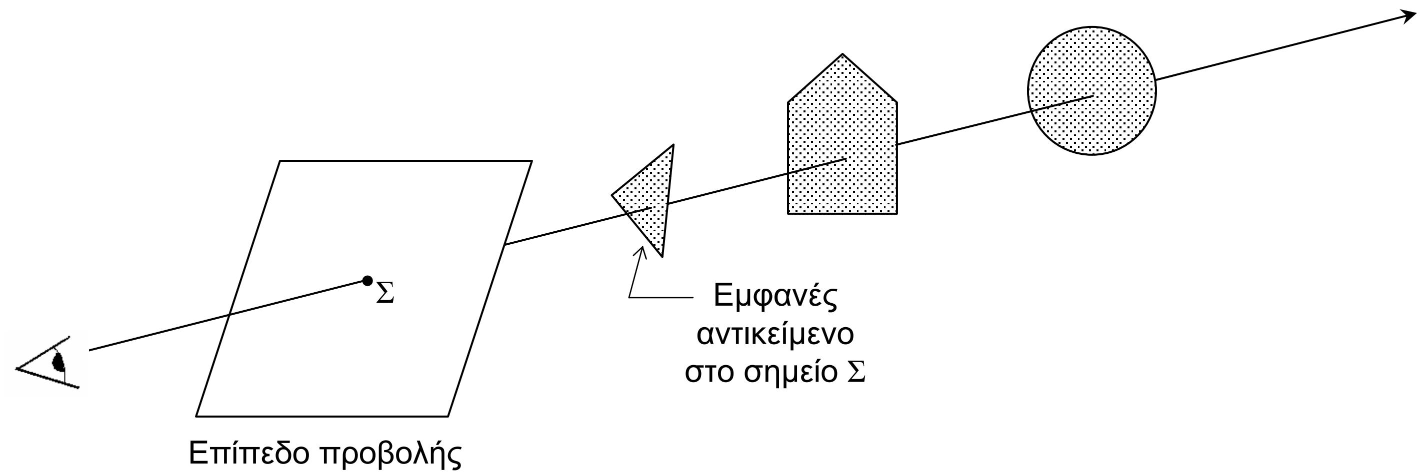
Πρόβλημα Απόκρυψης

- Ποιο είναι το εμφανές αντικείμενο (χρώμα) σε κάθε σημείο του επιπέδου προβολής;



- Χωρίζονται σε αλγόριθμους απόκρυψης ακμών και επιφανειών.
- Χρησιμοποιούν άμεσα ή έμμεσα ταξινόμηση στις διαστάσεις X Y και Z.

Αλγόριθμοι Απόκρυψης

- Χωρίζονται σε αλγόριθμους χώρου αντικειμένων και αλγόριθμους χώρου εικόνας.
- Αλγόριθμοι χώρου αντικειμένων συγκρίνουν αντικείμενα μεταξύ τους για να βρουν το πλησιέστερο σε κάθε σημείο του επιπέδου προβολής:

```
Για κάθε αντικείμενο π
     {Εύρεση των ορατών τμημάτων του π μέσω σύγκρισης (X,Y,Z)
       με όλα τα άλλα αντικείμενα;
      Χρωματισμός των ορατών τμημάτων του π στην οθόνη
     }
```

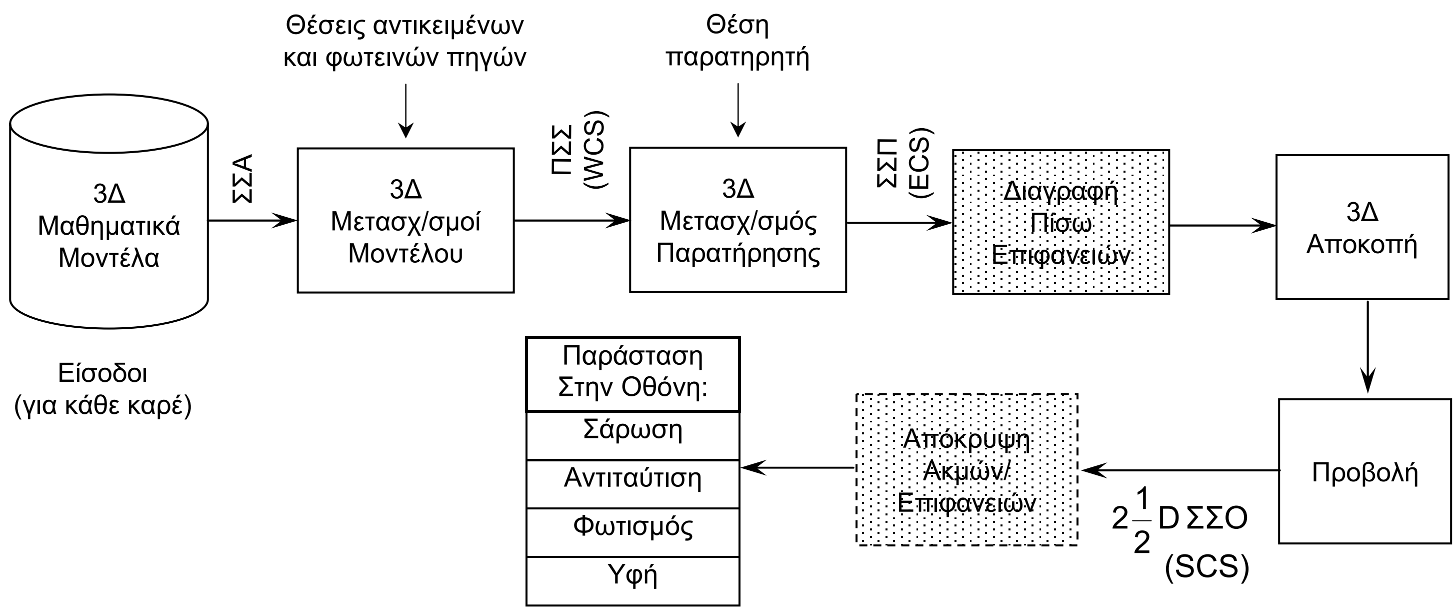
- Αλγόριθμοι χώρου εικόνας έχουν την εξής μορφή:

```
Για κάθε pixel ρ της εικόνας
     {Εύρεση του πλησιέστερου αντικειμένου που τέμνεται από την
       ακτίνα προβολής που περνά από το ρ;
      Χρωματισμός του ρ με το χρώμα του πλησιέστερου αντικειμένου στο
       σημείο τομής
     }
```

- Αλγόριθμοι χώρου αντικειμένων είναι $O(\Pi^2)$ ενώ αλγόριθμοι χώρου εικόνας είναι $O(\Pi \cdot P)$ όπου Π ο αριθμός των πολυγώνων και P ο αριθμός των pixels.

Αλγόριθμοι Απόκρυψης

- Υπολογιστική ακρίβεια.
  - Αλγόριθμοι χώρου εικόνας: ακρίβεια που απαιτεί η ανάλυση της εικόνας.
  - Αλγόριθμοι χώρου αντικειμένων: ακρίβεια ορισμού αντικειμένων (=ακρίβεια υπολογιστή).
- Θέση στη γραφική σωλήνωση εξόδου.
  - Αλγόριθμοι χώρου αντικειμένων: μετά την προβολή (διακεκομμένη γραμμή).
  - Αλγόριθμοι χώρου εικόνας: ενσωματώνονται στη διαδικασία παράστασης στην οθόνη.
  - Διαγραφή πίσω επιφανειών.

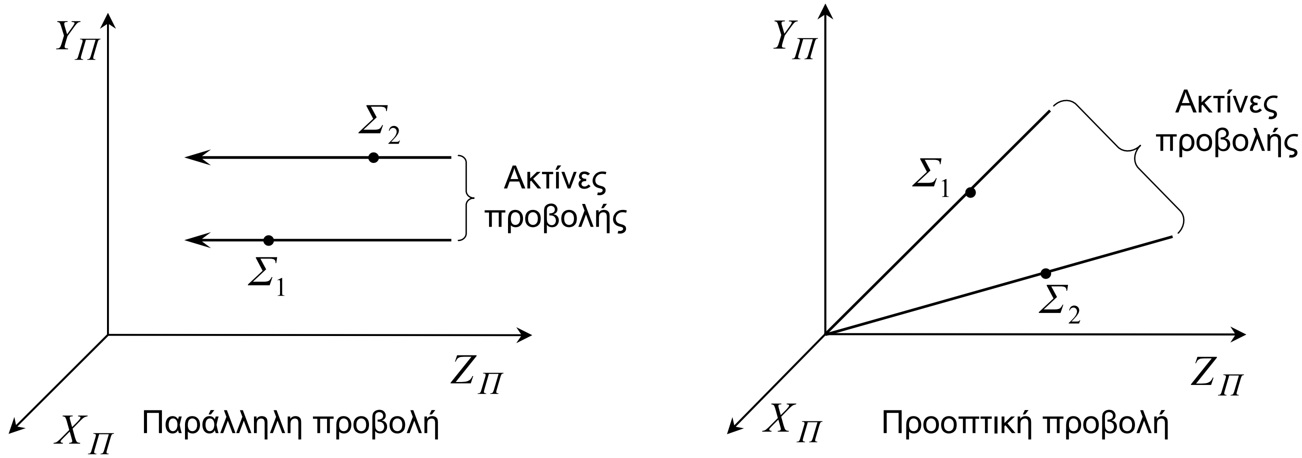


Αλγόριθμοι Απόκρυψης

- Τεχνικές βελτίωσης αποτελεσματικότητας αλγορίθμων απόκρυψης.
  - Εκμετάλλευση προοπτικού μετασχηματισμού.
  - Εκμετάλλευση συνάφειας.
  - Περιβάλλοντες όγκοι.
  - Διαμερισμός χώρου.

Προοπτικός Μετασχηματισμός

- Ζήτημα απόκρυψης μεταξύ δύο σημείων $\Sigma_1(x_1,y_1,z_1)$ και $\Sigma_2(x_2,y_2,z_2)$ υπάρχει μόνο αν τα δύο σημεία βρίσκονται πάνω στην ίδια ακτίνα προβολής.



- Συνθήκη απόκρυψης στην παράλληλη προβολή.

$$(x_1 = x_2)\ \&\ (y_1 = y_2)$$

- Συνθήκη απόκρυψης στην προοπτική προβολή.

$$(x_1 / z_1 = x_2 / z_2)\ \&\ (y_1 / z_1 = y_2 / z_2)$$

  – Απαιτεί (ακριβή) διαίρεση με z.

  – Η διαίρεση αυτή γίνεται κατά την προοπτική προβολή.

  – Η προοπτική προβολή μετατρέπει τις ακτίνες προβολής σε παράλληλες.

  – Φυλάμε z συντεταγμένη για σύγκριση βάθους.

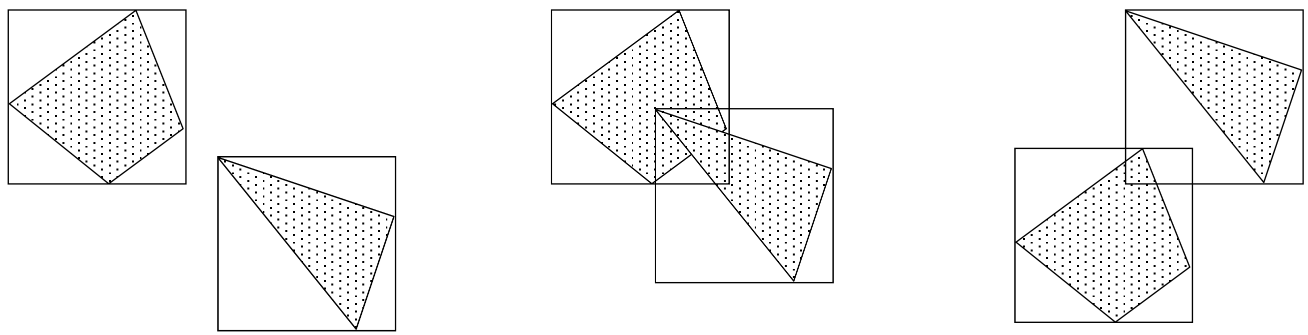
Συνάφεια

- Συνάφεια: ιδιότητα γεωμετρικών οντοτήτων να διατηρούν τοπικά σταθερές τις τιμές των χαρακτηριστικών τους ή να τις μεταβάλλουν ομαλά.
  – Π.χ. αυξητικός υπολογίσμος z σημείων πολυγώνου.

- Είδη συνάφειας:
  – Συνάφεια ακμής.
  – Συνάφεια επιφάνειας.
  – Συνάφεια γραμμών σάρωσης.
  – Συνάφεια καρέ.

Περιβάλλοντες Όγκοι

- Απλούστεροι όγκοι από τα αντικείμενα που περιβάλλουν για μείωση κόστους συγκρίσεων κατά την ταξινόμηση στις διαστάσεις X, Y και Z.
  - Συχνά έχουν τη μορφή ορθογώνιου παραλληλογράμμου (2Δ) ή ορθογώνιου παραλληλεπιπέδου (3Δ).
  - Οχι απαραίτητα κλειστοί.
- Αν οι περιβάλλοντες όγκοι δύο αντικειμένων δεν τέμνονται τότε ούτε τα αντικείμενα τέμνονται. Το αντίθετο δεν ισχύει πάντα.

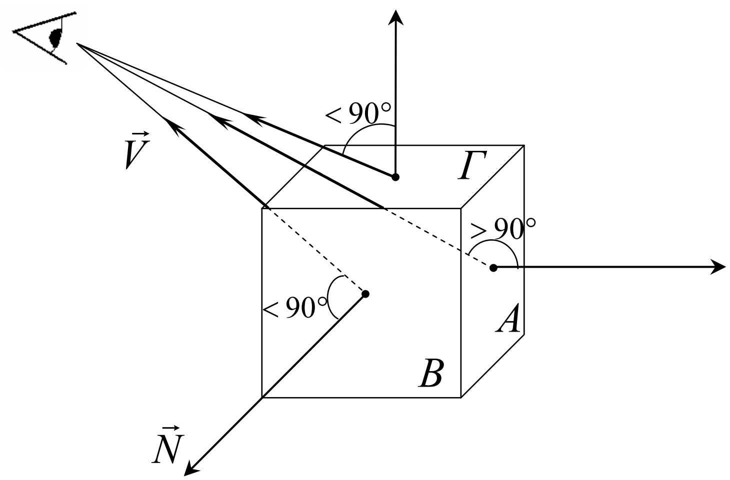


Διαμερισμός Χώρου

- Διαμέριση χώρου σε σύνολο διατεταγμένων μερών (π.χ. voxels).
- Τα μέρη του χώρου που καταλαμβάνει ένα αντικείμενο ορίζουν έμμεσα τη διάταξή του σε σχέση με άλλα αντικείμενα.

Διαγραφή Πίσω Επιφανειών

- Αν η γωνία μεταξύ $\vec{V}$ και $\vec{N}$ είναι $> 90°$ τότε η επιφάνεια είναι πίσω (αόρατη).

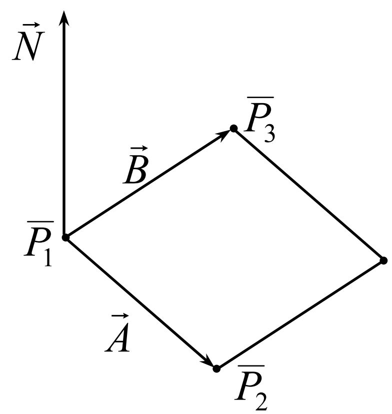


- Μια επιφάνεια είναι ορατή αν
$$\vec{V} \cdot \vec{N} = V_x \cdot N_x + V_y \cdot N_y + V_z \cdot N_z > 0$$
- Μειώνει όγκο δεδομένων κατά ~50%.
- Λύνει πρόβλημα απόκρυψης για ένα κυρτό αντικείμενο.

Διαγραφή Πίσω Επιφανειών

- Λειτουργεί στο χώρο αντικειμένων και είναι $O(\Pi)$.
- $\vec{V}$ μπορεί να υπολογισθεί από μία κορυφή $\overline{P}$ της επιφάνειας.
  - Αν ο παρατηρητής βρίσκεται στο κέντρο του ΣΣΠ τότε $\vec{V} = -\vec{P}$
- $\vec{N}$ μπορεί να υπολογισθεί από 3 διαδοχικές, μη συγγραμμικές κορυφές $\overline{P_1}$ $\overline{P_2}$ και $\overline{P_3}$

$$\vec{N} = \vec{A} \times \vec{B} = \left(\overline{P}_2 - \overline{P_1}\right) \times \left(\overline{P}_3 - \overline{P_1}\right)$$



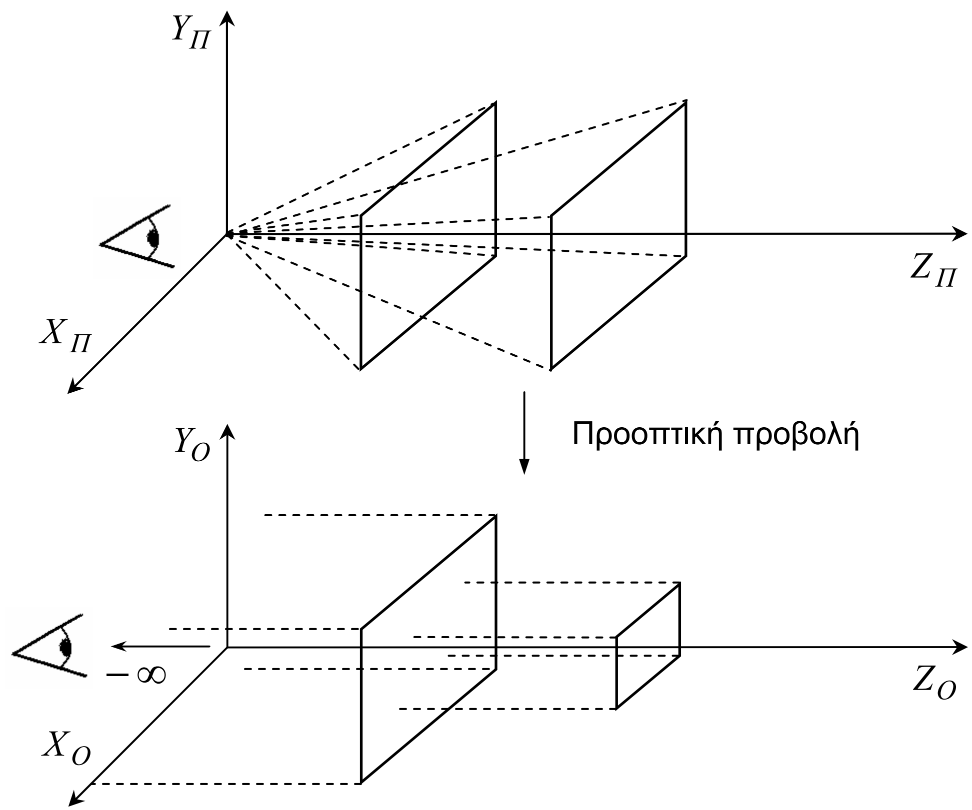
  - Προσοχή $\vec{A} \times \vec{B} \neq \vec{B} \times \vec{A}$. Χρήση σειράς κορυφών.

Αλγόριθμοι Απόκρυψης Επιφανειών

- Ακολούθησαν την εμφάνιση της πλεγματικής οθόνης.
- 4 Βασικές κατηγορίες.
  - z-buffer (πιο διαδεδομένος).
  - scanline.
  - ταξινόμηση κατά βάθος.
  - υποδιαίρεση επιφάνειας.

Αλγόριθμοι Απόκρυψης Επιφανειών

- Βασική πράξη: σύγκριση βάθους 2 στοιχείων με ίδιο XY.
  - Η προοπτική προβολή διευκολύνει.
  - Μετατρέπει ακτίνες προβολής ώστε να είναι παράλληλες του Z.
  - Ίσες αποστάσεις στον Z_π δεν μετασχηματίζονται σε ίσες αποστάσεις στον Z_o.
  - Z_o μεταβάλλεται ταχύτερα καθώς πλησιάζει τη μέγιστη τιμή του (1).



Αλγόριθμος z-buffer

- Απαιτεί ύπαρξη μνήμης βάθους (z-buffer) για κάθε pixel.
  - Οι τιμές βάθους βρίσκονται (για τα περισσότερα σχήματα) με παρεμβολή.
  - Έμμεση ταξινόμηση.
  - Για κάθε pixel ο z-buffer φυλάει την ελάχιστη (ως τώρα) τιμή βάθους στο pixel αυτό.
  - Αρχικοποίηση στο μέγιστο z (πίσω επίπεδο αποκοπής).

Αλγόριθμος z-buffer

```
/* Αλγόριθμος z-buffer */
/* Αρχικοποίηση: m,n οι διαστάσεις της οθόνης */
for (x=0; x<m; x++) {
    for (y=0; y<n; y++) {
      z_buffer[x,y]=f;                          /*μέγιστο βάθος*/
      frame_buffer[x,y]=background;             /*φόντο*/
     }
   }
/*Επεξεργασία πολυγώνων*/
for (π=0; π<number_of_polygons(); π++) {
/*Επεξεργασία scanlines πολυγώνου ymin…ymax*/
   for (y=ymin; y<=ymax; y++) {
   /*Βρες με γραμμική παρεμβολή μεταξύ των αντίστοιχων κορυφών,
     τις τιμές τομής με scanline y        :xleft,xright
     τις τιμές βάθους στις τομές          :zleft,zright
     τις τιμές χρωματισμού στις τομές     :cleft,cright*/
    for (x=xleft; x<=xright; x++) {
                /*Βρες με γραμμική παρεμβολή μεταξύ xleft και xright την τιμή
                 βάθους z και χρώματος c σε κάθε pixel x,y*/
                if (z<z_buffer[x,y]) {
                           z_buffer[x,y]=z;
                           frame_buffer[x,y]=c;
                          }
                }
     }
}
```

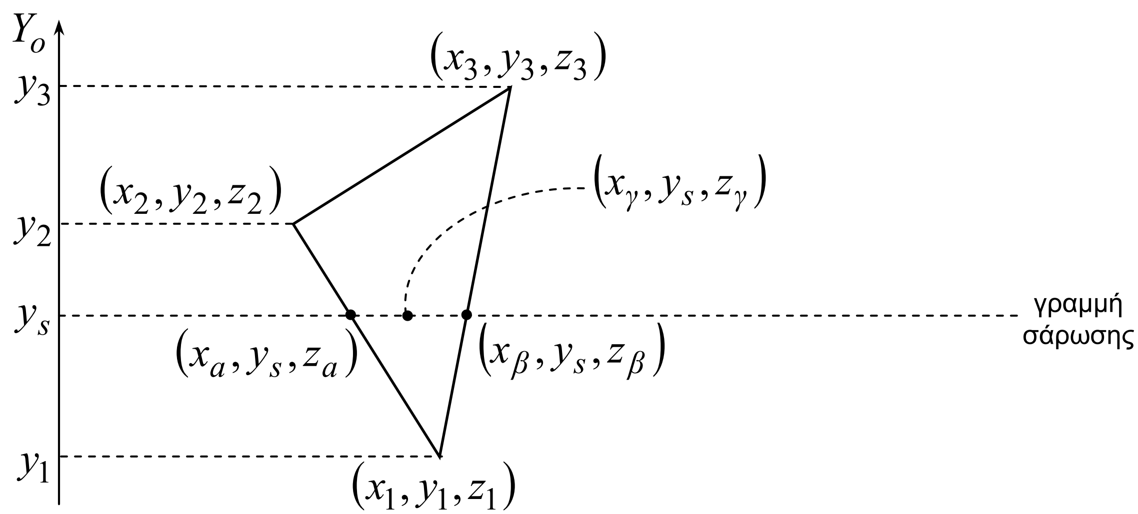
Αλγόριθμος z-buffer

- Τομές πλευρών πολυγώνου με scanlines βρίσκονται με λίστα ενεργών πλευρών.
- Γραμμική παρεμβολή z.

$$z_a = z_1 + (z_2 - z_1)\frac{y_s - y_1}{y_2 - y_1}$$

$$z_\beta = z_1 + (z_3 - z_1)\frac{y_s - y_1}{y_3 - y_1}$$

$$z_\gamma = z_\alpha + (z_\beta - z_\alpha)\frac{x_\gamma - x_a}{x_\beta - x_a}$$



Αλγόριθμος z-buffer

- $O(\Pi \cdot S)$, όπου Π ο αριθμός πολυγώνων και S ο μέσος αριθμός pixel ανά πολύγωνο.
  - Για τυπικές σκηνές $\Pi \cdot S \ \propto \ m \cdot n$ (σταθερό).
- Πλεονεκτήματα z-buffer:
  - Ευκολία υλοποίησης σε H/W ή S/W.
  - Επεξεργασία πολυγώνων με τυχαία σειρά.
  - Δυνατότητα χρήσης για μη πολυγωνικά αντικείμενα (με συνάρτηση βάθους).
- Ανάγκη σε μνήμη:
  - Ένας καλός z-buffer απαιτεί 32 bits/pixel.
  - Για ανάλυση 1024x1024 αυτό συνεπάγεται 4Mbytes.

Ειδικές Εφαρμογές z-buffer

- Συνδυασμός εικόνων
  - Έστω $(F_A, Z_A), (F_B, Z_B)$ οι frame και z-buffer εικόνων A και B.
  - A και B μπορεί να δημιουργήθηκαν χωριστά. Συνδυασμός:

```
for (x=0; x<m; x++)
   for (y=0; y<n; y++)
     {Z_C[x,y]=(Z_A[x,y]<Z_B[x,y])?Z_A[x,y]:Z_B[x,y];
      F_C[x,y]=(Z_A[x,y]<Z_B[x,y])?F_A[x,y]:F_B[x,y];
     }
```

- Τοποθέτηση 3Δ αντικειμένων σε σκηνή με χρήση z-buffer.
  - 3Δ cursor.
  - Δεν μεταβάλλουμε περιεχόμενα z-buffer.

Αλγόριθμοι Απόκρυψης Ακμών

- Ανάπτυξη την εποχή της διανυσματικής οθόνης.
- Σήμερα χρήσιμοι για παράσταση περιγραμμάτων (π.χ. δοκιμές animation).
- Εξετάζουμε γενικό αλγόριθμο για μη διαφανή πολυγωνικά αντικείμενα που δεν διαπερνά το ένα το άλλο:
  - Βήμα 1: Διαίρεση ακμών σε τμήματα που είναι είτε ολικώς εμφανή είτε ολικώς μη εμφανή.
  - Βήμα 2: Καθορισμός, για κάθε τέτοιο τμήμα, εάν είναι εμφανές.

Αλγόριθμος Απόκρυψης Ακμών

- Λειτουργεί στο χώρο αντικειμένων μετά τον αποκλεισμό πίσω επιφανειών, την αποκοπή και την προβολή:
  - Ενα τμήμα ακμής δεν είναι εμφανές μόνο αν κρύβεται από κάποια επιφάνεια (πολύγωνο).
- Υπολογισμός των σημείων τομής της προβολής κάθε ακμής με τις προβολές των ακμών όλων των άλλων πολυγώνων:
  - $O(e^2)$ για e ακμές.
  - Με χρήση της παραμετρικής εξίσωσης.
  - Εστω ακμή $k = \overline{P}_1\overline{P}_2$ και προβολή της $k' = \overline{P}_1'\overline{P}_2'$
    $$k' = \overline{P}_1' + \mu\left(\overline{P}_2' - \overline{P}_1'\right) \quad \text{με } \mu \in [0,1]$$
  - Τα $n(k')$ σημεία τομής της k' με άλλες ακμές δίνονται από τιμές της παραμέτρου $\mu_1, \mu_2, \ldots, \mu_{n(k')}$
    $$0 \le \mu_1 \le \mu_2 \le \ldots \le \mu_{n(k')} \le 1$$
  - Κάθε ζεύγος (μ_i , μ_{i+1}) ορίζει ένα τμήμα ακμής.

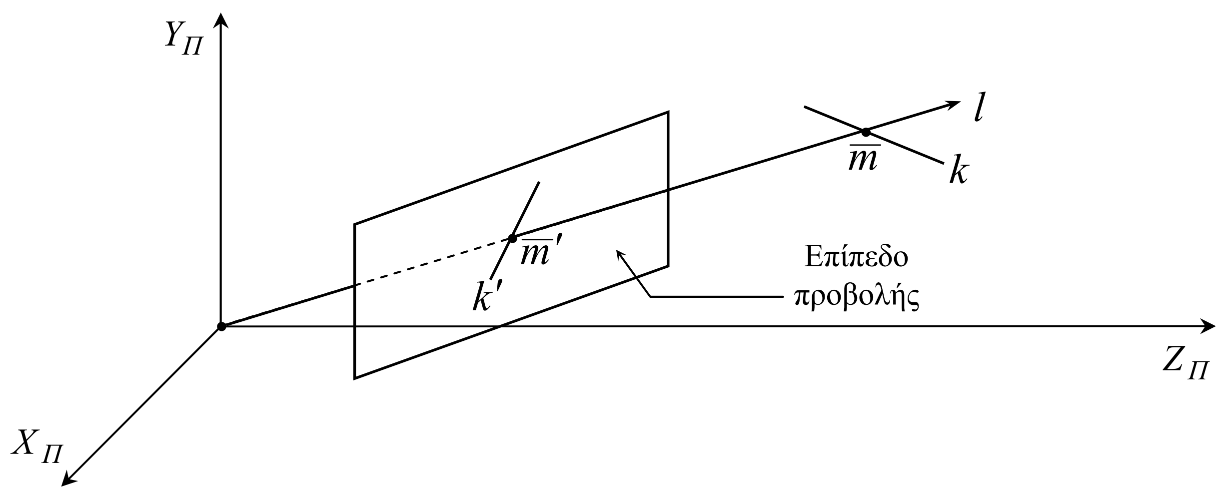
Αλγόριθμος Απόκρυψης Ακμών

- Καθορισμός ορατότητας τμήματος ακμής:
  - Αρκεί ο έλεγχος ορατότητας οποιουδήποτε σημείου του.
  - Εστω το μέσον $\overline{m}'$ του k' με παραμετρική τιμή:

$$\sigma = \frac{1}{2}\left(\mu_i + \mu_{i+1}\right)$$

$$\overline{m}' = \overline{P}_1' + \sigma\left(\overline{P}_2' - \overline{P}_1'\right) \qquad (8.1)$$

- Ορατότητα $\overline{m}'$ εξαρτάται από $\overline{m}$ της k:
  - $\overline{m}$ δεν είναι απαραίτητα μέσον της k.



Αλγόριθμος Απόκρυψης Ακμών

- Απειρα σημεία της l προβάλοντια στο $\overline{m}'$. Ομως γνωρίζουμε επιπλέον τα άκρα $\overline{P_1}$ και $\overline{P_2}$ της k:

$$\overline{m} = v \cdot \overline{m}' \qquad (8.2)$$

$$\overline{m} = \overline{P_1} + \mu\left(\overline{P_2} - \overline{P_1}\right) \quad \text{για κάποιο } \mu \qquad (8.3)$$

- Από τις (8.1), (8.2) και (8.3) :

$$v \cdot \left(\overline{P_1}' + \sigma\left(\overline{P_2}' - \overline{P_1}'\right)\right) = \overline{P_1} + \mu\left(\overline{P_2} - \overline{P_1}\right)$$

(σ δοσμένο, v, μ άγνωστα, μ το ζητούμενο)

- Από την προοπτική προβολή:

$$\overline{P_1}' = \frac{d}{z_1}\overline{P_1} \qquad \overline{P_2}' = \frac{d}{z_2}\overline{P_2}$$

- Αρα:

$$v \cdot d\left(\frac{\overline{P_1}}{z_1} + \sigma\left(\frac{\overline{P_2}}{z_2} - \frac{\overline{P_1}}{z_1}\right)\right) = \overline{P_1} + \mu\left(\overline{P_2} - \overline{P_1}\right) \qquad (8.4)$$

Αλγόριθμος Απόκρυψης Ακμών

- Η z-συντεταγμένη της (8.4) είναι:

$$v \cdot d = z_1 + \mu(z_2 - z_1)$$

- Λύνοντας ως προς v και αντικαθιστώντας στην (8.4):

$$\mu(\sigma-1)\overline{P_2} - \sigma(\mu-1)\frac{z_1}{z_2}\overline{P_2} + \sigma(\mu-1)\overline{P_1} - \mu(\sigma-1)\frac{z_2}{z_1}\overline{P_1} = 0 \qquad (8.5)$$

- Η (8.5) ισχύει για οποιαδήποτε $\overline{P_1}$ και $\overline{P_2}$. Χρησιμοποιώντας τα σημεία με $x_1=1$ και $x_2=0$ και παίρνοντας τη x-συντεταγμένη της (8.5):

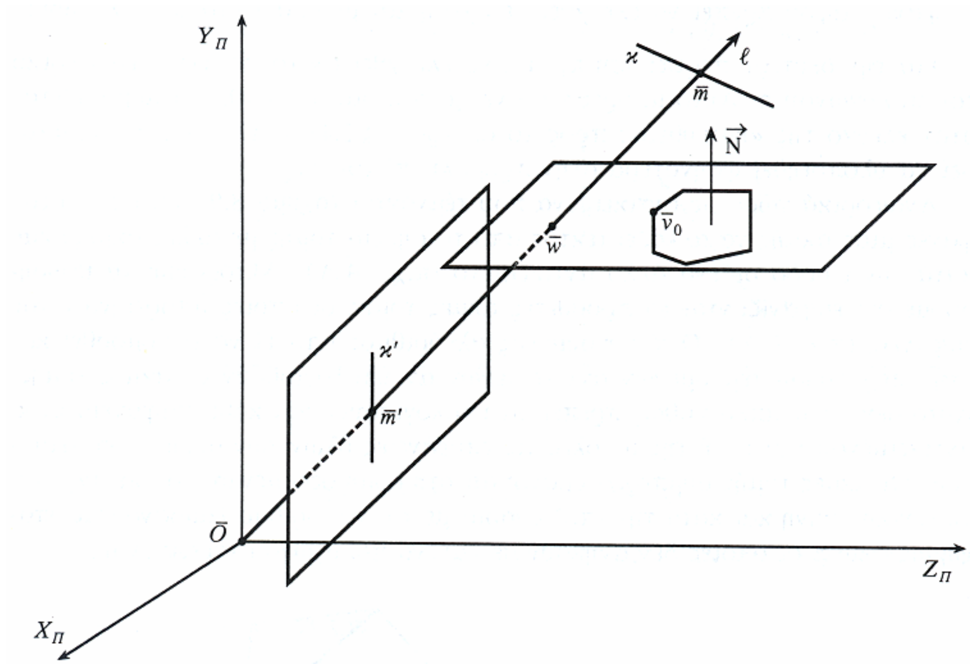
$$\sigma(\mu-1) - \mu(\sigma-1)\frac{z_2}{z_1} = 0$$

- Τελικά:

$$\mu = \frac{\sigma\, z_1}{\sigma(z_1 - z_2) + z_2}$$

Αλγόριθμος Απόκρυψης Ακμών

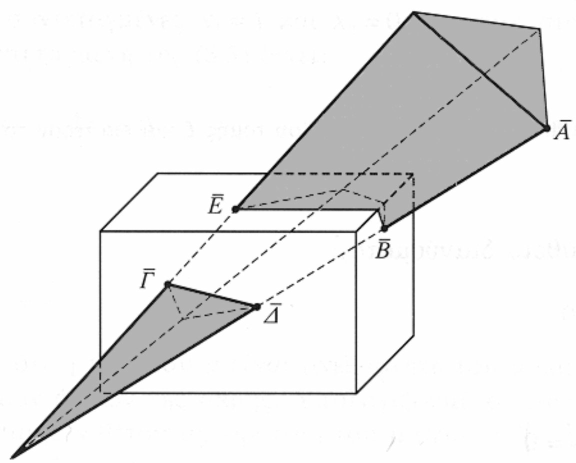
- Για διερεύνηση ορατότητας $\overline{m}$ (συνεπώς k) εξετάζουμε σημείο τομής $l = a \cdot \overline{m}$ με επίπεδο κάθε πολυγώνου π που δεν περιέχει την k.



- Η l τέμνει το επίπεδο του π στο $\overline{w} = a_w \cdot \overline{m}$
- Για $\overline{v}_0$ σημείο του π, ισχύει $(\overline{w} - \overline{v}_0) \cdot \vec{N} = 0$
- Αρα, υπολογίζουμε το α_w (και το $\overline{w}$) από την $(a_w \overline{m} - \overline{v}_0) \cdot \vec{N} = 0$

Αλγόριθμος Απόκρυψης Ακμών

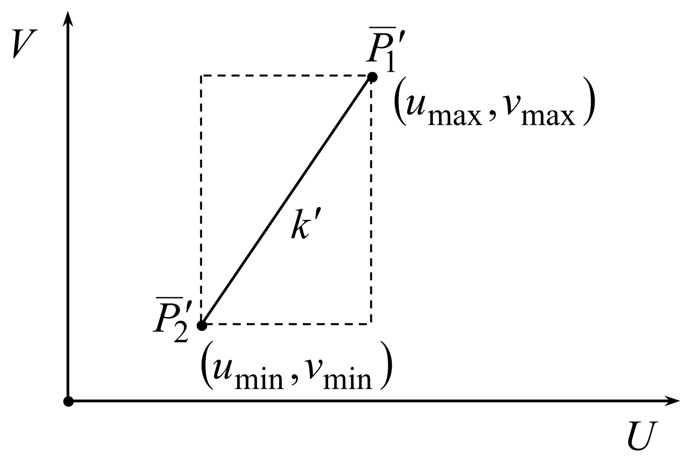
- Αν $a_w > 1$ τότε το π βρίσκεται πίσω από την k.
- Αν $a_w \leq 1$ τότε πρέπει να ελεγχθεί αν το $\overline{w}$ είναι εσωτερικό του π :
  - Αν είναι, τότε το π καλύπτει το τμήμα της k.
  - Αν όχι, τότε το π δεν καλύπτει το τμήμα της k (πρέπει βέβαια να εξετασθούν και τα άλλα πολύγωνα).
- Για αντικείμενα που διαπερνά το ένα το άλλο πρέπει να προστεθούν στη δομή παράστασης οι ακμές τομής.

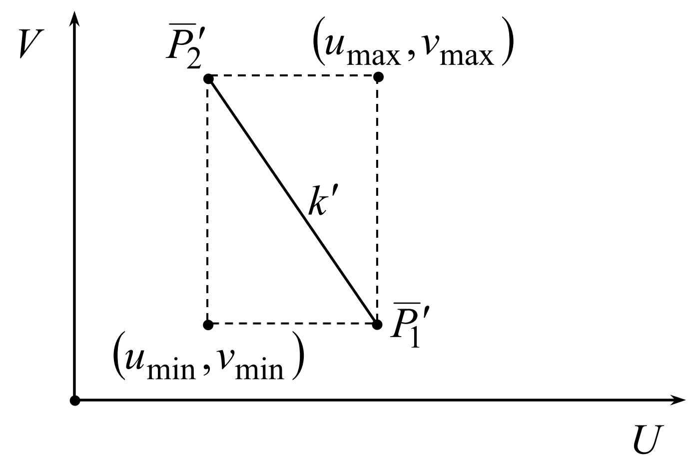


Αλγόριθμος Απόκρυψης Ακμών

- Βελτίωση απόδοσης (εύρεση τομών ακμών):
  - Χρήση περιβαλλόντων ορθογώνιων παραλληλογράμμων για μείωση κόστους υπολογισμού τομών ακμών:

$$u_{\min} = \min(u_1, u_2),\ v_{\min} = \min(v_1, v_2),\ u_{\max} = \max(u_1, u_2),\ v_{\max} = \max(v_1, v_2)$$





  - Δύο ευθύγραμμα τμήματα k_1' και k_2' δεν τέμνονται αν:

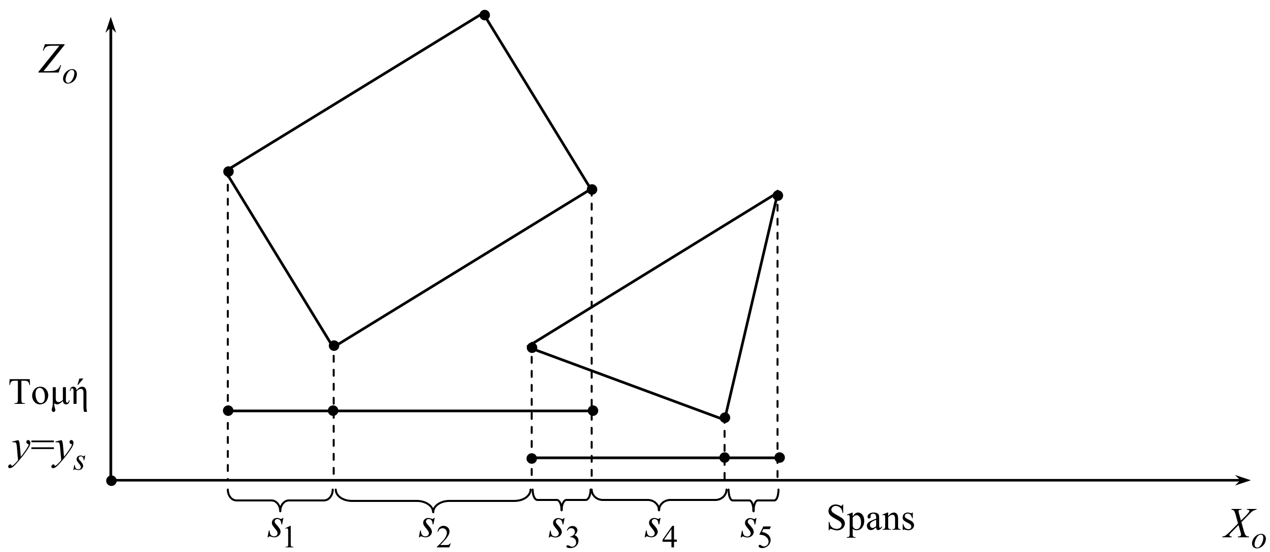
$$u_{\max,1} < u_{\min,2} \quad \text{ή} \quad u_{\max,2} < u_{\min,1} \quad \text{ή} \quad v_{\max,1} < v_{\min,2} \quad \text{ή} \quad v_{\max,2} < v_{\min,1}$$

Αλγόριθμος Απόκρυψης Ακμών

- Βελτίωση απόδοσης (εύρεση τομών ακμών):
  - Ταξινόμηση προβολών ακμών σύμφωνα με αυξανόμενες $u_{\min}$ τιμές.
  - Δυαδική αναζήτηση στην ταξινομημένη λίστα πριν από έλεγχο τομής.
  - Επεκτείνεται στις $u_{\max}$, $v_{\min}$ και $v_{\max}$ τιμές.
- Βελτίωση απόδοσης (έλεγχοι ορατότητας):
  - Χρήση περιβάλλοντος ορθογώνιου παραλληλεπίπεδου πολυγώνου π κατά τον έλεγχο τομής με l.

Αλγόριθμοι Scanline

- Επέκταση βασικού αλγόριθμου σάρωσης πολυγώνου (πολλά πολύγωνα):
  - Ιδια είδη συνάφειας.
  - Χώρος εικόνας.
- Τομές πολυγωνικών 3Δ αντικειμένων με επίπεδο $y=y_s$ είναι πολύγωνα:
  - Προβολές ακμών των πολυγώνων αυτών στο XY διαιρούν γραμμή σάρωσης σε “span”.
  - Σε κάθε span αντιστοιχεί 1 ορατή ακμή (χρώμα αντίστοιχου πολυγώνου).
  - Εύρεση ορατής ακμής μετά από ταξινόμηση ως προς z.
  - Απαγορεύονται πολύγωνα που τέμνονται.



Αλγόριθμοι Scanline

```
/*Αλγόριθμος scanline*/
Δημιούργησε με Bucket-sort ένα y-bucket με πληροφορίες για κάθε πλευρά του πολυγώνου (κεφ. 2);
/*Επανάληψη γραμμών σάρωσης*/
for (ys=0; ys<n; ys++)
{/*Επανάληψη ενεργών πολυγώνων (που τέμνουν ys)*/
  for (π=0; π<max_active; π++)
        {Εύρεση ακμής τομής π με επίπεδο y=ys;
         Εισαγωγή ακμής σε λίστα ακμών τομής ταξινομημένη ως προς xs;
         Καθορισμός span από λίστα ακμών τομής;
        /*Επανάληψη span*/
        for (s=0; s<max_span; s++)
                {Αποκοπή ενεργών ακμών (που τέμνουν s) στα όρια του s;
                 Εύρεση βάθους αποκομμένων ακμών σε ένα από τα άκρα του s;
                 Επιλογή ορατής αποκομμένης ακμής (ελάχιστο βάθος);
                 Χρωματισμός ορατής αποκομμένης ακμής;
                }
        }
}
```

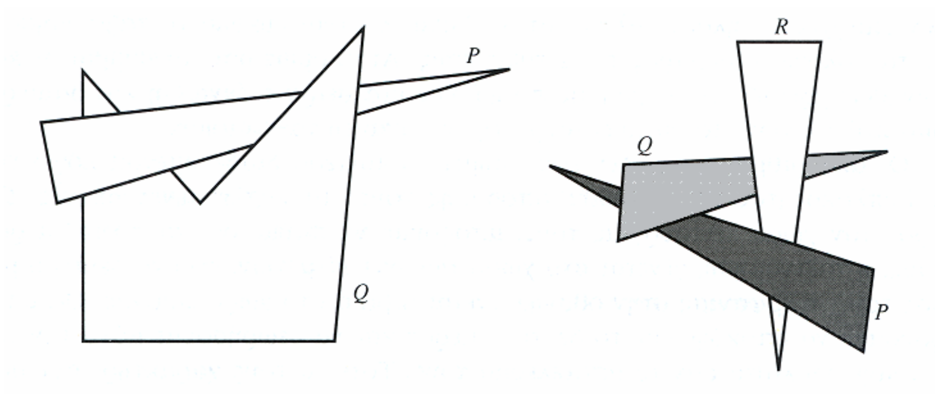
- Χρήση ΛΕΠ για αποδοτικότητα.
- Απόδοση εξαρτάται από # πολυγώνων και # span.

Αλγόριθμοι Λίστας Προτεραιότητας

- Ταξινόμηση πολυγώνων κατά βάθος και εμφάνιση με σωστή σειρά:
  - Οχι πάντα δυνατή (επικάλυψη z-έκτασης, τεμνόμενα πολύγωνα).
  - Διαμερισμός πολυγώνων.
- Μεγάλη πολυπλοκότητα.
- Χώρος αντικειμένων, αν αποτέλεσμα θεωρηθεί η ταξινομημένη λίστα.
- 1. Αλγόριθμος ταξινόμησης κατά βάθος (Newell 1972).
- 2. Αλγόριθμος BSP δένδρου (Fuchs 1983).

Αλγόριθμος Ταξινόμησης κατά Βάθος

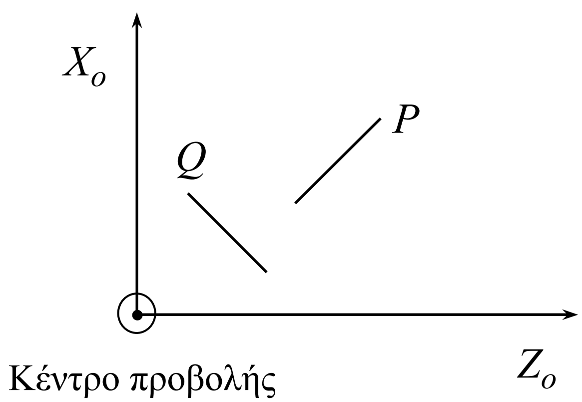
- Ταξινόμηση κατά φθίνουσα απόσταση από κέντρο προβολής.
- Σύγκριση P προ Q δεν είναι πάντα εύκολη.



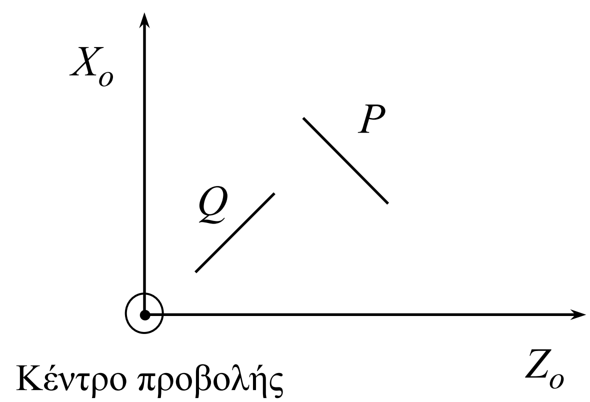
- Χρήση περιβάλλοντος παραλληλεπιπέδου $[x_{\min}(P), y_{\min}(P), z_{\min}(P)]$, $[x_{\max}(P), y_{\max}(P), z_{\max}(P)]$.

Αλγόριθμος Ταξινόμησης κατά Βάθος

- Απόφαση P προς Q με 5 ελέγχους αύξουσας πολυπλοκότητας:
  - 1. Δεν επικαλύπτονται x-εκτάσεις P και Q.
  - 2. Δεν επικαλύπτονται y-εκτάσεις P και Q.
  - 3. Το P βρίσκεται στον ημιχώρο του επιπέδου που ορίζεται από το Q στον οποίο δεν βρίσκεται το κέντρο προβολής (σχήμα Α):
    Πρόσημο $(A_Q \cdot x + B_Q \cdot y + C_Q \cdot z + D_Q) \neq$ Πρόσημο$(A_Q \cdot 0 + B_Q \cdot 0 + C_Q \cdot 0 + D_Q)$
    $\forall$κορυφή (x, y, z) του P.
  - 4. Το Q βρίσκεται στον ημιχώρο του επιπέδου που ορίζεται από το P στον οποίο βρίσκεται το κέντρο προβολής (σχήμα Β).
  - 5. Δεν επικαλύπτονται οι προβολές των P και Q στο XY.



A



B

Αλγόριθμος Ταξινόμησης κατά Βάθος

- Αν κανένας από τους παραπάνω 5 ελέγχους δεν ισχύει, γίνεται εναλλαγή των P και Q και επαναλαμβάνονται οι έλεγχοι 3 και 4.
- Αν και πάλι κανένας δεν ισχύει, κόβουμε το ένα πολύγωνο με το επίπεδο του άλλου και επαναλαμβάνουμε τους ελέγχους.

Αλγόριθμος BSP Δένδρου (Binary Space Partitioning)

- Εστω επίπεδο Ε που διαχωρίζει σύνολο πολυγώνων σκηνής:
  - Πολύγωνα που βρίσκονται στο ίδιο τμήμα με σημείο προβολής δεν μπορεί να αποκρύπτονται από πολύγωνα του άλλου τμήματος.
  - Χρήση διαχωριστικών επιπέδων που συνεπάγονται λιγότερες τομές πολυγώνων.
  - Διαχωρισμός σταματάει όταν ένα σύνολο περιέχει 1 πολύγωνο.
- Παράσταση σε δένδρο (BSP δένδρο):
  - Εσωτερικοί κόμβοι: επίπεδα διαχωρισμού.
  - Φύλλα = μη περαιτέρω διαχωριζόμενες περιοχές.
- Λύση προβλήματος απόκρυψης για θέση παρατήρησης $\bar{b}$ με ενδοδιατεταγμένη διαδρομή BSP δένδρου:
  - Εστω διαχωριστικό επίπεδο (πολύγωνο) P (κόμβος δένδρου).
  - Πρώτα παρίστανται τα πολύγωνα του τμήματος που δεν βρίσκεται το $\bar{b}$.
  - Μετά παριστάνεται το P.
  - Τέλος παρίστανται τα πολύγωνα του τμήματος που βρίσκεται το $\bar{b}$.
  - Αναδρομική διαδικασία.

Αλγόριθμοι Υποδιαίρεσης Επιφάνειας

- Αναδρομική διαίρεση επιπέδου προβολής:
  - Αν 1 πολύγωνο καλύπτει τμήμα πλήρως, τότε αυτό παίρνει το χρώμα του πολυγώνου.
  - Αναδρομή σταματά επίσης στο μέγεθος του pixel.
  - Εκμετάλλευση συνάφειας επιφάνειας (μεγάλα πολύγωνα, σταθερού χρώματος).
- Αλγόριθμος Warnock:
  - Αναδρομή σταματά αν από ένα τμήμα περνά το πολύ 1 ακμή πολυγώνου.
  - Συγκρίσεις προβολής πολυγώνου P με περιοχή οθόνης B. Χρήση περιβάλλοντων ορθογωνίων:

$$P: \ [x_{\min}(P), y_{\min}(P)], \ \ [x_{\max}(P), y_{\max}(P)]$$
$$B: \ [x_{\min}(B), y_{\min}(B)], \ \ [x_{\max}(B), y_{\max}(B)]$$

  - 4 περιπτώσεις σχέσης P και B:
    - (D: Disjoint) P εξωτερικό της B.
    - (C: Contained) P εσωτερικό της B.
    - (S: Surrounding) P καλύπτον πλήρως την B.
    - (I: Intersecting) P τέμνον την B.
  - Π.χ. σχέση D εξασφαλίζεται εάν ισχύει:

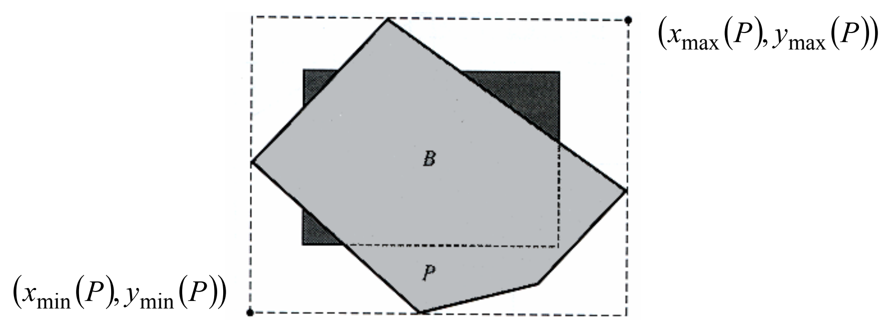
$$(x_{\min}(P) > x_{\max}(B)) \text{ ή } (x_{\max}(P) < x_{\min}(B)) \text{ ή}$$
$$(y_{\min}(P) > y_{\max}(B)) \text{ ή } (y_{\max}(P) < y_{\min}(B))$$

Αλγόριθμοι Υποδιαίρεσης Επιφάνειας

- Περιπτώσεις αλγόριθμου Warnock:
  - 1. Αν για όλα τα πολύγωνα ισχύει η σχέση D, τότε δίνεται στην B το χρώμα του φόντου.
  - 2. Αν υπάρχει μοναδικό P με σχέση C με την B, τότε δίνουμε αρχικά χρώμα φόντου στην B και μετά εμφανίζουμε το P.
  - 3. Αν υπάρχει P με σχέση S με την B, χωρίς να υπάρχουν άλλα πολύγωνα που να περιέχονται ή να τέμνουν την B, τότε η B παίρνει το χρώμα του P.
  - 4. Αν υπάρχει P με σχέση S με την B και όλα τα άλλα πολύγωνα που περιέχονται ή τέμνουν την B βρίσκονται σε μεγαλύτερη απόσταση από τον παρατηρητή, τότε η B παίρνει το χρώμα του P.
  - 5. Διαφορετικά γίνεται αναδρομική διαίρεση της B.
- Σχέση S προϋποθέτει:

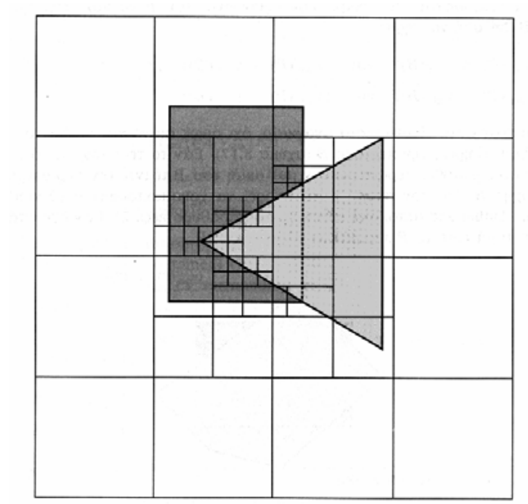
$$(x_{\min}(P) < x_{\min}(B)) \;\&\; (x_{\max}(P) > x_{\max}(B)) \;\&$$
$$(y_{\min}(P) < y_{\min}(B)) \;\&\; (y_{\max}(P) > y_{\max}(B))$$

και εξασφαλίζεται αν καμία ακμή του P δεν τέμνει την B (αλγόριθμος αποκοπής).

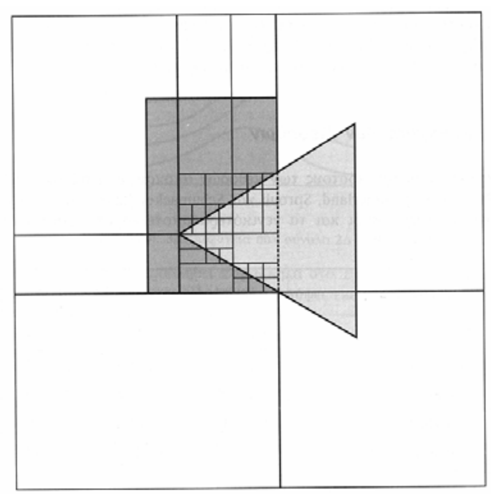


Αλγόριθμοι Υποδιαίρεσης Επιφάνειας

- Αναδρομικές διαιρέσεις Warnock:



- Μείωση υποδιαιρέσεων με βάση κορυφές πολυγώνων (Weiler - Atherton):



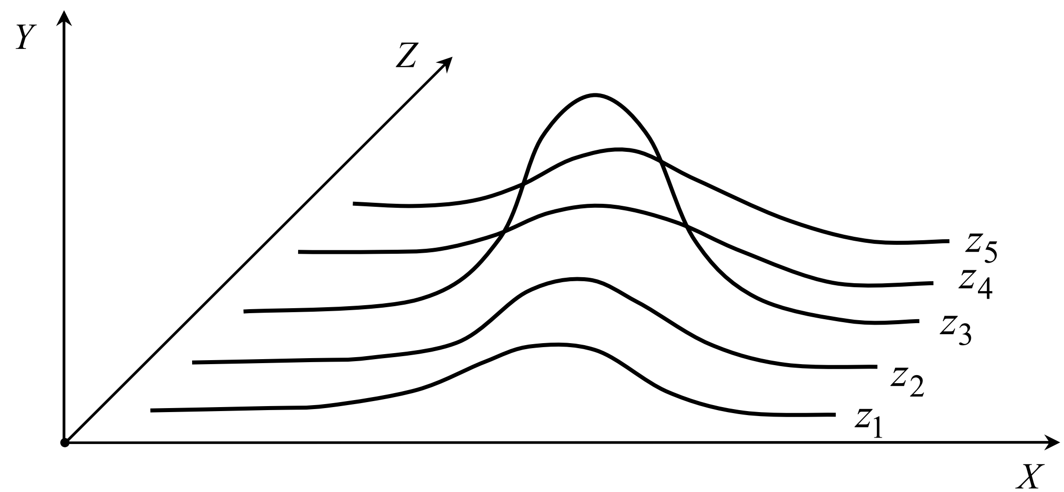
- Αλγόριθμοι υποδιαίρεσης περιοχής έχουν μεγάλο κόστος για πεπλεγμένες σκηνές.

Σύγκριση Μεθόδων Απόκρυψης

Αλγόριθμος	**Αριθμός Πολυγωνικών Επιφανειών**		
	200	5.000	120.000
depth - sort	$0,14 \bullet 10^6$	$1,4 \bullet 10^6$	$71 \bullet 10^6$
z-buffer	$7,5 \bullet 10^6$	$7,5 \bullet 10^6$	$7,5 \bullet 10^6$
Scan - line	$0,77 \bullet 10^6$	$2,9 \bullet 10^6$	$14 \bullet 10^6$
Υποδιαίρεση επιφάνειας	$1,6 \bullet 10^6$	$9 \bullet 10^6$	$43 \bullet 10^6$

Απόκρυψη 3Δ Μαθηματικών Επιφανειών

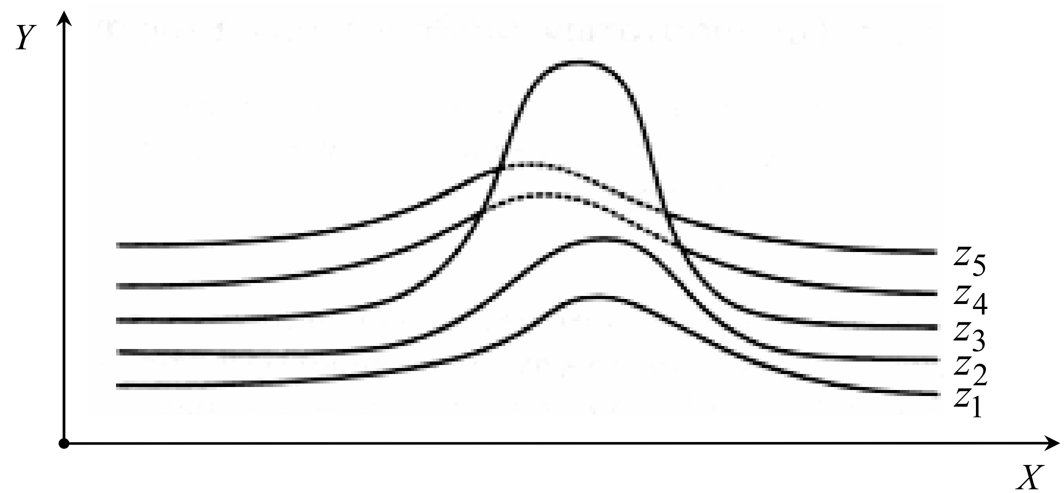
- Αλγόριθμος κινητού ορίζοντα:
  - Παράσταση $F(x, y, z)=0$ με σύνολο καμπυλών από τομές της F με επίπεδα παράλληλα στα XY, XZ ή YZ.



  - Πρέπει να αποκρυφθούν τα τμήματα καμπυλών που δεν φαίνονται.

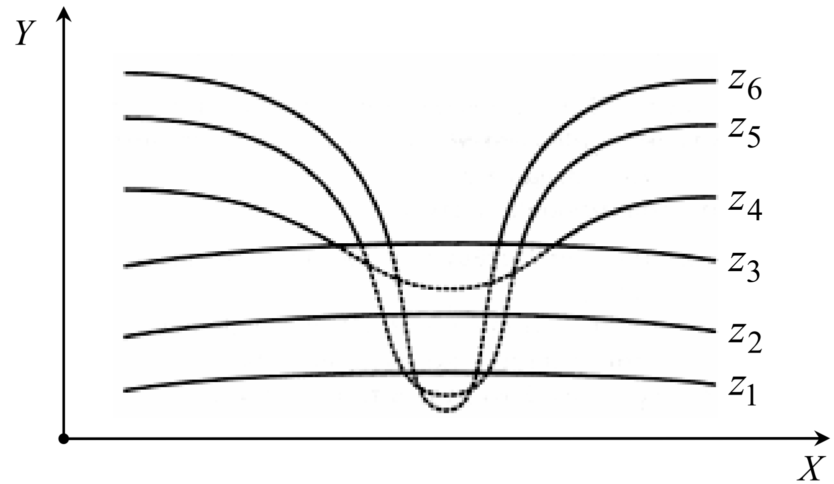
Αλγόριθμος Κινητού Ορίζοντα

- Βασικά βήματα (έστω τομές με επίπεδα παράλληλα στο XY):
  - Ταξινόμηση z-τιμών σύμφωνα με απόσταση από παρατηρητή.
  - Εύρεση σημείων της $y = f(x,z)$ $\forall\ z$ από πλησιέστερο προς πιο απομακρυσμένο, στην ανάλυση (x) της οθόνης.
  - Αν, για κάποιο x, η τιμή του y που παίρνουμε είναι μεγαλύτερη από οποιαδήποτε προηγούμενη τότε η καμπύλη είναι εμφανής στο σημείο αυτό (χρήση 1Δ array μέγιστων y-τιμών)



Αλγόριθμος Κινητού Ορίζοντα

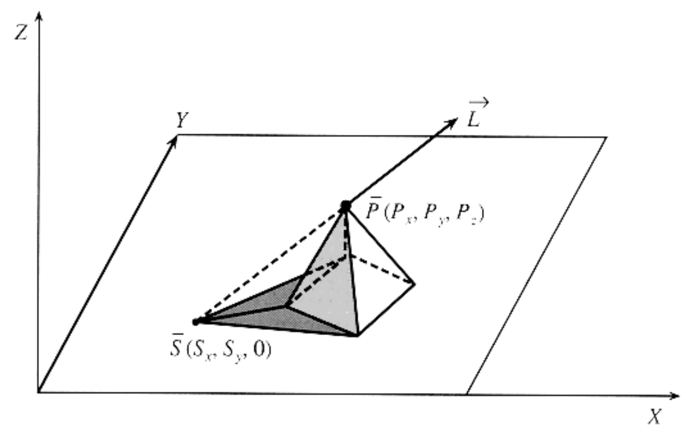
- Πρόβλημα για καμπύλες που εμφανίζονται στο κάτω μέρος της επιφάνειας (π.χ. z_5,z_6).



- Λύση με χρήση δεύτερου 1Δ array $y_{\min}$ τιμών και κατάλληλη τροποποίηση αλγορίθμου.

Σκιές

- Παραμελημένη περιοχή, παρά τη μεγάλη οπτική σημασία των σκιών.
- Εύρεση σκιών $\Leftrightarrow$ εύρεση μη ορατών επιφανειών ως προς φωτεινή πηγή (για σημειακή πηγή):
  - Χρήση αλγορίθμων απόκρυψης.
  - Για σταθερό σκηνικό και σταθερή θέση φωτεινής πηγής, εύρεση σκιών είναι ανεξάρτητη από θέση παρατηρητή.
- Απλός αλγόριθμος για σκιά αντικειμένου που βρίσκεται πάνω σε επίπεδη επιφάνεια (Blinn):



  - $\bar{S} = \bar{P} - a \cdot \vec{L}$
  - Για την z-συντεταγμένη ισχύει (S_z=0): $0 = P_z - a \cdot l_z$ ή $a = \dfrac{P_z}{l_z}$
  - Αρα $S_x = P_x - a \cdot l_x$ και $S_y = P_y - a \cdot l_y$

Απόκρυψη και Παρακολούθηση Ακτίνας (Ray Tracing)

- Αλγόριθμος παρακολούθησης ακτίνας (Whitted):
  - Ακτίνες που ορίζονται από σημείο παρατήρησης και κάθε pixel ακολουθούνται ώσπου να συναντήσουν αντικείμενο.
  - Εκεί διασπώνται ανάλογα με μοντέλο.
  - Ουσιαστικά λύνει πρόβλημα απόκρυψης.