

Κεφάλαιο 6°

Φωτορεαλισμός

Εισαγωγή

Στο Κεφάλαιο 2 αναλύσαμε τις μεθόδους απόδοσης χρωμάτων σε επιφάνειες κατά τη σχεδιάσή τους στη σκηνή. Ωστόσο, εάν ενδιαφερόμαστε για την απόδοση σκηνών που προσομοιώνουν τον πραγματικό κόσμο, τα εργαλεία που αναλύσαμε έως τώρα δεν επαρκούν. Προκειμένου να δημιουργήσουμε μια ρεαλιστική απεικόνιση, θα πρέπει να λάβουμε υπόψη και να προσομοιώσουμε τον τρόπο με τον οποίο παράγεται μια εικόνα στον πραγματικό κόσμο.

Δεδομένου ότι στην πράξη μια εικόνα σχηματίζεται βάσει των επίδρασης πηγών φωτισμού στις επιφάνειες των αντικειμένων, το πώς θα αποδοθεί η σκηνή καθορίζεται από τις ιδιότητες των πηγών φωτισμού αλλά και από τις ιδιότητες των επιφανειών.

Στο κεφάλαιο αυτό αναλύουμε το φυσικό μοντέλο φωτισμού στο οποίο βασίζονται οι βιβλιοθήκες γραφικών για την αναπαράσταση φωτορεαλιστικών σκηνών. Αναφέρονται οι κατηγορίες και οι ιδιότητες των πηγών φωτισμού, καθώς και οι ιδιότητες των επιφανειών σε ό,τι αφορά την ανακλαστικότητα τους. Επίσης, αναλύονται και οι τρόποι με τους οποίους οι παράμετροι αυτές καθορίζονται στο μοντέλο σκίασης της OpenGL.

6.1 Πηγές φωτισμού

Από πλευράς Φυσικής, μια εικόνα σχηματίζεται από την πρόσπτωση ακτινών φωτός σε μια φωτοευαίσθητη επιφάνεια (όπως λ.χ. στο ανθρώπινο μάτι ή στον αισθητήρα μιας κάμερας). Επομένως, προκειμένου να σχηματιστεί μια εικόνα απαιτείται η ύπαρξη φωτεινής ενέργειας, η οποία προέρχεται από μία ή περισσότερες πηγές εκπομπής (**πηγές φωτισμού**). Χωρίς πηγές φωτισμού δεν είναι εφικτός ο σχηματισμός εικόνας.

Στην OpenGL προκειμένου να χρησιμοποιήσουμε το μοντέλο σκίασης απαιτείται η ενεργοποίησή του δίνοντας στην εντολή **glEnable** το όρισμα **GL_LIGHTING**:

```
glEnable( GL_LIGHTING );
```

Προκειμένου να οριστεί μια πηγή φωτισμού, πρέπει να προσδιορίσουμε ένα σύνολο παραμέτρων που χαρακτηρίζουν τη συμπεριφορά της. Τα χαρακτηριστικά μιας πηγής φωτισμού διακρίνονται σε δύο κατηγορίες:

α) Χαρακτηριστικά που προσδιορίζονται με μια αριθμητική τιμή:

Αυτά ορίζονται με την εντολή *glLightf*

void glLightf (GLenum light, GLenum parameterName, GLfloat parameter);

Το όρισμα *light* είναι μια συμβολική σταθερά που προσδιορίζει την πηγή φωτισμού στην οποία αποδίδουμε το χαρακτηριστικό. Οι υλοποιήσεις της OpenGL υποστηρίζουν τουλάχιστον οκτώ πηγές φωτισμού. Επομένως, η σταθερά *light* μπορεί να δεχτεί τουλάχιστον τις τιμές από *GL_LIGHT0* έως *GL_LIGHT7*.

Το όρισμα *parameterName* είναι μια συμβολική σταθερά που καθορίζει το χαρακτηριστικό που αποδίδουμε. Στις επόμενες παραγράφους περιγράφονται αναλυτικά κάθε ένα από τα χαρακτηριστικά.

Το όρισμα *parameter* δηλώνει την αριθμητική τιμή που αποδίδουμε στην ιδιότητα *parameterName*.

β) Χαρακτηριστικά που προσδιορίζονται από ένα σύνολο αριθμητικών τιμών:

Αυτά ορίζονται με την εντολή *glLightfv*:

void glLightfv (GLenum light, GLenum parameterName, const GLfloat * parameterArray);

Το όρισμα *parameterArray* είναι δείκτης σε μητρώο. Το μητρώο αυτό περιέχει το σύνολο των τιμών που προσδιορίζουν το χαρακτηριστικό *parameterName*.

Αφού δηλωθούν τα χαρακτηριστικά των πηγών φωτισμού, απαιτείται επιπλέον η ενεργοποίηση κάθε μίας πηγής ξεχωριστά. Μια πηγή φωτισμού ενεργοποιείται δίνοντας το όνομά της ως όρισμα στην *glEnable*:

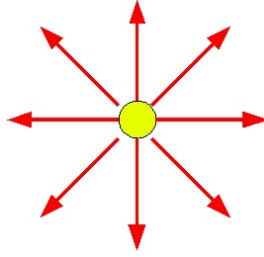
glEnable (GL_LIGHTx);

6.2 Κατηγορίες πηγών φωτισμού

Οι πηγές φωτισμού μπορούν να διακριθούν στις παρακάτω κατηγορίες.

6.2.1 Σημειακές πηγές

Οι σημειακές πηγές φωτισμού καταλαμβάνουν άπειρα μικρό χώρο και δηλώνονται ορίζοντας τη θέση τους στη σκηνή. Σε μια ομοιόμορφη σημειακή πηγή, οι ακτίνες διαδίδονται σφαιρικά προς όλες τις κατευθύνσεις και με την ίδια ένταση όπως φαίνεται στο Σχήμα 6.1.



Σχ. 6.1: Σημειακή ομοιόμορφη πηγή φωτισμού. Οι εκπεμπόμενες ακτίνες διαδίδονται σφαιρικά και με ίση ένταση προς όλες τις κατευθύνσεις

Για να δηλώσουμε μια σημειακή πηγή στη σκηνή, προσδιορίζουμε τη θέση της με την εντολή

```
glLightfv(lightName, GL_POSITION, positionVector);
```

όπου *positionVector* δείκτης σε μητρώο με τέσσερα στοιχεία. Τα τρία πρώτα στοιχεία του ορίζουν τη θέση και το τέταρτο στοιχείο ορίζει εάν η πηγή φωτός είναι σημειακή. Με μη μηδενική τιμή ορίζουμε μια σημειακή πηγή.

Πχ, με τις εντολές

```
GLfloat light0Position[] = {5,5,0,1}  
glLightfv (GL_LIGHT0, GL_POSITION, light0Position);
```

ορίζουμε ότι η πηγή *GL_LIGHT_0* είναι σημειακή (το τέταρτο στοιχείο του μητρώου *light0Position* είναι μη μηδενικό) και θέση της πηγής είναι στο σημείο $(x, y, z) = (5,5,0)$

6.2.2 Ακτινική εξασθένηση φωτεινότητας

Σε πραγματικές συνθήκες η ένταση των σημειακών πηγών εξασθενεί κατά την απομάκρυνσή μας από τη πηγή. Αυτή η εξασθένηση ονομάζεται **ακτινική εξασθένηση (radial attenuation)** και οφείλεται στο διασκορπισμό της εκπεμπόμενης ισχύος σε συνεχώς μεγαλύτερες σφαιρικές επιφάνειες, καθώς αυξάνεται η απόσταση από την πηγή. Συγκεκριμένα, η εξασθένηση της έντασης μιας σημειακής πηγής συναρτήσει της απόστασης ακολουθεί το νόμο του αντιστρόφου τετραγώνου, δηλαδή σε ένα σημείο που απέχει απόσταση d από την πηγή για τον υπολογισμό της έντασης λαμβάνεται υπόψη ο παράγοντας εξασθένησης

$$f_{radatt}(d) = \frac{1}{d^2}$$

Ωστόσο η υλοποίηση του μοντέλου αυτού για τη σύνθεση σκηνών με υπολογιστή δεν εξάγει ρεαλιστικά

αποτελέσματα, διότι το μοντέλο θεωρεί σημειακές πηγές απείρως μικρού μεγέθους. Τέτοιες πηγές δεν υφίστανται στην πράξη. Σε γραφικές εφαρμογές, αντί του προηγούμενου μοντέλου, συνήθως θεωρούμε ότι ο παράγοντας ακτινικής εξασθένησης περιγράφεται επιπλέον με ένα γραμμικό και έναν σταθερό όρο, δηλαδή:

$$f_{\text{radatt}}(d) = \frac{1}{a_0 + a_1 \cdot d + a_2 \cdot d^2}$$

Οι τιμές των συντελεστών a_0 , a_1 , a_2 επιλέγονται αυθαίρετα από τον προγραμματιστή, ούτως ώστε να προκύψει το επιθυμητό αισθητικό αποτέλεσμα. Επιπλέον, εάν χρησιμοποιούμε πολλαπλές πηγές φωτισμού, υπάρχει η δυνατότητα να ανατεθούν διαφορετικοί συντελεστές εξασθένησης σε κάθε μία.

Στην OpenGL, οι συντελεστές εξασθένησης για την πηγή φωτισμού *lightName* (*lightName=GL_LIGHT0, ..., GL_LIGHT7*) ορίζονται χρησιμοποιώντας τις εντολές

glLightf (lightName, GL_CONSTANT_ATTENUATION, a0);

για τον καθορισμό του σταθερού όρου a_0

glLightf (lightName, GL_LINEAR_ATTENUATION, a1);

για τον καθορισμό του γραμμικού όρου a_1

και

glLightf(lightName, GL_QUADRATIC_ATTENUATION, a2);

για τον καθορισμό του τετραγωνικού όρου a_2 .

Οι συντελεστές a_0 , a_1 , a_2 δίνονται ως αριθμοί κινητής υποδιαστολής. Εάν δεν δηλωθούν από τον προγραμματιστή, ισχύουν οι εξής προκαθορισμένες τιμές:

$$a_0 = 1$$

$$a_1 = 0$$

$$a_2 = 0$$

Αυτό σημαίνει ότι, με τις προκαθορισμένες τιμές, δεν υφίσταται ακτινική εξασθένηση.

6.2.3 Πηγές σε άπειρη απόσταση:

Σε ορισμένες περιπτώσεις θεωρούμε ότι η πηγή φωτισμού βρίσκεται σε άπειρη απόσταση από τη σκηνή.

Ακολουθούμε λ.χ. αυτή την παραδοχή όταν μοντελοποιούμε το φωτισμό ενός τοπίου από τον ήλιο. Στην περίπτωση πηγών σε άπειρη απόσταση, όλες οι εκπεμπόμενες ακτίνες ακολουθούν παράλληλες τροχιές σε αντίθεση με την περίπτωση σημειακών πηγών που οι ακτίνες διαδίδονται σφαιρικά.

Στην OpenGL, ορίζουμε μια πηγή σε άπειρη απόσταση με την εντολή

glLightfv(GL_LIGHTX, GL_POSITION, positionVector);

με το τέταρτο στοιχείο του μητρώου *positionVector* ίσο με μηδέν. Στην περίπτωση αυτή, τα τρία πρώτα στοιχεία του μητρώου *positionVector* δε δηλώνουν τη θέση της πηγής (εφόσον βρίσκεται σε άπειρη απόσταση) αλλά το διάνυσμα κατεύθυνσης των ακτίνων που αναχωρούν από την πηγή.

Π.χ. με τις εντολές

```
GLfloat light0Position[]={1,0,1,0};  
glLightfv(GL_LIGHT0, GL_POSITION, light0Position);
```

δηλώνουμε ότι η πηγή *GL_LIGHT0* βρίσκεται σε άπειρη απόσταση, διότι το τέταρτο στοιχείο του μητρώου *light0Position* ισούται με μηδέν. Επιπλέον δηλώνουμε ότι **οι ακτίνες της πηγής προέρχονται από την κατεύθυνση που ορίζει το διάνυσμα {1,0,1}**.

Το μοντέλο ακτινικής εξασθένησης δεν εφαρμόζεται σε πηγές φωτισμού που βρίσκονται σε άπειρη απόσταση, διότι τότε η ένταση του φωτός στη σκηνή θα μηδενιζόταν. Στην περίπτωση αυτή, ο συντελεστής ακτινικής εξασθένησης είναι ίσος με 1. Επομένως στη γενική περίπτωση ισχύει:

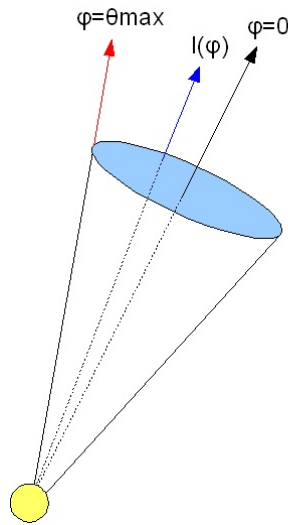
$$f_{\text{radatt}}(d) = \begin{cases} 1 & \text{πηγή σε άπειρη απόσταση} \\ \frac{1}{a_0 + a_1 \cdot d + a_2 \cdot d^2} & \text{σημειακή πηγή σε κοντινή απόσταση} \end{cases}$$

Εάν ο προγραμματιστής δεν ορίσει θέση πηγής φωτισμού, η μηχανή της OpenGL θεωρεί ως προκαθορισμένη επιλογή μια πηγή σε άπειρη απόσταση που οι ακτίνες της κατευθύνονται προς την αρνητική κατεύθυνση του άξονα z (0,0,1,0).

6.2.4 Κατευθυντικές πηγές (σποτ)

Μια σημειακή πηγή φωτισμού μπορεί να καθοριστεί ούτως ώστε να εκπέμπει σε ένα περιορισμένο γωνιακό εύρος, όπως ένα φωτιστικό σποτ. Σε αυτή την περίπτωση, το τμήμα της σκηνής που φωτίζεται από την κατευθυντική πηγή περικλείεται σε έναν κώνο. Το άνοιγμα του κώνου διάχυσης καθορίζεται από μια

γωνία αποκοπής θ_{\max} , όπως φαίνεται στο Σχ. 6.2. Σημεία της σκηής που βρίσκονται εκτός του κώνου διάχυσης δε φωτίζονται καθόλου από τη κατευθυντική πηγή

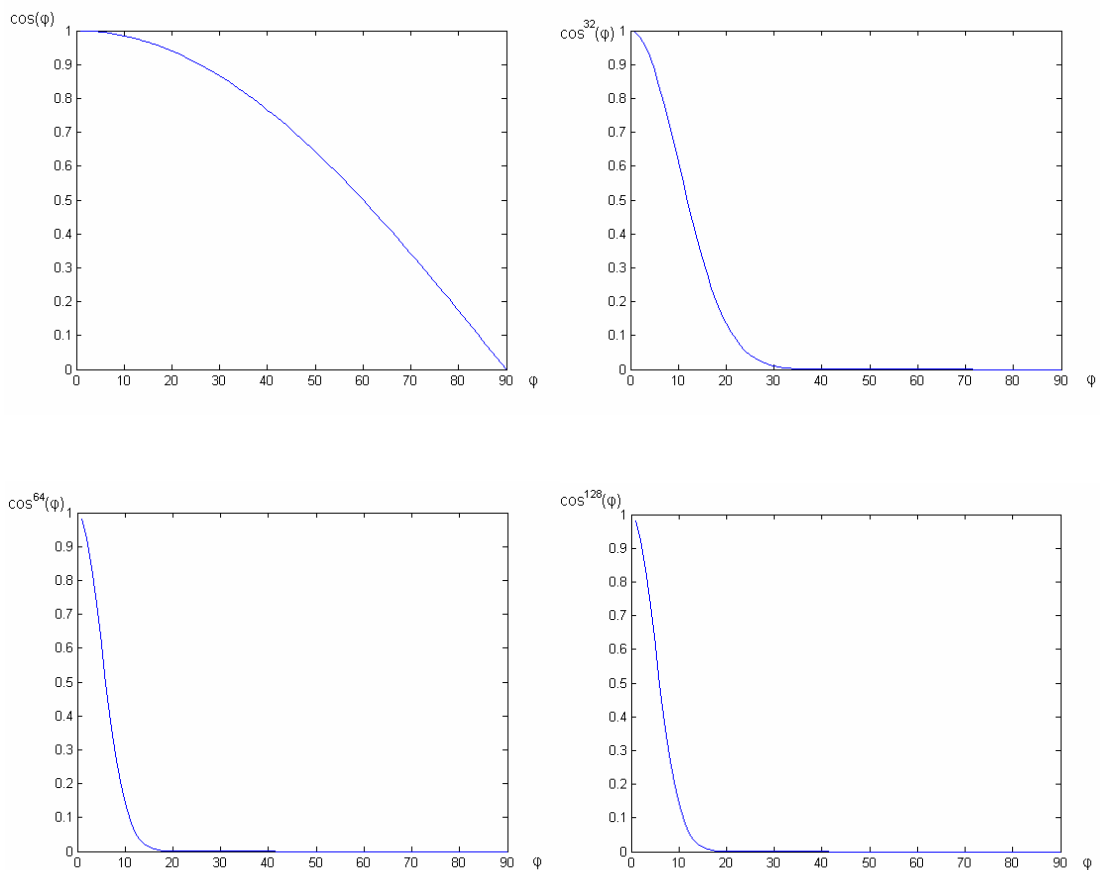


Σχ. 6.2 Κατευθυντική πηγή φωτισμού

Όπως στις σημειακές, έτσι και στις κατευθυντικές πηγές φωτισμού, κάθε εκπεμπόμενη ακτίνα εξασθενεί συναρτήσει της απόστασης από την πηγή. Επιπλέον όμως, εξασθενεί και συναρτήσει της γωνιακής της απόκλισης από τον άξονα του κώνου φωτισμού. Επομένως, ένας παρατηρητής παρατηρεί τη μέγιστη ένταση όταν βρίσκεται επί του άξονα του κώνου ($\phi = 0$). Όσο αποκλίνει γωνιακά από τον άξονα εκπομπής, η παρατηρούμενη ένταση $I(\phi)$ εξασθενεί, βάσει του ακόλουθου παράγοντα γωνιακής εξασθένησης (angular attenuation) $f_{angatt}(\phi)$:

$$f_{angatt}(\phi) = \cos^n(\phi)$$

Ο εκθέτης n καθορίζει κατά πόσο διαχέεται η εκπεμπόμενη ένταση γύρω από τη θέση μέγιστης ακτινοβολίας. Θέτοντας μεγάλες τιμές στον εκθέτη, ορίζουμε μια ισχυρά κατευθυντική πηγή και η εκπεμπόμενη ένταση μειώνεται ραγδαία συναρτήσει της γωνιακής απόκλισης ϕ . Μικρές τιμές του εκθέτη προκαλούν μικρότερη διακύμανση της εκπεμπόμενης έντασης στο γωνιακό εύρος που καλύπτει η πηγή. Τα διαγράμματα του Σχ. 6.3 παρουσιάζουν τις καμπύλες του συντελεστή εξασθένησης $\cos^n \phi$ για διάφορες τιμές του εκθέτη n .



Σχ 6.3: Καμπύλες συντελεστή γωνιακής εξασθένησης για τιμές εκθέτη $n = 1, 32, 64, 128$

6.3 Χαρακτηριστικά πηγών φωτισμού

Οι πηγές φωτισμού προσδιορίζονται με τις εξής ιδότητες:

6.3.1 Ένταση και χρώμα πηγής

Τόσο η ένταση όσο και το χρώμα καθορίζονται δίνοντας τα βάρη των χρωματικών συνιστωσών της πηγής στο μοντέλο RGB. Πχ. επιλέγοντας τα βάρη $\{1,1,1\}$ δημιουργούμε μια πηγή λευκού φωτός. Πηγές με ίσα βάρη $\{a,a,a\}$, ($0 < a < 1$), εκπέμπουν τόνους του γκριζου. Προφανώς μια πηγή φωτός με βάρη $\{0,0,0\}$ δεν εκπέμπει καθόλου φως. (Το μαύρο χρώμα δηλώνει έλλειψη φωτός καθώς όλες οι χρωματικές συνιστώσες είναι μηδενικές.)

Προφανώς, μπορούμε να δηλώσουμε πηγές χρωματισμένου φωτός, ορίζοντας διαφορετικές εντάσεις για κάθε χρωματική συνιστώσα της πηγής στο μοντέλο RGB. Πχ ορίζουμε μία πηγή έντονου κίτρινου φωτός δίνοντας τα βάρη $\{1,1,0\}$.

Δεδομένου ότι δήλωση έντασης και χρώματος πηγών προϋποθέτει μια αρχική εξοικείωση με το μοντέλο ανακλάσεων, οι αντίστοιχες εντολές θα αναλυθούν σε επόμενη παράγραφο του κεφαλαίου.

6.3.2 Θέση πηγής

Για την τοποθέτηση μιας πηγής φωτισμού στο χώρο, απλά δηλώνουμε τις συντεταγμένες σκηνής του σημείου τοποθέτησης. Επισημαίνουμε ότι η θέση μιας πηγής φωτισμού, όπως και όλα τα σημεία της σκηνής, υπόκειται τους μετασχηματισμούς συντεταγμένων που ορίζει το μητρώο μετασχηματισμού μοντέλου.

6.3.3 Κατεύθυνση εκπομπής - Γωνία αποκοπής

Η κατεύθυνση εκπομπής ορίζεται για κατευθυντικές πηγές και δηλώνεται με τη χρήση ενός διανύσματος κατεύθυνσης dir .

$$dir = (dir_x, dir_y, dir_z)$$

Η γωνία αποκοπής προσδιορίζει το εύρος της κωνικής περιοχής που καλύπτει η κατευθυντική πηγή και προσδιορίζεται με μια τιμή ϕ_{max} . Περιοχές που βρίσκονται έξω από τη γωνία φωτισμού ϕ_{max} υφίστανται μηδενική επίδραση από την πηγή.

6.4 Υλοποίηση κατευθυντικών πηγών

Στην OpenGL μια κατευθυντική πηγή υλοποιείται ορίζοντας τις εξής τρεις παραμέτρους: την κατεύθυνση μέγιστης εκπομπής, τη γωνία κάλυψης και τη γωνιακή εξασθένηση της εκπεμπόμενης έντασής της.

6.4.1 Κατεύθυνση μέγιστης εκπομπής

Η κατεύθυνση εκπομπής καθορίζει τον άξονα εκπομπής της πηγής και ορίζεται προσδιορίζοντας με την εντολή `glLightfv` την παράμετρο `GL_SPOT_DIRECTION`:

`glLightfv(GL_LIGHTX, GL_SPOT_DIRECTION, directionVector);`

όπου `directionVector` το διάνυσμα που καθορίζει τη διεύθυνση εκπομπής.

Εάν για μια κατευθυντική πηγή δεν καθορίσουμε διεύθυνση εκπομπής η μηχανή της OpenGL θεωρεί ως προκαθορισμένη επιλογή την κατεύθυνση προς τον αρνητικό ημιάξονα z $(0,0,-1)$.

6.4.2 Γωνία κάλυψης

Το γωνιακό εύρος που καλύπτει μια κατευθυντική πηγή ορίζεται προσδιορίζοντας με την εντολή ***glLightf*** την παράμετρο *GL_SPOT_CUTOFF*:

glLightf(GL_LIGHTX, GL_SPOT_CUTOFF, cutoffAngle);

όπου *cutoffAngle* η γωνία αποκοπής της πηγής. Αντικείμενα που αποκλίνουν από τον κύριο άξονα εκπομπής της πηγής κατά γωνίες μεγαλύτερες της *cutoffAngle* δε φωτίζονται καθόλου από αυτήν. Εάν για μια σημειακή πηγή δεν ορίσουμε γωνία αποκοπής, η μηχανή της OpenGL θεωρεί ως προκαθορισμένη γωνία αποκοπής τις 180 μοίρες. Επομένως στην περίπτωση αυτή η πηγή εκπέμπει φως προς όλες τις κατευθύνσεις.

6.4.3 Γωνιακή εξασθένηση έντασης

Η γωνιακή εξασθένηση καθορίζεται δηλώνοντας τον εκθέτη εξασθένησης που ορίζεται στο μοντέλο Phong. Η δήλωσή του γίνεται προσδιορίζοντας με την εντολή ***glLightf*** την παράμετρο *GL_SPOT_EXPONENT*:

glLightf(GL_LIGHTX, GL_SPOT_EXPONENT, exponent);

όπου *exponent* η τιμή του εκθέτη, η οποία κυμαίνεται μεταξύ των τιμών 0 και 128. Εάν δεν καθορίσουμε την τιμή του εκθέτη, η μηχανή της OpenGL θεωρεί ως προκαθορισμένη τιμή το 0. Επομένως, στην περίπτωση αυτή, η κατευθυντική πηγή εκπέμπει με ομοιόμορφη ένταση σε όλο το γωνιακό εύρος της.

6.5 Ανακλώσες επιφάνειες - Χαρακτηριστικά επιφανειών

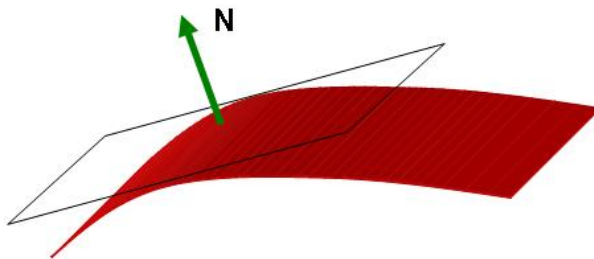
Στον πραγματικό κόσμο, ένας παρατηρητής αντιλαμβάνεται την παρουσία επιφανειών από τις ανακλάσεις που αυτές προκαλούν στις ακτίνες των πηγών φωτισμού. Κάθε επιφάνεια γίνεται αντιληπτή με διαφορετικά χαρακτηριστικά ανάλογα με τα χαρακτηριστικά της ανακλαστικότητάς της (χρώμα). Επιπλέον όμως, παίζει ρόλο και η σχετική θέση της κάθε επιφάνειας ως προς τις υπάρχουσες πηγές φωτισμού.

Δεδομένου ότι η ανάκλαση ακτίνων φωτός από επιφάνειες της σκηνής περιγράφεται βάσει συγκεκριμένων φυσικών νόμων, για τη ρεαλιστική απόδοση των φαινομένων αυτών, το μοντέλο σκίασης θα πρέπει να λαμβάνει υπόψη τους νόμους αυτούς.

6.5.1 Προσανατολισμός επιφανείας – Κανονικό διάνυσμα

Προκειμένου να περιγράψουμε τον προσανατολισμό μιας επιφάνειας σε κάθε σημείο της,

χρησιμοποιούμε την έννοια του **κανονικού διανύσματος**. Εάν θεωρήσουμε μια καμπύλη επιφάνεια στο χώρο, τότε, σε κάθε σημείο της επιφάνειας ορίζουμε ως κανονικό διάνυσμα το διάνυσμα που είναι κάθετο στο εφαπτόμενο επίπεδο του συγκεκριμένου σημείου (Σχ. 6.4).



Σχ. 6.4: Κανονικό διάνυσμα σημείου επιφάνειας. Το κανονικό διάνυσμα είναι κάθετο στο εφαπτόμενο επίπεδο που ορίζεται για το εκάστοτε σημείο της επιφάνειας.

Δεδομένου ότι σε εφαρμογές γραφικών οι επιφάνειες προσεγγίζονται από πολυγωνικά πλέγματα, κάθε στοιχειώδη πολυγωνική επιφάνεια συμπίπτει με το εφαπτόμενο επίπεδό της. Επομένως, εξ'ορισμού, το κανονικό διάνυσμα μιας πολυγωνικής επιφάνειας είναι κάθετο σε αυτήν. Εάν θεωρήσουμε λοιπόν τρεις κορυφές ενός του πολυγώνου (x_1, y_1, z_1) , (x_2, y_2, z_2) και (x_3, y_3, z_3) , το κανονικό διάνυσμά του προκύπτει από το εξωτερικό γινόμενο των διανυσμάτων που ορίζουν οι κορυφές του, δηλαδή

$$\vec{N} = \begin{vmatrix} \vec{x} & \vec{y} & \vec{z} \\ (x_2 - x_1) & (y_2 - y_1) & (z_2 - z_1) \\ (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) \end{vmatrix}$$

Ο παραπάνω τύπος του ορισμού κανονικού διανύσματος επιφάνειας εφαρμόζεται από την OpenGL, όταν ο προγραμματιστής δε δηλώνει τιμές για αυτό. Ωστόσο ο προγραμματιστής μπορεί να ρυθμίσει τον προσανατολισμό μιας επιφάνειας, δηλώνοντας τις συνιστώσες του κανονικού διανύσματος για κάθε κορυφή της επιφάνειας. Μπορούμε να αναθέσουμε σε κάθε κορυφή ένα κανονικό διάνυσμα με την εντολή **glNormal3***:

glNormal3{bsifd}(TYPE nx, TYPE ny TYPE nz);

όπου *nx*, *ny* και *nz* οι συνιστώσες του κανονικού διανύσματος.

Επίσης μπορούμε να δώσουμε τις συνιστώσες του κανονικού διανύσματος με τη μορφή μητρώου, έστω *normalVector*, με την εντολή **glNormal3{bsifd}v**.

void glNormal3{bsifd}v(TYPE *normalVector);

Από τη στιγμή που θα καθορίσουμε τον προσανατολισμό του κανονικού διανύσματος, αυτός ανατίθεται

σε κάθε κορυφή που δηλώνουμε στη συνέχεια του προγράμματος. Δηλαδή ο προσανατολισμός του κανονικού διανύσματος είναι μια μεταβλητή κατάσταση. Προκειμένου να ορίσουμε σε κάθε κορυφή μιας επιφάνειας διαφορετικό προσανατολισμό, απαιτείται η αλλαγή του κανονικού διανύσματος.

Π.χ. με τον κώδικα

```
GLfloat nv1[]={1,1,1};  
glNormal3fv(nv1);  
glVertex3fv(vertex1);
```

```
GLfloat nv2[]={1,1,0};  
glNormal3fv(nv2);  
glVertex3fv(vertex2);
```

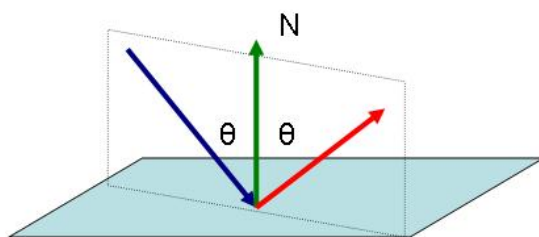
```
GLfloat nv3[]={0,1,1};  
glNormal3fv(nv3);  
glVertex3fv(vertex3);
```

αναθέτουμε στην κορυφή *vertex1* το κανονικό διάνυσμα *nv1*, στην κορυφή *vertex2* το κανονικό διάνυσμα *nv2* και στην κορυφή *vertex3* το κανονικό διάνυσμα *nv3*.

Αφού ορίζουμε ένα κανονικό διάνυσμα ανά κορυφή, τα κανονικά διανύσματα όλων των κορυφών μιας πολυγωνικής επιφάνειας χρησιμοποιούνται για τον προσδιορισμό της φωτεινότητας κάθε εσωτερικού σημείου της επιφάνειας, βάσει της τεχνικής που θα αναλυθεί στην ενότητα “Σκίαση Gouraud”.

6.5.2 Νόμος ανάκλασης – Συντελεστές ανάκλασης

Το κανονικό διάνυσμα κάθε σημείου βοηθάει στο να εντοπίσουμε τη διεύθυνση που θα ακολουθήσει μια ακτίνα φωτός, μετά την ανάκλασή της από το σημείο αυτό. Η πορεία της ανακλώμενης ακτίνας καθορίζεται από τη **γωνία πρόσπτωσης**, δηλαδή τη γωνία που σχηματίζει η προσπίπτουσα ακτίνα με το κανονικό διάνυσμα στο σημείο πρόσπτωσης. Σύμφωνα με το νόμο της ανάκλασης **η γωνία πρόσπτωσης και η γωνία ανάκλασης είναι ίσες**. Επιπλέον, στις τρεις διαστάσεις, το διάνυσμα της προσπίπτουσας ακτίνας, το διάνυσμα της ανακλώμενης ακτίνας και το κανονικό διάνυσμα βρίσκονται στο ίδιο επίπεδο, όπως φαίνεται στο Σχ. 6.5.



Σχ. 6.5: Νόμος ανάκλασης. Οι γωνίες πρόσπτωσης και ανάκλασης είναι ίσες. Επιπλέον, η προσπίπτουσα και η ανακλώμενη ακτίνα βρίσκονται στο ίδιο επίπεδο.

Επιπρόσθετα, η ένταση μιας ανακλώμενης ακτίνας, σε σχέση με την αντίστοιχη ένταση της προσπίπτουσας ακτίνας, παρουσιάζει εξασθένιση, λόγω απορρόφησης ενέργειας από την επιφάνεια. Αυτό το ποσοστό εξασθένισης ονομάζεται **συντελεστής ανάκλασης** και είναι ένα χαρακτηριστικό της επιφάνειας που επιλέγεται από τον προγραμματιστή (με τιμή μεταξύ 0 και 1). Μια επιφάνεια με συντελεστή ανάκλασης 1 ανακλά όλη την ενέργεια που προσπίπτει πάνω της, ενώ μια επιφάνεια με μηδενικό συντελεστή ανάκλασης παρουσιάζει τη μέγιστη απορρόφηση.

6.5.3 Χρώμα επιφανείας

Μια επιφάνεια μπορεί να προκαλεί διαφορετικό βαθμό εξασθένισης σε ακτίνες διαφορετικού χρώματος. Ουσιαστικά, με αυτό τον τρόπο ορίζεται το χρώμα μιας επιφάνειας. Έτσι, μία κόκκινη επιφάνεια εκλαμβάνεται από έναν παρατηρητή ως κόκκινη επειδή ανακλά μόνο την κόκκινη συνιστώσα της προσπίπτουσας ακτίνας φωτός. Οι συντελεστές ανάκλασης των άλλων δύο βασικών χρωμάτων είναι μηδενικοί για την επιφάνεια αυτή.

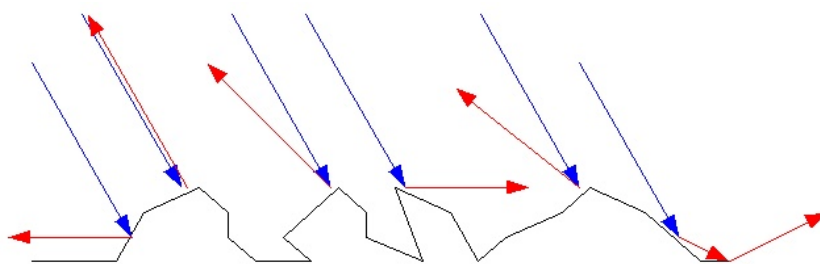
Επομένως, το χρώμα μιας επιφάνειας καθορίζεται από τους συντελεστές ανάκλασης των τριών βασικών χρωματικών συνιστωσών. Μπορούμε λοιπόν να καθορίσουμε το χρώμα μιας επιφάνειας δίνοντας ως συντελεστές ανάκλασης (k_r, k_g, k_b) τα βάρη των συνιστωσών που περιγράφουν το χρώμα στο μοντέλο RGB.

Πχ αν για μια επιφάνεια ορίσουμε συντελεστές ανάκλασης $\{0,1,0\}$, η επιφάνεια αυτή θα εμφανίζεται πράσινη στον παρατηρητή, εάν εκτεθεί σε λευκό φως.

Προφανώς, μεταβάλλοντας τη χρωματική σύνθεση της πηγής φωτισμού, μεταβάλλουμε και τον τρόπο με τον οποίο αποδίδεται μια επιφάνεια με συγκεκριμένο χρώμα. Πχ μια επιφάνεια που ανακλά μόνο το κόκκινο φως δεν ανακλά καθόλου φως εάν εκτεθεί σε πηγή που εκπέμπει μόνο πράσινο φως, καθώς δεν υπάρχει επικάλυψη στις χρωματικές συνιστώσες πηγής και επιφανείας. Στην περίπτωση αυτή η επιφάνεια θα φαίνεται μαύρη.

6.6 Διάχυτες ανακλάσεις

Όλες οι ανακλάσεις ακολουθούν το φυσικό νόμο που περιγράψαμε στην προηγούμενη παράγραφο. Ωστόσο, στην πράξη, οι ανακλώσες επιφάνειες, σε μικροσκοπικό επίπεδο, παρουσιάζουν τυχαιότητα στον προσανατολισμό τους. Αυτός ο τυχαίος προσανατολισμός της επιφάνειάς τους έχει ως αποτέλεσμα, προσπίπτουσες παράλληλες δέσμες να ακολουθούν διαφορετικές τροχιές μετά την ανάκλασή τους, καθώς, για κάθε μία δέσμη ορίζεται διαφορετική γωνία πρόσπτωσης. Το φαινόμενο αυτό ονομάζεται **διάχυση** (diffusion) και αναπαρίσταται στο Σχήμα 6.6



Σχ. 6.6: Διαχύση παράλληλων ακτινών. Κάθε σημείο επιφάνειας λόγω του τυχαιού προσανατολισμού του, ανακλά την ακτίνα που προσπίπτει σε αυτό προς διαφορετική κατεύθυνση

Επιφάνειες που διαφέρουν ως προς την τραχύτητά τους (ανάλογα με το υλικό τους) θα παρουσιάζουν αυτό το φαινόμενο διάχυσης με διαφορετική ένταση.

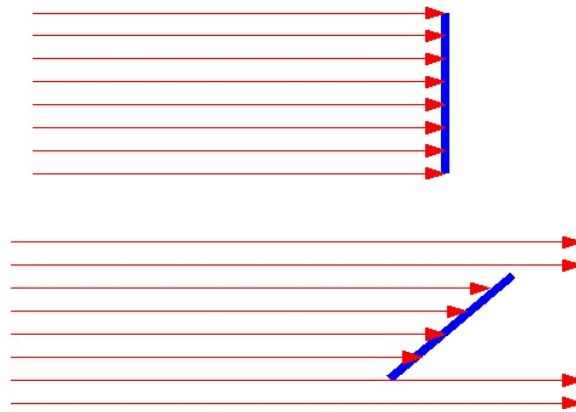
Στο μοντέλο φωτισμού που υλοποιούν οι βιβλιοθήκες γραφικών, θεωρούμε ιδανικές επιφάνειες που τα σημεία τους διαχέουν ομοιόμορφα την προσπίπτουσα ένταση προς όλες τις κατευθύνσεις. Αυτές οι ιδανικά διαχυτικές επιφάνειες ονομάζονται **Λαμπερτιανές** και τα σημεία τους παρουσιάζουν την ίδια φωτεινότητα από οποιαδήποτε οπτική γωνία και αν τα παρατηρήσουμε.

Οι διάχυτες ανακλάσεις εμφανίζονται μόνο στα τμήματα των επιφανειών που είναι ορατά από την πηγή φωτισμού. Επιπλέον, η ένταση που ανακλάται από μια ιδανικά διαχυτική επιφάνεια εξαρτάται από τον προσανατολισμό της σε σχέση με την πηγή φωτισμού. Εάν το κανονικό διάνυσμα της ανακλώσας επιφάνειας είναι παράλληλο με τη διεύθυνση διάδοσης της προσπίπτουσας ακτινοβολίας, τότε η επιφάνεια δέχεται τη μέγιστη ένταση από την πηγή φωτισμού. (Σχ. 6.7). Εάν όμως η διεύθυνση της προσπίπτουσας ακτίνας της πηγής και το κανονικό διάνυσμα σχηματίζουν γωνία ϕ τότε η προσπίπτουσα στην επιφάνεια ένταση φωτός καθορίζεται βάσει της σχέσης

$$I_{inc} = I_d \cdot \cos(\phi)$$

Συνεπώς η ένταση που τελικά διαχέεται από μια Λαμπερτιανή επιφάνεια εξαρτάται επίσης από τη συγκεκριμένη γωνία πρόσπτωσης της ακτίνας. Αυτό σημαίνει ότι μια επιφάνεια στην οποία προσπίπτει μία

ακτίνα πλαγίως θα εμφανίζεται σκοτεινότερη σε σχέση με μια πανομοιότυπη επιφάνεια που δέχεται κάθετη πρόσπτωση. (Σχ. 6.7)

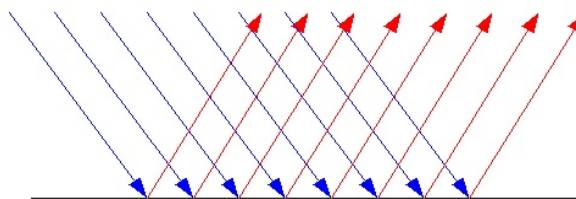


Σχ. 6.7: Η ένταση που προσπίπτει (και συνεπώς διαχέεται) από μια ιδανική διαχυτική επιφάνεια εξαρτάται από τη σχετική θέση της ως προς την πηγή φωτισμού

6.7 Κατοπτρικές ανακλάσεις – Μοντέλο Phong

Πέραν των ανακλάσεων διάχυσης, οι βιβλιοθήκες γραφικών ορίζουν έναν ακόμη τύπο ανακλάσεων που μοντελοποιεί τη συμπεριφορά λείων επιφανειών: τις **κατοπτρικές ανακλάσεις (specular reflections)**. Οι κατοπτρικές ανακλάσεις γίνονται αντιληπτές ως έντονες φωτεινές κηλίδες σε στιλπνές επιφάνειες όπως π.χ. σε λείες μεταλλικές επιφάνειες που εκτίθενται στο ηλιακό φως.

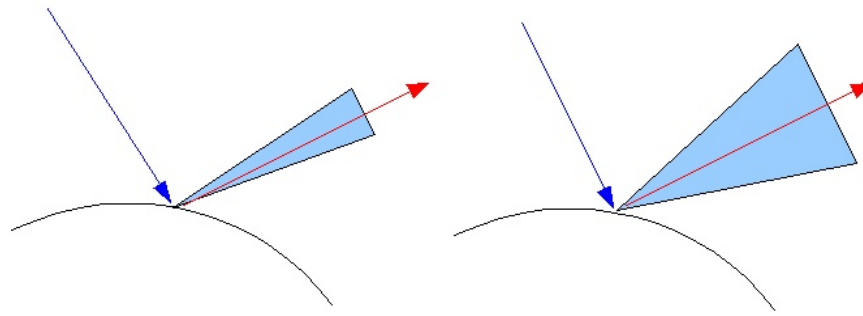
Σε αντίθεση με τις Λαμπερτιανές επιφάνειες, στις απόλυτα λείες επιφάνειες, παράλληλες παράλληλες δέσμες ανακλώνται κατά την ίδια κατεύθυνση, με αποτέλεσμα να συγκεντρώνουν της ανακλώμενη ενέργεια στην ίδια διεύθυνση διάδοσης και να γίνονται αντιληπτές ως περιοχές με ισχυρή ανακλώμενη ένταση (Σχ. 6.8).



Σχ. 6.8. Ιδανική κατοπτρική ανάκλαση (ιδανικός ανακλαστήρας). Όλες οι ανακλώμενες ακτίνες ακολουθούν παράλληλες διαδρομές

Στην πράξη δεν υπάρχουν απολύτως λείες επιφάνειες και οι πρακτικά στιλπνές επιφάνειες χαρακτηρίζονται από έναν βαθμό τραχύτητας. Αυτό έχει ως αποτέλεσμα, παράλληλες ακτίνες που προσπίπτουν πάνω σε μια “λεία” επιφάνεια να εμφανίζουν ελαφρές αποκλίσεις στη διεύθυνσή τους μετά την ανάκλασή τους. Η απόκλισή τους περιορίζεται σε μια κωνική περιοχή που το εύρος της εξαρτάται από την τραχύτητα της επιφανείας. Προφανώς όσο πιο λεία είναι η ανακλώσα επιφάνεια τόσο μικρότερο είναι το

εύρος της κωνικής περιοχής διάχυσης (Σχ. 6.9).



Σχ. 6.9: Εύρος διάχυσης κατοπτρικών ανακλάσεων. Λείες επιφάνειες παρουσιάζουν έντονα κατευθυντικές κατοπτρικές ανακλάσεις και μικρή διάχυση (αριστερά). Τραχιές επιφάνειες τείνουν να διαχέουν περισσότερο την προσπίπτουσα ακτινοβολία και εμφανίζουν λιγότερο κατευθυντικές κατοπτρικές ανακλάσεις (δεξιά).

Ο βαθμός τραχύτητας μιας επιφάνειας, άρα και η συμπεριφορά της ως προς τις κατοπτρικές ανακλάσεις, προσομοιώνεται βάσει του **μοντέλου Phong**. Σύμφωνα με το μοντέλο Phong, η ένταση μιας κατοπτρικά ανακλωμένης ακτίνας I_s εξασθενεί σε σχέση με την προσπίπτουσα ένταση I_{inc} βάσει του παράγοντα γωνιακής εξασθένσης

$$f_{ang}(\phi) = \cos^n(\phi)$$

δηλαδή ισχύει

$$I_s = k_s \cdot I_{inc} \cdot f_{ang}(\phi)$$

όπου k_s ο συντελεστής κατοπτρικής ανάκλασης της επιφάνειας και ϕ η γωνιακή απόκλιση της θέσης παρατήρησης από τον άξονα του κώνου διάχυσης

Ο εκθέτης n εξαρτάται από την **τραχύτητα** του υλικού. Όπως και στην περίπτωση των κατευθυντικών πηγών, η επίδραση του εκθέτη προκαλεί εξασθένιση στην ένταση του κατοπτρικά ανακλώμενου φωτός, όσο ο παρατηρητής αποκλίνει από τον άξονα του κώνου διάχυσης, με τον ίδιο τρόπο που δείχνουν τα γραφήματα διάχυσης στην ενότητα “Κατευθυντικές πηγές”. Μεγάλες τιμές του εκθέτη προσομοιώνουν στιλπνές επιφάνειες, στις οποίες έχουμε συγκέντρωση των ανακλώμενων ακτίνων σε μικρό γωνιακό εύρος. Μικρές τιμές του εκθέτη προσομοιώνουν τραχιές επιφάνειες, στις οποίες το φαινόμενο της διάχυσης είναι πιο έντονο.

6.8 Μοντελοποίηση πηγών φωτισμού στην OpenGL

Στο μοντέλο φωτισμού που υλοποιούν οι βιβλιοθήκες γραφικών, θεωρούμε ότι το φως που εκπέμπεται από μία πηγή μπορεί να διαχωριστεί σε τρεις συνιστώσες: μια συνιστώσα που συνεισφέρει αποκλειστικά

στον περιβάλλοντα φωτισμό της σκηνής (περισσότερα για τον περιβάλλοντα φωτισμό στη συνέχεια), μια συνιστώσα που προκαλεί αποκλειστικά ανακλάσεις διάχυσης και μια συνιστώσα που προκαλεί αποκλειστικά κατοπτρικές ανακλάσεις. Ο διαχωρισμός αυτός μας δίνει ευελιξία ως προς τα φαινόμενα ανακλάσεων που επιθυμούμε να εμφανίζονται στη σκηνή. Έχουμε λ.χ. τη δυνατότητα να εμφανίζουμε μόνο ανακλάσεις διάχυσης και να αποκλείσουμε την εμφάνιση κατοπτρικών ανακλάσεων.

6.8.1 Συνιστώσα περιβάλλοντος φωτισμού

Ο **περιβάλλον φωτισμός (ambient light)** εκτείνεται σε ολόκληρη τη σκηνή, έχει την ίδια ένταση σε όλη την έκταση της σκηνής και επιδρά σε όλες τις επιφάνειες. Ουσιαστικά, με τον περιβάλλοντα φωτισμό προσομοιώνουμε ένα γενικό επίπεδο φωτεινότητας που επιδρά ομοιόμορφα σε όλες τις επιφάνειες της σκηνής. Η ένταση και το χρώμα του περιβάλλοντος φωτισμού ορίζονται δηλώνοντας τις χρωματικές συνιστώσες του. Πχ στο μοντέλο RGB δηλώνεται με τρεις συνιστώσες έντασης

$$I_a = \{I_{ar}, I_{ag}, I_{ab}\}$$

Στην OpenGL η δήλωση γίνεται ως εξής.

glLightfv(GL_LIGHTX, GL_AMBIENT, emittedAmbient);

όπου *emittedAmbient* μητρώο που περιέχει τις συνιστώσες του περιβάλλοντος φωτισμού που εκπέμπει η πηγή *GL_LIGHTX*.

Πχ, με τις εντολές

```
GLfloat light1Ambient[] = {0, 0, 1, 0};  
glLightfv(GL_LIGHT1, GL_AMBIENT, light0Ambient);
```

δηλώνουμε ότι η πηγή *GL_LIGHT1* εκπέμπει συνιστώσα περιβάλλοντος φωτισμού μπλε χρώματος.

6.8.2 Συνιστώσα “διάχυτου φωτισμού”

Προκαλεί αποκλειστικά ανακλάσεις διάχυσης. **Επιδρά μόνο στις επιφάνειες που έχουν οπτική επαφή με την πηγή φωτισμού.** Στην OpenGL ορίζεται με την εντολή:

glLightfv(GL_LIGHTX, GL_DIFFUSE, emittedDiffuse);

όπου *emittedDiffuse* ο πίνακας με τα βάρη της διάχυτης συνιστώσας.

Π.χ. με τις εντολές

```
GLfloat light1Diffuse[] = {0, 1, 0, 0};  
glLightfv(GL_LIGHT1, GL_DIFFUSE, light1Diffuse);
```

ορίζουμε για την πηγή *GL_LIGHT1* μία διάχυτη συνιστώσα πράσινου χρώματος.

6.8.3 Συνιστώσα “κατοπτρικού φωτισμού”:

Συνεισφέρει αποκλειστικά στην εμφάνιση κατοπτρικών ανακλάσεων. Ορίζεται με τη δήλωση των χρωματικών συνιστωσών της.

$$I_s = \{I_{sr}, I_{sg}, I_{sb}\}$$

Η κατοπτρική συνιστώσα μιας πηγής φωτισμού ορίζεται με την ***glLightfv*** ως εξής:

glLightfv(GL_LIGHTX, GL_SPECULAR, emittedSpecular);

όπου *emittedSpecular* μητρώο με τα βάρη της “κατοπτρικής συνιστώσας φωτισμού”.

Π.χ. Με τις εντολές

```
GLfloat light1Specular[] = {1, 0, 0, 0};  
glLightfv(GL_LIGHT1, GL_SPECULAR, light1Specular);
```

ορίζουμε για την πηγή *GL_LIGHT1* μία κατοπτρική συνιστώσα κόκκινου χρώματος.

6.9 Καθολικές παράμετροι φωτισμού

Εκτός από τη ρύθμιση μεμονωμένων πηγών φωτισμού, η OpenGL υποστηρίζει και καθολικές παραμέτρους φωτισμού. Οι καθολικές παράμετροι δεν εντάσσονται σε κάποια από τις πηγές φωτισμού. Για να ρυθμιστούν οι καθολικές παράμετροι φωτισμού, χρησιμοποιείται η εντολή ***glLightModel{if}{n}***:

void glLightModel*(parameterName, parameterValue);

όπου *parameterName* η καθολική παράμετρος που ορίζουμε και *parameterValue* η τιμή ή το μητρώο τιμών

που προσδιορίζουν την παράμετρο *parameterName*.

Μια συχνά ρυθμιζόμενη καθολική παράμετρος είναι ο **καθολικός περιβάλλον φωτισμός**, μια παράμετρος περιβάλλοντος φωτισμού που δεν εντάσσεται σε κάποια από τις πηγές φωτισμού (*GL_LIGHT0*, *GL_LIGHT1*, ...) αλλά ρυθμίζεται ανεξάρτητα. Η καθολική παράμετρος περιβάλλοντος φωτισμού ορίζεται με την χρήση της εντολής ***glLightModelfv*** ως εξής:

```
GLfloat globalAmbient[] = {r, g, b};  
glLightModelfv( GL_LIGHT_MODEL_AMBIENT, globalAmbient );
```

όπου *globalAmbient* οι συνιστώσες του καθολικού περιβάλλοντος φωτισμού.

6.10 Προκαθορισμένες ιδιότητες πηγών φωτισμού

Η μηχανή της OpenGL θεωρεί προκαθορισμένες τιμές για τις ιδιότητες των πηγών φωτισμού, όταν αυτές δεν προσδιορίζονται από το χρήστη. Συγκεκριμένα:

- Για την καθολική παράμετρο περιβάλλοντος φωτισμού οι συνιστώσες RGBA έχουν τις προκαθορισμένες τιμές $\{0.2, 0.2, 0.2\}$ (σκούρο γκριζο).
- Για την πηγή φωτισμού *GL_LIGHT0* ορίζονται οι προκαθορισμένες συνιστώσες περιβάλλοντος φωτισμού $\{0.2, 0.2, 0.2\}$ (σκούρο γκριζο), οι συνιστώσες διάχυτου φωτισμού $\{1, 1, 1\}$ (το εντονότερο λευκό) και οι συνιστώσες κατοπτρικού φωτισμού $\{1, 1, 1\}$ (το εντονότερο λευκό)
- Όλες οι πηγές φωτισμού πλην της *GL_LIGHT0* (*GL_LIGHT1*, *GL_LIGHT_2* κ.ο.κ.) έχουν ως αρχικά καθορισμένες μηδενικές συνιστώσες “περιβάλλοντος”, “διάχυτου” και “κατοπτρικού” φωτισμού.

6.11 Μοντελοποίηση ανακλωσών επιφανειών στην OpenGL

Η εμφάνιση μιας σκηνής δεν εξαρτάται μόνο από τα χαρακτηριστικά των πηγών φωτισμού, αλλά και από τα ανακλαστικά χαρακτηριστικά των επιφανειών. Επομένως, η απόδοση μιας σκηνής καθορίζεται από το συνδυασμό των χαρακτηριστικών των πηγών και των επιφανειών.

Οι ιδιότητες των επιφανειών καθορίζονται με τρόπο παρόμοιο με αυτόν που ορίζονται οι ιδιότητες των πηγών. Ανάλογα με τον τύπο του χαρακτηριστικού, χρησιμοποιείται η εντολή

glMaterialf(GLenum face, GLenum property, GLfloat propertyValue);

για ιδιότητες που προσδιορίζονται με μία αριθμητική τιμή ή

glMaterialfv(GLenum face, GLenum property, GLfloat *propertyValues);

για ιδιότητες που περιγράφονται από ένα σύνολο αριθμητικών τιμών.

Το όρισμα *face* καθορίζει την όψη στην οποία αποδίδεται το εκάστοτε χαρακτηριστικό:

GL_FRONT: Το χαρακτηριστικό αποδίδεται στη μπροστινή όψη των επιφανειών.

GL_BACK: Το χαρακτηριστικό αποδίδεται στην πίσω όψη των επιφανειών.

GL_FRONT_AND_BACK: Το χαρακτηριστικό αποδίδεται και στις δύο όψεις των επιφανειών.

Το όρισμα *propertyValue* ή *propertyValues* είναι η αριθμητική τιμή ή ο δείκτης στο μητρώο των αριθμητικών τιμών που καθορίζουν το χαρακτηριστικό αντίστοιχα .

Σε κάθε χρονική στιγμή ορίζεται ένας μόνο ενεργός συνδυασμός χαρακτηριστικών επιφανείας. Δηλαδή, τα χαρακτηριστικά των επιφανειών λειτουργούν ως μεταβλητές κατάστασης. Προκειμένου λοιπόν να μεταβάλλουμε τις τρέχουσες ανακλαστικές ιδιότητες των επιφανειών επιφάνειας, πρέπει να μεταβάλλουμε τις τρέχουσες ενεργές τιμές.

Στη συνέχεια αναλύουμε τα χαρακτηριστικά των επιφανειών που ορίζονται με τις εντολές ***glMaterial ****.

6.12 Συντελεστές ανάκλασης επιφανειών στην OpenGL

Κάθε επιφάνεια χαρακτηρίζεται από τους συντελεστές ανάκλασής της. Οι συντελεστές ανάκλασης της επιφάνειας είναι αριθμητικές τιμές που καθορίζουν το ποσοστό της φωτεινής έντασης που επιστρέφει η επιφάνεια στη σκηνή. Δεδομένου ότι σε πραγματικές σκηνές η ανακλαστικότητα μιας επιφάνειας διαφέρει ανάλογα με τη συχνότητα (χρώμα) εκπομπής, ορίζουμε έναν συντελεστή ανάκλασης ανά βασικό χρώμα. Με αυτό τον τρόπο ουσιαστικά δηλώνουμε το χρώμα της επιφανείας.

Επιπρόσθετα, το μοντέλο φωτισμού της OpenGL θεωρεί ότι και η ανακλαστικότητα των επιφανειών περιγράφεται με τρεις “κατηγορίες” συντελεστών ανάκλασης: το “συντελεστή ανάκλασης περιβάλλοντος φωτισμού”, το “συντελεστή ανάκλασης διάχυτου φωτισμού” και το “συντελεστή ανάκλασης της κατοπτρικής συνιστώσας”. Κάθε συντελεστής ανάκλασης περιγράφεται από τις χρωματικές συνιστώσες του.

6.12.1 Συντελεστής ανάκλασης περιβάλλοντος φωτισμού

Αν μια επιφάνεια χαρακτηρίζεται από τους συντελεστές ανάκλασης περιβάλλοντος φωτισμού στο χρωματικό μοντέλο RGB:

$$k_a = \{k_{ar}, k_{ag}, k_{ab}\}$$

τότε οι συνιστώσες της έντασης που ανακλώνται από την επιφάνεια λόγω της επίδρασης περιβάλλοντος φωτισμού προκύπτουν από τα γινόμενα

$$I_{a,ref} = \{k_{ar} \cdot I_{ar}, k_{ag} \cdot I_{ag}, k_{ab} \cdot I_{ab}\}$$

Ο συντελεστής ανάκλασης περιβάλλοντος φωτισμού ορίζεται με την εντολή

glMaterialfv (face, GL_ AMBIENT, ambientCoefficients);

όπου *ambientCoefficients* το μητρώο με τις αντίστοιχες χρωματικές συνιστώσες.

Σκηνές που μοντελοποιούνται αποκλειστικά με την επίδραση περιβάλλοντος φωτισμού αποδίδουν ένα χρώμα σε κάθε επιφάνεια με ομοιόμορφο συντελεστή ανάκλασης. Αυτό έχει ως αποτέλεσμα να στερούνται φυσικότητας και αίσθησης βάθους, αφού ο περιβάλλον φωτισμός επιδρά με τον ίδιο τρόπο σε όλες τις επιφάνειες ανεξαρτήτως της θέσης και του προσανατολισμού τους. Γι' αυτό το λόγο, η μοντελοποίηση περιβάλλοντος φωτισμού χρησιμοποιείται σε συνδυασμό με πρόσθετες πηγές φωτισμού.

Παράδειγμα: Μοντελοποίηση περιβάλλοντος φωτισμού

```
#include <glut.h>

void init()
{
    glutInitWindowPosition(50,50);
    glutInitWindowSize(800,600);
    glutCreateWindow("Ambient reflection");
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glClearColor(0,0,0,0);

    glMatrixMode(GL_MODELVIEW);
    glTranslatef(0,0,-40);

    glMatrixMode(GL_PROJECTION);
    glOrtho(-80,80,-60,60,0,50);

    glEnable(GL_LIGHTING);

    //Defining global ambient lighting properties
    GLfloat globalAmbient[]={0.5,0.5,0.5};
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT,globalAmbient);
}
```

```

//Defining "ambient reflection" coefficients for surfaces
GLfloat ambientMat[]={0.4,0.6,0.7};
glMaterialfv(GL_FRONT_AND_BACK,GL_AMBIENT,ambientMat);
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    //Drawing a sphere
    glutSolidSphere(40,80,80);

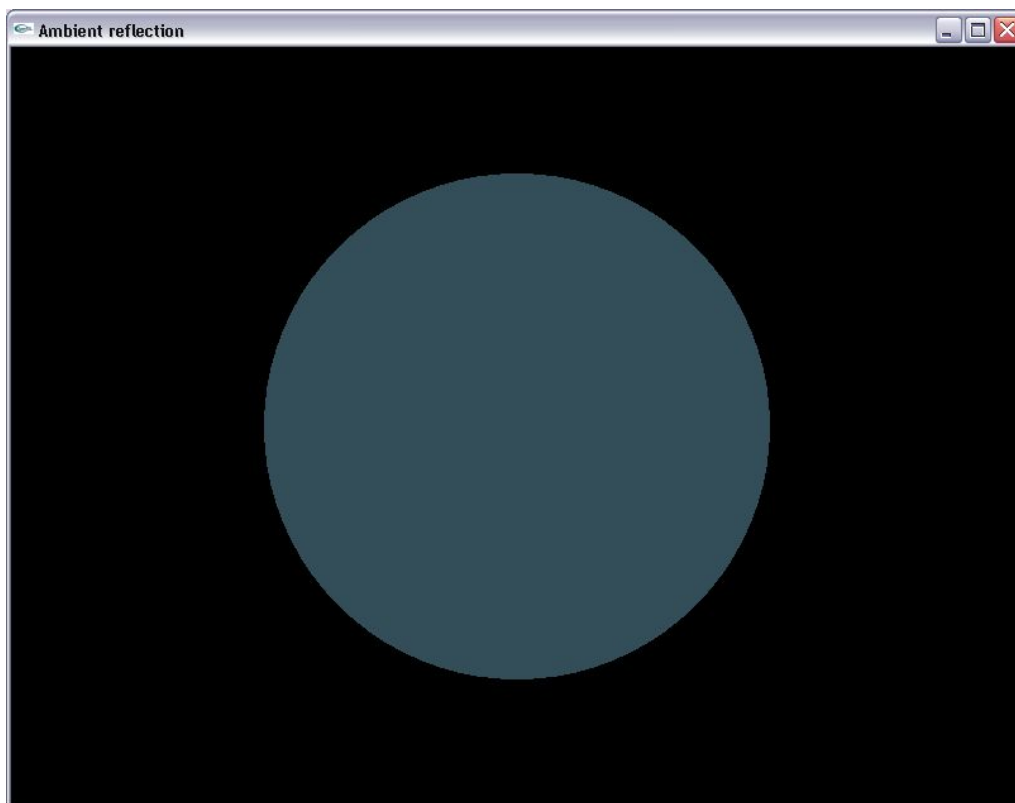
    glFlush();
}

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

    //Initializations
    init();

    glutDisplayFunc(display);
    glutMainLoop();
}

```



6.12.2 Συντελεστές ανάκλασης διάχυτου φωτισμού

Ορίζουν το ποσοστό της διάχυτης συνιστώσας της πηγής που ανακλάται από την επιφάνεια και δηλώνονται με την εντολή:

glMaterialfv(glFace, GL_DIFFUSE, diffuseCoefficients);

όπου *diffuseCoefficients* μητρώο που περιέχει τους συντελεστές ανάκλασης διάχυτου φωτισμού.

Για μια επιφάνεια που χαρακτηρίζεται από τους “συντελεστές ανάκλασης διάχυσης”

$$k_d = \{k_{dr}, k_{dg}, k_{db}\}$$

στην οποία προσπίπτει ένταση

$$I_d = \{I_{dr}, I_{dg}, I_{db}\}$$

και το κανονικό της διάνυσμα της οποίας αποκλίνει κατά γωνία ϕ από το ευθύγραμμο τμήμα που την ενώνει με την πηγή φωτισμού, η ανακλώμενη ένταση (για κάθε χρωματικό κανάλι) προκύπτει από το γινόμενο

$$I_{d,ref} = \{k_{dr}I_{dr} \cos \phi, k_{dg}I_{dg} \cos \phi, k_{db}I_{db} \cos \phi\}$$

Παράδειγμα: Μοντελοποίηση διάχυτης ανάκλασης

```
#include <glut.h>

void init()
{
    glutInitWindowPosition(50,50);
    glutInitWindowSize(800,600);
    glutCreateWindow("Diffuse reflection");
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glClearColor(0,0,0,0);

    glMatrixMode(GL_PROJECTION);
    glOrtho(-80,80,-60,60,0,50);

    glEnable(GL_LIGHTING);

    //Disabling global ambient light component
    GLfloat globalAmbient[]={0,0,0};
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT,globalAmbient);

    //Defining the position of the point light source GL_LIGHT0 at
    //(x,y,z)=(0,0,40)
    GLfloat light0Position[]={0,0,40,1};
    glLightfv(GL_LIGHT0,GL_POSITION,light0Position);

    //Defining "diffuse" lighting properties for GL_LIGHT0
    GLfloat light0Diffuse[]={1,1,1};
    glLightfv(GL_LIGHT0,GL_DIFFUSE,light0Diffuse);
```

```

//Defining "diffuse reflection" coefficients for surfaces
GLfloat diffuseMat[]={0.4,0.6,0.7};
glMaterialfv(GL_FRONT_AND_BACK,GL_DIFFUSE,diffuseMat);

glEnable(GL_LIGHT0);

//Translating all subsequent shapes by x=-40
glMatrixMode(GL_MODELVIEW);
glTranslatef(0,0,-40);
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    //Drawing a sphere
    glutSolidSphere(40,80,80);

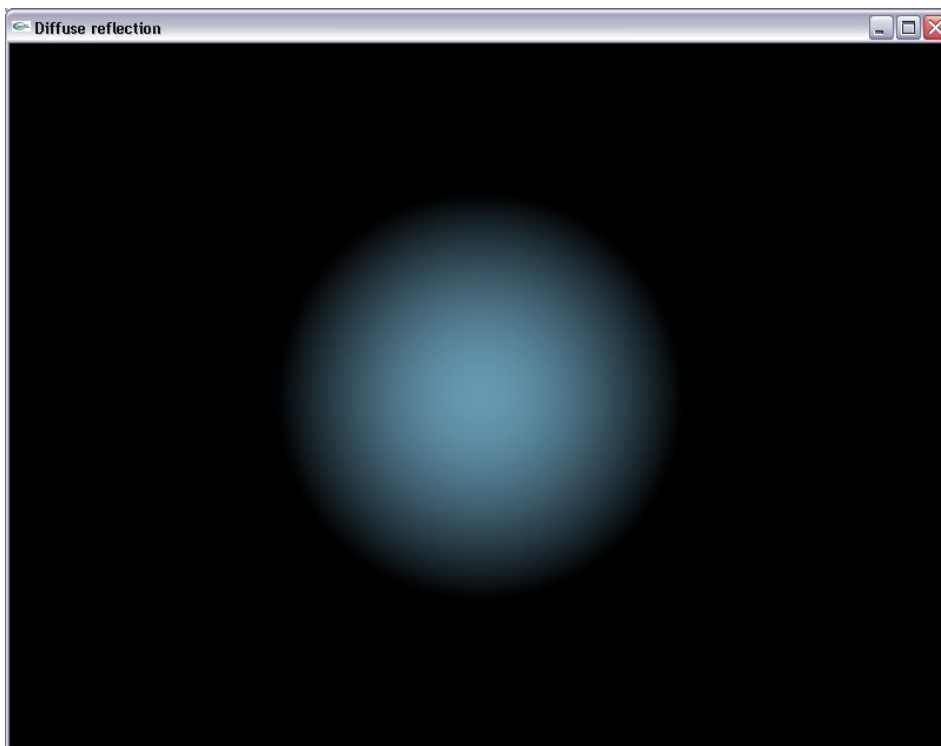
    glFlush();
}

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

    //Initializations
    init();

    glutDisplayFunc(display);
    glutMainLoop();
}

```



6.12.3 Συντελεστές κατοπτρικής ανάκλασης

Οι συντελεστές κατοπτρικής ανάκλασης ορίζουν το ποσοστό της κατοπτρικής συνιστώσας φωτισμού που ανακλά η επιφάνεια. Ορίζονται με την εντολή

glMaterialfv(face, GL_SPECULAR, specularCoefficients);

Επιπλέον, μπορούμε να καθορίσουμε την τραχύτητα της επιφάνειας (εύρος κώνου διάχυσης), δίνοντας τον εκθέτη του γωνιακού παράγοντα εξασθένησης με την εντολή:

glMaterialf(face, GL_SHININESS, n);

όπου n ο εκθέτης γωνιακής εξασθένησης, ο οποίος κυμαίνεται μεταξύ των τιμών 0 και 128. Εάν δε δηλωθεί εκθέτης γωνιακής εξασθένησης από τον προγραμματιστή, η μηχανή της OpenGL θεωρεί ως προκαθορισμένη την τιμή 0, οπότε η ανακλώσα επιφάνεια ανακλά ομοιόμορφα την προσπίπτουσα κατοπτρική συνιστώσα προς όλες τις κατευθύνσεις.

Για μια επιφάνεια με συντελεστές κατοπτρικής ανάκλασης $k_s = \{k_{sr}, k_{sg}, k_{sb}\}$, στην οποία προσπίπτει “κατοπτρική συνιστώσα” πηγής φωτός με ένταση $I_s = \{I_{sr}, I_{sg}, I_{sb}\}$, εάν ο παρατηρητής αποκλίνει κατά γωνία θ από τον άξονα του κώνου διάχυσης, η ανακλώμενη ένταση προς τον παρατηρητή $I_{s,ref}$ είναι:

$$I_{s,ref} = \{k_{sr} I_{sr} \cos^n \theta, k_{sg} I_{sg} \cos^n \theta, k_{sb} I_{sb} \cos^n \theta\}$$

Παράδειγμα: Μοντελοποίηση διάχυτης και κατοπτρικής ανάκλασης

```
#include <glut.h>

void init()
{
    glutInitWindowPosition(50,50);
    glutInitWindowSize(800,600);
    glutCreateWindow("Diffuse reflection");
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glClearColor(0,0,0,0);

    glMatrixMode(GL_PROJECTION);
    glOrtho(-80,80,-60,60,0,50);

    glEnable(GL_LIGHTING);

    //Disabling global ambient light
    GLfloat globalAmbient[]={0,0,0};
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT,globalAmbient);
}
```



```

    //Defining the position of the point light source GL_LIGHT0 at
(x,y,z)=(0,0,40)
    GLfloat light0Position[]={0,0,40,1};
    glLightfv(GL_LIGHT0,GL_POSITION,light0Position);

    //Defining "diffuse" lighting properties for GL_LIGHT0
    GLfloat light0Diffuse[]={1,1,1};
    glLightfv(GL_LIGHT0,GL_DIFFUSE,light0Diffuse);

    //Defining "specular" lighting properties for GL_LIGHT0
    GLfloat light0Specular[]={.5,1,1};
    glLightfv(GL_LIGHT0,GL_SPECULAR,light0Specular);

    //Defining "diffuse reflection" coefficients for surfaces
    GLfloat diffuseMat[]={0.6,0.5,0.1};
    glMaterialfv(GL_FRONT_AND_BACK,GL_DIFFUSE,diffuseMat);

    //Defining "specular reflection" coefficients for surfaces
    GLfloat specularMat[]={.6,.5,.1};
    glMaterialfv(GL_FRONT_AND_BACK,GL_SPECULAR,specularMat);

    //Defining material shininess exponent
    glMaterialf(GL_FRONT_AND_BACK,GL_SHININESS,64);

    glEnable(GL_LIGHT0);

    //Translating all subsequent shapes by x=-40
    glMatrixMode(GL_MODELVIEW);
    glTranslatef(0,0,-40);
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    //Drawing a sphere
    glutSolidSphere(40,80,80);

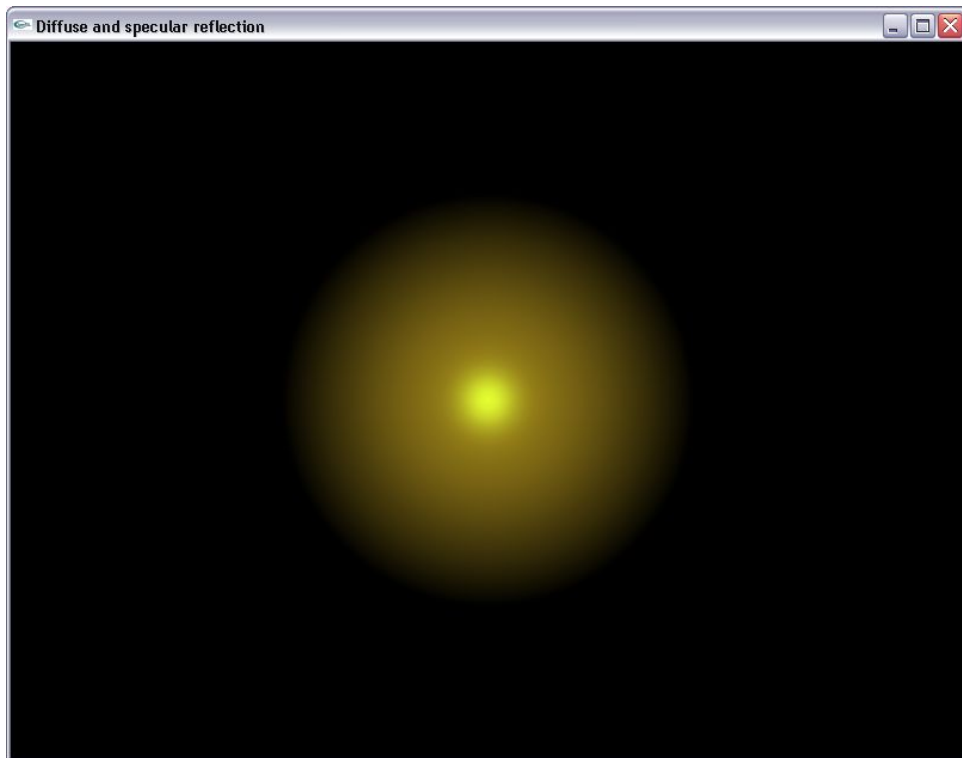
    glFlush();
}

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

    //Initializations
    init();

    glutDisplayFunc(display);
    glutMainLoop();
}

```



Σημείωση:

Ένας κανόνας για τη ρεαλιστική απόδοση μιας ανακλώσας επιφάνειας ορίζει ότι, οι συντελεστές διάχυτης ανάκλασης και οι συντελεστές ανάκλασης περιβάλλοντος φωτισμού είναι οι ίδιοι. Προκειμένου λοιπόν να αποδώσουμε τις ιδιότητες αυτές με μία εντολή, μπορούμε να χρησιμοποιήσουμε την εντολή *glMaterialfv* με το όρισμα *GL_AMBIENT_AND_DIFFUSE* ως εξής:

```
GLfloat ambientAndDiffuse[] = {r, g, b};  
glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE, ambientAndDiffuse)
```

6.13 Αυτόφωτες επιφάνειες

Συχνά, ορισμένες επιφάνειες της σκηνής λειτουργούν ως αυτόφωτα σώματα. Για να αποδώσουμε το φαινόμενο εκπομπής φωτός από μια επιφάνεια αυτή καθεαυτή, ορίζουμε τα βάρη του εκπεμπόμενου φωτός με την εντολή

glMaterialfv(face, GL_EMISSION, emissionComponents);

όπου *emissionComponents* μητρώο που περιέχει τις χρωματικές συνιστώσες του εκπεμπόμενου φωτός.

Π.χ. χρησιμοποιώντας τις εντολές

```
GLfloat emittedSurf[] = {1, 1, 0, 0};
glMaterialfv(GL_FRONT_AND_BACK, GL_EMISSION, emittedSurf);
```

ορίζουμε ότι οι εμπρός και πίσω όψεις των επιφανειών εκπέμπουν κίτρινο χρώμα.

Επισημαίνουμε ότι **το φως που εκπέμπεται από μια αυτόφωτη επιφάνεια δεν επιδρά στις υπόλοιπες ανακλώσες επιφάνειες της σκηνής και γενικά δε λαμβάνεται υπόψη από το μοντέλο φωτισμού της OpenGL ως πηγή φωτός**. Επομένως, εάν θέλουμε να μοντελοποιήσουμε την επίδραση μιας αυτόφωτης επιφάνειας ως πηγή φωτισμού στις υπόλοιπες επιφάνειες, απαιτείται επιπλέον η τοποθέτηση πηγής φωτισμού στη θέση της αυτόφωτης επιφάνειας.

6.14 Προκαθορισμένες ιδιότητες ανακλώσων επιφανειών

Οι συντελεστές ανάκλασης των επιφανειών, ως μεταβλητές κατάστασης, έχουν αρχικά καθορισμένες τιμές. Συγκεκριμένα:

- Οι προκαθορισμένοι συντελεστές ανάκλασης περιβάλλοντος φωτισμού είναι $\{0.2, 0.2, 0.2\}$.
- Οι προκαθορισμένοι συντελεστές διάχυτης ανάκλασης είναι: $\{0.8, 0.8, 0.8\}$.
- Οι προκαθορισμένοι συντελεστές κατοπτρικής ανάκλασης είναι $\{1, 1, 1\}$.
- Οι ανακλώσες επιφάνειες δεν είναι αυτόφωτες (εκπεμπόμενο φως $\{0, 0, 0\}$).

6.15 Υπέρθωση πηγών φωτισμού

Όταν για μια επιφάνεια ορίσουμε περισσότερους από έναν συντελεστές ανάκλασης (και επιπλέον έχουμε δηλώσει τις αντίστοιχες συνιστώσες σε μία πηγή φωτισμού), τότε, η ανακλώμενη ένταση της επιφάνειας προκύπτει από την υπέρθεση των συνιστωσών φωτισμού. Δηλαδή

$$I_{ref} = I_{a,ref} + I_{d,ref} + I_{s,ref} = k_a \cdot I_a + k_d \cdot I_d \cdot \cos \phi + k_s \cdot I_s \cdot \cos^n \theta$$

όπου ϕ η γωνία με την οποία προσπίπτει η “διάχυτη συνιστώσα” φωτισμού στην επιφάνεια και θ η γωνιακή απόκλιση του παρατηρητή από τον άξονα του κώνου διάχυσης των κατοπτρικά ανακλωμένων ακτίνων.

Επιπλέον, εάν ορίζονται πολλαπλές πηγές φωτισμού με συνιστώσες έντασης I_{ai} (περιβάλλουσα

συνιστώσα), I_{di} (διάχυτη συνιστώσα), I_{si} (κατοπτρική συνιστώσα), η ανακλώμενη ένταση σε κάθε σημείο μιας επιφάνειας προκύπτει από τη συνδυασμένη επίδραση των πηγών αυτών, δηλαδή

$$I_{ref} = k_a \cdot \sum_i I_{ai} + k_d \cdot \sum_i (I_{di} \cdot \cos \phi_i) + k_s \cdot \sum_i (I_{si} \cdot \cos^n \theta_i)$$

όπου ϕ_i η γωνία που σχηματίζει η ακτίνα της διάχυτης συνιστώσας της πηγής i με το κανονικό διάνυσμα της επιφάνειας και θ_i η απόκλιση του παρατηρητή από τον άξονα του κώνου διάχυσης που παράγεται υπό την επίδραση της πηγής i .

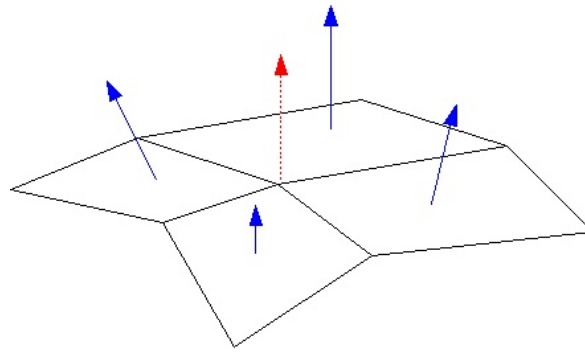
6.16 Σκίαση Gouraud

Όπως αναφέραμε στην ενότητα “Κανονικό διάνυσμα επιφάνειας”, κάθε στοιχειώδης πολυγωνική επιφάνεια χαρακτηρίζεται από το κανονικό της διάνυσμα, το οποίο χρησιμοποιείται για υπολογισμούς από το μοντέλο φωτισμού. Ωστόσο, μια πολυγωνική επιφάνεια δεν αποτελείται από ένα, αλλά από πολλά εσωτερικά σημεία. Δεδομένου ότι για τον υπολογισμό της φωτεινότητας έχουμε μόνο ένα διαθέσιμο κανονικό διάνυσμα ανά πολυγωνική επιφάνεια, το μοντέλο φωτισμού υπολογίζει μία τιμή φωτισμού για όλα τα pixels της πολυγωνικής επιφάνειας. Ωστόσο, η αναπαράσταση αυτή έχει μειωμένη λεπτομέρεια και δεν αναπαριστά ρεαλιστικά τη φωτιζόμενη επιφάνεια.

Μια πιο ρεαλιστική απόδοση της έντασης στα σημεία των επιφανειών αποδίδεται με τη μέθοδο σκίασης Gouraud. Στην τεχνική Gouraud, υπολογίζουμε ένα κανονικό διάνυσμα για κάθε μία κορυφή του πολυγώνου. Στην περίπτωση που δεν ορίζουμε τιμές για τα κανονικά διανύσματα των κορυφών, το κανονικοποιημένο (με μέτρο ίσο με 1) κανονικό διάνυσμα \vec{N} κάθε κορυφής προκύπτει από το μέσο όρο των κανονικών διανυσμάτων των γειτονικών της πολυγώνων (Σχήμα 6.10), δηλαδή:

$$\vec{N} = \frac{\sum_i \vec{N}_i}{\left| \sum_i \vec{N}_i \right|}$$

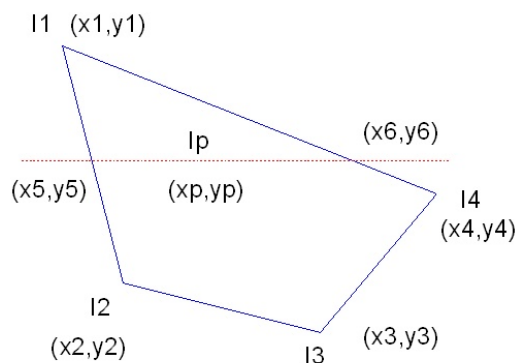
όπου \vec{N}_i το κανονικό διάνυσμα κάθε γειτονικής πολυγωνικής επιφάνειας.



Σχ. 6.10: Κανονικό διάνυσμα κορυφής. Όταν δεν αναθέτουμε τιμές, τα κανονικά διανύσματα των κορυφών προσεγγίζονται από το μέσο όρο των κανονικών διανυσμάτων των γειτονικών πολυγώνων.

Έχοντας βρει τα κανονικά διανύσματα των κορυφών υπολογίζουμε την ανακλώμενη ένταση στη θέση κάθε μίας κορυφής βάσει των σχέσεων που ορίζει το φωτορεαλιστικό μοντέλο. Κατόπιν, με βάση τις εντάσεις των κορυφών, υπολογίζουμε τις εντάσεις στα όρια των πολυγώνων εφαρμόζοντας την τεχνική της **γραμμικής παρεμβολής**. Με την τεχνική αυτή, προσεγγίζουμε την ένταση στα σημεία των πλευρών του πολυγώνων, ανάλογα με τις αποστάσεις τους από τα άκρα των ευθυγράμμων τμημάτων.

Πχ. στο διδιάστατο πολύγωνο του Σχ. 6.11 θεωρούμε μια τυχαία οριζόντια γραμμή σάρωσης με άκρα της τα σημεία (x_5, y_5) και (x_6, y_6) . Αφού υπολογιστούν οι εντάσεις των κορυφών του, προσεγγίζουμε την ένταση I_5 του αυθαίρετου σημείου (x_5, y_5) ως ένα συνδυασμό των εντάσεων I_1, I_2 και την ένταση I_6 του αυθαίρετου σημείου (x_6, y_6) ως ένα συνδυασμό των εντάσεων I_1 και I_4 .



Σχ. 6.11: Γραμμική παρεμβολή εντάσεων σε πολυγωνική επιφάνεια

Η ένταση I_5 του σημείου (x_5, y_5) καθορίζεται από τις αποστάσεις του από τις κορυφές βάσει της σχέσης γραμμικής παρεμβολής

$$I_5 = (1 - a) \cdot I_1 + a \cdot I_2$$

Η τιμή a καθορίζεται από την απόσταση του σημείου (x_5, y_5) από τις δύο κορυφές της πλευράς:

$$a = \frac{d_{1,5}}{d_{1,2}} = \frac{\sqrt{(x_5 - x_1)^2 + (y_5 - y_1)^2}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

Προφανώς ισχύει

$$0 \leq a \leq 1$$

Ομοίως για το σημείο (x_6, y_6) η έντασή του προκύπτει από τη σχέση

$$I_6 = (1 - a) \cdot I_1 + a \cdot I_4$$

με

$$a = \frac{d_{1,6}}{d_{1,4}} = \frac{\sqrt{(x_6 - x_1)^2 + (y_6 - y_1)^2}}{\sqrt{(x_4 - x_1)^2 + (y_4 - y_1)^2}}$$

Έχοντας υπολογίσει τις οριακές τιμές I_5 και I_6 μιας ακτίνας σάρωσης, υπολογίζουμε τις εντάσεις όλων των εσωτερικών σημείων της γραμμής σάρωσης με τον ίδιο τρόπο.

Η σκίαση Gouraud είναι ενεργοποιημένη εξ' αρχής στη μηχανή καταστάσεων της OpenGL. Ωστόσο, ο προγραμματιστής έχει τη δυνατότητα να επιλέξει την ενιαία απόδοση φωτεινότητας σε κάθε στοιχειώδη επιφάνεια με την εντολή *glShadeModel*. Δίνοντας την εντολή:

glShadeModel(GL_FLAT);

επιλέγουμε ενιαία σκίαση σε όλη την έκταση κάθε στοιχειώδους πολυγωνικής επιφανείας. Η χρήση της σκίασης Gouraud επαναφέρεται με την εντολή:

glShadeModel(GL_SMOOTH);

Παράδειγμα: Δήλωση κανονικού διανύσματος ανά κορυφή τριγώνου και εφαρμογή σκίασης Gouraud

```
#include <glut.h>

void init()
{
    glutInitWindowPosition(50, 50);
    glutInitWindowSize(800, 600);
    glutCreateWindow("Using normal vectors");
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glClearColor(0, 0, 0, 0);

    glMatrixMode(GL_PROJECTION);
    glOrtho(-40, 40, -30, 30, 1, 50);
```

```

glEnable(GL_LIGHTING);

//Disabling global ambient light
GLfloat globalAmbient[]={0,0,0};
glLightModelfv(GL_LIGHT_MODEL_AMBIENT,globalAmbient);

//Defining the position of the point light source GL_LIGHT0 at
//(x,y,z)=(0,0,40)
GLfloat light0Position[]={0,0,0,1};
glLightfv(GL_LIGHT0,GL_POSITION,light0Position);

GLfloat light0Ambient[]={0,0,0};
glLightfv(GL_LIGHT0,GL_AMBIENT,light0Ambient);

//Defining "diffuse" lighting properties for GL_LIGHT0
GLfloat light0Diffuse[]={1,1,1};
glLightfv(GL_LIGHT0,GL_DIFFUSE,light0Diffuse);

//Defining "diffuse" lighting properties for GL_LIGHT0
GLfloat light0Specular[]={0,0,0};
glLightfv(GL_LIGHT0,GL_SPECULAR,light0Specular);

//Defining "diffuse reflection" coefficients for surfaces
GLfloat diffuseMat[]={0.4,0.6,0.7};
glMaterialfv(GL_FRONT_AND_BACK,GL_DIFFUSE,diffuseMat);

glEnable(GL_LIGHT0);
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    //Drawing a triangle - Each is assigned a different normal vector

    glBegin(GL_TRIANGLES);

    //Normal vector of the first point nv1=(-1,0,1)
    GLfloat nv1[]={-1,0,1};
    glNormal3fv(nv1);
    glVertex3f(-10,0,-10);

    //Normal vector of the first point nv2=(0,1,2)
    GLfloat nv2[]={0,1,2};
    glNormal3fv(nv2);
    glVertex3f(10,-10,-10);

    //Normal vector of the first point nv3=(0,0,1)
    GLfloat nv3[]={0,0,1};
    glNormal3fv(nv3);
    glVertex3f(10,10,-10);
    glEnd();

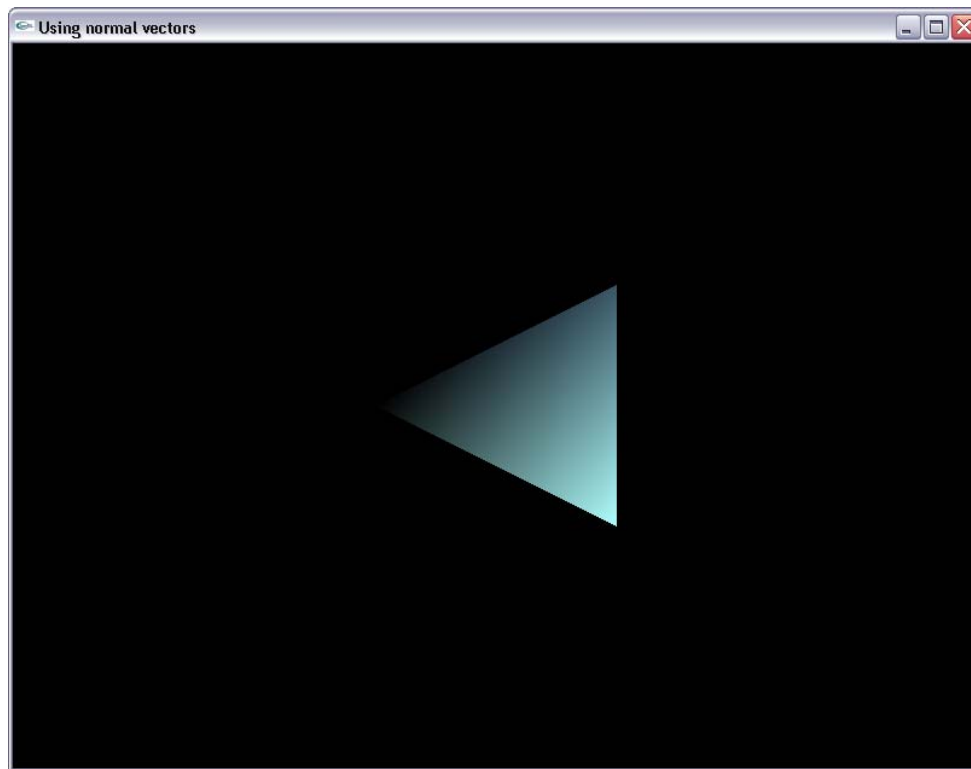
    glFlush();
}

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

    //Initializations
    init();
}

```

```
glutDisplayFunc(display);  
glutMainLoop();  
}
```



6.17 Ατμοσφαιρικά εφέ

Σε ορισμένες σκηνές (ιδιαίτερα στις περιπτώσεις σχεδίασης τοπίων) μας ενδιαφέρει να μοντελοποιήσουμε την παρουσία ομιχλώδους ατμόσφαιρας. Η μοντελοποίηση αυτή εκτελείται με την εξασθένηση της έντασης της ανακλώσας επιφάνειας που αντιλαμβάνεται ο παρατηρητής όσο απομακρύνεται από αυτήν. Έτσι, εάν δύο πανομοιότυπες επιφάνειες βρίσκονται σε διαφορετικές αποστάσεις η πιο απομακρυσμένη επιφάνεια θα παρουσιάζεται θολότερη απ' ότι η πλησιέστερη.

Για να αποδοθεί η αίσθηση της ομίχλης μπορούμε να θεωρήσουμε ότι μεταξύ της ανακλώσας επιφάνειας και του παρατηρητή παρεμβάλλεται ένα εμπόδιο (ατμόσφαιρα). Η διαφάνεια του ατμοσφαιρικού εμποδίου καθορίζεται από το πάχος που μεσολαβεί μεταξύ του παρατηρητή και της ανακλώσας επιφάνειας. Από μαθηματικής σκοπιάς, η ένταση μιας ανακλώσας επιφάνειας αποσβέννεται όσο ο παρατηρητής απομακρύνεται από αυτή και η ένταση της παρατηρούμενης ομίχλης αυξάνεται εφόσον το πάχος του ατμοσφαιρικού εμποδίου που παρεμβάλλεται αυξάνεται.

Για τη μοντελοποίηση φαινομένου ομίχλης συνήθως χρησιμοποιείται το μοντέλο εκθετικής απόσβεσης της ανακλώμενης έντασης. Δηλαδή, εάν θεωρήσουμε μια ανακλώσα επιφάνεια η οποία ανακλά προς τον παρατηρητή φως έντασης I_0 και η οποία βρίσκεται σε απόσταση d από το επίπεδο παρατήρησης, τότε η

ένταση $I_{fog}(d)$ που αντιλαμβάνεται ο παρατηρητής είναι

$$I_{fog}(d) = I_0 \cdot f_{fog}(d)$$

Όπου $f_{fog}(d)$ ο παράγοντα εξασθένησης (**συντελεστή διαφάνειας**):

$$f_{fog}(d) = e^{-pd}$$

Η τιμή p ονομάζεται **συντελεστής απόσβεσης**. Καθορίζει την πυκνότητα της ομίχλης και επιλέγεται από τον προγραμματιστή. Μεγάλες τιμές του συντελεστή απόσβεσης ενισχύουν το φαινόμενο ομίχλης.

Πέραν του μοντέλου εκθετικής απόσβεσης, χρησιμοποιούνται και τα μοντέλα τετραγωνικής εκθετικής απόσβεσης και γραμμικής απόσβεσης:

$$f_{fog}(d) = e^{-(pd)^2}$$

$$f_{fog}(d) = 1 - pd$$

Όταν προσομοιώνουμε ομιχλώδεις σκηνές, αποδίδουμε επίσης και το χρώμα της ομιχλώδους ατμόσφαιρας. Για να επιτύχουμε αυτό το συνδυασμό ομίχλης και τοπίου, αναμειγνύουμε το χρώμα της ομιχλώδους ατμόσφαιρας με το χρώμα των ανακλωσών επιφανειών. Δεδομένου ότι ο παράγοντας απόσβεσης $f_{fog}(d)$ καθορίζει το συντελεστή διαφάνειας της ατμόσφαιρας, ο παρατηρητής παρατηρεί σε κάθε σημείο του επιπέδου προβολής τις συνδυασμένες έντασεις:

$$I = f_{fog}(d) \cdot I_{obj} + [1 - f_{fog}(d)] \cdot I_{fog}$$

Δεδομένου ότι ο παράγοντας διαφάνειας της ομίχλης $f_{fog}(d)$ εξαρτάται από την απόσταση παρατηρητή-ανακλώσας επιφάνειας, σε κάθε σημείο του επιπέδου προβολής η διαφάνεια της ομιχλώδους ατμόσφαιρας παρουσιάζει και διαφορετική τιμή. Όσο μια επιφάνεια απομακρύνεται από τον παρατηρητή, τόσο μειώνεται η διαφάνεια της ομίχλης.

Αρχικά, η ενεργοποίηση του εφέ ομίχλης εκτελείται δίνοντας το όρισμα `GL_FOG` στην εντολή `glEnable`:

`glEnable(GL_FOG);`

Κατόπιν, η ρύθμιση των παραμέτρων ομίχλης επιτελείται με την εντολή **glFog***:

void glFogf{f|v}{n}(TYPE parameterName, TYPE parameterValue);

Για να προσδιοριστεί το εφφέ ομίχλης καθορίζουμε τρεις παραμέτρους: το χρώμα, την πυκνότητα και το μοντέλο εξασθένησής της.

α) Χρώμα

Το χρώμα που αποδίδεται στην ομίχλη ρυθμίζεται προσδιορίζοντας την παράμετρο *GL_FOG_COLOR*:

glFogfv(GL_FOG_COLOR, fogColorComponents);

όπου *fogColorComponents* το μητρώο με τις χρωματικές συνιστώσες.

Π.χ. με τις εντολές

```
GLfloat *fogColorArray={0.5,0.5,0.5};  
glFogfv(GL_FOG_COLOR, fogColorArray);
```

αποδίδουμε στην ομίχλη γκρίζο χρώμα.

Εάν δεν αποδώσουμε χρώμα στην ομίχλη, η προκαθορισμένη του τιμή είναι {0,0,0} (μαύρο).

β) Πυκνότητα

Η πυκνότητα *density* της ομίχλης ρυθμίζεται με την απόδοση τιμής στην παράμετρο *GL_FOG_DENSITY*

glFogf(GL_FOG_DENSITY, density);

γ) Μοντέλο εξασθένησης

Η επιλογή του μοντέλου εξασθένησης γίνεται προσδιορίζοντας την παράμετρο *GL_FOG_MODE*:

glFogi(GL_FOG_MODE, mode);

Η παράμετρος *mode* παίρνει τις εξής τιμές

GL_EXP: για την επιλογή του μοντέλου εκθετικής εξασθένησης

GL_EXP2: για την επιλογή του μοντέλου τετραγωνικής εκθετικής εξασθένησης

GL_LINEAR: για την επιλογή του μοντέλου γραμμικής εξασθένησης

Το αρχικά επιλεγμένο μοντέλο είναι το μοντέλο εκθετικής εξασθένησης (*GL_EXP*).

Πέραν της απόδοσης ομιχλωδών τοπίων, το φαινόμενο ομίχλης είναι χρήσιμο για τη σαφή απόδοση σκηνών που περιέχουν κλειστές επιφάνειες σε μορφή πλέγματος. Σε σκηνές στις οποίες αποδίδουμε το πλέγμα τρισδιάστατων κλειστών επιφανειών (σφαιρών, κυλίνδρων κλπ.), κατά την προβολή στο επίπεδο παρατήρησης, προκαλείται σύγχυση ως προς το ποια πλευρά βρίσκεται πλησιέστερα στον παρατηρητή. Στην περίπτωση αυτή, εάν εφαρμόσουμε το φαινόμενο ομίχλης, εξασθενούμε την ένταση με την οποία παρατηρούνται οι απομακρυσμένες ακμές του πλέγματος και αποδίδουμε την αίσθηση του βάθους (**depth cueing**).

Παράδειγμα: Μοντελοποίηση ομίχλης

```
#include <glut.h>

void init()
{
    glutInitWindowPosition(50,50);
    glutInitWindowSize(800,600);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutCreateWindow("Fog modelling");

    glMatrixMode(GL_PROJECTION);
    glOrtho(-80,80,-60,60,0,100);

    glClearColor(0,0,0,1);

    //Enabling hidden-surface removal
    glEnable(GL_DEPTH_TEST);

    //Enabling shading model
    glEnable(GL_LIGHTING);

    //Specifying a distant light source with light rays travelling along the z
axis
    GLfloat light0Position[]={0,0,1,0};
    glLightfv(GL_LIGHT0,GL_POSITION,light0Position);

    //Specifying material ambient and diffuse reflection properties
    GLfloat reflectionCoefficients[]={0.7,0.7,0.2};
    glMaterialfv(GL_FRONT,GL_AMBIENT_AND_DIFFUSE,reflectionCoefficients);

    glEnable(GL_LIGHT0);

    //Enabling atmospheric effects
    glEnable(GL_FOG);
}
```

```

//Setting fog color: (0.8, 0.8, 0.8)
GLfloat fogColor[]={0.8,0.8,0.8};
glFogfv(GL_FOG_COLOR,fogColor);

//Setting fog density to 0.015. An exponential attenuation model is
assumed.
glFogf(GL_FOG_DENSITY,0.015);
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);

    glMatrixMode(GL_MODELVIEW);

    //Drawing a sphere centered at position (-30,-0,-25)
    glLoadIdentity();
    glTranslatef(-45,0,-25);
    glutSolidSphere(20,80,80);

    //Drawing a sphere centered at a further position (0,0,-40)
    glLoadIdentity();
    glTranslatef(-15,0,-40);
    glutSolidSphere(20,80,80);

    //Drawing a sphere centered at position (30,0,-60)
    glLoadIdentity();
    glTranslatef(15,0,-60);
    glutSolidSphere(20,80,80);

    //Drawing a sphere centered at position (30,0,-80)
    glLoadIdentity();
    glTranslatef(45,0,-80);
    glutSolidSphere(20,80,80);

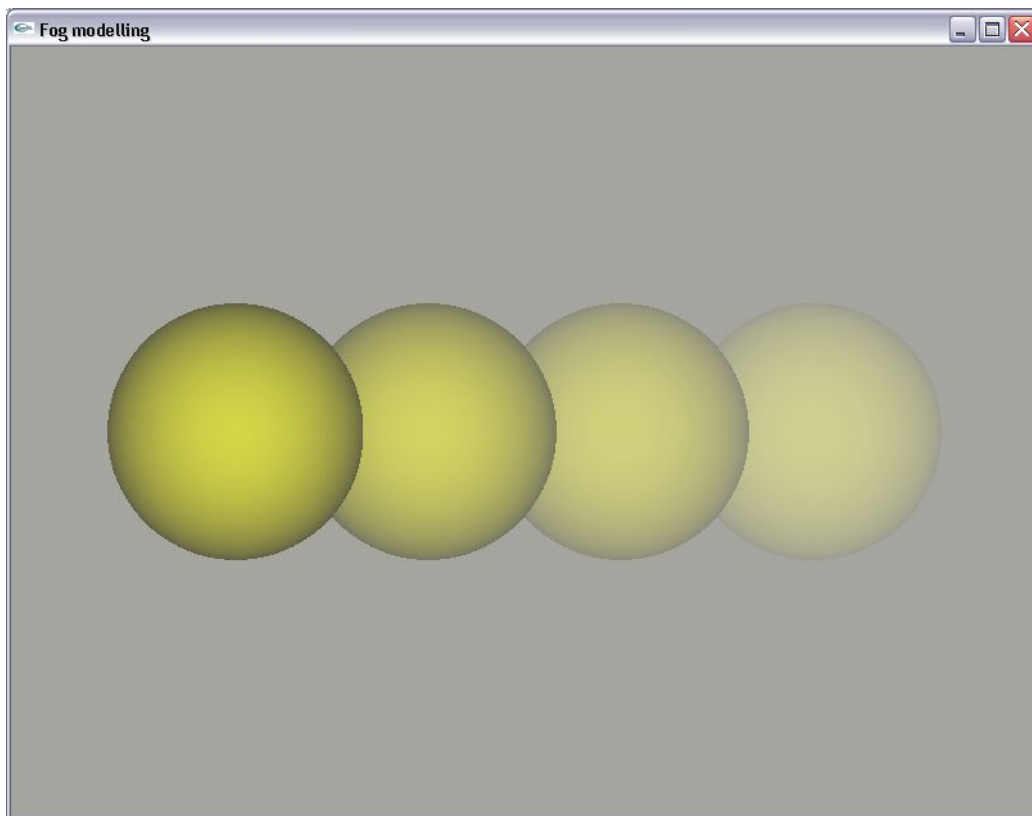
    //Drawing a "background plane" (z=-99)
    glLoadIdentity();
    glBegin(GL_QUADS);
    glVertex3f(-80,-60,-99);
    glVertex3f(80,-60,-99);
    glVertex3f(80,60,-99);
    glVertex3f(-80,60,-99);
    glEnd();

    glFlush();
}
int main(int argc, char **argv)
{
    glutInit(&argc,argv);

    init();

    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```



Παράδειγμα: Απόδοση βάθους σε πλέγμα τρισδιάστατου κυλινδρικού σχήματος

```
#include <glut.h>

GLUquadric *cylinder;

void init()
{
    glutInitWindowPosition(50,50);
    glutInitWindowSize(800,600);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutCreateWindow("Depth Cueing");

    glMatrixMode(GL_PROJECTION);
    glOrtho(-64,64,-48,48,0,50);

    glColor3f(0,1,0);
    glClearColor(0,0,0,0);

    //Setting line width to 3 pixels
    glLineWidth(3);

    //Enabling hidden surface removal
    glEnable(GL_DEPTH_TEST);

    //Enabling fog effects
    glEnable(GL_FOG);

    //Setting the fog color (0, 0.3, 0)
    GLfloat fogColor[]={0.0,0.3,0};
    glFogfv(GL_FOG_COLOR,fogColor);
    glFogf(GL_FOG_DENSITY,0.05);
}
```

```

//Initializing the GLUquadric object cylinder
cylinder=gluNewQuadric();

//All declared shapes will be rotated by 50 degrees by the x axis
//and translated by -30 along the z-axis
glMatrixMode(GL_MODELVIEW);
glTranslatef(0,0,-30);
glRotatef(50,1,0,0);
}
void display()
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);

    //Drawing a cylinder wireframe
    gluQuadricDrawStyle(cylinder, GLU_LINE);
    gluCylinder(cylinder, 20, 20, 20, 20, 5);

    glFlush();
}
int main(int argc, char **argv)
{
    glutInit(&argc, argv);

    init();
    glutDisplayFunc(display);
    glutMainLoop();

    return 0;
}

```

