλαγής εστίασης του αντικειμένου (zoom in/out).....	75
6.2.9 Συνάρτηση αρχικοποίησης – init().....	76
6.2.10 Συνάρτηση επανασχεδιασμού του παράθυρου απεικόνισης – winReshape().....	78
6.2.11 Συνάρτηση αποθήκευσης εικόνας – writetiff().....	79
6.2.12 Συνάρτηση main().....	79
6.3 Εφαρμογές και μελλοντικές προεκτάσεις του λογισμικού MeshViewer v1.0.....	80
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	81
ΠΑΡΑΡΤΗΜΑ.....	83

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία είχε σαν σκοπό την ανάπτυξη λογισμικού για την απεικόνιση και διαχείριση τρισδιάστατων πλεγμάτων και επιφανειών, το οποίο θα είναι ανεξάρτητο λειτουργικού συστήματος. Ο κώδικας αναπτύχθηκε στη γλώσσα προγραμματισμού C με τη χρήση συναρτήσεων της βιβλιοθήκης γραφικών OpenGL.

Κατά τη συγγραφή της παρούσας εργασίας, θεωρήθηκε ότι ο αναγνώστης έχει γνώσεις προγραμματισμού και μελέτης-σχεδίασης με χρήση ηλεκτρονικού υπολογιστή (CAD). Επιπλέον, μπορεί να ανατρέξει σε οποιαδήποτε βιβλιογραφική αναφορά για επεξηγήσεις ή περαιτέρω εμβάθυνση.

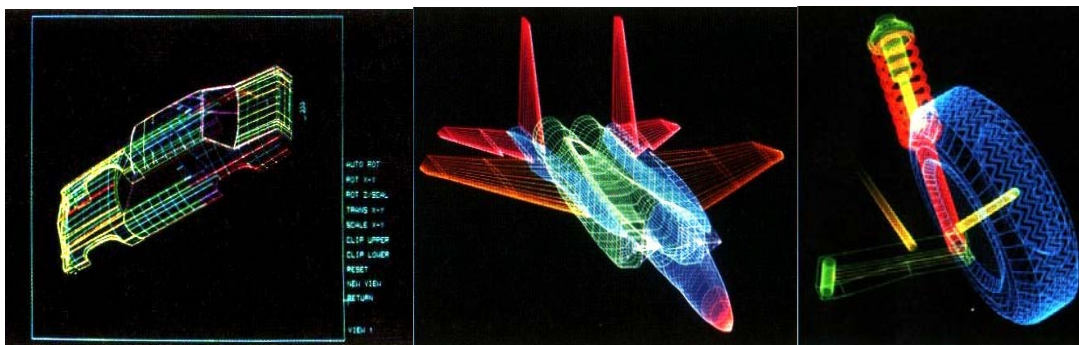
Στο παρόν σύγγραμμα αναφέρονται κάποιες γενικές έννοιες και εφαρμογές των ψηφιακών γραφικών, στη συνέχεια το μαθηματικό υπόβαθρο που χρησιμοποιήθηκε για την ανάπτυξη του λογισμικού, που ονομάστηκε MeshViewer, και τέλος παρουσιάζεται ο τρόπος λειτουργίας και οι δυνατότητες του.

Αναλυτικότερα, στο Κεφάλαιο 1 παρουσιάζεται μία εισαγωγή στο τομέα των ψηφιακών γραφικών, εφαρμογές τους, τυποποιήσεις λογισμικών, και ειδικότερη αναφορά στις βιβλιοθήκες γραφικών OpenGL και Microsoft DirectX. Το Κεφάλαιο 2 αναφέρεται αναλυτικότερα στη βιβλιοθήκη OpenGL, στη σύνταξη των συναρτήσεων της, καθώς και στη βοηθητική βιβλιοθήκη GLUT για τη διαχείριση των παράθυρων απεικόνισης. Τα Κεφάλαια 3 και 4 περιλαμβάνουν το μαθηματικό υπόβαθρο, που αφορά στα συστήματα συντεταγμένων και στους γεωμετρικούς μετασχηματισμούς αντικειμένων τόσο σε διδιάστατο όσο και σε τρισδιάστατο χώρο. Στο Κεφάλαιο 5 περιγράφονται οι βασικές έννοιες οπτικής και τα είδη των προβολών που χρησιμοποιούνται. Τέλος, στο Κεφάλαιο 6 αναλύεται η διάρθρωση του κώδικα που αναπτύχθηκε, καθώς και οι συναρτήσεις της OpenGL που χρησιμοποιήθηκαν. Επίσης, περιγράφονται οι λειτουργίες του λογισμικού MeshViewer v1.0, και πως ο χρήστης μπορεί να το χρησιμοποιήσει, ενώ στο Παράρτημα παρουσιάζονται οι συναρτήσεις της OpenGL που χρησιμοποιήθηκαν.

ΕΙΣΑΓΩΓΗ

1.1 Ψηφιακά Γραφικά (Computer Graphics)

Ο τομέας των ψηφιακών γραφικών παραμένει ένας από τους πιο ραγδαία εξελισσόμενους τομείς της μοντέρνας τεχνολογίας, καθώς τα ψηφιακά γραφικά αποτελούν καθεστώς πλέον στο λογισμικό εφαρμογών (software) και γενικά στα συστήματα υπολογιστών.

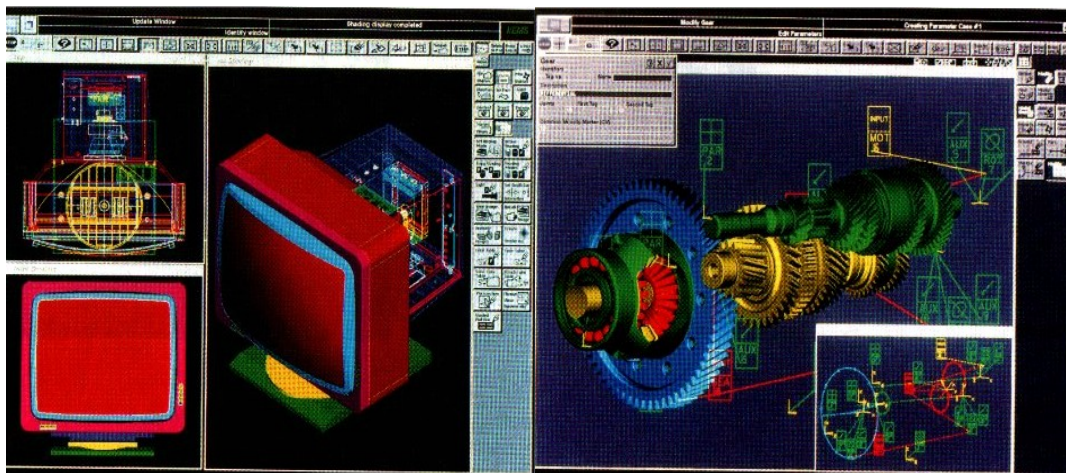


(α)

(β)

(γ)

*Εικόνα 1.1: Περιγράμματα τύπου σύρματος, χρωματικά κωδικοποιημένα.
(α) Αυτοκίνητο (Peritek Corporation) (β) Αεροσκάφος και (γ) Τροχός αυτοκινήτου
(Evans & Sutherland)*

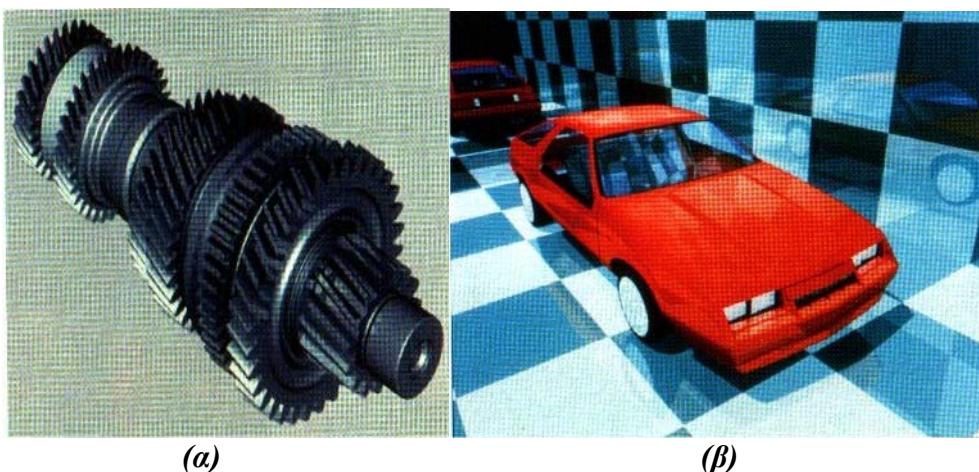


Εικόνα 1.2: Απεικονίσεις πολλαπλών παραθύρων, χρωματικά κωδικοποιημένες από λογισμικό CAD (Intergraph Corporation).

Η εφαρμογή μεθόδων κατασκευής ψηφιακών γραφικών είναι πλέον σύνηθες φαινόμενο στο σχεδιασμό των περισσότερων προϊόντων, στις εκπαιδευτικές συσκευές προσομοίωσης, στην παραγωγή μουσικών βίντεο και τηλεοπτικών διαφημίσεων, στις κινηματογραφικές ταινίες, στην ανάλυση δεδομένων, σε επιστημονικές μελέτες, σε ιατρικές διαδικασίες, όπως και σε αμέτρητες άλλες εφαρμογές. Ένα μεγάλο εύρος από μαθηματικές και προγραμματιστικές τεχνικές, καθώς και ηλεκτρονικές διατάξεις

(hardware) χρησιμοποιούνται ή βρίσκονται υπό ανάπτυξη για τα πεδία εφαρμογών που αναφέρθηκαν.

Ειδικότερα, το μεγαλύτερο μέρος της έρευνας στο τομέα των ψηφιακών γραφικών σήμερα αφορά στη βελτίωση της αποτελεσματικότητας, του ρεαλισμού και της ταχύτητας στην αναπαραγωγή των εικόνων. Για την παραγωγή μίας ρεαλιστικής όψης μίας φυσικής σκηνής, ένα πρόγραμμα γραφικών πρέπει να προσομοιώσει τις επιδράσεις των ανακλάσεων του πραγματικού φωτός και των διαθλάσεων από τα φυσικά αντικείμενα. Για το λόγο αυτό, η τρέχουσα τάση στο τομέα των ψηφιακών γραφικών είναι η συγχώνευση βελτιωμένων διαδικασιών προσέγγισης των φυσικών αρχών σε αλγόριθμους γραφικών, με αποτέλεσμα την καλύτερη προσομοίωση των πολύπλοκων αλληλεπιδράσεων μεταξύ των αντικειμένων και του φωτισμού του περιβάλλοντος. [3].



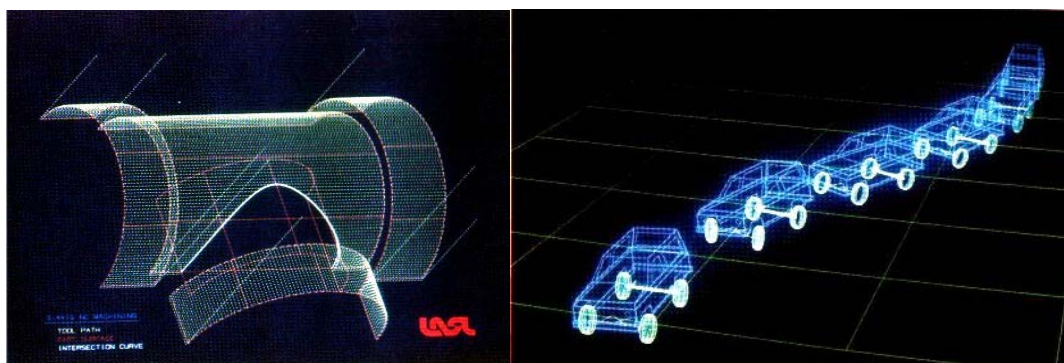
Εικόνα 1.3:Ρεαλιστικές φωτοσκιάσεις. (α)Κιβώτιο ταχυτήτων (Intergraph Corporation) και (β)Αυτοκίνητο (Eric Haines, Autodesk, Inc.)



Εικόνα 1.4:Σκηνή από την ταινία Final Fantasy: The Spirits Within. (Square Pictures, Inc.)

1.1.1 Εφαρμογές

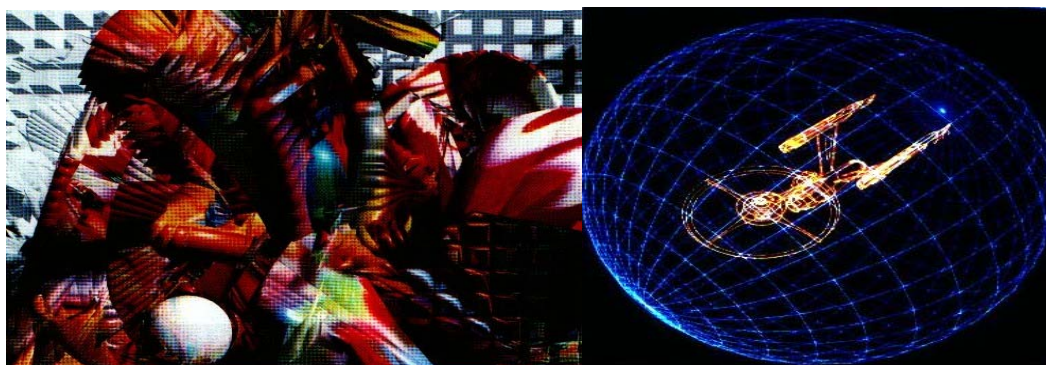
Οι ηλεκτρονικοί υπολογιστές έχουν μετατραπεί σε ισχυρά εργαλεία για την ταχύτατη και οικονομική παραγωγή εικόνων. Ουσιαστικά δεν υπάρχει περιορισμός στη χρήση γραφικών αναπαραστάσεων, οπότε δεν αποτελεί έκπληξη ότι η χρήση των ψηφιακών γραφικών είναι τόσο διαδεδομένη.



(α)

(β)

Εικόνα 1.5:(α)Σχέδιο CAD για τη περιγραφή της αριθμητικά ελεγχόμενης κατεργασίας του αντικειμένου (Los Alamos National Laboratory) και (β)Προσομοίωση των επιδόσεων οχήματος κατά τη διάρκεια αλλαγής πορείας.(Evans & Sutherland and Mechanical Dynamics, Inc.)



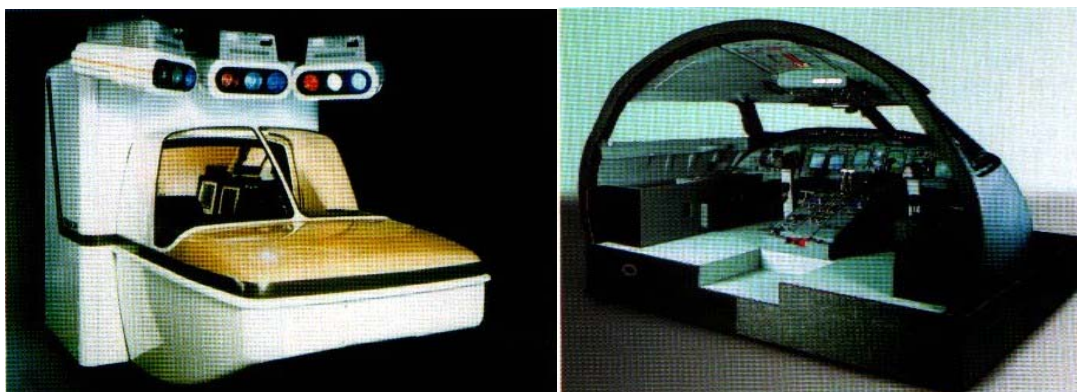
(α)

(β)

(γ)

Εικόνα 1.6:(α)Whigmalarie, ηλεκτρονικός πίνακας (Williams Gallery, 1992 Wynne Ragland,Jr.), (β)Γραφική απεικόνιση για την ταινία Star Trek (Evans & Sutherland), και (γ)Ψηφιακή αναπαραγωγή σκηνής, Knickknack (Pixar)

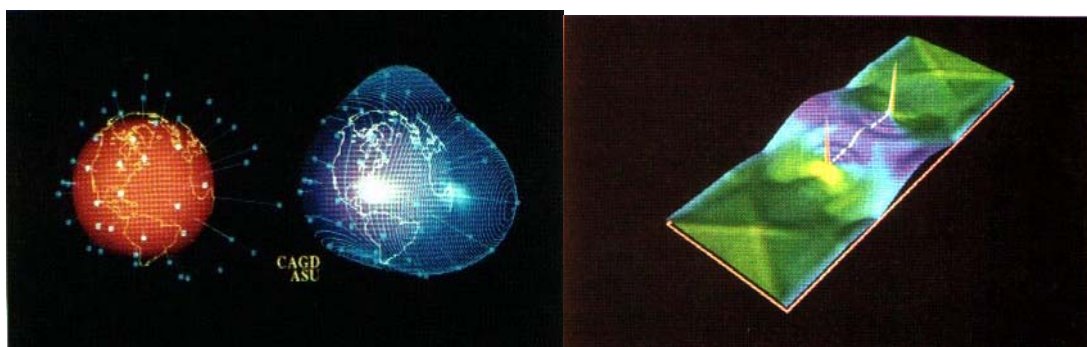
Αν και οι πρώιμες επιστημονικές και τεχνικές εφαρμογές των γραφικών βασίζονταν σε ακριβό και δύσχρηστο εξοπλισμό, τα επιτεύγματα της τεχνολογίας των υπολογιστών κατέστησαν πλέον τα αλληλεπιδρώντα (interactive) ψηφιακά γραφικά ένα ιδιαίτερα πρακτικό και εύχρηστο εργαλείο. Στις μέρες μας, τα ψηφιακά γραφικά χρησιμοποιούνται σε πολλά διαφορετικά πεδία, όπως στην επιστήμη, την τέχνη, τη μηχανική, τις επιχειρήσεις, τη βιομηχανία, την ιατρική, την ψυχαγωγία, τη διαφήμιση, την εκπαίδευση, σε οικιακές εφαρμογές, ενώ με την βοήθεια του Διαδικτύου (Internet) γίνεται εφικτή η μεταβίβασή τους σε ολόκληρο τον κόσμο.



(α)

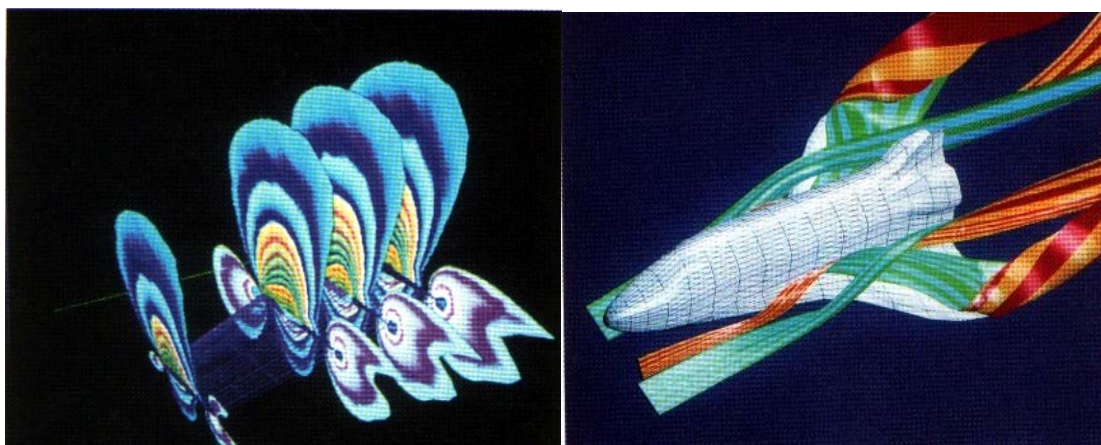
(β)

Εικόνα 1.7: Καμπίνες. (α) Μικρού προσομοιωτή πτήσης και (β) Προσομοιωτή πτήσης διπλού χειρισμού. (Frasca International).



(α)

(β)



(γ)

(δ)

Εικόνα 1.8: Οπτικοποίηση δεδομένων. (α) Μέθοδος απεικόνισης διανομής δεδομένων πάνω σε σφαιρική επιφάνεια (Greg Nelson, Computer Science Department, Arizona State University), (β) Χρωματικά κωδικοποιημένη απεικόνιση πυκνότητας ενεργειακών τάσεων σε διάδοση ρωγμής (National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign), (γ) Προσομοίωση ροής ρευστού σε πτέρυγα αεροσκάφους (Information Technology Institute, Republic of Singapore), και (δ) Οπτικοποίηση ροής γύρω από το διαστημικό λεωφορείο (Sam Uselton, NASA Ames Research Center).

Πιο συγκεκριμένα, τα ψηφιακά γραφικά χρησιμοποιούνται για τη δημιουργία γραφημάτων και διαγραμμάτων, στα συστήματα CAD (Computer-Aided Design), για

την ανάπτυξη περιβαλλόντων εικονικής πραγματικότητας (Virtual Reality), για την οπτικοποίηση δεδομένων, στην εκπαίδευση, στην τέχνη, στην ψυχαγωγία, στην επεξεργασία εικόνων, στην ανάπτυξη γραφικού περιβάλλοντος σε εφαρμογές λογισμικού (Graphical User Interface – GUI). Στις **Εικόνες 1.1-1.8** παρουσιάζονται χαρακτηριστικά παραδείγματα των προαναφερθέντων εφαρμογών.

1.2 Λογισμικό Γραφικών

Υπάρχουν δύο βασικές ταξινομήσεις για το λογισμικό ψηφιακών γραφικών: τα *πακέτα ειδικών εφαρμογών (special-purpose packages)* και τα *πακέτα γενικού προγραμματισμού (general programming packages)*. Η πρώτη κατηγορία σχεδιάστηκε για χρήστες χωρίς γνώσεις προγραμματισμού, που επιθυμούν να παράγουν εικόνες, γραφήματα ή διαγράμματα σε κάποια πεδία εφαρμογών, χωρίς να γνωρίζουν ή να εμπλέκονται στα πρωτόκολλα γραφικών που απαιτούνται για το σκοπό αυτό. Το μέσον αλληλεπίδρασης σε ένα πακέτο ειδικών εφαρμογών είναι τυπικά ένα σύνολο από μενού, που επιτρέπουν στους χρήστες να «επικοινωνούν» με το πρόγραμμα, σύμφωνα με τις ανάγκες τους. Παραδείγματα τέτοιων εφαρμογών είναι τα καλλιτεχνικά προγράμματα ζωγραφικής, καθώς και τα ποικίλα συστήματα CAD.

Στον αντίποδα, ένα πακέτο γενικού προγραμματισμού παρέχει μία βιβλιοθήκη με συναρτήσεις γραφικών, που μπορούν να χρησιμοποιηθούν σε μία γλώσσα προγραμματισμού όπως είναι οι C, C++, Java, ή Fortran. Οι βασικές συναρτήσεις μίας τυπικής βιβλιοθήκης γραφικών συμπεριλαμβάνουν εκείνες για τον καθορισμό των συστατικών της εικόνας (ευθείες γραμμές, πολύγωνα, σφαίρες κ.α.), την ανάθεση των τιμών των χρωμάτων, την επιλογή της οπτικής της σκηνής, καθώς και την εφαρμογή περιστροφών και άλλων μετασχηματισμών στα αντικείμενα. Μερικά παραδείγματα πακέτων γενικού προγραμματισμού είναι η βιβλιοθήκη GL (Graphics Library), η OpenGL, η DirectX, η γλώσσα VRML (Virtual-Reality Modeling Language), η Java 2D, και η Java 3D. [3].

Ένα σύνολο από συναρτήσεις γραφικών συνήθως καλείται *μέσον αλληλεπίδρασης εφαρμογής προγραμματισμού ψηφιακών γραφικών (Computer-Graphics Application Programming Interface – CG API)*, για το λόγο ότι η βιβλιοθήκη παρέχει ένα μέσον αλληλεπίδρασης λογισμικού ανάμεσα σε μία γλώσσα προγραμματισμού και στον υλικό εξοπλισμό (hardware). Έτσι, όταν αναπτύσσεται, για παράδειγμα, μία εφαρμογή σε C++, οι ρουτίνες γραφικών που χρησιμοποιούνται επιτρέπουν την κατασκευή και παρουσίαση μίας εικόνας σε μία συσκευή εξόδου (π.χ. οθόνη, εκτυπωτής).

1.2.1 Συναρτήσεις Γραφικών

Ένα πακέτο γραφικών γενικών εφαρμογών παρέχει στον χρήστη ένα εύρος συναρτήσεων για την δημιουργία και διαχείριση εικόνων. Οι ρουτίνες αυτές μπορούν να ταξινομηθούν γενικά ανάλογα με το αν μεταχειρίζονται έξοδο (output), είσοδο (input), χαρακτηριστικά, μετασχηματισμούς, όψεις, υποδιαίρεση εικόνων, ή γενικό έλεγχο των γραφικών.

Τα βασικά μέρη για την κατασκευή εικόνων αναφέρονται ως *πρωταρχικά δεδομένα εξόδου γραφικών (graphics output primitives)*. Περιλαμβάνουν αλφαριθμητικά χαρακτήρων (character strings) και γεωμετρικές οντότητες, όπως σημεία, ευθείες και καμπύλες, περιοχές πληρωμένες με χρώμα (συνήθως πολύγωνα), και σχήματα ορισμένα με πίνακες (arrays) από σημεία χρώματος. Επιπρόσθετα,

μερικά πακέτα γραφικών παρέχουν συναρτήσεις για την απεικόνιση περισσότερο πολύπλοκων σχημάτων, όπως σφαίρες, κώνοι, και κύλινδροι. Οι ρουτίνες για την δημιουργία πρωταρχικών δεδομένων εξόδου παρέχουν τα βασικά εργαλεία για την κατασκευή εικόνων.

Τα *χαρακτηριστικά γνωρίσματα* είναι οι ιδιότητες των πρωταρχικών δεδομένων εξόδου. Ένα χαρακτηριστικό γνώρισμα περιγράφει πως ένα συγκεκριμένο πρωταρχικό δεδομένο θα απεικονιστεί. Σε αυτά συμπεριλαμβάνονται οι προδιαγραφές των χρωμάτων, οι τύποι των γραμμών και του κειμένου, και τα σχέδια που θα γεμίσουν τις περιοχές.

Το μέγεθος, η θέση, ή ο προσανατολισμός ενός αντικειμένου στη σκηνή μπορούν να αλλάξουν κάνοντας χρήση *γεωμετρικών μετασχηματισμών* (*geometric transformations*). Κάποια πακέτα γραφικών παρέχουν ένα επιπρόσθετο σύνολο συναρτήσεων για την πραγματοποίηση *μετασχηματισμών μοντέλου* (*modeling transformations*), οι οποίες χρησιμοποιούνται για την κατασκευή μίας σκηνής όπου ξεχωριστές για κάθε αντικείμενο περιγραφές δίνονται σε τοπικές συντεταγμένες (βλ. Κεφάλαιο 4). Τέτοια πακέτα συνήθως παρέχουν ένα μηχανισμό για την περιγραφή πολύπλοκων αντικειμένων (όπως ένα ηλεκτρικό κύκλωμα ή ένα ποδήλατο) με μία *ιεραρχική δένδροειδή μορφή*. Άλλα πακέτα απλά παρέχουν τις ρουτίνες γεωμετρικών μετασχηματισμών και αφήνουν τις λεπτομέρειες της μοντελοποίησης στον προγραμματιστή.

Αφού η σκηνή έχει κατασκευαστεί, κάνοντας χρήση των ρουτινών για τον καθορισμό των σχημάτων του αντικειμένου και των χαρακτηριστικών τους, ένα πακέτο γραφικών στη συνέχεια προβάλλει μία όψη της εικόνας στη συσκευή εξόδου. *Μετασχηματισμοί όψεων* (*viewing transformations*) χρησιμοποιούνται για την επιλογή της όψης της σκηνής, το τύπο της προβολής που θα χρησιμοποιηθεί (βλ. Κεφάλαιο 5), και την τοποθεσία στην οθόνη όπου θα απεικονιστεί η όψη. Άλλες ρουτίνες είναι διαθέσιμες για την διαχείριση της περιοχής απεικόνισης σε μία οθόνη με τον καθορισμό της θέσης της, του μεγέθους και της δομής της. Για τρισδιάστατες σκηνές, αναγνωρίζονται τα ορατά αντικείμενα και εφαρμόζονται οι συνθήκες φωτισμού.

Οι εφαρμογές γραφικών με *αλληλεπίδραση* κάνουν χρήση διαφόρων ειδών συσκευών εισόδου, συμπεριλαμβανομένων ποντικιού, πινακίδας, μοχλού. Για τον έλεγχο και την επεξεργασία της ροής δεδομένων από τις συσκευές αυτές χρησιμοποιούνται συναρτήσεις εισόδου.

Τέλος, ένα πακέτο γραφικών περιλαμβάνει ένα αριθμό ρουτινών για απαραίτητες εργασίες, όπως είναι η εκκαθάριση της απεικόνισης στην οθόνη και η αρχικοποίηση παραμέτρων. Οι ρουτίνες αυτές περιγράφονται από τον γενικό τίτλο *λειτουργίες ελέγχου* (*control operations*). [3].

1.2.2 Τυποποιήσεις Λογισμικού

Πρωταρχικός στόχος της τυποποίησης του λογισμικού γραφικών είναι η *μεταφερισιμότητα* (*portability*). Όταν τα πακέτα σχεδιάζονται με τυποποιημένες συναρτήσεις γραφικών, το λογισμικό μπορεί να μεταφέρεται εύκολα από το ένα υπολογιστικό σύστημα στο άλλο και να χρησιμοποιείται σε διαφορετικές υλοποιήσεις και εφαρμογές. Χωρίς τυποποιήσεις, προγράμματα που σχεδιάζονται για έναν συγκεκριμένο εξοπλισμό συχνά δεν μπορούν να μεταφερθούν σε άλλο, χωρίς την εκτεταμένη αναπροσαρμογή τους.

Στην προσπάθεια ανάπτυξης ενός γενικά αποδεκτού προτύπου για τα ψηφιακά γραφικά συνεργάστηκαν διεθνείς και εθνικοί οργανισμοί ανάπτυξης προτύπων σε πολλές χώρες. Η προσπάθεια αυτή οδήγησε στην ανάπτυξη του *Graphical Kernel*

System (GKS) το 1984. Το σύστημα αυτό υιοθετήθηκε ως η πρώτη τυποποίηση για λογισμικό γραφικών από το *Διεθνή Οργανισμό Προτύπων (ISO)* και από πολλούς εθνικούς οργανισμούς προτύπων, συμπεριλαμβανομένου του *Εθνικού Αμερικάνικου Ινστιτούτου Προτύπων (ANSI)*. Αν και το GKS αρχικά σχεδιάστηκε σαν ένα δισδιάστατο πακέτο γραφικών, σύντομα αναπτύχθηκε και μία τρισδιάστατη επέκταση. Η δεύτερη τυποποίηση λογισμικού, που αναπτύχθηκε και εγκρίθηκε από τους διεθνείς οργανισμούς, ήταν το *Programmer's Hierarchical Interactive Graphics Standard (PHIGS)*, το οποίο αποτελεί μία επέκταση του GKS. Τα νέα χαρακτηριστικά που παρέχονταν από το PHIGS ήταν αυξημένες ικανότητες για ιεραρχική μοντελοποίηση αντικειμένων, προδιαγραφές χρωμάτων, φωτοσκίαση (rendering) επιφανειών, και διαχείριση εικόνων. Συνακολούθως, μία επέκταση του PHIGS, που ονομαζόταν PHIGS+, αναπτύχθηκε για να παρέχει δυνατότητες τρισδιάστατης φωτοσκίασης επιφανειών.

Καθώς αναπτύσσονταν τα πακέτα GKS και PHIGS, οι σταθμοί εργασίας για επεξεργασία γραφικών της εταιρίας Silicon Graphics Inc.(SGI) γίνονταν ολοένα και πιο δημοφιλείς. Οι σταθμοί αυτοί χρησιμοποιούσαν ένα σύνολο από ρουτίνες γραφικών, που ονομάζονταν *Graphics Library (GL)*, το οποίο πολύ σύντομα έγινε ένα ευρέως χρησιμοποιούμενο πακέτο στη διεθνή κοινότητα των γραφικών. Έτσι η βιβλιοθήκη εντολών GL κατέληξε να γίνει μια διεθνής τυποποίηση γραφικών. Οι ρουτίνες GL σχεδιάστηκαν για γρήγορη και σε πραγματικό χρόνο φωτοσκίαση και σύντομα το πακέτο αυτό επεκτάθηκε και σε άλλου τύπου υπολογιστικά συστήματα. Ως αποτέλεσμα, αναπτύχθηκε η βιβλιοθήκη *OpenGL* ως μία ανεξάρτητη υλικού εξοπλισμού έκδοση της GL στις αρχές της δεκαετίας του 1990. Το πακέτο αυτό των γραφικών διατηρήθηκε και εκσυγχρονίστηκε από τον οργανισμό *OpenGL Architecture Review Board*, που είναι μία διεθνής εταιρική συνεργασία με αντιπροσώπους από πολλές εταιρίες γραφικών και οργανισμούς. Η βιβλιοθήκη OpenGL έχει σχεδιαστεί ειδικά για αποδοτική επεξεργασία τρισδιάστατων εφαρμογών, αλλά μπορεί ωστόσο να διαχειριστεί περιγραφές δισδιάστατων σκηνών ως ειδική περίπτωση τρισδιάστατων, όπου όλες οι τιμές των z συντεταγμένων είναι μηδενικές.

Οι συναρτήσεις γραφικών σε οποιοδήποτε πακέτο τυπικά ορίζονται ως ένα σύνολο προδιαγραφών, που είναι ανεξάρτητες από οποιαδήποτε γλώσσα προγραμματισμού. Στη συνέχεια καθορίζεται ένας σύνδεσμος με τη συγκεκριμένη, υψηλού επιπέδου, γλώσσα προγραμματισμού. Ο σύνδεσμος αυτός παρέχει το συντακτικό για την πρόσβαση από τη γλώσσα στις διάφορες συναρτήσεις γραφικών. Ο σύνδεσμος κάθε γλώσσας καθορίζεται ώστε να κάνει βέλτιστη χρήση των δυνατοτήτων ανταπόκρισης της γλώσσας και να διαχειρίζεται διάφορα συντακτικά θέματα, όπως είναι τύποι δεδομένων, πέρασμα παραμέτρων και σφάλματα. Οι προδιαγραφές για την υλοποίηση ενός πακέτου γραφικών σε μία συγκεκριμένη γλώσσα καθορίζονται από τον *Διεθνή Οργανισμό Προτύπων*. Οι σύνδεσμοι της OpenGL για τις γλώσσες C/C++ είναι ίδιοι. Άλλοι σύνδεσμοι της OpenGL είναι επίσης διαθέσιμοι, όπως είναι εκείνοι για Ada και Fortran. [3].

1.2.3 Άλλα Πακέτα Γραφικών

Για τον προγραμματισμό ψηφιακών γραφικών έχουν αναπτυχθεί πολλές άλλες βιβλιοθήκες. Κάποιες παρέχουν γενικές ρουτίνες γραφικών, ενώ κάποιες άλλες στοχεύουν σε συγκεκριμένες εφαρμογές ή ειδικούς τομείς των ψηφιακών γραφικών, όπως είναι η δυναμική κίνηση (animation), η εικονική πραγματικότητα, ή τα γραφικά στο Διαδίκτυο.

Ένα πακέτο που καλείται *Open Inventor* παρέχει ένα σύνολο από αντικειμενοστραφείς ρουτίνες για την περιγραφή μίας σκηνής που θα απεικονιστεί με κλήσεις από την OpenGL. Η *Virtual-Reality Modeling Language (VRML)*, που ξεκίνησε ως υποσύνολο του Open Inventor, επιτρέπει τη δημιουργία τρισδιάστατων μοντέλων πραγματικών κόσμων στο Διαδίκτυο. Μπορούν επίσης να κατασκευαστούν εικόνες στον Παγκόσμιο Ιστό (Web), χρησιμοποιώντας βιβλιοθήκες γραφικών που αναπτύχθηκαν για την γλώσσα προγραμματισμού *Java*. Για παράδειγμα, με την *Java 2D* μπορούν να δημιουργηθούν δισδιάστατες σκηνές μέσα σε *Java* εφαρμογές, ή να παραχθούν τρισδιάστατες απεικονίσεις στο Διαδίκτυο με την *Java 3D*. Επίσης, με το *Renderman Interface* της γνωστής εταιρίας παραγωγής ταινιών *Pixar Corporation*, μπορούν να αναπαραχθούν σκηνές με τη χρήση μίας ποικιλίας από μοντέλα φωτισμού. Τέλος, βιβλιοθήκες γραφικών συχνά παρέχονται και σε άλλου τύπου εφαρμογές, όπως είναι τα λογισμικά μαθηματικών εφαρμογών *Mathematica*, *Matlab* και *Maple* [3].

1.2.4 Microsoft DirectX

Πριν από τα Windows, το πιο δημοφιλές λειτουργικό σύστημα για προσωπικούς υπολογιστές (PCs) ήταν το MS-DOS. Οι προγραμματιστές, με την εμφάνιση των Windows, αρχικά αρνήθηκαν να αναπτύξουν εφαρμογές για τη νέα πλατφόρμα λόγω των απαράδεκτων γραφικών και της ηχητικής απόδοσης του νέου λειτουργικού. Η άμεση πρόσβαση στο υλικό μέρος του H/Y (hardware) που προσέφερε το MS-DOS συνδυαζόταν με αρκετή πολυπλοκότητα. Οι εφαρμογές σε MS-DOS έπρεπε να υποστηρίζουν όλη την ποικιλία καρτών εικόνας και ήχου. Αυτό όμως εξανάγκασε τους προγραμματιστές να γράφουν πολύπλοκο κώδικα για την υποστήριξη δεκάδων διαφορετικών ρυθμίσεων μόνο για να παρέχουν ίδιας ποιότητας γραφικά και ήχο για όλους τους προσωπικούς υπολογιστές με όλους τους πιθανούς συνδυασμούς καρτών.

Με τον ερχομό του DirectX το 1995, η Microsoft εφοδίασε τα Windows με την απόδοση που πριν ήταν διαθέσιμη μόνο διαμέσου του MS-DOS, χωρίς πλέον την πολυπλοκότητα της υποστήριξης κάθε διαφορετικής κάρτας. Μέχρι εκείνη τη στιγμή, κάθε εταιρεία υλικού παρείχε τα προϊόντα της με οδηγούς (drivers) για Windows. Το DirectX αναπτύχθηκε από την Microsoft και είναι μία γενική βιβλιοθήκη ρουτινών που μεγιστοποιεί την απόδοση δισδιάστατων και τρισδιάστατων εφαρμογών. Υποστηρίζει πλήρως την επιτάχυνση γραφικών και ήχου με την υποστήριξη του αντίστοιχου υλικού μέρους (κάρτας). Η Microsoft ανέπτυξε το DirectX με στόχο την αγορά βιντεοπαιχνιδιών για τη πλατφόρμα των Windows. Λόγω της τεράστιας επιτυχίας του παραθυρικού λειτουργικού συστήματος της Microsoft, το DirectX εδραιώθηκε και στηρίχθηκε από όλες τις μεγάλες εταιρείες υλικού και λογισμικού. Οι βασικές βιβλιοθήκες του DirectX είναι οι DirectDraw, Direct3D, DirectInput, DirectSound και DirectPlay. Η DirectDraw είναι η θεμελιώδης βιβλιοθήκη γραφικών και διαχειρίζεται χαμηλού επιπέδου γραφικά, όπως διδιάστατα σχήματα. Η Direct3D είναι η βιβλιοθήκη που περιέχει ρουτίνες για τη φωτοσκίαση, τη διαχείριση τρισδιάστατων πολυγώνων, ειδών υφής, φωτορεαλισμού, μετασχηματισμών και της οπτικής της κάμερας. Η DirectInput διαχειρίζεται τις συσκευές εισόδου του υπολογιστή, η DirectSound τον ήχο και ότι σχετίζεται με αυτόν, ενώ τέλος η DirectPlay περιλαμβάνει ρουτίνες για την επικοινωνία δύο υπολογιστών μέσω δικτύου.

Η βιβλιοθήκη Direct3D, μέρος του DirectX, παρουσιάστηκε το 1996 στην έκδοση DirectX 2.0. Η βιβλιοθήκη αυτή σχεδιάστηκε για να δώσει πρόσβαση στις

αυξημένες δυνατότητες γραφικών των καρτών επιτάχυνσης τρισδιάστατων γραφικών, ενόσω προωθούσε την ανεξαρτησία συσκευών, παρέχοντας ένα κοινό μέσον αλληλεπίδρασης στους προγραμματιστές. Ως παράδειγμα, ένα μέρος κώδικα γραμμένο κατάλληλα για Direct3D θα δουλεύει σε συσκευές που υποστηρίζουν Direct3D τώρα αλλά και στο μέλλον.

Πηγαίνοντας λίγο πιο πίσω ιστορικά, στις αρχές της δεκαετίας του 1990, πολλές από τις τρισδιάστατες μηχανές γραφικών (3D graphics engine) προσωπικών υπολογιστών κατασκευάστηκαν στη Μεγάλη Βρετανία. Υπήρχε η πολύ γνωστή *Renderware* (www.renderware.com) και η *BRender* της εταιρίας *Argonaut* (www.argonaut.com), η οποία εισήχθη το 1994 στο λειτουργικό σύστημα OS/2, καθώς και μία μικρή βρετανική εταιρεία που ονομαζόταν *RenderMorphics*. Η *RenderMorphics* ιδρύθηκε το 1993 από τους Servan Keondjian, Kate Seekings, και Doug Rabson και παρήγαγε ένα προϊόν που ονομάστηκε *Reality Lab*. Στην έκθεση SIGGRAPH 94 που συμμετείχαν για πρώτη φορά, εντοπίστηκαν από την Microsoft, και η *RenderMorphics* αποκτήθηκε τον Φεβρουάριο του 1995. Μετά την απόκτηση της *RenderMorphics*, η Microsoft ανέμειξε το *Reality Lab* στην DirectX οικογένεια των μέσων αλληλεπίδρασης προγραμματιστικών εφαρμογών (*Application Programming Interfaces – APIs*). Το *Immediate Mode*, συστατικό του *Reality Lab*, απορρόφησε το πρότυπο τρισδιάστατο API των Windows τη δεδομένη στιγμή, το 3D-DDI, το οποίο είχε δημιουργηθεί από τον Michael Abrash.

Μέχρι τον ερχομό του DirectX 8.0, η βιβλιοθήκη Direct3D αποτελούταν από δύο ξεχωριστά APIs: το *Retained Mode* και το *Immediate Mode*. Εκείνη τη χρονική στιγμή, το *Immediate Mode* API ήταν δύσκολο στη χρήση, αλλά ήταν ένα ευέλικτο, χαμηλού επιπέδου API που “έτρεχε” όσο το δυνατόν πιο αποτελεσματικά. Το *Retained Mode* είχε κατασκευαστεί στη κορυφή του *Immediate Mode* και παρείχε επιπρόσθετες λειτουργίες, όπως ιεραρχία πλαισίων (frames) και δυναμική κίνηση εικόνων (animation). Το *Retained Mode* ήταν ευκολότερο στην εκμάθηση και χρήση από το *Immediate Mode*, αλλά οι προγραμματιστές επιθυμούσαν την επιπρόσθετη απόδοση και ευελιξία που παρείχε το *Immediate Mode*. Η ανάπτυξη του *Retained Mode* API πάγωσε με την έκδοση του DirectX 6.0. Οι μεγαλύτερες αλλαγές μεταξύ της έκδοσης 6.0 και 7.0 του Direct3D *Immediate Mode* ήταν η υποστήριξη μετασχηματισμών και φωτισμού με επιτάχυνση από το υλικό μέρος, και η αναδιοργάνωση φώτων, υλικών και οπτικών οδών (viewports) των αντικειμένων, που από τότε και στο εξής θέτονται κατευθείαν καλώντας τις μεθόδους του *IDirect3DDevice7* και μέσω ενός ειδικού μέσου αλληλεπίδρασης για την πρόσβαση σε είδη υφής (texture). Το μέσο αλληλεπίδρασης του *IDirect3DDrawSurface7* παρείχε επίσης ένα ευκολότερο τρόπο για τη διαχείριση των ειδών υφής.

Με τον ερχομό του DirectX 8.0 SDK ήρθαν και οι μεγαλύτερες βελτιώσεις στην ιστορία του Direct3D. Η αρχικοποίηση, κατανομή και διαχείριση δεδομένων απλοποιήθηκε, με την ενσωμάτωση των βιβλιοθηκών *DirectDraw* και *Direct3D* σε ένα μέσο αλληλεπίδρασης που ονομάζεται *DirectX Graphics*, και το οποίο οδήγησε σε λιγότερη δέσμευση μνήμης και απλούστερο προγραμματιστικό μοντέλο. [21].

1.2.5 Σύγκριση OpenGL και DirectX

Η OpenGL και το DirectX είναι τα δύο πιο δημοφιλή APIs που υποστηρίζονται από τις νέες συσκευές. Η σημαντικότερη διαφορά μεταξύ των δύο τυποποιήσεων είναι ότι το DirectX είναι ένα πλήρες API με ρουτίνες για τη διαχείριση δεδομένων εισόδου, ήχου, μουσικής, λειτουργιών δικτύου, και φυσικά γραφικών, προσανατολισμένο κυρίως σε εφαρμογές βιντεοπαιχνιδιών. Από την άλλη

μεριά, η OpenGL έχει μία πολύπλοκη συλλογή από συναρτήσεις γραφικών, αλλά διαχειρίζεται μόνο αυτά, αποδοτικότερα όμως συγκριτικά με το DirectX. Σχεδόν πάντα, όλες οι λειτουργίες θα είναι ταχύτερες και τα αποτελέσματα ποιοτικότερα με χρήση της OpenGL. Επιπρόσθετα, το DirectX είναι περιορισμένο να χρησιμοποιείται αποκλειστικά σε πλατφόρμες Windows, ενώ η OpenGL υιοθετήθηκε από τις περισσότερες βιομηχανίες ως μία ανεξάρτητη λειτουργικού συστήματος, επαγγελματικού επιπέδου, βιβλιοθήκη γραφικών [8].

Για τους λόγους αυτούς, και έπειτα από σύγκριση των δύο APIs στην πράξη, για την ανάπτυξη της εφαρμογής που θα περιγραφεί στην παρούσα έκθεση (βλ. Κεφάλαιο 6) χρησιμοποιήθηκε η βιβλιοθήκη συναρτήσεων της OpenGL.

ΕΙΣΑΓΩΓΗ ΣΤΗΝ OpenGL

Στην OpenGL παρέχεται μία βασική βιβλιοθήκη συναρτήσεων για τον καθορισμό των πρωταρχικών γραφικών, των χαρακτηριστικών, των γεωμετρικών μετασχηματισμών, των μετασχηματισμών όψεων κ.α. Όπως σημειώθηκε στο προηγούμενο κεφάλαιο, η OpenGL σχεδιάστηκε ώστε να είναι ανεξάρτητη υλικού εξοπλισμού (και λειτουργικού συστήματος), συνεπώς πολλές λειτουργίες, όπως οι ρουτίνες εισόδου και εξόδου δεδομένων, δεν συμπεριλαμβάνονται στη βασική βιβλιοθήκη. Εντούτοις, οι ρουτίνες εισόδου και εξόδου καθώς και πολλές επιπρόσθετες συναρτήσεις είναι διαθέσιμες σε βοηθητικές βιβλιοθήκες που αναπτύχθηκαν ειδικά για προγράμματα που χρησιμοποιούν την OpenGL.

2.1 Η Βασική Σύνταξη της OpenGL

Τα ονόματα των συναρτήσεων της βασικής βιβλιοθήκης της OpenGL (OpenGL basic library ή OpenGL core library) έχουν το πρόθεμα *gl*, και κάθε συνιστώσα λέξη στο όνομα της εκάστοτε συνάρτησης έχει το πρώτο της γράμμα κεφαλαίο, όπως για παράδειγμα οι εξής συναρτήσεις: **glBegin**, **glClear**, **glCopyPixels**, **glPolygonMode**. Συγκεκριμένες συναρτήσεις απαιτούν ένα (ή περισσότερα) από τα ορίσματά τους να προσδιορίζονται από μία συμβολική σταθερά. Ως παράδειγμα, το όνομα ή η τιμή μίας παραμέτρου, ή μία συγκεκριμένη κατάσταση. Όλες αυτές οι σταθερές ξεκινούν με τα κεφαλαία γράμματα GL. Επιπρόσθετα, οι συνιστώσες λέξεις στο όνομα μίας σταθεράς γράφονται με κεφαλαία γράμματα, και η υπογράμμιση (_) χρησιμοποιείται ως διαχωριστής ανάμεσά τους. Ακολουθούν μερικά παραδείγματα από τις εκατοντάδες των συμβολικών σταθερών που είναι διαθέσιμες για χρήση με τις συναρτήσεις της OpenGL: **GL_2D**, **GL_RGB**, **GL_CCW**, **GL_POLYGON**, **GL_AMBIENT_AND_DIFFUSE**.

Οι συναρτήσεις της OpenGL απαιτούν συγκεκριμένους τύπους δεδομένων. Ως παράδειγμα, μία παράμετρος μίας συνάρτησης OpenGL μπορεί να αναμένει μία τιμή που είναι ορισμένη ως ακέραιος 32-bit. Αλλά το προδιαγραφόμενο μέγεθος ενός ακεραίου μπορεί να διαφέρει σε διαφορετικά υπολογιστικά συστήματα. Για τη δήλωση ενός συγκεκριμένου τύπου δεδομένων, η OpenGL χρησιμοποιεί ειδικά κατασκευασμένα, ονόματα τύπων δεδομένων, όπως είναι: **GLbyte**, **GLshort**, **GLint**, **GLfloat**, **GLdouble**, **GLboolean**. Το όνομα του εκάστοτε τύπου δεδομένων αρχίζει με τα κεφαλαία γράμματα GL και το υπόλοιπο μέρος της λέξης είναι το όνομα ενός από τους τυποποιημένους τύπους δεδομένων, γραμμένος όμως με μικρά γράμματα.

Σε μερικά ορίσματα των συναρτήσεων της OpenGL μπορούν να ανατεθούν τιμές χρησιμοποιώντας ένα πίνακα, που περιέχει μία λίστα του συνόλου των τιμών των δεδομένων. Η εναλλακτική αυτή μέθοδος χρησιμοποιείται για τον καθορισμό μίας λίστας από τιμές ως δείκτη (pointer) σε ένα πίνακα (array), αποφεύγοντας έτσι το καθορισμό κάθε στοιχείου της λίστας ρητά ως δήλωση παραμέτρου. Ένα τυπικό παράδειγμα της χρήσης της συγκεκριμένης εναλλακτικής είναι στο καθορισμό των τιμών των xyz συντεταγμένων. [3].

2.2 Σχετικές Βιβλιοθήκες

Επιπλέον της βασικής βιβλιοθήκης της OpenGL, υπάρχει ένας αριθμός από συνεργαζόμενες βιβλιοθήκες, που χρησιμοποιούνται για τη διαχείριση ειδικών λειτουργιών. Η βιβλιοθήκη *OpenGL Utility (GLU)* παρέχει ρουτίνες για τη δημιουργία των πινάκων όψεων και προβολών, την περιγραφή πολύπλοκων αντικειμένων με προσεγγίσεις γραμμών και πολύγωνων, την απεικόνιση τετραγωνικών καμπύλων (quadratic curves) και B-Splines χρησιμοποιώντας γραμμικές προσεγγίσεις, τη διαχείριση των λειτουργιών της φωτοσκίασης επιφανειών, και άλλες πολύπλοκες εργασίες. Κάθε εγκατάσταση της OpenGL περιέχει τη βιβλιοθήκη GLU, ενώ τα ονόματα όλων των GLU συναρτήσεων ξεκινούν με το πρόθεμα *glu*. Υπάρχει επίσης ένα αντικειμενοστραφές σύνολο εργαλείων βασισμένο στην OpenGL και γραμμένο σε C++, που ονομάζεται *Open Inventor* και παρέχει ρουτίνες και προκαθορισμένα σχήματα αντικειμένων για την ανάπτυξη αλληλεπιδρώντων εφαρμογών τρισδιάστατης σχεδίασης.

Για τη δημιουργία μίας απεικόνισης γραφικών χρησιμοποιώντας την OpenGL, απαιτείται πρώτα να δημιουργηθεί ένα παράθυρο απεικόνισης (*display window*) στην οθόνη του Η/Υ. Το παράθυρο αυτό είναι απλά η ορθογωνική περιοχή της οθόνης, στην οποία θα πραγματοποιείται η απεικόνιση. Η δημιουργία του παραθύρου απεικόνισης δεν μπορεί να γίνει απευθείας με τις βασικές συναρτήσεις της OpenGL, αφού η συγκεκριμένη βιβλιοθήκη περιέχει μόνο συναρτήσεις γραφικών ανεξάρτητες συσκευών και εξοπλισμού, ενώ οι λειτουργίες διαχείρισης των παραθύρων εξαρτώνται από το υπολογιστικό σύστημα που χρησιμοποιείται. Παρ' όλα αυτά, υπάρχουν αρκετές βιβλιοθήκες για παραθυρικά συστήματα που υποστηρίζουν συναρτήσεις OpenGL για διάφορα υπολογιστικά μηχανήματα. Η *OpenGL Extension to the X Window System (GLX)* παρέχει ένα σύνολο ρουτινών που έχουν ως πρόθεμα τα γράμματα *glx*. Στα συστήματα της *Apple* χρησιμοποιείται το μέσον αλληλεπίδρασης *Apple GL (AGL)* για την διαχείριση των παραθύρων. Τα ονόματα των συναρτήσεων για αυτή τη βιβλιοθήκη έχουν ως πρόθεμα το *agl*. Για τα λειτουργικά συστήματα της οικογένειας *Microsoft Windows*, οι ρουτίνες *WGL* παρέχουν ένα *Windows-to-OpenGL* μέσον αλληλεπίδρασης και έχουν ως πρόθεμα το *wgl*. Το *Presentation Manager to OpenGL (PGL)* είναι ένα μέσον αλληλεπίδρασης για τα συστήματα *IBM OS/2*, και χρησιμοποιεί ως πρόθεμα για τις ρουτίνες της βιβλιοθήκης το *pgl*. Τέλος, το *OpenGL Utility Toolkit (GLUT)* παρέχει μία βιβλιοθήκη συναρτήσεων για την αλληλεπίδραση με οποιοδήποτε παραθυρικό περιβάλλον. Οι συναρτήσεις της βιβλιοθήκης GLUT έχουν ως πρόθεμα το *glut*, ενώ και αυτή η βιβλιοθήκη περιέχει μεθόδους για την περιγραφή και φωτοσκίαση τετραγωνικών καμπύλων και επιφανειών. [3].

Επειδή η βιβλιοθήκη GLUT είναι ένα μέσον αλληλεπίδρασης σε διαφορετικά παραθυρικά συστήματα, η χρήση του GLUT εξασφαλίζει ότι οι εφαρμογές θα είναι ανεξάρτητες συσκευών και λειτουργικού συστήματος. Κάτι τέτοιο ήταν άλλωστε και ένας από τους βασικούς στόχους στην ανάπτυξης της εφαρμογής, που θα περιγραφεί στην παρούσα έκθεση (βλ. Κεφάλαιο 6), συνεπώς προτιμήθηκε η χρήση της συγκεκριμένης βιβλιοθήκης. Πληροφορίες σχετικές με την τελευταία έκδοση της GLUT, καθώς και διαδικασίες φόρτωσης του πηγαίου κώδικα είναι διαθέσιμα στην ιστοσελίδα <http://reality.sgi.com/opengl/glut3/glut3.html>.

2.3 Αρχεία Επικεφαλίδων (Header Files)

Σε οποιοδήποτε κώδικα ενός προγράμματος γραφικών, απαιτείται να συμπεριληφθεί το αρχείο επικεφαλίδων για τον πυρήνα της βιβλιοθήκης OpenGL. Για τις περισσότερες εφαρμογές χρειάζεται και η βιβλιοθήκη GLU, ενώ πρέπει να προστεθεί και το αρχείο για το παραθυρικό σύστημα. Ως παράδειγμα, εάν χρησιμοποιείται κάποια πλατφόρμα Microsoft Windows, το αρχείο επικεφαλίδων που έχει πρόσβαση στις ρουτίνες της WGL είναι το *windows.h*. Το αρχείο αυτό θα πρέπει να τοποθετηθεί στο πηγαίο κώδικα πριν από τα αρχεία επικεφαλίδων των OpenGL και GLU, γιατί περιέχει μακροεντολές, που χρειάζονται από την έκδοση για τα Microsoft Windows των βιβλιοθηκών της OpenGL. Έτσι, ο πηγαίος κώδικας σε αυτή την περίπτωση θα ξεκινά:

```
#include <windows.h>
#include <GL/gl.h>
#include <GL/glu.h>
```

Ωστόσο, εάν χρησιμοποιηθεί το GLUT για την διαχείριση των παραθυρικών λειτουργιών, δεν χρειάζεται να συμπεριληφθούν τα αρχεία *gl.h* και *glu.h*, για το λόγο ότι το GLUT εξασφαλίζει ότι τα τελευταία θα περιληφθούν σωστά. Συνεπώς, τα αρχεία επικεφαλίδων για τις OpenGL και GLU μπορούν να αντικατασταθούν με:

```
#include <GL/glut.h>
```

Μπορούν να συμπεριληφθούν και τα αρχεία *gl.h* και *glu.h*, αλλά κάτι τέτοιο θα ήταν πλεονασμός και μπορεί να επηρεάσει την συμβατότητα του προγράμματος. Στη συνέχεια, κατά την γραφή του πηγαίου κώδικα, ακολουθούν τα υπόλοιπα αρχεία επικεφαλίδων, που απαιτούνται από την γλώσσα C/C++, όπως:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

2.4 Διαχείριση Παραθύρου Απεικόνισης Χρησιμοποιώντας το GLUT

Εφόσον γίνεται χρήση του OpenGL Utility Toolkit για την διαχείριση του παραθυρικού περιβάλλοντος και την απεικόνιση, το πρώτο βήμα είναι η αρχικοποίηση του GLUT. Η συνάρτηση αρχικοποίησης μπορεί επίσης να επεξεργαστεί οποιαδήποτε *ορίσματα γραμμής εντολών (command-line arguments)*. Η αρχικοποίηση του GLUT πραγματοποιείται με τη δήλωση:

```
glutInit (&argc, argv);
```

Στη συνέχεια, δηλώνεται ότι ένα παράθυρο απεικόνισης πρόκειται να δημιουργηθεί στην οθόνη με ένα δοσμένο όνομα, π.χ. *example program*, στη μπάρα τίτλου. Αυτό πραγματοποιείται με την συνάρτηση:

```
glutCreateWindow ("example program");
```

όπου το μοναδικό όρισμα για την συνάρτηση αυτή μπορεί να είναι οποιοσδήποτε αλφαριθμητικός χαρακτήρας.

Συνεχίζοντας, χρειάζεται να καθοριστεί τι πρόκειται να περιέχει το παράθυρο απεικόνισης. Για το σκοπό αυτό δημιουργείται μία εικόνα, χρησιμοποιώντας τις συναρτήσεις της OpenGL και τροφοδοτείται στην GLUT ρουτίνα *glutDisplayFunc*, η οποία αναθέτει την εικόνα στο παράθυρο απεικόνισης. Ως παράδειγμα, υποτίθεται ότι ο OpenGL κώδικας που περιγράφει ένα ευθύγραμμο τμήμα περιλαμβάνεται σε μία ρουτίνα που ονομάζεται *lineSegment*. Εν συνεχεία, η ακόλουθη κλήση συνάρτησης θα περάσει την περιγραφή του ευθύγραμμου τμήματος στο παράθυρο απεικόνισης:

```
glutDisplayFunc (lineSegment);
```

Ωστόσο, το παράθυρο απεικόνισης δεν βρίσκεται ακόμη στην οθόνη. Απαιτείται μία ακόμη συνάρτηση του GLUT για να ολοκληρώσει τις διαδικασίες επεξεργασίας του παραθύρου. Μετά την εκτέλεση της ακόλουθης δήλωσης, όλα τα παράθυρα απεικόνισης που έχουν δημιουργηθεί, συμπεριλαμβανομένου και του γραφικού περιεχομένου τους, ενεργοποιούνται:

```
glutMainLoop( );
```

Η συνάρτηση αυτή απαιτείται να είναι η τελευταία του προγράμματος. Απεικονίζει τα αρχικά γραφικά και θέτει το πρόγραμμα σε έναν ατέρμονο βρόγχο, που ελέγχει για δεδομένα από συσκευές εισόδου όπως το ποντίκι ή το πληκτρολόγιο. Εάν δεν υπάρξουν δεδομένα εισόδου, το πρόγραμμα θα συνεχίσει να παρουσιάζει την συγκεκριμένη εικόνα στο παράθυρο απεικόνισης, μέχρι το τελευταίο να κλείσει από τον χρήστη.

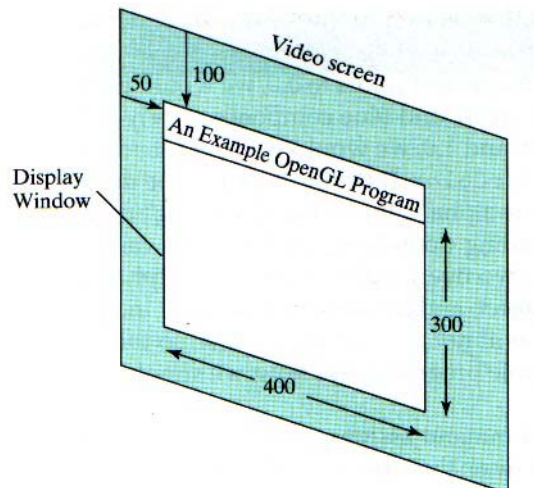
Αν και το παράθυρο απεικόνισης που δημιουργήθηκε θα εμφανίζεται σε προεπιλεγμένη θέση και μέγεθος, οι παράμετροι αυτές μπορούν να ρυθμιστούν, χρησιμοποιώντας επιπρόσθετες συναρτήσεις του GLUT. Κάνοντας χρήση της συνάρτησης *glutInitWindowPosition*, ορίζεται μία αρχική θέση για την άνω αριστερή γωνία του παραθύρου απεικόνισης. Η θέση αυτή προδιαγράφεται με ακέραιες συντεταγμένες οθόνης, η αρχή των οποίων είναι στην άνω αριστερή γωνία της οθόνης. Ως παράδειγμα, η ακόλουθη δήλωση καθορίζει ότι η άνω αριστερή γωνία του παραθύρου απεικόνισης θα τοποθετηθεί 50 εικονοστοιχεία (pixels) δεξιά από την αριστερή πλευρά της οθόνης και 100 κάτω από την άνω πλευρά της:

```
glutInitWindowPosition (50, 100);
```

Όμοια, η συνάρτηση *glutInitWindowSize* χρησιμοποιείται για να καθορίσει το αρχικό πλάτος και ύψος σε εικονοστοιχεία του παραθύρου απεικόνισης. Έτσι, ορίζεται ένα παράθυρο απεικόνισης με αρχικό πλάτος 400 εικονοστοιχείων και ύψος 300 (**Εικόνα 2.1**) με την δήλωση:

```
glutInitWindowSize (400, 300);
```

Αφότου το παράθυρο απεικόνισης βρίσκεται στην οθόνη, είναι δυνατό να αλλάξει η τοποθεσία και το μέγεθός του από το χρήστη.



Εικόνα 2.1: Παράμετροι παραθύρου απεικόνισης. [3]

Επιπρόσθετα, μπορούν να εφαρμοστούν και άλλες λειτουργίες για το παράθυρο απεικόνισης, όπως είναι η προσωρινή αποθήκευση στοιχείων (*buffering*) και η επιλογή μεθόδου χρωμάτων (*color mode*), με τη συνάρτηση *glutInitDisplayMode*. Τα ορίσματα για τη συνάρτηση αυτή είναι συμβολικές GLUT σταθερές. Ως παράδειγμα, η ακόλουθη εντολή ορίζει ότι θα χρησιμοποιηθεί μία *μονή προσωρινή αποθήκη στοιχείων ανανέωσης* (*single refresh buffer*) για το παράθυρο απεικόνισης και η RGB [red (κόκκινο), green (πράσινο), blue (μπλε)] μέθοδος χρώματος για την επιλογή των τιμών χρώματος:

```
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
```

Οι τιμές των σταθερών, που δηλώνονται σε αυτή τη συνάρτηση, συνδέονται χρησιμοποιώντας το λογικό τελεστή (*OR*). Για την ακρίβεια, η μονή προσωρινή αποθήκη στοιχείων και η RGB μέθοδος χρώματος είναι οι προεπιλεγμένες εναλλακτικές μέθοδοι. Μπορούν να χρησιμοποιηθούν σταθερές για την επιλογή διαφορετικής μεθόδου χρώματος, καθώς και εναλλακτικές μέθοδοι απεικόνισης, όπως η *διπλή προσωρινή αποθήκευση στοιχείων* (*double buffering*) [3].

ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΥΝΤΕΤΑΓΜΕΝΩΝ

3.1 Γενικά

Για την αναπαραγωγή μίας εικόνας, χρησιμοποιώντας ένα προγραμματιστικό πακέτο, αρχικά χρειάζεται να δοθούν οι γεωμετρικές περιγραφές των αντικειμένων που πρόκειται να απεικονιστούν. Οι περιγραφές αυτές καθορίζουν τις τοποθεσίες και τα σχήματα των αντικειμένων. Ως παράδειγμα, ένα κουτί ορίζεται από τις θέσεις των κορυφών του (vertices), ενώ μία σφαίρα από τη θέση του κέντρου και την ακτίνα της. Με ελάχιστες εξαιρέσεις, τα γενικά πακέτα γραφικών απαιτούν οι γεωμετρικές περιγραφές να καθορίζονται με ένα τυποποιημένο, δεξιόστροφο, Καρτεσιανό σύστημα συντεταγμένων. Εάν οι τιμές των συντεταγμένων για μία εικόνα δίνονται σε ένα άλλο σύστημα (σφαιρικό, υπερβολικό, κ.λπ.), πρέπει να μετασχηματιστούν σε Καρτεσιανές συντεταγμένες πριν να αποτελέσουν δεδομένα εισόδου στο πακέτο γραφικών. Κάποια πακέτα, που έχουν σχεδιαστεί για εξειδικευμένες εφαρμογές, μπορεί να επιτρέπουν τη χρήση άλλων συστημάτων συντεταγμένων, που είναι κατάλληλα για αυτές τις εφαρμογές.

Γενικά, στη διαδικασία κατασκευής και απεικόνισης μίας σκηνής χρησιμοποιούνται πολλά διαφορετικά Καρτεσιανά συστήματα αναφοράς. Αρχικά, ορίζονται τα σχήματα καθενός αντικειμένου ξεχωριστά, όπως είναι ένα δέντρο, ένα αεροσκάφος, σε ξεχωριστό σύστημα αναφοράς για το καθένα. Οι συντεταγμένες σε αυτά τα τοπικά συστήματα αναφοράς ονομάζονται *συντεταγμένες μοντέλου* (*modeling coordinates*), ή μερικές φορές *τοπικές* (*local coordinates*) ή *κύριες συντεταγμένες* (*master coordinates*). Μόλις καθοριστούν ξεχωριστά τα σχήματα των αντικειμένων, στη συνέχεια μπορεί να κατασκευαστεί η σκηνή, με την τοποθέτηση των αντικειμένων στις κατάλληλες τοποθεσίες με την χρήση ενός συστήματος αναφοράς της σκηνής. Οι αντίστοιχες συντεταγμένες ονομάζονται *παγκόσμιες συντεταγμένες* (*world coordinates*). Το βήμα αυτό περιλαμβάνει τον μετασχηματισμό των ανεξάρτητων συντεταγμένων μοντέλου σε συγκεκριμένες θέσεις και τους αναγκαίους προσανατολισμούς στο παγκόσμιο σύστημα αναφοράς. Ως παράδειγμα, ένα ποδήλατο μπορεί να κατασκευαστεί με τον καθορισμό κάθε εξαρτήματός του (τροχοί, σκελετός, σέλα, τιμόνι, γρανάζια, αλυσίδα, πετάλια) σε ένα ξεχωριστό σύστημα αναφοράς μοντέλου. Στη συνέχεια, τα εξαρτήματα συνδέονται στο παγκόσμιο σύστημα αναφοράς. Εάν και οι δύο τροχοί έχουν το ίδιο μέγεθος, χρειάζεται η περιγραφή του ενός σε ένα τοπικό σύστημα συντεταγμένων, έπειτα η περιγραφή του τροχού τοποθετείται στη περιγραφή του ποδηλάτου στο παγκόσμιο σύστημα σε δύο τοποθεσίες. Για σκηνές που δεν είναι τόσο πολύπλοκες, τα εξαρτήματα ενός αντικειμένου μπορούν να προσδιοριστούν κατευθείαν στις παγκόσμιες συντεταγμένες του ολικού αντικειμένου, παρακάμπτοντας τα βήματα των συντεταγμένων μοντέλου και των μετασχηματισμών. Οι γεωμετρικές περιγραφές σε συντεταγμένες μοντέλου και σε παγκόσμιες μπορούν να δοθούν σε οποιαδήποτε βολική μορφή ακεραίου ή κινητής υποδιαστολής, χωρίς να ληφθούν υπόψη οι περιορισμοί μίας συγκεκριμένης συσκευής εξόδου. Για κάποιες σκηνές, μπορεί να απαιτείται ο καθορισμός της γεωμετρίας του αντικειμένου σε κλάσματα του ποδιού, ενώ σε άλλες εφαρμογές να χρησιμοποιούνται χιλιοστά του μέτρου.

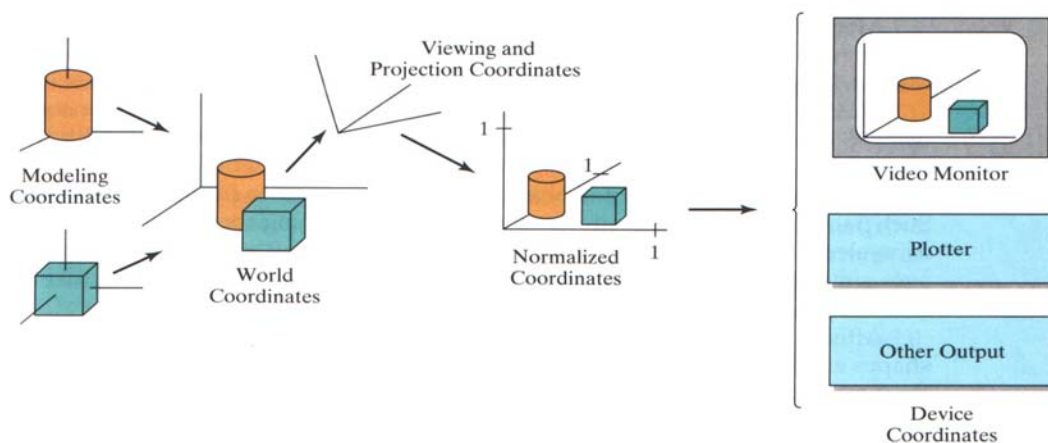
Αφότου όλα τα μέρη της σκηνής έχουν καθοριστεί, γίνεται επεξεργασία της ολικής σκηνής, μέσω διάφορων ρουτινών, σε ένα ή περισσότερα πλαίσια αναφοράς της συσκευής εξόδου προς απεικόνιση. Η διαδικασία αυτή καλείται *οπτική διαδικασία μεταφοράς στο χρήστη* (*viewing pipeline*). Οι θέσεις σε παγκόσμιες συντεταγμένες μετασχηματίζονται πρώτα σε *οπτικές συντεταγμένες* (*viewing coordinates*), που αντιστοιχούν στην επιθυμητή οπτική για τη σκηνή, βασιζόμενες στη θέση και τον προσανατολισμό μίας *υποθετικής κάμερας*. Στη συνέχεια, οι τοποθεσίες των αντικειμένων μετασχηματίζονται σε μία δισδιάστατη προβολή της σκηνής, που αντιστοιχεί στο τι θα απεικονίζεται στη συσκευή εξόδου (οθόνη). Ακολούθως, η σκηνή αποθηκεύεται σε *κανονικοποιημένες συντεταγμένες* (*normalized coordinates*), όπου κάθε τιμή συντεταγμένης κυμαίνεται μεταξύ -1 έως 1 ή 0 έως 1, ανάλογα με το υπολογιστικό σύστημα. Οι κανονικοποιημένες συντεταγμένες αναφέρονται επίσης και ως *κανονικοποιημένες συντεταγμένες συσκευής* (*normalized device coordinates*), αφού χρησιμοποιώντας αυτή την αναπαράσταση, ένα πακέτο γραφικών γίνεται ανεξάρτητο από το σύνολο τιμών συντεταγμένων οποιασδήποτε συσκευής εξόδου. Επίσης, απαιτείται να προσδιοριστούν οι ορατές επιφάνειες και να αφαιρεθούν τα μέρη της εικόνας που βρίσκονται έξω από τα όρια της οπτικής, που επιθυμείται να παρουσιαστεί στη συσκευή απεικόνισης. Τελικώς, η εικόνα σαρώνεται μετασχηματιζόμενη στη προσωρινή αποθήκη στοιχείων ανανέωσης ενός συστήματος αναπαραγωγής ειδώλου, για να απεικονισθεί. Τα συστήματα συντεταγμένων για τις συσκευές απεικόνισης γενικά ονομάζονται *συντεταγμένες συσκευής* (*device coordinates*), ή *συντεταγμένες οθόνης* (*screen coordinates*) στη περίπτωση οθόνης. Συχνά, οι κανονικοποιημένες και οι συντεταγμένες οθόνης καθορίζονται σε ένα αριστερόστροφο σύστημα αναφοράς, με σκοπό όταν θα αυξάνονται θετικά οι αποστάσεις από το επίπεδο-xy, που αναπαριστά την οθόνη ή το οπτικό επίπεδο, να ερμηνεύεται ως απομάκρυνση από τη θέση παρακολούθησης.

Στην **Εικόνα 3.1** παρουσιάζεται συνοπτικά η ακολουθία των μετασχηματισμών των συντεταγμένων, από τις συντεταγμένες μοντέλου στις συντεταγμένες συσκευής, για την απεικόνιση που περιέχει την όψη δύο τρισδιάστατων αντικειμένων. Μία αρχική θέση σε συντεταγμένες μοντέλου (x_{mc}, y_{mc}, z_{mc}) μεταφέρεται σε παγκόσμιες συντεταγμένες, στη συνέχεια σε οπτικές και συντεταγμένες προβολής, έπειτα σε αριστερόστροφες κανονικοποιημένες συντεταγμένες, και τέλος σε μία θέση συντεταγμένων συσκευής (x_{dc}, y_{dc}) με την ακόλουθη σειρά:

$$(x_{mc}, y_{mc}, z_{mc}) \rightarrow (x_{wc}, y_{wc}, z_{wc}) \rightarrow (x_{vc}, y_{vc}, z_{vc}) \rightarrow (x_{pc}, y_{pc}, z_{pc}) \rightarrow (x_{nc}, y_{nc}, z_{nc}) \rightarrow$$

$$(x_{dc}, y_{dc})$$

Οι συντεταγμένες συσκευής (x_{dc}, y_{dc}) είναι ακέραιοι, με τιμές που κυμαίνονται από (0,0) έως (x_{max}, y_{max}) για μία συγκεκριμένη συσκευή εξόδου. Επιπρόσθετα στις δισδιάστατες θέσεις (x_{dc}, y_{dc}) στην επιφάνεια παρακολούθησης, αποθηκεύονται πληροφορίες για το βάθος κάθε θέσης σε συντεταγμένες συσκευής, ώστε να χρησιμοποιηθούν σε ποικίλους αλγόριθμους ορατότητας και επεξεργασίας επιφανειών. [3].



Εικόνα 3.1: Ακολουθία μετασχηματισμών από συντεταγμένες μοντέλου σε συντεταγμένες συσκευής. [3]

3.2 Συστήματα Αναφοράς Συντεταγμένων

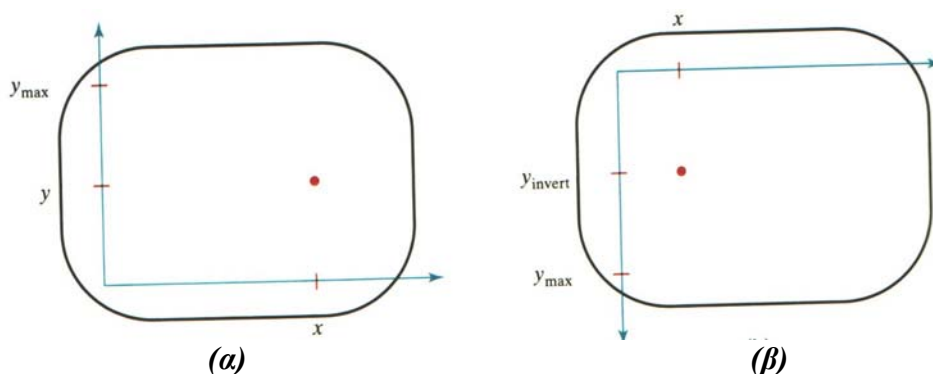
Τα Καρτεσιανά και μη-Καρτεσιανά συστήματα αναφοράς είναι συχνά πολύ χρήσιμα στις εφαρμογές ψηφιακών γραφικών. Τυπικά, οι συντεταγμένες σε ένα πρόγραμμα γραφικών καθορίζονται χρησιμοποιώντας ένα Καρτεσιανό σύστημα αναφοράς, αλλά η αρχική προδιαγραφή της σκηνής μπορεί να δίνεται σε ένα μη-Καρτεσιανό σύστημα αναφοράς. Σφαιρικές, κυλινδρικές, ή άλλες συμμετρίες χρησιμοποιούνται συχνά, για να απλοποιηθούν οι περιγραφές των αντικειμένων ή ο χειρισμός τους.

3.2.1 Δισδιάστατες Καρτεσιανές Συντεταγμένες Οθόνης

Για τις εντολές σε ένα πακέτο γραφικών που είναι ανεξάρτητες συσκευής, οι θέσεις σε συντεταγμένες οθόνης αναφέρονται στο πρώτο τεταρτημόριο ενός δισδιάστατου Καρτεσιανού συστήματος, σε μία πρότυπη θέση, όπως φαίνεται και στην **Εικόνα 3.2(α)**. Η αρχή των συντεταγμένων για αυτό το σύστημα αναφοράς βρίσκεται στη κάτω αριστερή γωνία της οθόνης. Ωστόσο, οι γραμμές σάρωσης της οθόνης αριθμούνται από 0 στην άνω πλευρά της. Έτσι οι τοποθεσίες στην οθόνη αναπαριστώνται εσωτερικά με βάση την άνω αριστερή γωνία της οθόνης. Για το λόγο αυτό, οι εντολές που συναρτώνται με τη συσκευή (όπως αυτές των δεδομένων εισόδου αλληλεπίδρασης και αυτών της διαχείρισης του παραθύρου απεικόνισης), συχνά αναφέρονται στις συντεταγμένες οθόνης, χρησιμοποιώντας το αντιστραμμένο Καρτεσιανό σύστημα αναφοράς που παρουσιάζεται στην **Εικόνα 3.2(β)**. Οι οριζόντιες τιμές των συντεταγμένων και στα δύο συστήματα είναι ίδιες, και μία αντιστραμμένη τιμή y μετασχηματίζεται σε μία τιμή y μετρούμενη από το κατώτατο μέρος της οθόνης, σύμφωνα με τον υπολογισμό:

$$y = y_{max} - y_{invert} \quad (3-1)$$

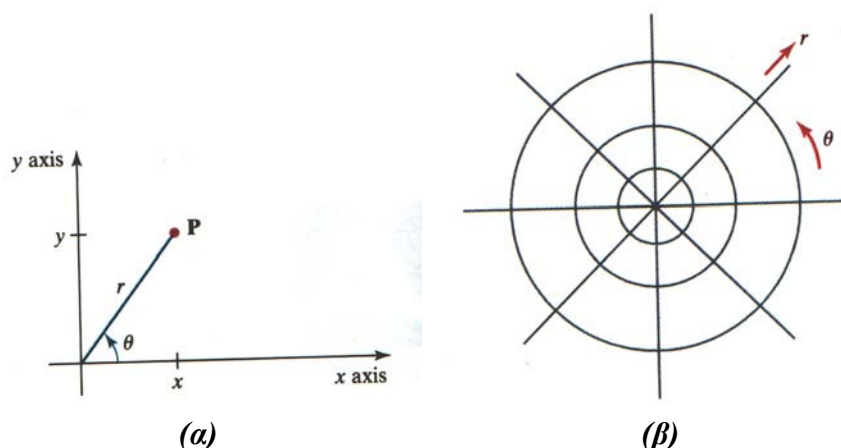
Σε κάποια πακέτα εφαρμογών, η αρχή των συντεταγμένων της οθόνης μπορεί να τοποθετηθεί σε μία αυθαίρετη θέση, όπως για παράδειγμα στο κέντρο της οθόνης.



Εικόνα 3.2: Οι θέσεις σε Καρτεσιανές συντεταγμένες οθόνης αναφέρονται ως προς (α) την κάτω αριστερή γωνία ή (β) την άνω αριστερή γωνία της οθόνης. [3]

3.2.2 Πρότυπα Διδιάστατα Καρτεσιανά Συστήματα Αναφοράς

Τα Καρτεσιανά συστήματα χρησιμοποιούνται σε μία πρότυπη θέση για τον καθορισμό παγκόσμιων συντεταγμένων (και άλλων αναφορών) κατά την δισδιάστατη οπτική διαδικασία μεταφοράς στο χρήστη. Οι συντεταγμένες σε αυτά τα πλαίσια μπορεί να είναι θετικές ή αρνητικές, με οποιοδήποτε σύνολο τιμών. Για την απεικόνιση μίας όψης μίας δισδιάστατης εικόνας, προσδιορίζονται ένα παράθυρο αποκόμματος (*clipping window*) και μία οπτική οδός (*viewport*), για τη χαρτογράφηση ενός τμήματος της εικόνας σε συντεταγμένες οθόνης.

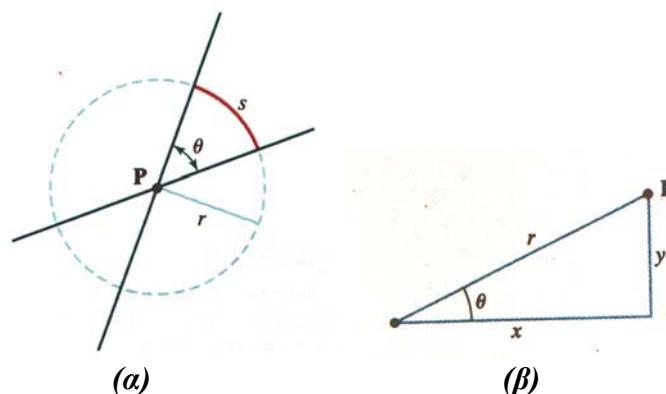


Εικόνα 3.3: (α) Συνάφεια μεταξύ πολικών και Καρτεσιανών συντεταγμένων και (β) Πολικό σύστημα συντεταγμένων [3].

3.2.3 Πολικές Συντεταγμένες στο $\chi\psi$ -Επίπεδο

Ένα συχνά χρησιμοποιούμενο δισδιάστατο μη-Καρτεσιανό σύστημα είναι το πολικό σύστημα συντεταγμένων (**Εικόνα 3.3(β)**), όπου οι συντεταγμένες μίας θέσης ορίζονται από την ακτινική απόσταση r από την αρχή των συντεταγμένων και από μία γωνιακή μετατόπιση θ από την οριζόντιο. Θετικές ακτινικές μετατοπίσεις λαμβάνονται οι αριστερόστροφες και αρνητικές οι δεξιόστροφες. Η σχέση μεταξύ Καρτεσιανών και πολικών συντεταγμένων παρουσιάζεται στην **Εικόνα 3.3(α)**. Θεωρώντας το τρίγωνο της **Εικόνας 3.4(β)**, και χρησιμοποιώντας τους ορισμούς των τριγωνομετρικών συναρτήσεων, πραγματοποιείται ο μετασχηματισμός από πολικές σε Καρτεσιανές συντεταγμένες με τη βοήθεια των σχέσεων:

$$x = r \cos\theta \quad y = r \sin\theta \quad (3-2)$$



Εικόνα 3.4:(α) γωνία θ που ορίζεται από κυκλικό τόξο μήκους s και ακτίνας r και (β) τρίγωνο με υποτείνουσα r , πλευρές x και y , και εσωτερική γωνία θ [3].

Ο αντίστροφος μετασχηματισμός, από Καρτεσιανές σε πολικές συντεταγμένες δίδεται:

$$r = \sqrt{x^2 + y^2} \quad \theta = \tan^{-1}\left(\frac{y}{x}\right) \quad (3-3)$$

Οι τιμές της γωνίας μπορούν να μετρηθούν είτε σε μοίρες είτε σε αδιάστατες μονάδες, όπως τα ακτίνια. Το ένα ακτίνιο ορίζεται ως η τιμή της γωνίας που αντιστοιχεί σε ένα κυκλικό τόξο του οποίου το μήκος είναι ίσο προς την ακτίνα του κύκλου. Ο ορισμός αυτός παρουσιάζεται στην **Εικόνα 3.4(α)**, η οποία δείχνει δύο τεμνόμενες γραμμές σε ένα επίπεδο και ένα κύκλο με κέντρο το σημείο τομής P. Για οποιοδήποτε κύκλο με κέντρο το P, η τιμή της γωνίας θ σε ακτίνια δίνεται από το λόγο:

$$\theta = \frac{s}{r} \quad (3-4)$$

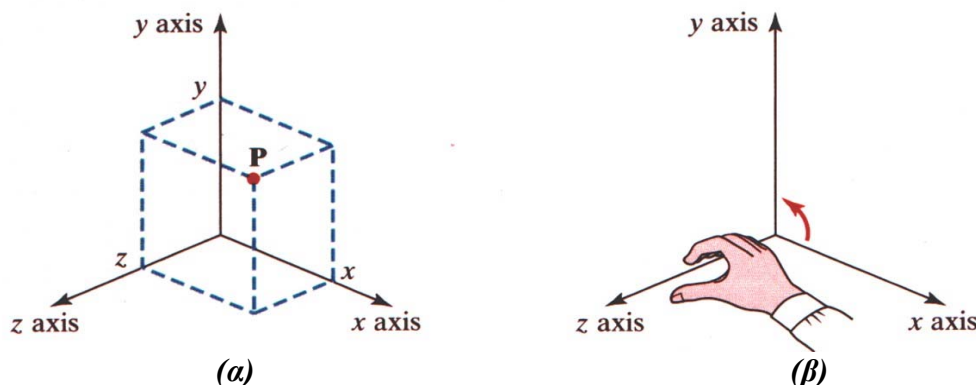
όπου s είναι το μήκος του τόξου που αντιστοιχεί στη γωνία θ , και r η ακτίνα του κύκλου. Η ολική γωνιακή απόσταση γύρω από το σημείο P είναι το μήκος της περιμέτρου του κύκλου ($2\pi r$) δια της ακτίνας r , δηλαδή 2π ακτίνια. Σε μοίρες, η περιφέρεια του κύκλου διαιρείται σε 360 τόξα ίσου μήκους, έτσι ώστε κάθε τόξο αντιστοιχεί σε γωνία 1 μοίρας, δηλαδή $360^\circ = 2\pi$ ακτίνια.

Άλλες κωνικές τομές, εκτός του κύκλου, μπορούν να χρησιμοποιηθούν για τον καθορισμό συντεταγμένων θέσεων. Ως παράδειγμα, χρησιμοποιώντας ομοαξονικές ελλείψεις αντί κύκλων, μπορούν να δοθούν συντεταγμένες θέσεων σε ελλειπτικές συντεταγμένες. Όμοια, άλλοι τύποι συμμετριών μπορούν να χρησιμοποιηθούν για τον καθορισμό υπερβολικών ή παραβολικών συντεταγμένων.

3.2.4 Πρότυπα Τρισδιάστατα Καρτεσιανά Συστήματα Αναφοράς

Στην **Εικόνα 3.5(α)** εμφανίζεται ο συμβατικός προσανατολισμός των αξόνων σε ένα τρισδιάστατο Καρτεσιανό σύστημα αναφοράς. Το σύστημα αυτό ονομάζεται δεξιόστροφο λόγω του ότι ο δεξιός αντίχειρας δείχνει στη θετική διεύθυνση του z-άξονα εάν υποθεθεί ότι τα δάχτυλα του δεξιού χεριού αγκαλιάζουν τον άξονα από την θετική διεύθυνση του x-άξονα προς τη θετική διεύθυνση του y-άξονα κατά 90° , όπως

παρουσιάζεται και στην **Εικόνα 3.5(β)**. Στα περισσότερα προγράμματα ψηφιακών γραφικών, οι περιγραφές των αντικειμένων, καθώς και άλλες παράμετροι συντεταγμένων, καθορίζονται σε δεξιόστροφες Καρτεσιανές συντεταγμένες. Στην παρούσα εργασία θεωρείται ότι όλα τα Καρτεσιανά συστήματα αναφοράς είναι δεξιόστροφα εκτός αν ορίζεται διαφορετικά.

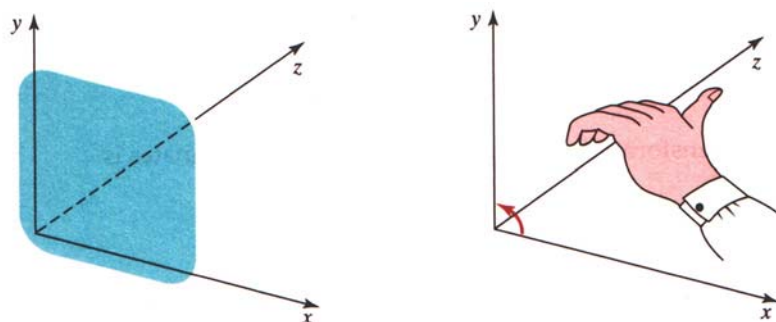


Εικόνα 3.5: Αναπαράσταση συντεταγμένων, σημείου $P(x, y, z)$ σε πρότυπο δεξιόστροφο Καρτεσιανό σύστημα αναφοράς [3].

Τα Καρτεσιανά συστήματα αναφοράς είναι ορθογώνια συστήματα, δηλαδή οι άξονες συντεταγμένων είναι κάθετοι μεταξύ τους. Επιπλέον, στα Καρτεσιανά συστήματα, οι άξονες είναι ευθείες γραμμές. Αλλά συστήματα συντεταγμένων με καμπύλους άξονες είναι χρήσιμα σε πολλές εφαρμογές. Τα περισσότερα από αυτά τα συστήματα είναι επίσης ορθογώνια με την έννοια ότι οι αξονικές κατευθύνσεις σε οποιοδήποτε σημείο του χώρου είναι κάθετες μεταξύ τους.

3.2.5 Τρισδιάστατες Καρτεσιανές Συντεταγμένες Οθόνης

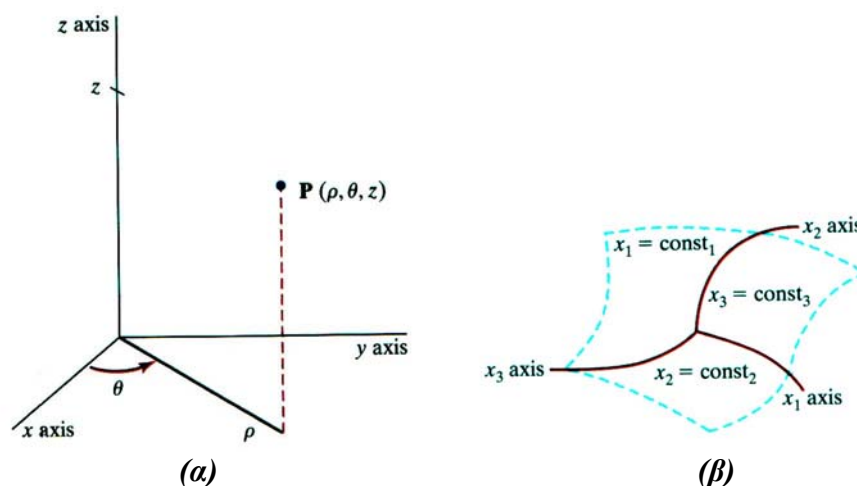
Όταν η όψη μίας τρισδιάστατης σκηνής απεικονίζεται στην οθόνη, αποθηκεύονται πληροφορίες για το βάθος καθενός σημείου. Η τρισδιάστατη θέση, που αντιστοιχεί σε κάθε σημείο της οθόνης, συχνά αναφέρεται σε αριστερόστροφο σύστημα αναφοράς, όπως φαίνεται στην **Εικόνα 3.6**. Στην περίπτωση αυτή, ο αριστερός αντίχειρας δείχνει στη θετική διεύθυνση του z-άξονα εάν υποθεθεί ότι τα δάχτυλα του αριστερού χεριού αγκαλιάζουν τον άξονα από την θετική διεύθυνση του x-άξονα προς τη θετική διεύθυνση του y-άξονα κατά 90° . Οι θετικές τιμές στον z-άξονα αντιστοιχούν σε θέσεις πίσω από την οθόνη για κάθε σημείο στο xy-επίπεδο, και μεγαλύτερες τιμές κατά μήκος της θετικής διεύθυνσης του z-άξονα ερμηνεύονται ως θέσεις που απομακρύνονται από τον χρήστη.



Εικόνα 3.6: Αριστερόστροφο Καρτεσιανό σύστημα συντεταγμένων τοποθετημένο πάνω σε οθόνη [3].

3.2.6 Τρισδιάστατα Καμπυλόγραμμα Συστήματα Συντεταγμένων

Κάθε μη-Καρτεσιανό σύστημα αναφοράς αναφέρεται ως καμπυλόγραμμα σύστημα συντεταγμένων. Η επιλογή του συστήματος συντεταγμένων για μία συγκεκριμένη εφαρμογή γραφικών εξαρτάται από ένα πλήθος παραγόντων, όπως είναι η συμμετρία, η ευκολία υπολογισμών, και τα πλεονεκτήματα της οπτικοποίησης. Η **Εικόνα 3.7(β)** παρουσιάζει ένα γενικό καμπυλόγραμμα σύστημα αναφοράς, που μορφοποιείται από τρεις επιφάνειες συντεταγμένων. Άξονες συντεταγμένων σε κάθε πλαίσιο αναφοράς είναι οι τεμνόμενες καμπύλες των επιφανειών συντεταγμένων. Εάν οι επιφάνειες συντεταγμένων τέμνονται παντού με δεξιόστροφες γωνίες, ορίζεται ένα ορθογώνιο καμπυλόγραμμα σύστημα συντεταγμένων. Μη-ορθογώνια, καμπυλόγραμμα συστήματα αναφοράς είναι χρήσιμα σε κάποιες εφαρμογές, όπως οπτικοποιήσεις κινήσεων που διέπονται από τους νόμους της γενικής σχετικότητας, αλλά χρησιμοποιούνται λιγότερο συχνά στα ψηφιακά γραφικά από τα ορθογώνια συστήματα.



Εικόνα 3.7: (α) Κυλινδρικές συντεταγμένες ρ , θ , z και (β) Γενικό καμπυλόγραμμα σύστημα αναφοράς. [3].

Ο καθορισμός των κυλινδρικών συντεταγμένων ενός σημείου στο χώρο εμφανίζεται στην **Εικόνα 3.7(α)** σε σχέση με ένα Καρτεσιανό σύστημα αναφοράς. Η επιφάνεια σταθερής ρ είναι ένας κατακόρυφος κύλινδρος, η επιφάνεια σταθερής θ είναι ένα κατακόρυφο επίπεδο που εμπεριέχει τον z -άξονα, και η επιφάνεια σταθερού z είναι ένα οριζόντιο επίπεδο παράλληλο στο Καρτεσιανό xy -επίπεδο. Ο μετασχηματισμός από τον ορισμό σε κυλινδρικές συντεταγμένες σε ένα Καρτεσιανό σύστημα αναφοράς πραγματοποιείται με τους εξής υπολογισμούς:

$$x = \rho \cos \theta \quad y = \rho \sin \theta \quad z = z \quad (3-5)$$

Ένας άλλος, συχνά χρησιμοποιούμενος καμπυλόγραμμα ορισμός συντεταγμένων είναι το σφαιρικό σύστημα συντεταγμένων της **Εικόνας 3.8(β)**. Οι σφαιρικές συντεταγμένες αναφέρονται κάποιες φορές ως πολικές συντεταγμένες στον τρισδιάστατο χώρο. Η επιφάνεια σταθερής r είναι σφαίρα, η επιφάνεια σταθερής θ είναι και πάλι ένα κατακόρυφο επίπεδο που περιλαμβάνει τον z -άξονα, και η επιφάνεια σταθερής ϕ είναι ένας κώνος με κορυφή την αρχή των συντεταγμένων. Εάν $\phi < 90^\circ$, ο κώνος είναι άνω του xy -επιπέδου. Εάν $\phi > 90^\circ$, ο κώνος είναι κάτω του xy -

επιπέδου. Ο μετασχηματισμός από τον ορισμό σε σφαιρικές συντεταγμένες σε ένα Καρτεσιανό σύστημα αναφοράς πραγματοποιείται με τους εξής υπολογισμούς:

$$x = \rho \cos\theta \sin\varphi \quad y = r \sin\theta \sin\varphi \quad z = r \cos\varphi \quad (3-6)$$

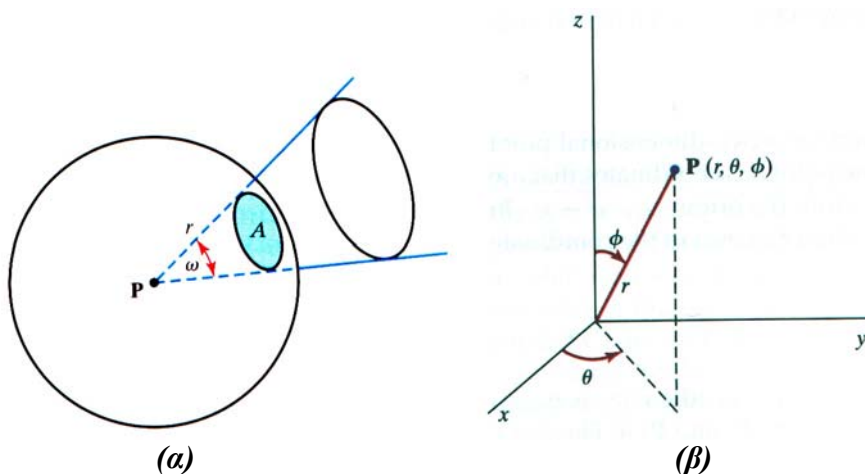
3.2.7 Συμπαγής Γωνία (Solid Angle)

Ο ορισμός για μία συμπαγή γωνία ω διατυπώνεται σε αναλογία με τον ορισμό για μία δισδιάστατη, μετρημένη σε ακτίνια, γωνία θ μεταξύ δύο τεμνόμενων γραμμών (3-4). Για μία τρισδιάστατη γωνία, ωστόσο, θεωρείται ένας κώνος με κορυφή ένα σημείο P και μία σφαίρα με κέντρο το P, όπως φαίνεται και στην **Εικόνα 3.8(α)**. Η συμπαγής γωνία ω μέσα στη, σχήματος κώνου, περιοχή με κορυφή στο P ορίζεται ως:

$$\omega = \frac{A}{r^2} \quad (3-7)$$

όπου A η περιοχή της σφαιρικής επιφάνειας που τέμνεται από τον κώνο και r η ακτίνα της σφαίρας.

Επίσης, σε αναλογία με τις δισδιάστατες πολικές συντεταγμένες, η αδιάστατη μονάδα μέτρησης για συμπαγείς γωνίες ονομάζεται *στερακτίνιο (steradian)*. Η ολική συμπαγής γωνία περί του σημείου P είναι η ολική περιοχή της σφαιρικής επιφάνειας ($4\pi r^2$) δια του r^2 , ή 4π στερακτίνια [3].



Εικόνα 3.8:(α)Συμπαγής γωνία ω ορισμένη από σφαιρικό επιφανειακό κομμάτι με επιφάνεια A και ακτίνα r, και (β)Σφαιρικές συντεταγμένες r, θ, και φ [3].

ΓΕΩΜΕΤΡΙΚΟΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ

4.1 Γενικά

Στην ενότητα αυτή θα παρουσιαστούν οι λειτουργίες μετασχηματισμών, που εφαρμόζονται στα αντικείμενα, για την επανατοποθέτηση και την αλλαγή του σχήματός τους. Οι λειτουργίες αυτές χρησιμοποιούνται επίσης στην ρουτίνες οπτικής, για την μετατροπή των παγκόσμιων συντεταγμένων της σκηνής σε συντεταγμένες κατάλληλες για την απεικόνιση στη συσκευή εξόδου. Επιπρόσθετα, χρησιμοποιούνται σε μία ποικιλία άλλων εφαρμογών, όπως στη σχεδιομελέτη με χρήση ηλεκτρονικών υπολογιστών (CAD) και στη ψηφιακή δυναμική κίνηση (computer animation). Οι λειτουργίες που εφαρμόζονται στη γεωμετρική περιγραφή ενός αντικειμένου για την αλλαγή της θέσης, του προσανατολισμού, ή του μεγέθους του ονομάζονται *γεωμετρικοί μετασχηματισμοί*.

Κάποιες φορές οι λειτουργίες γεωμετρικών μετασχηματισμών αναφέρονται επίσης και ως *μετασχηματισμοί μοντέλου (modeling transformations)*, αλλά κάποια πακέτα γραφικών διαχωρίζουν αυτούς τους δύο όρους. Γενικά, οι μετασχηματισμοί μοντέλου χρησιμοποιούνται για την κατασκευή της σκηνής ή για να δώσουν την ιεραρχική περιγραφή ενός πολύπλοκου αντικειμένου, που συντίθεται από πολλά μέρη, τα οποία με τη σειρά τους συντίθενται από απλούστερα. Ως παράδειγμα, ένα αεροσκάφος αποτελείται από πτέρυγες, ουραίο, σκελετό, κινητήρες, και άλλα μέρη, καθένα από τα οποία μπορεί να αναλυθεί σε κατώτερα ιεραρχικά επίπεδα. Έτσι, το αεροσκάφος μπορεί να περιγραφεί με ορισμούς των μερών που το συνθέτουν και τους αντίστοιχους μετασχηματισμούς μοντέλου για το καθένα επιμέρους τμήμα. Αυτοί περιγράφουν πώς το κάθε τμήμα συνδυάζεται στο ολικό σχέδιο του αεροσκάφους. Οι γεωμετρικοί μετασχηματισμοί, από την άλλη μεριά, μπορούν να χρησιμοποιηθούν για να περιγράψουν πώς τα αντικείμενα μπορεί να μετακινούνται στη σκηνή, κατά τη διάρκεια μίας διαδοχής κινούμενων εικόνων ή απλά για την θέασή τους από μία άλλη γωνία. Για το λόγο αυτό, κάποια πακέτα γραφικών παρέχουν δύο σύνολα ρουτινών μετασχηματισμών, ενώ άλλα πακέτα έχουν ένα μόνο σύνολο συναρτήσεων που μπορεί να χρησιμοποιηθεί για μετασχηματισμούς μοντέλου αλλά και γεωμετρικούς μετασχηματισμούς [3].

4.2 Βασικοί Δισδιάστατοι Γεωμετρικοί Μετασχηματισμοί

Οι συναρτήσεις γεωμετρικών μετασχηματισμών, που είναι διαθέσιμες σε όλα τα πακέτα γραφικών, είναι εκείνες για *μεταφορά (translation)*, *περιστροφή (rotation)* και *αλλαγή μεγέθους (scaling)*. Άλλες χρήσιμες ρουτίνες μετασχηματισμών, που μερικές φορές περιλαμβάνονται σε πακέτα γραφικών, είναι ο *συμμετρικός μετασχηματισμός (reflection)* και οι λειτουργίες *αποκοπής (shearing)*. Για την παρουσίαση των γενικών αρχών, που συνδέονται με τους γεωμετρικούς μετασχηματισμούς, αρχικά θεωρούνται οι λειτουργίες σε δύο διαστάσεις, και έπειτα αναλύεται πώς αυτές οι ιδέες επεκτείνονται στις τρισδιάστατες σκηνές.

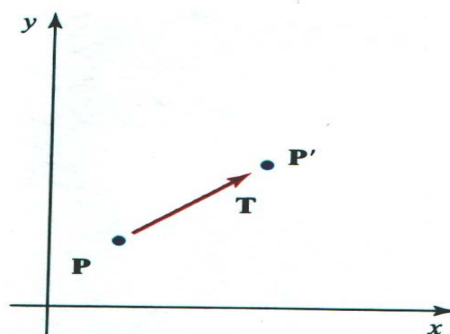
4.2.1 Δισδιάστατη Μεταφορά

Η μεταφορά (translation) ενός μόνο σημείου συντεταγμένων εκπληρώνεται με την προσθήκη των μετατοπίσεων στις συντεταγμένες του, έτσι ώστε να δημιουργηθεί μία νέα θέση συντεταγμένων. Στην ουσία, μετακινείται η αρχική θέση του σημείου πάνω σε μία ευθύγραμμη τροχιά, στη νέα τοποθεσία. Παρόμοια, η μεταφορά εφαρμόζεται σε ένα αντικείμενο που ορίζεται με πολλαπλές θέσεις συντεταγμένων, όπως ένα τετράπλευρο, με την επανατοποθέτηση όλων των θέσεων συντεταγμένων κατά μήκος παράλληλων τροχιών. Έπειτα το ολοκληρωμένο αντικείμενο απεικονίζεται στη νέα θέση.

Για τη μεταφορά μίας δισδιάστατης θέσης, προστίθενται οι αποστάσεις μεταφοράς t_x και t_y στις αρχικές συντεταγμένες (x, y) για την απόκτηση της νέας θέσης συντεταγμένων (x', y') , όπως φαίνεται και στην **Εικόνα 4.1**.

$$x' = x + t_x \quad y' = y + t_y \quad (4-1)$$

Το ζεύγος (t_x, t_y) της απόστασης μεταφοράς ονομάζεται *διάνυσμα μεταφοράς* (translation vector) ή *διάνυσμα μετατόπισης* (shift vector).



Εικόνα 4.1: Μεταφορά σημείου από μία θέση P σε μία θέση P' χρησιμοποιώντας το διάνυσμα μεταφοράς T [3].

Οι εξισώσεις (4-1) μπορούν να εκφραστούν σαν μία μόνο εξίσωση πινάκων, χρησιμοποιώντας τα ακόλουθα διανύσματα στήλης για την αναπαράσταση των θέσεων και του διανύσματος μεταφοράς:

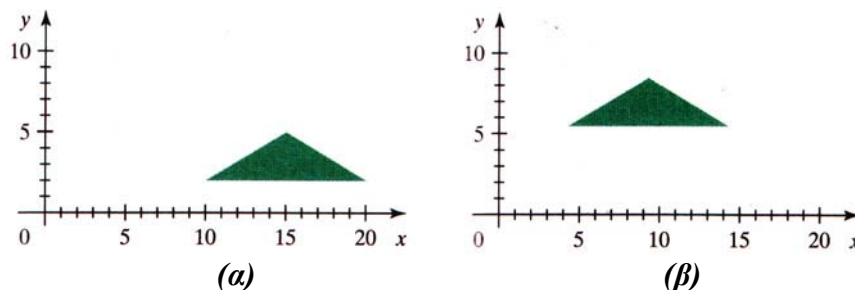
$$P = \begin{bmatrix} x \\ y \end{bmatrix} \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (4-2)$$

Αυτό επιτρέπει τη γραφή των δισδιάστατων εξισώσεων μεταφοράς σε μορφή πινάκων ως:

$$P' = P + T \quad (4-3)$$

Η μεταφορά είναι ένας μετασχηματισμός που μετακινεί τα αντικείμενα χωρίς παραμόρφωση. Δηλαδή, κάθε σημείο του αντικειμένου μεταφέρεται κατά την ίδια ποσότητα. Η μεταφορά ενός ευθύγραμμου τμήματος επιτυγχάνεται, εφαρμόζοντας την εξίσωση μετασχηματισμού (4-3) σε καθένα από τα δύο ακραία σημεία του και ξανασχεδιάζοντας το ευθύγραμμο τμήμα μεταξύ των νέων θέσεων των ακραίων σημείων. Η μεταφορά ενός πολυγώνου πραγματοποιείται με παρόμοιο τρόπο,

προσθέτοντας ένα διάνυσμα μεταφοράς στη θέση συντεταγμένων κάθε κορυφής και έπειτα αναπαράγοντας το πολύγωνο, χρησιμοποιώντας το νέο σύνολο των συντεταγμένων των κορυφών. Στην **Εικόνα 4.2** παρουσιάζεται η εφαρμογή ενός συγκεκριμένου διανύσματος μεταφοράς για την μετακίνηση ενός αντικειμένου από μία θέση σε άλλη.

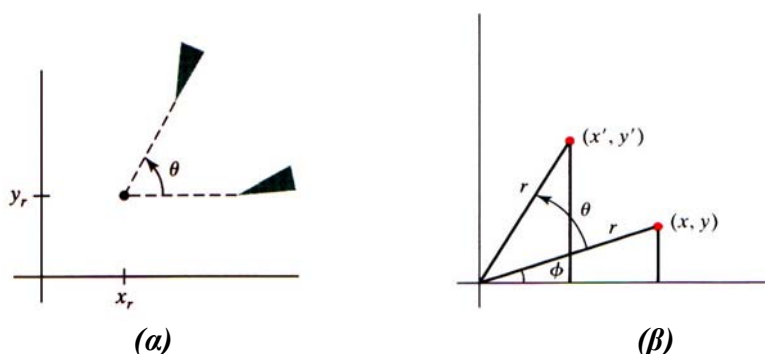


Εικόνα 4.2: Μεταφορά πολύγωνου από μία θέση (α) σε μία θέση (β) [3].

Παρόμοιες μέθοδοι χρησιμοποιούνται για τη μεταφορά άλλων αντικειμένων. Για την αλλαγή της θέσης ενός κύκλου ή μίας έλλειψης, μεταφέρονται οι συντεταγμένες του κέντρου και ξανασχεδιάζεται το σχήμα στη νέα τοποθεσία. Για μία καμπύλη spline, μεταφέρονται τα σημεία που ορίζουν τη τροχιά της καμπύλης και έπειτα ανακατασκευάζονται τα τμήματα της καμπύλης, μεταξύ των νέων θέσεων συντεταγμένων [3].

4.2.2 Δισδιάστατη Περιστροφή

Η περιστροφή (rotation) ενός αντικειμένου πραγματοποιείται με τον καθορισμό του άξονα περιστροφής (rotation axis) και της γωνίας περιστροφής (rotation angle). Στη συνέχεια, όλα τα σημεία του αντικειμένου μετασχηματίζονται στις νέες θέσεις με την περιστροφή τους υπό την καθορισμένη γωνία περί του άξονα περιστροφής.



Εικόνα 4.3: (α) Περιστροφή αντικειμένου κατά γωνία θ γύρω από σημείο περιστροφής (x_r, y_r) και (β) Περιστροφή σημείου από τη θέση (x, y) στη θέση (x', y') κατά γωνία θ ως προς την αρχή των συντεταγμένων. Η αρχική γωνιακή μετατόπιση του σημείου από το x-άξονα είναι ϕ [3].

Η δισδιάστατη περιστροφή ενός αντικειμένου εξασφαλίζεται με τον επαναπροσδιορισμό της θέσης του κατά μήκος μίας κυκλικής τροχιάς στο xy-επίπεδο. Στην περίπτωση αυτή, το αντικείμενο περιστρέφεται γύρω από έναν άξονα κάθετο στο xy-επίπεδο, ή διαφορετικά, παράλληλο στον z-άξονα των συντεταγμένων. Οι

παράμετροι για τη δισδιάστατη περιστροφή είναι η *γωνία περιστροφής* θ και η θέση (x_r, y_r) , που ονομάζεται *σημείο περιστροφής* (*rotation ή pivot point*), περί του οποίου το αντικείμενο θα περιστραφεί, όπως φαίνεται και στην **Εικόνα 4.3(α)**. Το σημείο περιστροφής είναι η θέση τομής του άξονα περιστροφής με το xy-επίπεδο. Θετική τιμή για τη γωνία θ ορίζει αριστερόστροφη περιστροφή περί του σημείου, ενώ η αρνητική τιμή ορίζει δεξιόστροφη.

Για την απλοποίηση της εξήγησης της βασικής μεθόδου, κατά πρώτον ορίζονται οι εξισώσεις μετασχηματισμού για την περιστροφή ενός σημείου P όταν το σημείο περιστροφής βρίσκεται στην αρχή των συντεταγμένων. Οι σχέσεις μεταξύ του αρχικού και του μετασχηματισμένου σημείου παρουσιάζονται στην **Εικόνα 4.3(β)**, όπου r είναι η σταθερή απόσταση του σημείου P από την αρχή των συντεταγμένων, η γωνία φ είναι η αρχική γωνιακή θέση του από την οριζόντιο, ενώ θ είναι η γωνία περιστροφής. Χρησιμοποιώντας τις γνωστές τριγωνομετρικές σχέσεις, οι μετασχηματισμένες συντεταγμένες εκφράζονται με παραμέτρους τις γωνίες θ και φ :

$$\begin{aligned}x' &= r \cos(\varphi + \theta) = r \cos\varphi \cos\theta - r \sin\varphi \sin\theta \\y' &= r \sin(\varphi + \theta) = r \cos\varphi \sin\theta + r \sin\varphi \cos\theta\end{aligned}\quad (4-4)$$

Οι αρχικές συντεταγμένες του σημείου P σε μορφή πολικών συντεταγμένων είναι:

$$x = r \cos\varphi \quad y = r \sin\varphi \quad (4-5)$$

Χρησιμοποιώντας τις σχέσεις (4-5) στις εξισώσεις (4-4), προκύπτουν οι εξισώσεις μετασχηματισμού για την περιστροφή ενός σημείου (x, y) κατά γωνία θ περί την αρχή των συντεταγμένων:

$$\begin{aligned}x' &= x \cos\theta - y \sin\theta \\y' &= x \sin\theta + y \cos\theta\end{aligned}\quad (4-6)$$

Με τη βοήθεια των αναπαράστασεων (4-2) (διανύσματα-στήλες) για τις θέσεις συντεταγμένων, οι εξισώσεις περιστροφής γράφονται σε μορφή πινάκων ως:

$$P' = R \cdot P \quad (4-7)$$

όπου ο πίνακας περιστροφής R δίδεται:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (4-8)$$

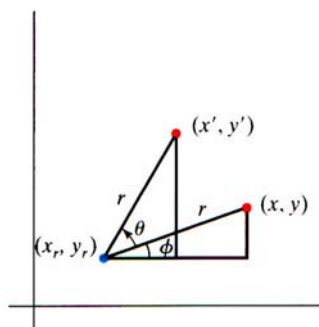
Η αναπαράσταση με διάνυσμα-στήλη για τις συντεταγμένες θέσης του σημείου P, όπως στις εξισώσεις (4-2), αποτελεί συνήθη πρακτική και είναι ορθή από μαθηματικής πλευράς. Ωστόσο, τα αρχικά συστήματα γραφικών μερικές φορές χρησιμοποιούσαν την αναπαράσταση διανύσματος-γραμμής. Κάτι τέτοιο αλλάζει τη σειρά με την οποία θα πραγματοποιηθεί ο πολλαπλασιασμός των πινάκων, για να εκτελεστεί η περιστροφή. Πλέον, τα πακέτα γραφικών όπως OpenGL, Java, PHIGS, και GKS ακολουθούν την σύμβαση διανύσματος-στήλης [3].

Η περιστροφή ενός σημείου γύρω από μία αυθαίρετη θέση περιστροφής περιγράφεται στην **Εικόνα 4.4**. Οι εξισώσεις (4-6) γενικεύονται στη μορφή των

εξισώσεων μετασχηματισμού για την περιστροφή ενός σημείου περί οποιουδήποτε καθορισμένου σημείου περιστροφής (x_r, y_r) ως:

$$\begin{aligned}x' &= x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta \\y' &= y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta\end{aligned}\quad (4-9)$$

Η σχέση (4-7) μπορεί να τροποποιηθεί για να περιλαμβάνει τις συντεταγμένες περιστροφής με τη προσθήκη στον πίνακα ενός διανύσματος στήλης, του οποίου τα στοιχεία περιέχουν του επιπρόσθετους όρους (μεταφοράς) των εξισώσεων (4-9). Ωστόσο, υπάρχουν καλύτεροι τρόποι για τη διατύπωση τέτοιων εξισώσεων πινάκων, και στην **Ενότητα 4.3** θα παρουσιαστεί μία συνεπέστερη διατύπωση για την αναπαράσταση την εξισώσεων μετασχηματισμών.



Εικόνα 4.4:Περιστροφή σημείου από τη θέση (x, y) στη θέση (x', y') κατά γωνία θ γύρω από σημείο περιστροφής (x_r, y_r) [3].

Όπως η μεταφορά, έτσι και η περιστροφή είναι ένας μετασχηματισμός άκαμπτου σώματος, που μετακινεί τα αντικείμενα χωρίς να τα παραμορφώνει. Κάθε σημείο του αντικειμένου περιστρέφεται υπό την ίδια γωνία. Ένα ευθύγραμμο τμήμα περιστρέφεται με την εφαρμογή των εξισώσεων (4-9) σε καθένα από τα δύο ακραία σημεία του και επανασχεδιάζοντάς το, μεταξύ των νέων θέσεων τους. Ένα πολύγωνο περιστρέφεται με την μετακίνηση κάθε κορυφής, χρησιμοποιώντας την καθορισμένη γωνία περιστροφής, και δημιουργώντας το πάλι, χρησιμοποιώντας τις νέες κορυφές. Η περιστροφή μίας καμπύλης πραγματοποιείται με την επανατοποθέτηση των σημείων ορισμού της και τον επανασχεδιασμό της [20].

4.2.3 Δισδιάστατη Αλλαγή Κλίμακας

Για τη μεταβολή του μεγέθους ενός αντικειμένου, εφαρμόζεται ο μετασχηματισμός αλλαγής κλίμακας (scaling). Μία απλή δισδιάστατη λειτουργία αλλαγής κλίμακας εκτελείται με τον πολλαπλασιασμό των συντεταγμένων του αντικειμένου (x, y) με τους παράγοντες αλλαγής κλίμακας s_x και s_y για την παραγωγή των μετασχηματισμένων συντεταγμένων (x', y') :

$$x' = x \cdot s_x \quad y' = y \cdot s_y \quad (4-10)$$

Ο παράγοντας αλλαγής κλίμακας s_x αλλάζει το αντικείμενο στη x-διεύθυνση, ενώ ο s_y το αλλάζει στην y-διεύθυνση. Οι βασικές δισδιάστατες εξισώσεις αλλαγής κλίμακας (4-10) μπορούν να γραφούν και σε μορφή πίνακα ως εξής:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (4-11)$$

ή

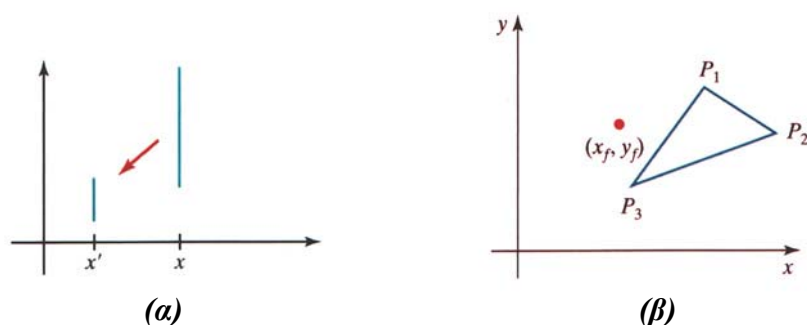
$$P' = S \cdot P \quad (4-12)$$

όπου S είναι ο 2×2 πίνακας αλλαγής κλίμακας της εξίσωσης (4-11).



Εικόνα 4.5: Δημιουργία ενός ορθογώνιου παραλληλογράμμου(β) από ένα τετράγωνο(α), χρησιμοποιώντας παράγοντες αλλαγής κλίμακας $s_x=2$ και $s_y=1$ [3].

Οποιαδήποτε θετική τιμή μπορεί να ανατεθεί στους παράγοντες αλλαγής κλίμακας s_x και s_y . Τιμές μικρότερες της μονάδας μειώνουν το μέγεθος των αντικειμένων, ενώ τιμές μεγαλύτερες προκαλούν μεγέθυνση. Καθορίζοντας την τιμή και των δύο παραγόντων ίση με τη μονάδα, το μέγεθος των αντικειμένων παραμένει ως έχει. Όταν στους παράγοντες s_x και s_y ανατεθεί η ίδια τιμή, παράγεται μία *ομοιόμορφη αλλαγή κλίμακας (uniform scaling)*, η οποία διατηρεί τη σχετική αναλογία του αντικειμένου. Άνισες τιμές για τους δύο παράγοντες έχουν ως αποτέλεσμα μία *διαφορική αλλαγή κλίμακας (differential scaling)*, η οποία χρησιμοποιείται συχνά σε εφαρμογές σχεδιασμού, όπου εικόνες κατασκευάζονται από ελάχιστα βασικά σχήματα, που μπορούν να προσαρμοστούν με μετασχηματισμούς αλλαγής κλίμακας και θέσης (**Εικόνα 4.5**). Σε κάποια συστήματα, μπορούν να καθοριστούν και αρνητικές τιμές για τις παραμέτρους αλλαγής κλίμακας, κάτι που όχι μόνο αλλάζει το μέγεθος του αντικειμένου, αλλά το καθρεπτίζει σε έναν ή περισσότερους άξονες συντεταγμένων.



Εικόνα 4.6:(α) Μία γραμμή που αλλάζει κλίμακα με την εξίσωση (4-12), χρησιμοποιώντας $s_x=s_y=0.5$, μικραίνει και πλησιάζει στην αρχή των αξόνων (β) Αλλαγή κλίμακας ως προς ένα επιλεγμένο σταθερό σημείο (x_f, y_f) [3].

Τα αντικείμενα, που μετασχηματίζονται με την εξίσωση (4-11), προκύπτουν συγχρόνως με αλλαγμένη και κλίμακα και θέση. Παράγοντες αλλαγής κλίμακας με απόλυτες τιμές μικρότερες της μονάδας μετακινούν τα αντικείμενα κοντύτερα στην αρχή των συντεταγμένων, ενώ απόλυτες τιμές μεγαλύτερες της μονάδας μετακινούν τις θέσεις συντεταγμένων μακρύτερα από την αρχή. Στην **Εικόνα 4.6(a)** φαίνεται η

αλλαγή στη κλίμακα ενός ευθύγραμμου τμήματος, αναθέτοντας την τιμή 0.5 στους παράγοντες s_x και s_y στην εξίσωση (4-11). Το μήκος, καθώς και η απόσταση του ευθύγραμμου τμήματος από την αρχή των συντεταγμένων, έχουν μειωθεί κατά τον παράγοντα $\frac{1}{2}$ [3].

Η τοποθέτηση ενός αντικειμένου, που έχει υποστεί αλλαγή κλίμακας, μπορεί να προσδιορισθεί, επιλέγοντας μία θέση, η οποία καλείται *σταθερό σημείο* (*fixed point*). Αυτό παραμένει αμετάβλητο έπειτα από το μετασχηματισμό αλλαγής κλίμακας. Οι συντεταγμένες του σταθερού σημείου (x_f , y_f) συνήθως επιλέγονται σε κάποια θέση του αντικειμένου, όπως για παράδειγμα το κέντρο βάρους του (*centroid*), αλλά γενικά μπορεί να επιλεγεί οποιαδήποτε θέση στο χώρο. Τα αντικείμενα στην περίπτωση αυτή αλλάζουν μέγεθος με αλλαγή στη κλίμακα των αποστάσεων μεταξύ των σημείων του αντικειμένου και του σταθερού σημείου, όπως παρουσιάζεται και στην **Εικόνα 4.6(β)**. Για συντεταγμένες (x , y), οι νέες συντεταγμένες (x' , y') αλλαγμένης κλίμακας υπολογίζονται από τις ακόλουθες σχέσεις:

$$x' - x_f = (x - x_f)s_x \quad y' - y_f = (y - y_f)s_y \quad (4-13)$$

Οι εξισώσεις (4-13) μπορούν να ξαναγραφούν ως αθροίσματα γινομένων:

$$x' = x \cdot s_x + x_f(1 - s_x) \quad y' = y \cdot s_y + y_f(1 - s_y) \quad (4-14)$$

όπου οι προσθετέοι $x_f(1 - s_x)$ και $y_f(1 - s_y)$ είναι σταθεροί για όλα τα σημεία του αντικειμένου.

Το να συμπεριληφθούν οι συντεταγμένες του σταθερού σημείου στις εξισώσεις αλλαγής κλίμακας είναι παρόμοιο με το να συμπεριληφθούν οι συντεταγμένες του σημείου περιστροφής στις ανάλογες εξισώσεις, όπως παρουσιάστηκε στην **Παράγραφο 4.2.2**. Μπορεί να σχηματιστεί ένα διάνυσμα-στήλη του οποίου τα στοιχεία είναι οι σταθεροί όροι στις εξισώσεις (4-14), και έπειτα να προστεθεί αυτό στο γινόμενο $S \cdot P$ της εξίσωσης (4-12). Στην **Ενότητα 4.3**, θα αναπτυχθεί μία διατύπωση για τις εξισώσεις μετασχηματισμών, που περιλαμβάνει μόνο πολλαπλασιασμούς πινάκων.

Η αλλαγή κλίμακας σε πολύγωνα επιτυγχάνεται με την εφαρμογή των μετασχηματισμών (4-14) σε κάθε κορυφή, και στη συνέχεια παράγεται ξανά το πολύγωνο, χρησιμοποιώντας τις μετασχηματισμένες κορυφές. Για άλλα αντικείμενα, οι εξισώσεις των μετασχηματισμών αλλαγής κλίμακας εφαρμόζονται στις παραμέτρους που ορίζουν τα αντικείμενα. Για την αλλαγή του μεγέθους ενός κύκλου, μπορεί να μεταβληθεί η κλίμακα της ακτίνας του και να υπολογισθούν οι νέες θέσεις συντεταγμένων γύρω από την περιφέρειά του. Και για να αλλάξει το μέγεθος μίας έλλειψης, εφαρμόζονται οι παράμετροι αλλαγής κλίμακας στους δύο άξονές της και έπειτα σχεδιάζονται οι νέες θέσεις της έλλειψης σε σχέση με τις συντεταγμένες του κέντρου της [20].

4.3 Αναπαραστάσεις Πινάκων και Ομογενείς Συντεταγμένες

Αρκετές εφαρμογές γραφικών περιλαμβάνουν μία σειρά από γεωμετρικούς μετασχηματισμούς. Μία δυναμική κίνηση αντικειμένων (*animation*) μπορεί να απαιτεί το αντικείμενο να μεταφερθεί και να περιστραφεί σε κάθε στάδιο της κίνησης. Στο σχεδιασμό αντικειμένων και σε εφαρμογές κατασκευής εικόνων, πραγματοποιούνται μεταφορές, περιστροφές και αλλαγές στη κλίμακα, ώστε να

τοποθετηθούν τα αντικείμενα που συνθέτουν την εικόνα στις κατάλληλες θέσεις τους. Οι μετασχηματισμοί οπτικής περιλαμβάνουν επίσης διαδοχικές μεταφορές και περιστροφές, κατά τη μετατροπή των πρωταρχικών προδιαγραφών της σκηνης σε απεικόνιση σε μία συσκευή εξόδου. Στην ενότητα αυτή παρουσιάζεται πώς οι αναπαραστάσεις πινάκων, που αναπτύχθηκαν στην **Ενότητα 4.2**, μπορούν να επαναπροσδιοριστούν, ώστε τέτοιες διαδοχές μετασχηματισμών να μπορούν να επεξεργάζονται αποτελεσματικά.

Στην **Ενότητα 4.2** αναλύθηκε ότι καθένας από τους τρεις βασικούς δισδιάστατους μετασχηματισμούς (μεταφορά, περιστροφή, αλλαγή κλίμακας) μπορεί να εκφραστεί σε μία γενική μορφή πινάκων ως εξής:

$$P' = M_1 \cdot P + M_2 \quad (4-15)$$

με τις θέσεις συντεταγμένων P και P' να αναπαριστώνται ως διανύσματα-στήλες. Ο πίνακας M_1 είναι 2×2 και περιέχει παράγοντες πολλαπλασιασμού, ενώ ο M_2 είναι ένας πίνακας στήλη δύο στοιχείων που περιέχει όρους μεταφοράς. Για τη μεταφορά, ο M_1 είναι ο μοναδιαίος πίνακας. Για την περιστροφή και την αλλαγή κλίμακας, ο M_2 περιέχει τους όρους μεταφοράς, που συνδέονται με το σημείο περιστροφής (pivot point) ή το σταθερό σημείο αλλαγής κλίμακας (fixed point) αντίστοιχα. Για την παραγωγή μίας ακολουθίας μετασχηματισμών με τις συγκεκριμένες εξισώσεις, όπως για παράδειγμα αλλαγή κλίμακας ακολουθούμενη από περιστροφή και έπειτα μεταφορά, υπάρχει η δυνατότητα υπολογισμού των μετασχηματισμένων συντεταγμένων κατά βήματα. Πρώτα, οι θέσεις συντεταγμένων αλλάζουν κλίμακα, στη συνέχεια οι συντεταγμένες αλλαγής κλίμακας περιστρέφονται, και τελικά, οι συντεταγμένες που έχουν περιστραφεί μεταφέρονται. Μία πιο αποτελεσματική προσέγγιση, ωστόσο, είναι ο συνδυασμός των μετασχηματισμών, έτσι ώστε οι τελικές θέσεις συντεταγμένων να προκύπτουν απ' ευθείας από τις αρχικές συντεταγμένες, χωρίς τον υπολογισμό ενδιάμεσων τιμών των συντεταγμένων. Αυτό μπορεί να επιτευχθεί επαναπροσδιορίζοντας την εξίσωση (4-15) για να εξαλειφθεί η λειτουργία πρόσθεσης των πινάκων [3].

4.3.1 Ομογενείς Συντεταγμένες

Οι όροι πολλαπλασιασμού και μεταφοράς ενός δισδιάστατου γεωμετρικού μετασχηματισμού μπορούν να συνδυαστούν σε ένα μόνο πίνακα, εάν επεκταθεί η αναπαράσταση σε πίνακες 3×3 . Έτσι, η τρίτη στήλη του πίνακα μετασχηματισμού μπορεί να χρησιμοποιηθεί για τους όρους μεταφοράς, και όλες οι εξισώσεις μετασχηματισμών μπορούν να εκφραστούν ως πολλαπλασιασμοί πινάκων. Αλλά για να πραγματοποιηθεί αυτό, χρειάζεται επίσης να επεκταθεί η αναπαράσταση με τη μορφή πίνακα της δισδιάστατης θέσης συντεταγμένων σε ένα πίνακα-στήλη τριών στοιχείων. Μία συνήθης τεχνική για την εκπλήρωση κάτι τέτοιου είναι η επέκταση κάθε αναπαράστασης διδιάστατης θέσης συντεταγμένων (x, y) σε μία αναπαράσταση τριών στοιχείων (x_h, y_h, h) , που ονομάζονται *ομογενείς συντεταγμένες* (*homogeneous coordinates*), όπου η ομογενής παράμετρος h είναι μία μη μηδενική τιμή, τέτοια ώστε:

$$x = \frac{x_h}{h}, \quad y = \frac{y_h}{h} \quad (4-16)$$

Για το λόγο αυτό, μία γενική δισδιάστατη αναπαράσταση ομογενών συντεταγμένων μπορεί επίσης να γραφεί ως $(h \cdot x, h \cdot y, h)$. Για τους γεωμετρικούς μετασχηματισμούς, η ομογενής παράμετρος h μπορεί να επιλεγεί ως οποιαδήποτε μη μηδενική τιμή. Συνεπώς, υπάρχει άπειρος αριθμός ισοδύναμων ομογενών αναπαραστάσεων για κάθε σημείο συντεταγμένων (x, y) . Μία εύκολη επιλογή είναι η ανάθεση $h = 1$. Κάθε δισδιάστατη θέση τότε αναπαριστάται με τις ομογενείς συντεταγμένες $(x, y, 1)$. Άλλες τιμές για τη παράμετρο h χρειάζονται, για παράδειγμα, στη διατύπωση πινάκων στους τρισδιάστατους μετασχηματισμούς οπτικής.

Η έκφραση των θέσεων με ομογενείς συντεταγμένες επιτρέπει την αναπαράσταση όλων των εξισώσεων γεωμετρικών μετασχηματισμών με πολλαπλασιασμούς πινάκων, η οποία είναι και η συνήθης μέθοδος που χρησιμοποιείται στα συστήματα γραφικών. Οι δισδιάστατες θέσεις συντεταγμένων αναπαριστώνται με διανύσματα-στήλες τριών στοιχείων, και οι λειτουργίες δισδιάστατων μετασχηματισμών εκφράζονται ως πίνακες 3×3 [3].

4.3.2 Δισδιάστατος Πίνακας Μεταφοράς

Χρησιμοποιώντας την προσέγγιση ομογενών συντεταγμένων, οι εξισώσεις για μία δισδιάστατη μεταφορά ενός σημείου συντεταγμένων μπορούν να αναπαρασταθούν, χρησιμοποιώντας τον ακόλουθο πολλαπλασιασμό πινάκων:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4-17)$$

Αυτή η λειτουργία μεταφοράς μπορεί να γραφεί στην συντμημένη μορφή:

$$P' = T(t_x, t_y) \cdot P \quad (4-18)$$

Όπου $T(t_x, t_y)$ είναι ο 3×3 πίνακας μεταφοράς της εξίσωσης (4-17). Σε περιπτώσεις που δεν υπάρχει αμφιβολία σχετικά με τις παραμέτρους μεταφοράς, ο πίνακας μεταφοράς μπορεί να γραφεί απλά ως T [19].

4.3.3 Δισδιάστατος Πίνακας Περιστροφής

Παρόμοια, οι εξισώσεις μετασχηματισμού για δισδιάστατη περιστροφή περί την αρχή των συντεταγμένων μπορούν να εκφραστούν στην ακόλουθη μορφή πινάκων:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4-19)$$

ή ως:

$$P' = R(\theta) \cdot P \quad (4-20)$$

Η μαθηματική συνάρτηση $R(\theta)$ του μετασχηματισμού περιστροφής είναι ο 3×3 πίνακας της εξίσωσης (4-19) με παράμετρο περιστροφής θ . Ο πίνακας περιστροφής μπορεί να γραφεί και απλά ως R .

Σε μερικές βιβλιοθήκες γραφικών, μία συνάρτηση δισδιάστατης περιστροφής παράγει περιστροφές μόνο ως προς την αρχή των συντεταγμένων, όπως στην εξίσωση (4-19). Η περιστροφή περί οποιουδήποτε άλλου σημείου πρέπει να εκτελεστεί στη συνέχεια, ως ακολουθία από λειτουργίες μετασχηματισμών. Μία εναλλακτική προσέγγιση σε ένα πακέτο γραφικών είναι να παρέχονται επιπρόσθετες παράμετροι στη ρουτίνα περιστροφής για τις συντεταγμένες του σημείου περιστροφής. Έπειτα, μία ρουτίνα περιστροφής, που περιέχει μία παράμετρο για το σημείο περιστροφής, δημιουργεί ένα γενικό πίνακα περιστροφής, χωρίς την ανάγκη υποστήριξης από μία διαδοχή συναρτήσεων μετασχηματισμών [19].

4.3.4 Δισδιάστατος Πίνακας Αλλαγής Κλίμακας

Ο μετασχηματισμός αλλαγής κλίμακας ως προς την αρχή των συντεταγμένων μπορεί να εκφραστεί ως πολλαπλασιασμός πινάκων:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4-21)$$

ή

$$P' = S(s_x, s_y) \cdot P \quad (4-22)$$

Ο όρος αλλαγής κλίμακας $S(s_x, s_y)$ είναι ο 3×3 πίνακας της εξίσωσης (4-21) με παραμέτρους s_x και s_y . Στις περισσότερες των περιπτώσεων, ο πίνακας αλλαγής κλίμακας μπορεί να αναπαρασταθεί απλά ως S .

Κάποιες βιβλιοθήκες παρέχουν μία συνάρτηση αλλαγής κλίμακας, που μπορεί να παράγει αλλαγή μόνο ως προς την αρχή των συντεταγμένων, όπως στην εξίσωση (4-21). Στην περίπτωση αυτή, ένας μετασχηματισμός αλλαγής κλίμακας σε σχέση με ένα άλλο σημείο αναφοράς αντιμετωπίζεται ως μία διαδοχή από λειτουργίες μετασχηματισμών. Ωστόσο, άλλα συστήματα εμπεριέχουν μία γενική ρουτίνα αλλαγής κλίμακας, που μπορεί να κατασκευάσει τον ομογενή πίνακα για αλλαγή κλίμακας με βάση ένα καθορισμένο σημείο [19].

4.4 Αντίστροφοι Μετασχηματισμοί

Για την μεταφορά, ο αντίστροφος πίνακας προκύπτει με την αλλαγή πρόσημου των αποστάσεων μεταφοράς. Έτσι, εάν οι αποστάσεις της διδιάστατης μεταφοράς είναι t_x και t_y , ο αντίστροφος πίνακας μεταφοράς θα είναι:

$$T^{-1} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4-23)$$

Ο πίνακας T^{-1} παράγει μία μεταφορά στην αντίθετη κατεύθυνση, και το γινόμενο ενός πίνακα μεταφοράς και του αντιστρόφου του παράγει τον μοναδιαίο πίνακα.

Μία αντίστροφη περιστροφή επιτυγχάνεται αντικαθιστώντας τη γωνία περιστροφής με την αρνητική της. Ως παράδειγμα, μία δισδιάστατη περιστροφή κατά μία γωνία θ περί την αρχή των συντεταγμένων έχει τον ακόλουθο αντίστροφο πίνακα μετασχηματισμού:

$$R^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-24)$$

Οι αρνητικές τιμές για τις γωνίες περιστροφής παράγουν δεξιόστροφες περιστροφές, έτσι ο μοναδιαίος πίνακας παράγεται όταν οποιοσδήποτε πίνακας περιστροφής πολλαπλασιάζεται με τον αντίστρόφό του. Αφού μόνο η συνάρτηση του ημίτονου επηρεάζεται από την αλλαγή του προσήμου της γωνίας περιστροφής, ο αντίστροφος πίνακας μπορεί επίσης να προκύψει με την εσωτερική αλλαγή σειρών και στηλών. Για το λόγο αυτό, ο αντίστροφος οποιουδήποτε πίνακα περιστροφής R μπορεί να υπολογιστεί από τον ανάστροφό του ($R^{-1} = R^T$).

Ο αντίστροφος πίνακας για οποιαδήποτε αλλαγή κλίμακας διατυπώνεται με την αντικατάσταση των παραμέτρων με τις αντίστροφές τους. Για δισδιάστατη αλλαγή κλίμακας με τις παραμέτρους s_x και s_y να σχετίζονται με την αρχή των συντεταγμένων, ο πίνακας του αντίστροφου μετασχηματισμού είναι:

$$S^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-25)$$

Ο αντίστροφος πίνακας παράγει έναν αντίθετο μετασχηματισμό αλλαγής κλίμακας, έτσι ώστε ο πολλαπλασιασμός οποιουδήποτε πίνακα αλλαγής κλίμακας με τον αντίστρόφό του να παράγει τον μοναδιαίο πίνακα [3].

4.5 Σύνθετοι Δισδιάστατοι Μετασχηματισμοί

Χρησιμοποιώντας την αναπαράσταση πινάκων, γίνεται δυνατή η δημιουργία μίας ακολουθίας μετασχηματισμών ως ένας σύνθετος πίνακας, με τον υπολογισμό του γινομένου των μεμονωμένων μετασχηματισμών. Η μορφοποίηση των γινομένων των μετασχηματισμών συχνά αναφέρεται ως *αλληλουχία* (*concatenation*), ή *σύνθεση* (*composition*) πινάκων. Μιας και μία θέση συντεταγμένων αναπαριστάται με έναν πίνακα-στήλη ομογενών συντεταγμένων, απαιτείται ένας εκ των προτέρων πολλαπλασιασμός του πίνακα-στήλη με τους πίνακες που αναπαριστούν οποιαδήποτε ακολουθία μετασχηματισμών. Εφόσον πολλές θέσεις στη σκηνή, τυπικά, έχουν μετασχηματιστεί με την ίδια ακολουθία, είναι αποτελεσματικότερο να πολλαπλασιαστούν πρώτα οι πίνακες μετασχηματισμών για τη δημιουργία ενός μόνο

σύνθετου πίνακα. Έτσι, εάν είναι επιθυμητό να εφαρμοσθούν δύο μετασχηματισμοί του σημείου P , η μετασχηματισμένη θέση θα υπολογιστεί ως:

$$P' = M_2 \cdot M_1 \cdot P = M \cdot P \quad (4-26)$$

Η θέση συντεταγμένων μετασχηματίζεται χρησιμοποιώντας τον σύνθετο πίνακα M και όχι εφαρμόζοντας ξεχωριστά τους μετασχηματισμούς M_1 και M_2 [3].

4.5.1 Σύνθετες Δισδιάστατες Μεταφορές

Εάν δύο διαδοχικά διανύσματα μεταφοράς, (t_{1x}, t_{1y}) και (t_{2x}, t_{2y}) εφαρμοσθούν σε μία δισδιάστατη θέση συντεταγμένων P , η τελική μετασχηματισμένη τοποθεσία P' υπολογίζεται ως:

$$P' = T(t_{2x}, t_{2y}) \cdot \{T(t_{1x}, t_{1y}) \cdot P\} = \{T(t_{2x}, t_{2y}) \cdot T(t_{1x}, t_{1y})\} \cdot P \quad (4-27)$$

όπου τα P και P' αναπαριστώνται ως διανύσματα-στήλες ομογενών συντεταγμένων τριών στοιχείων. Το αποτέλεσμα αυτό επαληθεύεται με τον υπολογισμό του γινομένου των πινάκων για τα δύο συσχετιζόμενα σύνολα ομαδοποιημένων στοιχείων. Επίσης, ο πίνακας σύνθετου μετασχηματισμού για την συγκεκριμένη ακολουθία μεταφορών δίδεται:

$$\begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \quad (4-28)$$

ή

$$T(t_{2x}, t_{2y}) \cdot T(t_{1x}, t_{1y}) = T(t_{1x} + t_{2x}, t_{1y} + t_{2y}) \quad (4-29)$$

όπου αποδεικνύεται ότι δύο διαδοχικές μεταφορές είναι προσθετικές [3].

4.5.2 Σύνθετες Δισδιάστατες Περιστροφές

Δύο διαδοχικές περιστροφές, που εφαρμόζονται σε ένα σημείο P , παράγουν τη μετασχηματισμένη θέση:

$$P' = R(\theta_2) \cdot \{R(\theta_1) \cdot P\} = \{R(\theta_2) \cdot R(\theta_1)\} \cdot P \quad (4-30)$$

Με τον πολλαπλασιασμό των δύο πινάκων περιστροφής, αποδεικνύεται ότι δύο διαδοχικές περιστροφές είναι προσθετικές:

$$R(\theta_2) \cdot R(\theta_1) = R(\theta_1 + \theta_2) \quad (4-31)$$

οπότε οι τελικές συντεταγμένες περιστροφής ενός σημείου μπορούν να υπολογιστούν με το σύνθετο πίνακα περιστροφής ως [3]:

$$P' = R(\theta_1 + \theta_2) \cdot P \quad (4-32)$$

4.5.3 Σύνθετες Δισδιάστατες Αλλαγές Κλίμακας

Οι πίνακες αλληλουχίας μετασχηματισμών για δύο διαδοχικές διαδικασίες αλλαγής κλίμακας στις δύο διαστάσεις παράγουν τον ακόλουθο σύνθετο πίνακα αλλαγής κλίμακας:

$$\begin{bmatrix} s_{2x} & 0 & 0 \\ 0 & s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{1x} & 0 & 0 \\ 0 & s_{1y} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{1x} \cdot s_{2x} & 0 & 0 \\ 0 & s_{1y} \cdot s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-33)$$

ή

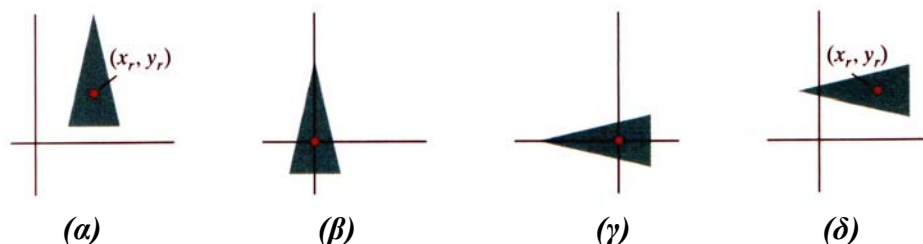
$$S(s_{2x}, s_{2y}) \cdot S(s_{1x}, s_{1y}) = S(s_{1x} \cdot s_{2x}, s_{1y} \cdot s_{2y}) \quad (4-34)$$

Ο πίνακας, που προκύπτει στην περίπτωση αυτή, υποδεικνύει ότι διαδοχικές λειτουργίες αλλαγής κλίμακας είναι αποτέλεσμα πολλαπλασιασμού. Ως παράδειγμα, εάν το μέγεθος ενός αντικειμένου τριπλασιασθεί δύο φορές διαδοχικά, το τελικό μέγεθος θα είναι εννέα φορές μεγαλύτερο του αρχικού [3].

4.5.4 Γενική Δισδιάστατη Περιστροφή Περί Σημείου (Pivot-Point Rotation)

Όταν ένα πακέτο γραφικών παρέχει μόνο μία συνάρτηση περιστροφής ως προς την αρχή των συντεταγμένων, μία δισδιάστατη περιστροφή γύρω από οποιοδήποτε άλλο σημείο (x_r, y_r) παράγεται με την εκτέλεση της ακόλουθης διαδοχής από λειτουργίες μεταφοράς – περιστροφής – μεταφοράς:

1. Μεταφορά του αντικειμένου, έτσι ώστε η θέση του σημείου περιστροφής να μετακινηθεί στην αρχή των συντεταγμένων.
2. Περιστροφή του αντικειμένου περί την αρχή των αξόνων.
3. Μεταφορά του αντικειμένου, έτσι ώστε το σημείο περιστροφής να επιστρέψει στην αρχική του θέση.



Εικόνα 4.7: Μία ακολουθία μετασχηματισμών για τη περιστροφή ενός αντικείμενου περί ενός καθορισμένου σημείου περιστροφής [3].

Αυτή η ακολουθία μετασχηματισμών παρουσιάζεται στην **Εικόνα 4.7**. Ο πίνακας σύνθετου μετασχηματισμού για την συγκεκριμένη ακολουθία προκύπτει από την αλληλουχία:

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos \theta & -\sin \theta & x_r(1 - \cos \theta) + y_r \sin \theta \\ \sin \theta & \cos \theta & y_r(1 - \cos \theta) - x_r \sin \theta \\ 0 & 0 & 1 \end{bmatrix} \quad (4-35)$$

η οποία μπορεί να εκφραστεί στη μορφή:

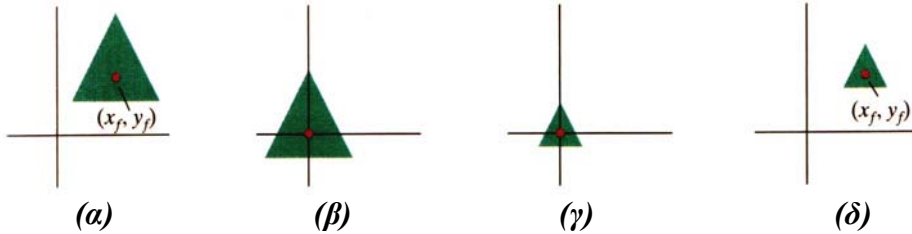
$$T(x_r, y_r) \cdot R(\theta) \cdot T(-x_r, -y_r) = R(x_r, y_r, \theta) \quad (4-36)$$

όπου, $T(-x_r, -y_r) = T^{-1}(x_r, y_r)$. Γενικά, μία συνάρτηση περιστροφής σε μία βιβλιοθήκη γραφικών μπορεί να δομηθεί ώστε να δέχεται ως παραμέτρους τις συντεταγμένες του σημείου περιστροφής, όπως επίσης και την ανάλογη γωνία, και να παράγει αυτόματα το πίνακα περιστροφής της εξίσωσης (4-35) [3].

4.5.5 Γενική Δισδιάστατη Αλλαγή Κλίμακας Βάση Σημείου (Fixed-Point Scaling)

Στην **Εικόνα 4.8** εμφανίζεται μία διαδοχή μετασχηματισμών, για την παραγωγή μίας δισδιάστατης αλλαγής κλίμακας με αναφορά σε μία επιλεγμένη σταθερή θέση (x_f, y_f) , όταν υπάρχει συνάρτηση που μπορεί να αλλάξει τη κλίμακα ενός αντικειμένου σε σχέση μόνο με την αρχή των συντεταγμένων. Η ακολουθία αυτή είναι:

1. Μεταφορά του αντικειμένου, έτσι ώστε το σταθερό σημείο να ταυτίζεται με την αρχή των συντεταγμένων.
2. Αλλαγή κλίμακας του αντικειμένου με αναφορά στην αρχή των συντεταγμένων.
3. Χρήση της αντίστροφης μεταφοράς του βήματος (1), για την επαναφορά του αντικειμένου στην αρχική θέση.



Εικόνα 4.8: Μία ακολουθία μετασχηματισμών, για την αλλαγή κλίμακας ενός αντικειμένου ως προς μία καθορισμένη σταθερή θέση [3].

Η αλληλουχία των πινάκων για τις τρεις αυτές λειτουργίες παράγει τον επιθυμητό πίνακα αλλαγής κλίμακας:

$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & x_f(1 - s_x) \\ 0 & s_y & y_f(1 - s_y) \\ 0 & 0 & 1 \end{bmatrix} \quad (4-37)$$

ή

$$T(x_f, y_f) \cdot S(s_x, s_y) \cdot T(-x_f, -y_f) = S(x_f, y_f, s_x, s_y) \quad (4-38)$$

Ο συγκεκριμένος μετασχηματισμός παράγεται αυτόματα σε συστήματα που παρέχουν συνάρτηση αλλαγής κλίμακας, που δέχεται συντεταγμένες για το σταθερό σημείο [3].

4.6 Γεωμετρικοί Μετασχηματισμοί Σε Τρισδιάστατο Χώρο

Οι μέθοδοι για τους γεωμετρικούς μετασχηματισμούς στις τρεις διαστάσεις είναι επέκταση των δισδιάστατων μεθόδων συμπεριλαμβάνοντας θεωρήσεις για τη z-συντεταγμένη. Η μεταφορά ενός αντικείμενου πραγματοποιείται με τον προσδιορισμό ενός τρισδιάστατου διανύσματος μεταφοράς, που προσδιορίζει κατά πόσο το αντικείμενο μετακινείται σε κάθε μία από τις τρεις διευθύνσεις συντεταγμένων. Παρόμοια, ένα αντικείμενο αλλάζει μέγεθος με την επιλογή ενός παράγοντα αλλαγής κλίμακας για καθεμία από τις τρεις Καρτεσιανές συντεταγμένες. Αλλά, η επέκταση των διδιάστατων μεθόδων περιστροφής στις τρεις διαστάσεις είναι περισσότερο πολύπλοκη.

Στη περιγραφή των δισδιάστατων περιστροφών στο xy-επίπεδο (**Ενότητα 4.2.2**), εξετάστηκαν μόνο περιστροφές περί αξόνων που ήταν κάθετοι στο xy-επίπεδο. Στον τρισδιάστατο χώρο είναι δυνατή η επιλογή οποιουδήποτε χωρικού προσανατολισμού για τον άξονα περιστροφής. Κάποια πακέτα γραφικών διαχειρίζονται την περιστροφή στις τρεις διαστάσεις ως μία σύνθεση από τρεις περιστροφές, μία περί τον καθένα από τους τρεις Καρτεσιανούς άξονες. Εναλλακτικά, είναι δυνατό να κατασκευαστούν γενικές εξισώσεις περιστροφής, δεδομένου του προσανατολισμού του άξονα και της απαιτούμενης γωνίας περιστροφής.

Μία θέση στον τρισδιάστατο χώρο, εκφρασμένη σε ομογενείς συντεταγμένες, αναπαριστάται ως ένα διάνυσμα-στήλη τεσσάρων στοιχείων. Έτσι, κάθε τελεστής γεωμετρικού μετασχηματισμού είναι πλέον ένας 4x4 πίνακας, ο οποίος πολλαπλασιάζει το διάνυσμα-στήλη των συντεταγμένων. Όπως στις δύο διαστάσεις, κάθε ακολουθία μετασχηματισμών αναπαριστάται από ένα μόνο πίνακα, ο οποίος διαμορφώνεται από την αλληλουχία πινάκων των μεμονωμένων μετασχηματισμών. Κάθε διαδοχικός πίνακας στην ακολουθία των μετασχηματισμών συνδέεται αλυσιδωτά στα αριστερά των προηγούμενων πινάκων [20].

4.6.1 Τρισδιάστατη Μεταφορά

Μία θέση $P = (x, y, z)$ σε ένα χώρο τριών διαστάσεων μεταφέρεται σε μία τοποθεσία $P' = (x', y', z')$ με την προσθήκη των αποστάσεων μεταφοράς t_x, t_y, t_z στις Καρτεσιανές συντεταγμένες του P :

$$x' = x + t_x \quad y' = y + t_y \quad z' = z + t_z \quad (4-39)$$

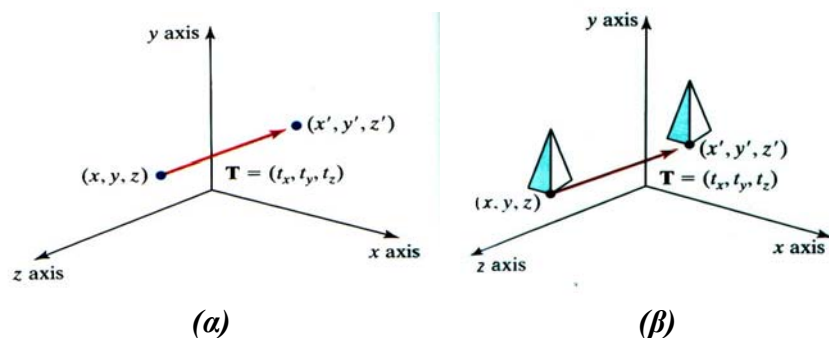
Στην **Εικόνα 4.9(α)** παρουσιάζεται η τρισδιάστατη μεταφορά ενός σημείου.

Οι λειτουργίες μεταφοράς στις τρεις διαστάσεις μπορούν να εκφραστούν με τη μορφή πίνακα όπως η εξίσωση (4-17). Στη περίπτωση αυτή, όμως, οι θέσεις συντεταγμένων, P και P' , αναπαριστώνται σε ομογενείς συντεταγμένες με πίνακες-στήλες τεσσάρων στοιχείων, ενώ ο τελεστής μεταφοράς T είναι ένας 4x4 πίνακας:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4-40)$$

ή

$$P' = T \cdot P \quad (4-41)$$



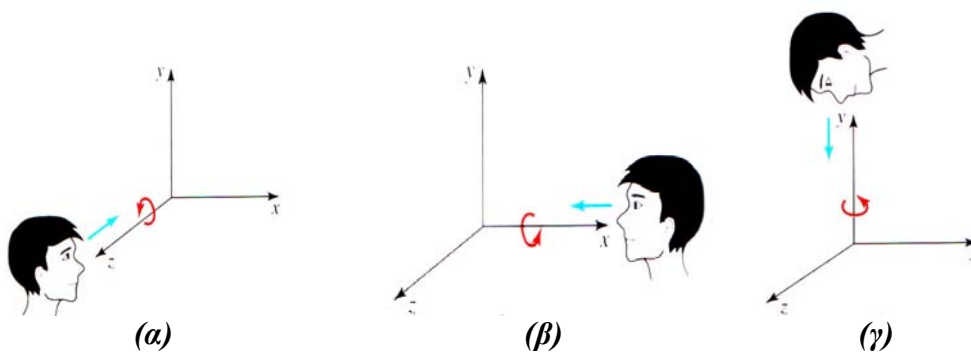
Εικόνα 4.9:(α) Μετακίνηση μίας θέσης συντεταγμένων με ένα διάνυσμα μεταφοράς $T=(t_x, t_y, t_z)$ και (β) αλλαγή θέσης ενός τρισδιάστατου αντικειμένου χρησιμοποιώντας διάνυσμα μεταφοράς T [3].

Στις τρεις διαστάσεις, ένα αντικείμενο μεταφέρεται με το μετασχηματισμό καθεμίας από τις θέσεις συντεταγμένων που το καθορίζουν, και έπειτα με την κατασκευή του αντικειμένου στη νέα τοποθεσία. Για ένα αντικείμενο, που περιγράφεται από ένα σύνολο πολυγωνικών επιφανειών, μεταφέρεται κάθε κορυφή για κάθε επιφάνεια (**Εικόνα 4.9(β)**) και απεικονίζονται ξανά οι πλευρές του πολυγώνου στις μεταφερθείσες θέσεις.

Η αντίστροφη διαδικασία με ένα πίνακα τρισδιάστατης μεταφοράς επιτυγχάνεται με τη χρήση όμοιων τεχνικών με αυτές που εφαρμόστηκαν στις δύο διαστάσεις. Δηλαδή, χρησιμοποιούνται οι αρνητικές τιμές των αποστάσεων μεταφοράς t_x , t_y και t_z . Η διαδικασία αυτή παράγει μία μεταφορά στην αντίθετη κατεύθυνση, ενώ το γινόμενο ενός πίνακα μεταφοράς με τον αντίστροφό του δίνει το μοναδιαίο πίνακα [20].

4.6.2 Τρισδιάστατη Περιστροφή

Στο χώρο, η περιστροφή ενός αντικειμένου μπορεί να γίνει γύρω από οποιοδήποτε άξονα, αλλά οι πιο εύκολοι στη διαχείριση άξονες περιστροφής είναι εκείνοι που είναι παράλληλοι στους άξονες του συστήματος Καρτεσιανών συντεταγμένων. Επιπλέον, μπορούν να χρησιμοποιηθούν συνδυασμοί περιστροφών περί των αξόνων συντεταγμένων (σε συνδυασμό με κατάλληλες μεταφορές) για τον καθορισμό περιστροφής γύρω από οποιοδήποτε ευθύγραμμο τμήμα στο χώρο. Έτσι, θα γίνει αναφορά πρώτα στις λειτουργίες περιστροφών περί των αξόνων του συστήματος συντεταγμένων, και έπειτα στους υπολογισμούς που απαιτούνται για άλλους, τυχαίους, άξονες περιστροφής.



Εικόνα 4.10: Οι θετικές περιστροφές περί ενός άξονα συντεταγμένων είναι αριστερόστροφες, για έναν παρατηρητή που κοιτάει κατά μήκος του θετικού μέρους του άξονα προς την αρχή [3].

Κατά σύμβαση, οι θετικές γωνίες περιστροφής παράγουν αριστερόστροφες περιστροφές γύρω από έναν άξονα συντεταγμένων, υποθέτοντας ότι ο παρατηρητής κοιτάει προς την αρνητική διεύθυνση κατά μήκος του άξονα (**Εικόνα 4.10**). Η σύμβαση αυτή συμφωνεί με την αναφορά στις περιστροφές στις δύο διαστάσεις (**Ενότητα 4.2.2**), όπου θετικές περιστροφές στο xy -επίπεδο θεωρούνται εκείνες που είναι αριστερόστροφες ως προς το σημείο περιστροφής (άξονας παράλληλος στον z -άξονα) [20].

4.6.2.1 Τρισδιάστατες Περιστροφές Περί Αξόνων Συστήματος Συντεταγμένων

Οι δισδιάστατες εξισώσεις περιστροφής περί τον z -άξονα επεκτείνονται εύκολα στις τρεις διαστάσεις ως εξής:

$$\begin{aligned} x' &= x \cos \theta - y \sin \theta \\ y' &= x \sin \theta + y \cos \theta \\ z' &= z \end{aligned} \quad (4-42)$$

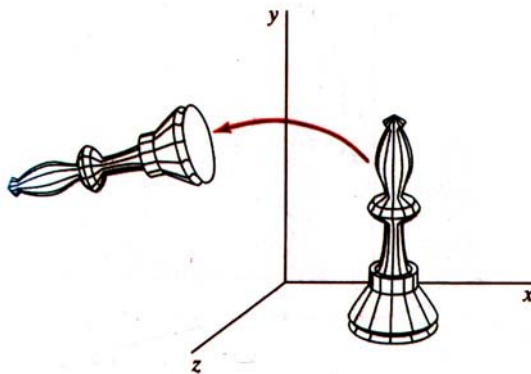
Η παράμετρος θ καθορίζει τη γωνία περιστροφής γύρω από τον z -άξονα, και η τιμή της z -συντεταγμένης παραμένει αμετάβλητη από αυτόν το μετασχηματισμό. Σε μορφή ομογενών συντεταγμένων, οι εξισώσεις της τρισδιάστατης περιστροφής γύρω από τον z -άξονα δίδονται:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4-43)$$

οι οποίες μπορούν να γραφούν σε πιο συμπτυκτωμένη μορφή ως:

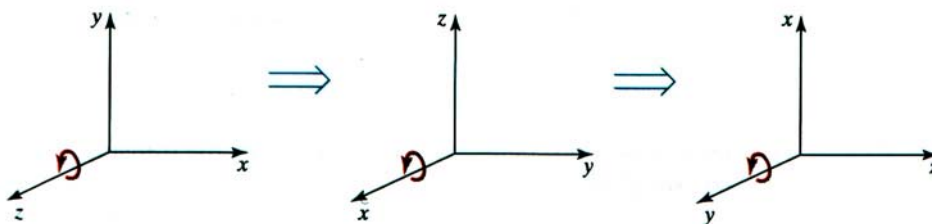
$$P' = R_z(\theta) \cdot P \quad (4-44)$$

Στην **Εικόνα 4.11** εμφανίζεται η περιστροφή ενός αντικειμένου περί του z -άξονα.



Εικόνα 4.11: Περιστροφή ενός αντικειμένου γύρω από τον z-άξονα [3].

Οι εξισώσεις μετασχηματισμών για περιστροφές γύρω από τους άλλους δύο άξονες συντεταγμένων προκύπτουν από κυκλική μετάθεση των παραμέτρων των συντεταγμένων x , y , και z στην εξίσωση (4-42), ($x \rightarrow y \rightarrow z \rightarrow x$). Συνεπώς, οι μετασχηματισμοί περιστροφής περί του x -άξονα και y -άξονα προκύπτουν με την κυκλική αντικατάσταση του x με y , του y με z , και του z με x , όπως εμφανίζεται στην **Εικόνα 4.12**.

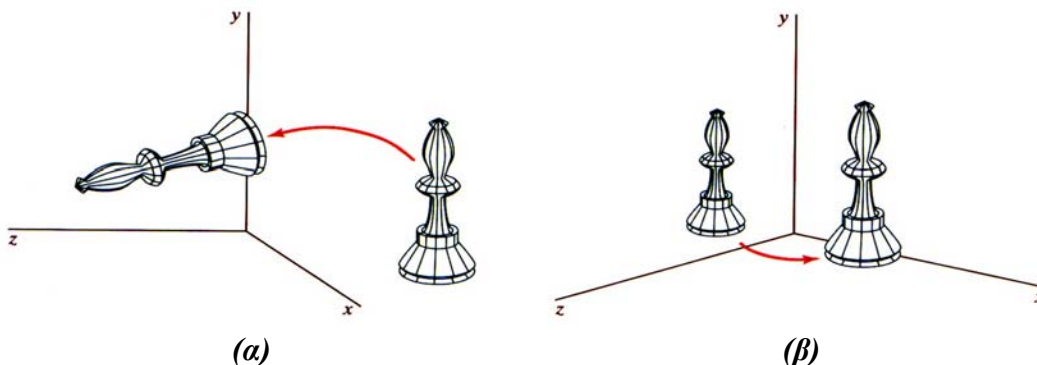


Εικόνα 4.12: Κυκλική μετάθεση των Καρτεσιανών αξόνων συντεταγμένων για τη παραγωγή τριών συνόλων από εξισώσεις περιστροφής γύρω από αυτούς [3].

Αντικαθιστώντας τις μεταθέσεις στην εξίσωση (4-42), προκύπτουν οι εξισώσεις περιστροφής γύρω από τον x -άξονα:

$$\begin{aligned} y' &= y \cos \theta - z \sin \theta \\ z' &= y \sin \theta + z \cos \theta \\ x' &= x \end{aligned} \quad (4-45)$$

Η περιστροφή ενός αντικειμένου γύρω από τον x -άξονα φαίνεται στην **Εικόνα 4.13(α)**.



Εικόνα 4.13: Περιστροφή αντικειμένου γύρω από (α) τον x -άξονα και (β) τον y -άξονα [3].

Η κυκλική μετάθεση των συντεταγμένων στις εξισώσεις (4-45) αποδίδει τις εξισώσεις μετασχηματισμού για περιστροφή γύρω από τον y-άξονα:

$$\begin{aligned} z' &= z \cos \theta - x \sin \theta \\ x' &= z \sin \theta + x \cos \theta \\ y' &= y \end{aligned} \quad (4-46)$$

Ένα παράδειγμα περιστροφής περί τον y-άξονα παρουσιάζεται στην **Εικόνα 4.13(β)**.

Ο αντίστροφος πίνακας τρισδιάστατης περιστροφής προκύπτει με τον ίδιο τρόπο όπως και στις δύο διαστάσεις. Απλά αντικαθιστάται η τιμή της γωνίας θ με $-\theta$. Οι αρνητικές τιμές για τις γωνίες περιστροφής παράγουν δεξιόστροφες περιστροφές, ενώ προκύπτει ο μοναδιαίος πίνακας όταν πολλαπλασιαστεί οποιοσδήποτε πίνακας περιστροφής με τον αντίστρόφό του. Εφόσον μόνο η συνάρτηση του ημίτονου επηρεάζεται από την αλλαγή του προσήμου της γωνίας περιστροφής, ο αντίστροφος πίνακας μπορεί επίσης να προκύψει από την εσωτερική αλλαγή σειρών και στηλών του πίνακα. Αυτό σημαίνει ότι ο αντίστροφος οποιουδήποτε πίνακα περιστροφής R προκύπτει σχηματίζοντας τον ανάστροφό του ($R^{-1} = R^T$) [3].

4.6.2.2 Γενικές Τρισδιάστατες Περιστροφές

Ο πίνακας περιστροφής γύρω από οποιοδήποτε άξονα, που δεν ταυτίζεται με κάποιο άξονα του συστήματος συντεταγμένων, μπορεί να κατασκευαστεί θεωρώντας ένα σύνθετο μετασχηματισμό, που περιλαμβάνει συνδυασμούς μεταφορών και περιστροφών γύρω από τους άξονες συντεταγμένων. Αρχικά, ο καθορισμένος άξονας περιστροφής μετακινείται έως ότου ταυτιστεί με κάποιον από τους άξονες του συστήματος συντεταγμένων και στη συνέχεια εφαρμόζεται ο κατάλληλος πίνακας περιστροφής για αυτόν τον άξονα. Το τελευταίο βήμα στη διαδοχή των μετασχηματισμών είναι η επαναφορά του άξονα περιστροφής στην αρχική του θέση.

Στην ειδική περίπτωση, που ένα αντικείμενο πρόκειται να περιστραφεί περί ενός άξονα, που είναι παράλληλος κάποιου άξονα συντεταγμένων, η επιθυμητή περιστροφή επιτυγχάνεται με την ακόλουθη διαδοχή μετασχηματισμών:

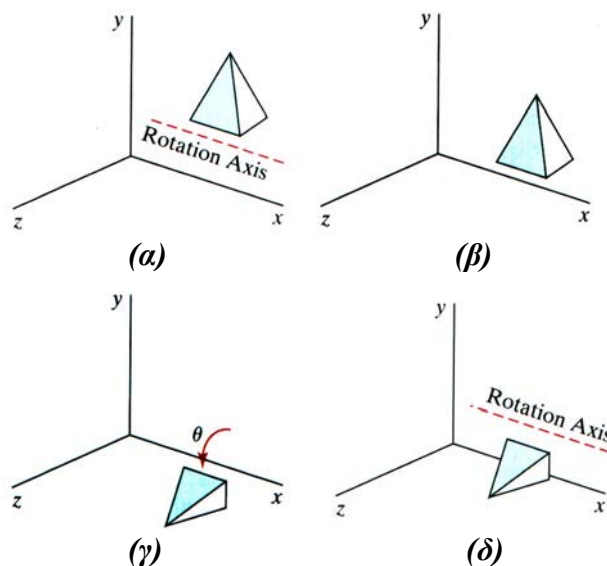
1. Μεταφορά του αντικειμένου, έτσι ώστε ο άξονας περιστροφής να ταυτιστεί με τον παράλληλό του άξονα συντεταγμένων.
2. Πραγματοποίηση της καθορισμένης περιστροφής περί του άξονα.
3. Μεταφορά του αντικειμένου έτσι ώστε ο άξονα περιστροφής να μετακινηθεί πίσω στην αρχική του θέση [19].

Τα βήματα της διαδικασίας αυτής εμφανίζονται στην **Εικόνα 4.14**. Μία θέση συντεταγμένων P μετασχηματίζεται, με τη σειρά που παρουσιάζεται στην εικόνα, ως εξής:

$$P' = T^{-1} \cdot R_x(\theta) \cdot T \cdot P \quad (4-47)$$

όπου ο σύνθετος πίνακας περιστροφής για το μετασχηματισμό δίδεται:

$$R(\theta) = T^{-1} \cdot R_x(\theta) \cdot T \quad (4-48)$$



Εικόνα 4.14: Ακολουθία μετασχηματισμών για τη περιστροφή αντικειμένου γύρω από άξονα παράλληλο στον x-άξονα [3].

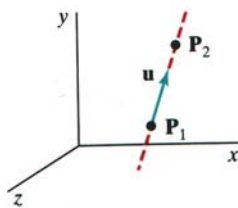
Ο σύνθετος αυτός πίνακας είναι της ίδιας μορφής με αυτόν της δισδιάστατης διαδοχής μετασχηματισμών, για την περίπτωση περιστροφής γύρω από άξονα, που είναι παράλληλος στον z-άξονα (σημείο περιστροφής που δεν βρίσκεται στην αρχή των αξόνων).

Όταν ένα αντικείμενο πρόκειται να περιστραφεί γύρω από έναν άξονα που δεν είναι παράλληλος με κάποιον από τους άξονες συντεταγμένων, πρέπει να πραγματοποιηθούν κάποιοι επιπρόσθετοι μετασχηματισμοί. Στη περίπτωση αυτή πρέπει επίσης οι περιστροφές να ευθυγραμμίζουν τον άξονα περιστροφής με έναν επιλεγμένο άξονα συντεταγμένων και έπειτα να τον επαναφέρουν στον αρχικό του προσανατολισμό. Δεδομένων των προδιαγραφών για τον άξονα και τη γωνία περιστροφής, είναι δυνατόν να επιτευχθεί η επιθυμητή περιστροφή σε πέντε βήματα:

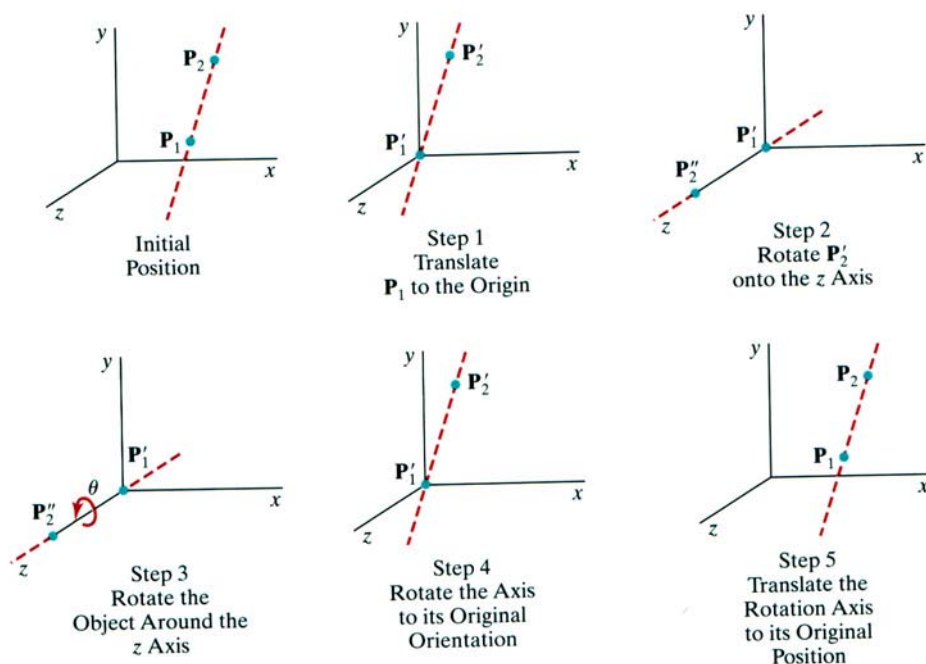
1. Μεταφορά του αντικειμένου, έτσι ώστε ο άξονας περιστροφής να διέρχεται από την αρχή των αξόνων.
2. Περιστροφή του αντικειμένου έτσι ώστε ο άξονας περιστροφής να ταυτιστεί με κάποιον από τους άξονες συντεταγμένων.
3. Πραγματοποίηση της καθορισμένης περιστροφής γύρω από τον επιλεγμένο άξονα συντεταγμένων.
4. Εφαρμογή αντίστροφης περιστροφής για την επαναφορά του άξονα περιστροφής στον αρχικό του προσανατολισμό.
5. Εφαρμογή αντίστροφης μεταφοράς για την επαναφορά του άξονα περιστροφής στην αρχική του θέση στο χώρο [19].

Ο άξονας περιστροφής μπορεί να μετασχηματιστεί σε οποιονδήποτε από του τρεις άξονες του συστήματος συντεταγμένων. Ο z-άξονας είναι συχνά η πιο βολική επιλογή. Στην **Εικόνα 4.16** θεωρείται μία διαδοχή μετασχηματισμών, χρησιμοποιώντας τον πίνακα περιστροφής περί τον z-άξονα.

Ένας άξονας περιστροφής μπορεί να οριστεί με δύο θέσεις συντεταγμένων, όπως φαίνεται στην **Εικόνα 4.15**, ή με ένα σημείο συντεταγμένων και τις γωνίες διευθύνσεων (ή συνημίτονα διευθύνσεων) μεταξύ του άξονα περιστροφής και δύο από του άξονες συντεταγμένων.



Εικόνα 4.15: Άξονας περιστροφής (διακεκομμένη γραμμή) που καθορίζεται από τα σημεία P_1 και P_2 [3].



Εικόνα 4.16: Πέντε βήματα μετασχηματισμού για να προκύψει ο σύνθετος πίνακας περιστροφής γύρω από τυχαίο άξονα, με τον άξονα περιστροφής να προβάλλεται επάνω στον z -άξονα [3].

Θεωρείται ότι ο άξονας περιστροφής ορίζεται από δύο σημεία, όπως απεικονίζεται, και ότι η διεύθυνση περιστροφής θα είναι αριστερόστροφη, καθώς ο παρατηρητής κοιτάει κατά μήκος του άξονα από το σημείο P_2 στο P_1 . Τα στοιχεία του διανύσματος περιστροφής του άξονα υπολογίζονται ως εξής:

$$V = P_2 - P_1 = (x_2 - x_1, y_2 - y_1, z_2 - z_1) \quad (4-49)$$

ενώ το μοναδιαίο διάνυσμα του άξονα περιστροφής δίδεται:

$$u = \frac{V}{|V|} = (a, b, c) \quad (4-50)$$

όπου τα στοιχεία a , b , και c είναι τα συνημίτονα διεύθυνσης για τους άξονες περιστροφής:

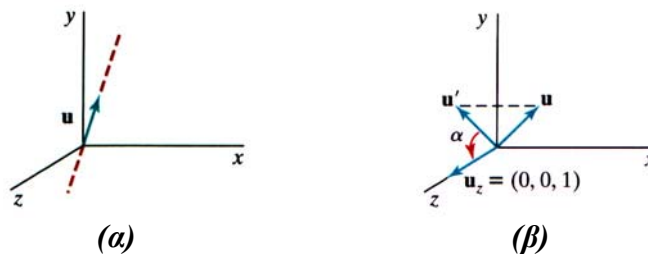
$$a = \frac{x_2 - x_1}{|V|}, \quad b = \frac{y_2 - y_1}{|V|}, \quad c = \frac{z_2 - z_1}{|V|} \quad (4-51)$$

Εάν η περιστροφή πρόκειται να πραγματοποιηθεί στην αντίθετη διεύθυνση (δεξιόστροφα για έναν παρατηρητή που κοιτάει από το P_2 προς το P_1), τότε θα αντιστραφεί το διάνυσμα V του άξονα και το μοναδιαίο u έτσι ώστε να δείχνουν στη διεύθυνση από το P_2 στο P_1 [9].

Το πρώτο βήμα στη σειρά για την πραγματοποίηση της περιστροφής είναι η κατασκευή του πίνακα μεταφοράς, ο οποίος επαναπροσδιορίζει τον άξονα περιστροφής, έτσι ώστε να διέρχεται από την αρχή των αξόνων του συστήματος συντεταγμένων. Εφόσον είναι επιθυμητή αριστερόστροφη περιστροφή για έναν παρατηρητή που κοιτάει κατά μήκος του άξονα από το P_2 στο P_1 (**Εικόνα 4.15**), μετακινείται το σημείο P_1 στην αρχή των συντεταγμένων. Εάν η περιστροφή είχε καθοριστεί προς την αντίθετη διεύθυνση, η μετακίνηση θα ήταν του σημείου P_2 στην αρχή των συντεταγμένων. Ο πίνακας μεταφοράς θα είναι:

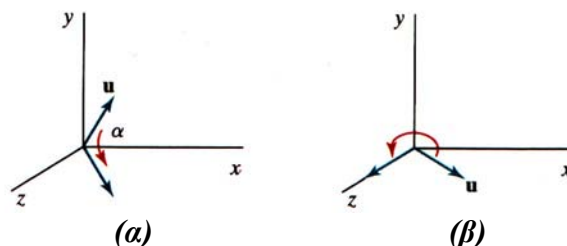
$$T = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-52)$$

ο οποίος επαναπροσδιορίζει τον άξονα περιστροφής και το αντικείμενο, όπως φαίνεται στην **Εικόνα 4.17(α)**.



Εικόνα 4.17:(α)Μεταφορά του άξονα περιστροφής στην αρχή των αξόνων. (β)Η περιστροφή του u περί του x -άξονα στο xz -επίπεδο επιτυγχάνεται με τη περιστροφή του u' κατά γωνία α μέχρι να συμπίψει στον z -άξονα [3].

Στη συνέχεια, μορφοποιούνται οι μετασχηματισμοί, που θα ταυτίσουν τον άξονα περιστροφής με τον z -άξονα. Μπορούν να χρησιμοποιηθούν οι περιστροφές των αξόνων συντεταγμένων για την επίτευξη της συγκεκριμένης στοίχισης σε δύο βήματα, ενώ υπάρχουν αρκετοί τρόποι για την πραγματοποίηση των δύο αυτών βημάτων. Για το συγκεκριμένο παράδειγμα, πρώτα γίνεται η περιστροφή περί τον x -άξονα και στη συνέχεια γύρω από τον y -άξονα. Η περιστροφή γύρω από τον x -άξονα φέρνει το διάνυσμα u στο xy -επίπεδο, και η περιστροφή περί τον y -άξονα το στρέφει γύρω από τον z -άξονα. Οι δύο αυτές περιστροφές παρουσιάζονται στην **Εικόνα 4.18**.



Εικόνα 4.18:(α)Το μοναδιαίο διάνυσμα u περιστρέφεται περί τον x -άξονα για να έρθει στο xz -επίπεδο. (β)Στη συνέχεια περιστρέφεται περί τον y -άξονα για να συμπίψει με τον z -άξονα [3].

Εφόσον οι υπολογισμοί περιστροφής περιλαμβάνουν τις συναρτήσεις ημίτονου και συνημίτονου, μπορούν να χρησιμοποιηθούν τυποποιημένες λειτουργίες των διανυσμάτων για να προκύψουν τα στοιχεία δύο πινάκων περιστροφής. Το εσωτερικό γινόμενο (dot product) διανυσμάτων μπορεί να χρησιμοποιηθεί για να προσδιορίσει τον όρο του συνημίτονου, και το εξωτερικό γινόμενο (cross product) για τον όρο του ημίτονου [19].

Ο πίνακας μετασχηματισμού για περιστροφή περί του x-άξονα προκύπτει με το καθορισμό των τιμών του ημίτονου και του συνημίτονου της απαραίτητης γωνίας περιστροφής για τη μεταφορά του διανύσματος u στο xy-επίπεδο. Αυτή η γωνία περιστροφής είναι η γωνία μεταξύ της προβολής του u στο yz-επίπεδο και του θετικού μέρους του z-άξονα (**Εικόνα 4.17(β)**). Εάν η προβολή του u στο yz-επίπεδο αναπαρασταθεί με το διάνυσμα $u' = (0, b, c)$, τότε το συνημίτονο της γωνίας περιστροφής α μπορεί να καθοριστεί από το εσωτερικό γινόμενο του u' με το μοναδιαίο διάνυσμα u_z κατά μήκος του z-άξονα:

$$\cos \alpha = \frac{u' \cdot u_z}{|u'| \cdot |u_z|} = \frac{c}{d} \quad (4-53)$$

όπου d είναι το μέτρο του διανύσματος u' :

$$d = \sqrt{b^2 + c^2} \quad (4-54)$$

Παρόμοια, προσδιορίζεται το ημίτονο της γωνίας α από το εξωτερικό γινόμενο του u' με το u_z . Η ανεξάρτητη συντεταγμένων μορφή του συγκεκριμένου εξωτερικού γινομένου δίδεται:

$$u' \times u_z = u_x |u'| |u_z| \sin \alpha \quad (4-55)$$

ενώ η Καρτεσιανή μορφή του εξωτερικού γινομένου δίνει:

$$u' \times u_z = u_x \cdot b \quad (4-56)$$

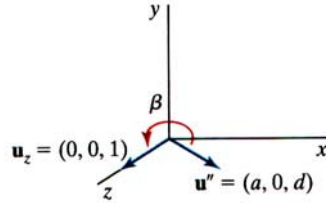
Εξισώνοντας το δεξιά μέρος των εξισώσεων (4-55) και (4-56) και σημειώνοντας ότι $|u_z|=1$ και $|u'|=d$, προκύπτει:

$$d \sin \alpha = b \text{ ή } \sin \alpha = \frac{b}{d} \quad (4-57)$$

Αφού έχουν καθοριστεί οι τιμές των $\cos \alpha$ και $\sin \alpha$ με παραμέτρους τα στοιχεία του διανύσματος u , είναι δυνατό να καθοριστούν τα στοιχεία του πίνακα για την περιστροφή αυτού του διανύσματος γύρω από τον x-άξονα και επάνω στο xz-επίπεδο:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{b}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-58)$$

Το επόμενο βήμα της μορφοποίησης της ακολουθίας των μετασχηματισμών είναι ο καθορισμός του πίνακα που θα περιστρέψει το μοναδιαίο διάνυσμα στο xz-επίπεδο αριστερόστροφα, γύρω από τον y-άξονα, μέχρι να συμπίσει με το θετικό μέρος του z-άξονα. Η **Εικόνα 4.19** παρουσιάζει τον προσανατολισμό του μοναδιαίου διανύσματος στο xz-επίπεδο, αποτέλεσμα της περιστροφής γύρω από τον x-άξονα [19].



Εικόνα 4.19: Περιστροφή του μοναδιαίου διανύσματος u'' γύρω από τον y-άξονα [3].

Το διάνυσμα αυτό, που συμβολίζεται u'' , έχει την τιμή a για το x στοιχείο του, εφόσον η περιστροφή περί τον x-άξονα αφήνει το στοιχείο x αμετάβλητο. Το z στοιχείο του είναι d (που είναι το μέτρο του u'), διότι το διάνυσμα u' έχει περιστραφεί μέχρι να συμπίσει με τον z-άξονα. Και το y στοιχείο του διανύσματος u'' είναι 0, διότι τώρα βρίσκεται στο xz-επίπεδο. Και στη περίπτωση αυτή είναι δυνατόν να προσδιοριστεί το συνημίτονο της γωνίας περιστροφής β από το εσωτερικό γινόμενο των μοναδιαίων διανυσμάτων u'' και u_z . Επομένως:

$$\cos \beta = \frac{u'' \cdot u_z}{|u''| |u_z|} = d \quad (4-59)$$

εφόσον $|u_z| = |u''| = 1$. Συγκρίνοντας την ανεξάρτητη συντεταγμένων μορφή του εξωτερικού γινομένου:

$$u'' \times u_z = u_y |u''| |u_z| \sin \beta \quad (4-60)$$

με την Καρτεσιανή μορφή:

$$u'' \times u_z = u_y \cdot (-a) \quad (4-61)$$

προκύπτει ότι:

$$\sin \beta = -a \quad (4-62)$$

Για το λόγο αυτό, ο πίνακας μετασχηματισμών για την περιστροφή του u'' περί τον y-άξονα δίδεται:

$$R_y(\beta) = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-63)$$

Με τους πίνακες μετασχηματισμών (4-52), (4-58), και (4-63), έχει πραγματοποιηθεί η στοίχιση του άξονα περιστροφής με το θετικό μέρος του z-άξονα. Η καθορισμένη γωνία περιστροφής θ μπορεί τώρα να εφαρμοστεί ως περιστροφή περί τον z-άξονα:

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-64)$$

Για την ολοκλήρωση της απαιτούμενης περιστροφής γύρω από τον δεδομένο άξονα, απαιτείται να μετασχηματιστεί ο άξονας περιστροφής πίσω στην αρχική του θέση. Αυτό πραγματοποιείται με την εφαρμογή αντιστροφής των μετασχηματισμών (4-52), (4-58), και (4-63). Ο πίνακας μετασχηματισμών για περιστροφή γύρω από έναν αυθαίρετο άξονα μπορεί να εκφραστεί ως η σύνθεση αυτών των επτά ανεξάρτητων μετασχηματισμών:

$$R(\theta) = T^{-1} \cdot R_x^{-1}(\alpha) \cdot R_y^{-1}(\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha) \cdot T \quad (4-65)$$

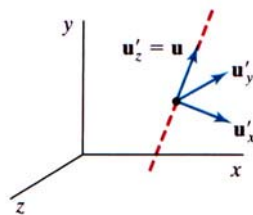
Μία πιο γρήγορη μέθοδος, αλλά ίσως λιγότερο διαισθητική, για να προκύψει ο σύνθετος πίνακας περιστροφής $R_y(\beta) \cdot R_x(\alpha)$ είναι η χρήση του γεγονότος ότι ο σύνθετος πίνακας οποιασδήποτε ακολουθίας τρισδιάστατων περιστροφών είναι της μορφής:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-66)$$

Ο άνω αριστερά 3x3 υποπίνακας αυτού του πίνακα είναι ορθογώνιος. Αυτό σημαίνει ότι οι σειρές (ή οι στήλες) του υποπίνακα αυτού διαμορφώνουν ένα σύνολο από ορθογώνια μοναδιαία διανύσματα, που περιστρέφονται από τον πίνακα R μέχρι να συμπίσουν στους x , y , και z άξονες, σύμφωνα με τη σχέση:

$$R \cdot \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad R \cdot \begin{bmatrix} r_{21} \\ r_{22} \\ r_{23} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad R \cdot \begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad (4-67)$$

Για το λόγο, είναι δυνατόν να καθοριστεί ένα τοπικό σύστημα συντεταγμένων με έναν από τους άξονές του στοιχισμένο με τον άξονα περιστροφής. Έπειτα, τα μοναδιαία διανύσματα των τριών αξόνων συντεταγμένων χρησιμοποιούνται για τη κατασκευή των στηλών του πίνακα περιστροφής. Υποθέτοντας ότι ο άξονας περιστροφής δεν είναι παράλληλος σε κάποιον από τους άξονες του συστήματος συντεταγμένων, μπορεί να σχηματιστεί το ακόλουθο σύνολο τοπικών μοναδιαίων διανυσμάτων (*Εικόνα 4.20*):



Εικόνα 4.20: Τοπικό σύστημα συντεταγμένων για έναν άξονα περιστροφής καθορισμένο από το μοναδιαίο διάνυσμα u [3].

$$u'_z = u, \quad u'_y = \frac{u \times u_x}{|u \times u_x|}, \quad u'_x = u'_y \times u'_z \quad (4-68)$$

Εάν τα στοιχεία των μοναδιαίων τοπικών διανυσμάτων για τον άξονα περιστροφής εκφραστούν ως:

$$u'_x = (u'_{x1}, u'_{x2}, u'_{x3}), \quad u'_y = (u'_{y1}, u'_{y2}, u'_{y3}), \quad u'_z = (u'_{z1}, u'_{z2}, u'_{z3}) \quad (4-69)$$

τότε ο απαιτούμενος σύνθετος πίνακας, ο οποίος είναι ίσος με το γινόμενο $R_y(\beta) \cdot R_x(\alpha)$, είναι ο:

$$R = \begin{bmatrix} u'_{x1} & u'_{x2} & u'_{x3} & 0 \\ u'_{y1} & u'_{y2} & u'_{y3} & 0 \\ u'_{z1} & u'_{z2} & u'_{z3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-70)$$

Ο πίνακας αυτός μετασχηματίζει τα μοναδιαία διανύσματα u'_x , u'_y , και u'_z ώστε να συμπίσουν επάνω στους x , y , και z άξονες, αντίστοιχα. Αυτό στοιχίζει τον άξονα περιστροφής με τον z -άξονα, διότι $u'_z = u$ [3].

4.6.3 Τρισδιάστατη Αλλαγή Κλίμακας

Η έκφραση με τη μορφή πίνακα για τον τρισδιάστατο μετασχηματισμό αλλαγής κλίμακας μίας θέσης $P = (x, y, z)$ ως προς την αρχή των αξόνων είναι απλή επέκταση της δισδιάστατης αλλαγής κλίμακας. Απλά συμπεριλαμβάνεται η παράμετρος για την αλλαγή κλίμακας στη z -διεύθυνση στον πίνακα μετασχηματισμού:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4-71)$$

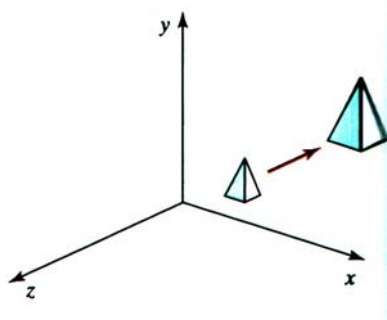
Ο τρισδιάστατος μετασχηματισμός αλλαγής κλίμακας για μία σημειακή θέση μπορεί να αναπαρασταθεί ως:

$$P' = S \cdot P \quad (4-72)$$

όπου οι παράμετροι αλλαγής κλίμακας s_x , s_y , και s_z μπορούν να πάρουν οποιαδήποτε θετική τιμή. Οι ρητές εκφράσεις για το μετασχηματισμό αλλαγής κλίμακας σε σχέση με την αρχή των αξόνων δίδονται:

$$x' = x \cdot s_x, \quad y' = y \cdot s_y, \quad z' = z \cdot s_z \quad (4-73)$$

Αλλάζοντας κλίμακα σε ένα αντικείμενο με τον μετασχηματισμό (4-71), αλλάζει η θέση του αντικειμένου ως προς την αρχή των αξόνων. Μία τιμή της παραμέτρου μεγαλύτερη από 1 απομακρύνει ένα σημείο από την αρχή των αξόνων προς την αντίστοιχη διεύθυνση συντεταγμένων. Παρόμοια, μία τιμή της παραμέτρου μικρότερη από 1 μετακινεί ένα σημείο κοντύτερα στην αρχή, στη διεύθυνση της συγκεκριμένης συντεταγμένης. Επιπλέον, εάν οι παράμετροι αλλαγής κλίμακας δεν είναι ίσοι, οι σχετικές διαστάσεις ενός μετασχηματισμένου αντικειμένου μεταβάλλονται. Το αρχικό σχήμα ενός αντικειμένου διατηρείται με την ομοιόμορφη αλλαγή κλίμακας: $s_x = s_y = s_z$. Το αποτέλεσμα της αλλαγής της κλίμακας ενός αντικειμένου ομοιόμορφα, με κάθε παράμετρο να είναι ίση με 2 φαίνεται στην **Εικόνα 4.21**.

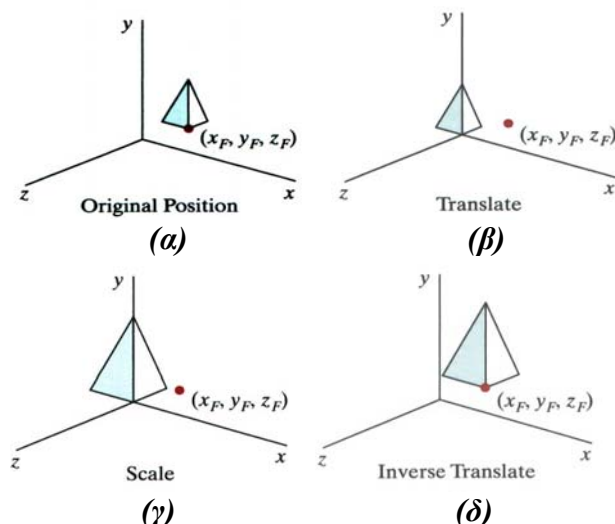


Εικόνα 4.21: Ο διπλασιασμός του μεγέθους ενός αντικειμένου με το μετασχηματισμό (4-71), το απομακρύνει από την αρχή των αξόνων [3].

Εφόσον κάποια πακέτα γραφικών παρέχουν μόνο μία ρουτίνα, που αλλάζει τη κλίμακα του αντικειμένου ως προς την αρχή των συντεταγμένων, μπορεί να κατασκευαστεί ένας μετασχηματισμός αλλαγής κλίμακας σε σχέση με οποιοδήποτε επιλεγμένο σταθερό σημείο (x_f, y_f, z_f) , χρησιμοποιώντας την εξής ακολουθία μετασχηματισμών.

1. Μεταφορά του σταθερού σημείου στην αρχή των αξόνων.
2. Εφαρμογή του μετασχηματισμού αλλαγής κλίμακας με αναφορά στην αρχή των συντεταγμένων χρησιμοποιώντας την εξίσωση (4-71).
3. Μεταφορά του σταθερού σημείου στην αρχική του θέση.

Η ακολουθία αυτή των μετασχηματισμών επιδεικνύεται στην **Εικόνα 4.22**. Η αναπαράσταση με τη μορφή πίνακα του μετασχηματισμού αλλαγής κλίμακας, με βάση ένα τυχαίο σταθερό σημείο, μπορεί να εκφραστεί ως αλληλουχία των μετασχηματισμών μεταφοράς – αλλαγής κλίμακας – μεταφοράς:



Εικόνα 4.22: Ακολουθία μετασχηματισμών για την αλλαγή κλίμακας αντικειμένου ως προς ένα επιλεγμένο σταθερό σημείο, χρησιμοποιώντας την εξίσωση (4-71) [3].

$$T(x_f, y_f, z_f) \cdot S(s_x, s_y, s_z) \cdot T(-x_f, -y_f, -z_f) = \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-74)$$

Για τη κατασκευή ενός πίνακα τρισδιάστατης αλλαγής κλίμακας είναι δυνατό να καθοριστούν προγραμματιστικές διαδικασίες, χρησιμοποιώντας είτε την ακολουθία μετασχηματισμών (μεταφορά – αλλαγή κλίμακας – μεταφορά) είτε την κατευθείαν ενσωμάτωση των συντεταγμένων του σταθερού σημείου.

Ο αντίστροφος πίνακας αλλαγής κλίμακας στις τρεις διαστάσεις καθορίζεται για τους πίνακες είτε της εξίσωσης (4-71) είτε της (4-74), με την αντικατάσταση κάθε παραμέτρου αλλαγής κλίμακας (s_x , s_y , και s_z) με τον αντίστροφό της ($\frac{1}{s_x}$, $\frac{1}{s_y}$, και $\frac{1}{s_z}$).

Αυτός όμως ο αντίστροφος μετασχηματισμός δεν ορίζεται εάν οποιαδήποτε παράμετρος έχει τιμή 0. Ο αντίστροφος πίνακας παράγει έναν αντίθετο μετασχηματισμό αλλαγής κλίμακας, ενώ το γινόμενο ενός πίνακα αλλαγής κλίμακας στις τρεις διαστάσεις με τον αντίστροφό του αποδίδει τον μοναδιαίο πίνακα [3].

4.6.4 Σύνθετοι Τρισδιάστατοι Μετασχηματισμοί

Όπως και στους μετασχηματισμούς στις δύο διαστάσεις, ένας σύνθετος τρισδιάστατος μετασχηματισμός μορφοποιείται με τον πολλαπλασιασμό των πινάκων, που αναπαριστούν τις ανεξάρτητες λειτουργίες στη διαδοχή των μετασχηματισμών. Οποιαδήποτε από τις διαδοχές μετασχηματισμών δύο διαστάσεων, που αναφέρθηκαν στην **Ενότητα 4.5**, μπορούν να πραγματοποιηθούν και στις τρεις διαστάσεις.

Μία διαδοχή μετασχηματισμών μπορεί να εκτελεστεί με την αλυσιδωτή σύνδεση των ανεξάρτητων πινάκων από τα δεξιά προς τα αριστερά ή από τα αριστερά προς τα δεξιά, ανάλογα με τη σειρά με την οποία έχουν οριστεί οι αναπαραστάσεις των πινάκων. Βέβαια, ο δεξιότερος όρος σε ένα γινόμενο πινάκων είναι πάντα ο

πρώτος μετασχηματισμός που εφαρμόζεται σε ένα αντικείμενο και ο αριστερότερος ο τελευταίος. Η σειρά για το γινόμενο των πινάκων είναι απαραίτητο να χρησιμοποιηθεί, διότι οι θέσεις συντεταγμένων αναπαριστώνται ως ένα διάνυσμα-στήλη τεσσάρων στοιχείων, το οποίο πολλαπλασιάζει τον 4x4 πίνακα του σύνθετου μετασχηματισμού [3].

ΟΠΤΙΚΗ ΚΑΙ ΠΡΟΒΟΛΕΣ

5.1 Εισαγωγή

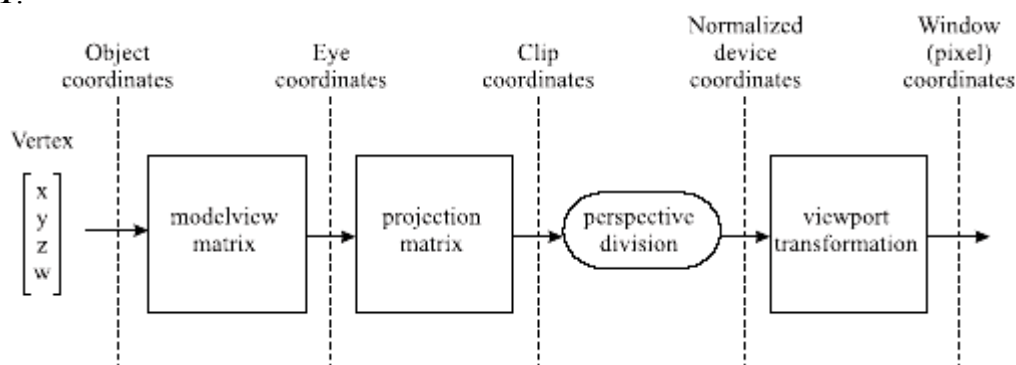
Στο κεφάλαιο 3 έγινε μία απλή αναφορά των ιδεών και λειτουργιών της οπτικής σε σχέση με τα συστήματα συντεταγμένων. Στο κεφάλαιο αυτό εξετάζονται με μεγαλύτερη λεπτομέρεια οι διαδικασίες για την απεικόνιση όψεων σε μία συσκευή εξόδου. Τυπικά, ένα πακέτο γραφικών επιτρέπει στο χρήστη να καθορίσει ποιο μέρος της καθορισμένης εικόνας θα απεικονιστεί και σε ποιο σημείο της συσκευής εξόδου. Οποιοδήποτε βολικό Καρτεσιανό σύστημα συντεταγμένων, που αναφέρεται στο χρήστη ως σύστημα παγκόσμιων συντεταγμένων, μπορεί να χρησιμοποιηθεί για να οριστεί η εικόνα.

Η ιδέα είναι απλή: κατασκευάζεται μία σκηνή (δισδιάστατη ή τρισδιάστατη) με χρήση των μετασχηματισμών μοντέλου (modeling transformations). Έπειτα λαμβάνεται μία εικόνα της σκηνής, χρησιμοποιώντας μία κάμερα, ενώ τέλος απεικονίζεται η εικόνα της κάμερας στην οθόνη επίδειξης (display screen). Για ευκολία, η διαδικασία μπορεί να χωριστεί σε τρία ξεχωριστά μέρη:

1. Αρχικά, διευκρινίζεται η θέση και ο προσανατολισμός της κάμερας.
2. Στη συνέχεια αποφασίζεται τι είδος εικόνας είναι επιθυμητό να δημιουργήσει η κάμερα. Συνήθως, για διδιάστατη γραφική αναπαράσταση χρησιμοποιείται ορθοκανονική προβολή, ενώ για τρισδιάστατη χρησιμοποιείται προοπτική.
3. Τέλος, περιγράφεται πώς θα χαρτογραφηθεί η εικόνα της κάμερας επάνω στην οθόνη απεικόνισης [20].

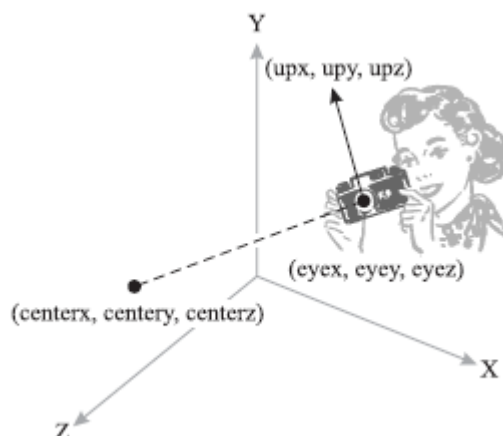
5.2 Έλεγχος της κάμερας

Μία τυπική, για τα περισσότερα πακέτα γραφικών, διαδικασία (ή κανάλι) μεταφοράς της οπτικής στο χρήστη (viewing pipeline) παρουσιάζεται στην **Εικόνα 5.1**.



Εικόνα 5.1: Κανάλι μεταφοράς της οπτικής στο χρήστη (viewing pipeline). Παρουσιάζεται η ακολουθία μετασχηματισμών και λειτουργιών που εφαρμόζεται σε ένα σημείο στις τρεις διαστάσεις. [20].

Οι παράμετροι, που χρησιμοποιούνται για τη ρύθμιση της θέσης και του προσανατολισμού της κάμερας, παρουσιάζονται στην **Εικόνα 5.2**. Η θέση της κάμερας στο χώρο, που μερικές φορές αποκαλείται και *eyepoint*, ορίζεται από το σημείο (*eyex*, *eyey*, *eyez*). Το σημείο (*centerx*, *centery*, *centerz*) καθορίζει το σημείο στο οποίο είναι στραμμένη η κάμερα. Μία τυπική επιλογή είναι το σημείο με το υψηλότερο ενδιαφέρον ή το κέντρο της σκηνής. Τα σημεία (*eyex*, *eyey*, *eyez*) και (*centerx*, *centery*, *centerz*) καθορίζουν ένα *διάνυσμα οπτικής* (*view vector*). Το σημείο (*upx*, *upy*, *upz*) ορίζει το άνω διάνυσμα της κάμερας, το οποίο και καθορίζει τον προσανατολισμό της κάμερας στο *eyepoint*.



Εικόνα 5.2: Τα σημεία για το καθορισμό της θέσης και του προσανατολισμού της κάμερας [20].

Το διάνυσμα οπτικής και το άνω διάνυσμα δεν είναι αναγκαίο να ορίζονται κάθετα μεταξύ τους (αν και εάν είναι παράλληλα θα προκύψουν παράξενες οπτικές). Συχνά το άνω διάνυσμα τίθεται προς μία σταθερή κατεύθυνση στη σκηνή, όπως για παράδειγμα δείχνοντας προς το θετικό μέρος του y-άξονα του παγκόσμιου συστήματος συντεταγμένων. Στη γενική περίπτωση, η κάμερα περιστρέφεται γύρω από τον άξονα του διανύσματος οπτικής, έως ότου το άνω μέρος της κάμερας συμπίσει με τη καθορισμένη κατεύθυνση προς τα επάνω όσο το δυνατόν περισσότερο [20].

Αυτό που γίνεται στην πράξη είναι να δημιουργηθεί ένας πίνακας μετασχηματισμού, που συμπεριλαμβάνει όλες της παραμέτρους της κάμερας, ο οποίος ονομάζεται *πίνακας οπτικής* (*viewing matrix*) ή *V*. Έπειτα, ο πίνακας αυτός πολλαπλασιάζει τον *πίνακα οπτικής του μοντέλου* (*modelview matrix*) *C* από δεξιά:

$$C \leftarrow C \cdot V \quad (5-1)$$

Στα περισσότερα πακέτα γραφικών, εάν ο χρήστης δεν τις αλλάξει, οι ρυθμίσεις προεπιλογής της κάμερας είναι οι ακόλουθες:

- τοποθετείται στην αρχή των αξόνων ($0, 0, 0$),
- είναι στραμμένη προς το αρνητικό μέρος του z-άξονα,
- και το άνω διάνυσμά της είναι παράλληλο στον y-άξονα [20].

5.3 Προβολές

Από τη στιγμή που η κάμερα έχει τοποθετηθεί και στραφεί προς την επιθυμητή διεύθυνση, το επόμενο βήμα είναι να διευκρινισθεί τι είδος εικόνας είναι επιθυμητή. Αυτό πραγματοποιείται χρησιμοποιώντας τον πίνακα προβολής (*projection matrix*) P . Συνήθως, ο μετασχηματισμός προβολής εφαρμόζεται αφότου έχει εφαρμοσθεί ο μετασχηματισμός οπτικής μοντέλου.

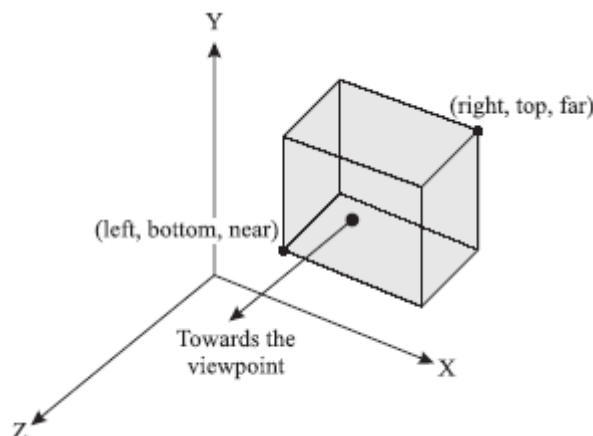
5.3.1 Όγκος οπτικής (view volume)

Ο αναγνώστης μπορεί να αναλογιστεί την αναλογία με μία φωτογραφική μηχανή, στην οποία επιλέγεται ο τύπος φακού που θα χρησιμοποιηθεί (π.χ. ευρυγώνιος φακός, κ.λπ.). Η επιλογή του φακού έχει επιπτώσεις στο οπτικό πεδίο, και καθορίζει ποιο μέρος του τρισδιάστατου κόσμου θα εμφανιστεί μέσα στα όρια της τελικής εικόνας. Ο όγκος του χώρου που εμφανίζεται τελικά στην εικόνα ονομάζεται *όγκος οπτικής* (*view volume* ή *frustum*). Όπως και στην απόρριψη των αντικειμένων που βρίσκονται έξω από το κάδρο μίας φωτογραφικής μηχανής, έτσι και οι λειτουργίες κάποιου πακέτου γραφικών επιβάλλουν τα όρια του πόσο μακριά πρέπει να είναι τα αντικείμενα από τη φωτογραφική μηχανή, προκειμένου να εμφανιστούν στην τελική εικόνα.

Το ακριβές τρισδιάστατο σχήμα του όγκου οπτικής εξαρτάται από ποιο είδος προβολής χρησιμοποιείται. Για ορθοκανονικές (παράλληλες) προβολές ο όγκος οπτικής έχει σχήμα κουτιού, ενώ οι προοπτικές προβολές διαμορφώνουν έναν όγκο οπτικής όπως μία περικομμένη πυραμίδα. Οι πλευρές, που περιβάλλουν τον όγκο οπτικής, καθορίζουν αποτελεσματικά έξι *επίπεδα αποκοπής* (*clipping planes*), τα οποία διαχωρίζουν το εσωτερικό του όγκου από τον αθέατο εξωτερικό κόσμο [20].

5.3.2 Ορθοκανονική προβολή

Αρχικά, δημιουργείται ο πίνακας για μία ορθοκανονική προβολή, και έπειτα αυτός πολλαπλασιάζει από αριστερά το τρέχοντα πίνακα, που είναι συνήθως ο πίνακας προβολής.



Εικόνα 5.3: Τα μεγέθη που ορίζουν έναν όγκο ορθοκανονικής οπτικής [20].

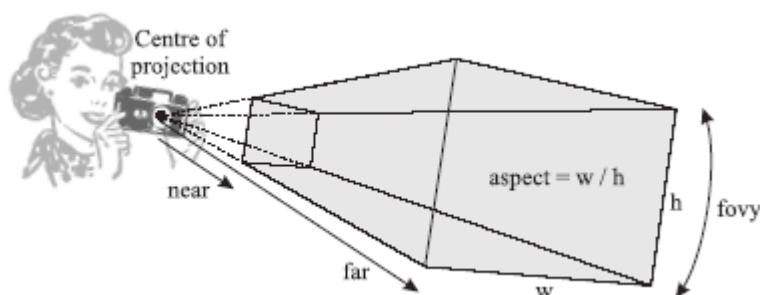
Στην **Εικόνα 5.3** παρουσιάζονται τα μεγέθη που ορίζουν μία ορθοκανονική προβολή. Οι τιμές των μεγεθών αυτών καθορίζουν έναν όγκο οπτικής σε σχήμα κουτιού, και είναι σημαντικό να τεθούν έτσι ώστε:

- $left < right$,
- $bottom < top$, και
- $near < far$.

Τα περιεχόμενα του όγκου οπτικής προβάλλονται επάνω σε μία ορθογώνια περιοχή στο xy-επίπεδο, με λόγο διαστάσεων $(right - left) / (top - bottom)$ [20].

5.3.3 Προοπτική προβολή

Αρχικά, δημιουργείται ο πίνακας για μία προοπτική προβολή, και έπειτα αυτός πολλαπλασιάζει από αριστερά το τρέχοντα πίνακα που είναι συνήθως ο πίνακας προβολής. Στην **Εικόνα 5.4** παρουσιάζονται τα μεγέθη, που ορίζουν μία προοπτική προβολή. Η *fovy* είναι η γωνία (σε μοίρες) του κάθετου πεδίου της οπτικής της εικόνας, *aspect* είναι ο λόγος διαστάσεων του όγκου οπτικής (το πλάτος του διαιρεμένο με το ύψος του), και τα *near* και *far* αντίστοιχα καθορίζουν τις θέσεις των κοντινών και μακρινών επιπέδων αποκοπής, μετρούμενα ως οι αποστάσεις τους από το κέντρο της προβολής (*eyepoint*). Τα *near* και *far* πρέπει να έχουν θετικές τιμές [20].

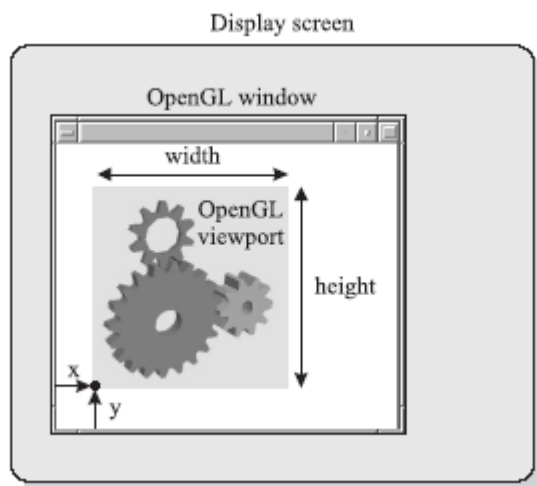


Εικόνα 5.4: Τα μεγέθη που ορίζουν έναν όγκο προοπτικής προβολής [20].

5.4 Ορισμός της οπτικής οδού (viewport)

Αρχικά, ορίζεται η θέση και το μέγεθος της *οπτικής οδού*. Η οπτική οδός είναι η ορθογώνια περιοχή στο παράθυρο επίδειξης, στην οποία απεικονίζεται η τελική εικόνα, όπως φαίνεται και στην **Εικόνα 5.5**:

Οι παράμετροι *x* και *y* καθορίζουν τη κάτω αριστερή γωνία της οπτικής οδού, και οι *width* και *height* το πλάτος και το ύψος της αντίστοιχα. Εάν η οπτική οδός δεν είναι σαφώς ορισμένη, καταλαμβάνει, ως προεπιλογή, ολόκληρο το παράθυρο απεικόνισης. Αυτό σημαίνει ότι εάν η αναλογία διαστάσεων του παραθύρου δεν ταιριάζει με αυτήν, που έχει καθοριστεί στον ορισμό του όγκου ορθοκανονικής ή προοπτικής προβολής (π.χ. μετά από αλλαγή σχήματος ή θέσης του παραθύρου απεικόνισης), η επιδειχθείσα εικόνα θα εμφανιστεί διαστρεβλωμένη. Επίσης, είναι δυνατό να σχεδιαστούν διάφορες χωριστές εικόνες μέσα σε ένα μόνο παράθυρο [20].



Εικόνα 5.5: Καθορισμός της οπτικής οδού σε ένα παράθυρο απεικόνισης [20].

5.5 Αντιστροφή του καναλιού μεταφοράς της οπτικής οδού

Μερικές φορές είναι επιθυμητό ο χρήστης να επιλέγει ένα σημείο (εικονοκύτταρο ή *pixel*) του παραθύρου απεικόνισης και να γνωρίζει σε ποιο σημείο (σε συντεταγμένες του αρχικού αντικειμένου) αντιστοιχεί. Κάτι τέτοιο επιτυγχάνεται με την αντιστροφή της οπτικής οδού, της προβολής και των μετασχηματισμών μοντέλου.

Δεδομένου ενός σημείου P_o του αντικειμένου, το αντίστοιχο εικονοκύτταρο του P_p δίνεται από τη σχέση:

$$P_p = M_{viewport} \cdot M_{projection} \cdot M_{modelview} \cdot P_o \quad (5-2)$$

Έτσι, εάν είναι γνωστό το P_p , το P_o μπορεί να προκύψει με την εφαρμογή του αντίστροφου του κάθε ενός από τους προηγούμενους μετασχηματισμούς:

$$P_o = M_{modelview}^{-1} \cdot M_{projection}^{-1} \cdot M_{viewport}^{-1} \cdot P_p \quad (5-3)$$

Σε αυτή όμως τη διαδικασία υπάρχει ένα πρόβλημα. Λόγω του ότι η θέση ενός εικονοκυττάρου της οθόνης ορίζεται στις δύο διαστάσεις, ενώ οι συντεταγμένες του αρχικού αντικειμένου στις τρεις, όλα τα σημεία που ορίζονται από διάνυσμα σε συντεταγμένες αντικειμένου μπορούν να προβληθούν στην ίδια θέση του δισδιάστατου επιπέδου της οθόνης. Αυτό σημαίνει ότι δεν είναι δυνατό να εκτελεσθεί μία σαφής αντιστροφή της προβολής από συντεταγμένες οθόνης σε παγκόσμιες συντεταγμένες. Έτσι, η εφαρμογή πρέπει να δίνει τη δυνατότητα να επιλέγεται μία τιμή z-συντεταγμένης για το εικονοκύτταρο, το οποίο βρίσκεται μεταξύ του κοντινού και του μακρινού επιπέδου αποκοπής [20].

ΛΟΓΙΣΜΙΚΟ MeshViewer v1.0

6.1 Εισαγωγή

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός γραφικού περιβάλλοντος με τη βιβλιοθήκη της OpenGL, για την απεικόνιση και διαχείριση τρισδιάστατων πλεγμάτων και επιφανειών. Πιο συγκεκριμένα, αναπτύχθηκε λογισμικό στη γλώσσα προγραμματισμού Microsoft Visual C++ 6.0, αλλά χρησιμοποιήθηκε η σύνταξη της C, προκειμένου να επιτευχθεί καλύτερη συμβατότητα με οποιοδήποτε λειτουργικό σύστημα. Για τον προηγούμενο λόγο άλλωστε επιλέχθηκε η βιβλιοθήκη γραφικών OpenGL, καθώς και η βιβλιοθήκη GLUT για την παραθυρική διαχείριση.

Το λογισμικό που αναπτύχθηκε διαχειρίζεται τρεις τύπους αρχείων:

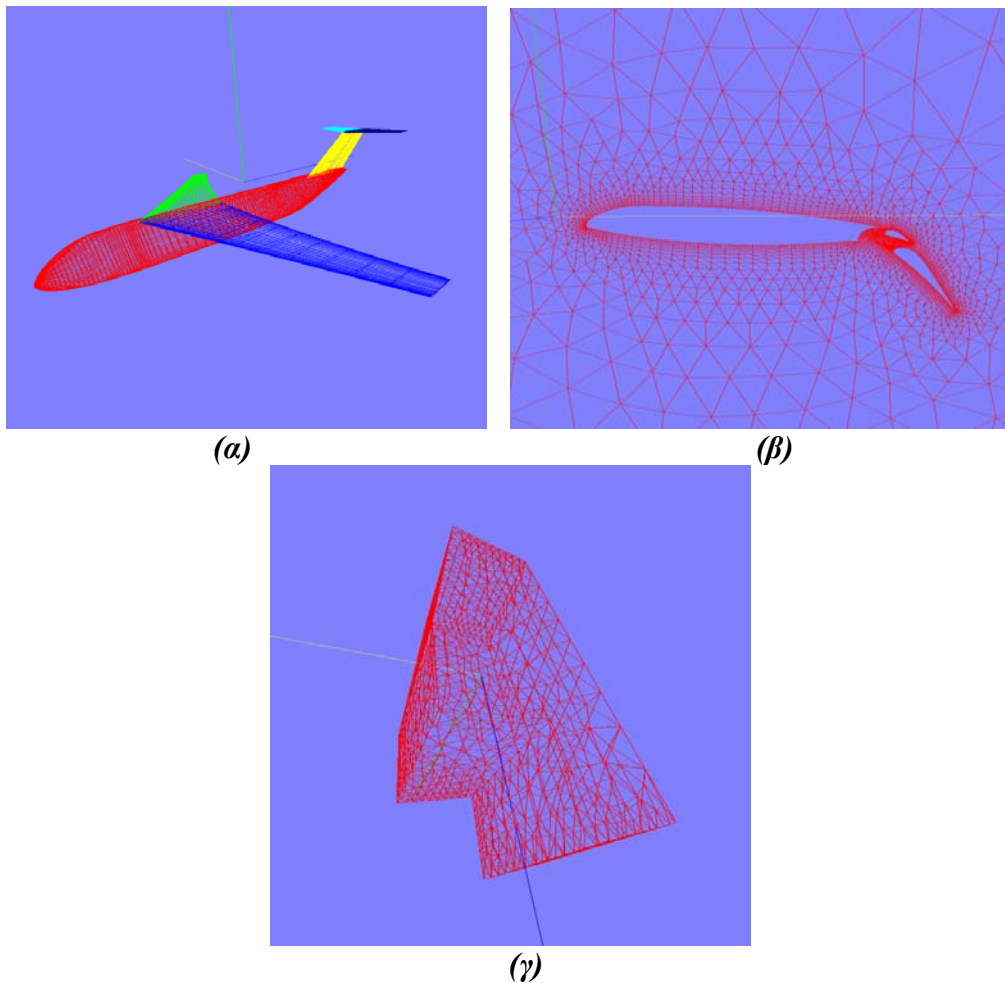
- αρχεία .DPL, τα οποία είναι αρχεία εξόδου του λογισμικού Delaundo v5.4. [Jens-Dominic Müller, University of Michigan, 1997].
- αρχεία .DPL τρισδιάστατου χώρου, τα οποία είναι αρχεία εξόδου του λογισμικού Mesh v3.5. [12].
- αρχεία .DAT, τα οποία είναι αρχεία εξόδου του λογισμικού Ge.P.A.S. v1.0. [18].

Η βασική λειτουργία του λογισμικού είναι να απεικονίζει γραφικά τα δεδομένα των παραπάνω τύπων αρχείων και να δίνει στο χρήστη την ικανότητα να μετασχηματίζει στο χώρο τα αντικείμενα, που έχουν απεικονιστεί. Τα αντικείμενα αυτά, ανάλογα με τον τύπο του αρχείου εισόδου, μπορεί να είναι είτε δισδιάστατα είτε τρισδιάστατα τριγωνικά πλέγματα, είτε βασικά σχέδια αεροσκαφών, όπως φαίνεται και στην **Εικόνα 6.1**.

6.2 Διάρθρωση του κώδικα

Η γενική δομή του κώδικα είναι η εξής: αρχικά δίνεται ο τύπος και το όνομα του αρχείου εισόδου. Στη συνέχεια καλείται η αντίστοιχη συνάρτηση, που διαβάζει το αρχείο και αποθηκεύει τα δεδομένα του σε κατάλληλες μεταβλητές. Ακολούθως, καλείται η συνάρτηση, που σχεδιάζει τα πλέγματα ή τις επιφάνειες, ανάλογα με τον τύπο του αρχείου, που έχει επιλεγεί και διαβαστεί. Στη συνέχεια, το αντικείμενο σχεδιάζεται αρχικά με μία γενική όψη, αλλά υπάρχουν και τέσσερις ακόμη βασικές όψεις, που μπορεί να επιλέξει ο χρήστης. Ο χρήστης έχει τη δυνατότητα να περιστρέψει, να μετακινήσει, να αλλάξει κλίμακα, ή να εστιάσει σε διαφορετικές αποστάσεις (zoom) το αντικείμενο, με κλήση των ανάλογων συναρτήσεων. Επίσης, υπάρχει η επιλογή το αντικείμενο να σχεδιαστεί είτε σε *συμπαγή μορφή (solid)* είτε σε *μορφή τύπου σύρματος (wire-frame)*. Κάθε απεικόνιση μπορεί να αποθηκευτεί ως αρχείο εικόνας τύπου .TIFF, με δυνατότητα τριών ειδών συμπίεσης. Τέλος, για την επιλογή των παραπάνω λειτουργιών έχει διαμορφωθεί μενού επιλογών, που εμφανίζεται ανά πάσα στιγμή με το πάτημα του δεξιού πλήκτρου του ποντικιού. Οι συναρτήσεις των λειτουργιών, που εμφανίζονται στο μενού επιλογών, καλούνται

όποτε ο χρήστης ενεργοποιήσει τη λειτουργία που επιθυμεί κατά την εκτέλεση του προγράμματος.



Εικόνα 6.1: Παράδειγματα αντικειμένων, που διαχειρίζεται το λογισμικό *MeshViewer v1.0*, ανάλογα με το αρχείο εισόδου.

Στη συνέχεια, θα παρουσιαστούν οι βασικές συναρτήσεις του κώδικα, ο τρόπος που επικοινωνούν μεταξύ τους και κυρίως οι συναρτήσεις της βιβλιοθήκης OpenGL που χρησιμοποιούνται.

6.2.1 Ανάγνωση αρχείων

Η ανάγνωση των τριών διαφορετικών τύπων αρχείων γίνεται με τις συναρτήσεις `openDPLFile()`, `open3D_DPLFile()`, και `openDATFile()`, ανάλογα με τον τύπο του αρχείου.

6.2.1.1 `openDPLFile()`

Η συνάρτηση `openDPLFile()` χρησιμοποιείται για την ανάγνωση και αποθήκευση των απαραίτητων μεταβλητών ενός `.DPL` αρχείου. Η μορφή αρχείων

.DPL είναι η τυποποιημένη μορφή εξόδου του λογισμικού Delaundo v5.4, που είναι ένα συμπαγές αρχείο ASCII.

Η δομή του είναι η εξής: αρχικά εμφανίζεται μία επικεφαλίδα με το τίτλο "unstructured grid data by Delaundo 5.4". Ακολουθούν τρεις αριθμοί, όπου ο πρώτος είναι ο αριθμός των κελιών και οι άλλες δύο θέσεις ακεραίων χρησιμοποιούνται για να κρατήσουν, για παράδειγμα, τον αριθμό των προηγούμενων επαναλήψεων και την μεταβλητή ελέγχου της σύγκλησης. Στη συνέχεια, για κάθε κελί αναγράφεται ο αριθμός των γωνιών που το αποτελούν, οι κόμβοι της κάθε γωνίας με αριστερόστροφη σειρά, τα γειτονικά κελιά με το πρώτο γειτονικό να είναι απέναντι από τον πρώτο κόμβο με τη σειρά που αναγράφονται, δηλαδή το πρώτο γειτονικό κελί και το τρέχον κελί μοιράζονται τον δεύτερο και τρίτο κόμβο, και τέλος ένας τρέχοντας μετρητής. Έπειτα, αναφέρεται ο αριθμός των κόμβων και στη συνέχεια, τέσσερις αναφορικές, ελεύθερης κατάστασης, ποσότητες και δύο θέσεις αριθμών κινητής υποδιαστολής. Συνεχίζοντας, εμφανίζονται για κάθε κόμβο, οι συντεταγμένες x και y , τέσσερις ποσότητες κατάστασης, και ένας τρέχοντας μετρητής. Ακολουθούν, ο αριθμός των τμημάτων του ορίου, ο αριθμός των γωνιών μεταξύ των τμημάτων του ορίου, οι αριθμοί κόμβων σε κάθε γωνία και το όνομα του αριστερά προσανατολισμένου ορίου για κάθε γωνία, δηλαδή το όνομα του ορίου που προσανατολίζεται με το χωρίο στα αριστερά, και έχει αυτούς τους κόμβους ως πρώτους. Έπειτα, αναγράφεται, για κάθε τμήμα, ο αριθμός των κόμβων κατά μήκος του τμήματος και το όνομα του τμήματος, και στη συνέχεια, για κάθε πλευρά του ορίου, οι δύο κόμβοι διαμόρφωσης, ο εσωτερικός γειτονικός κόμβος και ένας μετρητής. Σημειώνεται ότι ο προσανατολισμός οποιασδήποτε πλευράς αλλάζει κατά την παρατήρησή της από το άλλο κελί που την μοιράζεται. Για τα κελιά του ορίου αυτό σημαίνει ότι διασχίζοντας το πρώτο εμφανιζόμενο στη λίστα κόμβο ενός κελιού προς το δεύτερο, το χωρίο είναι προς τα δεξιά. Τέλος, υπάρχει ένας δείκτης ότι δεν υπάρχουν εξωτερικές πλευρές του ορίου και ένα ψευδές όνομα ορίου. Ακολουθεί ένα παράδειγμα της μορφής ενός .DPL αρχείου:

unstructured grid data by Delaundo 5.4

```

8 0 1
3 9 4 5 13 6 4 1
3 9 1 2 10 4 5 2
3 4 2 3 11 12 4 3
3 9 2 4 3 1 2 4
3 9 8 1 9 2 8 5
3 9 5 7 7 8 1 6
3 7 5 6 14 15 6 7
3 9 7 8 16 5 6 8
9
1. 1. 1. 1. 0. 0.
0.3000000E+01 0.0000000E+00 1. 1. 1. 1. 1
0.3000000E+01 0.2000000E+01 1. 1. 1. 1. 2
0.3000000E+01 0.3000000E+01 1. 1. 1. 1. 3
0.2000000E+01 0.3000000E+01 1. 1. 1. 1. 4
0.1000000E+01 0.3000000E+01 1. 1. 1. 1. 5
0.0000000E+00 0.3000000E+01 1. 1. 1. 1. 6
0.0000000E+00 0.2000000E+01 1. 1. 1. 1. 7
0.0000000E+00 0.0000000E+00 1. 1. 1. 1. 8
0.1267948E+01 0.1000000E+01 1. 1. 1. 1. 9
4 4 1 3 6 8 2 4 3 1
1 2

```

```

1 8 5 9
2 4
2 1 2 10
3 2 3 11
3 3
4 3 3 12
5 4 1 13
6 5 7 14
2 1
7 6 7 15
8 7 8 16
0 -1

```

Η συνάρτηση `openDPLFile()` αρχικά ζητάει από τον χρήστη την τοποθεσία και το όνομα του αρχείου και στην συνέχεια το ανοίγει προς ανάγνωση των επιθυμητών δεδομένων. Για την απεικόνιση του πλέγματος, από την πληθώρα δεδομένων που παρέχει το αρχείο, χρειάζονται ο αριθμός των κελιών του πλέγματος, από ποιους κόμβους αποτελείται το κάθε κελί, ο αριθμός των κόμβων του πλέγματος, και τέλος οι x και y συντεταγμένες του κάθε κόμβου. Τα δεδομένα αυτά αποθηκεύονται στις κατάλληλες μεταβλητές και στη συνέχεια η συνάρτηση υπολογίζει και απεικονίζει στην οθόνη τις συντεταγμένες του κέντρου μάζας του πλέγματος.

6.2.1.2 open3D_DPLFile()

Η συνάρτηση `open3D_DPLFile()` χρησιμοποιείται για την ανάγνωση και αποθήκευση των απαραίτητων μεταβλητών ενός .DPL αρχείου κατασκευασμένο για απεικόνιση στον τρισδιάστατο χώρο. Η μορφή αρχείων .DPL τριών διαστάσεων του λογισμικού Mesh v3.5 είναι παρόμοια με τη τυποποιημένη μορφή εξόδου του λογισμικού Delaundo v5.4. Ακολουθεί ένα παράδειγμα της μορφής ενός .DPL αρχείου τρισδιάστατου χώρου:

Unstructured grid by Mesh v3.5.

```

166
3 69 27 28 194 64 113 1
3 54 23 24 190 119 34 2
.....
3 83 41 82 157 156 166 165
3 83 67 41 134 165 40 166
104
0.4933333E+01 0.0000000E+00 0.1080000E+01
0.8266667E+01 0.0000000E+00 0.4000000E-01
.....
0.1399667E+02 0.1000000E+02 0.2965000E+00
0.9149945E+01 0.1000000E+02 0.1069204E+01

```

Η συνάρτηση `open3D_DPLFile()` αρχικά ζητάει από το χρήστη την τοποθεσία και το όνομα του αρχείου και στην συνέχεια το ανοίγει προς ανάγνωση των επιθυμητών δεδομένων. Για την απεικόνιση του τρισδιάστατου πλέγματος χρησιμοποιείται ο αριθμός των κελιών του πλέγματος, από ποιους κόμβους αποτελείται το κάθε κελί, ο αριθμός των κόμβων του πλέγματος, και τέλος οι x , y , και z συντεταγμένες του κάθε κόμβου. Τα δεδομένα αυτά αποθηκεύονται στις

κατάλληλες μεταβλητές και στη συνέχεια η συνάρτηση υπολογίζει και απεικονίζει στην οθόνη τις συντεταγμένες του κέντρου μάζας του πλέγματος.

6.2.1.3 openDATFile()

Η συνάρτηση openDATFile() χρησιμοποιείται για την ανάγνωση και αποθήκευση των απαραίτητων μεταβλητών ενός .DAT αρχείου. Η μορφή αρχείων .DAT είναι μία από τις τυποποιημένες μορφές εξόδου του λογισμικού Ge.P.A.S. v1.0, και είναι ένα συμπαγές αρχείο ASCII, που περιλαμβάνει τα απαραίτητα δεδομένα για τη σχεδίαση ενός αεροσκάφους, όπως προκύπτουν από τον καθορισμό των παραμέτρων του μέσω του λογισμικού Ge.P.A.S. v1.0 [18].

Η δομή του αρχείου προκύπτει από το τρόπο με τον οποίο σχεδιάζεται το αεροσκάφος, μέσω του λογισμικού Ge.P.A.S. v1.0. Έτσι, το αεροσκάφος χωρίζεται σε ζώνες, μερικές από τις οποίες είναι η άτρακτος, το αριστερό και το δεξί πτερύγιο, το ουραίο κάθετο πτερύγιο, το αριστερό και το δεξί οριζόντιο ουραίο πτερύγιο. Η κάθε ζώνη με τη σειρά της σχεδιάζεται με βάση το σχήμα των διατομών κατά μήκος της, ενώ η κάθε διατομή από τα σημεία που την αποτελούν.

Η δομή του αρχείου είναι η εξής: ανάλογα με τον αριθμό των ζωνών που αποτελείται το σχέδιο του αεροσκάφους, το .DAT αρχείο είναι χωρισμένο σε τμήματα όμοια σε δομή. Το κάθε τμήμα του αρχείου εμφανίζει αρχικά τη λέξη "ZONE" και στη συνέχεια τα ονόματα των μεταβλητών I, J, και K της κάθε ζώνης ακολουθούμενα από τη τιμή τους. Ακολουθούν, οι x, y, και z συντεταγμένες του κάθε I σημείου της κάθε J διατομής και τρεις τρέχοντες μετρητές με σειρά I, J, K. Ακολουθεί ένα παράδειγμα της μορφής ενός .DAT αρχείου:

```

ZONE I= 41 J= 81 K=1
0      0      0      1 1 1
0.19   -1.23913043478261E-02    0.39      2 1 1
0.36   -2.34782608695652E-02    0.76      3 1 1
.....
0.551999999999998    -0.023478260869564    9.24000000000001    39 81 1
0.432999999999997    -1.23913043478253E-02    9.61000000000001    40 81 1
0.3      0      10      41 81 1
ZONE I= 11 J= 61 K=1
0      -0.034      8.06      1 1 1
0.4     -0.03094    8.26130143703768      2 1 1
0.8     -0.02788    8.46260287407537      3 1 1
.....
3.2     -9.5199999999999E-03    9.67041149630147      9 61 1
3.6     -6.45999999999993E-03    9.87171293333915     10 61 1
4       -3.39999999999997E-03    10.0730143703768     11 61 1
ZONE I= 11 J= 61 K=1
0      -0.034      8.06      1 1 1
-0.4    -0.03094    8.26130143703768      2 1 1
-0.8    -0.02788    8.46260287407537      3 1 1
.....
-3.2    -9.5199999999999E-03    9.67041149630147      9 61 1
-3.6    -6.45999999999993E-03    9.87171293333915     10 61 1
-4      -3.39999999999997E-03    10.0730143703768     11 61 1

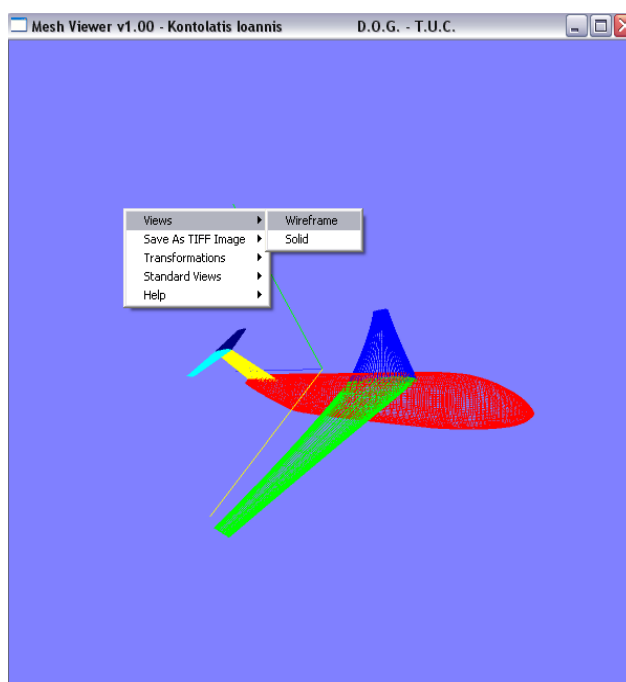
```

Η συνάρτηση openDATFile() αρχικά ζητάει από το χρήστη τη τοποθεσία και το όνομα του αρχείου και στην συνέχεια το ανοίγει προς ανάγνωση των επιθυμητών δεδομένων. Για την απεικόνιση του σχεδίου του αεροσκάφους, η συνάρτηση διαβάζει από το .DAT αρχείο και αποθηκεύει στις κατάλληλες μεταβλητές τις τιμές I, J, και K

κάθε ζώνης, καθώς και τις x , y , και z συντεταγμένες του κάθε I σημείου της κάθε J διατομής. Τέλος, η συνάρτηση υπολογίζει και απεικονίζει στην οθόνη τις συντεταγμένες του κέντρου μάζας του αεροσκάφους.

6.2.2 Σχεδιασμός πλεγμάτων και επιφανειών – Συνάρτηση designMesh()

Για την απεικόνιση των πλεγμάτων, δισδιάστατων ή τρισδιάστατων, και των σχεδίων των αεροσκαφών αναπτύχθηκε η συνάρτηση designMesh(). Η συνάρτηση περιλαμβάνει εντολές για το σχεδιασμό του αντικειμένου είτε σε συμπαγή μορφή (solid) είτε σε μορφή τύπου σύρματος (wire-frame), όπως παρουσιάζεται και στο μενού επιλογών μορφής αντικειμένου του λογισμικού στην **Εικόνα 6.2**.



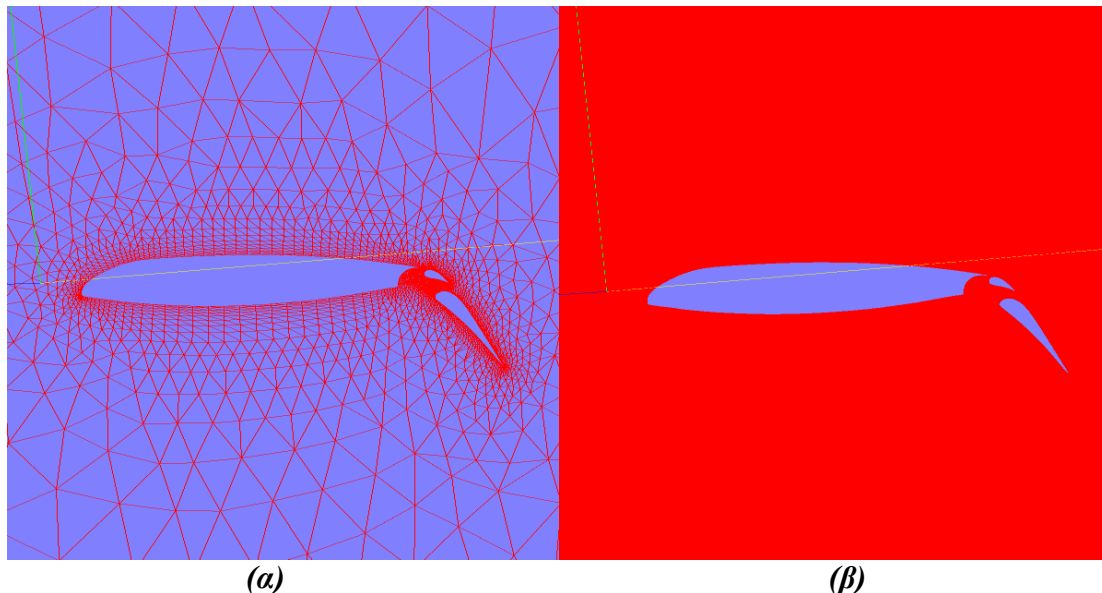
Εικόνα 6.2: Μενού επιλογών μορφής αντικειμένου του λογισμικού *MeshViewer v1.0*.

Εάν ο χρήστης επιλέξει το σχεδιασμό ενός δισδιάστατου πλέγματος, τότε χρησιμοποιείται η συνάρτηση glBegin() της OpenGL με όρισμα GL_LINE_LOOP, για την κατασκευή διαδοχικών ευθύγραμμων τμημάτων κλειστού χωρίου, που ορίζονται από κορυφές. Για τη συγκεκριμένη εφαρμογή, οι κορυφές είναι οι κόμβοι κάθε τριγωνικού κελιού του πλέγματος και, για τον καθορισμό της θέσης τους στο επίπεδο, χρειάζονται οι συντεταγμένες που αποθηκεύτηκαν κατά την ανάγνωση του αντίστοιχου αρχείου. Ο βρόγχος σχεδιασμού ενός τριγωνικού κελιού κλείνει με τη συνάρτηση glEnd(), και οι γραμμές κώδικα που εκτελούν τη παραπάνω λειτουργία είναι οι εξής:

```
glBegin(GL_LINE_LOOP);
glVertex3f(x1, y1, 0);
glVertex3f(x2, y2, 0);
glVertex3f(x3, y3, 0);
glEnd();
```

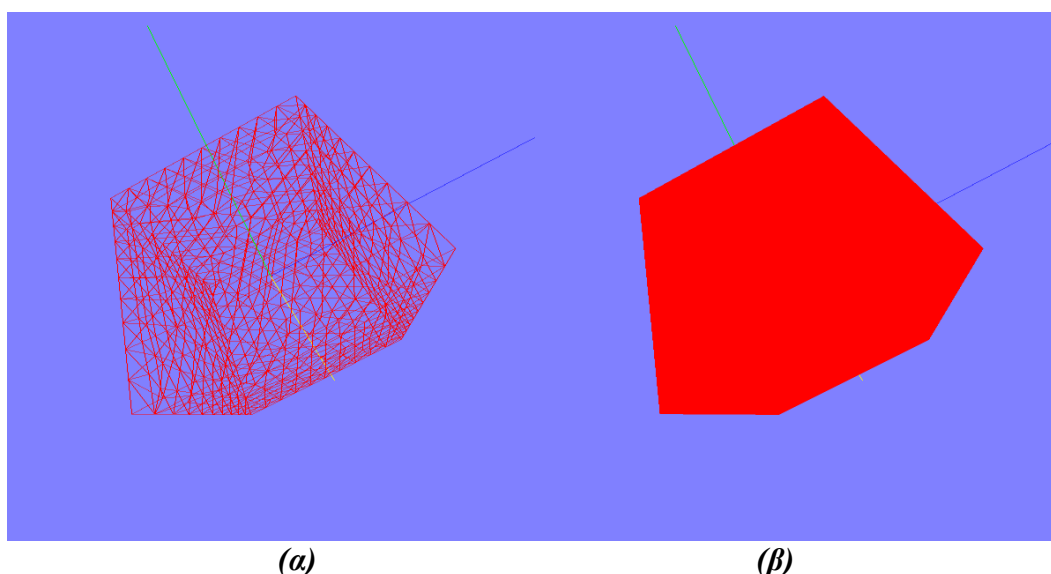
Με την προηγούμενη λειτουργία, κατασκευάζεται μία επιφάνεια με πλέγμα ή σε μορφή τύπου σύρματος, όπως παρουσιάζεται στην **Εικόνα 6.3(α)**. Εάν είναι

επιθυμητό η μορφή να είναι συμπαγής (**Εικόνα 6.3(β)**), τότε το όρισμα της συνάρτησης `glBegin()` αλλάζει σε `GL_TRIANGLE_STRIP` και, μετά τον ορισμό της τρίτης κορυφής, επαναλαμβάνεται η πρώτη για να κλείσει ο κύκλος. Η συνάρτηση `glVertex3f(x, y, z)` στη γενική της μορφή ορίζει κορυφή (vertex) με τρεις συντεταγμένες (3), κινητής υποδιαστολής (f).



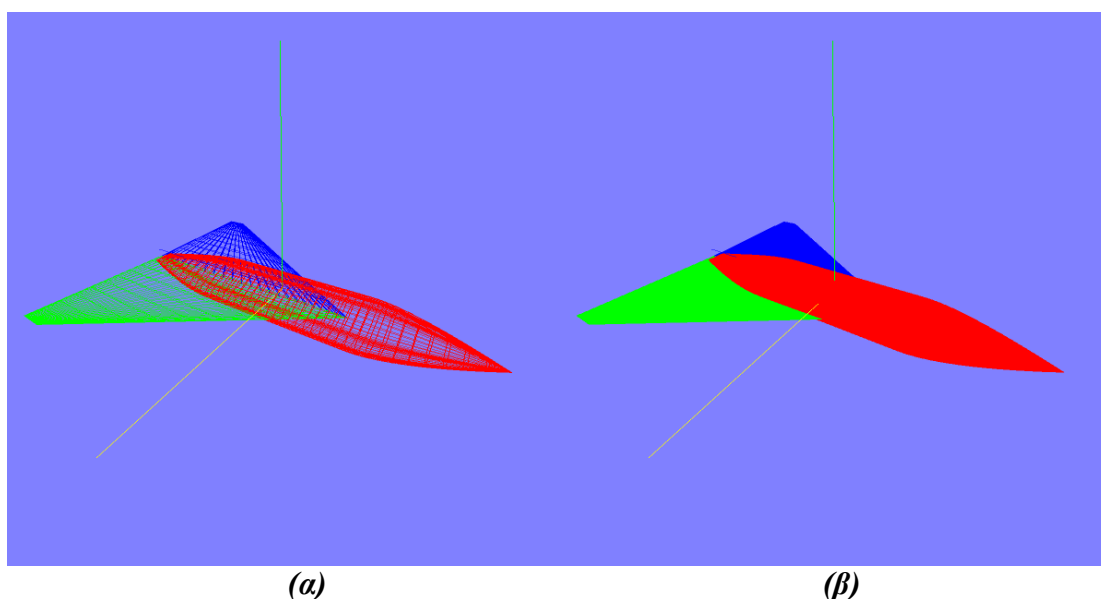
Εικόνα 6.3: Απεικόνιση διατομής πτέρυγας είτε (α) σε μορφή πλέγματος είτε (β) σε συμπαγή μορφή.

Στη περίπτωση που ο χρήστης επιλέξει την απεικόνιση ενός τρισδιάστατου πλέγματος (**Εικόνα 6.4(α)**), η συνάρτηση είναι όμοια με προηγουμένως, απλά στον ορισμό των κορυφών η τιμή της συντεταγμένης z μπορεί να είναι και διάφορη του μηδενός, σύμφωνα με τις τιμές που έχουν αποθηκευτεί κατά την ανάγνωση του αντίστοιχου αρχείου. Αν επιπλέον ο χρήστης επιλέξει το αντικείμενο να εμφανιστεί σε συμπαγή μορφή (**Εικόνα 6.4(β)**), ισχύουν τα παραπάνω.



Εικόνα 6.4: Απεικόνιση κύβου είτε (α) σε μορφή τρισδιάστατου πλέγματος είτε (β) σε συμπαγή μορφή.

Τέλος, εάν ο χρήστης επιλέξει την απεικόνιση του σχεδίου ενός αεροσκάφους σε μορφή τύπου σύρματος (*Εικόνα 6.5(α)*), τότε στη συνάρτηση `glBegin()` χρησιμοποιείται σαν όρισμα το `GL_LINE_STRIP` και οι κορυφές των ευθύγραμμων τμημάτων είναι κάθε σημείο *I* της *J* διατομής, δηλαδή δημιουργείται ένας βρόγχος με σκοπό τον σχεδιασμό της κάθε διατομής από τα ευθύγραμμα τμήματα που ενώνουν τα σημεία που την αποτελούν. Οι συντεταγμένες των τελευταίων έχουν αποθηκευτεί στις κατάλληλες μεταβλητές κατά την ανάγνωση του αντίστοιχου αρχείου. Η διαδικασία αυτή επαναλαμβάνεται για κάθε ζώνη, μέχρι την ολοκλήρωση του σχεδίου. Εάν επιλεγεί το αεροσκάφος να εμφανιστεί σε συμπαγή μορφή (*Εικόνα 6.5(β)*), τότε το όρισμα της `glBegin()` είναι `GL_QUADS` και οι κορυφές ορίζονται ομοίως.



Εικόνα 6.5: Απεικόνιση σχεδίου αεροσκάφους είτε (α) σε μορφή τύπου σύρματος είτε (β) σε συμπαγή μορφή.

6.2.3 Σχεδιασμός αξόνων συστήματος συντεταγμένων – Συνάρτηση `designAxis()`

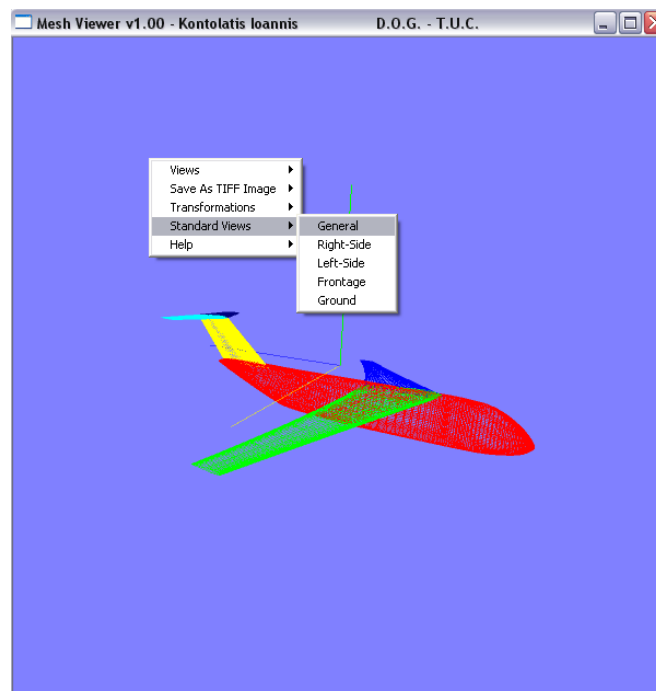
Για τη διευκόλυνση του χρήστη, ταυτόχρονα με την απεικόνιση οποιουδήποτε αντικειμένου, σχεδιάζονται και οι άξονες ενός Καρτεσιανού συστήματος συντεταγμένων με κέντρο το κέντρο μάζας του αντικειμένου και μοναδιαίο μήκος, όπως φαίνεται και στην *Εικόνα 6.5*. Οποιοσδήποτε μετασχηματισμός εφαρμόζεται στον αντικείμενο, εφαρμόζεται και στο σύστημα των αξόνων. Ο x-άξονας απεικονίζεται με κίτρινο χρώμα, ο y-άξονας με πράσινο και ο z-άξονας με μπλε.

Για το σχεδιασμό των αξόνων χρησιμοποιείται η συνάρτηση `glBegin()` με όρισμα `GL_LINES`, που σχεδιάζει ευθύγραμμα τμήματα ανοικτού χωρίου με άκρα τις δηλωμένες κορυφές.

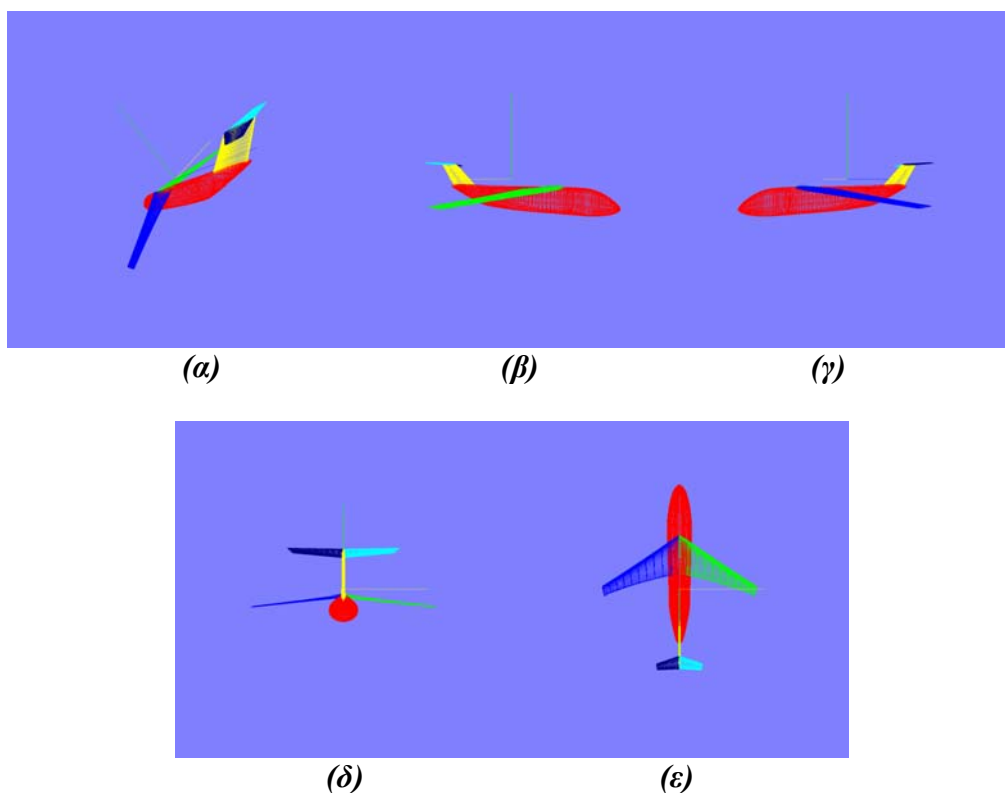
6.2.4 Σχεδιασμός των 5 βασικών όψεων

Ο αρχικός σχεδιασμός οποιουδήποτε αντικειμένου γίνεται με γενική οπτική, όπως φαίνεται και στην *Εικόνα 6.7(α)*. Στη συνέχεια, ο χρήστης έχει τη δυνατότητα να χρησιμοποιήσει οποιαδήποτε από τις τέσσερις βασικές όψεις, που του παρέχονται από το μενού επιλογών με τίτλο "Standard Views", οι οποίες είναι η δεξιά και αριστερή πλάγια όψη, πρόσοψη και κάτοψη, ή ακόμη να επιστρέψει στη γενική, όπως

εμφανίζεται και στην **Εικόνα 6.6**. Στις **Εικόνες 6.7(β)-(ε)** παρουσιάζονται παραδείγματα των όψεων για το σχέδιο ενός αεροσκάφους.



Εικόνα 6.6: Μενού επιλογών τυποποιημένων όψεων του λογισμικού *MeshViewer* v1.0.



Εικόνα 6.7: Παραδείγματα βασικών όψεων που παρέχει το λογισμικό *MeshViewer* v1.0. (α)Γενική, (β)Δεξιά πλάγια όψη, (γ)Αριστερή πλάγια όψη, (δ)Πρόσοψη, και (ε)Κάτοψη.

Για το σχεδιασμό των όψεων αναπτύχθηκαν οι 5 συναρτήσεις `drawGeneral()`, `drawRight()`, `drawLeft()`, `drawFront()`, και `drawGround()`, οι οποίες στην ουσία μετασχηματίζουν τις αρχικές συντεταγμένες των αντικειμένων, όπως αυτές έχουν αποθηκευτεί από το αντίστοιχο αρχείο, ώστε το κάθε αντικείμενο να σχεδιαστεί ξανά, σύμφωνα με την επιθυμητή όψη.

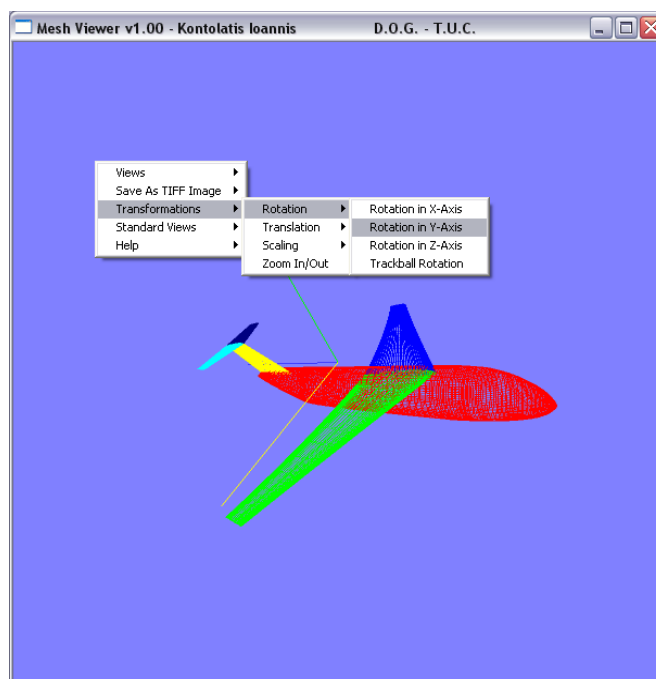
Για το σκοπό αυτό έχουν οριστεί 5 δομές με τη μορφή πίνακα 8 στοιχείων (`generalprims[8]`, `rightprims[8]`, `leftprims[8]`, `frontprims[8]`, `groundprims[8]`), που η κάθε μία από αυτές περιλαμβάνει τη κλήση των συναρτήσεων `designAxis()` και `designMesh()`, και τους πίνακες μετασχηματισμών ανάλογα με την επιλεγμένη όψη για το κάθε αντικείμενο που αυτές σχεδιάζουν. Ακολουθεί ένα παράδειγμα της συνάρτησης `drawLeft()` και της δομής `leftprims[8]`:

```
/* structure for design the left view. */
struct primitive leftprims[8] = {
    {
        designAxis,
        {
            0, 0, 0,
            0, 1, 0, 1,
            5, 5, 5
        }
    },
    {
        designMesh,
        {
            0, 0, 0,
            0, 1, 0, 1,
            1, 1, 1
        }
    }
};

void drawLeft(int i)
{
    struct primitive *l = &leftprims[i];
    struct transformation *xform = &l->xform;
    glPushMatrix();
    drawXform(xform, TRUE);
    l->draw();
    glPopMatrix();
}
```

6.2.5 Συναρτήσεις περιστροφής

Από τη στιγμή που πραγματοποιηθεί η απεικόνιση του αντικειμένου, το λογισμικό MeshViewer v1.0 παρέχει στο χρήστη τη δυνατότητα να περιστρέψει το αντικείμενο είτε γύρω από τον καθένα από τους τρεις άξονες του συστήματος συντεταγμένων είτε γύρω από τους άξονες x και y ταυτόχρονα. Με την τελευταία επιλογή, το αντικείμενο θεωρείται ότι βρίσκεται μέσα σε μία σφαίρα με κέντρο το κέντρο μάζας του αντικειμένου και περιστρέφεται όπως θα περιστρεφόταν η σφαίρα. Η επιλογή ονομάζεται "Trackball Rotation" και είναι ο προεπιλεγμένος τύπος περιστροφής του λογισμικού, ενώ οι υπόλοιποι τρεις τύποι περιστροφής ονομάζονται «Rotation in X-Axis», «Rotation in Y-Axis», «Rotation in Z-Axis» αντίστοιχα, όπως παρουσιάζεται και στην **Εικόνα 6.8**.



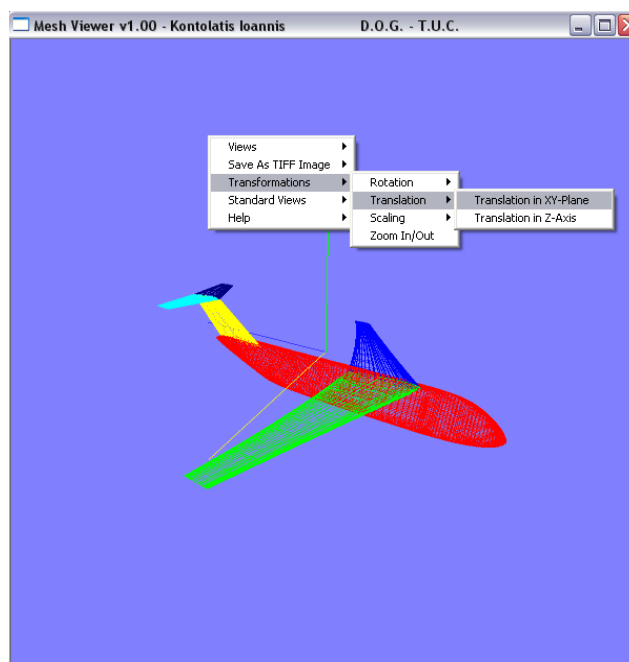
Εικόνα 6.8: Μενού επιλογών περιστροφής του λογισμικού *MeshViewer v1.0*.

Για την πραγματοποίηση των περιστροφών χρησιμοποιείται η συνάρτηση `glRotate*(theta, vx, vy, vz)` της OpenGL, που παράγει τον 4x4 πίνακα περιστροφής (βλ. **Ενότητα 4.6.2**), όπου το διάνυσμα $v = (vx, vy, vz)$ μπορεί να έχει οποιεσδήποτε τιμές αριθμών κινητής υποδιαστολής για τα στοιχεία του. Το διάνυσμα αυτό καθορίζει τον προσανατολισμό ενός άξονα περιστροφής, που περνά από την αρχή των συντεταγμένων. Εάν το διάνυσμα v δεν είναι καθορισμένο ως μοναδιαίο διάνυσμα, τότε κανονικοποιείται αυτόματα, πριν υπολογιστούν τα στοιχεία του πίνακα περιστροφής. Η κωδικοποίηση της κατάληξης μπορεί να είναι f (float) ή d (double - διπλής ακρίβειας), και η παράμετρος $theta$ ορίζεται ως η γωνία περιστροφής σε μοίρες, την οποία η ρουτίνα μετασχηματίζει σε ακτίνια για τους τριγωνομετρικούς υπολογισμούς. Για παράδειγμα, η δήλωση `glRotatef(90.0, 0.0, 0.0, 1.0)` προετοιμάζει ένα πίνακα για περιστροφή 90° γύρω από τον z-άξονα [3].

Για κάθε τύπο περιστροφής γύρω από έναν μόνο άξονα έχει αναπτυχθεί μία συνάρτηση, η οποία δέχεται την τιμή της διαφοράς της y-συντεταγμένης του τελικού μείον του σημείου που βρισκόταν ο δείκτης του ποντικιού όταν πιάστηκε αρχικά το αριστερό πλήκτρο του και ξεκίνησε η διαδικασία. Το αριστερό πλήκτρο πρέπει να παραμένει πατημένο καθ' όλη τη διάρκεια της διαδικασίας, ενώ μόλις αφεθεί το αντικείμενο ακινητοποιείται. Η τιμή αυτή καθορίζει την τιμή της γωνίας περιστροφής του αντικειμένου γύρω από το δεδομένο άξονα στη συνάρτηση `glRotate()`. Η διαφορά της συνάρτησης για τη σφαιρική περιστροφή είναι ότι δέχεται και τη διαφορά της x-συντεταγμένης για την τιμή της γωνίας γύρω από τον άλλον άξονα. Τα ονόματα των συναρτήσεων που αναπτύχθηκαν είναι `rotateX()`, `rotateY()`, `rotateZ()`, και `rotateTrackball()`.

6.2.6 Συναρτήσεις μεταφοράς

Για τη μεταφορά του αντικειμένου έχουν επιλεγεί δύο τύποι. Ο χρήστης μπορεί να επιλέξει μεταφορά του αντικειμένου στο xy-επίπεδο ή μεταφορά κατά μήκος του z-άξονα, όπως παρουσιάζεται και στην **Εικόνα 6.9**.



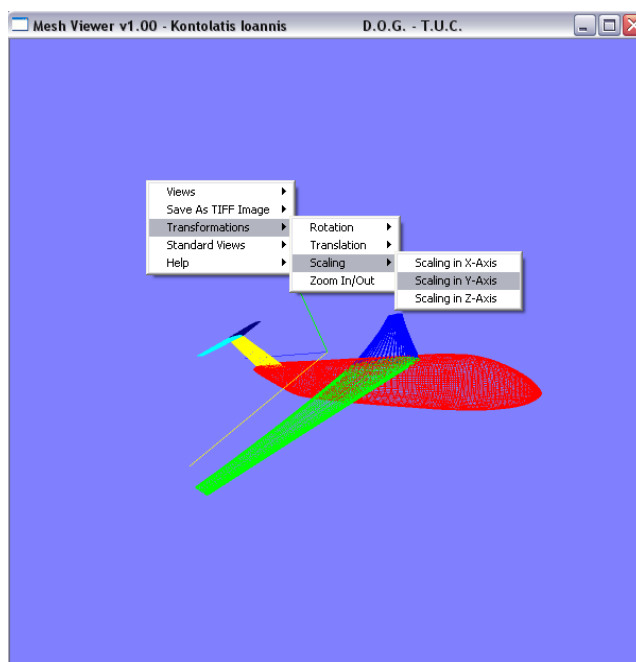
Εικόνα 6.9: Μενού επιλογών μεταφοράς του λογισμικού *MeshViewer v1.0*.

Η συνάρτηση της βιβλιοθήκης OpenGL που χρησιμοποιείται είναι η `glTranslate*(tx, ty, tz)`, η οποία κατασκευάζει τον 4x4 πίνακα μεταφοράς (βλ. **Ενότητα 4.6.1**). Οι παράμετροι μεταφοράς tx , ty , και tz μπορούν να έχουν την τιμή οποιουδήποτε πραγματικού αριθμού, ενώ η κωδικοποίηση της κατάληξης μπορεί να είναι είτε f (float - κινητής υποδιαστολής) είτε d (double - διπλής ακρίβειας). Για δισδιάστατες εφαρμογές, η παράμετρος tz τίθεται ίση με μηδέν (0.0), εξάλλου μία θέση στο δισδιάστατο χώρο αναπαριστάται ως ένα διάνυσμα στήλης τεσσάρων στοιχείων με το στοιχείο z ίσο με μηδέν (0.0). Ο πίνακας μεταφοράς, που δημιουργείται από τη συνάρτηση αυτή, χρησιμοποιείται για το μετασχηματισμό των θέσεων αντικειμένων, που έχουν καθοριστεί πριν κληθεί αυτή. Για παράδειγμα, ένα αντικείμενο μεταφέρεται 25 μονάδες στην x-διεύθυνση και -10 μονάδες στην y-διεύθυνση με τη δήλωση `glTranslatef(25.0, -10.0, 0.0) [3]`.

Κατά τη μεταφορά στο xy-επίπεδο, χρησιμοποιείται η διαφορά της x-συντεταγμένης για τη μετακίνηση του αντικειμένου κατά μήκος του x-άξονα, η διαφορά της y-συντεταγμένης κατά μήκος του y-άξονα, ενώ κατά τη μεταφορά κατά μήκος του z-άξονα χρησιμοποιείται πάλι η y-συντεταγμένη. Η διαφορά των συντεταγμένων ορίζεται, όπως και στη συνάρτηση περιστροφής, από τη τελική και αρχική θέση του δείκτη του ποντικιού, και καθορίζει πόσες μονάδες θα μετακινηθεί το αντικείμενο στην ανάλογη διεύθυνση.

6.2.7 Συναρτήσεις αλλαγής κλίμακας

Το λογισμικό MeshViewer v1.0 παρέχει στον χρήστη τη δυνατότητα να αλλάξει την κλίμακα του αντικειμένου ως προς της τρεις διευθύνσεις των αξόνων του συστήματος συντεταγμένων, όπως φαίνεται και στο μενού επιλογών, που παρουσιάζεται στην **Εικόνα 6.10**.



Εικόνα 6.10: Μενού επιλογών αλλαγής κλίμακας του λογισμικού MeshViewer v1.0.

Για την πραγματοποίηση των αλλαγών κλίμακας στο εκάστοτε αντικείμενο, χρησιμοποιήθηκε η συνάρτηση της OpenGL $glScale*(sx, sy, sz)$, η οποία δημιουργεί ένα 4x4 πίνακα αλλαγής κλίμακας ως προς την αρχή του συστήματος συντεταγμένων (βλ. **Ενότητα 4.6.3**). Η κωδικοποίηση της κατάληξης είναι και σε αυτή την περίπτωση f ή d , ενώ οι παράμετροι αλλαγής κλίμακας μπορούν να έχουν οποιαδήποτε τιμή πραγματικού αριθμού. Για το λόγο αυτό, η συνάρτηση αυτή μπορεί να επιτύχει την παραγωγή ειδώλου όταν οι παράμετροι έχουν αρνητικές τιμές. Για παράδειγμα, η δήλωση $glScalef(2.0, -3.0, 1.0)$ αλλάζει την κλίμακα του αντικειμένου κατά το παράγοντα 2 στη x-διεύθυνση, κατά 3 στη y-διεύθυνση, και δημιουργεί είδωλο ως προς το x-άξονα. Η τιμή μηδέν για οποιαδήποτε παράμετρο αλλαγής κλίμακας μπορεί να προκαλέσει σφάλμα επεξεργασίας (processing error), λόγω του ότι δεν μπορεί να οριστεί ο αντίστροφος πίνακας [3].

Όμοια με τις συναρτήσεις των προηγούμενων μετασχηματισμών, το πόσες μονάδες θα αλλάξει η κλίμακα του αντικειμένου ως προς την ανάλογη διεύθυνση καθορίζεται από τη διαφορά της y-συντεταγμένης της αρχικής και τελικής θέσης του δείκτη του ποντικιού.

6.2.8 Συνάρτηση αλλαγής εστίασης του αντικειμένου (zoom in/out)

Όπως φαίνεται και στην **Εικόνα 6.10**, στο μενού επιλογών των μετασχηματισμών (transformations menu) υπάρχει η δυνατότητα εστίασης του αντικειμένου σε διαφορετικές αποστάσεις (zoom in/out). Για την επίτευξη της συγκεκριμένης λειτουργίας έχει χρησιμοποιηθεί και σε αυτή τη περίπτωση η συνάρτηση αλλαγής κλίμακας $glScalef()$ με ταυτόχρονη αλλαγή της κλίμακας του αντικειμένου κατά την ίδια ποσότητα και στις τρεις διευθύνσεις, σύμφωνα με τη διαφορά της y-συντεταγμένης της αρχικής και τελικής θέσης του δείκτη του ποντικιού.

6.2.9 Συνάρτηση αρχικοποίησης – init()

Για τη σωστή λειτουργία ενός κώδικα με χρήση συναρτήσεων της βιβλιοθήκης OpenGL χρειάζεται η δημιουργία μίας συνάρτησης αρχικοποίησης με συγκεκριμένη δομή, η οποία θα καλείται μέσα από τη κεντρική συνάρτηση του κώδικα, τη `main()`. Η συνάρτηση αυτή συνήθως καλείται `init()`, χωρίς όμως να υπάρχει περιορισμός σε αυτό.

Η συνάρτηση `init()` περιέχει ρουτίνες της OpenGL για τον καθορισμό του χρώματος του φόντου του παράθυρου απεικόνισης, των πινάκων της προβολής που θα χρησιμοποιηθεί και της θέσης της κάμερας, καθώς και άλλων παραμέτρων που απαιτούν αρχικοποίηση και διαφέρουν σε κάθε εφαρμογή. Ακολουθεί ένα παράδειγμα μίας τέτοιας συνάρτησης:

```
void init(void)
{
    glClearColor (0.5, 0.5, 1.0, 0.0);

    glMatrixMode(GL_PROJECTION);
    glFrustum(xwmin, xwmax, ywmin, ywmax, dnear, dfar);

    glMatrixMode(GL_MODELVIEW);
    gluLookAt(0.0, 0.0, 10.0, 0, 0, 0, 1, 0);
}
```

Η συνάρτηση για τον καθορισμό του χρώματος του φόντου είναι η `glClearColor(red, green, blue, alpha)`. Κάθε στοιχείο χρώματος του ορισμού του RGB κώδικα, καθώς και η παράμετρος *alpha*, αποκτούν μία τιμή αριθμού κινητής υποδιαστολής με σύνολο τιμών από 0.0 έως 1.0. Η προεπιλεγμένη τιμή και για τις τέσσερις παραμέτρους είναι η 0.0, η οποία και παράγει το μαύρο χρώμα. Εάν κάθε στοιχείο χρώματος τεθεί ίσο με 1.0, τότε παράγεται το άσπρο χρώμα. Αποχρώσεις του γκριζου προκύπτουν με όμοιες τιμές για τα στοιχεία χρώματος με τιμές μεταξύ 0.0 και 1.0. Η τέταρτη παράμετρος, η *alpha*, παρέχει την επιλογή ανάμειξης του προηγούμενου χρώματος με το τρέχον. Κάτι τέτοιο μπορεί να πραγματοποιηθεί εάν ενεργοποιηθεί το χαρακτηριστικό της ανάμειξης της OpenGL, και αυτό γίνεται μόνο εάν τα χρώματα έχουν οριστεί με το κώδικα RGB [3].

Η συνάρτηση `glFrustum(xwmin, xwmax, ywmin, ywmax, dnear, dfar)` χρησιμοποιείται για τον καθορισμό μίας προοπτικής προβολής η οποία είτε έχει ένα συμμετρικό όγκο οπτικής είτε ένα κεκλιμένο. Όλες οι παράμετροι σε αυτή τη συνάρτηση είναι τιμές αριθμών κινητής υποδιαστολής, διπλής ακρίβειας. Το κοντινό επίπεδο (near plane) είναι το επίπεδο της οπτικής (view plane) και το σημείο αναφοράς της προβολής είναι στη θέση της οπτικής (π.χ. αρχή των αξόνων). Οι τέσσερις πρώτες παράμετροι θέτουν τις συντεταγμένες για το παράθυρο αποκοπής (*clipping window*) στο κοντινό επίπεδο, και οι δύο τελευταίες παράμετροι, οι οποίες πρέπει να είναι θετικοί αριθμοί, καθορίζουν τις αποστάσεις από την αρχή των συντεταγμένων έως το κοντινό και μακρινό επίπεδο αποκοπής, κατά μήκος της διεύθυνσης των αρνητικών τιμών του z_{view} -άξονα (βλ. **Ενότητα 5.3.3**). Οι τοποθεσίες του κοντινού και μακρινού επιπέδου υπολογίζονται ως $z_{near} = -dnear$ και $z_{far} = -dfar$, αντίστοιχα. Το παράθυρο αποκοπής μπορεί να οριστεί οπουδήποτε στο κοντινό επίπεδο. Εάν επιλεγεί το παράθυρο αποκοπής να έχει συντεταγμένες $xw_{min} = -xw_{max}$ και $yw_{min} = -yw_{max}$, τότε προκύπτει ένας συμμετρικός όγκος οπτικής ως προς τον άξονα z_{view} . Πριν από τη συνάρτηση `glFrustum`, προηγείται η `glMatrixMode(GL_PROJECTION)` για το καθορισμό του πίνακα μετασχηματισμών προβολής (*projection – transformation matrix*). Έπειτα, όταν δίνεται μία εντολή

μετασχηματισμού, ο πίνακας που προκύπτει συνδέεται αλυσιδωτά με τον τρέχοντα πίνακα προβολής [3].

Όταν καθορίζονται οι παράμετροι οπτικής στην OpenGL, διαμορφώνεται ένας πίνακας και συνδέεται αλυσιδωτά με τον τρέχοντα πίνακα απεικόνισης του μοντέλου. Συμπερασματικά, ο πίνακας αυτός της οπτικής συνδυάζεται με οποιουσδήποτε γεωμετρικούς μετασχηματισμούς, που έχουν επιπλέον καθοριστεί. Ο σύνθετος αυτός πίνακας στη συνέχεια χρησιμοποιείται για να μετασχηματίσει την περιγραφή του αντικειμένου από παγκόσμιες συντεταγμένες σε συντεταγμένες οπτικής. Η κατάσταση οπτικής μοντέλου καθορίζεται από τη δήλωση `glMatrixMode (GL_MODELVIEW)`.

Οι παράμετροι οπτικής καθορίζονται από την ακόλουθη συνάρτηση της βιβλιοθήκης GLU, η οποία είναι σε αυτή τη βιβλιοθήκη γιατί επικαλείται τις ρουτίνες μεταφοράς και περιστροφής της βασικής βιβλιοθήκης OpenGL, με σύνταξη `gluLookAt (x0, y0, z0, xref, yref, zref, Vx, Vy, Vz)`. Οι τιμές όλων των παραμέτρων σε αυτή τη συνάρτηση τίθενται αριθμοί κινητής υποδιαστολής διπλής ακρίβειας. Η συνάρτηση αυτή προσδιορίζει την αρχή του πλαισίου αναφοράς της οπτικής (viewing reference frame) ως τη θέση σε παγκόσμιες συντεταγμένες $P_0 = (x_0, y_0, z_0)$, τη θέση αναφοράς ως $P_{ref} = (x_{ref}, y_{ref}, z_{ref})$, και το διάνυσμα άνω όψης ως $V = (Vx, Vy, Vz)$. Η θετική διεύθυνση του z_{view} -άξονα για το πλαίσιο οπτικής είναι στη διεύθυνση $N = P_0 - P_{ref}$, και τα μοναδιαία διανύσματα των αξόνων για το πλαίσιο αναφοράς της οπτικής υπολογίζονται από τις εξισώσεις:

$$\begin{aligned} n &= \frac{N}{|N|} = (n_x, n_y, n_z) \\ u &= \frac{V \times n}{|V \times n|} = (u_x, u_y, u_z) \\ v &= n \times u = (v_x, v_y, v_z) \end{aligned} \quad (6-1)$$

Εφόσον η διεύθυνση οπτικής είναι κατά μήκος του z_{view} -άξονα, η θέση αναφοράς P_{ref} αναφέρεται επίσης και ως "look-at point". Συνήθως θεωρείται ότι είναι κάποια θέση στο κέντρο της σκηνής, η οποία μπορεί να χρησιμοποιηθεί ως αναφορά για τον καθορισμό των παραμέτρων προβολής. Η θέση αναφοράς μπορεί να θεωρείται ως το σημείο στο οποίο είναι επιθυμητό να στοχεύει η κάμερα, η οποία είναι τοποθετημένη στην αρχή του συστήματος συντεταγμένων της οπτικής. Ο άνω προσανατολισμός της κάμερας προσδιορίζεται από το διάνυσμα V , το οποίο προσαρμόζεται σε μία διεύθυνση κάθετη στο N (βλ. **Ενότητα 5.2**).

Οι παράμετροι οπτικής, που καθορίζονται με τη συνάρτηση `gluLookAt`, χρησιμοποιούνται για τη διαμόρφωση του πίνακα μετασχηματισμών οπτικής, ο οποίος έχει τη μορφή:

$$M_{wc,vc} = R \cdot T = \begin{bmatrix} u_x & u_y & u_z & -u \cdot P_0 \\ v_x & v_y & v_z & -v \cdot P_0 \\ n_x & n_y & n_z & -n \cdot P_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6-2)$$

όπου,

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{και} \quad T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6-3)$$

οι πίνακες σύνθετης περιστροφής.

Εάν δεν χρησιμοποιηθεί η συνάρτηση `glLookAt`, οι προεπιλεγμένες παράμετροι οπτικής της OpenGL είναι $P_0 = (0, 0, 0)$, $P_{ref} = (0, 0, -1)$, και $V = (0, 1, 0)$. Για αυτές τις προεπιλεγμένες τιμές, το πλαίσιο αναφοράς οπτικής είναι το ίδιο με το παγκόσμιο πλαίσιο, με τη διεύθυνση οπτικής κατά μήκος του αρνητικού τμήματος του z_{world} -άξονα. Σε πολλές εφαρμογές, οι προεπιλεγμένες τιμές των παραμέτρων οπτικής είναι πολύ βολικές [3].

6.2.10 Συνάρτηση επανασχεδιασμού του παράθυρου απεικόνισης –`winReshape()`

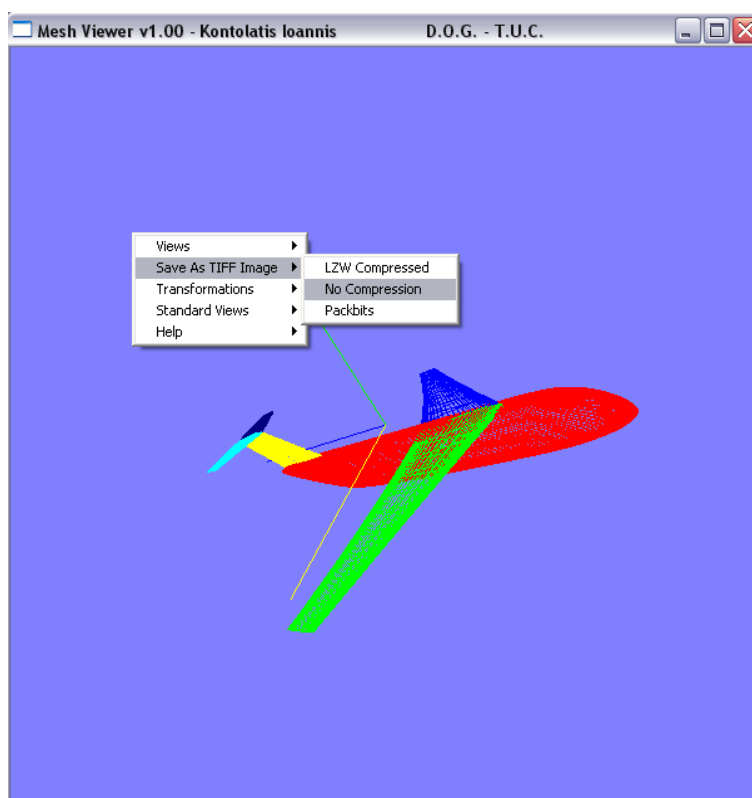
Για να υπάρχει η δυνατότητα ο χρήστης να αλλάζει το μέγεθος και τη θέση του παράθυρου απεικόνισης, καθώς και να το ελαχιστοποιεί/μεγιστοποιεί, θα πρέπει να αναπτυχθεί μία συνάρτηση για το σκοπό αυτό. Η συνάρτηση αυτή, που στη συγκεκριμένη εφαρμογή έχει ονομαστεί `winReshape()`, λειτουργεί ως όρισμα της συνάρτησης `glutReshapeFunc()` που καλείται στη `main()`. Ένα παράδειγμα για την ανάπτυξη της `winReshape()` είναι το ακόλουθο:

```
void winReshape(int width, int height)
{
    glViewport(0, 0, width, height);
    winWidth = width;
    winHeight = height;
    glutPostRedisplay();
}
```

Η συνάρτηση `glViewport(xvmin, yvmin, vpWidth, vpHeight)` χρησιμοποιείται για τον καθορισμό των παραμέτρων της οπτικής οδού, όπου οι τιμές όλων των παραμέτρων δίνονται σε ακέραιες συντεταγμένες οθόνης, σχετικές με το παράθυρο απεικόνισης. Οι παράμετροι `xvmin` και `yvmin` καθορίζουν τη θέση της κάτω αριστερής γωνίας της οπτικής οδού σχετικά με την κάτω αριστερή γωνία του παραθύρου απεικόνισης. Το πλάτος και μήκος σε εικονοστοιχεία (pixels) της οπτικής οδού καθορίζονται με τις παραμέτρους `vpWidth` και `vpHeight`. Εάν δεν συμπεριληφθεί η συνάρτηση `glViewport` στο πρόγραμμα, οι προεπιλεγμένες τιμές για το μέγεθος και τη θέση της οπτικής οδού είναι ίδιες με εκείνες του παράθυρου απεικόνισης. Στη συγκεκριμένη εφαρμογή, η συνάρτηση `winReshape` δέχεται ως όρισμα ένα επιθυμητό πλάτος και ύψος για το παράθυρο απεικόνισης, τα οποία ανατίθενται και ως αρχικές τιμές για το πλάτος και ύψος της οπτικής οδού. Η συνάρτηση `glutPostRedisplay()` της βιβλιοθήκης GLUT χρησιμοποιείται για τη δήλωση ότι τα περιεχόμενα του τρέχοντος παράθυρου απεικόνισης θα πρέπει να ανανεωθούν. Επίσης, η ρουτίνα αυτή χρησιμοποιείται όταν ένα επιπρόσθετο αντικείμενο, όπως ένα ξαφνικά εμφανιζόμενο μενού (pop-up menu) πρόκειται να απεικονιστεί στο παράθυρο [3].

6.2.11 Συνάρτηση αποθήκευσης εικόνας – writetiff()

Το λογισμικό meshViewer v1.0 παρέχει στο χρήστη τη δυνατότητα να αποθηκεύσει οποιαδήποτε στιγμή ότι εμφανίζεται στο παράθυρο απεικόνισης σε εικόνα τύπου TIFF με τρεις διαφορετικούς τύπους συμπίεσης, όπως παρουσιάζεται στην **Εικόνα 6.11**. Για το σκοπό αυτό χρησιμοποιήθηκε η συνάρτηση writetiff(), που έχει αναπτυχθεί από τον Mark J. Kilgard (Copyright (c) Mark J. Kilgard, 1997) και χρησιμοποιεί τη βιβλιοθήκη libtiff.lib. Η βιβλιοθήκη libtiff.lib έχει αναπτυχθεί και μεταγλωττιστεί από το Dongho Kim (dkim@seas.gwu.edu). Ο πηγαίος κώδικας, καθώς και οδηγίες διορθώσεων αυτής βρίσκονται στην ηλεκτρονική διεύθυνση www.libtiff.org/. Η ανάπτυξη μίας εκ νέου συνάρτησης θεωρήθηκε ότι ξεφεύγει από το σκοπό του παρόντος συγγράμματος και αποφεύχθηκε.



Εικόνα 6.11: Μενού επιλογών αποθήκευσης σε μορφή εικόνας τύπου TIFF του λογισμικού MeshViewer v1.0.

6.2.12 Συνάρτηση main()

Η συνάρτηση main() είναι η κεντρική συνάρτηση του κώδικα ενός προγράμματος γραμμένου σε C/C++ και είναι αδύνατο να παραληφθεί, αφού όλες οι υπόλοιπες συναρτήσεις καλούνται μέσω αυτής. Πέρα από αυτό, όταν χρησιμοποιούνται συναρτήσεις της βιβλιοθήκης OpenGL, για την απεικόνιση των γραφικών, απαιτείται η κλήση συγκεκριμένων συναρτήσεων και ρουτινών από τη συνάρτηση main(), ώστε ο κώδικας να λειτουργεί σωστά. Ένα παράδειγμα είναι το ακόλουθο:

```
void main(int argc, char **argv)
{
    glutInit(&argc, argv);
```

```
    glutInitWindowSize(600, 600);  
    glutInitWindowPosition (100,100);  
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);  
    glutCreateWindow("Mesh Viewer v1.00 - Kontolatis Ioannis           D.O.G. - T.U.C.");  
    glutDisplayFunc(redraw);  
    glutKeyboardFunc(keyboard);  
    glutMotionFunc(motion);  
    glutMouseFunc(button);  
    glutReshapeFunc(winReshape);  
  
    init();  
    glutMainLoop();  
}
```

Για τις ρουτίνες αυτές, που καλούνται στη συνάρτηση `main()` και αφορούν στη πλειοψηφία τους λειτουργίες παραθυρικής διαχείρισης, ο αναγνώστης μπορεί να ανατρέξει στην **Ενότητα 2.4**.

6.3 Εφαρμογές και μελλοντικές προεκτάσεις του λογισμικού MeshViewer v1.0

Στη συγκεκριμένη του έκδοση (v1.0) το λογισμικό MeshViewer χρησιμοποιείται για την απεικόνιση και διαχείριση τρισδιάστατων πλεγμάτων και επιφανειών, συλλέγοντας πληροφορίες από τους τύπους των αρχείων που αναφέρθηκαν στην **Ενότητα 6.1**. Αν και οι εφαρμογές του διακρίνονται εξειδικευμένες, ωστόσο είναι εύκολο να προσαρμοστεί, μέσω των συναρτήσεων `designMesh ()` (βλ. **Ενότητα 6.2.2**) και `drawGeneral ()`, κτλ. (βλ. **Ενότητα 6.2.4**) για την απεικόνιση και διαχείριση οποιουδήποτε αντικειμένου ή δεδομένων μέσω αρχείου.

Το λογισμικό MeshViewer αναπτύχθηκε με σκοπό να πλαισιώσει τις εφαρμογές Mesh v3.5 [12] και Ge.P.A.S. v2.0 [18] ως οπτικό βοήθημα, αλλά και για την εισαγωγή του γράφοντος στο χώρο των ψηφιακών γραφικών και της ανάπτυξης συστημάτων CAD. Ουσιαστικά, έγινε κατανοητό, μέχρι ενός επιπέδου, με ποιο τρόπο ο φυσικός κόσμος μπορεί να ψηφιοποιηθεί και να αναπαρασταθεί σε μία οπτική συσκευή, όπως η οθόνη του ηλεκτρονικού υπολογιστή.

Μελλοντικά, το λογισμικό MeshViewer μπορεί να εμπλουτιστεί με ρουτίνες που διαχειρίζονται και άλλους τύπους αρχείων, λειτουργίες φωτορεαλισμού και σκίασης, πολλαπλά παράθυρα απεικόνισης, κ.α.. Όμως, η κεντρική ιδέα της ανάπτυξης της παρούσας εφαρμογής ήταν ένα εργαλείο ευέλικτο, γρήγορο, με μικρές απαιτήσεις σε πόρους συστήματος, και κυρίως ανεξάρτητο συσκευών και λειτουργικού συστήματος. Γίνεται λοιπόν κατανοητό ότι αν προστεθούν πολλές λειτουργίες επιπλέον, θα αλλοιωθεί ο πρωταρχικός χαρακτήρας και τα πλεονεκτήματα της εφαρμογής.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] *Adobe Systems Incorporated, PostScript Language Reference Manual, Addison-Wesley, Reading, Mass., 1986.*
- [2] *Adrian Nye, X Window System User's Guide, volume 3 of The Definitive Guides to the X Window System, O'Reilly and Associates, Sebastapol, Ca., 1987.*
- [3] *Donald Hearn and M. Pauline Baker, Computer Graphics with OpenGL, Third Edition, Pearson Prentice Hall, 2004.*
- [4] *Garry Wiegand and Bob Covey, HOOPS Reference Manual, Version 3.0, Ithaca Software, 1991.*
- [5] *Gerald Farin, Curves and Surfaces for Computer Aided Geometric Design, Academic Press, Boston, Ma., Second edition, 1990.*
- [6] *International Standards Organization, International standard information processing systems - computer graphics - graphical kernel system for three dimensions (GKS-3D) functional description, Technical Report ISO Document Number 9905:1988(E), American National Standards Institute, New York, 1988.*
- [7] *Jeff Stevenson, PEXlib specification and C language binding, version 5.1P, The X Resource, Special Issue B, September 1992.*
- [8] *Jonathan S. Harbour, Microsoft Visual Basic Game Programming with DirectX, Premier Press, 2002.*
- [9] *Mark Segal and Kurt Akeley, The OpenGL graphics system: A specification, Technical report, Silicon Graphics Computer Systems, Mountain View, Ca., 1992.*
- [10] *Mark Segal and Kurt Akeley, The OpenGL graphics interface, Silicon Graphics Computer Systems, Mountain View, Ca., 1992.*
- [11] *Μπιλλάλης Νικόλαος, Μελέτη Σχεδίαση Με Χρήση Η/Υ (CAD), Πολυτεχνείο Κρήτης, 2003.*
- [12] *Ντιπτένη Μαρίνα, Κατασκευή Μη Δομημένων Πλεγμάτων σε Τρισδιάστατες Επιφάνειες με Χρήση της Μεθόδου Frontal Delaunay, Πολυτεχνείο Κρήτης, 2004.*
- [13] *Paula Womack, ed. PEX protocol specification and encoding, version 5.1P. The X Resource, Special Issue A, May 1992.*
- [14] *Paul Haeberli and Kurt Akeley, The accumulation buffer: Hardware support for high-quality rendering, In Proceedings of SIGGRAPH '90, pages 309-318, 1990.*
- [15] *Paul S. Heckbert, A survey of texture mapping, IEEE CG & A, pages 56-67, November 1986.*

- [16] *Paul S. Strauss and Rikk Carey, An object-oriented 3D graphics toolkit, In Proceedings of SIGGRAPH '92, pages 341-349, 1992.*
- [17] *PHIGS+ Committee, Andries van Dam, chair. PHIGS+ functional description, revision 3.0, Computer Graphics, 22(3):125-218, July 1988.*
- [18] *S.Sarakinos, E.Amoiralis, I.Valakos, I.K.Nikolos, Generic Parameterized Aircraft Surface Generation for Design Optimization, 7th National Congress on Mechanics, Chania, Greece, June 24-26, 2004.*
- [19] *Silicon Graphics, OpenGL Programming Guide, Second Edition, Addison-Wesley Publishing Company, 1997.*
- [20] *Toby Howard and Alan Murta, Graphics Programming with OpenGL, Tutorial and Reference Manual, Department of Computer Science - University of Manchester, 1999.*
- [21] *Wolfgang F. Engel and Amir Geva, Beginning Direct3D Game Programming, Prima Tech, 2001.*

ΠΑΡΑΡΤΗΜΑ

ΚΑΤΑΛΟΓΟΣ ΣΥΝΑΡΤΗΣΕΩΝ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ OpenGL	
Συνάρτηση	Περιγραφή
gluOrtho2D	Καθορίζει τις συντεταγμένες του παράθυρου αποκοπής ως παραμέτρους για μία διδιάστατη ορθογώνια προβολή.
glVertex*	Επιλέγει μία θέση συντεταγμένων. Η συνάρτηση αυτή πρέπει να τοποθετηθεί ανάμεσα στο ζευγάρι συναρτήσεων glBegin/glEnd.
glBegin (GL_POINTS);	Σχεδιασμός ενός ή περισσότερων θέσεων σημείων, που το καθένα έχει καθοριστεί σε μία glVertex συνάρτηση. Η λίστα των θέσεων κλείνει έπειτα με τη δήλωση glEnd.
glBegin (GL_LINES);	Απεικονίζει ένα σύνολο από ευθύγραμμα τμήματα, των οποίων οι συντεταγμένες των τελικών τους σημείων καθορίζονται σε glVertex συναρτήσεις. Η λίστα των τελικών σημείων κλείνει έπειτα με τη δήλωση glEnd.
glBegin (GL_LINE_STRIP);	Απεικονίζει μία polyline, που καθορίζεται χρησιμοποιώντας την ίδια δομή όπως στη GL_LINES.
glBegin (GL_LINE_LOOP);	Απεικονίζει μία κλειστή polyline, που καθορίζεται χρησιμοποιώντας την ίδια δομή όπως στη GL_LINES.
glRect*	Απεικονίζει ένα γεμάτο ορθογώνιο παραλληλόγραμμο στο xy-επίπεδο.
glBegin (GL_POLYGON);	Απεικονίζει ένα γεμάτο πολύγωνο, του οποίου οι κορυφές δίνονται σε glVertex συναρτήσεις και τερματίζεται με τη δήλωση glEnd.
glBegin (GL_TRIANGLES);	Απεικονίζει ένα σύνολο γεμάτων τριγώνων χρησιμοποιώντας την ίδια δομή με τη GL_POLYGON.
glBegin (GL_TRIANGLE_STRIP);	Απεικονίζει ένα πλέγμα γεμάτων τριγώνων, που καθορίζεται χρησιμοποιώντας την ίδια δομή με τη GL_POLYGON.
glBegin (GL_TRIANGLE_FAN);	Απεικονίζει ένα πλέγμα γεμάτων τριγώνων σε σχήμα βεντάλιας με όλα τα τρίγωνα να συνδέονται στη πρώτη κορυφή, το οποίο καθορίζεται με την ίδια δομή όπως η GL_POLYGON.
glBegin (GL_QUADS);	Απεικονίζει ένα σύνολο από γεμάτα τετράπλευρα, που καθορίζεται με την ίδια

	δομή όπως στη GL_POLYGON.
glBegin (GL_QUAD_STRIP);	Απεικονίζει ένα πλέγμα γεμάτων τετραπλεύρων, που καθορίζεται με την ίδια δομή όπως στη GL_POLYGON.
glNewList (listID, listMode)	Καθορίζει ένα σύνολο εντολών ως λίστα απεικόνισης, που τερματίζεται με τη δήλωση glEndList.
glCallList	Εκτελεί μία μοναδική λίστα απεικόνισης.
glCallLists	Εκτελεί πολλαπλές λίστες απεικόνισης.
glDeleteLists	Εξαλείφει μία καθορισμένη ακολουθία λιστών απεικόνισης.
glReadPixels	Αποθηκεύει ένα τμήμα εικονοστοιχείων σε ένα επιλεγμένο μητρώο.
glCopyPixels	Αντιγράφει ένα τμήμα εικονοστοιχείων από μία θέση αποθήκευσης σε άλλη.
glutReshapeFunc	Καθορίζει ενέργειες που γίνονται όταν οι διαστάσεις ενός παράθυρου απεικόνισης αλλάζουν.
glutInitDisplayMode	Επιλογή της μεθόδου χρώματος, η οποία μπορεί να είναι είτε GLUT_RGB είτε GLUT_INDEX.
glColor*	Καθορίζει ένα RGB ή RGBA χρώμα.
glIndex*	Καθορίζει ένα χρώμα χρησιμοποιώντας ένα πίνακα χρωμάτων για ευρετήριο.
glPointSize (size);	Καθορίζει το μέγεθος ενός σημείου.
glLineWidth (width);	Καθορίζει το πλάτος μίας γραμμής.
glEnable (GL_LINE_STIPPLE);	Ενεργοποιεί την επιλογή τύπου γραμμής.
glEnable (GL_POLYGON_STIPPLE);	Ενεργοποιεί την επιλογή τύπου γεμίσματος.
glPolygonMode	Απεικονίζει τη μπροστά ή πίσω πλευρά είτε ως ένα σύνολο ακμών είτε ως κορυφών.
glFrontFace	Καθορίζει τη σειρά των κορυφών της μπροστινής όψης ως GL_CCW ή GL_CW.
glTranslate*	Καθορίζει της παραμέτρους μεταφοράς.
glRotate*	Καθορίζει της παραμέτρους της περιστροφής περί οποιουδήποτε άξονα διέρχεται από την αρχή των αξόνων.
glScale*	Καθορίζει της παραμέτρους αλλαγής κλίμακας ως προς την αρχή των συντεταγμένων.
glMatrixMode	Καθορίζει το τρέχοντα πίνακα για γεωμετρικούς μετασχηματισμούς οπτικής, για μετασχηματισμούς προβολών, υψής ή χρώματος.
glLoadIdentity	Θέτει το τρέχοντα πίνακα ως μοναδιαίο.
glLoadMatrix* (elems);	Θέτει τα στοιχεία του τρέχοντα πίνακα.
glMultMatrix* (elems);	Μετα-πολλαπλασιάζει το τρέχοντα

	πίνακα με το καθορισμένο πίνακα.
glPushMatrix	Αντιγράφει τον άνω πίνακα του σωρού και αποθηκεύει ένα αντίγραφο στη δεύτερη θέση του σωρού.
glPopMatrix	Διαγράφει τον άνω πίνακα του σωρού και μετακινεί το δεύτερο στη κορυφή του σωρού.
glViewport	Καθορίζει τις παραμέτρους συντεταγμένων οθόνης για μία οπτική οδό.
glGetIntegerv	Χρησιμοποιεί τα ορίσματα GL_VIEWPORT και vpArray για τη λήψη των παραμέτρων για τη τρέχουσα ενεργή οπτική οδό.
glutInit	Αρχικοποιεί τη βιβλιοθήκη GLUT.
glutInitWindowPosition	Καθορίζει τις συντεταγμένες της άνω αριστερής γωνίας του παράθυρου απεικόνισης.
glutInitWindowSize	Καθορίζει πλάτος και ύψος του παράθυρου απεικόνισης.
glutCreateWindow	Δημιουργεί ένα παράθυρο απεικόνισης (το οποίο προσδιορίζεται με έναν ακέραιο αναγνωριστή) και καθορίζει ένα τίτλο για αυτό.
glClearColor	Καθορίζει το χρώμα του φόντου για ένα παράθυρο απεικόνισης σε κώδικα RGB.
glClearIndex	Καθορίζει το χρώμα του φόντου για ένα παράθυρο απεικόνισης σε κώδικα ευρετηρίου χρώματος.
glutDestroyWindow	Καθορίζει έναν αριθμό αναγνωριστή για ένα παράθυρο απεικόνισης που πρόκειται να διαγραφεί.
glutSetWindow	Καθορίζει έναν αριθμό αναγνωριστή για ένα παράθυρο απεικόνισης που πρόκειται να είναι το τρέχων παράθυρο.
glutPositionWindow	Επαναφέρει στην αρχική κατάσταση τη τοποθεσία στην οθόνη για το τρέχων παράθυρο απεικόνισης.
glutReshapeWindow	Επαναφέρει στην αρχική κατάσταση το πλάτος και ύψος του τρέχοντος παράθυρου απεικόνισης.
glutFullScreen	Θέτει το τρέχων παράθυρο απεικόνισης στο μέγεθος της οθόνης.
glutSetWindowTitle	Καθορίζει ένα νέο τίτλο για το τρέχων παράθυρο απεικόνισης.
glutPopWindow	Μετακινεί το τρέχων παράθυρο απεικόνισης μπροστά από τα υπόλοιπα παράθυρα.
glutPushWindow	Μετακινεί το τρέχων παράθυρο

	απεικόνισης πίσω από τα υπόλοιπα παράθυρα.
glutShowWindow	Επιστρέφει το τρέχων παράθυρο απεικόνισης στην οθόνη.
glutCreateSubWindow	Κατασκευάζει ένα παράθυρο δευτέρου επιπέδου μέσα στο παράθυρο απεικόνισης.
glutSetCursor	Επιλέγει το σχήμα του κέρσορα της οθόνης.
glutDisplayFunc	Θέτει εν ενεργεία μία συνάρτηση για τη δημιουργία μίας εικόνας μέσα στο τρέχων παράθυρο απεικόνισης.
glutPostRedisplay	Ανανεώνει τα περιεχόμενα του τρέχοντος παράθυρου.
glutMainLoop	Εκτελεί το πρόγραμμα των γραφικών.
glutIdleFunc	Καθορίζει μία συνάρτηση που θα εκτελείται όταν το σύστημα είναι αδρανές.
glutGet	Εξετάζει το σύστημα σχετικά με μία καθορισμένη παράμετρο κατάστασης.
gluLookAt	Καθορίζει τις τρισδιάστατες παραμέτρους οπτικής.
glOrtho	Καθορίζει τις παραμέτρους για ένα παράθυρο αποκοπής, καθώς και το μακρινό και κοντινό επίπεδο αποκοπής για μία ορθογώνια προβολή.
gluPerspective	Καθορίζει τη γωνία του πεδίου οπτικής και άλλες παραμέτρους για μία συμμετρική προοπτική προβολή.
glFrustum	Καθορίζει τις παραμέτρους για ένα παράθυρο αποκοπής, καθώς και το κοντινό και μακρινό επίπεδο αποκοπής για μία προοπτική προβολή (συμμετρική ή πλάγια).
glClipPlane	Καθορίζει τις παραμέτρους για ένα εναλλακτικό επίπεδο αποκοπής.

Για περισσότερες πληροφορίες ο αναγνώστης μπορεί να ανατρέξει στις εξής διευθύνσεις στο διαδίκτυο:

- www.opengl.org/
- www.sgi.com/products/software/opengl/
- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/opengl.asp>