



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

Γραφικά Ι

*Ενότητα 5: Αλγόριθμοι Περικοπής και
Απομάκρυνσης Κρυμμένων Επιφανειών*

Θεοχάρης Θεοχάρης

Σχολή Θετικών Επιστημών

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Ενότητα 5

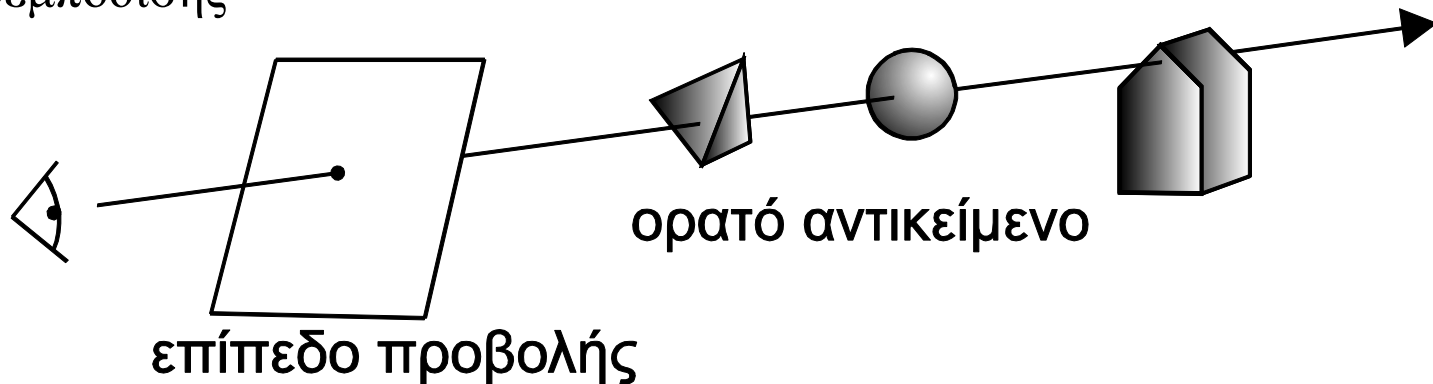
*Αλγόριθμοι Περικοπής και Απομάκρυνσης
Κρυμμένων Επιφανειών*

Εισαγωγή

- Το οπτικό μας πεδίο είναι περιορισμένο ενώ παράλληλα υπάρχει παρεμπόδιση μεταξύ αντικειμένων \rightarrow βλέπουμε ένα πολύ μικρό κομμάτι των αντικειμένων που αποτελούν τον κόσμο
- Οι αλγόριθμοι περικοπής αφαιρούν αντικείμενα που δεν σχετίζονται με την δημιουργία της εικόνας συγκεκριμένου καρέ επειδή:
 - Βρίσκονται εκτός του οπτικού πεδίου (*περικοπή στο οπτικό πεδίο*)
 - Παρεμποδίζονται από άλλα αντικείμενα (*περικοπή παρεμποδιζομένων*)
 - Παρεμποδίζονται από έμπροσθεν όψεις του ίδιου αντικειμένου (*περικοπή πίσω όψεων*)
- Περικοπή στο οπτικό πεδίο:
 - Αφαιρεί τα στοιχειώδη αντικείμενα εκτός του πεδίου παρατήρησης
 - Υλοποιείται με αλγόριθμους αποκοπής στις 3Δ

Εισαγωγή (2)

- Περικοπή πίσω όψεων:
 - Αφαιρεί τα στοιχειώδη σχήματα που αποκρύπτονται από μπροστινά στοιχειώδη σχήματα του ίδιου αντικειμένου
 - Χρησιμοποιεί γι' αυτό τα κανονικά διανύσματα
- Το πρόβλημα της παρεμπόδισης:
 - Καθορισμός των ορατών αντικειμένων σε κάθε τμήμα της εικόνας
 - Επιλύεται με τον υπολογισμό του πρώτου αντικειμένου που τέμνεται από κάθε ακτίνα προερχόμενη από το σημείο παρατήρησης
 - Για σωστή δημιουργία εικόνας πρέπει να λυθεί το πρόβλημα της παρεμπόδισης



Εισαγωγή (3)

- Αλγόριθμοι Απομάκρυνσης Κρυμμένων Επιφανειών (ΑΚΕ)
- **Σκοπός:** Εξάλειψη των κρυμμένων επιφανειών των αντικειμένων \rightarrow διαχείριση του προβλήματος παρεμπόδισης
- Οι αλγόριθμοι ΑΚΕ περιλαμβάνουν ταξινόμηση των στοιχειωδών αντικειμένων:
 - Η ταξινόμηση στην διάσταση Z (βάθος) είναι αναγκαία καθώς η ορατότητα εξαρτάται από τη διάταξη βάθους
 - Η ταξινόμηση στις διαστάσεις X, Y επιταχύνει την ταξινόμηση της διάστασης Z , καθώς τα στοιχειώδη αντικείμενα που δεν επικαλύπτονται στις διαστάσεις X, Y δεν παρεμποδίζονται
- Οι αλγόριθμοι ΑΚΕ κατηγοριοποιούνται σύμφωνα με τον χώρο λειτουργίας τους:
 - Χώρος αντικειμένων
 - Χώρος εικόνας

Εισαγωγή (4)

- Γενική μορφή ΑΚΕ χώρου αντικειμένων:

for κάθε στοιχώδες αντικείμενο

{

Βρες το ορατό μέρος //(σύγκριση με όλα τα στοιχώδη αντικείμενα)

Σχεδίασε το ορατό μέρος

}

- Η πολυπλοκότητα είναι $O(P^2)$, όπου P # στοιχειωδών αντικειμένων
- Γενική μορφή ΑΚΕ χώρου εικόνας:

for κάθε εικονοστοιχείο

{

Βρες το πλησιέστερο στοιχώδες αντικείμενο

Σχεδίασε το εικονοστοιχείο με το χρώμα του πλησιέστερου
στοιχώδους αντικειμένου

}

- Η πολυπλοκότητα είναι $O(pP)$, όπου p # εικονοστοιχείων εικόνας

Εισαγωγή (5)

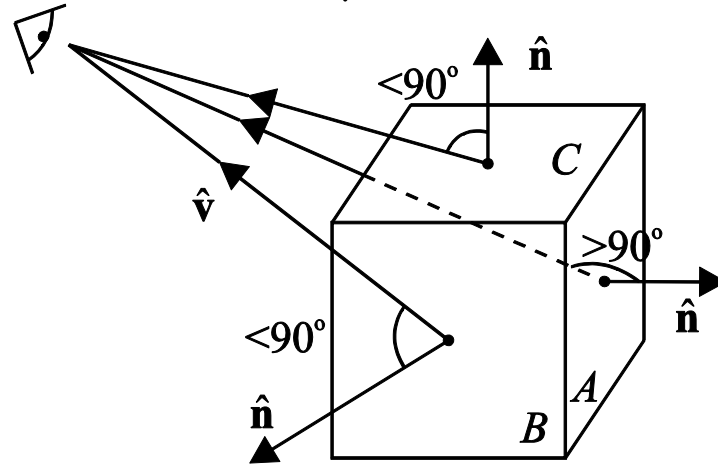
- Το υπολογιστικό κόστος των αλγορίθμων ΑΚΕ είναι μεγάλο παρά την υλοποίηση σε υλικό και την παράλληλη επεξεργασία
- Πολλά στοιχειώδη αντικείμενα μπορούν να απαλειφθούν χωρίς την χρήση αλγορίθμων ΑΚΕ
- Η περικοπή πίσω όψεων εξαλείφει περίπου τα μισά στοιχειώδη αντικείμενα (πίσω όψεις) με έλεγχο κόστους $O(P)$
- Η περικοπή στο οπτικό πεδίο αφαιρεί τα στοιχειώδη αντικείμενα που βρίσκονται εκτός του οπτικού πεδίου με κόστος $O(Pv)$, όπου v το μέσο πλήθος κορυφών ανά στοιχειώδες αντικείμενο
- Η περικοπή παρεμποδιζομένων κοστίζει επίσης $O(P)$
- ΣΥΜΠΕΡΑΣΜΑ: Πρόβλημα είναι η απόδοση των αλγορίθμων ΑΚΕ ($O(P^2)$ ή $O(pP)$ ανάλογα με τον χώρο λειτουργίας τους)

Περικοπή Πίσω Όψεων

- Τα ορατά πολύγωνα ενός αντικειμένου είναι αυτά που βρίσκονται στο 'έναντι του παρατηρητή' ημισφαίριο των αντικειμένων
- Αν, από κατασκευή μοντέλων, οι πίσω όψεις τους δεν είναι ορατές \rightarrow περικοπή αυτών. Οι απαιτήσεις είναι:
 - Να μην υπάρχει σύνορο
 - Να είναι 2D πολλαπλότητες
 - Να είναι αδιαφανή
- Η κυρτότητα δεν αποτελεί εμπόδιο
- Οι πίσω όψεις ανιχνεύονται με τον υπολογισμό της γωνίας που σχηματίζεται από το κανονικό διάνυσμα \hat{n} του πολυγώνου (κατευθυνόμενο από το αδιαφανές στερεό προς τα έξω) και το διάνυσμα παρατήρησης \hat{v}

Περικοπή Πίσω Όψεων (2)

- Αν η γωνία $> 90^\circ$ τότε το πολύγωνο είναι πίσω όψη



- Ο έλεγχος αυτός γίνεται με το εσωτερικό γινόμενο διανυσμάτων: $\hat{\mathbf{v}} \cdot \hat{\mathbf{n}} < 0$
- Η περικοπή πίσω όψεων είναι πολύ αποτελεσματική καθώς εξαλείφει περίπου το 50% των πολυγώνων
- Καθώς ο έλεγχος πίσω όψης & ο υπολογισμός του κανονικού διανύσματος & του διανύσματος παρατήρησης κάθε πολυγώνου απαιτούν σταθερό χρόνο \rightarrow η πολυπλοκότητα είναι $O(P)$, όπου P το πλήθος των πολυγώνων

Περικοπή στο Οπτικό Πεδίο

- Ο μετασχηματισμός παρατήρησης καθορίζει το οπτικό πεδίο του παρατηρητή
- Το πεδίο οριοθετείται από μέγιστη & ελάχιστη τιμή βάθους, και ορίζει ένα 3Δ στερεό που ονομάζεται *στερεό παρατήρησης*
- Η μορφή του στερεού παρατήρησης μπορεί να είναι:
 - Κόλουρη πυραμίδα {προοπτική προβολή}
 - Ορθογώνιο παραλληλεπίπεδο {ορθογραφική ή παράλληλη προβολή}
- **Σκοπός:** Εξάλειψη στοιχειωδών αντικειμένων εκτός του στερεού παρατήρησης
- Λαμβάνει χώρα μετά τον μετασχηματισμό από ΣΣΠ σε ΚΧΟ, και πριν την διαίρεση με w (για την προοπτική προβολή)
- Η περικοπή στο οπτικό πεδίο γίνεται στις 3Δ με χρήση 3Δ αποκοπής

Περικοπή στο Οπτικό Πεδίο (2)

- Αντικείμενα προς αποκοπή:
 - Σημεία
 - Ευθύγραμμα Τμήματα
 - Πολύγωνα
- Η αποκοπή σημείου είναι τετριμμένη
- Η αποκοπή ευθύγραμμου τμήματος & πολυγώνου ανάγεται στον υπολογισμό των τομών ενός ευθυγράμμου τμήματος με τα επίπεδα του αντικειμένου αποκοπής
- Στις 3Δ, το εσωτερικό του αντικειμένου αποκοπής ορίζεται ως:

$$\begin{aligned}x_{min} \leq x \leq x_{max} \\ y_{min} \leq y \leq y_{max} \\ z_{min} \leq z \leq z_{max}\end{aligned} \quad (1)$$

Περικοπή στο Οπτικό Πεδίο (3)

- Στην ορθογραφική ή παράλληλη προβολή χρησιμοποιείται ο πίνακας $\mathbf{M}_{\text{SSP}^{\text{R}}\text{KCO}}^{\text{ORTHO}}$ που απεικονίζει τα επίπεδα αποκοπής στο -1 & 1 έτσι ώστε:

$$x_{min} = y_{min} = z_{min} = -1$$

$$x_{max} = y_{max} = z_{max} = 1$$

- Στην προοπτική προβολή, ο πίνακας $\mathbf{M}_{\text{SSP}^{\text{R}}\text{KCO}}^{\text{PERSP}}$ (πριν τη διαίρεση με w) απεικονίζει τα επίπεδα αποκοπής στο $-w$ & w έτσι ώστε:

$$x_{min} = y_{min} = z_{min} = -w$$

$$x_{max} = y_{max} = z_{max} = w$$

- Η τιμή του w δεν είναι σταθερή (είναι το z_e του σημείου)
- Η αποκοπή με βάση το w ονομάζεται *ομογενής αποκοπή*

Περικοπή στο Οπτικό Πεδίο (4)

- Για ευθύγραμμο τμήμα σε παραμετρική μορφή:

$$\mathbf{I}(t) = (1-t)\mathbf{p}_1 + t\mathbf{p}_2$$

από το $\mathbf{p}_1 = [x_1, y_1, z_1, w_1]^T$ στο $\mathbf{p}_2 = [x_2, y_2, z_2, w_2]^T$,

η τιμή του w μπορεί να βρεθεί με παρεμβολή ως εξής:

$$(1-t)w_1 + tw_2$$

και οι ανισότητες **(1)** μπορούν να χρησιμοποιηθούν για να ορίσουν το κομμάτι του ευθυγράμμου τμήματος που βρίσκεται εντός του αντικειμένου αποκοπής:

$$- ((1-t)w_1 + tw_2) \leq (1-t)x_1 + tx_2 \leq (1-t)w_1 + tw_2$$

$$- ((1-t)w_1 + tw_2) \leq (1-t)y_1 + ty_2 \leq (1-t)w_1 + tw_2 \quad \mathbf{(2)}$$

$$- ((1-t)w_1 + tw_2) \leq (1-t)z_1 + tz_2 \leq (1-t)w_1 + tw_2$$

Περικοπή στο Οπτικό Πεδίο (5)

- Επιλύοντας τις 6 ανισότητες ως προς t βρίσκουμε τα 6 σημεία τομής του ευθυγράμμου τμήματος με τα επίπεδα αποκοπής:

$$\text{αριστερο: } t = \frac{x_1 + w_1}{(x_1 - x_2) + (w_1 - w_2)} \quad \text{δεξι: } t = \frac{x_1 - w_1}{(x_1 - x_2) + (w_2 - w_1)}$$

$$\text{κάτω: } t = \frac{y_1 + w_1}{(y_1 - y_2) + (w_1 - w_2)} \quad \text{πάνω: } t = \frac{y_1 - w_1}{(y_1 - y_2) + (w_2 - w_1)} \quad (3)$$

$$\text{εμπροσθεν: } t = \frac{z_1 + w_1}{(z_1 - z_2) + (w_1 - w_2)} \quad \text{οπισθεν: } t = \frac{z_1 - w_1}{(z_1 - z_2) + (w_2 - w_1)}$$

Αλγόριθμοι Αποκοπής στις 3Δ

- Οι περισσότεροι αλγόριθμοι αποκοπής 2Δ επεκτείνονται εύκολα στις 3Δ διευθετώντας κατάλληλα:
 - Τον υπολογισμό των τομών
 - Τον έλεγχο εντός / εκτός

Αποκοπή Ευθυγράμμου Τμήματος Cohen – Sutherland στις 3Δ

- Στις 3Δ, χρησιμοποιούνται 6 ψηφία για την κωδικοποίηση των 27 διαμερίσεων του 3Δ χώρου, που ορίζονται από τα επίπεδα του αντικειμένου αποκοπής:
 - 1^ο ψηφίο: Θέσε 1 για $z > z_{\max}$, αλλιώς 0
 - 2^ο ψηφίο: Θέσε 1 για $z < z_{\min}$, αλλιώς 0
 - 3^ο ψηφίο: Θέσε 1 για $y > y_{\max}$, αλλιώς 0
 - 4^ο ψηφίο: Θέσε 1 για $y < y_{\min}$, αλλιώς 0
 - 5^ο ψηφίο: Θέσε 1 για $x > x_{\max}$, αλλιώς 0
 - 6^ο ψηφίο: Θέσε 1 για $x < x_{\min}$, αλλιώς 0.

Αλγόριθμοι Αποκοπής στις 3Δ (2)

- Ένας 6-ψήφιος κωδικός ανατίθεται σε ένα 3Δ σημείο ανάλογα με ποια από τις 27 διαμερίσεις του 3Δ χώρου βρίσκεται
- Έστω c_1, c_2 οι 6-ψήφιοι κωδικοί των κορυφών $\mathbf{p}_1, \mathbf{p}_2$ ενός ευθυγράμμου τμήματος:
 - Αν $c_1 \dot{\vee} c_2 = 000000$ αποδοχή ευθυγράμμου τμήματος
 - Αν $c_1 \Omega c_2 \dot{\vee} 000000$ απόρριψη ευθυγράμμου τμήματος
- Ψευδοκώδικας του CS στις 3Δ:

```
CS_Clip_3D ( vertex p1, p2 )
```

```
{
```

```
int c1, c2;          vertex I;          plane R;
```

```
c1 = mkcode (p1);  c2 = mkcode (p2);
```

```
if ((c1 | c2) == 0) /* p1p2 βρίσκεται εντός */
```

```
else if ((c1 & c2) != 0) /* p1p2 βρίσκεται εκτός */
```

```
else { R = /* επίπεδο αντικ. αποκοπής με (ψηφίο c1 != ψηφίο c2) */
```

```
    i = intersect_plane_line (R, (p1, p2));
```

```
    if outside (R, p1) CS_Clip_3D(i, p2); else CS_Clip_3D(p1, i);}
```

```
}
```


Αλγόριθμοι Αποκοπής στις 3Δ (3)

- Διαφέρει από τον 2Δ αλγόριθμο ως προς:
 - Τον υπολογισμό των τομών
 - Τον έλεγχο εντός / εκτός
- Υπολογισμός τομών:
 - Υπολογισμός τομής επιπέδου-ευθείας στις 3Δ αντί του υπολογισμού τομής ευθείας-ευθείας στις 2Δ
 - Τα όρια αποκοπής δεν δίνονται στον ψευδοκώδικα:
 - ◆ Για την ορθογραφική ή παράλληλη προβολή, είναι σταθερά επίπεδα & χρησιμοποιούνται αλγόριθμοι τομής επιπέδου-ευθείας
 - ◆ Για την προοπτική προβολή (ομογενείς συντεταγμένες), χρησιμοποιούνται οι εξισώσεις **(3)**
- Έλεγχος εντός / εκτός αντικειμένου αποκοπής:
 - Μπορεί να γίνει με έλεγχο προσήμου στην τιμή της εξίσωσης επιπέδου R για τις συντεταγμένες του \mathbf{p}_1

Αλγόριθμοι Αποκοπής στις 3Δ (4)

Αποκοπή Ευθύγραμμου Τμήματος Liang – Barsky 3Δ

- Το ευθύγραμμο τμήμα προς αποκοπή δίνεται με τις κορυφές του \mathbf{p}_1 και \mathbf{p}_2 (όπως πριν)
- Στην ορθογραφική ή παράλληλη προβολή, το αντικείμενο αποκοπής είναι ένας κύβος και στις 3Δ προστίθεται η ανισότητα για την z συντεταγμένη: $z_{min} \leq z_1 + tDz \leq z_{max}$
- Στην προοπτική προβολή (ομογενείς συντεταγμένες), χρησιμοποιούνται οι ανισότητες (2) που ορίζουν το τμήμα του παραμετρικού ευθύγραμμου τμήματος εντός του αντικειμένου αποκοπής:
 - $(w_1 + tDw) \leq x_1 + tDx \leq w_1 + tDw$
 - $(w_1 + tDw) \leq y_1 + tDy \leq w_1 + tDw$
 - $(w_1 + tDw) \leq z_1 + tDz \leq w_1 + tDw$όπου $Dx = x_2 - x_1, Dy = y_2 - y_1, Dz = z_2 - z_1$

Αλγόριθμοι Αποκοπής στις 3Δ (5)

- Αυτές οι ανισότητες έχουν την κοινή μορφή $tp_i \leq q_i$ για $i=1,2,\dots,6$ όπου:

$$p_1 = -Dx - Dw \qquad q_1 = x_1 + w_1$$

$$p_2 = Dx - Dw \qquad q_2 = w_1 - x_1$$

$$p_3 = -Dy - Dw \qquad q_3 = y_1 + w_1$$

$$p_4 = Dy - Dw \qquad q_4 = w_1 - y_1$$

$$p_5 = -Dz - Dw \qquad q_5 = z_1 + w_1$$

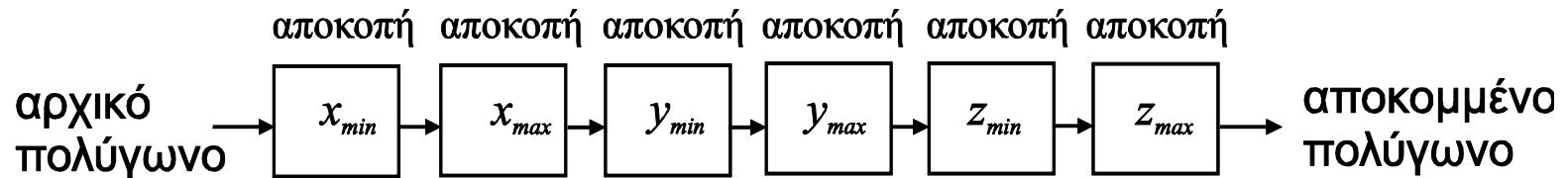
$$p_6 = Dz - Dw \qquad q_6 = w_1 - z_1.$$

- Οι λόγοι (q_i / p_i) αντιστοιχούν στις παραμετρικές τιμές της τομής του ευθύγραμμου τμήματος με το επίπεδο αποκοπής i
- Ο υπόλοιπος αλγόριθμος παραμένει όπως στις 2Δ

Αλγόριθμοι Αποκοπής στις 3Δ (6)

Αποκοπή Πολυγώνου Sutherland – Hodgman 3Δ

- Στις 3Δ το αντικείμενο αποκοπής είναι το στερεό παρατήρησης, αντί ενός κυρτού πολυγώνου
- Ο αλγόριθμος αποτελείται από 6 αλυσιδωτά στάδια, ένα για κάθε όψη του στερεού παρατήρησης

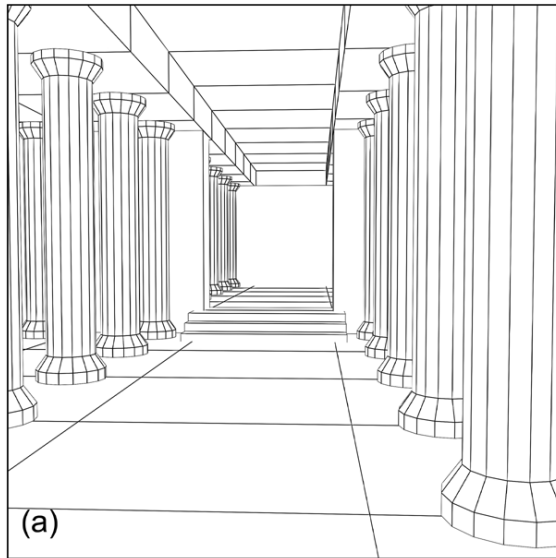


- Διαφορές από τον 2Δ αλγόριθμο:
 - Η υπορουτίνα `inside_test` τροποποιείται ώστε να ελέγχει αν ένα σημείο βρίσκεται εντός επιπέδου \rightarrow ισοδύναμο με έλεγχο προσήμου εξίσωσης του επιπέδου για τις συντεταγμένες του σημείου
 - Η υπορουτίνα `intersect_lines` πρέπει να αντικατασταθεί από την `intersect_plane_line` για τον υπολογισμό τομής ακμής πολυγώνου με επίπεδο του όγκου αποκοπής \rightarrow οι εξισώσεις (3) χρησιμοποιούνται για την προοπτική προβολή

Περικοπή Παρεμποδιζομένων

- Απόρριψη στοιχειωδών αντικειμένων που κρύβονται από άλλα στοιχειώδη αντικείμενα πλησιέστερα στον παρατηρητή
- **Σκοπός:** Αποδοτική απόρριψη μεγάλου πλήθους στοιχειωδών αντικειμένων πριν την εφαρμογή ακριβών αλγορίθμων ΑΚΕ
- *Σύνολο ορατών επιφανειών:* Υποσύνολο στοιχειωδών αντικειμένων που αποτυπώνονται σε τουλάχιστον ένα εικονοστοιχείο της εικόνας
- Υπολογισμός σφιχτού υπερσυνόλου του συνόλου ορατών επιφανειών ώστε τα υπόλοιπα στοιχειώδη αντικείμενα να περικοπούν
 - Αυτό το υπερσύνολο καλείται *δυναμικά ορατό σύνολο* (ΔΟΣ)
- Η περικοπή παρεμποδιζομένων δεν εξακριβώνει ποια τμήματα των στοιχειωδών αντικειμένων αποκρύπτονται → αλγόριθμοι ΑΚΕ
 - Αντίθετα εξακριβώνει ποια στοιχειώδη αντικείμενα ΔΕΝ είναι (εξ' ολοκλήρου) ορατά και τα απορρίπτει, υπολογίζοντας το ΔΟΣ

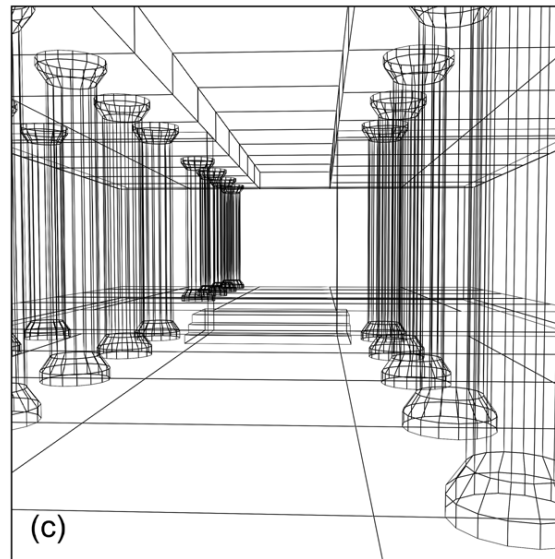
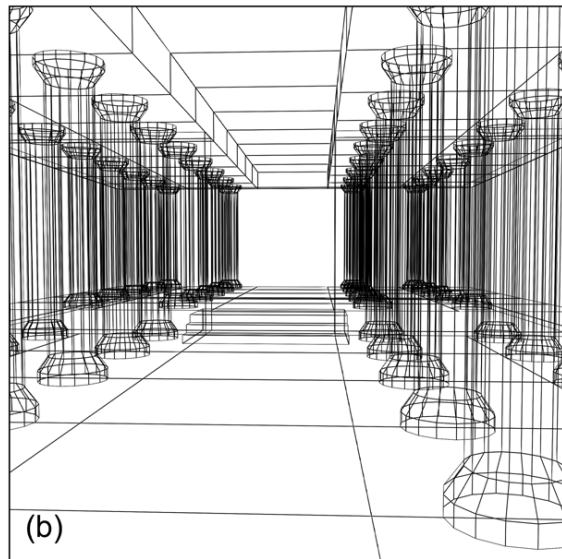
Περικοπή Παρεμποδιζομένων (2)



(a) σύνολο ορατών επιφανειών

(b) όλα τα στοιχειώδη αντικείμενα

(c) ΔΟΣ



Περικοπή Παρεμποδιζομένων (3)

- Το ΔΟΣ χρησιμοποιείται σαν είσοδος στους αλγόριθμους ΑΚΕ
- Η περικοπή παρεμποδιζομένων κοστίζει $O(P)$, όπου P το # στοιχειωδών αντικειμένων
- 2 κατηγορίες περικοπής παρεμποδιζομένων:
 - Περικοπή Παρεμποδιζομένων από-σημείο:
 - ◆ Επιλύει το πρόβλημα παρεμπόδισης για ένα σημείο παρατήρησης
 - ◆ Κατάλληλη για εξωτερικές σκηνές
 - Περικοπή Παρεμποδιζομένων από-περιοχή:
 - ◆ Επιλύει το πρόβλημα περικοπής για μια περιοχή του χώρου
 - ◆ Κατάλληλη για πυκνές εσωτερικές σκηνές
 - ◆ Κατάλληλη για στατικές σκηνές εξαιτίας των υπολογισμών προεπεξεργασίας

Περικοπή Παρεμποδιζομένων από-Περιοχή

- Πολλές εφαρμογές αποτελείται από σύνολο κυρτών περιοχών (κελιά), συνδεδεμένα με πύλες
 - Απλή μορφή: σκηνή αποτελείται από 2Δ σχέδιο δαπέδου, τα κελιά και οι πύλες είναι παράλληλα με τον x και τον y άξονα
- Τα στοιχειώδη αντικείμενα είναι ορατά μεταξύ κελιών μόνο μέσω των πυλών
 - Το κελί c_a μπορεί να είναι ορατό από το κελί c_b μέσω του c_m , αν υπάρχουν ευθείες ορατότητας που ενώνουν κατάλληλα τις πύλες τους
- Ο αλγόριθμος απαιτεί ένα βήμα προ-επεξεργασίας
 - Εκτελείται μόνο μια φορά αν τα κελιά και οι πύλες είναι στατικά
 - Κατασκευάζει τον πίνακα ΔΟΣ & ένα δέντρο BSP
- Πίνακας ΔΟΣ:
 - Δίνει το ΔΟΣ για κάθε κελί που μπορεί να βρίσκεται ο παρατηρητής
 - Η ορατότητα είναι συμμετρική \rightarrow ο πίνακας ΔΟΣ είναι συμμετρικός

Περικοπή Παρεμποδιζομένων από-Περιοχή (2)

- Κατασκευάζεται, ξεκινώντας από κάθε κελί c & αναδρομικά επισκέπτεται όλα τα προσβάσιμα κελιά από τον γράφο γειτνίασης, όσο υπάρχουν ευθείες ορατότητας που επιτρέπουν την ορατότητα από το c
- Έτσι κατασκευάζεται το δέντρο ορατότητας (*stab tree*) του c , που ορίζει το ΔΟΣ

• Δέντρο BSP:

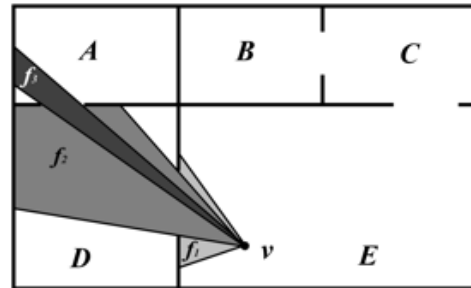
- Χρησιμοποιεί διαχωριστικά επίπεδα, που μπορεί να είναι όρια κελιών, για να διαμερίσει αναδρομικά την σκηνή
- Τα φύλλα αναπαριστούν τα κελιά
- Ένα ισορροπημένο δέντρο BSP χρησιμοποιείται για τον εντοπισμό του κελιού που περιέχει ένα σημείο, σε χρόνο $O(\log_2 n_c)$, όπου n_c το # των κελιών

• Κατά την εκτέλεση, τα βήματα που οδηγούν στη δημιουργία εικόνας του ΔΟΣ για ένα σημείο παρατήρησης \mathbf{v} είναι:

- 1. Προσδιορισμός κελιού c του \mathbf{v} χρησιμοποιώντας το δέντρο BSP
- 2. Προσδιορισμός του ΔΟΣ του c χρησιμοποιώντας τον πίνακα ΔΟΣ
- 3. Αποτύπωση του ΔΟΣ στην οθόνη

Περικοπή Παρεμποδιζομένων από-Περιοχή (3)

(a) Σκηνή μοντελοποιημένη με κελιά και πύλες



(a)



(b)

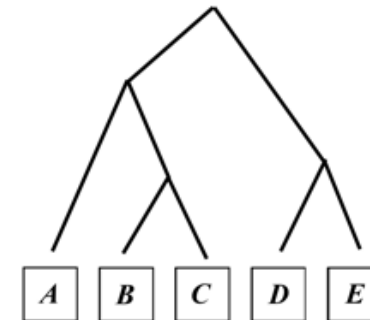
(b) Δέντρα τομής (stab trees) κελιών

	A	B	C	D	E
A	1			1	1
B		1	1		1
C		1	1	1	1
D	1		1	1	1
E	1	1	1	1	1

(c)

(c) Πίνακας ΔΟΣ

(d) δέντρο BSP



(d)

- Το ΔΟΣ δεν αλλάζει όσο το v παραμένει στο ίδιο κελί
- Τα πρώτα 2 βήματα εκτελούνται μόνο όταν το v περνά το όριο ενός κελιού
- Κατά την εκτέλεση, χρησιμοποιούνται μόνο το δέντρο BSP και ο πίνακας ΔΟΣ

Περικοπή Παρεμποδιζομένων από-Περιοχή (4)

- Η περικοπή παρεμποδιζομένων μπορεί να βελτιστοποιηθεί αν συνδυαστεί με περικοπή στο οπτικό πεδίο και πίσω όψεων
 - Η δημιουργία εικόνας μπορεί να περιοριστεί σε στοιχειώδη αντικείμενα που βρίσκονται ταυτόχρονα εντός του οπτικού πεδίου και του ΔΟΣ
 - Το στερεό παρατήρησης δημιουργείται αναδρομικά και μικραίνει από κελί σε κελί του δέντρου ορατότητας
- Ψευδοκώδικας:

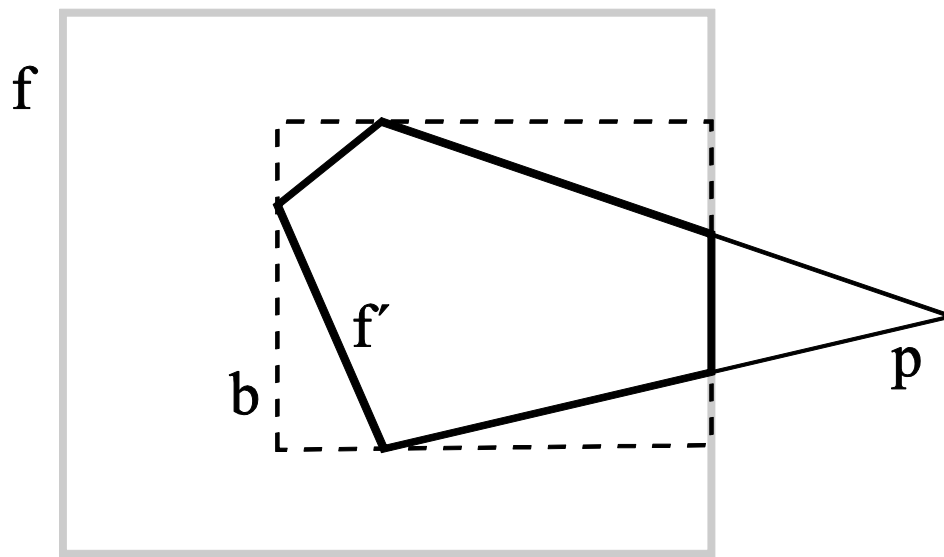
```
main()  
{  
    προσδιορισμός κελιού c του σημείου παρατήρησης με χρήση δέντρου BSP;  
    προσδιορισμός του ΔΟΣ κελιού c με χρήση πίνακα ΔΟΣ;  
    f = αρχικό οπτικό πεδίο;  
    portal render(c, f, PVS); /* PVS = ΔΟΣ*/  
}
```

Περικοπή Παρεμποδιζομένων από-Περιοχή (5)

```
portal render(cell c, frustum f, list PVS);
{
  for κάθε πολύγωνο R στο c {
    if ((R είναι πύλη) & (c' ∈ ΔΟΣ)) {
      /* η πύλη R οδηγεί στο κελί c' */
      /* υπολογισμός νέου οπτικού πεδίου f' */
      f' = clip frustum(f, R);
      if (f' <> empty) portal render(c', f', PVS);
    }
    else if (R είναι πύλη) {}
    else { /* R δεν είναι πύλη*/
      /* εφαρμογή περικοπής πίσω όψεων*/
      if (!back face(R)) {
        /* εφαρμογή περικοπής στο οπτικό πεδίο*/
        R' = clip poly(f, R);
        if (R' <> empty) render(R' );
      }
    }
  }
}
```

Περικοπή Παρεμποδιζομένων από-Περιοχή (6)

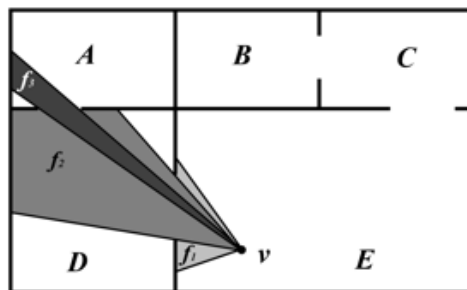
- Η εντολή $f' = \text{clip frustum}(f, R)$ υπολογίζει την τομή του τρέχοντος οπτικού πεδίου f & του όγκου που σχηματίζεται από το σημείο παρατήρησης & του πολυγώνου πύλης R
 - Αν παράγονται περίεργα κυρτά σχήματα, μπορεί να μην υπάρξει υποστήριξη από το υλικό
 - Λύση: Αντικατάσταση του f' από το περιβάλλον κιβώτιό του



Περικοπή Παρεμποδιζομένων από-Περιοχή (7)

Παράδειγμα:

- Προσδιορίζεται το κελί (E), στο οποίο βρίσκεται ο παρατηρητής v , με BSP
- Τα αντικείμενα του κελιού περικόπτονται ως προς το αρχικό οπτικό πεδίο f_1
- Η πρώτη πύλη, που οδηγεί στο κελί D του ΔΟΣ, περιορίζει το οπτικό πεδίο στο f_2
- Τα αντικείμενα του κελιού D περικόπτονται ως προς το f_2
- Η δεύτερη πύλη που οδηγεί στο κελί A περιορίζει το οπτικό πεδίο στο f_3
- Τα αντικείμενα του κελιού A περικόπτονται ως προς το f_3
- Η αναδρομή σταματά εδώ \rightarrow δεν υπάρχουν άλλα πολύγωνα πύλες στο f_3



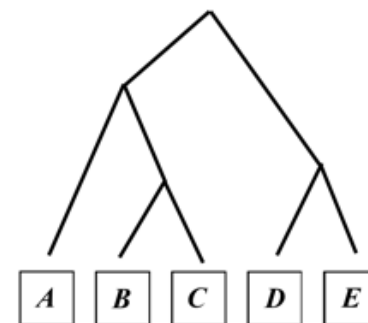
(a)



(b)

	A	B	C	D	E
A	1			1	1
B		1	1		1
C		1	1	1	1
D	1		1	1	1
E	1	1	1	1	1

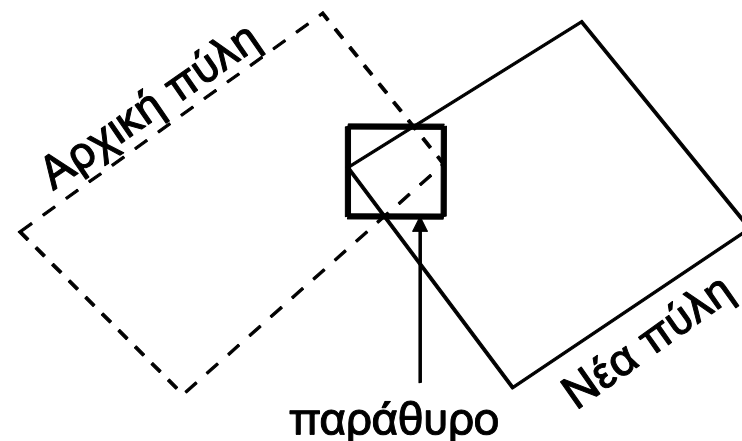
(c)



(d)

Περικοπή Παρεμποδιζομένων από-Σημείο

- Luebke: Εσωτερικές σκηνές με κελιά και πύλες
- Η εικόνα δημιουργείται ξεκινώντας από το τρέχον κελί
- Τα στοιχειώδη αντικείμενα πρέπει να είναι ορατά μέσω της προβολής των πυλών, αν αυτές βρίσκονται εντός των ορίων αποκοπής
- Αναδρομικές κλήσεις για κελιά στα οποία οδηγούν οι πύλες
- Σε κάθε βήμα, οι νέες πύλες τέμνονται με τις παλιές μέχρι να μην απομείνει τίποτα
- Υπολογίζεται μια υπερεκτίμηση (παράθυρο) της τομής των πυλών, ώστε να μειωθεί η πολυπλοκότητα

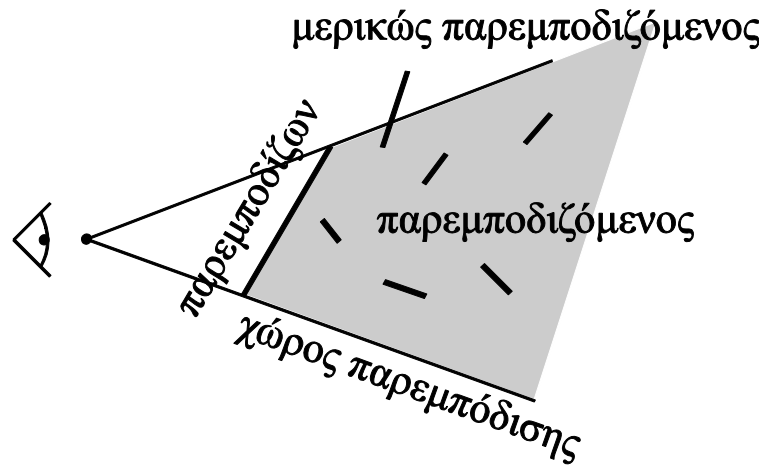


Περικοπή Παρεμποδιζομένων από-Σημείο (2)

- Δεν μπορούμε να υποθέσουμε την ύπαρξη κελιών & πυλών για εξωτερικές σκηνές
 - Οι διαμερισμένες περιοχές τέτοιων σκηνών δεν θα είχαν συνοχή σχετικά με τα χαρακτηριστικά παρεμπόδισης
- Η περικοπή παρεμποδιζομένων από-σημείο επιλύει το πρόβλημα για ένα σημείο παρατήρησης \rightarrow το ΔΟΣ δεν υπολογίζεται, άρα δεν απαιτείται πολλή προ-επεξεργασία εν αντιθέσει με την περικοπή παρεμποδιζομένων από-περιοχή
- Η ιδέα πίσω από αυτή την τεχνική είναι ο *παρεμποδιστής*:
 - Ο παρεμποδιστής είναι ένα στοιχειώδες αντικείμενο, ή συνδυασμός στοιχειωδών αντικειμένων, που παρεμποδίζουν μεγάλο πλήθος άλλων στοιχειωδών αντικειμένων (*παρεμποδιζόμενοι*) ως προς συγκεκριμένο σημείο παρατήρησης
 - Η περιοχή που ορίζεται από το σημείο παρατήρησης και τον παρεμποδιστή είναι ο *χώρος παρεμπόδισης*

Περικοπή Παρεμποδιζομένων από-Σημείο (3)

- Τα στοιχειώδη αντικείμενα εντός του χώρου παρεμπόδισης περικόπτονται
- Τα μερικώς παρεμποδιζόμενα αντικείμενα προωθούνται στους αλγορίθμους ΑΚΕ



- Δύο βήματα απαιτούνται για την περικοπή παρεμποδιζομένων ως προς σημείο παρατήρησης \mathbf{v} :
 - Δημιουργία συνόλου καλών παρεμποδιστών για το \mathbf{v}
 - Περικοπή παρεμποδιζομένων χρησιμοποιώντας τους παρεμποδιστές

Περικοπή Παρεμποδιζομένων από-Σημείο (4)

- Επίπεδοι παρεμποδιστές, ταξινομημένοι ανάλογα με την επιφάνεια προβολής τους, προτείνονται από τους Coorg & Teller:

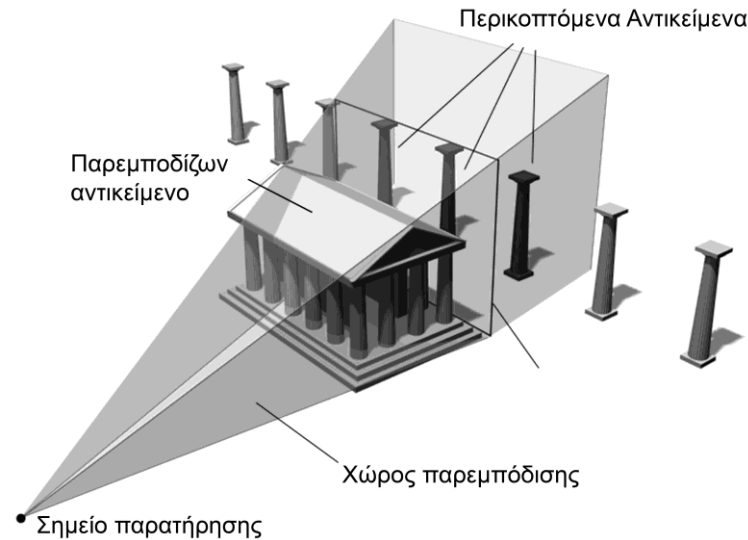
- Η συνάρτηση ταξινόμησης:

$$f_{\text{επίπεδο}} = \frac{-A(\hat{\mathbf{n}} \cdot \hat{\mathbf{v}})}{|\vec{\mathbf{v}}|^2}$$

όπου το A η επιφάνεια του επίπεδου παρεμποδιστή, $\hat{\mathbf{n}}$ το μοναδιαίο κανονικό του διάνυσμα και $\hat{\mathbf{v}}$ το διάνυσμα από το σημείο παρατήρησης προς το κέντρο του επίπεδου παρεμποδιστή

- Ένα σύνηθες είδος επιπέδου παρεμποδιστή είναι ο *αντιπρόσωπος* ενός αντικειμένου
- Ο αντιπρόσωπος είναι ένα κυρτό πολύγωνο, κάθετο στην κατεύθυνση παρατήρησης, εγγεγραμμένο εντός του χώρου παρεμπόδισης του παρεμποδιστή

Περικοπή Παρεμποδιζομένων από-Σημείο (5)

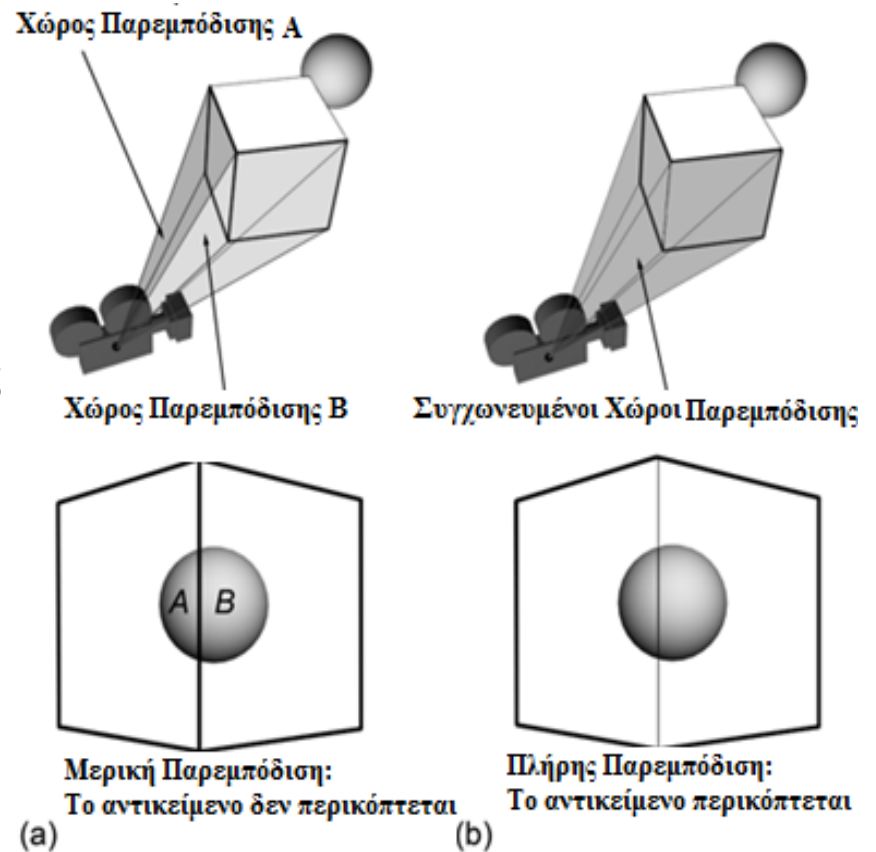


- Η περικοπή παρεμποδιζομένων επιταχύνεται αν υπάρχει μια ιεραρχική περιγραφή της σκηνής με περιβάλλοντες όγκους
- Ξεκινώντας από το υψηλότερο επίπεδο, ένας περιβάλλον όγκος που βρίσκεται εξ' ολοκλήρου εντός/εκτός του χώρου παρεμπόδισης αγνοείται/αποτυπώνεται στην οθόνη
- Ένας περιβάλλον όγκος που τέμνει τον χώρο παρεμπόδισης, διασπάται στο επόμενο επίπεδο

Περικοπή Παρεμποδιζομένων από-Σημείο (6)

- Η περικοπή παρεμποδιζομένων πάσχει από το πρόβλημα της *μερικής παρεμπόδισης*:

- Ένα αντικείμενο μπορεί να μην βρίσκεται στον χώρο παρεμπόδισης κανενός στοιχειώδους αντικειμένου → δεν περικόπτεται, παρόλο που μπορεί να βρίσκεται στον χώρο παρεμπόδισης κάποιου συνδυασμού γειτονικών στοιχειωδών αντικειμένων



- Γι αυτό, έχουν αναπτυχθεί αλγόριθμοι που συγχωνεύουν χώρους παρεμπόδισης :

- Ο Παπαϊωάννου πρότεινε μια επέκταση του επίπεδου παρεμποδιστή, τον *στερεό παρεμποδιστή*, που διευθετεί το πρόβλημα της μερικής παρεμπόδισης δημιουργώντας έναν επίπεδο παρεμποδιστή για ολόκληρο τον όγκο ενός αντικειμένου

Απομάκρυνση Κρυμμένων Επιφανειών

- Οι αλγόριθμοι ΑΚΕ δίνουν την τελική λύση στο πρόβλημα της παρεμπόδισης (ορατότητας)
- Τα ορατά στοιχειώδη αντικείμενα (ή τμήματά τους) πρέπει να σχεδιάζονται απευθείας
- Οι αλγόριθμοι ΑΚΕ ταξινομούν τα αντικείμενα που τέμνονται από τις ακτίνες προβολής
- Αυτή η ταξινόμηση καταλήγει στη σύγκριση 2 σημείων:

$$\mathbf{p}_1=[x_1, y_1, z_1]^T \text{ and } \mathbf{p}_2=[x_2, y_2, z_2]^T$$

- Αν είναι πάνω στην ίδια ακτίνα τότε αποτελούν ένα ζεύγος παρεμπόδισης
→ το πλησιέστερο παρεμποδίζει το άλλο

- 2 περιπτώσεις:

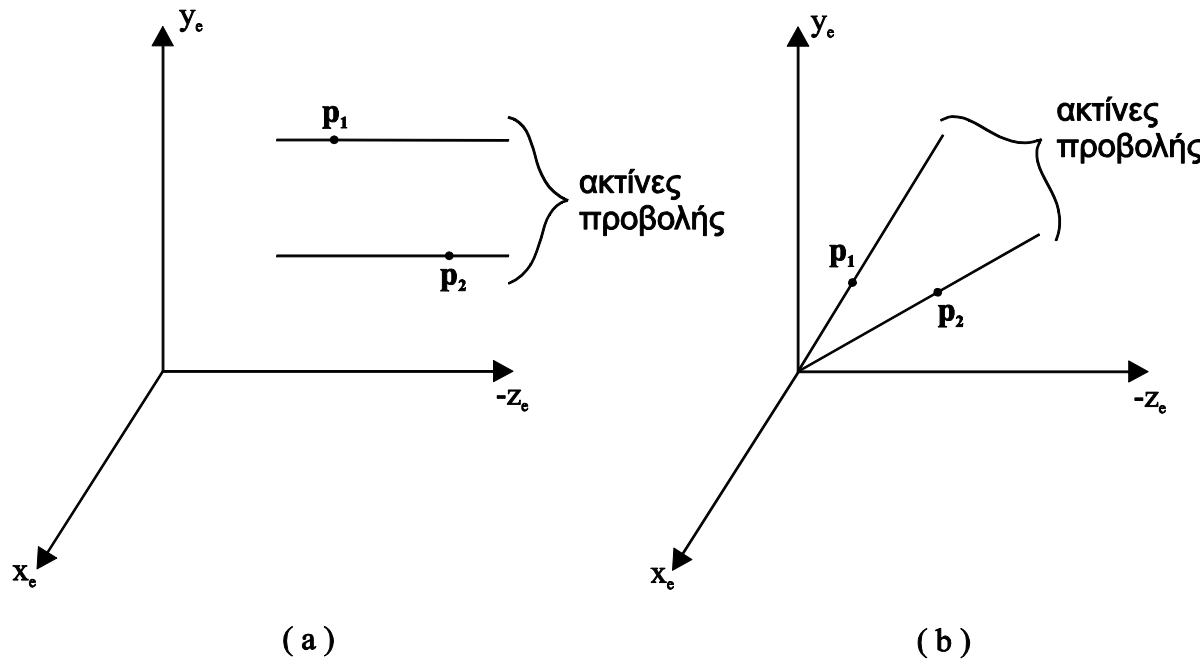
- Ορθογραφική προβολή: αν οι ακτίνες προβολής είναι παράλληλες στο z , τότε 2 σημεία είναι ζεύγος παρεμπόδισης αν:

$$(x_1 = x_2) \text{ και } (y_1 = y_2)$$

Απομάκρυνση Κρυσμένων Επιφανειών (2)

- Προοπτική προβολή: πρέπει να γίνει η προοπτική διαίρεση για να βρεθεί αν 2 σημεία είναι ζεύγος παρεμπόδισης. Η συνθήκη είναι:

$$(x_1 / z_1 = x_2 / z_2) \text{ και } (y_1 / z_1 = y_2 / z_2)$$



- Στην προοπτική προβολή, η ακριβή προοπτική διαίρεση γίνεται έτσι κι' αλλιώς κατά την μετάβαση από ΣΣΠ σε ΚΧΟ, εντός του μετασχηματισμού παρατήρησης

Απομάκρυνση Κρυμμένων Επιφανειών (3)

- Μετασχηματίζει το προοπτικό στερεό παρατήρησης σε ορθογώνιο παραλληλεπίπεδο \rightarrow γίνονται συγκρίσεις των x και y συντεταγμένων για τον καθορισμό των ζευγών παρεμπόδισης
- Για αυτό το λόγο η ΑΚΕ γίνεται μετά το μετασχηματισμό παρατήρησης στον ΚΧΟ
- Διατήρηση της z συντεταγμένης για την ΑΚΕ, κατά τον μετασχηματισμό παρατήρησης
- Οι περισσότεροι αλγόριθμοι ΑΚΕ εκμεταλλεύονται τη *συνάφεια* \rightarrow υπολογισμοί τομής αντικαθίστανται με αυξητικούς υπολογισμούς:
 - Συνάφεια επιφάνειας
 - Συνάφεια αντικειμένων
 - Συνάφεια γραμμών σάρωσης
 - Συνάφεια ακμής
 - Συνάφεια μεταξύ καρέ

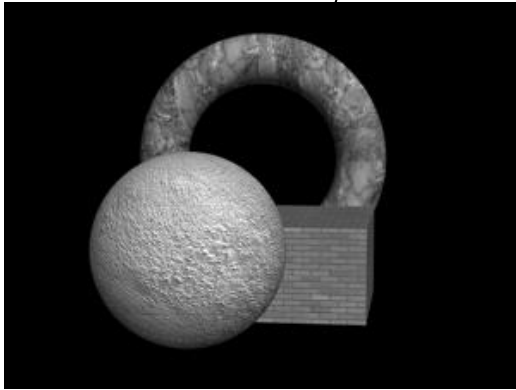
Αλγόριθμος Z-buffer

- Αρχικά απορρίφθηκε λόγω της υψηλής απαίτησης σε μνήμη
- Σήμερα είναι υλοποιημένος σε κάθε κάρτα γραφικών
- Διατηρεί 2Δ μνήμη τιμών βάθους (Z-buffer) με την ίδια χωρική ανάλυση με τον καταχωρητή εικόνας
- 1:1 αντιστοιχία μεταξύ των στοιχείων των καταχωρητών εικόνας & βάθους
- Κάθε στοιχείο του καταχωρητή βάθους φυλάσσει το ελάχιστο βάθος μέχρι στιγμής για το αντίστοιχο εικονοστοιχείο του καταχωρητή εικόνας
- Ο καταχωρητής βάθους αρχικοποιείται στη μέγιστη τιμή βάθους, συνήθως στο όπισθεν επίπεδο αποκοπής
- Κατά τη σχεδίαση υπολογίζονται για κάθε στοιχειώδες αντικείμενο τα (z_p, c_p) για κάθε εικονοστοιχείο $p = (x_p, y_p)$, όπου z_p το βάθος του αντικειμένου στο p , και c_p το χρώμα του στο p

Αλγόριθμος Z-buffer (2)

Καταχωρητής

Εικόνας

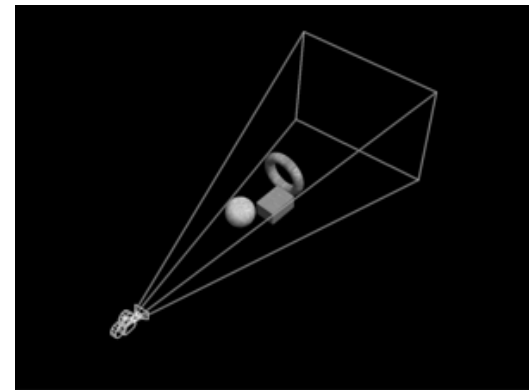


Καταχωρητής

Βάθους



3Δ σκηνή



- Αν οι τιμές βάθους μειώνονται καθώς απομακρυνόμαστε από το σημείο παρατήρησης, ο κύριος έλεγχος του Z-buffer είναι:

```
if (z-buffer[xp, yp] < zp)
```

```
{
```

```
  f-buffer[xp, yp] = cp; /* ενημέρωση καταχωρητή εικόνας */
```

```
  z-buffer[xp, yp] = zp; /* ενημέρωση καταχωρητή βάθους */
```

```
}
```

Αλγόριθμος Z-buffer (3)

- Για τον υπολογισμό του βάθους z_p σε κάθε εικονοστοιχείο που καλύπτει ένα αντικείμενο, τίθεται θέμα απόδοσης:
 - Ο υπολογισμός τομής του αντικειμένου με την ακτίνα που ορίζεται από το σημείο παρατήρησης και το εικονοστοιχείο είναι ακριβός
 - Εκμεταλλευόμαστε τη συνάρτηση επιφάνειας. Έστω η εξίσωση επιπέδου του στοιχειώδους αντικειμένου:

$$F(x, y, z) = ax + by + cz + d = 0$$

Για το βάθος παίρνουμε:

$$F'(x, y) = z = -d/c - (a/c)x - (b/c)y$$

F' υπολογίζεται αυξητικά από το εικονοστοιχείο (x, y) στο $(x+1, y)$ αφού:

$$F'(x+1, y) - F'(x, y) = -a/c$$

- Προσθέτοντας την σταθερή πρώτη διαφορά της F' στο x ή y , υπολογίζεται το βάθος από εικονοστοιχείο σε εικονοστοιχείο με κόστος μιας πρόσθεσης

Αλγόριθμος Z-buffer (4)

- Στην πράξη, οι τιμές βάθους στις κορυφές ενός επίπεδου στοιχειώδους αντικειμένου παρεμβάλλονται κατά μήκος των ακμών και των γραμμών σάρωσης
- Το ίδιο συμβαίνει και για το χρωματικό διάνυσμα
- Η πολυπλοκότητα του αλγορίθμου Z-buffer είναι $O(Ps)$, όπου P # των στοιχειωδών αντικειμένων και s ο μέσος # των εικονοστοιχείων που καλύπτει ένα αντικείμενο
- Στην πράξη, όσο αυξάνει το P , το s μειώνεται αναλογικά $\rightarrow O(p)$, όπου p το # των εικονοστοιχείων
- Πλεονεκτήματα αλγορίθμου Z-buffer :
 - Απλός
 - Σταθερή απόδοση
- Μειονεκτήματα αλγορίθμου Z-buffer :
 - Δυσκολία στην αντιμετώπιση κάποιων ειδικών εφέ (π.χ. διαφάνεια)

Αλγόριθμος Z-buffer (5)

- Το αποτέλεσμα έχει σταθερή ανάλυση, που οφείλεται στη φύση του χώρου εικόνας
- *Z-fighting*: προβλήματα αριθμητικής σύγκρισης βάθους, αν τα όρια αποκοπής ορίζουν μεγάλη διάσταση βάθους

• Ο καταχωρητής βάθους που υπολογίζεται κατά τη σχεδίαση, μπορεί να χρησιμοποιηθεί με πολλούς τρόπους:

- Επιτρέπει τη συγχώνευση βάθους 2 ή περισσότερων εικόνων. Έστω (f_a, z_a) και (f_b, z_b) οι καταχωρητές εικόνας και βάθους 2 τμημάτων μιας σκηνής. Μπορούν να συγχωνευθούν με την επιλογή του τμήματος με την μικρότερη τιμή βάθους σε κάθε εικονοστοιχείο:

```
for (x=0; x<XRES; x++) {  
    for (y=0; y<YRES; y++) {  
        Fc[x, y] = (Za[x, y]>Zb[x, y])? Fa[x, y]:Fb[x, y];  
        Zc[x, y] = (Za[x, y]>Zb[x, y])? Za[x, y]:Zb[x, y];  
    }  
}
```

Αλγόριθμος Z-buffer (6)

- Πολλοί άλλοι υπολογισμοί μπορούν να γίνουν με τον Z-buffer:
 - Προσδιορισμός σκιών
 - Μετατροπή επιφανειών σε ογκοστοιχεία
 - Υπολογισμοί Voronoi
 - Ανακατασκευή 3D αντικειμένων
 - Ανίχνευση συμμετρίας
 - Ανάκτηση 3D αντικειμένων

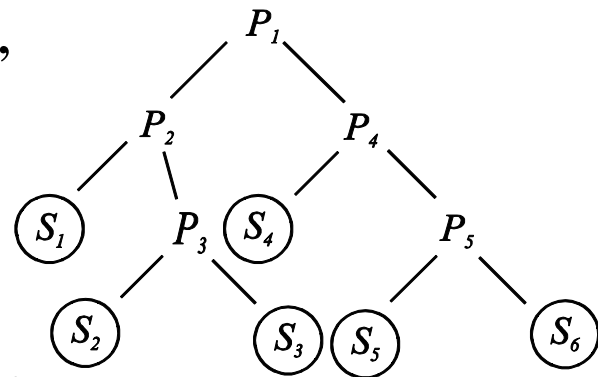
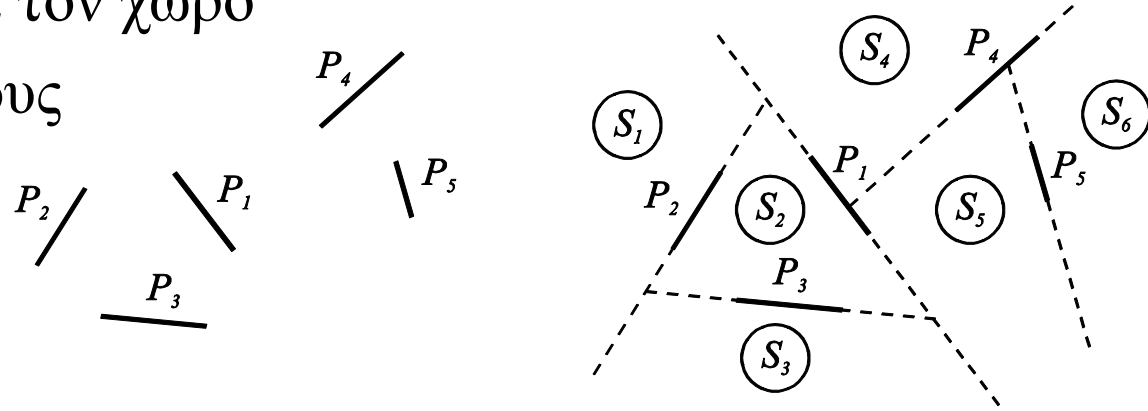
Αλγόριθμος BSP

- Η δυαδική διαμέριση χώρου (BSP) είναι ένας αλγόριθμος χώρου αντικειμένων που χρησιμοποιεί ένα δυαδικό δέντρο για να υποδιαιρεί αναδρομικά τον χώρο

- Τα δεδομένα στους κόμβους αντιπροσωπεύουν τα πολύγωνα της σκηνής

- Οι εσωτερικοί κόμβοι διαμερίζουν το χώρο σε σχέση (a)

με το επίπεδο του πολυγώνου τους, ώστε οι απόγονοι που είναι στο αριστερό υποδέντρο να είναι στη μια μεριά του επιπέδου και εκείνοι που είναι στο δεξί να είναι στην άλλη



(c)

Αλγόριθμος BSP (2)

- Αλγόριθμος για την κατασκευή δέντρου BSP:

```
BuildBSP(BSPnode, polygonDB);
```

```
{
```

```
    Επιλογή πολυγώνου (επιπέδου)  $P_i$  από polygonDB;
```

```
    Ανάθεση  $P_i$  στο BSPnode;
```

```
    /* Διαμερισμός των πολυγώνων της σκηνής ανάλογα με τη μεριά που βρίσκονται σε  
    σχέση με το επίπεδο  $P_i$ , διαίρεση των πολυγώνων που τέμνουν το  $P_i$  */
```

```
    Partition( $P_i$ , polygonDB, polygonDBL, polygonDBR);
```

```
    if (polygonDBL != empty) BuildBSP(BSPnode->Left, polygonDBL);
```

```
    if (polygonDBR != empty) BuildBSP(BSPnode->Right, polygonDBR);
```

```
}
```

- Σημαντική η επιλογή του επιπέδου διαμερισμού P_i , ώστε να δημιουργηθεί ένα ισοσταθμισμένο δέντρο BSP
- Επιλέγεται ένα επίπεδο, που διαιρεί τη σκηνή σε 2 τμήματα που κατά προσέγγιση έχουν την ίδια πληθικότητα

Αλγόριθμος BSP (3)

- Κατά τη διαμέριση, πολύγωνα που τέμνουν το επίπεδο διαιρούνται → επιτυγχάνεται με την επέκταση ενός αλγορίθμου αποκοπής ώστε να δίνει τόσο τα “εσωτερικά” όσο και τα “εξωτερικά” τμήματα του πολυγώνου
- Δέντρα BSP χρησιμοποιούνται για την εμφάνιση μιας σκηνής χωρίς τις κρυμμένες επιφάνειες
- Για ένα σημείο παρατήρησης v και έναν κόμβο του δέντρου BSP, όλα τα πολύγωνα που βρίσκονται στην ίδια διαμέριση με το v δεν μπορούν να κρύβονται από πολύγωνα της άλλης διαμέρισης → τα πολύγωνα της άλλης διαμέρισης σχεδιάζονται πρώτα

Αλγόριθμος BSP (4)

- Ψευδοκώδικας:

```
DisplayBSP (BSPnode, v);
```

```
{
```

```
  if IsLeaf (BSPnode) Render (BSPnode->Polygon)
```

```
  else if (v στην 'άριστερή' διαμέριση του BSPnode->Polygon) {
```

```
    DisplayBSP (BSPnode->Right, v);
```

```
    Render (BSPnode->Polygon);
```

```
    DisplayBSP (BSPnode->Left, v);
```

```
  }
```

```
  else /* v στην 'δεξιά' διαμέριση του BSPnode->Polygon */ {
```

```
    DisplayBSP (BSPnode->Left, v);
```

```
    Render (BSPnode->Polygon);
```

```
    DisplayBSP (BSPnode->Right, v);
```

```
  }
```

```
}
```

Αλγόριθμος BSP (5)

- Ο αλγόριθμος `DisplayBSP` κοστίζει $O(P)$
- Ο αλγόριθμος `BuildBSP` κοστίζει $O(P^2)$
- Η συνολική πολυπλοκότητα του αλγορίθμου BSP είναι $O(P^2)$
- Για στατικές σκηνές, η `BuildBSP` δε χρησιμοποιείται συχνά. Για κάθε νέα θέση του σημείου παρατήρησης, μόνο η `DisplayBSP` χρησιμοποιείται \rightarrow ο αλγόριθμος BSP είναι κατάλληλος για στατικές σκηνές, αλλά ΟΧΙ κατάλληλος για δυναμικές σκηνές

Αλγόριθμος Ταξινόμησης Κατά Βάθος

- Ταξινομεί τα πολύγωνα ανάλογα με την απόστασή τους από τον παρατηρητή και τα σχεδιάζει με αντίστροφη σειρά (από πίσω προς τα εμπρός)
- Ονομάζεται επίσης *αλγόριθμος ζωγράφου*
- Για την ταξινόμηση χρησιμοποιούνται οι ελάχιστες τιμές βάθους:

DepthSort (polygonDB) ;

{

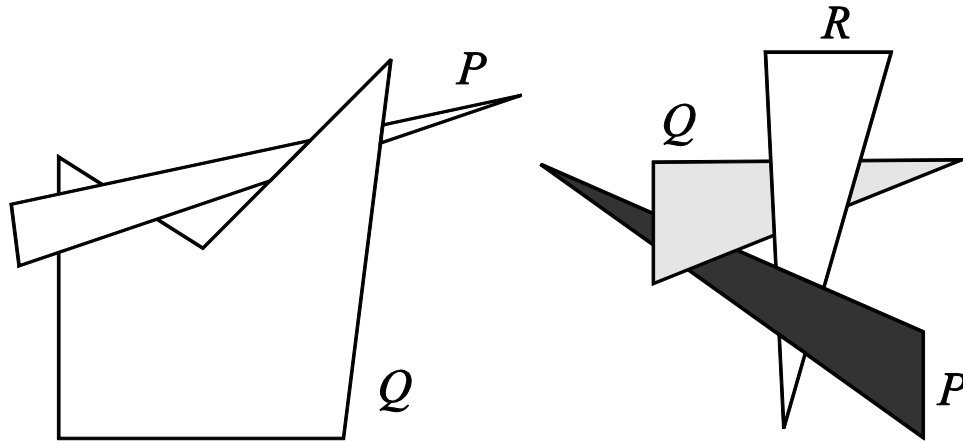
```
/* Ταξινόμησηση την polygonDB σύμφωνα με το ελάχιστο z */  
for κάθε πολύγωνο στο polygonDB βρες MINZ και MAXZ ;  
ταξινόμησηση το polygonDB σύμφωνα με το MINZ ;  
επίλυση επικαλύψεων στο z ;  
σχεδίασε τα πολύγωνα με τη σειρά της ταξινομημένης λίστας ;
```

}

- Επικαλύψεις στο z έχουμε όταν επικαλύπτονται οι εκτάσεις z των πολυγώνων \rightarrow δύσκολη ταξινόμηση αφού άγνωστο ποιο είναι πλησιέστερο \rightarrow μερικές φορές αδύνατη η ταξινόμηση

Αλγόριθμος Ταξινόμησης Κατά Βάθος (2)

- Πολύγωνα που δεν μπορούν να ταξινομηθούν:



- Όταν επικαλύπτονται οι z εκτάσεις δυο πολύγωνων R και Q , γίνονται μια σειρά ελέγχων αυξανόμενης πολυπλοκότητας για να βρεθεί η σωστή σειρά στη λίστα σχεδίασης
- Αν ένας από τους παρακάτω ελέγχους είναι θετικός τότε το Q δεν αποκρύπτεται από το R :
 1. Δεν επικαλύπτονται οι x -εκτάσεις των R και Q
 2. Δεν επικαλύπτονται οι y -εκτάσεις των R και Q

Αλγόριθμος Ταξινόμησης Κατά Βάθος (3)

3. Το R βρίσκεται ολόκληρο στον ημιχώρο του επιπέδου του Q στον οποίο **δεν** βρίσκεται το σημείο παρατήρησης \mathbf{v} . Αυτό διαπιστώνεται ελέγχοντας αν το πρόσημο της εξίσωσης επιπέδου του Q είναι ίδιο για όλες τις κορυφές του R και διαφορετικό από αυτό για το \mathbf{v} :

$$\text{sign}(f_Q(\mathbf{r}_i)) \neq \text{sign}(f_Q(\mathbf{v})), \quad \forall \mathbf{r}_i \in R$$

όπου

$$f_Q(x, y, z) = a_Q x + b_Q y + c_Q z + d_Q = 0$$

είναι η εξίσωση επιπέδου του πολύγωνου Q

4. Το Q βρίσκεται ολόκληρο στον ημιχώρο του επιπέδου του R στον οποίο βρίσκεται και το σημείο παρατήρησης \mathbf{v} . Αυτό διαπιστώνεται ελέγχοντας αν το πρόσημο της εξίσωσης επιπέδου του R είναι ίδιο για όλες τις κορυφές του Q καθώς και για το \mathbf{v} :

$$\text{sign}(f_R(\mathbf{q}_i)) = \text{sign}(f_R(\mathbf{v})), \quad \forall \mathbf{q}_i \in Q$$

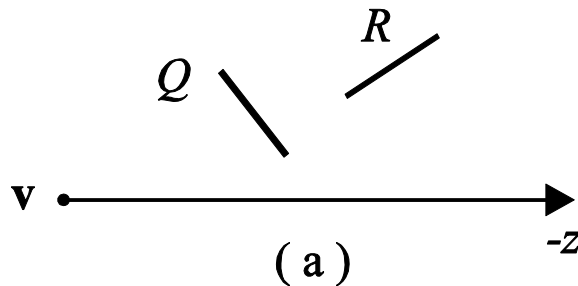
όπου

$$f_R(x, y, z) = a_R x + b_R y + c_R z + d_R = 0$$

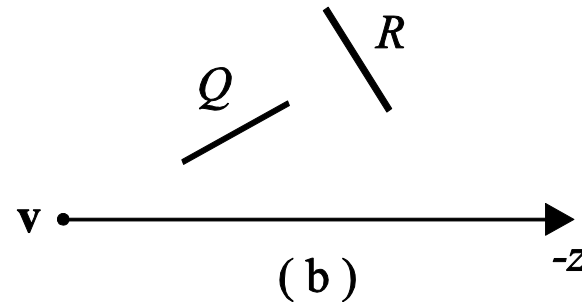
είναι η εξίσωση επιπέδου του πολύγωνου R

Αλγόριθμος Ταξινόμησης Κατά Βάθος (4)

(a) R πίσω από Q



(b) Q μπροστά από R



5. Δεν επικαλύπτονται οι προβολές των R και Q

- Αν κανένας από τους 5 ελέγχους δεν ισχύει, εναλλάσσουμε τα R και Q και επαναλαμβάνονται οι έλεγχοι 3, 4
- Οι έλεγχοι 1, 2, 5 είναι συμμετρικοί \rightarrow δε χρειάζονται επανάληψη
- Αν και πάλι δε βρεθεί η σωστή σειρά, το R χωρίζεται σε 2 πολύγωνα από το επίπεδο του Q , τα νέα πολύγωνα αντικαθιστούν το R και η παραπάνω διαδικασία επαναλαμβάνεται

Αλγόριθμος Ταξινόμησης Κατά Βάθος (5)

- Η ταξινόμηση κατά βάθος είναι αλγόριθμος χώρου αντικειμένων εκτός από το τελευταίο βήμα (χώρος εικόνας)
- Μια βελτιστοποίηση είναι να σχεδιάζονται τα πολύγωνα από εμπρος προς τα πίσω:
 - Διαδοχικά πολύγωνα σχεδιάζονται μόνο σε εικονοστοιχεία που δεν έχουν ήδη καλυφθεί από πλησιέστερα πολύγωνα
 - Η σχεδίαση σταματά όταν όλα τα εικονοστοιχεία της εικόνας έχουν καλυφθεί τουλάχιστον μια φορά
- Κόστος βήματος ταξινόμησης $O(P \log_2 P)$
- Η επίλυση των z επικαλύψεων κοστίζει (χειρότερη) $O(P^2)$
- Ο αλγόριθμος είναι αργός για πολύπλοκες σκηνές
- Θετικό:
 - Μπορεί να αντιμετωπίσει εύκολα τη διαφάνεια

Αλγόριθμος Εκπομπής Ακτίνων

- Για κάθε εικονοστοιχείο p παρακολουθείται μια ακτίνα
- Η ακτίνα καθορίζεται από το σημείο παρατήρησης v και το διάνυσμα $\vec{p} - \vec{v}$
- Υπολογισμός τομών με όλα τα στοιχειώδη αντικείμενα της σκηνής και η πλησιέστερη στο v ορίζει το ορατό αντικείμενο
- Βασικός αλγόριθμος:

```
RayCasting(primitiveDB, v); {  
  for κάθε εικονοστοιχείο p {  
    minp = MAXINT;  
    for κάθε αντικείμενο R στην primitiveDB {  
      /* υπολογισμός τομής της ακτίνας (v,p) με το R */  
      I = intersect_primitive_ray(R,v,p); /* MAXINT αν κανένα */  
      if (|i-v| < minp) {  
        p -> nearest_primitive = R;  
        minp = |i-v|  
      }  
    }  
  }  
}
```


Αλγόριθμος Εκπομπής Ακτίνων (2)

- Η εκπομπή ακτίνων δεν εκμεταλλεύεται τη συνάφεια $\rightarrow O(pP)$
- Πολύ γενική μέθοδος
- Μπορεί να επιταχυνθεί με παράλληλη επεξεργασία
- Ο αλγόριθμος εκπομπής ακτίνων εφαρμόζεται πριν ή μετά την προοπτική προβολή:
 - Στην 1^η περίπτωση, οι ακτίνες είναι οι ακτίνες προβολής
 - Στη 2^η περίπτωση, οι ακτίνες είναι παράλληλες μεταξύ τους και κάθετες στο επίπεδο προβολής
- Η εκπομπή ακτίνων κατατάσσεται είτε ως αλγόριθμος χώρου αντικειμένων είτε χώρου εικόνας

Περίληψη

- Στον παρακάτω πίνακα εμφανίζονται οι πολυπλοκότητες και ο χώρος λειτουργίας των αλγορίθμων ΑΚΕ που παρουσιάστηκαν:

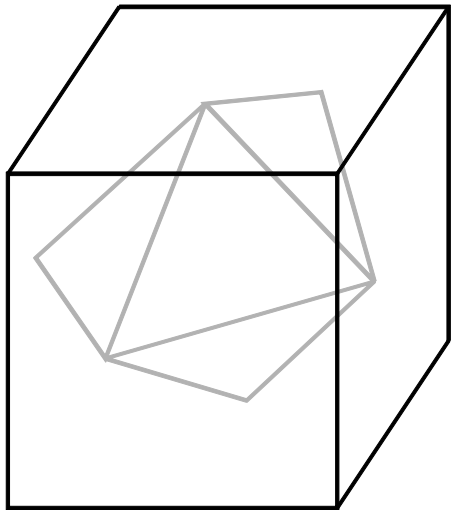
Αλγόριθμοι ΑΚΕ	Πολυπλοκότητα	Χώρος Λειτουργίας
Z-Buffer	$O(Ps) \simeq O(p)$	Εικόνας
BSP	$O(P^2)$	Αντικειμένου
Ταξινόμηση κατά βάθος	$O(P^2)$	Αντικειμένου
Εκπομπής Ακτίνων	$O(pP)$	Εικόνας/Αντικειμένου

Θέματα Απόδοσης

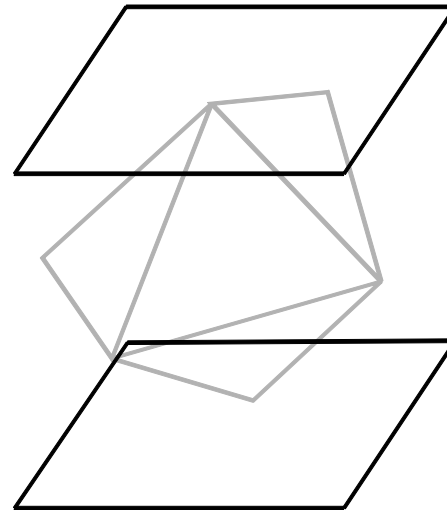
- Αλγόριθμοι Περικοπής, ΑΚΕ, Παρακολούθησης Ακτίνας και άλλοι, χρειάζονται υπολογισμούς τομής
- Οι υπολογισμοί τομής είναι ακριβοί
- Τεχνικές που επιταχύνουν τους υπολογισμούς τομής:
 - Περιβάλλοντες Όγκοι (Bounding volumes)
 - Υποδιαίρεση Χώρου

Περιβάλλοντες Όγκοι

- Ένας τρόπος να μειωθεί το κόστος υπολογισμού τομών με ένα πολύπλοκο μοντέλο, είναι να ομαδοποιήσουμε τα στοιχειώδη αντικείμενα σε ένα περιβάλλοντα όγκο
- Μπορεί π.χ. να είναι ένα ορθογώνιο παραλληλεπίπεδο, ή σφαίρα
- Μπορεί να μην είναι κλειστός



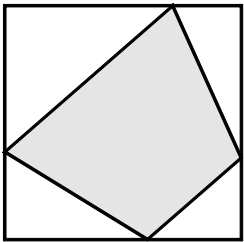
(a)



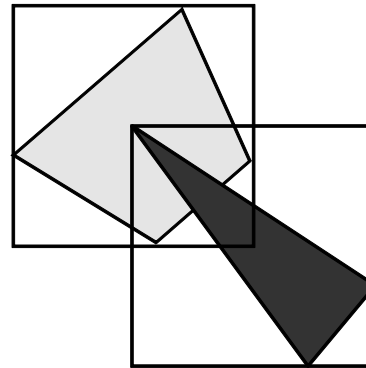
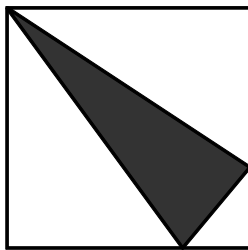
(b)

Περιβάλλοντες Όγκοι (2)

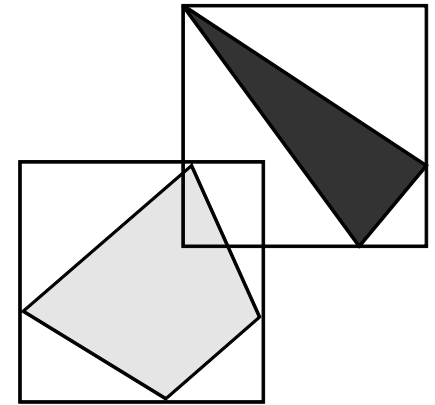
- Τομή με τον περιβάλλοντα όγκο **δεν συνεπάγεται απαραίτητα** τομή με το μοντέλο, αφού υπάρχει *κενός χώρος* μεταξύ του όγκου και του μοντέλου
- Αντίθετα, μη τομή με τον περιβάλλοντα όγκο **συνεπάγεται μη** τομή με το μοντέλο που περιέχει



(a)



(b)



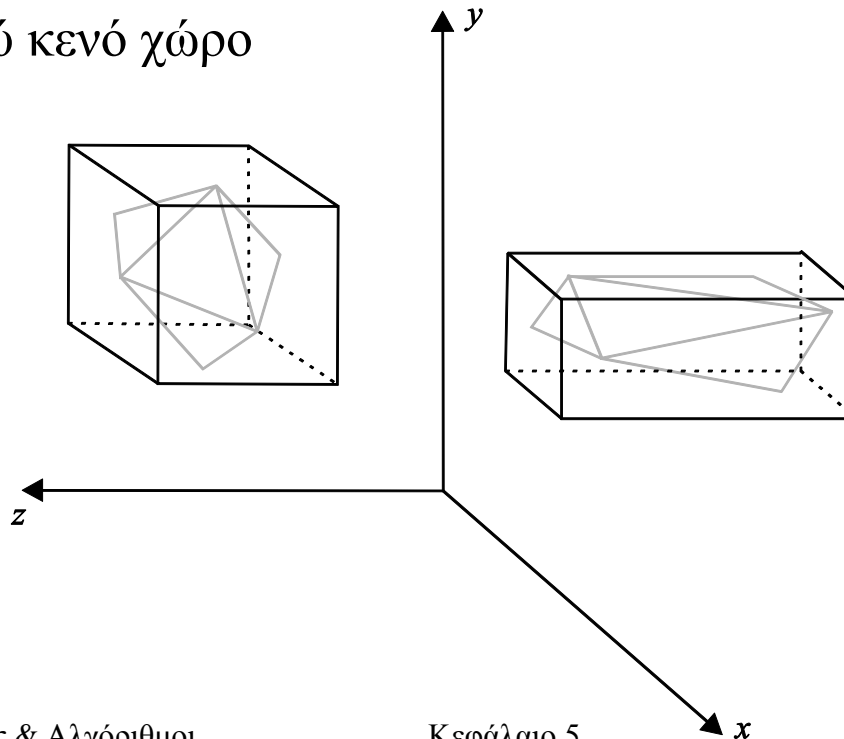
(c)

Περιβάλλοντες Όγκοι (3)

- Όταν δεν τέμνεται ο περιβάλλοντας όγκος, δεν γίνεται έλεγχος τομής με το μοντέλο → εξοικονόμηση υπολογισμών
- Ένας καλός περιβάλλον όγκος πρέπει:
 - Να είναι απλός
 - Να ελαχιστοποιεί τον κενό χώρο
- Με το πρώτο οι έλεγχοι τομής γίνονται αποτελεσματικοί
- Με το δεύτερο θα υπάρχουν κατά το δυνατόν λιγότερες λάθος προειδοποιήσεις
- Αυτά τα 2 είναι συνήθως αντικρουόμενα:
 - Πρέπει να επιτευχθεί συμβιβασμός

Περιβάλλοντες Όγκοι (4)

- Ο περιβάλλον όγκος ορθογώνιο παραλληλεπίπεδο σχηματίζεται παίρνοντας το μέγιστο και το ελάχιστο των συντεταγμένων των κορυφών του μοντέλου
- Ορίζονται από 6 επίπεδα κάθετα στους κύριους άξονες και γι' αυτό ονομάζονται *περιβάλλοντα κιβώτια ευθυγραμμισμένα με τους άξονες (AABBs)*
 - Περιέχουν πολύ κενό χώρο

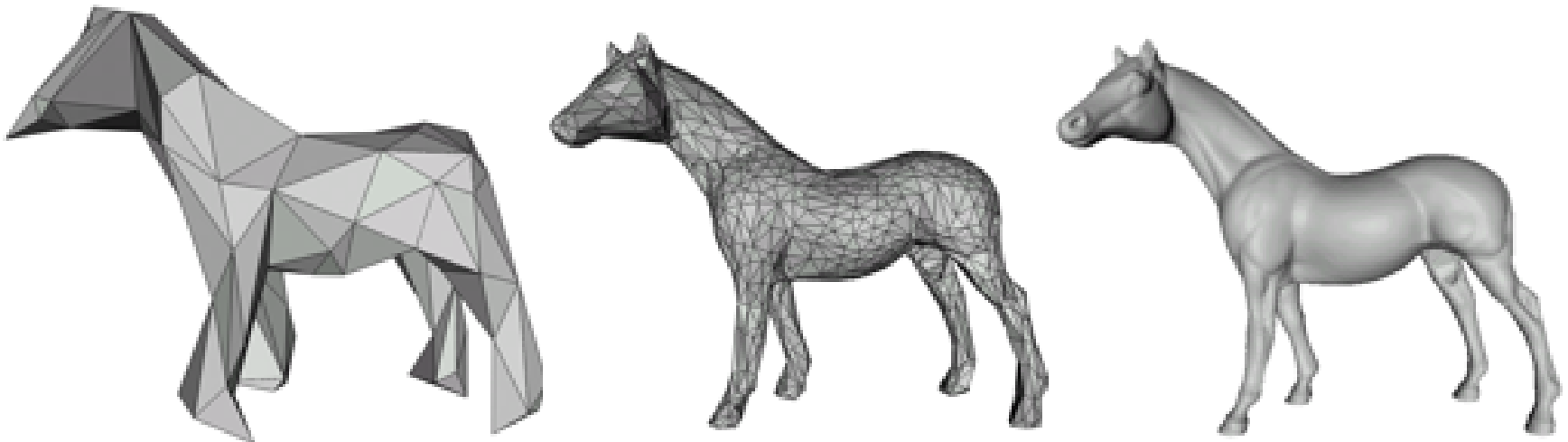


Περιβάλλοντες Όγκοι (5)

- Προσανατολισμένα περιβάλλοντα κιβώτια (OBBs) είναι ορθογώνια παραλληλεπίπεδα προσανατολισμένα αυθαίρετα
 - OBBs περιέχουν λιγότερο κενό χώρο από τα AABBs
- Ιεραρχικοί περιβάλλοντες όγκοι (hierarchical bounding volumes) συμβιβάζουν καλύτερα την απλότητα και τον κενό χώρο
 - Περιλαμβάνουν τις ιεραρχίες k-DOPs (πολύεδρα που έχουν πλευρές με προκαθορισμένους προσανατολισμούς) και ιεραρχίες OBBs
 - Δημιουργούνται δέντρα με εγκυβωτισμένους όγκους
 - ◆ Ρίζα: περιβάλλον όγκος που περιέχει όλο το μοντέλο
 - ◆ Κόμβοι: μικρότεροι όγκοι που εσωκλείουν τμήματα του μοντέλου
 - ◆ Φύλλα: στοιχειώδη αντικείμενα
 - Η δομή δέντρου περιορίζει την περιοχή των πιθανών τομών

Περιβάλλοντες Όγκοι (6)

- Άλλη ιεραρχική μέθοδος είναι τα προοδευτικά περιβλήματα
 - Ακολουθία από περιβλήματα που περικλείουν πιο στενά το μοντέλο
 - Κάθε περίβλημα περικλείει όλα τα επόμενα περιβλήματα
 - Όλα τα περιβλήματα περικλείουν ολόκληρο το μοντέλο
 - Τα εξωτερικά περιβλήματα είναι πιο απλά αλλά με περισσότερο κενό χώρο
 - Τα εσωτερικά περιβλήματα είναι πιο σύνθετα αλλά με λιγότερο κενό χώρο
 - Έναρξη από τα εξωτερικά περιβλήματα (πιο απλά), όσο υπάρχουν τομές



Περιβάλλοντες Όγκοι (7)

- Ψευδοκώδικας για ιεραρχικό έλεγχο τομής ενός μοντέλου M

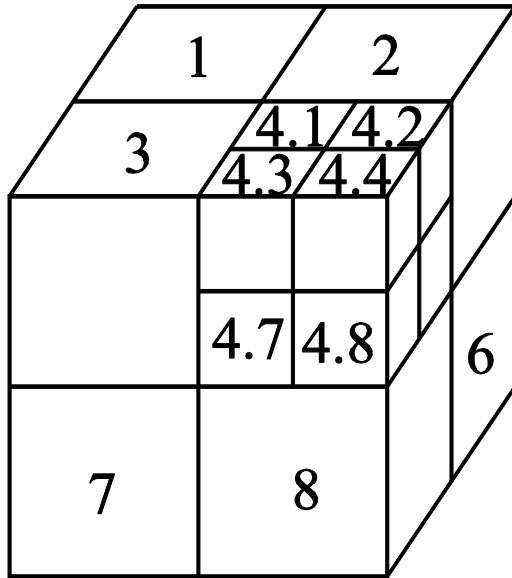
```
IntersectionTest (M) {  
  if BottomLevel (M)      return (LLIntersectionTest (M))  
  else if LLIntersectionTest (BoundingVolume (M)) {  
    v = false;  
    for κάθε τμήμα  $M \rightarrow C$   
      v = (v || IntersectionTest (M  $\rightarrow$  C));  
    return(v);  
  }  
  else return (false);  
}
```

- LLIntersectionTest: εκτελεί εξαντλητικό έλεγχο τομής για κάθε στοιχειώδες αντικείμενο της παραμέτρου του
- $M \rightarrow C$: αντιπροσωπεύει ένα τμήμα στο αμέσως πιο κάτω επίπεδο της ιεραρχίας

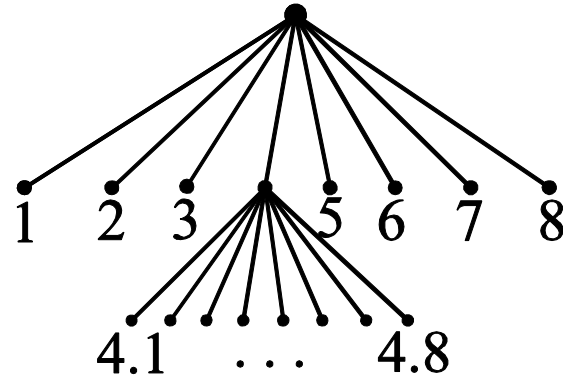
Υποδιαίρεση Χώρου

- Υποδιαίρεση 3Δ χώρου σε ένα διατεταγμένο σύνολο *κελιών*
- Τα κελιά προσδιορίζουν έμμεσα τη χωρική σχέση των αντικειμένων που τα καταλαμβάνουν
 - 2 αντικείμενα ενδέχεται να τέμνονται αν καταλαμβάνουν κοινά κελιά
 - Η διάταξη των κελιών προσδιορίζει την ενδεχόμενη παρεμπόδιση ενός αντικειμένου από κάποιο άλλο αντικείμενο
- Οι τεχνικές υποδιαίρεσης χώρου απαιτούν ειδικές δομές δεδομένων και επεξεργασία για να καταχωρηθούν τα αντικείμενα στις δομές αυτές
- *Οκταδικό δέντρο*: ιεραρχική τεχνική υποδιαίρεσης 3Δ χώρου
 - Υποδιαιρεί αναδρομικά ένα αρχικό κελί σε 8 υποκελιά
 - Η υποδιαίρεση σταματά:
 - ◆ Όταν φτάσει σε κελί με στοιχειώδης μέγεθος (*ογκοστοιχείο*)
 - ◆ Όταν η πολυπλοκότητα του αντικειμένου μέσα στο κελί είναι μικρότερη από δεδομένο όριο

Υποδιαίρεση Χώρου (2)



Πεπερασμένος χώρος 3Δ



- Στην περικοπή, αγνοούνται τα μοντέλα που δεν καλύπτουν τα κελιά που μας ενδιαφέρουν:
 - ◆ Στην περικοπή στο οπτικό πεδίο, λαμβάνονται υπόψη μόνο τα κελιά που είναι κοινά με το οπτικό πεδίο
 - ◆ Στην περικοπή παρεμποδιζομένων, ελέγχονται μόνο αντικείμενα που καταλαμβάνουν κελιά με τις ίδιες x, y συντατεγμένες
- Το οκταδικό δέντρο έχει περισσότερα επίπεδα για σκηνές με μεγαλύτερη πολυπλοκότητα

Τέλος Ενότητας

*Αλγόριθμοι Περικοπής και Απομάκρυνσης
Κρυμμένων Επιφανειών*

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

Σημείωμα Αναφοράς

Copyright Εθνικών και Καποδιστριακών Πανεπιστημίων Αθηνών, Θεοχάρης Θεοχάρης. «Γραφικά Ι. Ενότητα 5: Αλγόριθμοι Περικοπής και Απομάκρυνσης Κρυμμένων Επιφανειών». Έκδοση: 1.01. Αθήνα 2015.

Διαθέσιμο από τη δικτυακή διεύθυνση:
<http://opencourses.uoa.gr/courses/DI104/>.



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.



Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση του ακόλουθου έργου:

«Γραφικά και Οπτικοποίηση. Αρχές και Αλγόριθμοι.» Θ. Θεοχάρης, Τ. Παπαϊωάννου, Ν. Πλατής, Ν. Μ. Πατρικαλάκης.

