

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
«ΠΛΗΡΟΦΟΡΙΚΗ»

# ΓΡΑΦΙΚΑ ΜΕ ΥΠΟΛΟΓΙΣΤΕΣ – ΕΙΚΟΝΙΚΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ

**Σημειώσεις Διδασκαλίας**

**Θ. Παναγιωτόπουλος**

Καθηγητής

Τμήμα Πληροφορικής

Πανεπιστήμιο Πειραιά

**Δρ Γ. Αναστασάκης**

Τμήμα Πληροφορικής

Πανεπιστήμιο Πειραιά

**Πειραιάς 2013**

# Περιεχόμενα

Γραφικά με Υπολογιστές.....	4
Εισαγωγή.....	4
Συστήματα raster απεικόνισης.....	5
Συστήματα διανυσματικής απεικόνισης.....	9
Σύγκριση συστημάτων raster και διανυσματικής απεικόνισης.....	9
Μαθηματικό υπόβαθρο.....	10
Διανύσματα.....	10
Πίνακες.....	11
Εικονική Πραγματικότητα.....	15
Θεμελιώδη φυσιολογικά χαρακτηριστικά.....	15
Ακοή και όραση.....	15
Η αίσθηση του βάθους στην όραση.....	18
Τρισδιάστατος ήχος.....	19
Γράφημα σκηνής.....	20
X3D/VRML.....	20
Java3D.....	22
OpenSceneGraph.....	23
OpenSG.....	24
Άλλα παραδείγματα γραφημάτων σκηνής.....	25
Αναπαραστάσεις χαμηλού επιπέδου.....	25
Η Γλώσσα VRML.....	27
Επισκόπηση της γλώσσας.....	27
Άξονες συντεταγμένων και γεωμετρικοί μετασχηματισμοί.....	30
Στοιχειώδη γεωμετρικά σχήματα.....	33
Κόμβοι κειμένου .....	36
Σημεία κατόπτρευσης .....	38
Μορφοποίηση .....	39
Πηγές φωτός.....	40
Επιφανειακά χαρακτηριστικά.....	44
Τεχνικές σκίασης.....	45

# Γραφικά με Υπολογιστές

## Εισαγωγή

Τα γραφικά με υπολογιστές (*computer graphics, CG*) είναι ένας από τους πιο εντυπωσιακούς τομείς της επιστήμης της Πληροφορικής σήμερα. Είναι, όμως, και ένας από τους πιο χρήσιμους, καθώς αξιοποιείται σε πλήθος άλλων τομέων, όπως:

**Συστήματα διεπαφής (user interfaces).** Τα συστήματα διεπαφής των ηλεκτρονικών υπολογιστών υπάρχουν χάρη στα γραφικά με υπολογιστές. Αυτό αφορά παλαιότερα συστήματα τα οποία βασίζονταν στην απλή εμφάνιση αδιαμόρφωτου κειμένου στην οθόνη του υπολογιστή, συστήματα τα οποία εμφανίστηκαν στη συνέχεια και βασίζονταν στις έννοιες της επιφάνειας εργασίας (*desktop*), του παραθύρου (*window*), του μενού (*menu*) και του δείκτη (*pointer*), αλλά και τα πλέον σύγχρονα συστήματα τα οποία επιτρέπουν προηγμένες μορφές προβολής και διαχείρισης της επιφάνειας εργασίας και των παραθύρων σε τρεις διαστάσεις. Καθώς το σύνολο, σχεδόν, της αλληλεπίδρασης ανθρώπου-υπολογιστή σήμερα γίνεται μέσα από συστήματα διεπαφής, τα γραφικά με υπολογιστές αποκτούν τεράστια σημασία.

**Επιστημονική απεικόνιση (scientific visualisation).** Τα γραφικά με υπολογιστές χρησιμοποιούνται για την απεικόνιση δισδιάστατων και τρισδιάστατων γραφημάτων στη φυσική, στα μαθηματικά, στην ιατρική, στη στατιστική, στη διοίκηση επιχειρήσεων, στο marketing, στα χρηματοοικονομικά και σε άλλους τομείς όπου είναι απαραίτητη η απεικόνιση τάσεων, προτύπων, κ.ά., όπως αυτά προκύπτουν από δεδομένα, με ευνόητο και συνεπή τρόπο, ώστε να διευκολύνεται η κατανόηση σύνθετων φαινομένων και η λήψη αποφάσεων.

**Ιατρική.** Πέραν των δυνατοτήτων απεικόνισης που προσφέρουν, τα γραφικά με υπολογιστές έχουν πολλές εφαρμογές σε χώρους όπως η διαγνωστική ιατρική και η τηλεϊατρική, όπου επιτρέπουν την εξέταση τρισδιάστατων μοντέλων οργάνων, την εμφάνιση αποτελεσμάτων μετά από επεξεργασία εικόνων και ακτινοσκοπήσεων, την παρουσίαση πληροφοριών κατά τη διάρκεια επεμβάσεων, κ.ά.

**Σχεδίαση με Υπολογιστές (Computer-Aided Design, CAD).** Χάρη στα γραφικά με υπολογιστές, η σχεδίαση με τη χρήση υπολογιστή έχει πλέον γίνει μία διαδραστική διαδικασία, με εφαρμογές σε χώρους όπως η αρχιτεκτονική, η σχεδιασμός ολοκληρωμένων κυκλωμάτων μεγάλης κλίμακας, ο βιομηχανικός σχεδιασμός, κ.ά.

**Γεωγραφικά Πληροφοριακά Συστήματα (Geographical Information Systems, GIS).** Στον τομέα αυτό ανήκουν τα πληροφοριακά συστήματα που επιτρέπουν την προβολή και επεξεργασία χαρτογραφικών δεδομένων. Χάρη στα γραφικά με υπολογιστές, είναι δυνατή η απεικόνιση και η

αλληλεπίδραση με ψηφιακούς χάρτες, αλλά και η εμφάνιση σχολιασμών, οδικών δικτύων, σημείων ενδιαφέροντος, κ.ά. Επιπλέον, είναι δυνατή η δημιουργία τρισδιάστατων απεικονίσεων που παρέχουν πρόσθετη πληροφόρηση σχετικά με τη μορφολογία του εδάφους.

**Πολυμέσα (multimedia).** Σχεδόν κάθε σύστημα πολυμέσων παρέχει οπτικές απεικονίσεις πληροφορίας (εικόνες, διαγράμματα, video, κ.ά.) η δημιουργία των οποίων είναι εφικτή χάρη στα γραφικά με υπολογιστές.

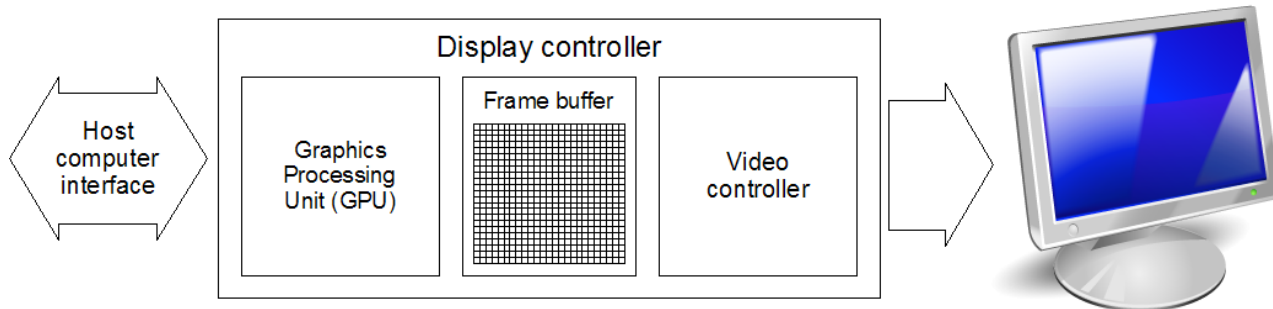
**Ψυχαγωγία.** Η συμβολή των γραφικών με υπολογιστές στο χώρο της ψηφιακής ψυχαγωγίας είναι τεράστια: ειδικά εφέ στον κινηματογράφο και την τηλεόραση, κινηματογραφικά έργα δημιουργημένα εν μέρει ή εξ' ολοκλήρου με τη χρήση γραφικών, αλλά και παιχνίδια με υπολογιστές (μία αγορά με πρωτοφανή άνοδο τα τελευταία χρόνια), είναι λίγα μόνο από τα εντυπωσιακά αποτελέσματα της χρήσης γραφικών με υπολογιστές στο χώρο της ψυχαγωγίας.

**Εικονική Πραγματικότητα (Virtual Reality, VR).** Η σχέση της *εικονικής πραγματικότητας* (*virtual reality, VR*) με τα γραφικά με υπολογιστές είναι άμεση. Κάθε σύστημα εικονικής πραγματικότητας βασίζεται, μεταξύ άλλων, στην οπτική απεικόνιση εικονικών κόσμων προκειμένου να παρέχει στο χρήστη την ψευδαίσθηση της εμπύθισης. Η συμβολή των γραφικών με υπολογιστές είναι ακόμα μεγαλύτερη σε συστήματα όπου χρησιμοποιείται ειδικός εξοπλισμός τρισδιάστατης προβολής για την επίτευξη ακόμα μεγαλύτερων επιπέδων εμπύθισης, όπως επίσης και σε εφαρμογές όπου απαιτείται αυξημένος οπτικός ρεαλισμός. Εξ' άλλου, τα εικονικά σώματα χρηστών αλλά και συνθετικών πρακτόρων – απαραίτητα συστατικά κάθε εικονικού κόσμου – διαθέτουν συχνά κάποιας μορφής σκελετική δομή και είναι ικανά για ρεαλιστική κίνηση, χαρακτηριστικά για τα οποία γίνεται εκτενής χρήση δομών και τεχνικών από τα γραφικά με υπολογιστές, όπως το γράφημα σκηνής και οι τρισδιάστατοι γεωμετρικοί μετασχηματισμοί.

Φυσικά, ένα δεδομένο που επικεντρώνει την προσοχή ακόμα περισσότερο στο χώρο των γραφικών με υπολογιστές είναι η εντυπωσιακή πρόοδος στον τομέα του υλικού γραφικών με υπολογιστές (*computer graphics hardware*) που παρατηρείται τα τελευταία χρόνια. Αξίζει να σημειωθεί πως, σήμερα, ο συγκεκριμένος τομέας έχει να επιδείξει *μονάδες επεξεργασίας γραφικών* (*graphics processing units, GPUs*) οι οποίες, όχι απλά είναι ισάξιες σε υπολογιστική ισχύ με τις παραδοσιακές *κεντρικές μονάδες επεξεργασίας* (*central processing units, CPUs*), αλλά, επιπλέον, μπορούν να αναλάβουν και υπολογιστικά καθήκοντα γενικού χαρακτήρα και όχι αποκλειστικά τη δημιουργία γραφικών.

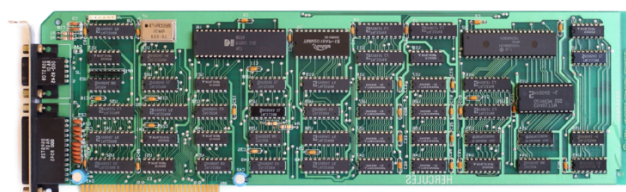
## **Συστήματα raster απεικόνισης**

Ένα τυπικό *σύστημα raster απεικόνισης* βασίζεται στο παρακάτω αφηρημένο σχεδιαστικό πρότυπο:



Εικόνα 1 – Τυπικό σύστημα απεικόνισης βασισμένο στην έννοια του frame buffer

Το βασικό συστατικό του παραπάνω σχήματος είναι αυτό που τιτλοφορείται *display controller*. Το συστατικό αυτό δεν είναι άλλο από τη γνωστή *κάρτα γραφικών* που περιλαμβάνουν όλοι οι υπολογιστές τύπου personal computer (PC) σήμερα<sup>1</sup>. Είναι, με άλλα λόγια, ένα ηλεκτρονικό κύκλωμα πλήρως εξειδικευμένο στη δημιουργία γραφικών, η δημοφιλής ονομασία του οποίου οφείλεται στη συνήθη μορφή που έχει, δηλαδή, της πλακέτας η οποία τοποθετείται πάνω στη μητρική πλακέτα του υπολογιστή.



Hercules (1982)  
by Hercules Computer Technology  
Maximum resolution: 720x348  
Maximum number of colours: 2 (black, white)



NVIDIA GeForce 8800 GT (2006)  
by NVIDIA Corporation  
Maximum resolution: 2048x1536  
Maximum number of colours: 2<sup>32</sup>

Εικόνα 2 – Παλαιότερες και σύγχρονες κάρτες γραφικών

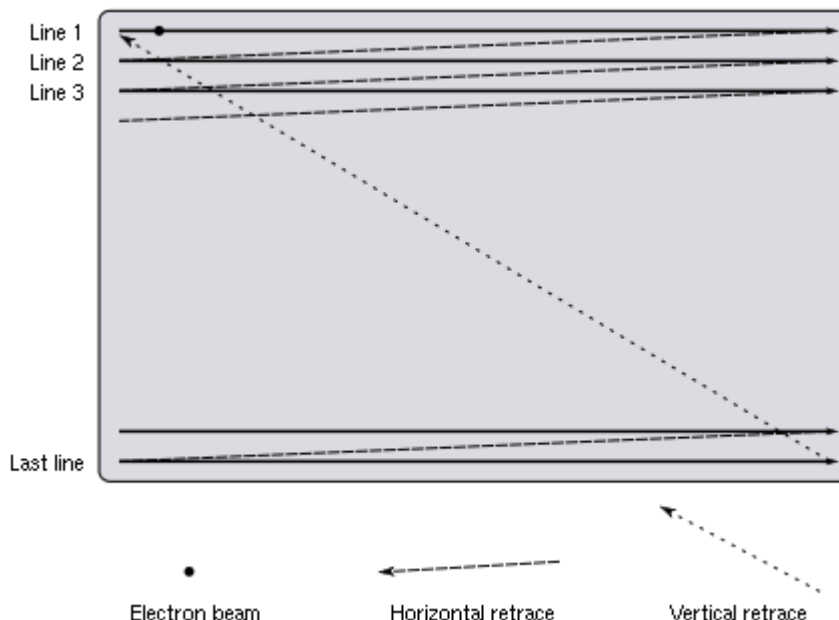
Όπως φαίνεται στο σχήμα, το *display controller* αποτελείται από τρία συστατικά, η λειτουργία των οποίων εξηγείται στη συνέχεια.

**Graphics processing unit.** Είναι ένα ηλεκτρονικό κύκλωμα το οποίο αναλαμβάνει να μεταφράσει οδηγίες του υπολογιστή οι οποίες αφορούν τη δημιουργία γραφικών. Οι λειτουργίες αυτές ανήκουν σε ένα ιδιαίτερα ευρύ φάσμα, από απλές εντολές απεικόνισης σημείων και απλών γεωμετρικών σχημάτων, μέχρι ιδιαίτερα σύνθετες ακολουθίες εντολών δημιουργίας τρισδιάστατων γραφικών,

<sup>1</sup> Η έννοια του *display controller* και το αφηρημένο σχεδιαστικό πρότυπο που δίνεται γι' αυτόν δεν αφορά μόνο υπολογιστές τύπου PC αλλά το σύνολο των υπολογιστών που διαθέτουν μηχανισμούς raster απεικόνισης.

κίνησης, οπτικών εφέ, κ.ά. Το αποτέλεσμα της λειτουργίας αυτού του συστατικού είναι η δημιουργία δεδομένων τα οποία αντικατοπτρίζουν την εικόνα που πρέπει να εμφανιστεί στο μέσο προβολής (π.χ., οθόνη, προβολέας, κ.τ.λ.), σύμφωνα με τις οδηγίες του υπολογιστή και με βάση συγκεκριμένη κωδικοποίηση.

**Frame buffer.** Το συστατικό αυτό περιέχει δεδομένα τα οποία αντικατοπτρίζουν το τι ακριβώς πρέπει να εμφανιστεί στο μέσο προβολής, τα οποία είναι το αποτέλεσμα της λειτουργίας του GPU. Η ύπαρξή του είναι το στοιχείο που ουσιαστικά καθορίζει τη φύση του συστήματος απεικόνισης ως raster. Συγκεκριμένα, είναι δομημένο με τέτοιο τρόπο ώστε να κωδικοποιεί το χρώμα καθενός από τα  $(m \times n)$  στοιχεία μίας επίπεδης επιφάνειας προβολής, τα οποία ονομάζονται και *εικονοστοιχεία* (*pixels*, από το *picture elements*). Οι επιφάνειες προβολής που είναι οργανωμένες ως πίνακες στοιχείων σε κατακόρυφη/οριζόντια διάταξη ονομάζονται *raster επιφάνειες προβολής* και, κατ' επέκταση, τα συστήματα απεικόνισης που τις οδηγούν ονομάζονται *συστήματα raster απεικόνισης*. Αν και στο σχήμα το frame buffer φαίνεται να έχει και αυτό διάταξη πίνακα, στην πραγματικότητα είναι μιας μορφής μνήμη όπως η κύρια μνήμη του υπολογιστή, και, έτσι, έχει και αυτό σειριακή διάταξη. Η οργάνωση των περιεχομένων του είναι σε γραμμές ξεκινώντας από το πάνω-αριστερά pixel. Αυτό υιοθετήθηκε αρχικά για να εξυπηρετεί τη διαδικασία σάρωσης των *οθονών καθοδικού σωλήνα* (*cathode ray tube displays, CRT displays*) που κυριαρχούσαν ως το μόνο ευρέως διαθέσιμο μέσο προβολής μέχρι και πριν μερικά χρόνια, η οποία γινόταν όπως φαίνεται στο σχήμα.



Εικόνα 3 – Σάρωση frame buffer

Έστω ένα μέσο προβολής το οποίο αποτελείται από  $(m \times n)$  pixels, κάθε ένα από τα οποία μπορεί

να απεικονίσει  $c$  χρώματα. Για το μέσο απεικόνισης αυτό ορίζεται ότι η *ανάλυση* (*resolution*) του είναι  $(m \times n)$ . Επιπλέον, προκειμένου το frame buffer να μπορεί να κωδικοποιήσει  $c$  διαφορετικές τιμές χρώματος για το παραπάνω μέσο προβολής, πρέπει να μπορεί να διαθέσει ικανό αριθμό *bits* ανά *pixel* (*bits-per-pixel*, *BPP*). Συγκεκριμένα, για  $c$  χρώματα, απαιτούνται  $BPP = \log_2 c$  bits, καθώς  $2^{BPP} = c$ . Η τιμή *BPP* ονομάζεται και *χρωματικό βάθος* (*colour depth*). Τυπικές τιμές της είναι:

Colour depth	Αριθμός χρωμάτων	Συνήθης ονομασία
1	2	Μονοχρωματικά μέσα προβολής (monochrome displays), μαύρο και άσπρο
3	8	Ένα bit ανά βασικό χρώμα (κόκκινο, πράσινο και μπλέ)
8	256	Χρήση παλέτας (indexed mode)
24	$2^{24}$	Truecolor mode

Table 1 - Τυπικές τιμές colour depth και αριθμού υποστηριζόμενων χρωμάτων

Τις περισσότερες φορές, μία τιμή *BPP* bits ορίζει τις επιμέρους τιμές των τριών βασικών χρωμάτων τα οποία μπορούν να απεικονίσουν τα pixels των περισσότερων μέσων προβολής (κόκκινο/πράσινο/μπλε, red/green/blue, γι' αυτό και τα μέσα προβολής αυτά ονομάζονται *RGB displays*) ως συστατικών του ζητούμενου χρώματος (π.χ., η τιμή 101 αντιστοιχεί στο συνδυασμό κόκκινο/μπλε που δίνει μια απόχρωση του μωβ). Πολλές φορές όμως, όπως στην περίπτωση 8 bits ανά pixel όπου δεν είναι δυνατή η αντιστοίχιση του ίδιου ακεραίου πλήθους bits σε κάθε ένα από τα τρία βασικά χρώματα, η τιμή που αποθηκεύεται ανά pixel δεν αντιστοιχεί στο πραγματικό του χρώμα αλλά είναι δείκτης σε μία παλέτα χρωμάτων (*colour palette*). Τελικά, η συνολική απαίτηση σε μνήμη  $M$  (σε bytes) για το frame buffer και την παλέτα ενός συστήματος απεικόνισης που οδηγεί μέσα προβολής *ανάλυσης*  $(m \times n)$  και πλήθους χρωμάτων  $c$  είναι  $M = m * n * \log_2 c / 8 + P$ , όπου  $P$  το μέγεθος της περιοχής μνήμης σε bytes που περιέχει τις πληροφορίες της παλέτας ανάλογα με την χρωματική κωδικοποίηση που έχει επιλεγεί για την κάθε περίπτωση (RGB, HSV, κ.τ.λ.)

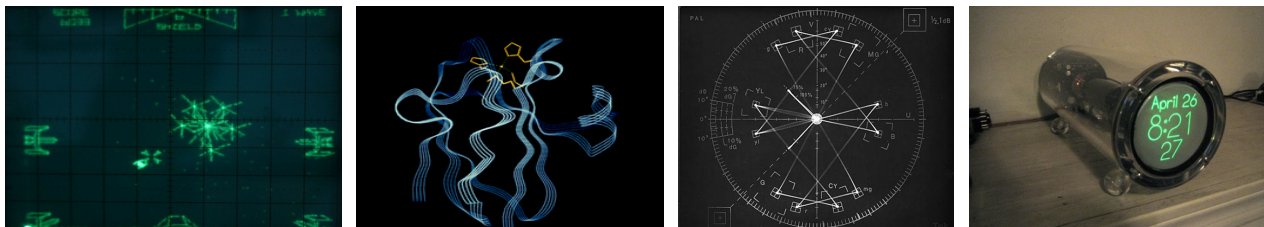
**Video controller.** Η λειτουργία του συστατικού αυτού είναι απλή, και περιορίζεται στο να καθοδηγεί το μέσο προβολής ώστε να απεικονίζεται σωστά η εικόνα που κωδικοποιεί το frame buffer. Ουσιαστικά, αναλαμβάνει τη μετατροπή των δεδομένων που περιέχονται στο frame buffer σε κατάλληλα για το μέσο προβολής ηλεκτρικά σήματα. Λειτουργεί επαναληπτικά, συνήθως σε συχνότητες 60Hz – 120Hz.

Αξίζει να σημειωθεί πως, ενώ η τεχνολογία του υλικού γραφικών με υπολογιστή έχει προοδεύσει σημαντικά, οι κύριες έννοιες στις οποίες βασίζεται έχουν παραμείνει οι ίδιες. Έτσι, κάθε raster display controller, από τα πρώτα που εμφανίστηκαν μέχρι τα σημερινά, περιλαμβάνει GPU, frame buffer και video controller, παρά το γεγονός ότι, σε κάθε περίπτωση, τα τρία αυτά συστατικά έχουν τελείως διαφορετικές δυνατότητες και λειτουργούν και ανταλλάσσουν δεδομένα μεταξύ τους και με

τον υπολογιστή με τελείως διαφορετικό τρόπο.

## Συστήματα διανυσματικής απεικόνισης

Τα συστήματα απεικόνισης δεν είχαν πάντα την παραπάνω μορφή. Πριν την εμφάνιση των συστημάτων raster απεικόνισης χρησιμοποιούνταν τα λεγόμενα συστήματα διανυσματικής απεικόνισης (*vector display systems*).

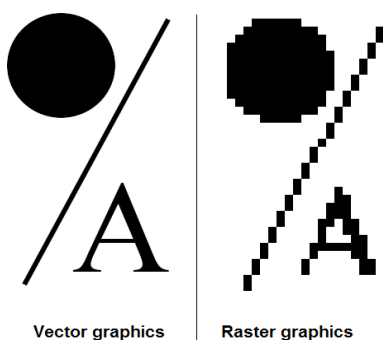


Εικόνα 4 – Vector displays (design + examples)

Στα συστήματα αυτά, το μέσο προβολής δεν είναι οργανωμένο σαν πίνακας ( $m \times n$ ) pixels, αλλά επιτρέπει την εμφάνιση σημείων σε οποιοδήποτε σημείο της επιφάνειάς του. Απαιτεί τη χρήση οθονών καθοδικού σωλήνα και, για το λόγο αυτό, έχει σχεδόν εγκαταλειφθεί εδώ και δεκαετίες, ενώ χρησιμοποιείται μόνο σε εξειδικευμένες εφαρμογές (π.χ., ραντάρ, άλλα συστήματα σάρωσης).

## Σύγκριση συστημάτων raster και διανυσματικής απεικόνισης

Ένα από τα μεγαλύτερα μειονεκτήματα των συστημάτων διανυσματικής απεικόνισης, το οποίο οδήγησε στην σχεδόν ολοκληρωτική υιοθέτηση των συστημάτων raster απεικόνισης, είναι η αδυναμία των πρώτων να δημιουργήσουν απεικονίσεις οι οποίες περιέχουν σκιασμένες περιοχές, όπως επίσης και ο περιορισμένος αριθμός χρωμάτων που μπορούν να απεικονίσουν. Επιπλέον, ένα



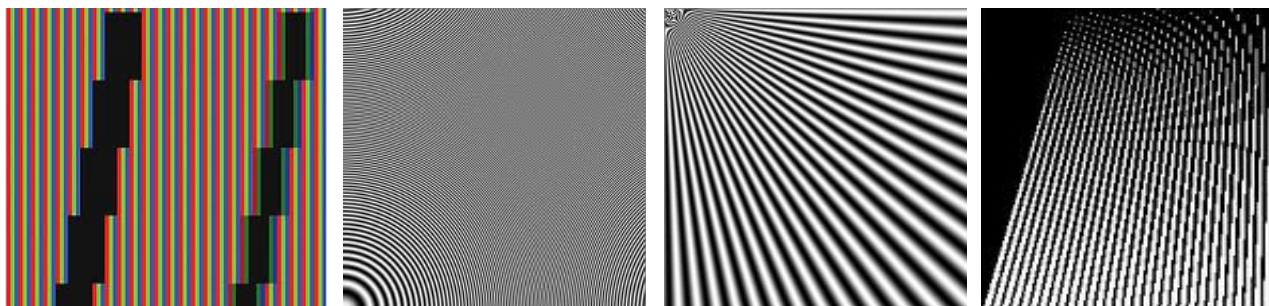
Εικόνα 5 – Vector vs. raster graphics (σε μεγέθυνση)

άλλο σημαντικό μειονέκτημά τους είναι ότι ο χρόνος που χρειάζονται για να απεικονίσουν πλήρως μία εικόνα εξαρτάται από την πολυπλοκότητα και το μέγεθός της, κάτι που καθιστά απαγορευτική τη χρήση τους σε συστήματα όπου απαιτείται απεικόνιση σε πραγματικό χρόνο. Εκεί όπου τα συστήματα raster απεικόνισης είναι η διακριτή φύση των πρώτων. Το γεγονός αυτό δημιουργεί την αναγκαιότητα αντιμετώπισης όλων ανεξαιρέτως των σχημάτων προς απεικόνιση από συστήματα raster απεικόνισης

ως σύνολα σημείων σε συγκεκριμένη διάταξη μεταξύ τους. Κατ' επέκταση, δημιουργείται η ανάγκη ορισμού βασικών γεωμετρικών σχημάτων, όπως το ευθύγραμμο τμήμα, το τρίγωνο, ο κύκλος, κ.τ.λ., με βάση συγκεκριμένα σημεία αναφοράς (π.χ., για το ευθύγραμμο τμήμα, τα δύο άκρα του,

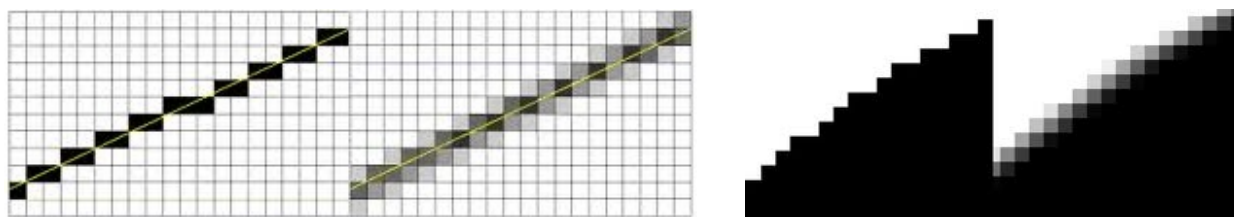


ενώ για τον κύκλο, το κέντρο και ένα σημείο της περιφέρειάς του) και ο αλγοριθμικός υπολογισμός όλων των ενδιάμεσων. Ακόμα, δημιουργούνται και προβλήματα ποιότητας στην απεικόνιση λόγω της μη επαρκούς, πολλές φορές, ανάλυσης του συστήματος απεικόνισης. Τα προβλήματα αυτά μπορεί να εμφανίζονται με διάφορες μορφές, όπως οι “πριονωτές” ακμές (jagged edges, jaggies) και ο σχηματισμός ειδώλων (aliasing), ιδιαίτερα μετά από μεγέθυνση.



Εικόνα 6 – Jaggies και aliasing

Η εξάλειψη του φαινομένου της δημιουργίας ειδώλων απαιτεί τη χρήση ειδικών τεχνικών (anti-aliasing), γεγονός που εισάγει ένα πρόσθετο, πολλές φορές σημαντικό, υπολογιστικό κόστος.



Εικόνα 7 – Anti-aliasing

Σήμερα, το υπολογιστικό κόστος της εφαρμογής τεχνικών anti-aliasing μειώνεται σημαντικά χάρη στην εγγενή υποστήριξη που παρέχουν γι' αυτό τα συστήματα απεικόνισης (κάρτες γραφικών) σε επίπεδο υλικού.

## **Μαθηματικό υπόβαθρο**

Στα γραφικά με υπολογιστές χρησιμοποιούνται πλήθος μαθηματικών δομών, οι κυριότερες από τις οποίες περιγράφονται στη συνέχεια.

### **Διανύσματα**

Στα γραφικά με υπολογιστές, ένα  $n$ -διάστατο διάνυσμα  $u$  ορίζεται ως μία πλειάδα  $n$  πραγματικών αριθμών, όπου το  $n$  είναι ίσο με 2 για δισδιάστατα διανύσματα, 3 για τρισδιάστατα, κ.ο.κ. Ένα διάνυσμα  $u$  με  $n$  στοιχεία συμβολίζεται όπως φαίνεται στη συνέχεια.

### Συμβολισμός διανυσμάτων:

$$u = (u_1, u_2, \dots, u_n), \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{bmatrix} .$$

Οι πράξεις διανυσμάτων με τη μεγαλύτερη σημασία για τα γραφικά με υπολογιστές είναι η πρόσθεση διανυσμάτων, ο πολλαπλασιασμός διανύσματος με πραγματικό αριθμό και το εσωτερικό γινόμενο διανυσμάτων. Σημειώνεται ότι η πρόσθεση διανυσμάτων είναι αντιμεταθετική και προσεταιριστική και διαθέτει ουδέτερο στοιχείο το διάνυσμα  $\theta = (0, 0, \dots, 0)$ . Επίσης, για τα διανύσματα  $u, v$  και τους πραγματικούς αριθμούς  $\alpha, \beta$  ισχύουν:  $\alpha u = u\alpha$ ,  $(\alpha\beta)u = \alpha(\beta u)$ ,  $1u = u$ ,  $(\alpha + \beta)u = \alpha u + \beta u$  και  $\alpha(u + v) = \alpha u + \alpha v$ .

### Πρόσθεση διανυσμάτων:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix}, \quad u + v = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \dots \\ u_n + v_n \end{bmatrix} .$$

### Πολλαπλασιασμός διανύσματος με πραγματικό αριθμό:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{bmatrix}, \quad \alpha \cdot u = \begin{bmatrix} \alpha \cdot u_1 \\ \alpha \cdot u_2 \\ \dots \\ \alpha \cdot u_n \end{bmatrix} .$$

### Εσωτερικό γινόμενο διανυσμάτων:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix}, \quad u \cdot v = u_1 v_1 + u_2 v_2 + \dots + u_n v_n .$$

Τα διανυσμάτων έχουν πάρα πολλές χρήσεις στα γραφικά με υπολογιστές, με κυριότερες την αναπαράσταση σημείων και προσανατολισμού επιφανειών.

## Πίνακες

Στα γραφικά με υπολογιστές, ένας πίνακας  $m \times n$  είναι μία ορθογώνια διάταξη  $mn$  αριθμών σε  $m$  γραμμές και  $n$  στήλες. Ένας πίνακας  $A$  διαστάσεων  $m \times n$  συμβολίζεται όπως στη συνέχεια.

### Συμβολισμός πινάκων:

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \dots & \dots & \dots & \dots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mn} \end{bmatrix}, \text{ όπου } \alpha_{ij} \text{ το στοιχείο του πίνακα } A \text{ στη γραμμή } i \text{ και στη στήλη } j.$$

Ένας πίνακας διαστάσεων  $n \times n$  ονομάζεται *τετραγωνικός πίνακας τάξης  $n$* . Ο τετραγωνικός πίνακας τάξης  $n$  του οποίου όλα τα στοιχεία είναι 0 εκτός από αυτά της κύριας διαγωνίου τα οποία είναι 1 συμβολίζεται με  $I_n$  και ονομάζεται *μοναδιαίος*.

### Μοναδιαίος πίνακας:

$$I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Οι πράξεις πινάκων με τη μεγαλύτερη σημασία για τα γραφικά με υπολογιστές είναι ο πολλαπλασιασμός πίνακα με πραγματικό αριθμό και ο πολλαπλασιασμός πινάκων, ενώ μεγάλη σημασία έχουν επίσης ο ανάστροφος και ο αντίθετος πίνακας.

### Πολλαπλασιασμός πίνακα με πραγματικό αριθμό:

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \dots & \dots & \dots & \dots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mn} \end{bmatrix}, \beta \cdot A = \begin{bmatrix} \beta \alpha_{11} & \beta \alpha_{12} & \dots & \beta \alpha_{1n} \\ \beta \alpha_{21} & \beta \alpha_{22} & \dots & \beta \alpha_{2n} \\ \dots & \dots & \dots & \dots \\ \beta \alpha_{m1} & \beta \alpha_{m2} & \dots & \beta \alpha_{mn} \end{bmatrix}$$

### Πολλαπλασιασμός πινάκων:

Για να είναι εφικτός ο πολλαπλασιασμός δύο πινάκων  $A, B$  πρέπει οι διαστάσεις του  $A$  να είναι  $m \times p$  και οι διαστάσεις του  $B$  να είναι  $p \times n$ , δηλαδή, ο αριθμός των στηλών του  $A$  να ισούται με τον αριθμό των γραμμών του  $B$ . Το γινόμενο των  $A, B$  είναι ένας πίνακας διαστάσεων  $m \times n$ .

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1p} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2p} \\ \dots & \dots & \dots & \dots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mp} \end{bmatrix}, B = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2n} \\ \dots & \dots & \dots & \dots \\ \beta_{p1} & \beta_{p2} & \dots & \beta_{pn} \end{bmatrix}, C = A \cdot B = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix},$$

όπου  $c_{ij} = \sum_{k=1}^p \alpha_{ik} \cdot \beta_{kj}, 1 \leq i \leq m, 1 \leq j \leq n.$

**Αντίθετος πίνακας:**

Ο αντίθετος ενός τετραγωνικού πίνακα  $A$  τάξης  $n$  είναι ένας τετραγωνικός πίνακας  $A^{-1}$  τάξης  $n$  για τον οποίο ισχύει  $A \cdot A^{-1} = A^{-1} \cdot A = I_n$ .

**Ανάστροφος πίνακας:**

Ο ανάστροφος ενός πίνακα  $A$  διαστάσεων  $m \times n$  είναι ένας πίνακας  $A^T$  διαστάσεων  $n \times m$  του οποίου τα στοιχεία των γραμμών είναι τα αντίστοιχα στοιχεία των στηλών του  $A$ .

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \dots & \dots & \dots & \dots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mn} \end{bmatrix}, A^T = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1m} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2m} \\ \dots & \dots & \dots & \dots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nm} \end{bmatrix},$$

Οι πίνακες έχουν πάρα πολλές χρήσεις στα γραφικά με υπολογιστές, με κυριότερες την αναπαράσταση γεωμετρικών μετασχηματισμών.



## **Εικονική Πραγματικότητα**

Η φυσική αναπαράσταση του εικονικού κόσμου περιγράφει την οπτική και ακουστική του παρουσίαση, όπως επίσης και, σε αρκετές περιπτώσεις, την απευθυνόμενη σε άλλες ανθρώπινες αισθήσεις (αφή, κ.τ.λ.) μορφοποίησή του. Επιπλέον, καθορίζει την σκελετική (ή την όποιας άλλης προσέγγισης) δομή των συστατικών του (π.χ. σε περιπτώσεις εικονικών σωμάτων και άλλων σύνθετων αντικειμένων), όπως επίσης και τη λογική μεταβολής και εξέλιξης των παραπάνω στο χρόνο. Έτσι, δεν αποτελεί μόνο ένα μέσο παρουσίασης του εικονικού κόσμου, αλλά μία πλήρη και ακριβή περιγραφή της χωρικής του υπόστασης και της δομής του. Η φυσική αναπαράσταση του εικονικού κόσμου είναι, ουσιαστικά, η εικονική μεταφορά της φυσικής υπόστασης την οποία θα είχε ο αναπαριστώμενος κόσμος αν αποτελούσε μέρος του φυσικού κόσμου.

## **Θεμελιώδη φυσιολογικά χαρακτηριστικά**

Ο τρόπος με τον οποίο αντιλαμβανόμαστε τις διάφορες μορφές πληροφορίας που ενσωματώνονται σε ένα Σύστημα Πολυμέσων είναι μέσα από τις αισθήσεις μας. Το κεφάλαιο αυτό θα παρουσιάσει ορισμένα θεμελιώδη φυσιολογικά χαρακτηριστικά των αισθήσεων μας με έμφαση στην όραση και στην ακοή μιας και σε αυτές απευθύνονται οι περισσότερες εφαρμογές πολυμέσων. Οποιοδήποτε αισθητηριακό ερέθισμα μπορεί να παρασταθεί σαν ένα σύνολο μεταβλητών η τιμή των οποίων είναι συνάρτηση του χρόνου ή του χώρου. Για παράδειγμα κάθε ήχος μπορεί να περιγραφεί από τη μεταβολή της πίεσης στο τύμπανο του παρατηρητή. Η συχνότητα με την οποία μεταβάλλεται η πίεση καθορίζει τη συχνότητα του ήχου που ακούμε ενώ το μέγεθος της μεταβολής καθορίζει την ένταση του.

## **Ακοή και όραση**

Το αυτί μας είναι τρομερά ικανό να διακρίνει μεταξύ ενός συνόλου ηχητικών σημάτων που δέχεται συγχρόνως όπως και να αντιλαμβάνεται πολύ μικρές διαφορές στη λήψη του ιδίου σήματος. Για παράδειγμα το αυτί μας είναι ικανό να παρακολουθεί δύο ή περισσότερες συζητήσεις συγχρόνως. Επίσης είναι ικανό να αντιλαμβάνεται διακοπές στην παραγωγή ενός ηχητικού σήματος που αρχίζουν από τα 40 ms. Τέλος το αυτί μας αντιλαμβάνεται παροδικές παραμορφώσεις συχνότητας που αρχίζουν από το 10% της εκπεμπόμενης συχνότητας. Οι τρομερές αυτές διακριτικές ικανότητες του αυτιού μας μπορεί να γίνουν κατανοητές και από την ευκολία με την οποία αντιλαμβανόμαστε ανεπαίσθητους και ασυνήθιστους ήχους στο σπίτι μας.

Στην περίπτωση της όρασης τα πράγματα γίνονται περισσότερο πολύπλοκα. Ας υποθέσουμε ότι

παρατηρούμε μια επίπεδη δισδιάστατη εικόνα που αποτελείται μόνο από αποχρώσεις του ίδιου χρώματος (π.χ. αποχρώσεις του μπλε). Μια τέτοια εικόνα ανακλά ηλεκτρομαγνητική ακτινοβολία που προσπίπτει πάνω της από το περιβάλλον και οι ανακλάσεις που καταλήγουν στο μάτι του παρατηρητή αποτελούν το φως που προέρχεται από την εικόνα. Η ένταση του φωτός είναι συνάρτηση της θέσης στο χώρο του σημείου της εικόνας από το οποίο έχει ανακλαστεί. Η εικόνα λοιπόν που εξετάζουμε μπορεί να περιγραφεί από τη τιμή της έντασης του φωτός που προέρχεται από την εικόνα και που είναι συνάρτηση της θέσης των σημείων της εικόνας στο χώρο.

Αν η εικόνα που παρατηρούμε δεν είναι μονοχρωματική αλλά ανακλά περισσότερα από ένα προσπίπτοντα μήκη κύματος τότε η περιγραφή της θα αποτελείται από τη σύζευξη των εντάσεων όλων των μηκών που ανακλώνται από τη εικόνα δηλαδή από μια συνάρτηση έντασης ( $C(\lambda)$ ) για καθένα από τα μήκη κύματος που ανήκουν στο φάσμα των μηκών κύματος που ανακλώνται από την εικόνα. Ευτυχώς για μας δε χρειάζεται να χρησιμοποιήσουμε τόσο πολύπλοκες περιγραφές της αίσθησης που μας δίνει κάθε εικόνα μιας και η θεωρία χρωμάτων καθορίζει ότι η αίσθηση οποιαδήποτε χρώματος μπορεί να αναπαραχθεί με την ανάμιξη σε κατάλληλες αναλογίες τριών κύριων μονοχρωματικών πηγών φωτός. Κατά σύμβαση η τριάδα των χρωμάτων που χρησιμοποιείται για το σκοπό αυτό αποτελείται από το κόκκινο (μήκος κύματος  $\lambda = 700$  nm), το πράσινο ( $\lambda=546,1$  nm) και το μπλε ( $\lambda=435.8$  nm). Χρησιμοποιώντας αυτή τη τριάδα κάθε επίπεδη έγχρωμη εικόνα μπορεί να παρασταθεί από ένα σύνολο τριών δισδιάστατων συναρτήσεων έντασης.

Πέρα από το γεγονός ότι αντιλαμβανόμαστε διαφορές στο μήκος κύματος του φωτός σε διαφορετικά χρώματα, το μάτι μας είναι περισσότερο ευαίσθητο σε ορισμένα μήκη κύματος από ότι σε άλλες. Για παράδειγμα μονοχρωματικές εικόνες διαφορετικών χρωμάτων με την ίδια όμως ένταση μπορεί να δημιουργήσουν διαφορετικές εντυπώσεις «λαμπρότητας». Για παράδειγμα το μάτι μας είναι πιο ευαίσθητο στο κίτρινο ή στο πρασινοκίτρινο χρώμα από ότι στο κόκκινο ή στο μπλε.

Η χρωματική αίσθηση μιας πηγής καθορίζεται όμως και από τρία άλλα χαρακτηριστικά: την *απόχρωση* (*lightness*), τη *λαμπρότητα* (*brightness*) και τη *φωτεινότητα* (*;*) (*luminance*).

Η απόχρωση μετρά την απόκριση του ματιού στο σύνολο του φάσματος των μηκών κύματος που περιέχονται σε μια πηγή. Ουσιαστικά η απόχρωση μετρά την απόκριση του ματιού στο σύνολο της φωτεινής ενέργειας που εκπέμπεται από μια πηγή.

Η φωτεινότητα, από την άλλη μεριά, μετρά την αίσθηση της “ασπρίλας” που αντιλαμβανόμαστε από μια φωτισμένη εικόνα. Με άλλα λόγια αντιστοιχεί στην αίσθηση ότι ένα αντικείμενο αντανακλά περισσότερο ή λιγότερο από το φως που προσπίπτει πάνω του. Η φωτεινότητα είναι πιο εύκολο να περιγράψει την αίσθηση του “μαύρου” ή του “άσπρου” που παράγουν άχρωμα

αντικείμενα, δηλαδή αντικείμενα που ανακλούν σε ίσες ποσότητες όλα τα μήκη κύματος που δέχονται. Στην περίπτωση αυτή, άχρωμα αντικείμενα που ανακλούν λιγότερο από το 30% του προσπίπτοντος φωτός μας φαίνονται μαύρα, ενώ αυτά που ανακλούν περισσότερο από το 80% του φωτός μας φαίνονται άσπρα. Μεταξύ των δύο αυτών άκρων το χρώμα των αντικειμένων αυτών λαμβάνει διάφορες αποχρώσεις του γκρι.

Τέλος για να αντιληφθούμε την αίσθηση που μας δίνει η λαμπρότητα ενός αντικειμένου αρκεί να φανταστούμε μια σελίδα ομοιόμορφου γκρι χρώματος. Η σελίδα αυτή έχει μια συγκεκριμένη και σταθερή φωτεινότητα. Αν φωτίσουμε το κέντρο της σελίδας με ένα φακό τότε η περιοχή που θα φωτιστεί θα μας φανεί πιο λαμπρή χωρίς όμως να μεταβληθεί η φωτεινότητα της. Η λαμπρότητα λοιπόν ενός αντικειμένου είναι ένα ακόμα χαρακτηριστικό της χρωματικής αίσθησης που υποδηλώνει ότι ένα αντικείμενο εκπέμπει περισσότερο ή λιγότερο φως. Η ιδιότητα αυτή εξαρτάται από το περιβάλλον φως και όχι από τις ιδιότητες αυτού καθαυτού του αντικειμένου.

Για τον τρόπο παραγωγής χρωμάτων από άλλα ισχύουν τα παρακάτω:

1. Οποιοδήποτε χρώμα μπορεί να παραχθεί με την ανάμιξη κατάλληλων ποσοτήτων από όχι περισσότερα από τρία οποιαδήποτε αλλά διαφορετικά μεταξύ τους χρώματα. Ο μόνος περιορισμός που θα πρέπει να ισχύει είναι ότι κανένα από τα χρώματα στη τριάδα δεν έχει παραχθεί από την ανάμιξη των άλλων δυο.
2. Αν σε δυο χρώματα που μας φαίνονται τα ίδια προσθέσουμε την ίδια ποσότητα του ίδιου χρώματος τότε τα χρώματα που θα παραχθούν θα συνεχίζουν να είναι τα ίδια.

Σε αντίθεση με το αντί μας το μάτι μας δεν έχει σημαντική διακριτική ικανότητα. Τρία παραδείγματα μπορούν να μας βοηθήσουν να κατανοήσουμε τους περιορισμούς της όρασης μας:

1. Είναι εξαιρετικά δύσκολο να αντιληφθούμε ως διαφορετικές δύο ή περισσότερες εικόνες που επικαλύπτονται.
2. Περιοδικές παραμορφώσεις στη συχνότητα ή παύσεις μετάδοσης εικόνας που μπορεί να φτάσουν μέχρι και δευτερόλεπτα περνούν απαρατήρητες κατά τη διάρκεια παρακολούθησης ενός βίντεο.
3. Σε αντίθεση με τη δυνατότητα αντίληψης ανεπαίσθητων αλλά ασυνήθιστων ήχων στο σπίτι μας πολλές φορές αλλαγές στη διαρρύθμιση του σπιτιού (π.χ. ξεκρέμασμα ενός κάδρου) περνούν απαρατήρητες για αρκετό καιρό.

Κατά συνέπεια η σχεδίαση μιας εφαρμογής πολυμέσων θα πρέπει να λάβει υπόψη της το γεγονός ότι οι άνθρωποι είναι πιο ευαίσθητοι σε παραμορφώσεις ηχητικών παρά οπτικών σημάτων. Κατά συνέπεια εφαρμογές που συμπιέζουν ή μεταδίδουν σε δίκτυα ηχητικά ή/και οπτικά σήματα θα



πρέπει να είναι περισσότερο προσεκτικοί ώστε οι παραμορφώσεις και καθυστερήσεις που εισάγονται από τη συμπίεση και μετάδοση ηχητικών σημάτων να είναι μέσα σε ανεκτά όρια.

Πέρα όμως από τα φυσιολογικά χαρακτηριστικά που ασφαλώς θα πρέπει να λαμβάνονται υπ' όψη κατά τη σχεδίαση μιας εφαρμογής πολυμέσων το μεγάλο ερώτημα είναι εάν οι τρόποι παρουσίασης των πληροφοριών και συμμετοχής των χρηστών που υλοποιούνται στα Συστήματα Πολυμέσων διευκολύνουν πραγματικά τους χρήστες στην επίτευξη των πληροφοριακών σκοπών τους. Δυστυχώς δεν έχει δοθεί ακόμα ικανοποιητική απάντηση στο ερώτημα αυτό έστω και αν οι ενδείξεις που υπάρχουν είναι ενθαρρυντικές.

Πιο συγκεκριμένα ορισμένες έρευνες έχουν καταδείξει ότι οι άνθρωποι αποκτούν περίπου το 80% των “γνώσεων” τους μέσω της όρασης, το 11% μέσω της ακοής, το 3.5% μέσω της όσφρησης και μεταξύ 1% με 1,5% μέσω της γεύσης και της αφής. Επιπλέον, άλλες έρευνες έχουν καταδείξει ότι οι άνθρωποι απομνημονεύουν μόλις το 20% των οπτικών πληροφοριών που δέχονται, μεταξύ του 20% και 30% των ηχητικών πληροφοριών, μεταξύ του 40% με 50% των πληροφοριών που τους διαβιβάζονται με ταυτόχρονη οπτική και ηχητική μορφή και περίπου 80% των πληροφοριών με τις οποίες αλληλεπιδρούν τόσο με το οπτικό όσο και με το ηχητικό τους περιεχόμενο. Θεωρώντας ότι οι έρευνες αυτές είναι σωστές (πράγμα που είναι συζητήσιμο) καταλήγουμε στα εξής συμπεράσματα:

1. Οι οπτικές μορφές πληροφορίας αποτελούν τη σπουδαιότερη μέθοδο ανάκτησης πληροφορίας για τους ανθρώπους. Επομένως ένα σύστημα πολυμέσων θα πρέπει να εκμεταλλεύεται στο έπακρο τις οπτικές μορφές πληροφορίας που διαθέτει.
2. Οι ηχητικές πηγές πληροφορίας παρόλο που είναι λιγότερο διαδεδομένες απομνημονεύονται με μεγαλύτερη ευκολία. Επομένως οι ηχητικές πηγές πληροφορίας μπορεί να αποτελέσουν αποτελεσματική μέθοδο ανάκτησης πληροφορίας.
3. Η ανάμιξη των οπτικών και ηχητικών πληροφοριών διευκολύνει την απομνημόνευση πληροφοριών.
4. Η ενεργός συμμετοχή του χρήστη σε αυτό που βλέπει και ακούει διευκολύνει τη απομνημόνευση και κατανόηση της παρουσιαζόμενης πληροφορίας.

## **Η αίσθηση του βάθους στην όραση**

Η ανθρώπινη όραση χρησιμοποιεί δύο μηχανισμούς για να υπολογίζει τη σχετική απόσταση των αντικειμένων στο χώρο. Ο πρώτος μηχανισμός βασίζεται στη σύγκλιση μεταξύ των ευθειών παρατήρησης για το καθένα από τα δύο μάτια. Πιο συγκεκριμένα όταν παρατηρούμε ένα αντικείμενο από κοντά οι ευθείες παρατήρησης του αντικειμένου για το κάθε μάτι μας δεν είναι

παράλληλες αλλά συγκλίνουν πάνω στο αντικείμενο. Όσο πιο κοντά βρίσκεται το αντικείμενο τόσο μεγαλύτερος βαθμός σύγκλισης είναι απαραίτητος για την παρατήρηση του. Όσο μεγαλύτερος όμως είναι και ο βαθμός σύγκλισης τόσο και μεγαλύτερη είναι η πίεση που εξασκούν οι μύες του ματιού για να το υποχρεώσουν να κινηθεί προς την επιθυμητή κατεύθυνση. Ο εγκέφαλος μετρά την πίεση των μυών του ματιού και έτσι λαμβάνει μια ένδειξη της απόστασης του αντικειμένου από το μάτι μας. Η δεύτερη ένδειξη που χρησιμοποιεί ο εγκέφαλος μας προέρχεται από την απόκλιση μεταξύ της εικόνας του κόσμου που προέρχεται από το αριστερό και το δεξί μας μάτι. Πιο συγκεκριμένα όταν τα μάτια μας εστιάζουν στο ίδιο σημείο του χώρου η εικόνα που μεταδίδουν στον εγκέφαλο διαφέρει καθώς η εικόνα του δεξιού ματιού είναι μετατοπισμένη προς τα δεξιά και η εικόνα του αριστερού ματιού προς τα αριστερά. Όταν εστιάζουμε σε ένα μακρινό αντικείμενο η διαφορά μεταξύ των δύο αυτών εικόνων είναι αμελητέα ενώ όσο το αντικείμενο πλησιάζει προς το μάτι μας η διαφορά αυτή αυξάνεται. Ο εγκέφαλος συνδυάζει τη πληροφορία αυτή με την πίεση των μυών του ματιού για να δημιουργήσει την αίσθηση του βάθους.

## Τρισδιάστατος ήχος

Η αίσθηση της κατεύθυνσης από την οποία προέρχεται ένας ήχος είναι μια διαδικασία που βασίζεται στη μέτρηση της έντασης και την ανάλυση των συχνοτήτων του ηχητικού σήματος που αντιλαμβανόμαστε. Πιο συγκεκριμένα όταν ακούμε κάτι η ένταση του ήχου που μετριέται στο καθένα από τα αυτιά μας διαφέρει ανάλογα με την πλευρά (δεξιά ή αριστερά) από την οποία προέρχεται ο ήχος. Ένας ήχος που προέρχεται από τα δεξιά θα έχει μεγαλύτερη ένταση στο δεξί από ότι στο αριστερό μας αυτί και αντιστρόφως. Επιπλέον θα υπάρχει και μια μικρή χρονική διαφορά μεταξύ των χρόνων που ο ίδιος ήχος γίνεται αντιληπτός από το καθένα αυτί μας. Ο συνδυασμός της έντασης και της χρονικής διαφοράς λήψης του ήχου από το καθένα αυτί μας χρησιμοποιείται από τον εγκέφαλο για τον καθορισμό της πλευράς από την οποία προήλθε ο ήχος αυτός.

Έχοντας εντοπίσει την πλευρά από την οποία προέρχεται ο ήχος, ο εγκέφαλος θα πρέπει να καθορίσει και την κατεύθυνση εκπομπής του (πάνω, κάτω, μπροστά, πίσω). Ο προσδιορισμός αυτός βασίζεται στην ικανότητα του αυτιού μας να μετασχηματίζει τον ήχο που λαμβάνει ανάλογα με την κατεύθυνση εκπομπής του. Ο μετασχηματισμός αυτός αποτελείται από το φιλτράρισμα ορισμένων συχνοτήτων ανάλογα με την κατεύθυνση εκπομπής.

Ένα σύστημα εικονικής πραγματικότητας εκμεταλλεύεται τα χαρακτηριστικά της ανθρώπινης ακοής που προαναφέραμε για να δημιουργήσει τη ψευδαίσθηση του τρισδιάστατου ήχου. Η αληθοφάνεια μιας τέτοιας διεργασίας βελτιώνεται και με την εξομοίωση των ακουστικών

χαρακτηριστικών του εικονικού χώρου από τον οποίο εκπέμπεται ο ήχος όπως π.χ. με τον υπολογισμό των διαφόρων ανακλάσεων που θα έχει ένα ηχητικό σήμα στα διάφορα αντικείμενα σε ένα εικονικό κόσμο.

## **Γράφημα σκηνής**

Ο χώρος των Γραφικών με Υπολογιστές έχει να επιδείξει πλήθος προσεγγίσεων ικανών να αντιμετωπίσουν το πρόβλημα της φυσικής αναπαράστασης εικονικών κόσμων. Από τις πλέον διαδεδομένες και ευρέως χρησιμοποιούμενες σε εφαρμογές εικονικής πραγματικότητας και εικονικών περιβαλλόντων είναι η *αναπαράσταση γραφήματος σκηνής* (*scene-graph* ή *scenegraph representation*).

Το γράφημα σκηνής είναι μία δενδροειδής ιεραρχική δομή η οποία περιγράφει πλήρως μία σκηνή. Τα γραφήματα σκηνής αφορούν, κατά κανόνα, τρισδιάστατες σκηνές (εντούτοις, οι αρχές και οι έννοιες στις οποίες βασίζονται έχουν γενικότερη ισχύ). Για τη μορφοποίηση ενός στιγμιότυπου της σκηνής, το υποσύστημα γραφικών διατρέχει ολόκληρο το γράφημα σκηνής ξεκινώντας από τη ρίζα, διαδικασία η οποία επαναλαμβάνεται ανά καρέ μορφοποίησης. Οι κόμβοι ενός γραφήματος σκηνής είναι διαφόρων τύπων, με κυριότερο σημείο διαφοροποίησης τη θέση τους στο γράφημα: οι κόμβου-φύλλα συνήθως αναπαριστούν απλά, πρωταρχικά στοιχεία γεωμετρίας (σημείο, ευθύγραμμο τμήμα, τρίγωνο, κ.τ.λ.), σύνθετα στοιχεία γεωμετρίας (σφαίρα, κύβος, κύλινδρος, επιφάνεια, κ.τ.λ.), χαρακτηριστικά απεικόνισης (υλικά, υφή, φωτισμός, κ.τ.λ.) και άλλα στοιχεία με συγκεκριμένη αναπαράσταση (ήχος, σημεία αλληλεπίδρασης, κίνηση, κ.τ.λ.): οι εσωτερικοί κόμβοι ομαδοποιούν μέρη του γραφήματος σε συγκεκριμένα υποδέντρα, στα οποία μπορούν – ανάλογα με τον τύπο τους – να εφαρμόζουν συγκεκριμένους χωρικούς μετασχηματισμούς (μετατόπιση, περιστροφή, κλιμάκωση).

Η δενδροειδής δομή του γραφήματος σκηνής και η ικανότητά του να εφαρμόζει μετασχηματισμούς επικεντρωμένους σε συγκεκριμένα μέρη του, το καθιστά μία ιδανική προσέγγιση για την αναπαράσταση σκηνών στις οποίες είναι απαραίτητη, εκτός από την απλή απεικόνιση, η μοντελοποίηση της δομής του εικονικού κόσμου και η λογική ομαδοποίηση των συστατικών του. Για τον ίδιο λόγο, το γράφημα σκηνής είναι η προτιμητέα επιλογή για εφαρμογές με κίνηση (animation) και εντοπισμένη αλληλεπίδραση με συγκεκριμένα, μεμονωμένα μέρη του εικονικού κόσμου.

## **X3D/VRML**

Το X3D (eXtensible 3D graphics) είναι μία αρχιτεκτονική αναπαράστασης τρισδιάστατων σκηνών

βασισμένη στην έννοια του γραφήματος σκηνής. Τα κύρια χαρακτηριστικά του είναι τα παρακάτω:

- ✓ Είναι αποσυνδεδεμένο από την υλοποίηση καθώς είναι προσανατολισμένο αποκλειστικά στην εννοιολογική περιγραφή της σκηνής, γεγονός που του δίνει μεγάλη μεταφερσιμότητα και αυξάνει τις επιλογές υλοποίησης.
- ✓ Βασίζεται σε μία ολοκληρωμένη και λεπτομερή προδιαγραφή και έτσι παράγει συνεπή και ομοιόμορφα αποτελέσματα ανεξάρτητα – σε μεγάλο βαθμό – από την υλοποίηση.
- ✓ Υποστηρίζει διάφορες μορφές κωδικοποίησης αρχικών δεδομένων, έχοντας τη δυνατότητα να επεξεργαστεί ορισμούς σκηνών σε XML όσο και με βάση την παλαιότερη γλώσσα VRML 2.0 (ή VRML97, όπως είναι επίσης γνωστή), της οποίας αποτελεί διάδοχο. Επιπλέον, αναγνωρίζει δεδομένα σε συμπιεσμένη μορφή.
- ✓ Είναι τμηματοποιημένο, επιτρέποντας σε υλοποιήσεις να ενσωματώσουν μόνο τα μέρη εκείνα της λειτουργικότητάς του που τις αφορούν.
- ✓ Υποστηρίζει, σε επίπεδο προδιαγραφής, μορφοποίηση σε πραγματικό χρόνο.
- ✓ Είναι ανοικτής αρχιτεκτονικής και διατίθεται δωρεάν.

Όπως τα περισσότερα γραφήματα σκηνής, το X3D διαθέτει τύπους κόμβων για τα συνηθέστερα συστατικά μίας σκηνής, όπως χωρικούς μετασχηματισμούς (π.χ. Transform), πρωταρχικά γεωμετρικά σχήματα (π.χ. Box, Sphere, Cylinder και Cone), χαρακτηριστικά υλικών, σκίασης και υφής (π.χ. Appearance, Material και ImageTexture), πλέγματα και επιφάνειες (π.χ. IndexedLineSet και IndexedFaceSet), ήχο και πολυμέσα (π.χ. Sound, AudioClip και MovieTexture), φωτισμό (π.χ. PointLight και DirectionalLight), κ.τ.λ. Επιπλέον, διαθέτει εξειδικευμένους κόμβους για αλληλεπίδραση με το χρήστη (κόμβους τύπου «Sensor»), παρεμβολή τιμών (κόμβους τύπου «Interpolator»), ορισμό συμπεριφοράς με χρήση των γλωσσών JavaScript και Java (Script), ορισμό NURB καμπύλων και επιφανειών, σκηνές με γεωαναφορά, κ.ά. Ακόμα, διαθέτει τη δυνατότητα ορισμού μεταδεδομένων ανά κόμβο, κάτι που έχει μεγάλη σημασία για τον ορισμό εικονικών κόσμων. Τέλος, χάρη στην παλαιότερη εξάπλωση της VRML, στο Internet υπάρχει μεγάλη διαθεσιμότητα μοντέλων, τόσο υπό τη μορφή εμπορικών προϊόντων όσο και δωρεάν.

Χάρη στα παραπάνω χαρακτηριστικά, το πρότυπο X3D, είναι ικανό να περιγράψει εικονικούς κόσμους με μεγάλη γεωμετρική ακρίβεια, ικανοποιητικά επίπεδα απεικόνισης και αυξημένες δυνατότητες συμπεριφοράς και αλληλεπίδρασης. Σε συνδυασμό με ένα ολοκληρωμένο μηχανισμό εξωτερικού ελέγχου της σκηνής («scene authoring interface», SAI) τον οποίο παρέχει, αποτελεί μία ελκυστική επιλογή για το υπόβαθρο φυσικής αναπαράστασης εικονικών κόσμων σε εικονικά περιβάλλοντα. Για το λόγο αυτό, έχει ήδη χρησιμοποιηθεί σε αρκετά εικονικά περιβάλλοντα.

## Java3D

Το Java3D είναι ένα γενικής χρήσης γράφημα σκηνής υλοποιημένο σε Java. Τα κύρια χαρακτηριστικά του είναι τα παρακάτω:

- ✓ Παρέχει τη δυνατότητα αξιοποίησης διαφόρων βιβλιοθηκών χαμηλού επιπέδου για μορφοποίηση γραφικών, όπως OpenGL και Direct3D, ανάλογα με την πλατφόρμα.
- ✓ Σε αντίθεση με τα OpenSceneGraph και OpenSG, δεν αποτελεί ένα διαμεσολαβητή μεταξύ της εφαρμογής και της βιβλιοθήκης μορφοποίησης· αντίθετα, υιοθετεί μία πλήρως αντικειμενοστρεφή προσέγγιση, προσανατολισμένη στη σκηνή και όχι στο υπόβαθρο μορφοποίησης.
- ✓ Διαθέτει εγγενή υποστήριξη για ήχο με χωρικά χαρακτηριστικά, όπως επίσης και για εξοπλισμό απτικής αλληλεπίδρασης και στερεοσκοπικής προβολής, συμπεριλαμβανομένων συστημάτων CAVE και HMDs.
- ✓ Υποστηρίζει προηγμένες τεχνικές μορφοποίησης και σκίασης (shaders), ενώ λειτουργεί τόσο σε άμεση όσο και διατηρούμενη κατάσταση.
- ✓ Μπορεί να επεξεργαστεί αρχεία ορισμού σκηνών σε διάφορες μορφές, συμπεριλαμβανομένων VRML και X3D.
- ✓ Είναι, πλέον, ανοικτής αρχιτεκτονικής, διατίθεται δωρεάν ενώ υποστηρίζεται από ομάδες χρηστών.

Όπως τα περισσότερα γραφήματα σκηνής, το Java3D διαθέτει τύπους κόμβων για τα συνηθέστερα συστατικά μίας σκηνής, όπως χωρικούς μετασχηματισμούς (π.χ. Transform), γεωμετρία (π.χ. Shape3D), χαρακτηριστικά υλικών, σκίασης και υφής (π.χ. Appearance, Material, Texture2D και Texture3D), ήχο και πολυμέσα (π.χ. Sound, AudioClip και AuralAttributes), φωτισμό (π.χ. PointLight και DirectionalLight), κ.τ.λ. Επιπλέον, διαθέτει εξειδικευμένους κόμβους για αλληλεπίδραση με το χρήστη (InputDevice και Sensor), παρεμβολή τιμών (κόμβους τύπου «Interpolator»), κ.ά. Τέλος, χάρη στον μεγάλο αριθμό διαθέσιμων μετατροπέων και την ποικιλία μορφών αρχικών δεδομένων τις οποίες υποστηρίζει, ωφελείται από τη μεγάλη διαθεσιμότητα μοντέλων στο Internet, τόσο υπό τη μορφή εμπορικών όσο και δωρεάν προϊόντων.

Χάρη στα παραπάνω χαρακτηριστικά, το Java3D είναι ικανό να καλύψει το σύνολο των υπολογιστικών αναγκών που άπτονται της φυσικής αναπαράστασης εικονικών κόσμων σε εικονικά περιβάλλοντα. Σε συνδυασμό με το γεγονός ότι είναι ένα σύγχρονο προϊόν με ιδιαίτερα ευρεία και έντονη υποστήριξη, μπορεί να σταθεί ως μία ολοκληρωμένη και πολυεπίπεδης λειτουργικότητας επιλογή.

## OpenSceneGraph

Το OpenSceneGraph είναι ένα πλαίσιο ανάπτυξης εφαρμογών γραφικών υψηλής απόδοσης βασισμένο στην έννοια του γραφήματος σκηνής. Τα κύρια χαρακτηριστικά του είναι τα παρακάτω:

- ✓ Στηρίζεται στη γλώσσα προγραμματισμού C++ και στη βιβλιοθήκη OpenGL. Σαν αποτέλεσμα, δεσμεύει σε σημαντικό βαθμό το χρήστη με προκαθορισμένες επιλογές υλοποίησης, με τις οποίες αρνητικές συνέπειες έχει μία τέτοια δέσμευση να αντισταθμίζονται από την αυξημένη απόδοση, τις προηγμένες δυνατότητες και την ευρεία μεταφερσιμότητα των συγκεκριμένων επιλογών.
- ✓ Υποστηρίζει μία μεγάλη ποικιλία μορφών αρχείων και βάσεων δεδομένων για τον ορισμό σκηνών, συμπεριλαμβανομένων και των X3D/VRML. Οι δυνατότητές του σε αυτό τον τομέα είναι επεκτάσιμες μέσω εξειδικευμένων τμημάτων λογισμικού (plugins).
- ✓ Επιτρέπει σε υλοποιήσεις να ενσωματώνουν εξειδικευμένα υποσύνολα της λειτουργικότητάς του (NodeKits): συστήματα σωματιδίων (particle systems), σκιά, έδαφος (terrain), κίνηση (animation), μορφοποίηση όγκου (volume rendering), κ.ά.
- ✓ Εκμεταλλεύεται την εξάρτησή του από την OpenGL για να παρέχει ιδιαίτερα υψηλές επιδόσεις όπως επίσης και υποστήριξη για τις προηγμένες δυνατότητες μορφοποίησης της συγκεκριμένης βιβλιοθήκης, συμπεριλαμβανομένων των πιο πρόσφατων επεκτάσεων (OpenGL extensions).
- ✓ Είναι ανοικτής αρχιτεκτονικής και διατίθεται δωρεάν.

Όπως τα περισσότερα γραφήματα σκηνής, το OpenSceneGraph διαθέτει τύπους κόμβων για τα συνηθέστερα συστατικά μίας σκηνής, όπως χωρικούς μετασχηματισμούς (π.χ. `osg::Transform`), γεωμετρία (π.χ. `osg::Geometry`, `osg::DVRTriangle` και `osg::DVRSlice`), χαρακτηριστικά υλικών, σκίασης και υφής (π.χ. `osg::SimpleMaterial`, `osg::PhongMaterial`, `osg::FresnelMaterial` και `osg::SimpleTexturedMaterial`), πλέγματα και επιφάνειες (π.χ. `osg::Surface` και `osg::ExtrusionSurface`), φωτισμό (π.χ. `osg::PointLight` και `osg::DirectionalLight`), κ.τ.λ., καθώς και βοηθητικές συναρτήσεις δημιουργίας απλών γεωμετρικών σχημάτων. Ακόμα, χάρη στην μεγάλη ποικιλία μορφών αρχικών δεδομένων τις οποίες υποστηρίζει, ωφελείται από τη μεγάλη διαθεσιμότητα μοντέλων στο Internet, τόσο υπό τη μορφή εμπορικών προϊόντων όσο και δωρεάν.

Χάρη στα παραπάνω χαρακτηριστικά, το OpenSceneGraph, είναι ικανό να περιγράψει εικονικούς κόσμους με μεγάλη γεωμετρική ακρίβεια και εντυπωσιακά επίπεδα απεικόνισης. Σε συνδυασμό με κώδικα επιπέδου εφαρμογής σε C++ για κίνηση, συμπεριφορά και αλληλεπίδραση, αποτελεί μία ολοκληρωμένη και υψηλής απόδοσης επιλογή για το υπόβαθρο φυσικής αναπαράστασης εικονικών κόσμων σε εικονικά περιβάλλοντα.

## OpenSG

Το OpenSG είναι ένα υψηλής απόδοσης γράφημα σκηνής. Τα κύρια χαρακτηριστικά του είναι τα παρακάτω:

- ✓ Δίνει μεγάλη έμφαση σε παράγοντες όπως η επεκτασιμότητα, η μεταφερσιμότητα, η ασφάλεια σε πολυνηματικές υλοποιήσεις και οι δυνατότητες ομαδοποίησης (clustering), με κύρια πλεονεκτήματα τη δυνατότητα μορφοποίησης πολύ μεγάλων ή σύνθετων σκηνών όπως επίσης και τη δυνατότητα συνεργασίας με συστήματα απεικόνισης μεγάλων διαστάσεων.
- ✓ Στηρίζεται στη βιβλιοθήκη OpenGL. Σαν αποτέλεσμα, επιβάλλει μία προκαθορισμένη επιλογή υλοποίησης, με τις όποιες αρνητικές συνέπειες έχει μία τέτοια δέσμευση να αντισταθμίζονται από την αυξημένη απόδοση, τις προηγμένες δυνατότητες και την ευρεία μεταφερσιμότητά της.
- ✓ Υποστηρίζει μία μεγάλη ποικιλία μορφών αρχείων για τον ορισμό σκηνών, συμπεριλαμβανομένων και των X3D/VRML. Επίσης, ορίζει τη μορφή αρχείου BIN η οποία είναι δυαδική και, έτσι, επιτρέπει πολύ γρηγορότερη επεξεργασία σε σχέση με τις υπόλοιπες μορφές αρχείων, αλλά, καθώς είναι μία ad hoc προσέγγιση που αφορά αποκλειστικά το OpenSG, δεν διακρίνεται από ανάλογη διαθεσιμότητα μοντέλων και συνεργασία μεταξύ πακέτων σχεδίασης.
- ✓ Επικεντρώνεται περισσότερο στη δομή του γραφήματος σκηνής, με στόχο την υψηλή απόδοση κατά τη μορφοποίηση και την βελτιστοποιημένη αναζήτηση, διατηρώντας την ανά κόμβο πληροφορία (π.χ., γεωμετρία, μετασχηματισμούς, κ.τ.λ.) σε ανεξάρτητες, επαναχρησιμοποιήσιμες δομές (cores).
- ✓ Είναι ανοικτής αρχιτεκτονικής και διατίθεται δωρεάν.

Όπως τα περισσότερα γραφήματα σκηνής, το OpenSceneGraph διαθέτει τύπους κόμβων για τα συνηθέστερα συστατικά μίας σκηνής, όπως χωρικούς μετασχηματισμούς (π.χ. `osg::Transform`), γεωμετρία (π.χ. `osg::Geometry`), χαρακτηριστικά υλικών, σκίασης και υφής (π.χ. `osg::SimpleMaterial`), φωτισμό (π.χ. `osg::PointLight` και `osg::DirectionalLight`), κ.τ.λ., καθώς και βοηθητικές συναρτήσεις δημιουργίας απλών γεωμετρικών σχημάτων. Ακόμα, χάρη στην μεγάλη ποικιλία μορφών αρχικών δεδομένων τις οποίες υποστηρίζει, ωφελείται από τη μεγάλη διαθεσιμότητα μοντέλων στο Internet, τόσο υπό τη μορφή εμπορικών προϊόντων όσο και δωρεάν. Είναι αντίστοιχης λογικής και μοιράζεται κάποια κοινά στοιχεία (π.χ. το namespace «`osg`») με το OpenSceneGraph που περιγράφεται παραπάνω καθώς και τα δύο λειτουργούν, ουσιαστικά, ως διαμεσολαβητές (mediators) μεταξύ της εφαρμογής και της βιβλιοθήκης OpenGL.

Χάρη στα παραπάνω χαρακτηριστικά, το OpenSG, είναι ικανό να περιγράψει εικονικούς κόσμους

με μεγάλο εύρος και πολυπλοκότητα, ενώ, ως μέρος κάποιου ολοκληρωμένου πλαισίου ανάπτυξης εφαρμογών εικονικής πραγματικότητας και με τις απαραίτητες επεκτάσεις, μπορεί να σταθεί ως μία ολοκληρωμένη και υψηλής απόδοσης επιλογή για το υπόβαθρο φυσικής αναπαράστασης εικονικών κόσμων σε εικονικά περιβάλλοντα. Για το λόγο αυτό, έχει ήδη χρησιμοποιηθεί σε αρκετές εφαρμογές εικονικών περιβαλλόντων.

## **Άλλα παραδείγματα γραφημάτων σκηνής**

Το OpenGL Performer (παλαιότερα γνωστό ως IRIS Performer) είναι ένα από τα αρχαιότερα ολοκληρωμένα γραφήματα σκηνής. Έχει αποτελέσει τη βάση για πολλές άλλες προτάσεις, όπως το OpenSceneGraph ενώ συνεχίζει να υποστηρίζεται ως εμπορικό προϊόν. Το ίδιο ισχύει και για το Open Inventor, ένα γράφημα σκηνής εξ' ίσου μεγάλης ιστορικής σημασίας.

Ιδιαίτερη μνεία αξίζει να γίνει στο PHIGS, καθώς ήταν το πρώτο ολοκληρωμένο σύστημα αναπαράστασης γραφικών για υπολογιστές με βάση την έννοια του γραφήματος σκηνής. Δημιουργήθηκε στις αρχές της δεκαετίας του 1980 ενώ αργότερα υιοθετήθηκε και ως ANSI και ISO standard. Στα επόμενα χρόνια σταμάτησε σταδιακά να χρησιμοποιείται, καθώς δεν μπόρεσε να ανταγωνιστεί την ανερχόμενη OpenGL σε θέματα απόδοσης, παρά το ότι η τελευταία υιοθετεί μία θεμελιωδώς διαφορετική προσέγγιση.

## **Αναπαραστάσεις χαμηλού επιπέδου**

Τα γραφήματα σκηνής είναι υψηλού επιπέδου αναπαραστάσεις, με λιγότερη ή περισσότερη έμφαση, σε κάθε περίπτωση, στα εννοιολογικά χαρακτηριστικά της σκηνής. Για το λόγο αυτό έχουν και ελάχιστες ως καμία εξαρτήσεις από το υλικό των υποσυστημάτων γραφικών, ήχου και αλληλεπίδρασης, στο οποίο και απευθύνονται διαφανώς, αξιοποιώντας κάποια χαμηλότερου επιπέδου αναπαράσταση, όπως είναι τα Direct3D και OpenGL.

Οι αναπαραστάσεις αυτές βασίζονται αποκλειστικά σε απλά, πρωταρχικά στοιχεία μοντελοποίησης και χωρικούς μετασχηματισμούς. Απαιτούν την επαναλαμβανόμενη και σειριακή τροφοδότηση δεδομένων στα σχετικά υποσυστήματα (λειτουργία *άμεσης κατάστασης*, *immediate mode*) σε αντίθεση με τα γραφήματα σκηνής στα οποία η μορφοποίηση της σκηνής σε κάθε καρτέ γίνεται επαναλαμβανόμενα βάσει ενός ορισμένου μοντέλου η κατάσταση του οποίου διατηρείται μεταξύ στιγμιότυπων (λειτουργία *διατηρούμενης κατάστασης*, *retained mode*).

Κατά κανόνα, αναπαραστάσεις σαν τα Direct3D και OpenGL δε διαθέτουν δομές για τον ορισμό γεωμετρικής, σκελετικής ή λογικής δομής των συστατικών της σκηνής. Χρησιμοποιούνται σπάνια απ' ευθείας, και μόνο σε περιπτώσεις όπου η εννοιολογική υπόσταση της σκηνής δεν είναι τόσο



σημαντική όσο η ταχύτητα και η απόδοση, χαρακτηριστικό εξ' ορισμού ασύμβατο με τη γενικότερη λογική των εικονικών περιβαλλόντων.

# Η Γλώσσα VRML

## Επισκόπηση της γλώσσας

Η VRML (Virtual Reality Modeling Language) αποτελεί μία μεταφέρσιμη γλώσσα περιγραφής τρισδιάστατων αντικειμένων και συνδυασμού των αντικειμένων αυτών σε σκηνές ή κόσμους. Η γλώσσα υποστηρίζει εφαρμογές εικονικής πραγματικότητας που υλοποιούνται σε προσωπικούς υπολογιστές χωρίς ειδικά συστήματα διαπροσωπείας παρά μόνο με τη χρήση ενός προγράμματος περιήγησης (browser) σε VRML. Η γλώσσα παρέχει τη δυνατότητα της γεωμετρικής περιγραφής των αντικειμένων από τα οποία αποτελείται ένας εικονικός κόσμος.

Πιο συγκεκριμένα ένα VRML αρχείο περιγράφει ένα σύνολο από αντικείμενα διατεταγμένα στο χώρο. Αρκετά από τα αντικείμενα έχουν μια τρισδιάστατη αναπαράσταση που αποτελείται από το γεωμετρικό σχήμα στο οποίο αντιστοιχούν, διάφορες ιδιότητες της επιφανείας του (χρώμα, ομαλότητα, λαμπρότητα κλπ) και τη θέση του στον τρισδιάστατο χώρο. Άλλα πάλι αντικείμενα μπορεί είτε να περιέχουν διάφορους ήχους που ακούγονται κατά τη διάρκεια της αλληλεπίδρασης με το σύστημα είτε να περιγράφουν τον τρόπο με τον οποίο φωτίζεται η σκηνή είτε ακόμη να καθορίζουν τις θέσεις στον χώρο (viewpoints) από τις οποίες είναι δυνατή η παρατήρηση του κόσμου από το χρήστη.

Το θεμελιώδες δομικό στοιχείο της VRML είναι ο κόμβος (node). Υπάρχουν κόμβοι που αντιστοιχούν σε αντικείμενα (π.χ. Cylinder, Box, SpotLight κλπ) και υπάρχουν και άλλοι που χρησιμοποιούνται για να ομαδοποιούν αντικείμενα (π.χ. Group). Οι κόμβοι αποτελούνται με τη σειρά τους από πεδία (fields) που περιγράφουν τις ιδιότητες τους. Π.χ. ο κόμβος Shape ο οποίος περιγράφει τρισδιάστατα αντικείμενα περιέχει ένα γεωμετρικό πεδίο (geometry) και ένα πεδίο εμφάνισης (appearance). Κάθε κόμβος μπορεί να περιέχει άλλους κόμβους κάτω από ορισμένες προϋποθέσεις για τις οποίες θα μιλήσουμε παρακάτω.

Πέρα από τους κόμβους που έχουν οπτικό ανάλογο σε έναν εικονικό κόσμο υπάρχουν και μια σειρά από άλλους κόμβους που επιτρέπουν τη σύνδεση πολλών αρχείων μεταξύ τους, την ανίχνευση συγκρούσεων μεταξύ των αντικειμένων ή την ενσωμάτωση συνδέσεων σε διάφορες διευθύνσεις στον Ιστό (WWW). Στη συνέχεια θα παρουσιάσουμε τη δομή ενός απλού προγράμματος σε VRML. Κάθε αρχείο που περιέχει ένα πρόγραμμα VRML θα πρέπει να περιέχει στην αρχή του την παρακάτω δήλωση:

```
#VRML V2.0 utf8
```

που καθορίζει τον τύπο του αρχείου που θα ακολουθήσει και τη συμβολοσειρά που θα πρέπει να χρησιμοποιήσει ο περιηγητής για να διαβάσει το αρχείο. Η δήλωση utf8 ορίζει ότι το αρχείο έχει

κωδικοποιηθεί με βάση το πρότυπο UTF-8.

Στη συνέχεια το πρόγραμμα περιέχει μια σειρά από κόμβους που περιγράφουν τις ιδιότητες των αντικειμένων που συνθέτουν τον εικονικό κόσμο. Για παράδειγμα ένα πρόγραμμα που περιγράφει ένα ορθογώνιο παραλληλεπίπεδο μπορεί να έχει την παρακάτω μορφή:

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
    }
  }
  geometry Box {
  }
}
```

Κάθε κόμβος περιέχει μια σειρά από πεδία τα οποία καθορίζουν τα γεωμετρικά του χαρακτηριστικά, το μοντέλο της επιφάνειας του κ.ο.κ. Στο συγκεκριμένο παράδειγμα ο κόμβος Shape έχει δυο τέτοια πεδία: το γεωμετρικό (geometry) και το επιφανειακό (appearance). Η τιμή του γεωμετρικού πεδίου δίνεται από τον κόμβο Box που αντιστοιχεί στη γεωμετρική αναπαράσταση ενός ορθογωνίου παραλληλεπιπέδου. Κάθε φορά που δημιουργείται ένας κόμβος τα διάφορα πεδία του λαμβάνουν τιμές που έχουν προκαθοριστεί από τον περιηγητή. Ο χρήστης μπορεί να αλλάξει αυτές τις προκαθορισμένες (default) τιμές συμπληρώνοντας το όνομα του πεδίου και την τιμή που θέλει να του αναθέσει στο συγκεκριμένο πρόγραμμα. Για παράδειγμα, ο κόμβος Box έχει ένα πεδίο που ονομάζεται size και το οποίο καθορίζει σε μέτρα το μέγεθος της πλευράς του στους άξονες συντεταγμένων x, y και z. Η προκαθορισμένη τιμή για το πεδίο αυτό είναι 2 m σε κάθε άξονα και το μέγεθος αυτό καθορίζεται από το πεδίο size το οποίο λαμβάνει σαν τιμή ένα διάλυσμα τριών αριθμών που καθορίζουν και το μέγεθος του παραλληλεπιπέδου. Για παράδειγμα ένα πρόγραμμα σχεδίασης ενός παραλληλεπιπέδου με διαστάσεις 3, 2 και 1 μέτρα στους άξονες x, y και z αντίστοιχα έχει την ακόλουθη μορφή:

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
    }
  }
  geometry Box {
    size 3 2 1
  }
}
```

Η εμφάνιση ενός κόμβου περιγράφεται από μία σειρά από πεδία που καθορίζουν τα χρωματικά χαρακτηριστικά του κάθε κόμβου. Στα δύο παραπάνω προγράμματα τα χαρακτηριστικά αυτά καθορίστηκαν από το κόμβο Appearance και από τα πεδία material του κόμβου που έλαβε σα τιμή ένα κόμβο τύπου Material με τις προκαθορισμένες από τον περιηγητή τιμές. Ορισμένα από τα πεδία ενός κόμβου Material είναι τα diffuseColor και shininess που καθορίζουν το χρώμα του φωτός που ανακλάται από το αντικείμενο και τη φωτεινότητα του χρώματος αυτού αντίστοιχα. Το χρώμα του ανακλώμενου φωτός περιγράφεται σαν ένα διάνυσμα τριών τιμών από 0 έως 1 που αντιστοιχεί στις τιμές με τις οποίες συνδυάζονται τα κύρια χρώματα (κόκκινο, πράσινο και μπλε) στη χρωματική κλίμακα RGB για την παραγωγή του τελικού χρώματος, ενώ η φωτεινότητα μετράται με μια τιμή από 0 έως 1. Για παράδειγμα ένα ορθογώνιο παραλληλεπίπεδο χρώματος κόκκινου και μεσαίας φωτεινότητας περιγράφεται από το ακόλουθο πρόγραμμα:

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 1 0 0
      shininess .5
    }
  }
  geometry Box {
    size 3 2 1
  }
}
```

Εκτός από τους κόμβους που περιγράφουν την τρισδιάστατη αναπαράσταση ενός αντικειμένου ένα VRML πρόγραμμα μπορεί να αποτελείται και από κόμβους που ομαδοποιούν τις περιγραφές των αντικειμένων της σκηνής. Τέτοιοι κόμβοι είναι οι: Group, Transform, LOD, Switch, Anchor, Inline και Collision. Οι κόμβοι που ομαδοποιούνται περιγράφονται στο πεδίο children των κόμβων αυτών.

Ο πιο γενικός κόμβος ομαδοποίησης είναι ο Group. Ο κόμβος Transform είναι και αυτός αρκετά χρήσιμος μιας και επιτρέπει τη δημιουργία ενός συνόλου αντικειμένων στο οποίο στη συνέχεια μπορούν να εφαρμοστούν μια σειρά από γεωμετρικοί μετασχηματισμοί όπως μετατόπιση (translation) ή περιστροφή (rotation). Για παράδειγμα μπορούμε να μετατοπίσουμε το παραλληλεπίπεδο του προηγούμενου προγράμματος ένα μέτρο προς τα δεξιά μας με τον ακόλουθο κώδικα:

```
#VRML V2.0 utf8
Transform {
  translation 1 0 0
```

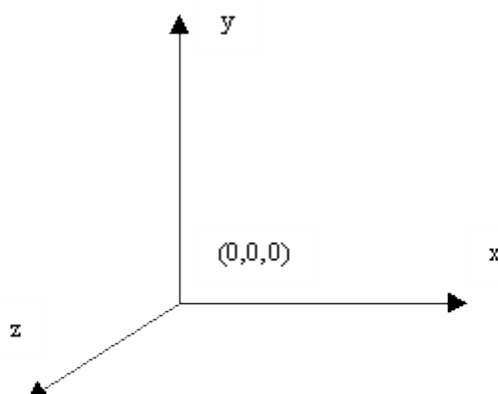
```
children [  
  Shape {  
    appearance Appearance {  
      material Material {  
      }  
    }  
    geometry Box {  
      size 3 2 1  
    }  
  }  
]  
}
```

Το πεδίο translation στο προηγούμενο πρόγραμμα λαμβάνει σαν τιμές ένα διάνυσμα από τρεις αριθμούς που περιγράφουν τη μετατόπιση του αντικειμένου σε μέτρα κατά τους άξονες x, y και z του συστήματος συντεταγμένων που περιγράφουν τη σκηνή μας.

## **Άξονες συντεταγμένων και γεωμετρικοί μετασχηματισμοί.**

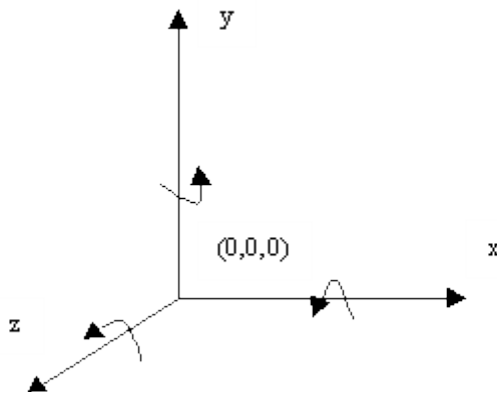
Η σχεδίαση τρισδιάστατων αντικειμένων χρησιμοποιεί μια σειρά από τεχνικές για τη γεωμετρική περιγραφή των αντικειμένων από τα οποία συντίθεται ο χώρος όπως και για τη μετακίνηση, περιστροφή και μεταβολή των διαστάσεων των αντικειμένων αυτών. Η VRML όπως και όλες σχεδόν οι γλώσσες περιγραφής των γεωμετρικών χαρακτηριστικών των αντικειμένων στο χώρο χρησιμοποιεί ένα καθολικό καρτεσιανό σύστημα συντεταγμένων. Το παρακάτω σχήμα απεικονίζει τη μορφή του συστήματος αυτού. Ο άξονας z στο σχήμα κατευθύνεται προς το χρήστη.

Οι γεωμετρικοί μετασχηματισμοί που είναι δυνατοί στη γλώσσα αυτή ορίζονται ως προς το σύστημα συντεταγμένων που περιγράψαμε ή ως προς τοπικά συστήματα συντεταγμένων και αποτελούνται από την μετατόπιση (translation), την περιστροφή (rotation) και την συρρίκνωση/μεγέθυνση (scaling) των διαστάσεων του αντικειμένου.



*Σχήμα 4.1- Το καρτεσιανό σύστημα συντεταγμένων που χρησιμοποιείται στη VRML*

Η μετατόπιση είναι ο πιο απλός από τους γεωμετρικούς μετασχηματισμούς που υπάρχουν και μπορεί να πραγματοποιηθεί κατά μήκος είτε ενός είτε περισσότερων αξόνων ταυτόχρονα στο σύστημα συντεταγμένων. Η σειρά με την οποία εφαρμόζονται μια σειρά από διαδοχικές μετατοπίσεις σε ένα αντικείμενο δεν επηρεάζει την τελική θέση του αντικειμένου που θα είναι η ίδια σε όλες τις περιπτώσεις. Για παράδειγμα μπορούμε να μετατοπίσουμε ένα αντικείμενο πρώτα +5 μέτρα στον άξονα x και στη συνέχεια +3 μέτρα στον άξονα y είτε πρώτα +3 μέτρα στον άξονα y και στη συνέχεια +5 μέτρα στον άξονα x. Η τελική θέση του αντικειμένου θα είναι η ίδια και στις δύο αυτές περιπτώσεις. Η μετατόπιση ενός αντικειμένου περιγράφεται από την τιμή του πεδίου translation που χρησιμοποιείται σε κόμβους ομαδοποίησης όπως ο Transform. Η τιμή αυτή αποτελείται από ένα διάνυσμα τριών τιμών που περιγράφει τη μετατόπιση του αντικειμένου κατά τους x, y και z άξονες. Η περιστροφή χρησιμοποιείται για την περιστροφή ενός ή περισσότερων αντικειμένων γύρω από ένα κέντρο και άξονα περιστροφής. Η γωνία περιστροφής μετριέται σε ακτίνια (radians). Σε ένα δεξιόστροφο σύστημα συντεταγμένων όπως αυτό που χρησιμοποιείται στην VRML η θετική φορά περιστροφής είναι αυτή στην οποία κλείνουν τα δάκτυλα του δεξιού χεριού μας όταν ο αντίχειρας λαμβάνει την κατεύθυνση του άξονα περιστροφής. Η περιστροφή ενός αντικειμένου περιγράφεται από την τιμή του πεδίου rotation που χρησιμοποιείται σε κόμβους ομαδοποίησης όπως ο Transform. Η τιμή αυτή αποτελείται από ένα διάνυσμα τεσσάρων τιμών. Οι πρώτες τρεις από αυτές αντιστοιχούν στις συντεταγμένες ενός σημείου στο χώρο. Ο άξονας περιστροφής ταυτίζεται με την ευθεία που συνδέει την αρχή των αξόνων με το σημείο αυτό με κατεύθυνση προς το σημείο. Η τέταρτη τιμή αντιστοιχεί στο μέγεθος σε ακτίνια της γωνίας περιστροφής. Το κέντρο περιστροφής στη VRML βρίσκεται πάντα στο κέντρο του συστήματος συντεταγμένων (στο σημείο (0,0,0)).



*Σχήμα 4.2- Θετική φορά περιστροφής σε καθέναν από τους ορθογώνιους άξονες*

Η μεγέθυνση/συρρίκνωση ενός αντικειμένου στην VRML μας επιτρέπει να μεταβάλλουμε το μέγεθος ενός ή περισσότερων αντικειμένων κατά ένα ποσοστό των αρχικών διαστάσεων του σε καθέναν από τους τρεις ορθογώνιους άξονες. Η μεγέθυνση/συρρίκνωση ενός αντικειμένου περιγράφεται από την τιμή του πεδίου `scaling` που χρησιμοποιείται σε κόμβους ομαδοποίησης όπως ο `Transform`. Η τιμή αυτή αποτελείται από ένα διάνυσμα τριών τιμών που περιγράφει το ποσοστό της μεταβολής των διαστάσεων του αντικειμένου κατά τους x, y και z άξονες.

Η σειρά με την οποία εφαρμόζονται οι γεωμετρικοί μετασχηματισμοί που περιέχονται στον ίδιο κόμβο στην VRML είναι: μεγέθυνση/συρρίκνωση - περιστροφή - μετατόπιση. Για παράδειγμα η εκτέλεση του παρακάτω προγράμματος:

```
#VRML V2.0 utf8
Transform {
  translation 1 0 0
  children [
    Transform {
      translation .5 .5 .1
      rotation 0 1 0 1.57
      scale .5 1 1.2
      children [
        Shape {
          appearance Appearance {
            material Material {
            }
          }
          geometry Box {
            size 3 2 1
          }
        }
      ]
    }
  ]
}
```

```
]
}
]
}
```

θα εφαρμόσει τους παρακάτω μετασχηματισμούς σε ένα ορθογώνιο παραλληλεπίπεδο:

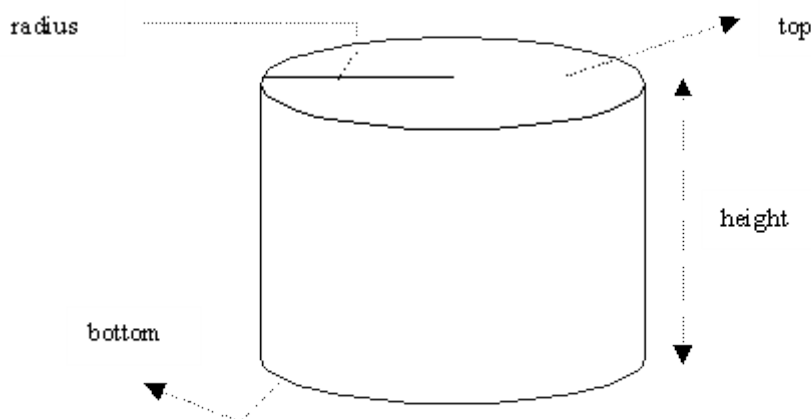
1. Μεγέθυνση/Συρρίκνωση 50%, 100%, 120% σε καθένα από τους ορθογώνιους άξονες.
2. Περιστροφή κατά +90 μοίρες γύρω από τον άξονα y.
3. Μετατόπιση κατά 0.5, 0.5 και .1 μέτρα ως προς τον καθένα από τους άξονες x, y και z.
4. Μετατόπιση κατά 1 μέτρο ως προς τον άξονα x.

### Στοιχειώδη γεωμετρικά σχήματα

Η VRML όπως και η πλειοψηφία των περιβαλλόντων ανάπτυξης χώρων εικονικής πραγματικότητας προσφέρει ένα σύνολο από στοιχειώδη γεωμετρικά αντικείμενα τα οποία μπορούν να χρησιμοποιηθούν για τη δημιουργία πιο πολύπλοκων αντικειμένων. Στη συνέχεια θα παρουσιάσουμε καθένα από τους κόμβους που αντιστοιχούν στα αντικείμενα αυτά.

Ο κόμβος *Box* όπως αναφέραμε και στην προηγούμενη παράγραφο μας δίνει τη δυνατότητα να ορίσουμε ορθογώνια παραλληλεπίπεδα στον εικονικό χώρο. Ο κόμβος αυτός έχει ένα μόνο πεδίο (το *size*) το οποίο και καθορίζει τις διαστάσεις του αντικειμένου σε καθένα από τους άξονες x, y και z του καθολικού συστήματος συντεταγμένων.

Ο κόμβος *Cylinder* χρησιμοποιείται για την εισαγωγή κυλινδρικών αντικειμένων στο χώρο.



Σχήμα 4.3- Γεωμετρική περιγραφή κυλίνδρου

Τα πεδία του κόμβου αυτού διαφέρουν από αυτά του *Box* μιας και αντιστοιχούν στα ακόλουθα χαρακτηριστικά:



ΠΕΔΙΟ	ΕΠΕΞΗΓΗΣΗ
radius	Η ακτίνα του κυλίνδρου
height	Το ύψος του κυλίνδρου
top	Λογικό πεδίο το οποίο καθορίζει την ύπαρξη ή όχι ενός κυλινδρικού δίσκου στην κορυφή του κυλίνδρου. Να λογικό πεδίο μπορεί να πάρει μία από δύο πιθανές τιμές: TRUE ή FALSE.
bottom	Λογική μεταβλητή η οποία καθορίζει την ύπαρξη ή όχι ενός κυλινδρικού δίσκου στη βάση του κυλίνδρου.

Για παράδειγμα ο παρακάτω κώδικας:

```
#VRML V2.0 utf8
Transform {
  children [
    DEF MyFirstCylinder Shape {
      appearance Appearance {
        material Material {
        }
      }
      geometry Cylinder {
        radius 1.4
        height 2.3
        top TRUE
        bottom FALSE
      }
    }
  ]
}
```

περιγράφει έναν κύλινδρο ακτίνας 1.4 m, ύψους 2.3 m κλειστό ως προς την κορυφή του και χωρίς πάτο. Η εντολή DEF που περιέχεται στον κώδικα μας επιτρέπει να δώσουμε ένα όνομα σε ένα κόμβο του προγράμματος μας. Στην περίπτωση μας η εντολή αυτή μας επιτρέπει να ονομάσουμε τον κύλινδρο MyFirstCylinder και στο υπόλοιπο πρόγραμμα μπορούμε να αναφερόμαστε στον συγκεκριμένο κύλινδρο με το όνομα αυτό.

Η VRML περιέχει και δύο άλλους κόμβους που περιγράφουν σφαιρικά (*Sphere*) και κωνικά (*Cone*) αντικείμενα. Ο κόμβος Cone αντιστοιχεί σε έναν κώνο με κεντρικό άξονα που ταυτίζεται με τον τοπικό άξονα y. Ο κόμβος περιέχει δύο αριθμητικά πεδία που περιγράφουν την ακτίνα του κάτω μέρους του (*bottomRadius*) και το ύψος του (*height*) όπως και δύο λογικά πεδία που καθορίζουν αν το πλευρικό τμήμα του κώνου (*side*) ή ο πάτος του (*bottom*) θα είναι ορατά. Ο κόμβος Sphere περιέχει ένα μόνο πεδίο που καθορίζει το μήκος της ακτίνας της σφαίρας (*radius*).

Για την απεικόνιση αντικειμένων με γεωμετρικά χαρακτηριστικά που δε μπορούν να αναλυθούν με

ευκολία στα βασικά σχήματα που αναφέραμε, η VRML παρέχει δύο κόμβους που επιτρέπουν τον ορισμό των γεωμετρικών ιδιοτήτων ενός αντικειμένου με τη χρήση είτε ενός συνόλου πολυγώνων (*IndexedFaceSet*) είτε ενός συνόλου ευθυγράμμων τμημάτων (*IndexedLineSet*).

Ο κόμβος *IndexedFaceSet* επιτρέπει τον ορισμό των γεωμετρικών ιδιοτήτων ενός αντικειμένου με τον καθορισμό μιας σειράς από πολύγωνα από τα οποία αποτελείται. Το καθένα από τα πολύγωνα αυτά περιγράφεται από μια λίστα που περιέχει τις συντεταγμένες των κορυφών του. Η λίστα αυτή περιέχεται σε ένα ειδικό κόμβο που λέγεται *Coordinate*. Το πεδίο *coord* ενός κόμβου *IndexedFaceSet* παίρνει σαν τιμή ένα κόμβο *Coordinate* που περιέχει τις συντεταγμένες όλων των κορυφών των πολυγώνων που απαρτίζουν το σχήμα.

Πιο συγκεκριμένα ο κόμβος *Coordinate* περιέχει ένα πεδίο, το *point*, το οποίο αποτελείται από μία λίστα από συντεταγμένες στον τρισδιάστατο χώρο που περιγράφουν τη θέση αντίστοιχων σημείων στο χώρο. Για παράδειγμα ο ακόλουθος κώδικας:

```
Coordinate {
  Point [
    1.2 2 3, 3 4.1 5, -2 3 5
  ]
}
```

περιγράφει τη θέση τριών σημείων στο χώρο.

Εκτός από το πεδίο *coord*, ένας κόμβος *IndexedFaceSet* περιέχει και τις κορυφές από τις οποίες αποτελείται κάθε πολύγωνο του αντικειμένου. Οι κορυφές κάθε πολυγώνου περιέχονται στο πεδίο *coordIndex* και καθορίζονται από μια λίστα δεικτών που αντιστοιχούν στη σειρά εμφάνισης της κάθε κορυφής του στην λίστα από σημεία που περιέχονται στην τιμή του πεδίου *coord*.

Για παράδειγμα ο παρακάτω κώδικας:

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
    }
  }
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        -1 -1 0,
        1 -1 0,
        1 1 0,
        -1 1 0
      ]
    }
  }
}
```

```

    ]
  }
  coordIndex [
    0, 1, 2, 3, -1
  ]
}
}

```

περιγράφει ένα ορθογώνιο παραλληλόγραμμο στο επίπεδο xy (οι συντεταγμένες όλων των κορυφών του σχήματος ως προς z είναι 0). Ο δείκτης 1 στο πεδίο coordIndex του παραπάνω κώδικα χρησιμοποιείται για να διαχωρίζει τις κορυφές που ανήκουν σε κάθε πολύγωνο.

Ο κόμβος IndexedLineSet ορίζεται όπως ακριβώς και ο IndexedFaceSet με τη διαφορά ότι αντί να σχεδιάζει τα πολύγωνα από τα οποία αποτελείται ένα αντικείμενο σχεδιάζει μόνο τα ευθύγραμμα τμήματα που συνδέουν τις κορυφές των πολυγώνων από τα οποία αποτελείται. Για να αντιληφθείτε τη διαφορά μπορείτε να αντικαταστήσετε στον παραπάνω κώδικα τον κόμβο IndexedFaceSet με τον κόμβο IndexedLineSet διατηρώντας τα υπόλοιπα πεδία αμετάβλητα.

Τέλος η VRML περιέχει τον κόμβο PointSet με τον οποίο μπορεί να απεικονιστεί ένα σύνολο από σημεία τα οποία περιέχονται στο πεδίο coord του κόμβου.

## Κόμβοι κειμένου

Εκτός από τους κόμβους που αντιστοιχούν σε στοιχειώδη γεωμετρικά σχήματα και πολυγωνικές επιφάνειες η VRML μας δίνει τη δυνατότητα να εισάγουμε δισδιάστατο κείμενο σε μια τρισδιάστατη σκηνή. Ο ορισμός του κειμένου γίνεται με τη χρησιμοποίηση του κόμβου *Text*. Ο κόμβος ορίζει μια δισδιάστατη συμβολοσειρά που είναι ορατή και από τις δύο όψεις της επιφάνειας και η οποία βρίσκεται στο επίπεδο  $z = 0$ . Ο *Text* περιέχει τέσσερα πεδία: *string*, *fontstyle*, *length* και *maxExtent*.

Το πεδίο *string* καθορίζει τη συμβολοσειρά την οποία θέλουμε να προβάλλουμε στη σκηνή μας. Το πεδίο *fontstyle* λαμβάνει σαν τιμή ένα κόμβο *FontStyle* τα πεδία του οποίου καθορίζουν τον τύπο (*family*, π.χ. SERIF), το μέγεθος (*size*), τον τρόπο εμφάνισης (*style* π.χ. PLAIN, BOLD κλπ), τον τρόπο στοίχισης (*justify*, π.χ. BEGIN), την κατεύθυνση της γραφής (*horizontal*, *leftToRight*, *topToBottom* όλα λογικά πεδία) τα διάστιχα (*spacing*) κλπ.

Το πεδίο *length* καθορίζει το μήκος του κειμένου μας στο χώρο. Σε περίπτωση που το κείμενο μας έχει μήκος μικρότερο από αυτό που καθορίζεται από τον κόμβο η απόσταση μεταξύ των γραμμμάτων θα μεγαλώσει έτσι ώστε το μήκος της συμβολοσειράς να γίνει ίσο με την τιμή του πεδίου. Σε αντίθετη περίπτωση η απόσταση μεταξύ των γραμμμάτων θα ελαττωθεί. Το πεδίο

maxExtent καθορίζει το μέγιστο μήκος του κειμένου μας. Σε περίπτωση που το μήκος του κειμένου μας είναι μεγαλύτερο από το maxExtent η απόσταση μεταξύ των γραμμάτων θα μικρύνει έτσι ώστε το μήκος του κειμένου να γίνει ίσο με maxExtent. Μια τιμή ίση με το 0 στα πεδία length και maxExtent δεν επιβάλλει κανένα περιορισμό στο μήκος του κειμένου. Για παράδειγμα το ακόλουθο πρόγραμμα:

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 0 0 0
      specularColor 0 0 0
      ambientIntensity 0.0
      shininess 0.0
      emissiveColor .5 .5 .9
    }
  }
  geometry Text {
    string "University of Piraeus"
  }
}
```

θα εμφανίσει το μήνυμα "University of Piraeus" στον περιηγητή σας σε μια οριζόντια γραμμή από τα δεξιά προς τα αριστερά. Το πρόγραμμα:

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 0 0 0
      specularColor 0 0 0
      ambientIntensity 0.0
      shininess 0.0
      emissiveColor .5 .5 .9
    }
  }
  geometry Text {
    string "University of Piraeus"
    fontStyle FontStyle {
      horizontal FALSE
    }
  }
}
```

θα εμφανίσει το ίδιο μήνυμα από πάνω προς τα κάτω.

Η χρήση του κόμβου Text θα πρέπει να γίνεται με σύνεση καθώς η σχεδίαση του κόμβου αυτού χρησιμοποιεί ένα μεγάλο αριθμό πολυγώνων που επιβαρύνουν σημαντικά τον υπολογισμό των οπτικών χαρακτηριστικών μίας σκηνής (rendering).

## Σημεία κατόπτρευσης

Εκτός από τη θέση των διαφόρων αντικειμένων που απαρτίζουν μια τρισδιάστατη σκηνή, η VRML μας παρέχει τη δυνατότητα να ορίσουμε τα σημεία παρατήρησης του χώρου από το χρήστη. Ο προσδιορισμός των σημείων αυτών γίνεται με τη χρήση του κόμβου *Viewpoint*. Πιο συγκεκριμένα ο κόμβος καθορίζει τη θέση και τον προσανατολισμό μιας κάμερας με την οποία ο χρήστης παρατηρεί τον κόσμο. Τα πεδία του κόμβου που μας ενδιαφέρουν για την ώρα είναι τα: *position*, *orientation* και *fieldOfView*.

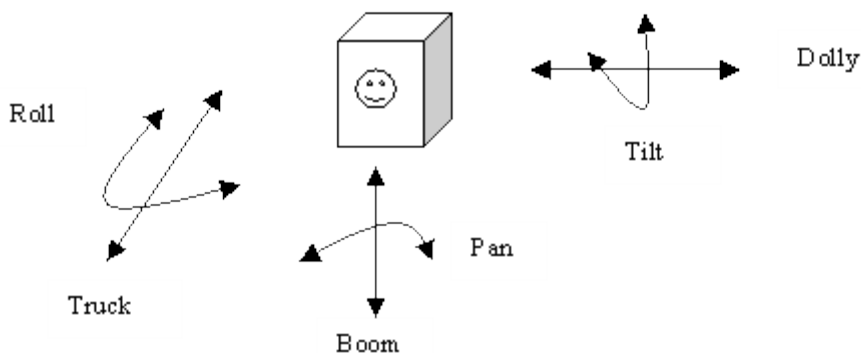
Τα πεδία *position* και *orientation* περιέχουν μια λίστα από 3 και 4 αριθμούς αντίστοιχα που περιγράφουν τις συντεταγμένες στους άξονες x, y και z και τον άξονα και τη γωνία περιστροφής της κάμερας αντίστοιχα. Το πεδίο *fieldOfView* περιέχει το οπτικό πεδίο που προσφέρει ο φακός και παίρνει τιμές σε ακτίνια από 0 έως  $\pi$ . Τιμές κοντά στο 0 αντιστοιχούν στη χρήση φακών τηλεκατόπτρευσης ενώ τιμές κοντά στο  $\pi$  αντιστοιχούν στη χρήση ευρυγώνιων φακών. Για παράδειγμα το ακόλουθο πρόγραμμα:

```
#VRML V2.0 utf8
Viewpoint {
  position 2 0 5
  orientation 1 0 0 .2
  fieldOfView 1.2
  description "InitialCameraPosition"
}
Transform {
  children [
    DEF MyShape Shape {
      appearance Appearance {
        material Material {
          diffuseColor .5 .5 .1
        }
      }
      geometry Box {
        size 1 2 1
      }
    }
  ]
}
```

μας επιτρέπει να παρατηρήσουμε το σχήμα MyShape από αρχική θέση (2,0,5) και με κλίση +0.2 ακτίνια ως προς τον άξονα x.

Όπως θα εξηγήσουμε αργότερα ο κόμβος Viewpoint μας επιτρέπει να υλοποιήσουμε στη VRML εφαρμογές συνθετικής κίνησης του χρήστη καθώς η εκτέλεση μιας σειράς από γεωμετρικούς μετασχηματισμούς μετατόπισης ή περιστροφής πάνω στον κόμβο αυτό μας επιτρέπουν να ελέγχουμε τι βλέπει ο χρήστης κάθε χρονική στιγμή στη σκηνή μας.

Η κίνηση του χρήστη σε περιβάλλοντα τρισδιάστατων γραφικών περιγράφεται με μια σειρά από όρους που είναι δανεισμένοι από την κινηματογραφική διάλεκτο. Το παρακάτω σχήμα περιγράφει τους όρους που χρησιμοποιούνται για να περιγράψουν την κίνηση της κάμερας στα περιβάλλοντα αυτά.



Σχήμα 4.4 - Κίνηση σε τρισδιάστατο χώρο

## Μορφοποίηση

Ένα περιβάλλον τρισδιάστατων γραφικών πέρα από την θέση και τη γεωμετρία των αντικειμένων από τα οποία αποτελείται ο χώρος θα πρέπει να είναι σε θέση να υπολογίσει και τα οπτικά χαρακτηριστικά τους όπως το χρώμα ή τη λαμπρότητα τους. Η διαδικασία αυτή είναι γνωστή σαν *μορφοποίηση* ή *rendering* και συνήθως αποτελείται από την ακόλουθη σειρά βημάτων:

1. Υπολόγισε τα γεωμετρικά χαρακτηριστικά των αντικειμένων του χώρου
2. Όρισε το σημείο κατόπτρευσης του χώρου
3. Όρισε τις πηγές φωτός που θα χρησιμοποιηθούν
4. Όρισε τα επιφανειακά χαρακτηριστικά όλων των αντικειμένων του χώρου
5. Διάλεξε την τεχνική σκίασης που θα χρησιμοποιήσεις
6. Υπολόγισε τα οπτικά χαρακτηριστικά του χώρου σου

Στη συνέχεια θα παρουσιάσουμε τον τρόπο με τον οποίο υλοποιείται καθένα από τα βήματα αυτά σε ένα περιβάλλον VRML. Επειδή έχουμε ήδη παρουσιάσει στο προηγούμενο κεφάλαιο

ορισμένους τρόπους με τους οποίους ορίζουμε τα γεωμετρικά χαρακτηριστικά και τη θέση της κάμερας στο χώρο στο κεφάλαιο αυτό θα αναλύσουμε τα υπόλοιπα 4 βήματα.

## Πηγές φωτός

Η VRML χρησιμοποιεί τρεις τύπους κόμβων για να καθορίσει το φωτισμό μίας σκηνής. Οι κόμβοι αυτοί είναι οι: *DirectionalLight*, *PointLight* και *SpotLight*. Στους κόμβους *PointLight* και *SpotLight* η ακτίνα του χώρου που φωτίζεται από την κάθε πηγή καθορίζεται από την τιμή του πεδίου *radius*. Στον κόμβο *DirectionalLight* τα αντικείμενα που φωτίζονται από αυτόν είναι μόνο αυτά που αποτελούν παιδιά του ιδίου κόμβου ομαδοποίησης. Υπάρχουν 4 πεδία τα οποία είναι κοινά για όλες τις πηγές φωτός. Τα ονόματα των πεδίων αυτών είναι: *on*, *color*, *intensity* και *ambientIntensity*.

Το λογικό πεδίο *on* καθορίζει αν η πηγή φωτός είναι αναμμένη ή όχι. Το πεδίο *color* καθορίζει το χρώμα του εκπεμπόμενου φωτός και λαμβάνει σαν τιμή μια λίστα από τρεις αριθμούς που καθορίζουν το ποσοστό ανάμειξης των βασικών χρωμάτων στο σύστημα RGB. Το πεδίο *intensity* καθορίζει τη λαμπρότητα της φωτεινής πηγής και παίρνει τιμές από το 0 (μηδενική λαμπρότητα) μέχρι το 1 (μέγιστη λαμπρότητα). Τέλος το πεδίο *ambientIntensity* καθορίζει τη συνεισφορά της κάθε πηγής φωτός στον *περιβάλλοντα φωτισμό (ambient light)* που υπάρχει σε μία σκηνή. Σαν *περιβάλλοντα φωτισμό* ορίζουμε το φως που δε προέρχεται απευθείας από μια πηγή φωτός σε μία σκηνή παρά προέρχεται από τη διάχυση και την ανάκλαση του φωτός από όλες τις φωτεινές πηγές της σκηνής μας. Αν μία φωτεινή πηγή είναι αναμμένη τότε η συνεισφορά της στο συνολικό *περιβάλλοντα φωτισμό* υπολογίζεται για κάθε μία από τις συνιστώσες του RGB σαν το γινόμενο:

$$\text{intensity} * \text{ambientIntensity} * \text{τιμή της RGB συνιστώσας}$$

Για παράδειγμα η συνεισφορά της πηγής που περιγράφεται από τον παρακάτω κώδικα:

```
DEF MyLightSource PointLight {  
  on TRUE  
  intensity .6  
  ambientIntensity .8  
  color .2 .3 .4  
}
```

θα είναι  $.2 * .8 * .6 = .096$  για την κόκκινη συνιστώσα του RGB,  $.144$  για την πράσινη συνιστώσα και  $.192$  για τη μπλε. Οι τιμές των συνιστωσών για όλες τις φωτεινές πηγές συνδυάζονται κατάλληλα για τον καθορισμό του χρώματος του περιβάλλοντος φωτός σε κάθε σημείο της σκηνής.

Ο κόμβος *DirectionalLight* αντιστοιχεί σε μία φωτεινή πηγή που βρίσκεται στο άπειρο και για την

οποία υποθέτουμε ότι όλες οι εκπεμπόμενες ακτίνες φωτός είναι παράλληλες μεταξύ τους. Η κατεύθυνση στην οποία θεωρούμε ότι εκπέμπονται οι ακτίνες αυτές καθορίζονται από το πεδίο *direction* που ορίζει ένα διάνυσμα κατεύθυνσης στον τρισδιάστατο χώρο.

Ο κόμβος *PointLight* αντιστοιχεί σε μια πηγή φωτός που ακτινοβολεί προς όλες τις κατευθύνσεις όπως μία λάμπα. Ο χώρος στον οποίο ακτινοβολεί μια τέτοια πηγή φωτός είναι μια σφαίρα με κέντρο την πηγή και ακτίνα που καθορίζεται από τη τιμή του πεδίου *radius*. Η θέση ενός τέτοιου κόμβου καθορίζεται από την τιμή του πεδίου *location* ενώ η εξασθένηση της έντασης του φωτός που εκπέμπεται στο χώρο αυτό καθορίζεται από την τιμή του πεδίου *attenuation*. Ο υπολογισμός της εξασθένησης της έντασης σε μια ορισμένη απόσταση από τη θέση της πηγής γίνεται πολλαπλασιάζοντας την ένταση της πηγής με τον παράγοντα:

$$1/(\text{attenuation}[0] + \text{attenuation}[1]*r + \text{attenuation}[2]*r^2)$$

όπου *r* είναι η απόσταση του σημείου που μας ενδιαφέρει από την πηγή.

Ο κόμβος *SpotLight* χρησιμοποιείται για φωτεινές πηγές που βρίσκονται σε μια ορισμένη θέση και εκπέμπουν φως στο εσωτερικό ενός κώνου με καθορισμένο προσανατολισμό. Η ένταση του φωτός εξασθενεί εκθετικά όσο προχωρούμε προς την επιφάνεια του κώνου από την επιφάνεια ενός εσωτερικού κώνου με ακτίνα *beamWidth* ενώ λαμβάνει τη μέγιστη τιμή της σε κάθε σημείο στο εσωτερικό του μικρότερου αυτού κώνου. Η μέγιστη γωνία στην οποία εκπέμπεται φως από την πηγή καθορίζεται από την τιμή του πεδίου *cutOffAngle*.

Το παρακάτω πρόγραμμα μπορεί να αποτελέσει μια αρχή για την μελέτη των επιπτώσεων της χρήσης φωτεινών πηγών σε ένα τρισδιάστατο περιβάλλον. Το πρόγραμμα απεικονίζει ένα κύβο που φωτίζεται από δύο διαφορετικές κατευθύνσεις από ένα κόμβο *PointLight* και ένα κόμβο *SpotLight*.

```
#VRML V2.0 utf8
PointLight {
  on TRUE
  intensity 0.75
  ambientIntensity 0.5
  color 0 0 1
}
SpotLight {
  on TRUE
  intensity 1
  ambientIntensity 0.5
  color 1 0 0
  direction 0 0 -1
  location 2 2 0
```



```

}
Group {
  children [
    Transform {
      translation 2 2 -2
      children [
        Shape {
          appearance Appearance {
            material Material {
            }
            texture NULL
            textureTransform NULL
          }
          geometry Box {
          }
        }
      ]
    }
  ]
}

```

Η επίδραση που έχει ένα κόμβος DirectionalLight μπορεί να μελετηθεί χρησιμοποιώντας τον παρακάτω κώδικα στον οποίο δύο ορθογώνια παραλληλεπίπεδα με κοινή γεωμετρία φωτίζονται από δύο πηγές SpotLight και PointLight αλλά μόνο το ένα από αυτά φωτίζεται από ένα κόμβο DirectionalLight. Στην περίπτωση αυτή το χρώμα των δύο αντικειμένων είναι διαφορετικό παρόλο που τα δύο σχήματα έχουν επικαλυπτόμενη γεωμετρία.

```

#VRML V2.0 utf8
PointLight {
  on TRUE
  intensity 0.75
  ambientIntensity 0.5
  color 0 0 1
}
SpotLight {
  on TRUE
  intensity 1
  ambientIntensity 0.5
  color 1 0 0
  direction 0 0 -1
  location 2 2 0
}

```

```

Transform {
  translation 2 1 -2
  children [
    Shape {
      appearance Appearance {
        material Material {
        }
        texture NULL
        textureTransform NULL
      }
      geometry Box {
      }
    }
  ]
}
Group {
  children [
    Transform {
      translation 2 2 -2
      children [
        Shape {
          appearance Appearance {
            material Material {
            }
            texture NULL
            textureTransform NULL
          }
          geometry Box {
          }
        }
        DirectionalLight {
          on TRUE
          color 1 0 0
          intensity 1
          ambientIntensity .2
        }
      ]
    }
  ]
}

```

Η υπολογιστική επιβάρυνση που προέρχεται από τη χρήση φωτεινών πηγών που έχουν μοντελοποιηθεί με βάση έναν από τους τρεις κόμβους που αναφέραμε είναι μεγαλύτερη για τον

κόμβο SpotLight που έχει και τα πιο πολύπλοκα οπτικά χαρακτηριστικά (κωνικό πεδίο δράσης, εξασθένιση κλπ), μικρότερη για τον κόμβο PointLight και ακόμα μικρότερη για τον κόμβο DirectionalLight που έχει και τα πιο απλά οπτικά χαρακτηριστικά (παράλληλες ακτίνες, πεδίο δράσης ένα καθορισμένο σύνολο από αντικείμενα κλπ).

## Επιφανειακά χαρακτηριστικά

Τα επιφανειακά χαρακτηριστικά ενός κόμβου περιγράφονται από έναν κόμβο τύπου *Appearance*. Κόμβοι τέτοιου είδους χρησιμοποιούνται σαν τιμές στο πεδίο *appearance* κόμβων Shape και καθορίζουν τα επιφανειακά χαρακτηριστικά των σχημάτων στα οποία αντιστοιχούν οι κόμβοι Shape. Τα πεδία του κόμβου που μας ενδιαφέρουν αυτή τη στιγμή είναι τα εξής: *material* και *texture*.

Το πεδίο *material* παίρνει σαν τιμές κόμβους *Material* οι οποίοι περιγράφουν οπτικά χαρακτηριστικά του αντικειμένου όπως είναι η λαμπρότητα ή η διαφάνεια του. Πιο συγκεκριμένα ένας κόμβος *Material* περιέχει έξι πεδία: *diffuseColor*, *ambientIntensity*, *specularColor*, *emissiveColor*, *shininess* και *transparency*. Το πεδίο *diffuseColor* καθορίζει στο σύστημα RGB το χρώμα που ανακλάται περισσότερο από το αντικείμενο όταν πέφτει πάνω του φως από οποιαδήποτε πηγή φωτός. Το πεδίο αυτό αντιστοιχεί στην έννοια του 'χρώματος' ενός αντικειμένου όπως αυτή χρησιμοποιείται στην καθημερινή ζωή. Το πεδίο *ambientIntensity* καθορίζει το ποσοστό του φωτός που ανακλά η επιφάνεια του σώματος. Η ένταση του φωτός στην συγκεκριμένη περίπτωση καθορίζεται σαν:

$ambientIntensity * diffuseColor$

Το πεδίο *specularColor* καθορίζει το χρώμα με το οποίο καθρεφτίζεται πάνω στην επιφάνεια μια πηγή φωτός που φωτίζει απευθείας το σώμα. Πιο συγκεκριμένα όταν η γωνία μεταξύ του σημείου κατόπτρευσης και μιας επιφάνειας πλησιάζει τη γωνία μεταξύ μιας φωτεινής πηγής που φωτίζει απευθείας την επιφάνεια και της επιφάνειας αυτής τότε το χρώμα που ορίζεται στο πεδίο *specularColor* συμμετέχει στον καθορισμό του χρώματος του αντικειμένου μαζί με το *diffuseColor* και το χρώμα του περιβάλλοντος φωτός. Το πεδίο *specularColor* χρησιμοποιείται για να προσομοιώσει τη συμπεριφορά αντικειμένων όπως οι καθρέπτες στα οποία το φως ανακλάται μόνο σε συγκεκριμένες διευθύνσεις που εξαρτώνται από τη γωνία με την οποία προσπίπτει το φως πάνω στην επιφάνεια.

Το πεδίο *emissiveColor* καθορίζει το χρώμα που εκπέμπουν αυτόφωτα αντικείμενα. Το πεδίο *shininess* χρησιμοποιείται για να προσομοιώσει τη λαμπρότητα επιφανειών και κυμαίνεται από το 0 για τραχιές επιφάνειες οι οποίες ανακλούν λίγο από το προσπίπτον φως και προς όλες τις

διευθύνσεις ως το 1 για γυαλισμένες επιφάνειες οι οποίες ανακλούν αρκετό από το προσπίπτον φως αλλά σε περιορισμένο εύρος διευθύνσεων. Τέλος το πεδίο *transparency* καθορίζει το βαθμό διαφάνειας μιας επιφάνειας και μπορεί να πάρει τιμές από 0 για πλήρως αδιαφανή σώματα έως 1 για πλήρως διαφανή.

Το πεδίο *texture* ενός κόμβου *Appearance* καθορίζει την υφή της επιφάνειας ενός σχήματος. Υπάρχουν δύο τρόποι για να οριστεί η υφή μιας επιφάνειας. Ο πρώτος τρόπος στηρίζεται στη προσομοίωση των γεωμετρικών χαρακτηριστικών μίας επιφάνειας. Η υλοποίηση μίας τέτοιας προσομοίωσης έχει δύο μειονεκτήματα. Το πρώτο μειονέκτημα προέρχεται από την ανάγκη να αναπαρασταθούν με λεπτομέρεια και σε πολύ μικρή κλίμακα μεγέθους οι μεταβολές των γεωμετρικών χαρακτηριστικών μίας επιφάνειας. Το δεύτερο μειονέκτημα προέρχεται από την πολυπλοκότητα του υπολογισμού των οπτικών χαρακτηριστικών μίας τέτοιας επιφάνειας.

Ο δεύτερος τρόπος στηρίζεται στη δημιουργία δισδιάστατων εικόνων (*textures*) που φωτογραφίζουν την υφή μίας επιφάνειας και την προβολή αυτών των εικόνων πάνω στην επιφάνεια ενός τρισδιάστατου σώματος. Πέρα από την αληθοφάνεια που μπορεί να έχουν οι εικόνες αυτές η χρησιμοποίησή τους δε μεταβάλλει τα γεωμετρικά χαρακτηριστικά των αντικειμένων στα οποία προβάλλονται διευκολύνοντας έτσι τον υπολογισμό των οπτικών τους χαρακτηριστικών. Ο τρόπος με τον οποίο ορίζονται και προβάλλονται τέτοιες εικόνες στη VRML είναι με τη χρήση ενός κόμβου *ImageTexture* σαν τιμή του πεδίου *texture* του κόμβου *Appearance*. Ένα από τα πεδία του κόμβου *ImageTexture*, το *url*, δέχεται σαν τιμή μια συμβολοσειρά που καθορίζει τη διεύθυνση ενός αρχείου στον Ιστό ή σε έναν από τους τοπικούς δίσκους το οποίο και περιέχει τη δισδιάστατη εικόνα. Το αρχείο αυτό μπορεί να είναι κωδικοποιημένο είτε σε JPEG είτε σε PNG. Επιπροσθέτως το πεδίο *texture* μπορεί να λάβει σαν τιμές κόμβους *MovieTexture* και *PixelTexture*. Ο κόμβος *MovieTexture* επιτρέπει την προβολή βίντεο κωδικοποιημένο σε MPEG-1 πάνω σε μία επιφάνεια ενώ ο κόμβος *PixelTexture* μας επιτρέπει να ορίσουμε δικά μας *textures* σε ένα πρόγραμμα VRML.

## Τεχνικές σκίασης

Η σκίαση αποτελεί το σημείο εκείνο της διαδικασίας υπολογισμού των οπτικών χαρακτηριστικών των αντικειμένων μας κατά το οποίο υπολογίζεται ο προσανατολισμός των ορατών επιφανειών κάθε αντικειμένου στη σκηνή. Η εύρεση του προσανατολισμού των επιφανειών αυτών χρησιμεύει για τον υπολογισμό της γωνίας πρόσπτωσης του φωτός από διάφορες φωτεινές πηγές έτσι ώστε να καθοριστεί το χρώμα κάθε τέτοιου σημείου όπως και για τον καθορισμό του τρόπου μεταβολής των χρωματικών χαρακτηριστικών ενός αντικειμένου. Στην τελευταία αυτή περίπτωση η τεχνική σκίασης που χρησιμοποιείται καθορίζει αν θα υπάρχει μια συνεχής μεταβολή των χρωματικών

χαρακτηριστικών μίας επιφάνειας ή αυτά θα μεταβάλλονται ασυνεχώς αποκαλύπτοντας και τα πολύγωνα από τα οποία έχουν συντεθεί τα αντικείμενα της σκηνής μας. Γενικά οι τεχνικές σκίασης μπορεί να κατηγοριοποιηθούν σε ομαλές (*smooth* ή *interpolative*) και σε πλευρικές (*faceted*). Για την εφαρμογή και των δύο αυτών κατηγοριών τεχνικών θεωρούμε ότι η επιφάνεια κάθε αντικειμένου αποτελείται από μια σειρά από πολύγωνα. Επίσης όλες αυτές οι τεχνικές χρησιμοποιούν την έννοια του μοναδιαίου διανύσματος προσανατολισμού (*normal*) για κάθε σημείο του χώρου. Το διάνυσμα αυτό έχει κατεύθυνση κάθετη προς το επίπεδο της επιφάνειας στην οποία ανήκει το σημείο.

Οι πλευρικές τεχνικές είναι οι πιο απλές και οι πιο γρήγορες από τις δύο κατηγορίες σκίασης που αναφέραμε. Οι τεχνικές αυτές υπολογίζουν ένα μόνο διάνυσμα προσανατολισμού για όλα τα σημεία ενός πολυγώνου. Το διάνυσμα αυτό σε ορισμένες περιπτώσεις υπολογίζεται με βάση τον προσανατολισμό του κέντρου του πολυγώνου ενώ σε άλλες περιπτώσεις προέρχεται από τον υπολογισμό της μέσης τιμής των διανυσμάτων προσανατολισμού όλων των κορυφών του πολυγώνου. Σε αντικείμενα στα οποία έχει εφαρμοστεί μια πλευρική τεχνική σκίασης είναι δυνατόν να διακρίνουμε τα στοιχειώδη πολύγωνα από τα οποία αποτελούνται επομένως δίνουν την αίσθηση μιας ασυνεχούς επιφάνειας. Η πιο γνωστή μέθοδος πλευρικής σκίασης είναι το μοντέλο του *Lambert* (*Lambert shading*).

Οι ομαλές τεχνικές σκίασης υπολογίζουν διανύσματα προσανατολισμού που μεταβάλλονται με ομαλό τρόπο μεταξύ των πολυγώνων που απαρτίζουν μια σκηνή. Οι περισσότερες από τις τεχνικές που ανήκουν σε αυτήν την κατηγορία υπολογίζουν αρχικά τα μοναδιαία διανύσματα προσανατολισμού στο κέντρο κάθε πολυγώνου. Στη συνέχεια υπολογίζονται τα διανύσματα προσανατολισμού για κάθε κορυφή ενός πολυγώνου με βάση τη μέση τιμή των διανυσμάτων προσανατολισμού των πολυγώνων τα οποία συναντώνται στην κορυφή αυτή. Τέλος τα διανύσματα προσανατολισμού για τα υπόλοιπα σημεία κάθε πολυγώνου υπολογίζονται με βάση τις τιμές των διανυσμάτων προσανατολισμού στις κορυφές του πολυγώνου. Οι ομαλές τεχνικές σκίασης δημιουργούν επιφάνειες οι οποίες δίνουν την εντύπωση ομαλών επιφανειών καθώς οι μεταβολές των διανυσμάτων προσανατολισμού μεταξύ των πολυγώνων που τις απαρτίζουν είναι βαθμωτές και όχι απότομες όπως στην περίπτωση των πλευρικών τεχνικών.

Οι πιο γνωστές τεχνικές ομαλής σκίασης είναι η *σκίαση κατά Gouraud* (*Gouraud shading*) και η *σκίαση κατά Phong* (*Phong shading*). Η κύρια διαφορά μεταξύ των δύο αυτών τεχνικών είναι ότι η σκίαση κατά Phong μπορεί να αποδώσει καλύτερα από τη σκίαση κατά Gouraud τις *ανταύγειες* (*highlights*) που παράγονται από γυαλιστερές επιφάνειες σε περιοχές της επιφάνειας οι οποίες φωτίζονται απευθείας από φωτεινές πηγές.

Η VRML παρέχει τη δυνατότητα χρησιμοποίησης των πεδίων *normal*, *normalIndex* και *normalPerVertex* στον κόμβο *IndexedFaceSet* για να ορίσει ομαλές ή πλευρικές τεχνικές σκίασης. Πιο συγκεκριμένα για κάθε πολύγωνο που περιλαμβάνεται σε ένα κόμβο *IndexedFaceSet* μπορούμε να ορίσουμε τα διανύσματα προσανατολισμού που θα χρησιμοποιηθούν για τη σκίαση του. Στην περίπτωση αυτή η τιμή του λογικού πεδίου *normalPerVertex* καθορίζει αν τα διανύσματα αυτά θα εφαρμοστούν μόνο στις κορυφές (*normalPerVertex=TRUE*) ή σε όλα τα σημεία του πολυγώνου (*normalPerVertex=FALSE*). Στην πρώτη περίπτωση η VRML θα χρησιμοποιήσει μία ομαλή μέθοδο σκίασης για να υπολογίσει τα διανύσματα προσανατολισμού για τα υπόλοιπα σημεία κάθε πολυγώνου.

Το παρακάτω πρόγραμμα εφαρμόζει μια πλευρική μέθοδο σκίασης:

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
    }
  }
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        0 0 0,
        1 0 0.5,
        2 0 0.5,
        3 0 0,
        0 1 0,
        1 1 0.5,
        2 1 0.5,
        3 1 0
      ]
    }
    coordIndex [
      0, 1, 5, 4, -1,
      1, 2, 6, 5, -1,
      2, 3, 7, 6, -1
    ]
    normal Normal {
      vector [
        -0.45 0 0.89,
        0 0 1,
        0.45 0 0.89
      ]
    }
  }
}
```

```

    ]
  }
  normalPerVertex FALSE
}
}

```

ενώ το ακόλουθο πρόγραμμα εφαρμόζει μία ομαλή μέθοδο σκίασης στο ίδιο σχήμα με το προηγούμενο πρόγραμμα:

```

#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
    }
  }
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        0 0 0,
        1 0 0.5,
        2 0 0.5,
        3 0 0,
        0 1 0,
        1 1 0.5,
        2 1 0.5,
        3 1 0
      ]
    }
  }
  coordIndex [
    0, 1, 5, 4, -1,
    1, 2, 6, 5, -1,
    2, 3, 7, 6, -1
  ]
  normal Normal {
    vector [
      -0.45 0 0.89,
      -0.23 0 0.97,
      0.23 0 0.97,
      0.45 0 0.89,
      -0.45 0 0.89,
      -0.23 0 0.97,
      0.23 0 0.97,
    ]
  }
}

```

```
    0.45 0 0.89
  ]
}
normalPerVertex TRUE
}
}
```