Αλγόριθμοι Παρακολούθησης Ακτίνας (Ray tracing)

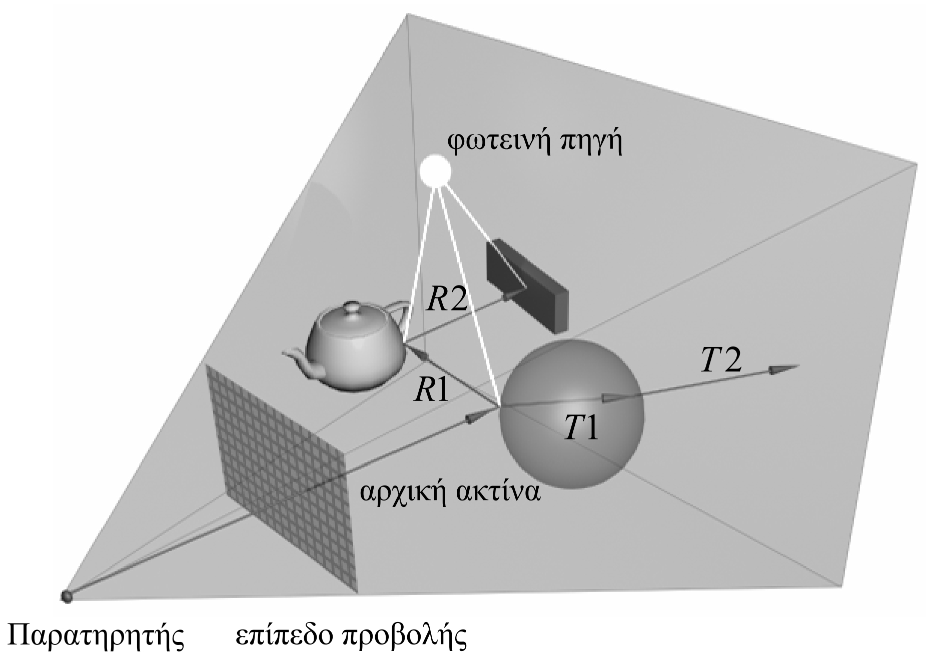
- Τα μοντέλα τοπικού φωτισμού (π.χ. Phong) δεν ασχολούνται με τον έμμεσο φωτισμό των αντικειμένων.
- Τα μοντέλα γενικού φωτισμού λαμβάνουν υπ’ όψη τους και τον έμμεσο φωτισμό:
  - Παρακολούθηση Ακτίνας.
  - Radiosity.
- Αλγόριθμοι παρακολούθηση ακτίνας: βασίζονται στη φυσική διαδικασία φωτισμού των αντικειμένων
  - Ακτίνα ξεκινά από φωτεινή πηγή και μετά από ανακλάσεις και διαθλάσεις με τα αντικείμενα του χώρου καταλήγει στον παρατηρητή.

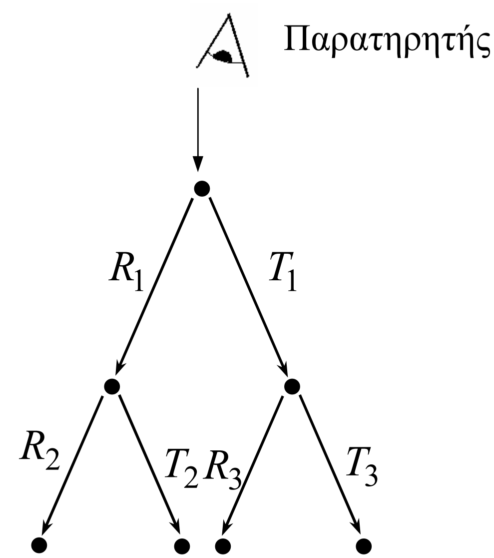
Αναδρομική Παρακολούθηση Ακτίνας

- Η παρακολούθηση ακτίνων από τη φωτεινή πηγή είναι υπολογιστικά πολύ δύσκολη:
  - Απειρες ακτίνες.
  - Λίγες από αυτές καταλήγουν στον παρατηρητή.
- Αναδρομική παρακολούθηση ακτίνας: ακτίνες ξεκινούν, ανάποδα, από τον παρατηρητή και ακολουθούνται μέσα στη σκηνή
  - Συνήθως 1 (ή n για antialiasing) ακτίνα(ες) για κάθε pixel του επιπέδου προβολής.
  - Αν δεν συναντά κανένα σώμα παίρνει χρώμα φόντου.
  - Αλλιώς βρίσκουμε σημείο τομής με το πρώτο αντικείμενο και η ακτίνα ακολουθείται (αναδρομικά) από αυτό το σημείο.
  - Οι ακτίνες θεωρούνται στοιχειώδους πάχους.

Αναδρομική Παρακολούθηση Ακτίνας

- Οταν η ακτίνα συναντά κάποιο αντικείμενο αυτή αναλύεται σε 2 νέες ακτίνες:
  - Την ανακλώμενη R (reflected).
  - Την διαθλώμενη T (transmitted / refracted).
- Κάθε μία από αυτές παρακολουθείται αναδρομικά και έτσι σχηματίζεται ένα δένδρο ακτίνων.





Αναδρομική Παρακολούθηση Ακτίνας

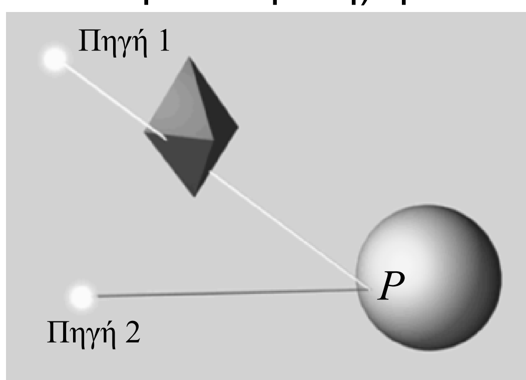
- Οι ακτίνες ανάκλασης και διάθλασης αποτελούν τους "γενικούς" όρους φωτισμού, ενώ υπάρχει και μία "τοπική" συνιστώσα I_L λόγω απευθείας φωτισμού από την πηγή:

$$I(\overline{\Sigma}) = I_L + k_r I_R + k_t I_T$$

  - Οι I_R και I_T υπολογίζονται αναδρομικά σαν φωτεινές τιμές των σημείων $\overline{\Sigma}_R$ και $\overline{\Sigma}_T$ που χτυπούν οι ακτίνες R και T κ.ο.κ.

$$I(\overline{\Sigma}) = I_L + k_r I(\overline{\Sigma}_R) + k_t I(\overline{\Sigma}_T)$$

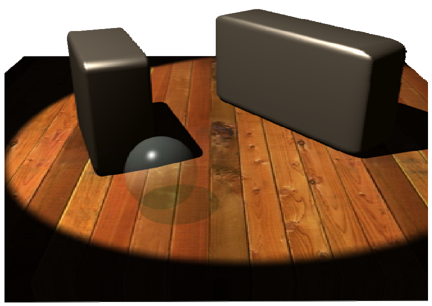
- Για την τοπική συνιστώσα, πριν εφαρμόσουμε το μοντέλο τοπικού φωτισμού, πρέπει να βεβαιωθούμε ότι υπάρχει οπτική επαφή μεταξύ φωτεινής πηγής και $\overline{\Sigma}$:
  - Στέλνουμε άλλη μία ακτίνα από το $\overline{\Sigma}$ προς την φωτεινή πηγή (ακτίνα σκίασης, shadow feeler).
  - Αν στην πορεία της συναντά άλλα σώματα το $\overline{\Sigma}$ σκιάζεται οπότε δεν υπολογίζεται (ή μειώνεται για διαφανή) η τοπική συνιστώσα.



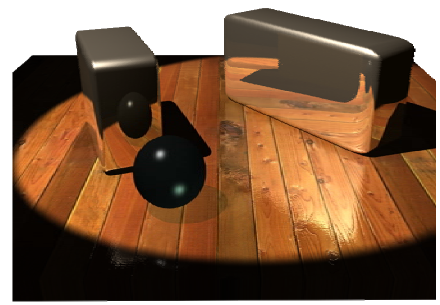
  - Για πολλαπλές φωτεινές πηγές χρειαζόμαστε ισάριθμες ακτίνες σκίασης.

Αναδρομική Παρακολούθηση Ακτίνας

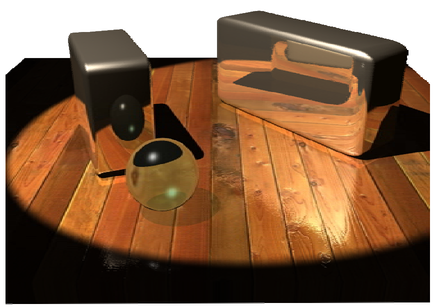
- Ψευδοκώδικας:
  - Αναδρομή σταματά είτε όταν η ακτίνα δεν τέμνεται με κανένα σώμα είτε αν ξεπεράσουμε το MAX_DEPTH επίπεδο.



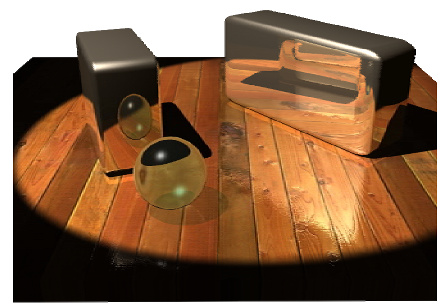
(α) Βάθος 0: καθόλου ανακλάσεις – διαθλάσεις.



(β) Βάθος 1: πρώτη ανάκλαση.



(γ) Βάθος 2: εμφανίζεται το εσωτερικό της σφαίρας (χρειάζονται δύο διαθλάσεις) καθώς και διπλές ανακλάσεις.



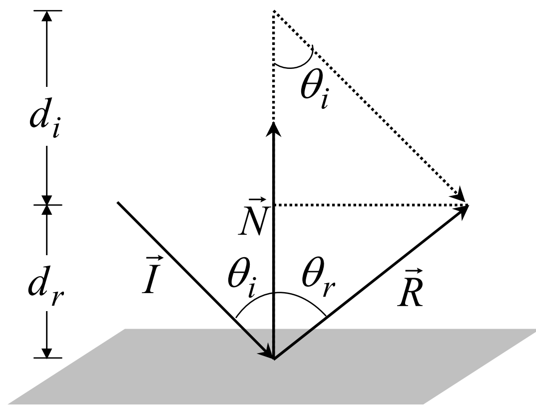
(δ) Βάθος 3: εμφανίζεται η ανάκλαση του εσωτερικού της σφαίρας, καθώς και τριπλές ανακλάσεις.

```
function RayTrace(r, depth) //r:ακτίνα, depth:βάθος αναδρομής {
      //αν έχουμε υπερβεί το επιτρεπτό βάθος αναδρομής θέσε ως τρέχον χρώμα το μαύρο
      if (depth>MAX_DEPTH)
          I=BLACK;
      else {
          //βρες το πλήθος num_of_objects των αντικειμένων που συναντά η ακτίνα και
          //τοποθέτησέ τα σε μία λίστα object_list
          num_of_objects=Intersections(r, object_list);
          //αν δεν υπάρχουν σημεία τομής, θέσε ως τρέχον χρώμα αυτό του υπόβαθρου
          if (num_of_objects==0)
               I=BACKGROUND;
          else {
            //βρες το πλησιέστερο σημείο τομής Σ και το αντίστοιχο αντικείμενο A
            (Σ,A)=ClosestIntersection(r, object_list);
            //αν το Σ δεν σκιάζεται, υπολόγισε τον τοπικό φωτισμό με βάση τις
            //ιδιότητες του A
            if !(InShadow(Σ))
               I_L=CalculateLocal(Σ,A);
            else I_L=BLACK;
            //υπολόγισε την ανακλώμενη και τη διαθλώμενη ακτίνα
            R=CalculateReflection(r,Σ,A);
            T=CalculateRefraction(r,Σ,A);
            //ακολούθησε τις νέες ακτίνες
            I_R=RayTrace(R,depth+1);
            I_T=RayTrace(T,depth+1);
            //υπολόγισε την ολική φωτεινότητα
            I=Combine(I_L,I_R,I_T,A); }
          }
      return(I);
    }
```

Υπολογισμός Ακτίνων Ανάκλασης και Διάθλασης

- Από την φυσική γνωρίζουμε ότι οι διευθύνσεις πρόσπτωσης $\vec{I}$, ανάκλασης $\vec{R}$ και διάθλασης $\vec{T}$ είναι συνεπίπεδες και μάλιστα το επίπεδο αυτό είναι κάθετο στην επιφάνεια (στο σημείο) πρόσπτωσης, άρα περιέχει το $\vec{N}$
  - Υπολογισμός της $\vec{R}$ όπως και στο μοντέλο Phong

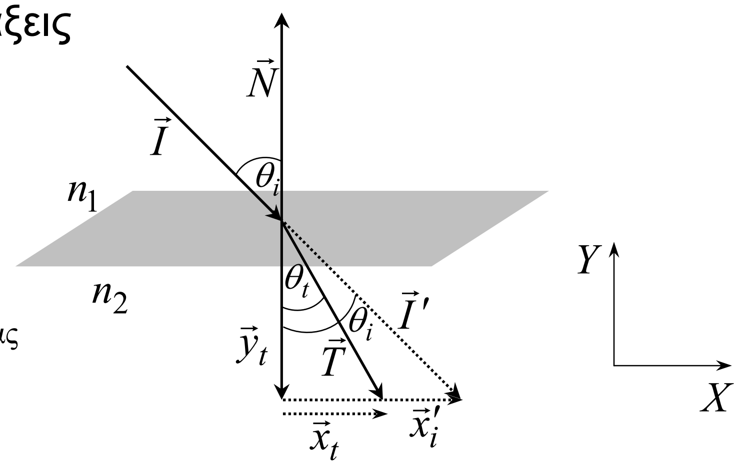
$$\vec{R} = \vec{I} - 2\left(\vec{I} \cdot \vec{N}\right)\vec{N}$$



$\vec{I}$: διάνυσμα κατεύθυνσης προσπίπτουσας ακτίνας
$\vec{R}$: διάνυσμα κατεύθυνσης ανακλώμενης ακτίνας
$\vec{N}$: κανονικό διάνυσμα επιφάνειας
θ_i : γωνία πρόσπτωσης
θ_r : γωνία ανάκλασης

  - Υπολογισμός της $\vec{T}$, μετά από πράξεις

$$\vec{T} = n_{12}\vec{I} + \left[-n_{12}\vec{I} \cdot \vec{N} - \sqrt{1+\left(\left(\vec{I} \cdot \vec{N}\right)^2 - 1\right)n_{12}^2}\right]\vec{N}$$



$\vec{T}$: διάνυσμα κατεύθυνσης διαθλώμενης ακτίνας
θ_t : γωνία διάθλασης
n_1, n_2 : δείκτες διάθλασης υλικών

Υπολογισμός Τοπικής Φωτεινότητας

- Η τοπική συνιστώσα I_L που αποδίδεται στον απευθείας φωτισμό από πηγή, χρησιμοποιεί συνήθως το μοντέλο Phong

$$I_L = k_a I_a + I_i f(d)\left[k_d\left(\vec{N}\cdot\vec{L}\right)+k_s\left(\vec{V}\cdot\vec{R}_L\right)^{nr}\right]$$

(προσοχή: $\vec{R}_L$ είναι η ανάκλαση της φωτεινής πηγής, ενώ ως εδώ $\vec{R}$ ήταν η ανάκλαση της διεύθυνσης παρατήρησης)

  – Το μοντέλο Phong υποθέτει αδιαφανή αντικείμενα και δεν έχει όρο φωτισμού από διάθλαση.

  – Πρόσθεση όρου διάθλασης $k_{st}\left(\vec{V}\cdot\vec{T}_L\right)^{nt}$ που εξαρτάται από γωνία της $\vec{V}$ σχετικά με ιδανική ακτίνα διάθλασης $\vec{T}_L$

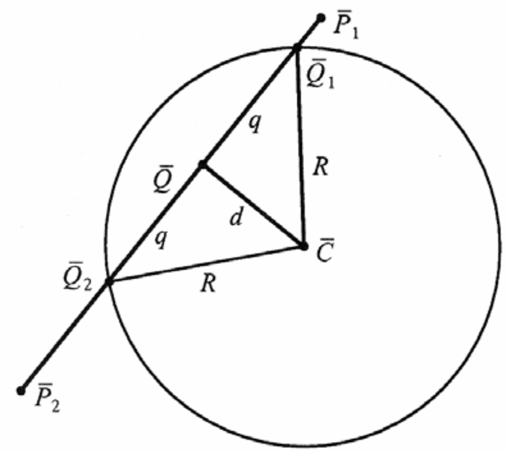
$$I_L = k_a I_a + I_i f(d)\left[k_d\left(\vec{N}\cdot\vec{L}\right)+k_{sr}\left(\vec{V}\cdot\vec{R}_L\right)^{nr}+k_{st}\left(\vec{V}\cdot\vec{T}_L\right)^{nt}\right]$$

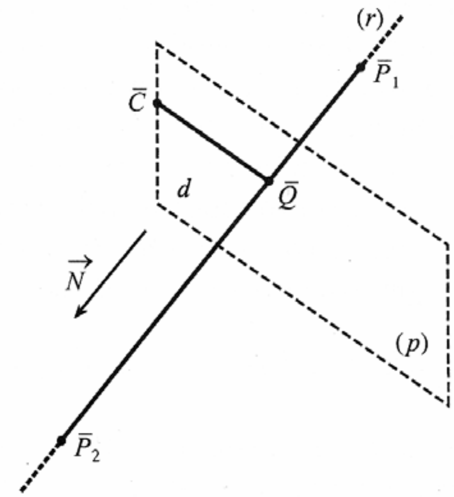
  – Οι εκθέτες nr και nt δείχνουν ότι λαμβάνουμε υπ' όψη όχι μόνο τις ιδανικές κατευθύνσεις ανάκλασης και διάθλασης.

  – Ο όρος $k_a I_a$ διατηρείται, παρόλο που έχουμε μοντέλο γενικού φωτισμού, καθώς οι ακτίνες $\vec{R}$ και $\vec{T}$ υπολογίζονται αυστηρά σε μία κατεύθυνση ελάχιστου πάχους ενώ κανονικά υπάρχει και “διάχυση” από κοντινά αντικείμενα.

Τομή Ακτίνας με Σφαίρα

- Η γεωμετρική λύση αποτελεί την ταχύτερη μέθοδο:
  - Εστω ημιευθεία (r) που ορίζεται από $\overline{P_1}$ και $\overline{P_2}$ και σφαίρα κέντρου $\overline{C}$ και ακτίνας R.
  - Εστω ότι η κάθετη στην (r) που διέρχεται από το $\overline{C}$ τέμνει την (r) στο $\overline{Q}$
  - Υπολογισμός $\overline{Q}$ (βιβλίο). Εστω d η απόσταση $\overline{Q}\,\overline{C}$

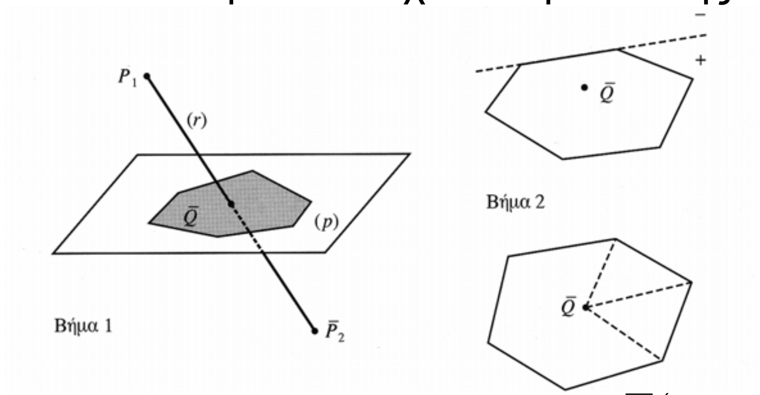




  - Αν $d>R$ δεν υπάρχει τομή ακτίνας-σφαίρας.
  - Αν $d=R$ η ακτίνα εφάπτεται της σφαίρας. Υπολογίζουμε αν το σημείο επαφής βρίσκεται μετά το $\overline{P_1}$ στην ακτίνα.
  - Αν $d<R$ υπάρχει τομή. Υπολογίζουμε τα σημεία τομής $\overline{Q_1}, \overline{Q_2}$ και ελέγχουμε αν είναι μετά το $\overline{P_1}$ (βιβλίο).

Τομή Ακτίνας με Πολύγωνο

- Πολύγωνο: το πιο διαδεδομένο στοιχείο παράστασης αντικειμένων



  – Βήμα 1: Εύρεση τομής ακτίνας (r) με άκρα $\overline{P_1}(x_1, y_1, z_1)$ και $\overline{P_2}(x_2, y_2, z_2)$ και παραμετρική εξίσωση

    $$x = x_1 + t(x_2 - x_1) \quad y = y_1 + t(y_2 - y_1) \quad z = z_1 + t(z_2 - z_1)$$

    με επίπεδο πολυγώνου $\alpha x+\beta y+\gamma z+\delta=0$: $\quad t = -\dfrac{\alpha x_1 + \beta y_1 + \gamma z_1 + \delta}{\alpha(x_2 - x_1) + \beta(y_2 - y_1) + \gamma(z_2 - z_1)}$

    » Αν ο παρανομαστής είναι 0 τότε η (r) είναι παράλληλη με το επίπεδο

    » Αν $t<0$ ή $t>1$ η τομή δεν βρίσκεται μεταξύ $\overline{P_1}$ και $\overline{P_2}$

    » Διαφορετικά υπολογίζουμε το σημείο τομής $\overline{Q}$

  – Βήμα 2: Ελεγχος αν το σημείο τομής είναι εντός του πολυγώνου

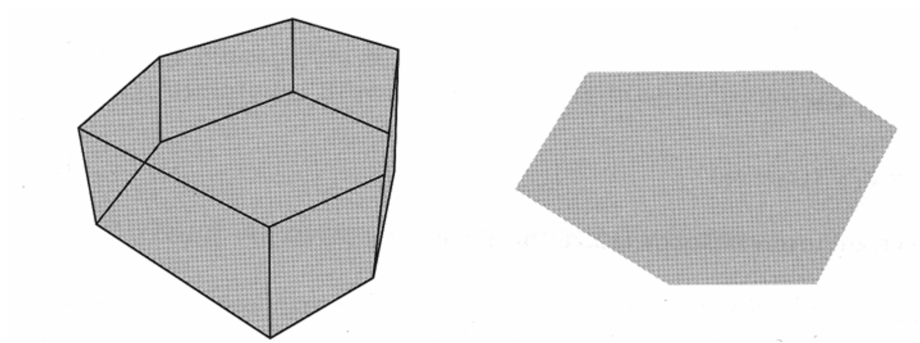
    » Ελεγχος προσήμων εξισώσεων ακμών πολυγώνου για $\overline{Q}$ (κυρτό)

    » Αθροισμα γωνιών κορυφών ως προς $\overline{Q} = 2\pi$ αν $\overline{Q}$ εντός (κυρτό)

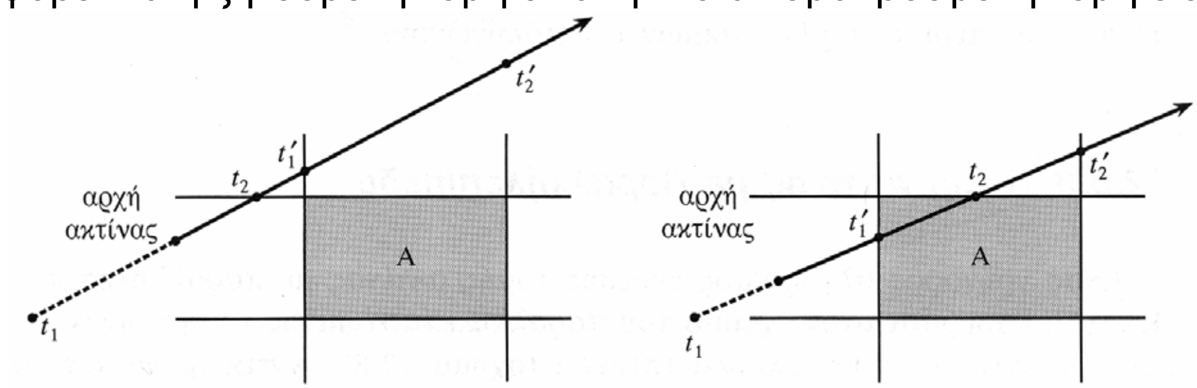
    » Γενικότερες και πιο πολύπλοκες μέθοδοι διατίθενται.

Τομή Ακτίνας με Παραλληλεπίπεδο

- Το παραλληλεπίπεδο συχνά χρησιμοποιείται ως περιβάλλον όγκος (bounding slab)



  - Για κάθε ζεύγος παράλληλων επιπέδων, καθορίζεται ποιο συναντά η ακτίνα πρώτο και ποιο δεύτερο καθορίζοντας έτσι 2 σύνολα εισόδου και εξόδου.
  - Αν η πιο απομακρυσμένη τομή εισόδου είναι μακρύτερα από την πλησιέστερη τομή εξόδου, τότε δεν τέμνει η ακτίνα το παραλληλεπίπεδο.
  - Διαφορετικά η ζητούμενη τομή είναι η πιο απομακρυσμένη τομή εισόδου:



  - Επεκτείνεται για τυχαία κυρτά πολύεδρα.

Τομή Ακτίνας με Παραλληλεπίπεδο

```
function BoxIntersection(r,A) //r:ακτίνα, A:αντικείμενο {
    O=Origin(r);    //το σημείο αρχής της ακτίνας
    //βρες το σύνολο P των παράλληλων επιπέδων που ορίζουν το A
    P=FindPlanes(A);
    //για κάθε ζεύγος παράλληλων επιπέδων
    for(each pair P_i){
          //έλεγξε αν η ακτίνα είναι παράλληλη με τα δύο επίπεδα
          if(Parallel(P_i,r))
          //αν η αρχή της ακτίνας O δεν βρίσκεται ανάμεσα στα δύο επίπεδα δεν μπορεί
          //να τέμνει το A
                    if!(Inside(P_i,O))
                       return(-1);
          //αν η ακτίνα δεν είναι παράλληλη με τα επίπεδα, υπολόγισε τις τομές
          else
                    (t_1i,t_2i)=Intersections(r,P_i);}
    //βρες το μέγιστο t_1 από τα t_11,...,t_1n
    t_1max=max(t_11,t_12,...,t_1n);
    //βρες το ελάχιστο t_2 από τα t_21,...,t_2n
    t_2min=min(t_21,t_22,...,t_2n);
    //έλεγξε τη σχέση των δύο τιμών
    if(t_1max> t_2min)
          return(-1);
    //έλεγξε αν το αντικείμενο είναι «πίσω» από την ακτίνα
    if(t_2min<0)
          return(-1);
    return(t_1max);}
```

Τομή Ακτίνας με Επιφάνειες 2ου Βαθμού

- Γενική εξίσωση επιφάνειας 2ου βαθμού:

$$Ax^2 + By^2 + \Gamma z^2 + \Delta xy + Exz + Hyz + \Theta x + Iy + Kz + \Lambda = 0$$

- Αντικατάσταση των x,y,z από παραμετρική εξίσωση ευθύγραμμου τμήματος μας δίνει:

$$at^2 + \beta t + \gamma = 0$$

όπου:

$$\begin{aligned} a &= A(x_2 - x_1)^2 + B(y_2 - y_1)^2 + \Gamma(z_2 - z_1)^2 \\ &\quad + \Delta(x_2 - x_1)(y_2 - y_1) + E(x_2 - x_1)(z_2 - z_1) + H(y_2 - y_1)(z_2 - z_1) \\ \beta &= 2Ax_1(x_2 - x_1) + 2By_1(y_2 - y_1) + 2\Gamma z_1(z_2 - z_1) \\ &\quad + \Delta(x_1(y_2 - y_1) + (x_2 - x_1)y_1) + E(x_1(z_2 - z_1) + (x_2 - x_1)z_1) \\ &\quad + H(y_1(z_2 - z_1) + (y_2 - y_1)z_1) + \Theta(x_2 - x_1) + I(y_2 - y_1) + K(z_2 - z_1) \\ \gamma &= Ax_1^2 + By_1^2 + \Gamma z_1^2 + \Delta x_1 y_1 + Ex_1 z_1 + Hy_1 z_1 + \Theta x_1 + Iy_1 + Kz_1 + \Lambda \end{aligned}$$

  – Επιλύουμε ως προς t.
  – Αν υπάρχουν 2 λύσεις $0<t_1<t_2$ τότε η t_1 μας δίνει το σημείο εισόδου της ακτίνας στο αντικείμενο.

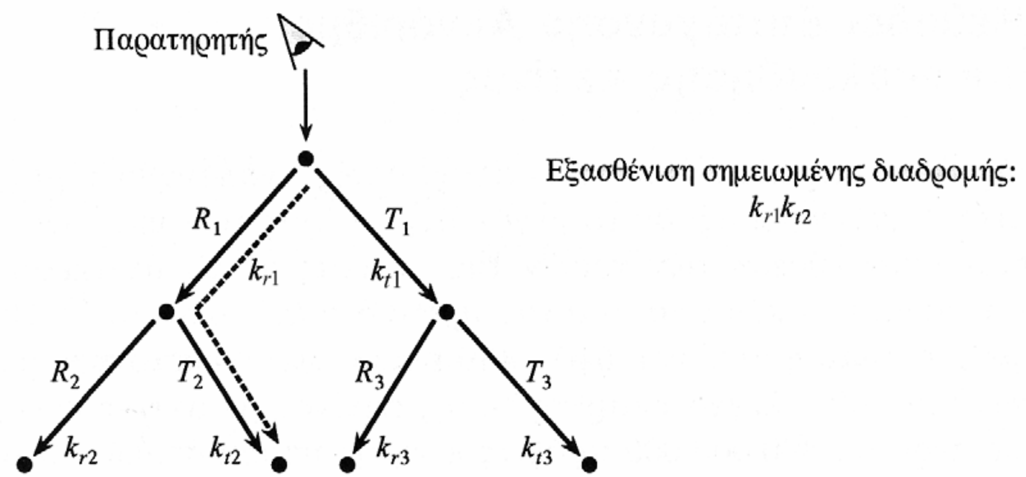
- Πολλά χρήσιμα σχήματα παριστάνονται από επιφάνειες 2ου βαθμού:
  – Κύλινδρος, κώνος, παραβολοειδές, ελλειψοειδές κλπ.

Μέθοδοι Επιτάχυνσης Αλγόριθμου Παρακολούθησης Ακτίνας

- Η παρακολούθηση ακτίνας είναι χρονοβόρα:
  - Π.χ. για 100 αντικείμενα σε ανάλυση 480×640 με 10 πράξεις / έλεγχο τομής και 1 ακτίνα / pixel έχουμε ~300.000.000 πράξεις.
  - Από την αρχή ύπαρξης της μεθόδου μελετώνται τρόποι επιτάχυνσης.
  - Μεγάλο μέρος του κόστους αφορά στις τομές ακτίνας με αντικείμενα.
  - Πολλές μέθοδοι επιτάχυνσης επικεντρώνονται στη μείωση του κόστους τομών.
- Μέθοδοι επιτάχυνσης:
  - Προσαρμοστικός έλεγχος βάθους.
  - Καταχωρητές φωτός.
  - Περιβάλλοντες όγκοι.
  - Διαμέριση χώρου.
  - Συνάφεια ακτίνων.

Προσαρμοστικός Ελεγχος Βάθους

- Κανονικό τέλος αναδρομής: ακτίνα δεν συναντά κανένα αντικείμενο ή συναντά απορροφητικό αντικείμενο ή φτάνουμε στο μέγιστο αριθμό αναδρομών:
  - Ομως με κάθε διάθλαση / ανάκλαση ή διαδρομή μέσα από απορροφητικό μέσο (π.χ. νερό) η ακτίνα εξασθενεί.
  - Π.χ. η μαρκαρισμένη διαδρομή: $k_{r1}k_{t2}I$

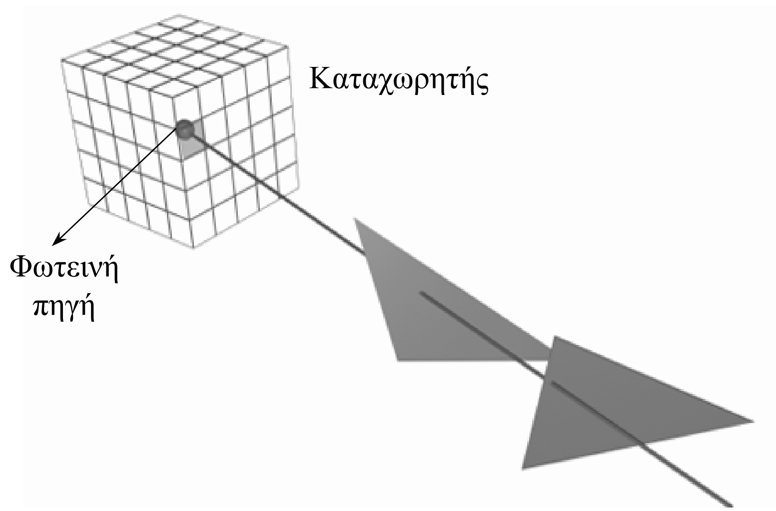


  - Προσαρμοστικός έλεγχος βάθους: παρακολούθηση σταματά όταν η εξασθένιση περάσει κάποιο κατώφλι.

```
function RayTrace(r,Σ_old,depth,weight){
//r: ακτίνα, Σ_old: προηγούμενο σημείο τομής
//depth: βάθος αναδρομής, weight: συνολικός συντελεστής εξασθένησης
    if (depth>MAX_DEPTH)
          I=BLACK;
//αν ο συντελεστής εξασθένησης είναι κάτω από το ελάχιστο, θέσε ως τρέχον χρώμα το μαύρο
    else if (weight<MIN_WEIGHT)
          I=BLACK;
    else{
          num_of_objects=Intersections(r,object_list);
          if (num_of_objects==0)
               I=BACKGROUND;
          else{
               (Σ,A)=ClosestIntersection(r,object_list);
               if !(InShadow(Σ))
                      I_L=CalculateLocal(Σ,A);
               else I_L=BLACK;
               //βρες τους συντελεστές ανάκλασης και διάθλασης στο σημείο Σ
               k_r=ReflectionCoefficient(Σ);
               k_t=RefractionCoefficient(Σ);
               //υπολόγισε την απόσταση από το προηγούμενο σημείο τομής
               d=Distance(Σ,Σ_old);
               //βρες το συντελεστή εξασθένησης λόγω διάδοσης στην τρέχουσα περιοχή που
               //διανύει η ακτίνα ως συνάρτηση του μέσου και της απόστασης που έχει διανυθεί
               k_trans=TransmittanceCoefficient(material,d);
               //υπολόγισε την ανακλώμενη και τη διαθλώμενη ακτίνα
               R=CalculateReflection(r,Σ,A);
               T=CalculateRefraction(r,Σ,A);
               //ακολούθησε τις νέες ακτίνες
               I_R=RayTrace(R,Σ,depth+1,weight*k_r*k_trans);
               I_T=RayTrace(T,Σ,depth+1,weight*k_t*k_trans);
               //υπολόγισε την ολική φωτεινότητα
               I=Combine(I_L,I_R,I_T,A);
               //υπολόγισε την εξασθένηση λόγω διάδοσης
               I=Attenuate(I,d,k_trans);}
          }
          return(I);}
```

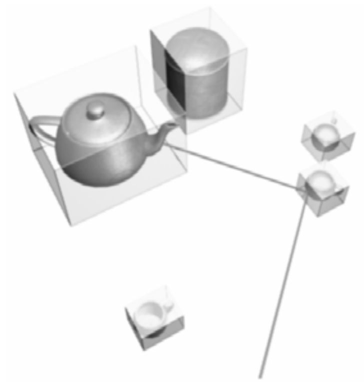
Καταχωρητές Φωτός (Light Buffers)

- Ακτίνες σκίασης: μία για κάθε σημείο τομής (ακτίνας, αντικειμένου)×κάθε φωτεινή πηγή
  - Μεγάλο υπολογιστικό κόστος.
  - Μείωση με χρήση καταχωρητή φωτός:
    - » Κύβος με κέντρο φωτεινή πηγή και επιφάνεια διαχωρισμένη σε “κελιά”.
    - » Ολα τα πολύγωνα προβάλλονται στην επιφάνεια του κύβου με κέντρο προβολής τη φωτεινή πηγή.
    - » Κάθε πολύγωνο τοποθετείται στη λίστα των κελιών που τέμνει η προβολή του.
    - » Ετσι μία ακτίνα που περνά από την πηγή και ένα κελί μπορεί μόνο να τέμνει τα πολύγωνα της λίστας του κελιού αυτού:



Περιβάλλοντες Ογκοι

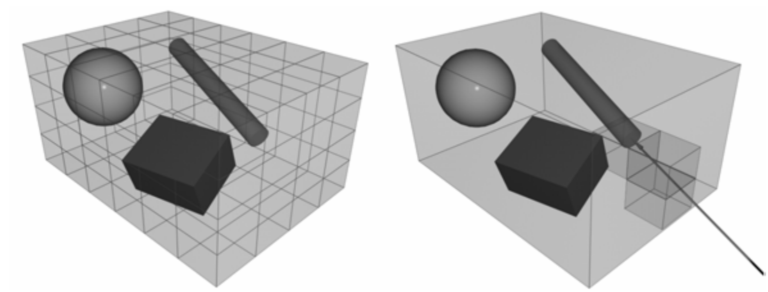
- 2 σημαντικές παρατηρήσεις:
  - Οσο πιο πολύπλοκο είναι ένα αντικείμενο, τόσο πιο χρονοβόρα η εύρεση τομής του με μία ακτίνα.
  - Μία ακτίνα που δεν τέμνει τον περιβάλλοντα όγκο, αποκλείεται να τέμνει το περιβαλλόμενο αντικείμενο.
- Ορισμός απλών περιβαλλόντων όγκων π.χ. σφαίρα, ορθογώνιο παραλληλεπίπεδο:



  - Παρά τη μείωση του κόστους κάθε ελέγχου, εξακολουθούν να χρειάζονται N έλεγχοι, όπου N ο αριθμός των αντικειμένων.
- Περαιτέρω μείωση κόστους με χρήση ιεραρχικών περιβαλλόντων όγκων:
  - Ομαδοποιούμενα αντικείμενα πρέπει να είναι κοντά, διαφορετικά αυξάνεται ο “κενός χώρος”, άρα και ο αριθμός των περιττών ελέγχων.
- Κριτήρια επιλογής περιβάλλοντος όγκου:
  - Απλότητα σχήματος.
  - Ελαχιστοποίηση “ κενού χώρου”.

Διαμέριση Χώρου

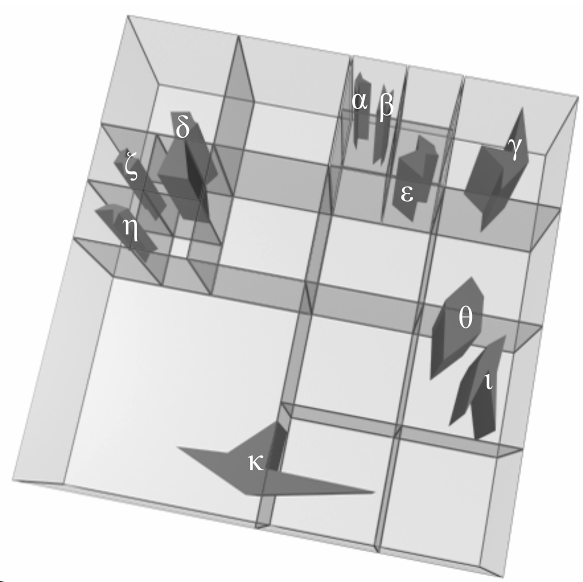
- Μείωση ελέγχων τομής με εκ διαμέτρου αντίθετη λογική περιβαλλόντων όγκων:
  - Ο χώρος διαμερίζεται σε κανονικές, μη επικαλυπτόμενες, περιοχές (voxels).
  - Ενα αντικείμενο κατατάσσεται σε ένα voxel αν το καλύπτει μερικά ή ολικά.
  - Η διαδρομή μίας ακτίνας παρακολουθείται μέσα στο χώρο των voxels και αναζητώνται τομές μόνο με τα αντικείμενα που περικλείονται σε κάθε voxel της διαδρομής αυτής:

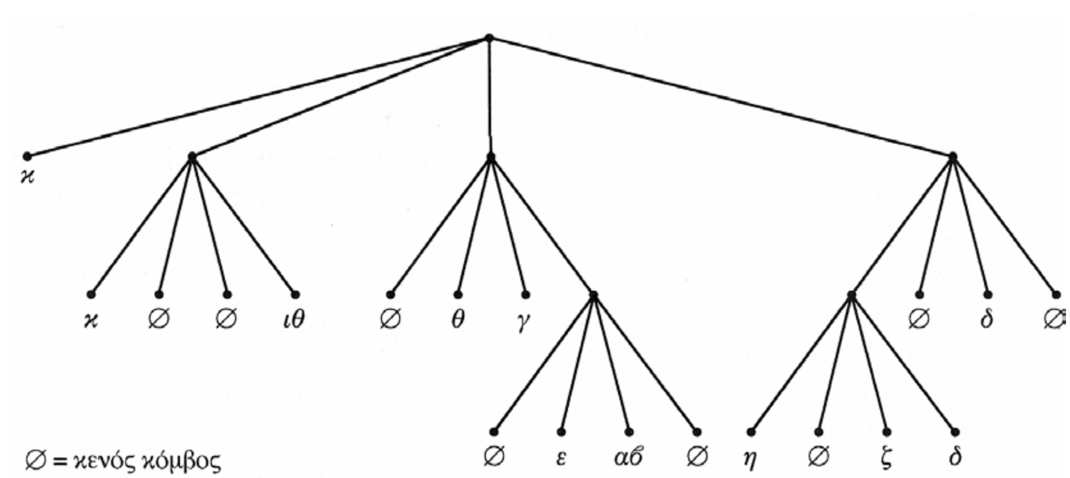


  - Τα voxels συνήθως είναι κύβοι αλλά μπορούν να έχουν οποιοδήποτε άλλο σχήμα με την προϋπόθεση της πλήρους κάλυψης του χώρου και της μη αλληλοεπικάλυψης.
- Σημαντικό πλεονέκτημα η εκμετάλλευση της χωρικής συνάφειας:
  - Υπολογίζουμε μόνο τομές αντικειμένων που βρίσκονται στο χώρο που ορίζουν τα voxels της διαδρομής της.
  - Επίσης αποφεύγεται η εξέταση πιο απομακρυσμένων αντικειμένων όταν βρεθεί το πρώτο αντικείμενο τομής.
- Η πολυπλοκότητα εξαρτάται κυρίως από την ανάλυση του χώρου voxel.

Διαμέριση Χώρου

- Επιτάχυνση με χρήση ιεραρχικών δομών:
  - Octree: δενδροειδής δομή που υποδιαιρεί κάθε voxel σε 8 ώσπου ο συνολικός αριθμός αντικειμένων σε ένα τμήμα να μην υπερβαίνει τα N.
  - Τα octrees αποτελούν προσαρμοστική υποδιαίρεση του χώρου όπου ο βαθμός διαμέρισης εξαρτάται από την κάλυψη μίας περιοχής.
  - Παρακολούθηση ακτίνας σε octree: κάθε φορά που η ακτίνα “βγαίνει” από το σημείο $\overline{P}$ ενός voxel, ψάχνουμε το octree με δεδομένο το $\overline{P}$ για να βρούμε το voxel στο οποίο “μπαίνει”:





Υβριδικές Μέθοδοι

- Περιβάλλοντες όγκοι:

  + : Ακολουθούν πιστά τα αντικείμενα

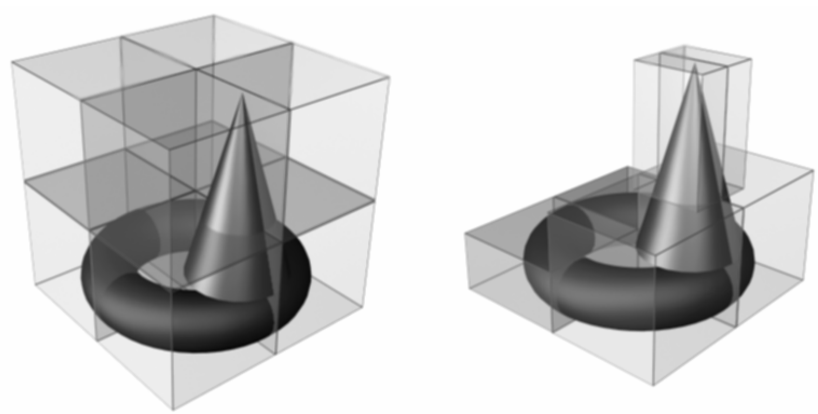
  - : Δεν προκύπτει κατάταξη τομών (θέλουμε 1^n), ούτε κατατάσσονται εύκολα σε ιεραρχίες (αλληλοεπικαλυπτόμενοι όγκοι)

- Διαμέριση χώρου

  + : Εύκολα δίνει την 1^n τομή και εύκολα δημιουργούνται ιεραρχίες

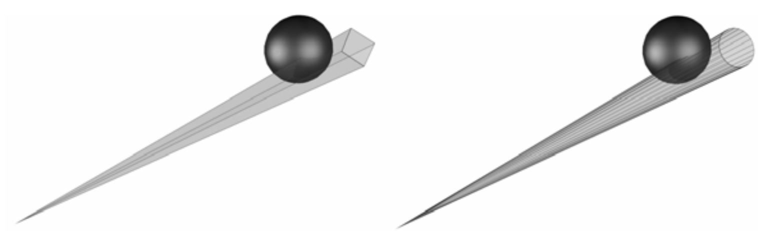
  - : Τα voxels δεν περιβάλλουν με τον καλύτερο τρόπο τα αντικείμενα

- Υβριδικές μέθοδοι: προσπάθεια συνδυασμού παραπάνω πλεονεκτημάτων
  – Αρχικά δημιουργείται ένας κατάλληλος περιβάλλων όγκος.
  – Αυτός διαχωρίζεται στα όρια των voxel κάθε επιπέδου (αποκοπή).
  – Περιβάλλοντες "υπο-όγκοι" σε κάθε επίπεδο είναι τμήματα voxel.
  – Δημιουργείται έτσι ιεραρχική δομή που ακολουθεί πιστά το αντικείμενο.



Συνάφεια Ακτίνων

- Συχνά ομάδες παραπλήσιων ακτίνων (ως προς σημείο εκκίνησης και κατεύθυνση) τέμνουν τα ίδια σώματα.
- Τρόποι εκμετάλλευσης συνάφειας ακτίνων:
  - Για επαναληπτική μέθοδο εύρεσης τομών: αρχικοποίηση δεδομένων ακτίνας με στοιχεία τομής γείτονός της.
  - Χωρική διαμέριση: παραπλήσια μονοπάτια στο χώρο voxel.
  - Παράλληλη παρακολούθηση πολλών, συναφών, ακτίνων (beam tracing) σε σχήμα πυραμίδας ή κώνου:



Παρακολύθηση Ακτίνας και Animation

- Κόστος παραγωγής πολλαπλών καρέ με παρακολούθηση ακτίνας πολύ υψηλό.
- Δυνατότητα εκμετάλλευσης χρονικής συνάφειας (temporal coherence) μεταξύ εικόνων ή αντικειμένων.
- Χρονική συνάφεια εικόνων: δημιουργούνται μόνο κάποια καρέ-κλειδιά ενώ τα υπόλοιπα προκύπτουν από ομοιότητα
  - Χρήση παρακολούθησης ακτίνας σε τμήματα εικόνας που δεν προκύπτουν από ομοιότητα.
  - Λειτουργεί μόνο σε περιορισμένες περιπτώσεις μεγάλης ομοιότητας μεταξύ καρέ, όχι κατάλληλη για μεταβολή σημείου παρατήρησης.
- Χρονική συνάφεια αντικειμένων: εξαγωγή πληροφοριών από την κίνηση των αντικειμένων μεταξύ καρέ
  - 4Δ περιβάλλοντες όγκοι.
  - Επαναπροβολή (reprojection):
    - » 1^{o} καρέ δημιουργείται με κανονική παρακολούθηση ακτίνας.
    - » Καρέ $k+1$ δημιουργείται με επαναπροβολή των σημείων του καρέ k ως προς τη νέα θέση παρατήρησης και τους μετασχηματισμούς των αντικειμένων: