

Γραφική με υπολογιστές: Καμπύλες και Επιφάνειες

Κ. Δελήμπασης

Παραμετρικός ορισμός καμπύλης

- Καμπύλη στο επίπεδο:

- Πεπλεγμένη (implicit) μορφή

$$f(x, y) = 0$$

- Παραμετρικός ορισμός (parametric)

$$\vec{p}(x, y) = (x(t), y(t))^T, t \in [0, 1]$$

- Καμπύλη στο χώρο:

- Πεπλεγμένη (implicit) μορφή

$$f(x, y, z) = 0$$

- Παραμετρικός ορισμός (parametric)

$$\vec{p}(x, y) = (x(t), y(t), z(t))^T, t \in [0, 1]$$

- Παράδειγμα: παραμετρική εξίσωση ευθείας 3D που διέρχεται από το σημείο (x_0, y_0, z_0) και είναι παράλληλη στο διάνυσμα (d_1, d_2, d_3) :

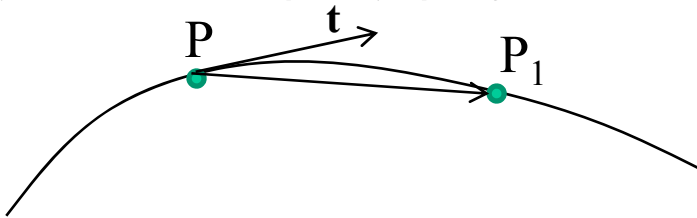
$$x(t) = x_0 + td_1$$

$$y(t) = y_0 + td_2$$

$$z(t) = z_0 + td_3$$

Εφαπτόμενο και κάθετο διάνυσμα καμπύλης σε 3D

- Το εφαπτόμενο διάνυσμα μίας καμπύλης στο σημείο P ορίζεται ως το διάνυσμα της 1^{ης} παραγώγου ως προς την παράμετρο της καμπύλης \rightarrow είναι ίσο με το PP_1 για $P_1 \rightarrow P$. Το εφαπτόμενο διάνυσμα κανονικοποιείται για να αποκτήσει μήκος =1.


$$\mathbf{t}(u) = \frac{\mathbf{f}'(u)}{|\mathbf{f}'(u)|} = \frac{(x'(u), y'(u), z'(u))}{|(x'(u), y'(u), z'(u))|}$$

- Αν θεωρήσουμε τρέχον σημείο και δύο σημεία επί της καμπύλης P_1, P_2 . Αν $P_1 \rightarrow P$ και $P_2 \rightarrow P$, τότε τα P_1, P_2, P καθορίζουν ένα επίπεδο το οποίο περιέχει το εφαπτόμενο διάνυσμα (tangent) στο P . Η πρώτη και δεύτερη παράγωγος της καμπύλης είναι διανύσματα που περιέχονται στο επίπεδο αυτό.

- Το διάνυσμα που περνά από το P και είναι κάθετο στο επίπεδο λέγεται δικάθετο διάνυσμα (binormal) και ορίζεται ως:

$$\mathbf{b}(u) = \frac{\mathbf{f}'(u) \times \mathbf{f}''(u)}{|\mathbf{f}'(u) \times \mathbf{f}''(u)|}$$

- Το διάνυσμα που είναι κάθετο στο εφαπτόμενο και το δικάθετο διάνυσμα είναι το κάθετο (normal) διάνυσμα της καμπύλης.

$$\mathbf{n}(u) = \frac{\mathbf{b}(u) \times \mathbf{f}'(u)}{|\mathbf{b}(u) \times \mathbf{f}'(u)|}$$

- Το Εφαπτόμενο μοναδιαίο διάνυσμα, το Δικάθετο και το κάθετο διάνυσμα αποτελούν μία ορθοκανονική βάση του \mathbb{R}^3 η οποία λέγεται τρίεδρο Frenet.

Παραδείγματα

- Παραμετρική εξίσωση 3D έλικας με κέντρο $(0,0)$, ακτίνα R και βήμα b . Η 1^η και 2^η παράγωγος υπολογίζεται:

$$\mathbf{f}(t) = (R \cos(t), R \sin(t), bt) \Rightarrow \begin{cases} \mathbf{f}'(t) = (-R \sin(t), R \cos(t), b) \\ \mathbf{f}''(t) = (-R \cos(t), -R \sin(t), 0) \end{cases}$$

- Το δικάθετο και το κάθετο διάνυσμα υπολογίζονται ως εξής:

$$\mathbf{b}(t) = \mathbf{f}'(t) \times \mathbf{f}''(t) = (Rb \sin t, -Rb \cos t, R^2)$$

$$\mathbf{n}(t) = \mathbf{b}(t) \times \mathbf{f}'(t) = (-R(R^2 + b^2) \cos t, -R(R^2 + b^2) \sin t, 0)$$

- Παρατηρούμε ότι το κάθετο διάνυσμα είναι παράλληλο με την \mathbf{f}'' .

- Παραμετρική εξίσωση κύκλου με κέντρο (x_0, y_0) και ακτίνα R :

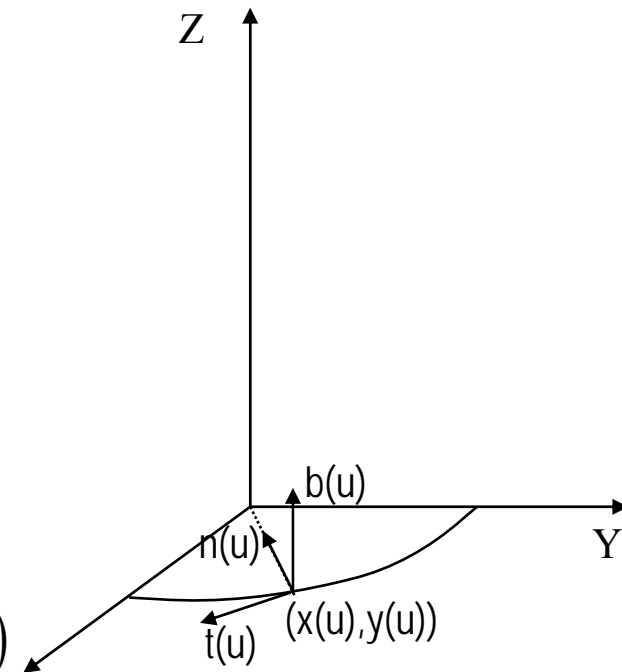
$$\mathbf{f}(t) = (R \sin(2\pi t) + x_0, R \cos(2\pi t) + y_0, 0)$$

$$\mathbf{f}'(t) = (R \cos(2\pi t), -R \sin(2\pi t), 0)$$

$$\mathbf{f}''(t) = (-R \sin(2\pi t), -R \cos(2\pi t), 0)$$

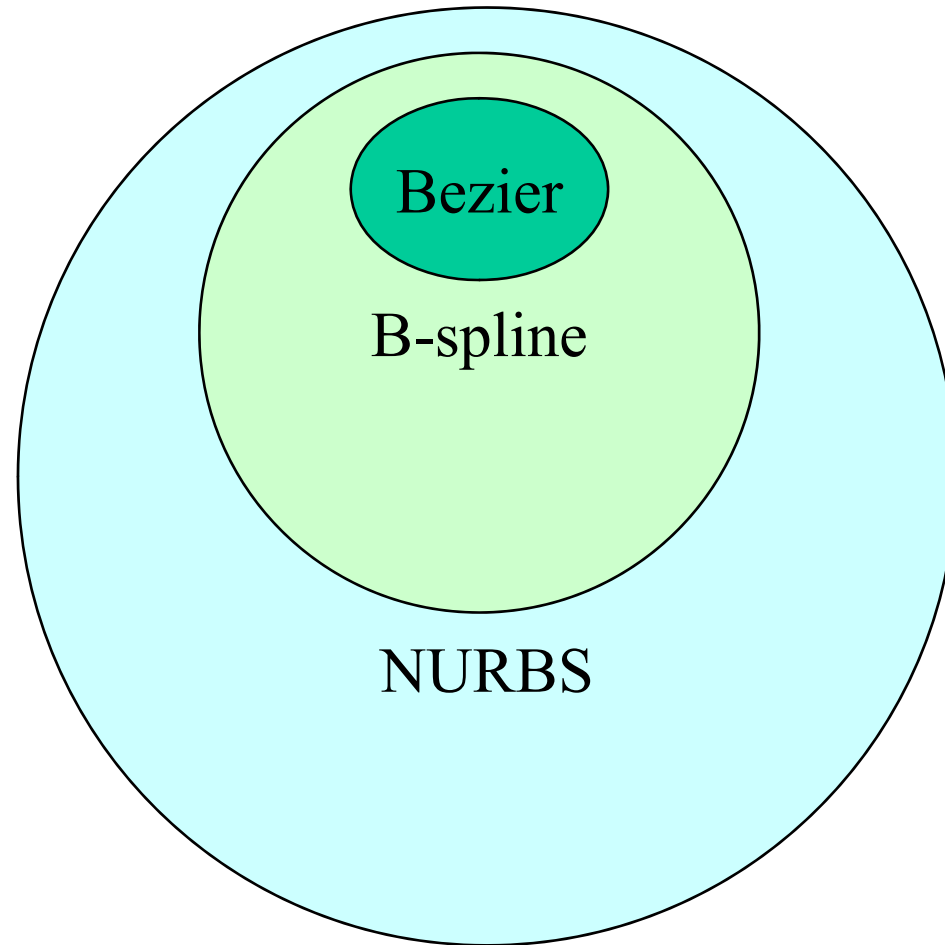
$$\mathbf{b}(t) = \mathbf{f}'(t) \times \mathbf{f}''(t) = (0, 0, R^2)$$

$$\mathbf{n}(t) = \mathbf{b}(t) \times \mathbf{f}'(t) = (R^3 \sin 2\pi t, R^3 \cos 2\pi t, 0)$$



- Προκύπτει ότι το δικάθετο είναι πάντα παράλληλο στον Z και το κάθετο είναι πάντα παράλληλο στην f .

Καμπύλες Bezier και Splines



Κ. Δελήμπασης

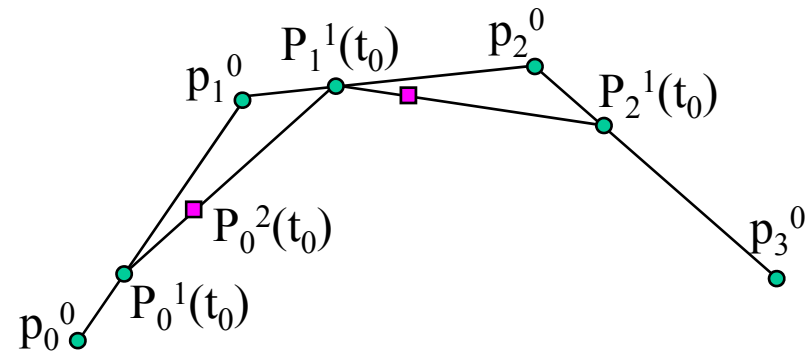
Καμπύλες Bezier

- Παράγονται από γραμμική παρεμβολή σημείων ελέγχου
 - Η θέση των σημείων καθορίζει το σχήμα της καμπύλης
 - Το πλήθος καθορίζει τον βαθμό της καμπύλης
- Γραμμική παρεμβολή 2 σημείων p_1, p_2 : η παραμετρική εξίσωση του ευθύγραμμου τμήματος που τα ενώνει

$$p = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_1 + t(x_2 - x_1) \\ y_1 + t(y_2 - y_1) \end{pmatrix}, t \in [0,1]$$

Καμπύλες Bezier βαθμού 2 και βαθμού n

- κηγκξγ



$$\left. \begin{aligned} p_0^1(t) &= p_0^0 + t(p_1^0 - p_0^0) \\ p_1^1(t) &= p_1^0 + t(p_2^0 - p_1^0) \end{aligned} \right\} \Rightarrow p_0^2(t) = p_0^0(1-t)^2 + p_1^0 2t(1-t) + p_2^0 t^2$$

$$p^n(t) = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} p_i^0$$

Κ. Δελήμπασης

Υπολογισμός σημείων καμπύλης

$$P_{ij} = (1-u)P_{i-1,j} + uP_{i-1,j+1}$$

Αναδρομικός υπολογισμός σημείου καμπύλης

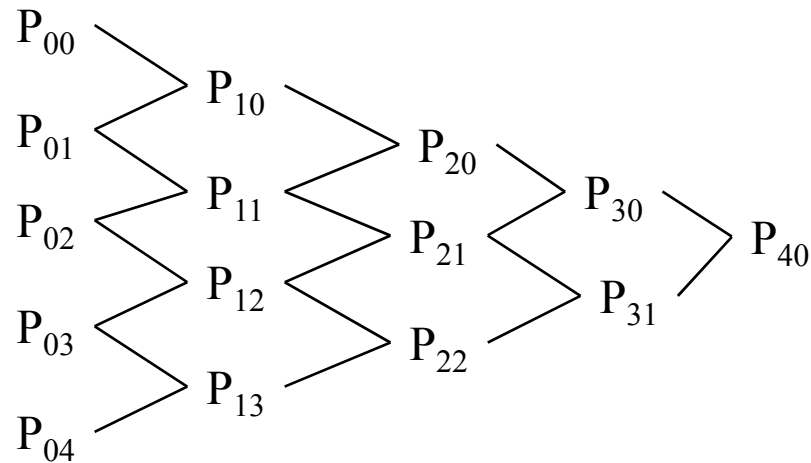
```
function de_Cast(i,j)
    if i=0 then return Pij else
    return (1-u)*De_Cast(i-1,j)+u*de_Cast(i-1,j+1)
end
```

- Αποτελεσματικός αλγόριθμος υπολογισμού σημείων Bezier βαθμού n-1

```

De_casteljau(Q[], u)
for k=1 to n
  for i=1 to n-k+1
    Q(i)=(1-u)*Q(i)+u*Q(i+1)
  end;
end;

```



Ιδιότητες των καμπύλων Bezier

- Έστω σημεία ελέγχου $\{P_i\}$ και η καμπύλη:

$$C(u) = \sum_{i=0}^n B_{n,i} P_i$$

$$B_{n,i} = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-1}$$

- Υπολογισμός της παραγώγου

$$\frac{d}{du}(C(u)) = \sum_{i=0}^n \frac{d}{du}(B_{n,i}) P_i$$

$$\frac{d}{du}(B_{n,i}) = n(B_{n-1,i-1}(u) - B_{n-1,i}(u))$$

$$\frac{d}{du}(C(u)) = \sum_{i=0}^n B_{n-1,i}(u)(n(P_{i+1} - P_i))$$

Η παράγωγος της C στο πρώτο ($u=0$) και τελευταίο σημείο ($u=1$) ισούται με την κλίση του 1ου και τελευταίου ευθύγραμμου τμήματος των σημείων ελέγχου.

- Διέλευση από το 1^ο και τελευταίο σημείο ελέγχου: η καμπύλη Bezier διέρχεται υποχρεωτικά από τα P_1 και P_n
 - $u=0 \rightarrow$ η καμπύλη Bezier ξεκινά από το P_1
 - $u=1 \rightarrow$ η καμπύλη Bezier τελειώνει στο P_n
- Η απόδειξη του παραπάνω είναι τετριμμένη, αρκεί να θεωρήσουμε ότι $0^0=1$.
- Μετακίνηση ενός σημείου ελέγχου: Εστω ότι το $\mathbf{p}_k \rightarrow \mathbf{p}_k + \mathbf{v}$. Τότε:

$$C(u) = \sum_{i=0}^n B_{n,i}(u) P_i^0, B_{n,i} = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-1}$$

$$C_1(u) = \sum_{i=0}^{k-1} B_{n,i}(u) P_i^0 + B_{n,k}(P_k + \mathbf{v}) + \sum_{i=k+1}^n B_{n,i}(u) P_i^0 =$$

$$\sum_{i=0}^n B_{n,i}(u) P_i^0 + B_{n,k} \mathbf{v} = C(u) + B_{n,k} \mathbf{v}$$

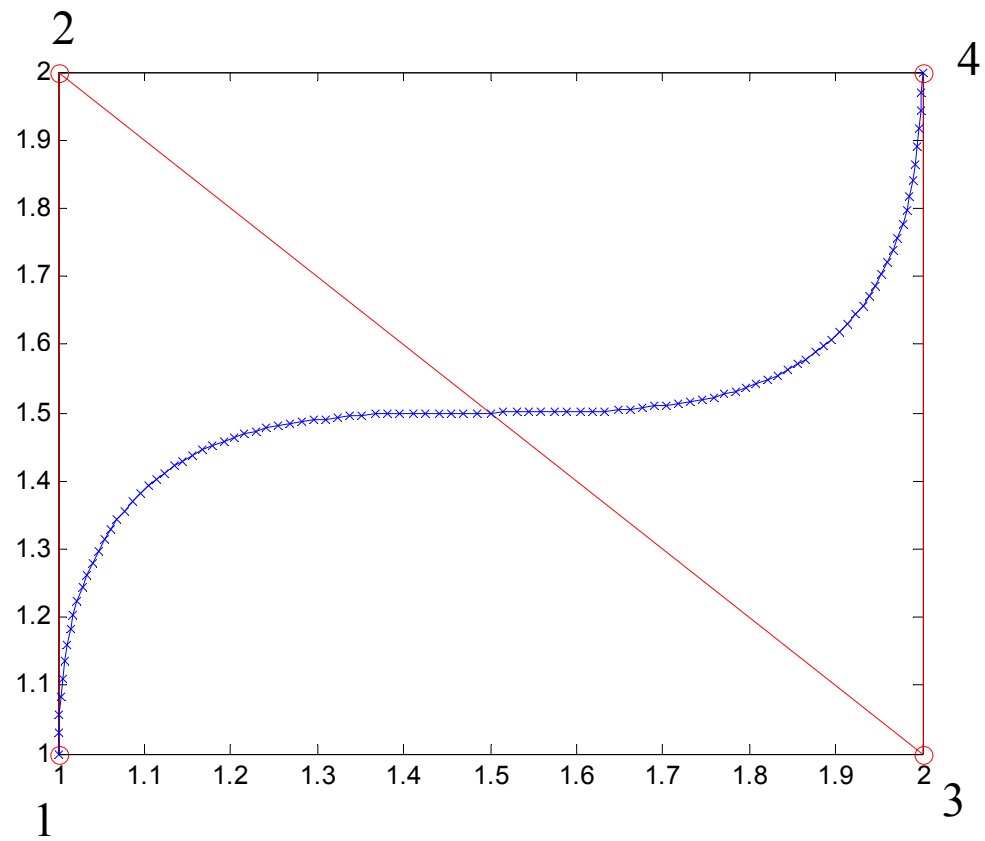
Κ. Δελήμπασης

- Μετακίνηση όλων των σημείων ελέγχου κατά \mathbf{v} :

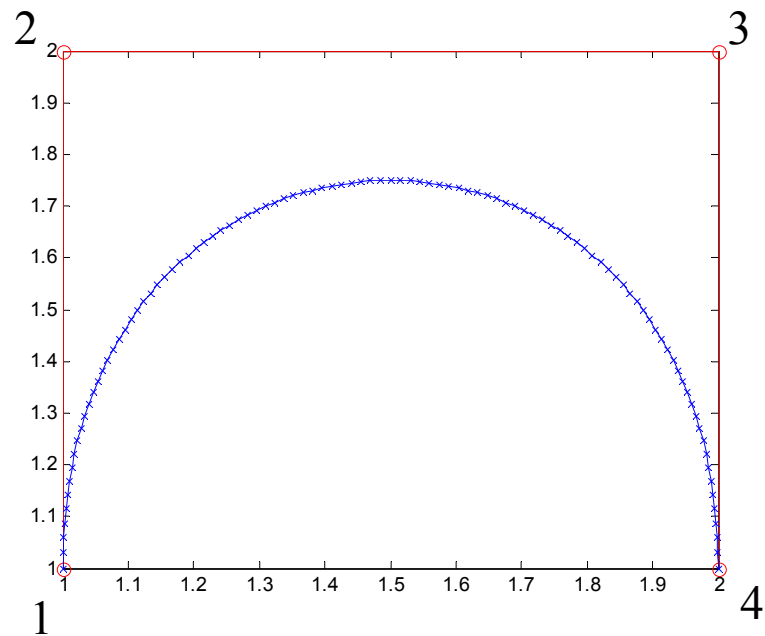
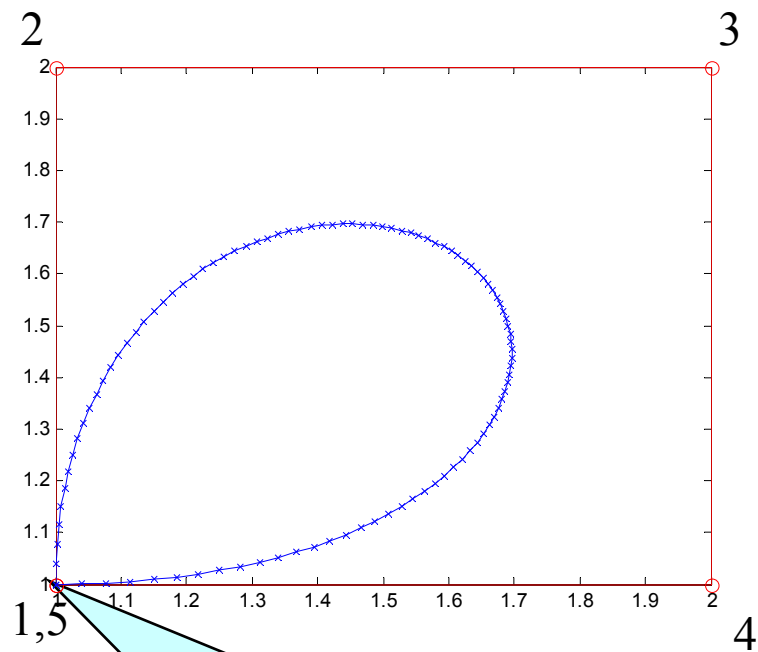
$$C(u) = \sum_{i=0}^n B_{n,i}(u) P_i^0, B_{n,i} = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-1}$$

$$C_1(u) = \sum_{i=0}^n B_{n,i}(u) (P_i^0 + \mathbf{v}) = \sum_{i=0}^n B_{n,i}(u) P_i^0 + \mathbf{v} \sum_{i=0}^n B_{n,i}(u) = C(u) + \mathbf{v}$$

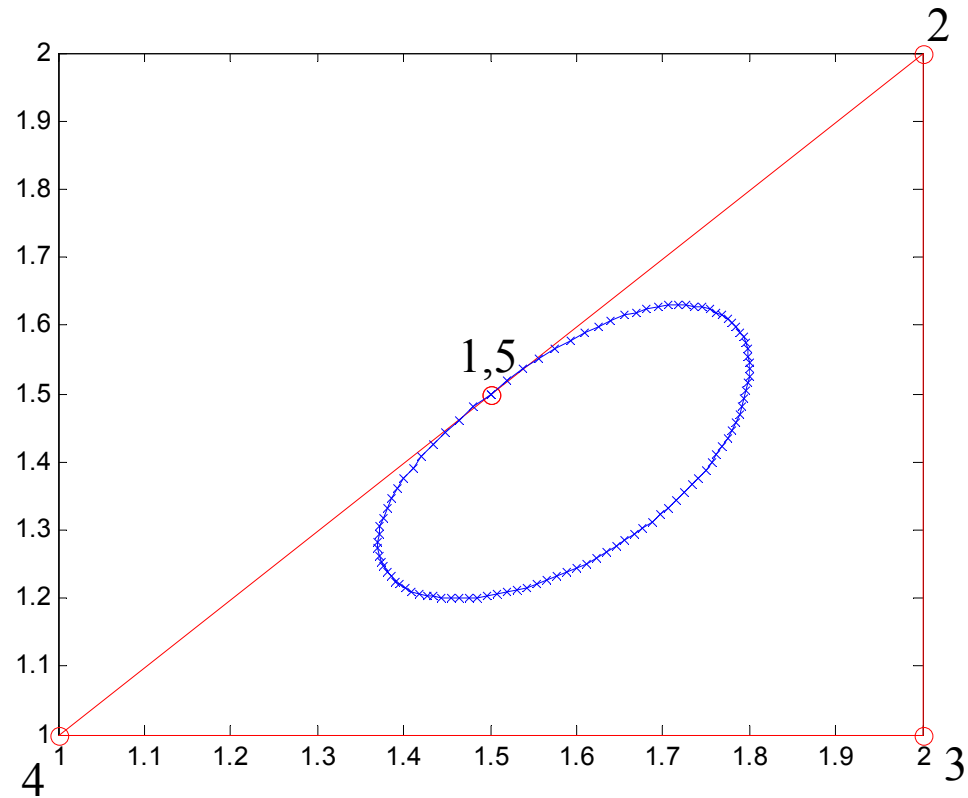
- Παρατηρούμε ότι αν εφαρμοστεί μετατόπιση στα σημεία ελέγχου, η ίδια μετατόπιση εφαρμόζεται σε όλα τα σημεία της καμπύλης.
- Η ιδιότητα αυτή ισχύει για κάθε συσχετισμένο μετασχηματισμό.



Κ. Δελήμπασης



Αρχικό = τελικό
 σημείο ελέγχου →
 Συνέχεια C^0



Συνθήκη για κλειστή καμπύλη με συνέχεια C^2 (ίσες 1ες παράγωγοι):
 συγγραμμικότητα σημείων 1, n-1, n.

Καμπύλες B-Splines

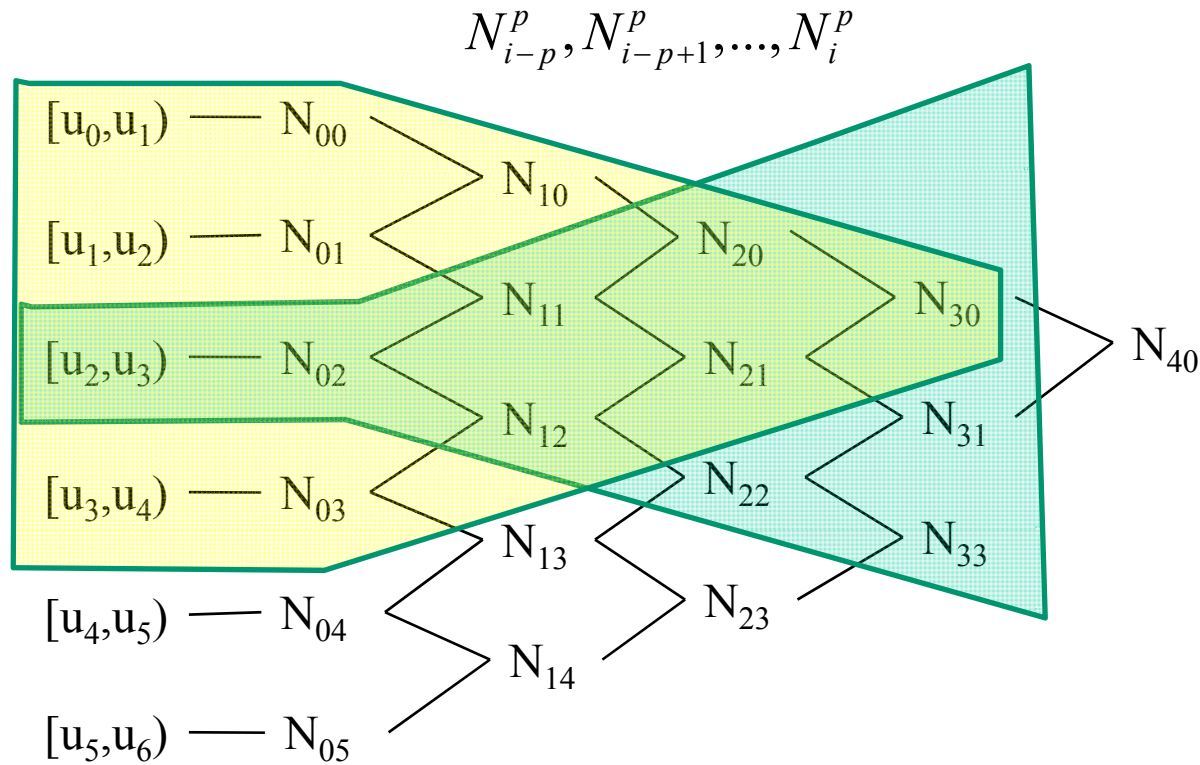
- Οι καμπύλες spline αποτελούνται από ένα αριθμό πολυωνυμικών τμημάτων βαθμού k τα οποία ορίζονται σε μη επικαλυπτόμενα και συνεχόμενα διαστήματα μίας παραμέτρου u .
- Τα πολυωνυμικά τμήματα ενώνονται με συνέχεια βαθμού $k-1$ C^{k-1} .
- Εστω $p_i, i=0 \dots n$ σημεία ελέγχου $(n+1)$ της καμπύλης.
- Τα πολυωνυμικά τμήματα καθορίζονται από ένα διάνυσμα κόμβων (knot vector) το οποίο περιέχει με αύξουσα σειρά τις τιμές της παραμέτρου u η οποία καθορίζει την καμπύλη.
$$(u_0, u_1, \dots, u_n)$$
$$[u_0, u_1), \dots, [u_i, u_{i+1}), \dots, [u_{n-1}, u_n)$$
- Αν το μήκος του διανύσματος κόμβων είναι $m+1$ και ο αριθμός των ανεξάρτητων πολυωνυμικών τμημάτων είναι $n+1$, τότε $m=n+k+1$.

- Το διάνυσμα των κόμβων μπορεί να περιλαμβάνει κάθε τιμή των κόμβων μόνο μία φορά, ή κάποια από τις τιμές του να εμφανίζονται $m > 1$ φορές. Στην δεύτερη περίπτωση ο συγκεκριμένος κόμβος έχει πολλαπλότητα m .
- Αν κάθε κόμβος δεν εμφανίζει πολλαπλότητα και οι κόμβοι διαιρούν το διάστημα της παραμέτρου σε ίσα μέρη, τότε το διάνυσμα κόμβων λέγεται ομογενές.
- Η βάση των συναρτήσεων B-Splines ορίζεται αναδρομικά (Cox de Boor recursive), ως εξής:

$$N_i^0 = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & u \notin [u_i, u_{i+1}) \end{cases}$$

$$N_i^p = \frac{u - u_i}{u_{i+p} - u_i} N_i^{p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u)$$

- Η συνάρτηση $N_{p,i}$ είναι μη μηδενική στα διαστήματα τιμών της παραμέτρου της καμπύλης $[u_i, u_{i+p+1})$, όπου p ο βαθμός του πολυωνύμου και i ο αύξων αριθμός.
- Αντίστοιχα το διάστημα $[u_i, u_{i+1})$ επηρεάζει τις $n+1$ συναρτήσεις βάσης:



Παράδειγμα

- Εστω η παραμετροποίηση καμπύλης B-Spline $U=\{0,1,2,3\}$. Υπολογίστε τις συναρτήσεις βάσης B-Spline βαθμού 0, 1 και 2.

$$p=1, i=0 \Rightarrow N_0^1(u) = \frac{u-u_0}{u_1-u_0} N_0^0(u) + \frac{u_2-u}{u_2-u_1} N_1^0(u) = uN_0^0(u) + (2-u)N_1^0(u)$$

$$p=1, i=1 \Rightarrow N_1^1(u) = \frac{u-u_1}{u_2-u_1} N_1^0(u) + \frac{u_3-u}{u_3-u_2} N_2^0(u) = (u-1)N_1^0(u) + (3-u)N_2^0(u)$$

$$p=2, i=0 \Rightarrow N_0^2(u) = \frac{u-u_0}{u_2-u_0} N_0^1(u) + \frac{u_3-u}{u_3-u_1} N_1^1(u) = \frac{u}{2} N_0^1(u) + \frac{(3-u)}{2} N_1^1(u) =$$

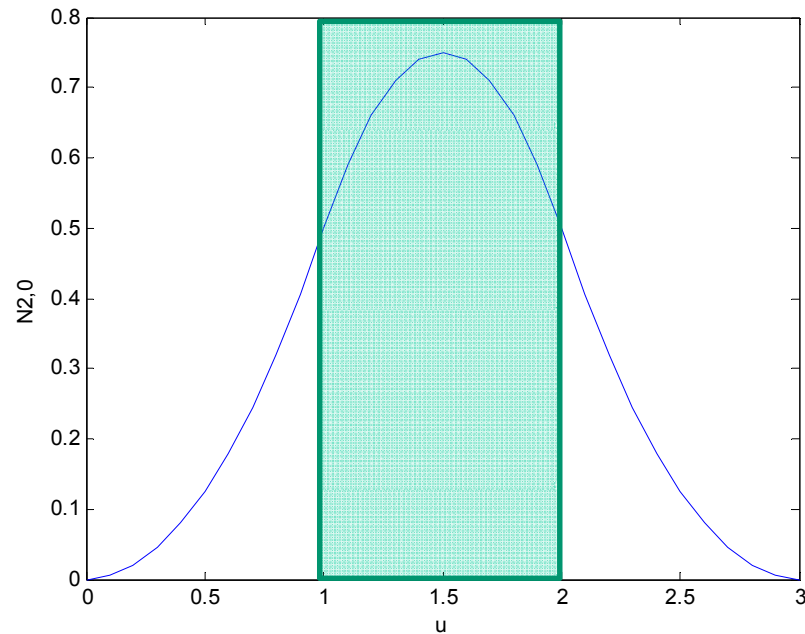
$$\frac{u^2}{2} N_0^0(u) + \frac{1}{2} [u(2-u) - (3-u)(u-1)] N_1^0(u) + \frac{(3-u)^2}{2} N_2^0(u)$$

$$N_0^2(u) = \begin{cases} u \in [0,1) \Rightarrow N_0^2(u) = \frac{u^2}{2} \\ u \in [1,2) \Rightarrow N_0^2(u) = \frac{1}{2}(-2u^2 + 6u - 3) \\ u \in [2,3) \Rightarrow N_0^2(u) = \frac{1}{2}(3-u)^2 \end{cases}$$

Κ. Δελήμπασης

- Παρατηρούμε ότι υπάρχει συνέχεια της συνάρτησης και της παραγώγου της στα σημεία ένωσης των υποδιαστημάτων:

$$\lim_{u \rightarrow 1^+} (N_0^2(u)) = \lim_{u \rightarrow 1^-} (N_0^2(u))$$

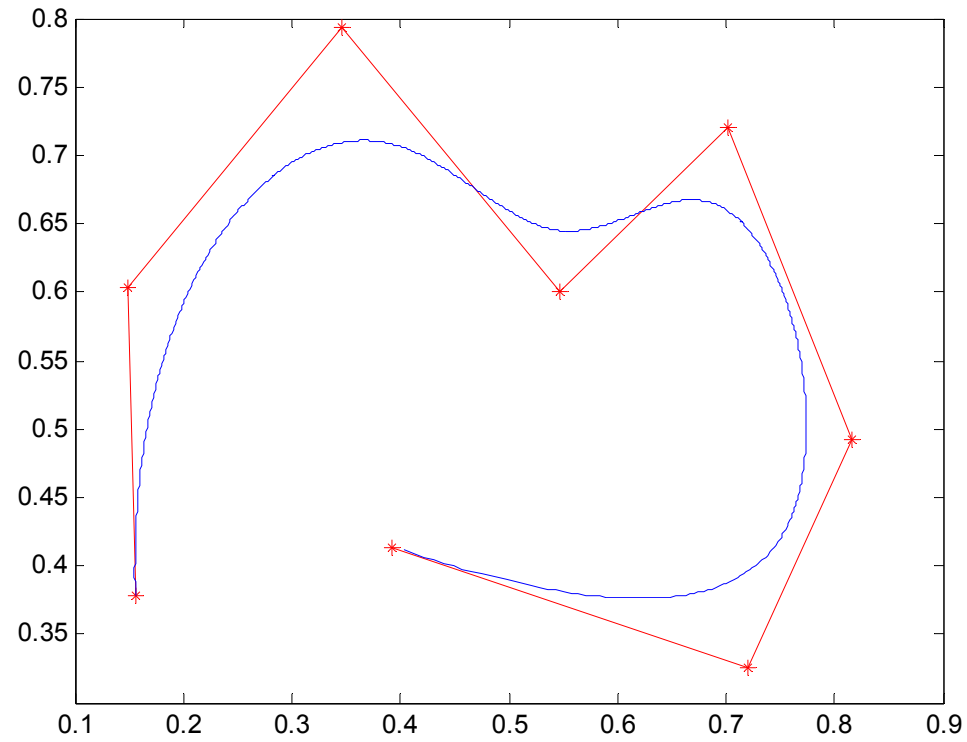


Ορισμός καμπυλών B-splines

- Δεδομένων $n+1$ σημείων ελέγχου $\{\mathbf{p}_i\}$, $i=0,1,\dots,n$, η καμπύλη B-spline βαθμού p ορίζεται ως εξής:

$$\mathbf{p}(u) = \sum_{i=0}^n N_i^p(u) \mathbf{p}_i$$

- Ανοικτές καμπύλες B-spline: αν το knot vector δεν έχει επαναλαμβανόμενους κόμβους, τότε η καμπύλη που παράγεται δεν θα διέρχεται από το 1^ο και το τελευταίο σημείο ελέγχου και δεν θα υπάρχει κάποια προβλεψη για την εφαπτομένη της καμπύλης για $u=0$ και $u=1$.
- Αν απαιτείται η παραγόμενη καμπύλη να διέρχεται από το 1^ο και το τελευταίο σημείο ελέγχου και η εφαπτιμένη της καμπύλης για $u=0$ και $u=1$ να είναι παράλληλη με το 1^ο και το τελευταίο ευθύγραμμο τμήμα ελέγχου, τότε οι κόμβοι u_0 και u_m πρέπει να επαναλαμβάνονται $p+1$ φορές, όπου p ο βαθμός της καμπύλης.



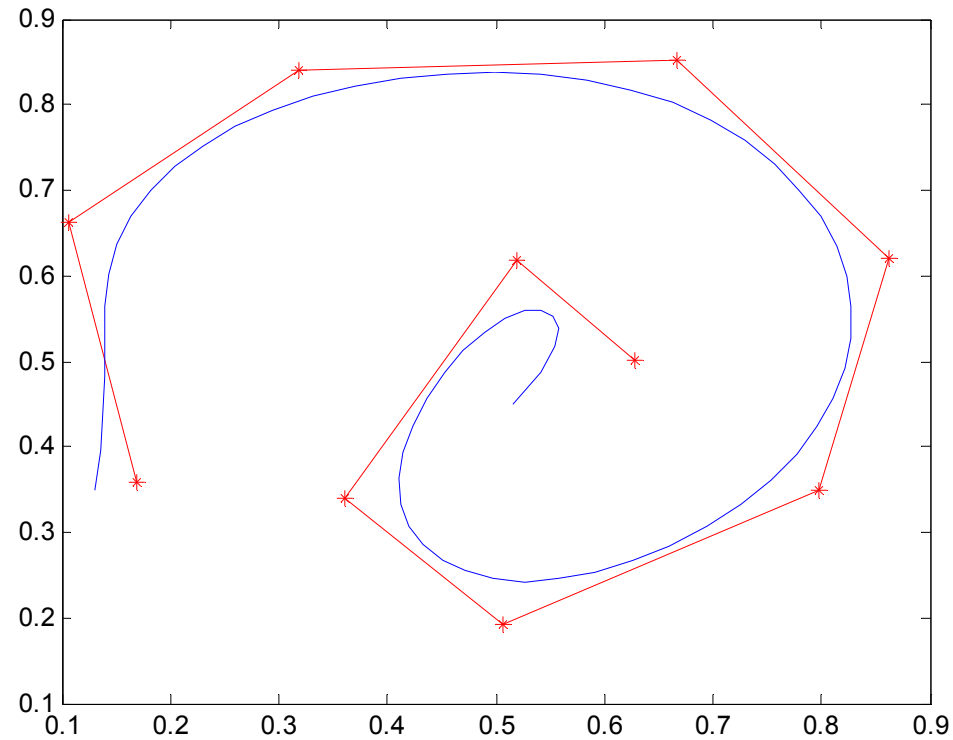
Βαθμός της καμπύλης: $p=3$

Πλήθος σημείων ελέγχου $n+1=8$

Διάσταση του knot vector: $m+1=12$ ($=n+1+p+1$)

Knot vector=[0 0 0 0 2 3 4 5 6 6 6 6]

Κ. Δελήμπασης



Βαθμός της καμπύλης: $p=3$

Πλήθος σημείων ελέγχου $n+1=10$

Διάσταση του knot vector: $m+1=14$ ($=n+1+p+1$)

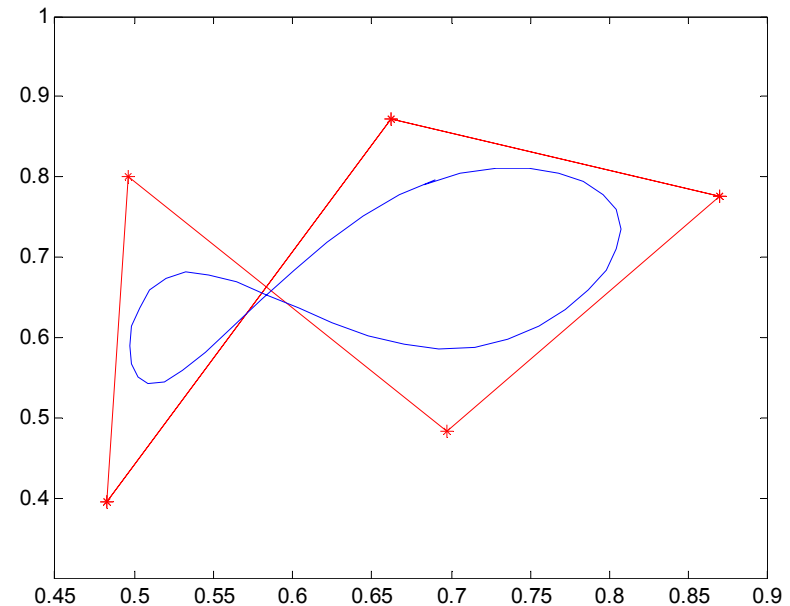
Knot vector=[0, 0.0714, 0.1429, 0.2143, 0.2857, 0.3571, 0.4286,
0.5000, 0.5714, 0.6429, 0.7143, 0.7857, 0.8571, 0.9286, 1.0000]

Κ. Δελήμπασης

Κατασκευή κλειστών καμπύλων B-Spline

- Εστω καμπύλη βαθμού k , με $n+1$ σημεία ελέγχου και $m+1$ knots, όπου $m=n+k+1$.
- Η καμπύλη που παράγεται είναι κλειστή αν για τα σημεία ελέγχου ισχύει η ακόλουθη σχέση:

$$p_i = p_{n-k+i+1}, i = 1, 2, \dots, p - 1$$



Εφαρμογή: προσέγγιση δεδομένων σημείων με καμπύλες Splines

- Συχνά είναι ιδιαίτερα χρήσιμο να προσεγγίσουμε ένα δεδομένο αριθμό σημείων $\{c_i\}$, $i=0, \dots, M-1$ με καμπύλη Splines. Τα βήματα που ακολουθούμε είναι τα ακόλουθα:
 - Καθορίζουμε το βαθμό d της καμπύλης
 - Καθορίζουμε τον αριθμό των σημείων ελέγχου $n+1$
 - Καθορίζουμε το διάνυσμα κόμβων (knot vector) $\{t_j\}$, $j=0, \dots, m$, όπου $m=n+d+1$.
 - Καθορίζουμε την παραμετροποίηση της καμπύλης, δηλ. τις τιμές $\{s_i\}$, $i=0, \dots, M-1$. Προσοχή στο γεγονός ότι οι παράμετροι $\{s_i\}$ παίρνουν τιμές στο διάστημα $[t_0, t_m]$, αλλά δεν πρέπει να συγχέονται με το διάνυσμα κόμβων.
 - Υπολογίζουμε τα σημεία ελέγχου $\{c_i\}$, $i=0, \dots, n$.
- Συνήθως $M \gg n$, άρα η χρησιμότητα της μεθόδου είναι στην ακριβή περιγραφή ενός συνόλου σημείων μέσω αναλυτικής μορφής.

- Καθορισμός του βαθμού d της καμπύλης: συνήθως επιλέγεται σταθερή τιμή ίση με 3
- Καθορισμός του αριθμού των σημείων ελέγχου $n+1$: αυτός γίνεται από το χρήστη. Αν είναι μικρός \rightarrow ακρίβεια της προσέγγισης δεν θα είναι ικανοποιητική. Αν είναι μεγάλος \rightarrow θα παραχθεί ασταθής καμπύλη.
- Καθορίζουμε το διάνυσμα κόμβων (knot vector) $\{t_j\}, j=0, \dots, m$, όπου $m=n+d+1$. Συνήθως ακολουθείται η ομογενής παραμετροποίηση τηρώντας πολλαπλότητα 3 στη πρώτη και τελευταία τιμή του:

$$t_0 = t_1 = t_2, t_{m-2} = t_{m-1} = t_m$$

- Καθορισμός της παραμετροποίησης της καμπύλης, δηλ. τις τιμές $\{s_i\}, i=0, \dots, M-1$: η παράμετρος για το πρώτο και τελευταίο σημείο τίθεται ίση με την αρχική και τελική τιμή του knot vector αντίστοιχα, τα δε ενδιάμεσα σημεία υπολογίζονται βάσει της απόστασης τους από το αρχικό σημείο, επί της τεθλασμένης που ορίζουν τα σημεία $\{c_i\}$.

$$s_i = t_0 + \frac{t_m - t_0}{\sum_{i=0}^n D_i} D_i, D_i = \begin{cases} 0, & i = 0 \\ D_{i-1} + \|c_i, c_{i-1}\|, & i > 0 \end{cases}$$

Κ. Δελήμπασης

- Υπολογίζουμε τα σημεία ελέγχου, λύνοντας

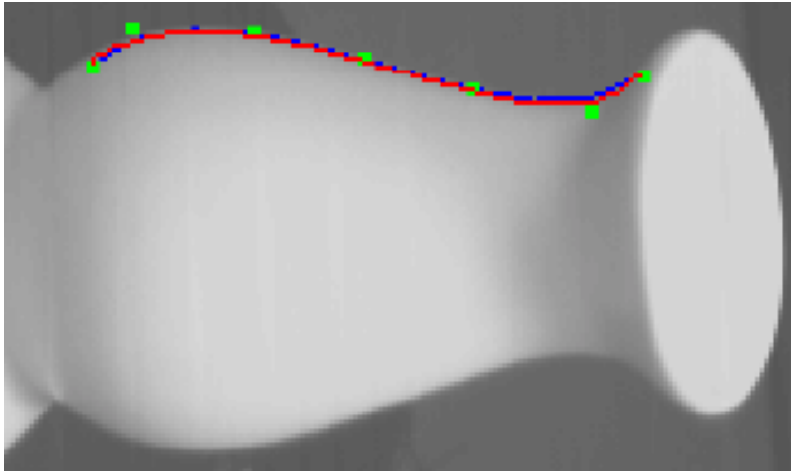
$$\mathbf{P}(s_i) = \mathbf{c}_i \Rightarrow \sum_{k=0}^{n-1} B_k^d(t_i) \mathbf{p}_k = \mathbf{c}_i, \quad 0 \leq i < M$$

- Η προηγούμενη σχέση περιγράφει ένα σύστημα M εξισώσεων:

$$\underbrace{\begin{bmatrix} B_0^d(t_0) & B_1^d(t_0) & \dots & B_{N-1}^d(t_0) \\ B_0^d(t_1) & B_1^d(t_1) & \dots & B_{N-1}^d(t_1) \\ \vdots & \vdots & \dots & \vdots \\ B_0^d(t_{M-1}) & B_1^d(t_{M-1}) & \dots & B_{N-1}^d(t_{M-1}) \end{bmatrix}}_{\mathbf{B}} \times \underbrace{\begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{N-1} \end{bmatrix}}_{\mathbf{P}} = \underbrace{\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{M-1} \end{bmatrix}}_{\mathbf{C}} \Leftrightarrow \mathbf{BP} = \mathbf{C}$$

- Επειδή συνήθως $M > N$, το σύστημα είναι υπερκαθορισμένο και λύνεται με προσέγγιση ελαχίστων τετραγώνων.

Παράδειγμα προσέγγισης ακμής εικόνας με καμπύλες Splines, 3^{ου} βαθμού



- Σημεία προς προσέγγιση
- Υπολογισμένη Spline
- Σημείο ελέγχου

Παρεμβολή (interpolation) σημείων καμπύλες Splines

- Εστω $n+1$ σημεία $\{\mathbf{P}_i^0\}$, $i=0,1,\dots,n$. Αναζητούμε μία κυβική καμπύλη $Q(t)$ η οποία παρεμβάλλει τα $\{\mathbf{P}_i^0\}$, δηλ $Q(t_i)=\mathbf{P}_i^0$.
- Οι τιμές της παραμέτρου t_i θα χρησιμοποιηθούν και για το διάλυμα δεσμών της καμπύλης.
- Αφού έχουμε $n+1$ σημεία προς παρεμβολή θα χρησιμοποιήσουμε $n+1$ σημεία ελέγχου για την καμπύλη (αν χρησιμοποιούσαμε λιγότερα \rightarrow καμπύλη προσέγγισης). Αρα θα χρειαστούν $m+1$ σημεία για το διάλυμα δεσμών της καμπύλης, όπου $m=n+3(\text{βαθμός})+1$.
- Δεδομένου ότι πρόκειται για παρεμβολή, $Q(t_0)=\mathbf{P}_0^0$ και $Q(t_n)=\mathbf{P}_n^0$, \rightarrow οι ακραίες τιμές των κόμβων θα πρέπει να έχουν πολλαπλότητα 3.

- Βάσει του ορισμού de Boor των συναρτήσεων B Splines, ισχύει για τις συναρτήσεις βαθμού 2:

$$N_i^2(t) = \begin{cases} \frac{t-t_i}{t_{i+1}-t_i} \frac{t-t_{i+1}}{t_{i+2}-t_{i+1}}, & t \in [t_i, t_{i+1}) \\ \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} \frac{t-t_i}{t_{i+2}-t_i} + \frac{t-t_{i+1}}{t_{i+2}-t_{i+1}} \frac{t_{i+3}-t}{t_{i+3}-t_{i+1}}, & t \in [t_{i+1}, t_{i+2}) \\ \frac{t_{i+3}-t}{t_{i+3}-t_{i+2}} \frac{t_{i+3}-t}{t_{i+3}-t_{i+1}}, & t \in [t_{i+2}, t_{i+3}) \\ 0 & \text{διαφορετικά} \end{cases}$$

- Στην περίπτωση της παρεμβολής, το t μπορεί να πάρει συγκεκριμένες τιμές: για την i -οστή συνάρτηση, τρεις δυνατές διαφορετικές τιμές: $t=t_i, t_{i+1}, t_{i+2}$. (για οποιεσδήποτε άλλες τιμές $N_i^2(t)=0$).

$$N_i^2(t_j) = \begin{cases} 0 & , j = i + 1 \\ \frac{t_{i+1}-t_i}{t_{i+2}-t_i}, & j = i + 2 \\ \frac{t_{i+3}-t_{i+2}}{t_{i+3}-t_{i+1}}, & j = i + 3 \\ 0 & \text{διαφορετικά} \end{cases}$$

- Εφαρμόζοντας τον αναδρομικό ορισμό των συναρτήσεων βάσης Splines αποκτούμε την έκφραση των συναρτήσεων βάσης Splines 3^{ου} βαθμού, για $t=t_{i+1}, t_{i+2}, t_{i+3}$:

$$N_i^3(t_j) = \frac{t_j - t_i}{t_{i+3} - t_i} N_i^2(t_j) + \frac{t_{i+4} - t_j}{t_{i+4} - t_{i+1}} N_{i+1}^2(t_j) = \begin{cases} j = i \Rightarrow 0 \\ j = i + 1 \Rightarrow \frac{t_j - t_i}{t_{i+3} - t_i} \frac{t_{i+1} - t_i}{t_{i+2} - t_i} \\ j = i + 2 \Rightarrow \frac{t_{i+2} - t_i}{t_{i+3} - t_i} \frac{t_{i+3} - t_{i+2}}{t_{i+3} - t_{i+1}} + \frac{t_{i+4} - t_{i+2}}{t_{i+4} - t_{i+1}} \frac{t_{i+2} - t_{i+1}}{t_{i+3} - t_{i+1}} \\ j = i + 3 \Rightarrow \frac{t_{i+4} - t_{i+3}}{t_{i+4} - t_{i+1}} \frac{t_{i+4} - t_{i+3}}{t_{i+4} - t_{i+2}} \\ \text{αλλιώς} \Rightarrow 0 \end{cases}$$

$$N_{j-1}^3(t_j) = \frac{t_j - t_{j-1}}{t_{i+3} - t_i} \frac{t_j - t_{j-1}}{t_{j+1} - t_{j-1}}$$

$$N_{j-2}^3(t_j) = \frac{t_j - t_{j-2}}{t_{j+1} - t_{j-2}} \frac{t_{j+1} - t_j}{t_{j+1} - t_{j-1}} + \frac{t_{j+2} - t_j}{t_{j+2} - t_{j-1}} \frac{t_j - t_{j-1}}{t_{j+1} - t_{j-1}}$$

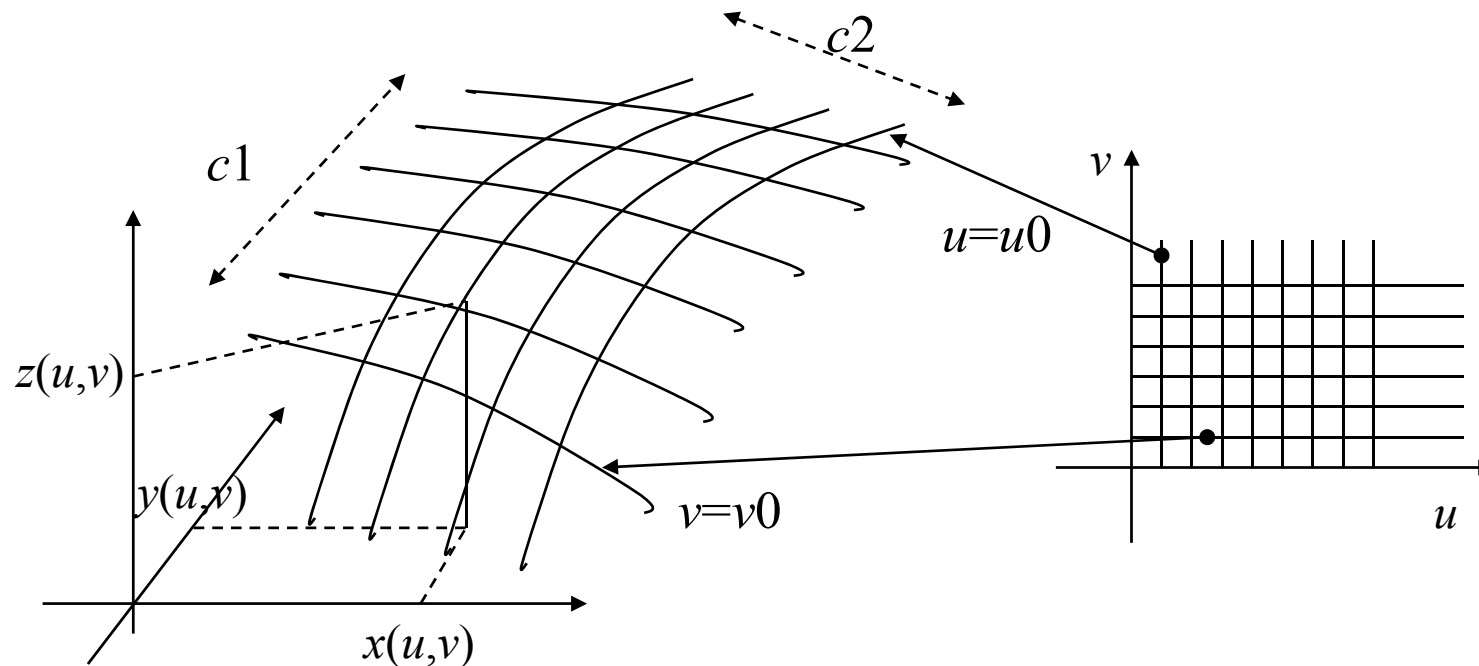
$$N_{j-3}^3(t_j) = \frac{t_{j+1} - t_j}{t_{j+1} - t_{j-2}} \frac{t_{j+1} - t_j}{t_{j+1} - t_{j-1}}$$

Κ. Δελήμπασης

Επιφάνειες

- Πεπλεγμένη μορφή (implicit form): $f(x,y,z)=0$
- Παραμετρική μορφή (parametric form):
 - $f(u,v)=(x(u,v), y(u,v), z(u,v))$
- Μπορεί να αποδειχθεί ότι η πεπλεγμένη μορφή μπορεί να αναπαραστήσει επιφάνειες που δεν αναπαρίστανται σε παραμετρική μορφή. Αντίθετα από την παραμετρική μορφή είναι πάντα δυνατό να εξάγουμε την πεπλεγμένη μορφή με απαλειφή των παραμέτρων u,v .
- Οι παράμετροι u,v ορίζονται στο πεδίο $[0,1] \times [0,1]$
- Ισοπαραμετρικές καμπύλες επί της επιφάνειας $c1=f(u_0,v), c2=f(u,v_0)$.

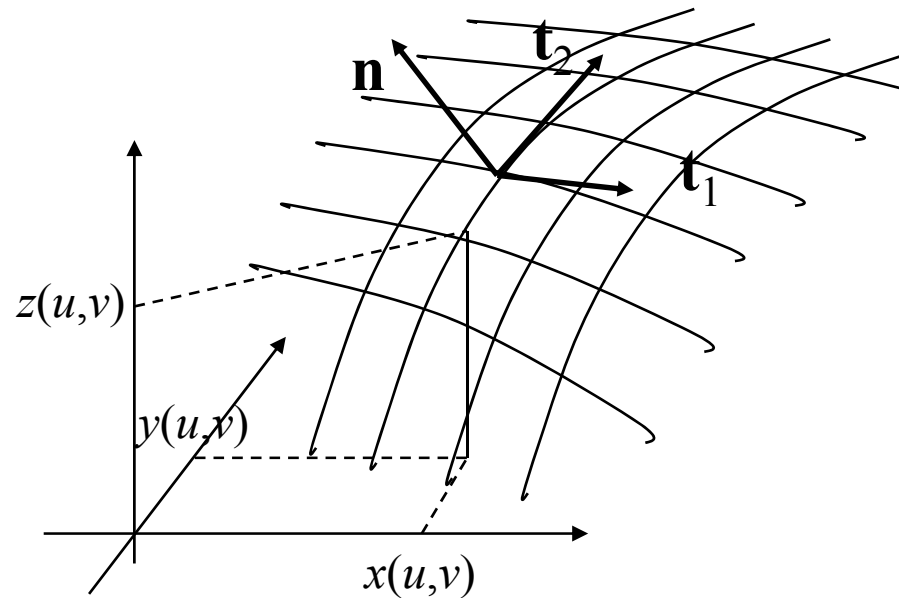
- Από τον παραμερικό χώρο (u,v) στον Ευκλείδειο χώρο.
- Ορισμός ισοπαραμετρικών καμπυλών.



Ορισμός εφαπτόμενων διανυσμάτων και κάθετου διανύσματος παραμετρικής επιφάνειας

- khgjk

$$\vec{t}_1 = \frac{\partial f}{\partial u}, \vec{t}_2 = \frac{\partial f}{\partial v}$$
$$\vec{n} = \frac{\vec{t}_1 \times \vec{t}_2}{|\vec{t}_1 \times \vec{t}_2|}$$



Παράδειγμα

- Επιβεβαιώστε την παραμετρική αναπαράσταση της σφαίρας S και υπολογίστε ότι το κάθετο διάνυσμα σε κάθε σημείο της S έχει σταθερό μέτρο και είναι παράλληλο προς την ακτίνα R .

$$\mathbf{f}(u, v) = \mathbf{f}(\theta, \varphi) = (R \sin \varphi \cos \theta, R \sin \varphi \sin \theta, R \cos \varphi)$$

$$\mathbf{f}_u = \frac{\partial \mathbf{f}}{\partial \theta} = (-R \sin \varphi \sin \theta, R \sin \varphi \cos \theta, 0)$$

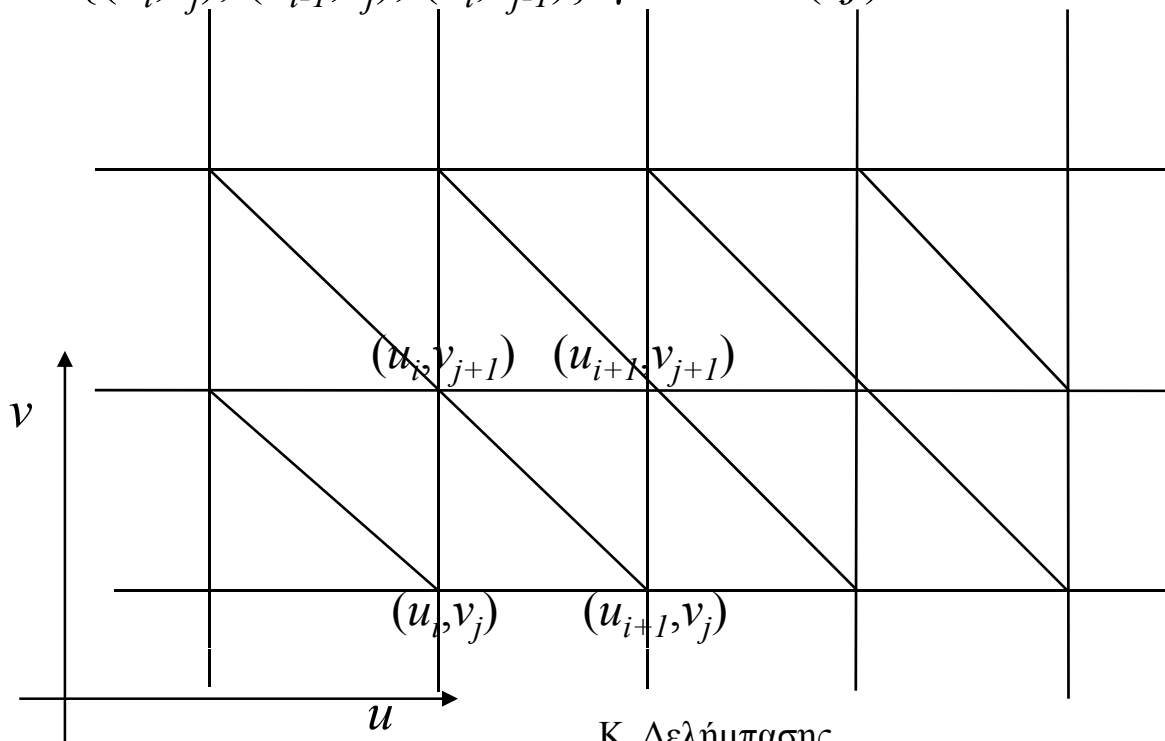
$$\mathbf{f}_v = \frac{\partial \mathbf{f}}{\partial \varphi} = (R \cos \varphi \cos \theta, R \cos \varphi \sin \theta, -R \sin \varphi)$$

$$\mathbf{n} = \mathbf{f}_u \times \mathbf{f}_v = (-R^2 \sin^2 \varphi \cos \theta, -R^2 \sin^2 \varphi \sin \theta, -R^2 \sin \varphi \cos \varphi) = -R \sin \varphi \mathbf{f}$$

Οπτικοποίηση (visualization) επιφανειών

- Η οπτικοποίηση επιφανειών απαιτεί το διαμερισμό – **tessellation**- της επιφάνειας σε στοιχειώδη τρίγωνα ή πολύγωνα, ώστε σε κάθε σημείο να υπολογίζεται το κάθετο διάνυσμα στην επιφάνεια. Τα πολύγωνα δεν χρειάζεται να είναι συνεπίπεδα.
- Στην περίπτωση επιφάνειας σε πεπλεγμένη μορφή, η οπτικοποίηση μπορεί να γίνει είτε μέσω τριγωνοποίησης των σημείων της επιφάνειας, είτε μέσω τεχνικών raytracing.
 - Η τριγωνοποίηση των σημείων της επιφάνειας αποτελεί δύσκολο εγχείρημα
 - Οι τεχνικές raytracing είναι υπολογιστικά απαιτητικές και δύσκολο να αναπτυχθούν
- **Tessellation του χώρου παραμέτρων (u,v)** : γίνεται με χρήση της τοπολογίας «δικτύου» της διάταξης των (u,v) .
 - Αποτελεί την ευκολότερη τριγωνοποίηση / πολυγωνοποίηση.
 - Απαιτεί να υπάρχει αναλυτική παραμετρική αναπαράσταση της επιφάνειας

- Α) πολυγωνοποίηση (τετραγωνοποίηση) στο χώρο των παραμέτρων: $\{(u_i, v_j), (u_{i+1}, v_j), (u_{i+1}, v_{j+1}), (u_i, v_{j+1})\}$ για κάθε (i, j)
- Τριγωνοποίηση –triangulation:
 - $\{(u_i, v_j), (u_{i+1}, v_j), (u_i, v_{j+1})\}$ για κάθε (i, j)
 - $\{(u_i, v_j), (u_{i-1}, v_j), (u_i, v_{j-1})\}$ για κάθε (i, j)



Δομές δεδομένων

- Για μία τριγωνοποιημένη / πολυγωνοποιημένη επιφάνεια απαιτούνται:
 - Οι κορυφές (σημεία) – vertices-
 - Η διατεταγμένη ακολουθία κορυφών που ορίζουν το κάθε τρίγωνο / πολύγωνο
- Πχ Matlab: ορίζουμε τον πίνακα vertices $n \times 3$ και
 - Τον πίνακα faces $k \times 3$ (τριγωνοποίηση) ή $k \times m$ (πολυγωνοποίηση), κάθε γραμμή του οποίου περιέχει τους δείκτες (indexes) του πίνακα vertices των σημείων που αποτελούν το πολύγωνο.
 - Η ίδια λογική χρησιμοποιείται και άλλα format πχ VRML (θα το δούμε πιο αναλυτικά παρακάτω)

Παράδειγμα

- Εκφράστε τον μοναδιαίο κύβο σαν πολυγωνοποιημένη επιφάνεια

- vertices

0 0 0

1 0 0

1 1 0

0 1 0

0 0 1

1 0 1

0 1 1

1 1 1

- faces

1 2 6 5

2 3 8 6

3 4 7 8

4 1 5 7

1 2 3 4

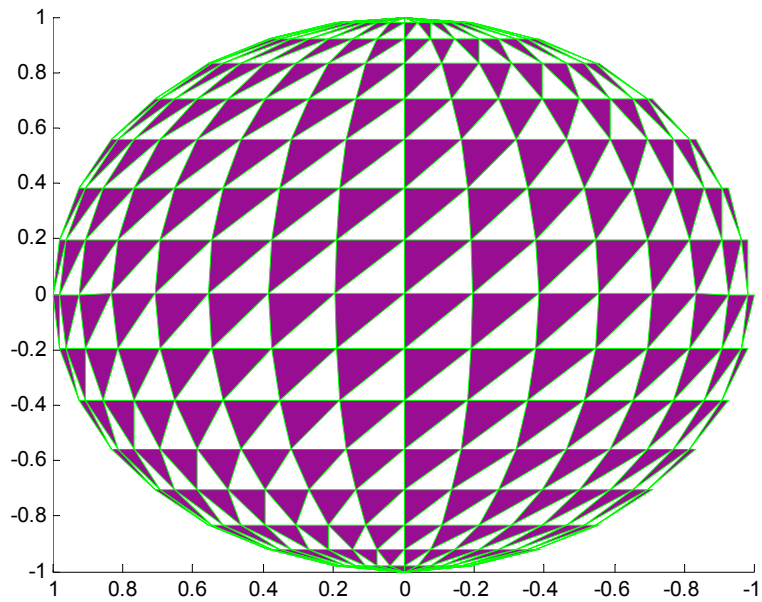
5 6 8 7

```
patch('Vertices',vertices,'Faces',faces,'Facecolor',rand(1,3),  
      'Edgecolor','green');
```

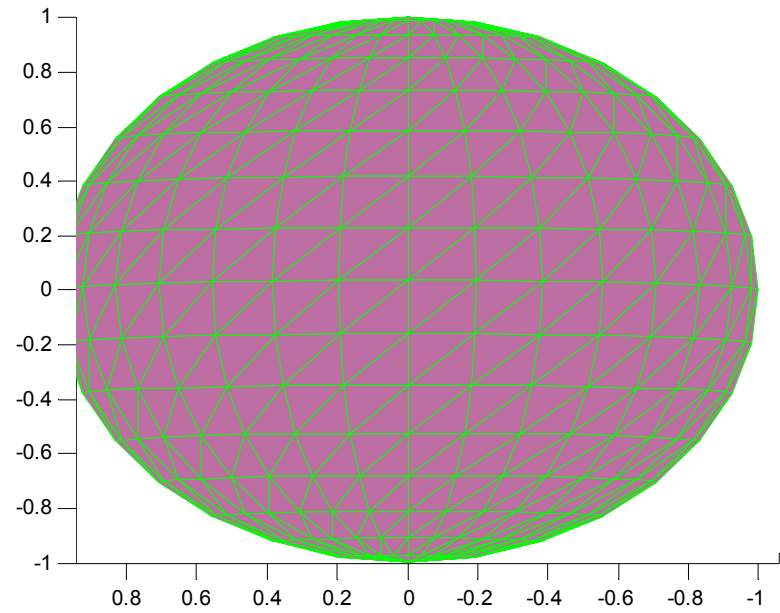
Κ. Δελημπασης

Παράδειγμα

- Υπολογίστε την τριγωνοποιημένη επιφάνεια μίας σφαίρας, βάσει της παραμετρικής αναπαράστασης της.
 - Τριγωνοποιείτε στο χώρο των παραμέτρων
 - Χρησιμοποιείτε τη δομή δεδομένων vertices/faces για αναπαράσταση στο Matlab.



Δημιουργία μισού αριθμού
τριγώνων



Πλήρης τριγωνοποίηση

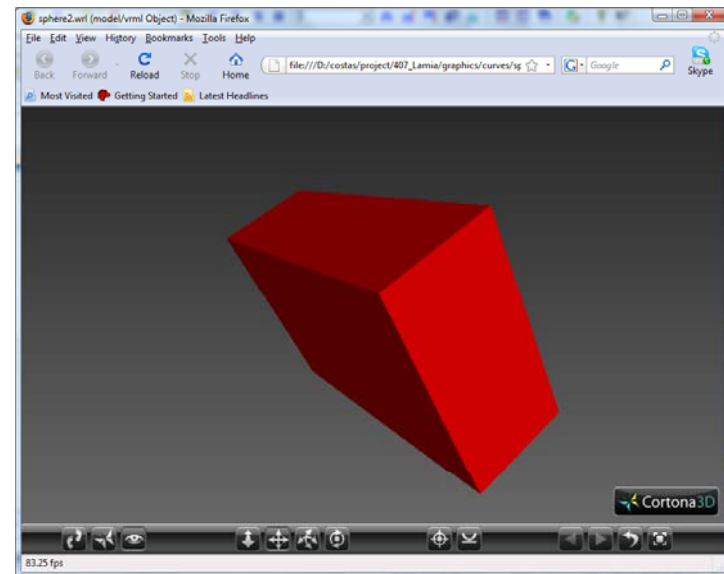
το φορμάτ αρχείων WRL

- VRML (*Virtual Reality Modeling Language*): μία μεταφέρσιμη γλώσσα περιγραφής τρισδιάστατων αντικειμένων και συνδυασμού αυτών σε σκηνές ή κόσμους
- Το VRML είναι WWW compatible
- θεμελιώδες δομικό στοιχείο της VRML είναι ο κόμβος (*node*)
- αποτελούνται με τη σειρά τους από πεδία (*fields*) που περιγράφουν τις ιδιότητες τους

Ιεράρχιση των κόμβων

- Οι απλούστεροι κόμβοι που περιλαμβάνει ένα αρχείο VRML είναι το Transform και Shape.
- Κάτω από κάθε κόμβο
- VRML σκηνή/κόσμος
 1. Transform node
 1. Shape node 1
 2. Transform node 1
 1. Shape node 2
 2. Shape node 3
 3. Shape node 4
 2. Shape node 5

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



VRML κόσμος που περιέχει ένα κόκκινο ορθογώνιο παραλληλεπίπεδο με διαστάσεις 3x2x1.

Κ. Δελήμπασης

```

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

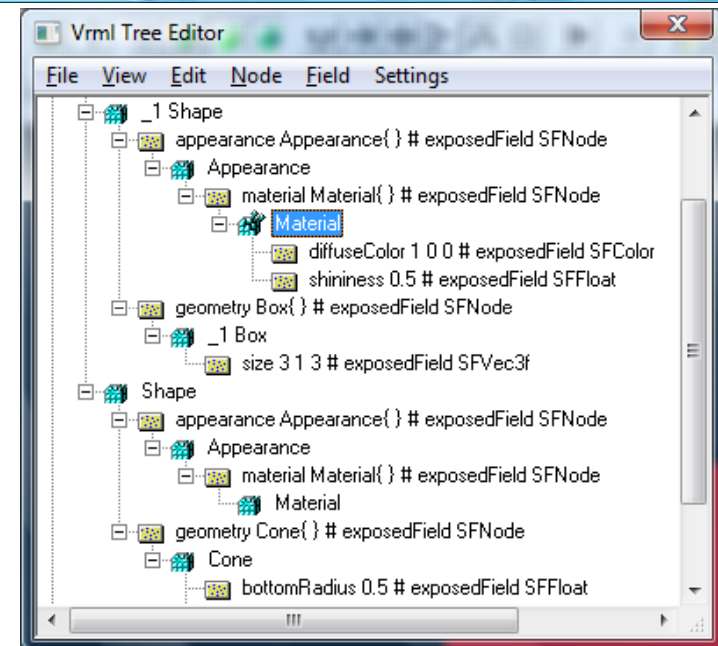
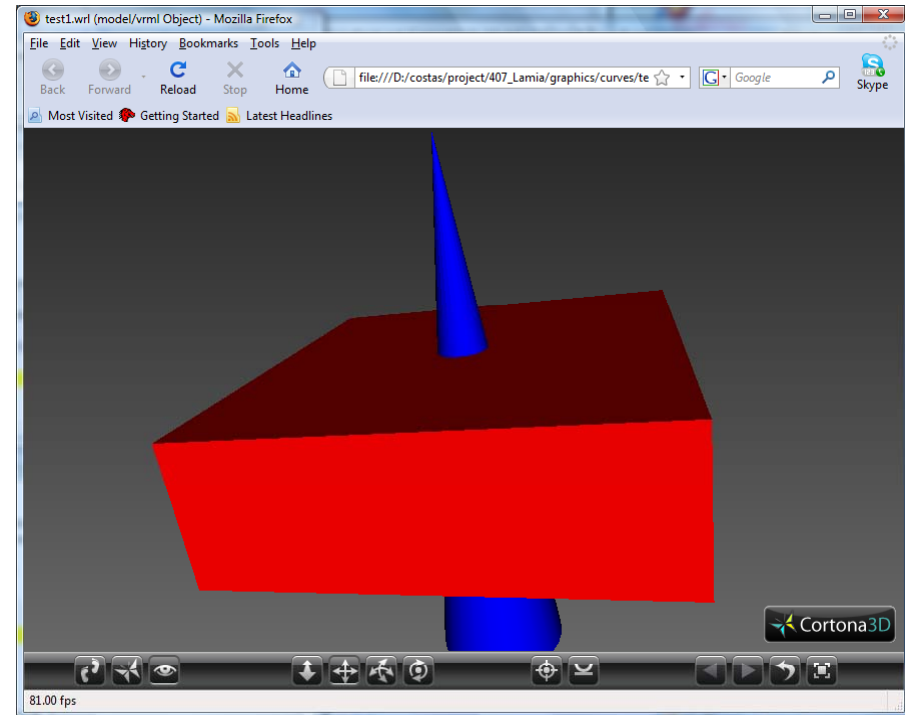
```

```

Shape {
  appearance Appearance {
    material Material {
      diffuseColor 0 0 1
      shininess .9
    }
  }
  geometry Cone {
    bottomRadius 0.5
    height 4
  }}

```

Προσθέτουμε έναν κώνο, μπλε χρώματος. Παρατηρείστε τις αλλαγές στο δέντρο των κόμβων.



Κ. Δελήμασης

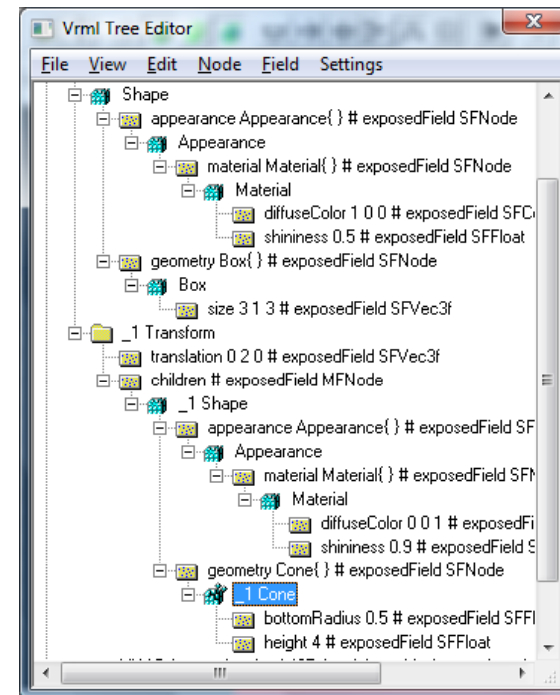
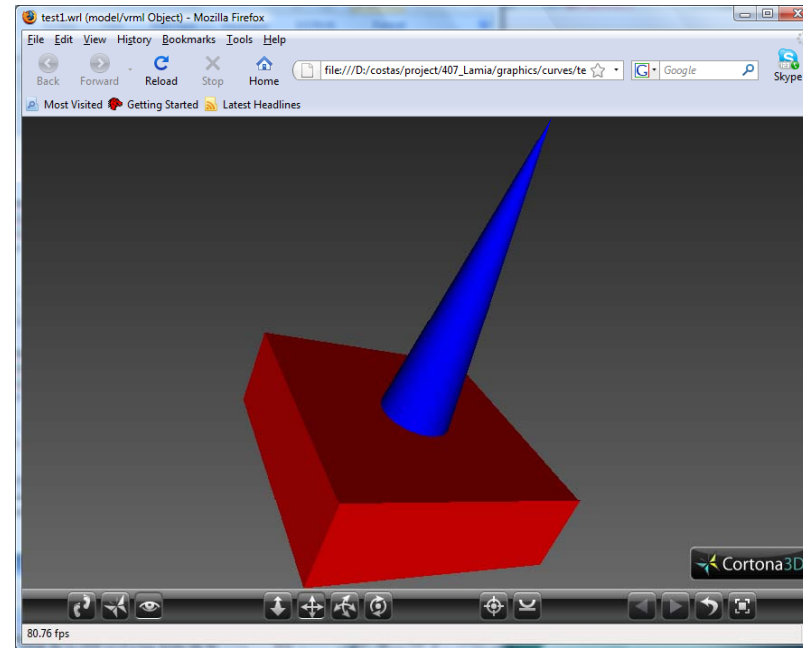

```

#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 1 0 0
      shininess .5
    }
  }
  geometry Box {
    size 3 1 3 }
  Transform {
    translation 0 2 0
    children Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0 0 1
          shininess .9
        }
      }
      geometry Cone {
        bottomRadius 0.5
        height 4
      }
    }
  }
}

```

Εφαρμόζουμε γεωμετρικό μετασχηματισμό στον κώνο. Παρατηρείστε τις αλλαγές στο δέντρο των κόμβων.

Κ. Δελήμασης

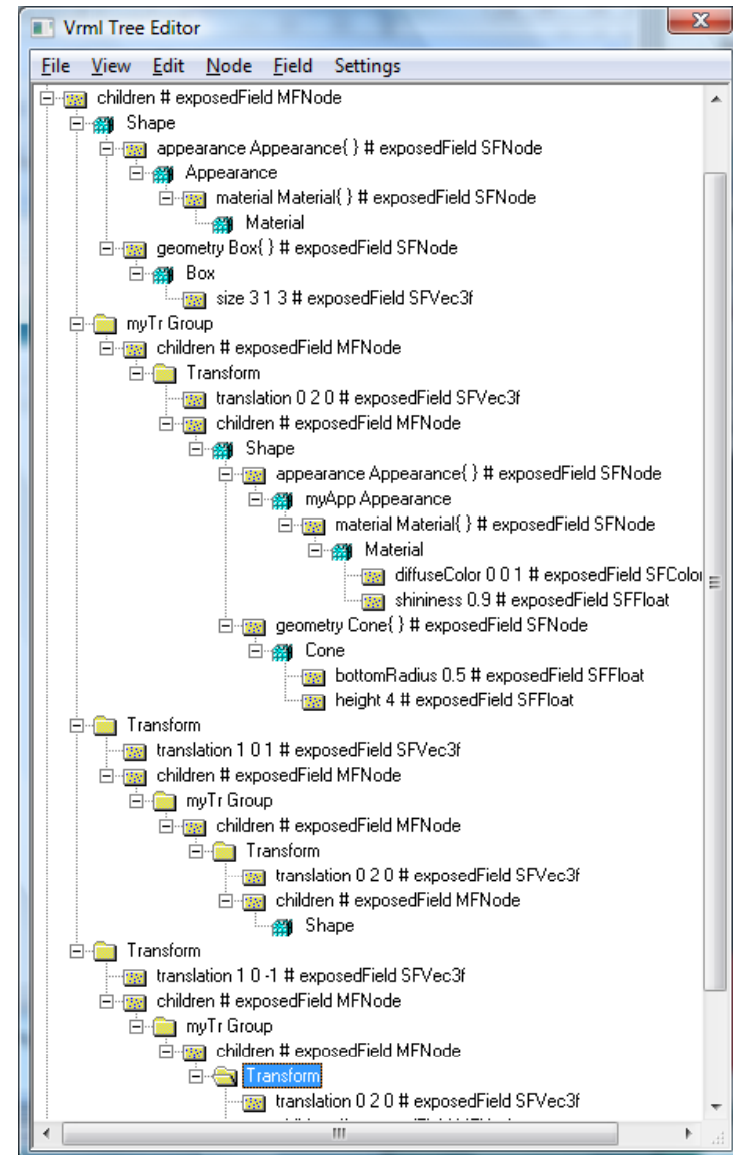
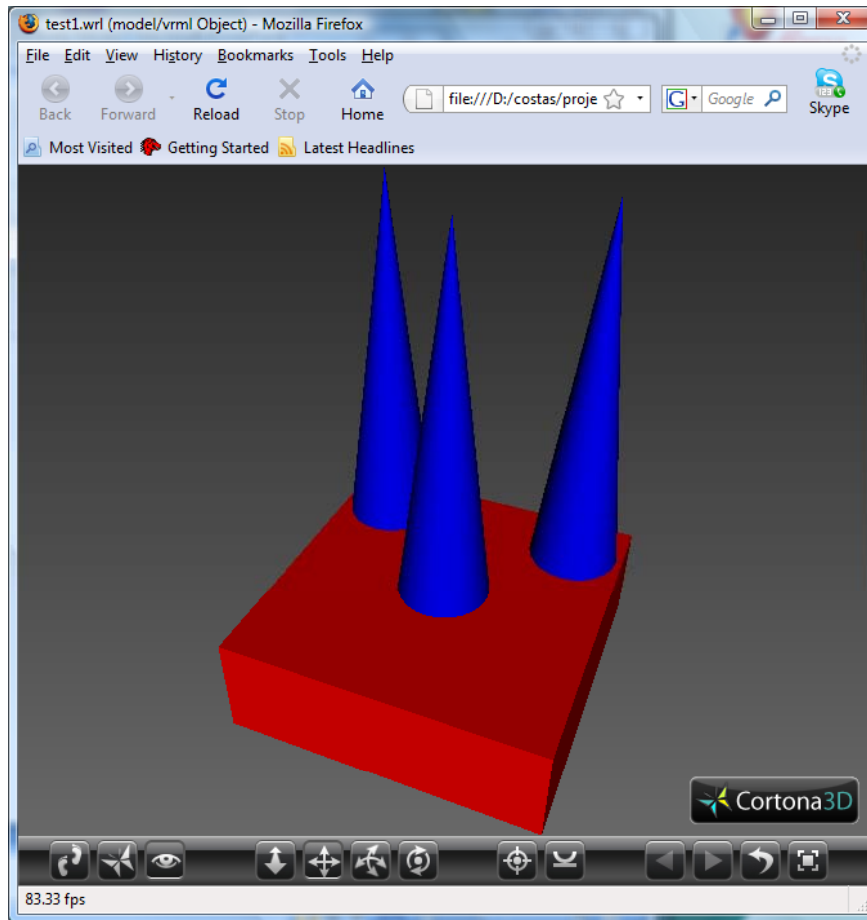


Επαναχρησιμοποίηση VRML κώδικα: DEF και USE

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 1 0 0
      shininess .5
    }
  }
  geometry Box {
    size 3 1 3}
}
DEF myTr Group {
  children [
    Transform {
      translation 0 2 0
      children Shape {
        appearance DEF myApp Appearance {
          material Material {
            diffuseColor 0 0 1
            shininess .9}
        }
        geometry Cone {
          bottomRadius 0.5
          height 4}
      }
  ]
}
Transform {
  translation 1 0 1
  children USE myTr}
Transform {
  translation 1 0 -1
  children USE myTr}
}}}]}
```

Δημιουργούμε άλλους 2 κόνους με το ίδιο χρώμα και ειδιότητες εμφάνισης. Οι 2 νέοι κόνουι είναι μετατοπισμένοι σε σχέση με τον πρώτο. Παρατηρείστε τις αλλαγές στο δέντρο των κόμβων.

Κ. Δελήμπασης



Κ. Δελήμασης

Αναπαράσταση τριγωνοποιημένων επιφανειών με πολύπλοκη γεωμετρία σε VRML

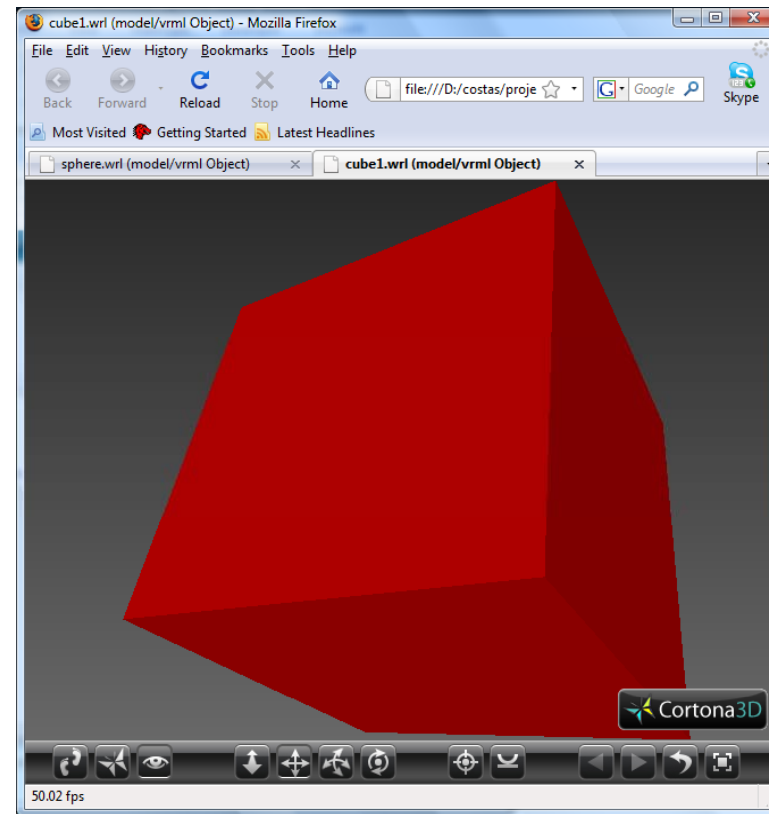
- Τα γεωμετρικά αντικείμενα που είναι διαθέσιμα από τη γλώσσα VRML στον κόμβο `shape`, είναι πολύ περιορισμένα (σφαίρα, κύλινδρος, παραλληλεπίπεδο και κώνος).
- Συχνά είναι απαραίτητο να αναπαριστούμε αντικείμενα με πολύπλοκη γεωμετρία

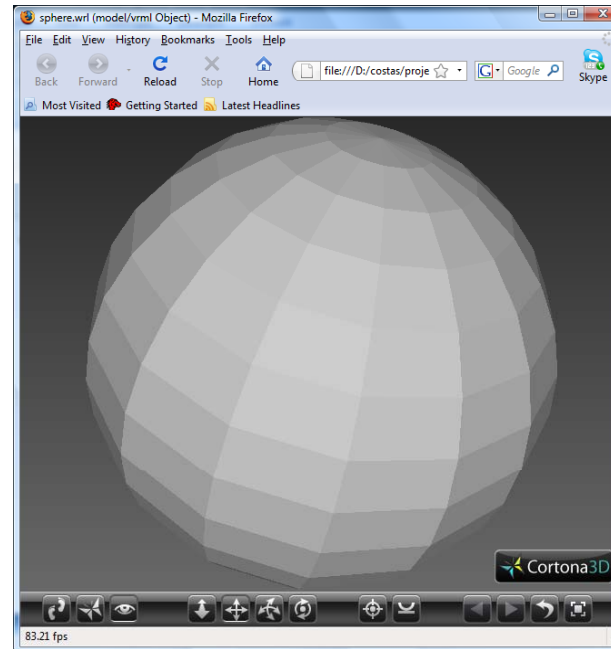
WRL: ο κόμβος IndexedFaceSet

```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 1 0 0 }}
  geometry IndexedFaceSet {
    coord DEF C Coordinate {
      point[
        # λίστα με συντεταγμένες κορυφών
        .....
      ]
    }
    coordIndex [
      # λίστα με ακολουθίες κορυφών που σχηματίζουν πολύγωνα
      .....
    ]
  }
}
```

Παράδειγμα: ο μοναδιαίος κύβος σαν επιφάνεια αποτελούμενη από 6 τετράπλευρα

```
#VRML V2.0 utf8
Shape {
  geometry IndexedFaceSet {
    solid FALSE
    coord Coordinate{
      point [
        0 0 0,
        1 0 0,
        0 1 0,
        1 1 0,
        0 0 1,
        1 0 1,
        0 1 1,
        1 1 1 ] }
    coordIndex [
      0, 1, 3, 2, -1,
      4, 6, 7, 5, -1,
      7, 3, 2, 6, -1,
      1, 5, 4, 0, -1,
      4, 6, 2, 0, -1,
      1, 3, 7, 5, -1]
  }
}
```





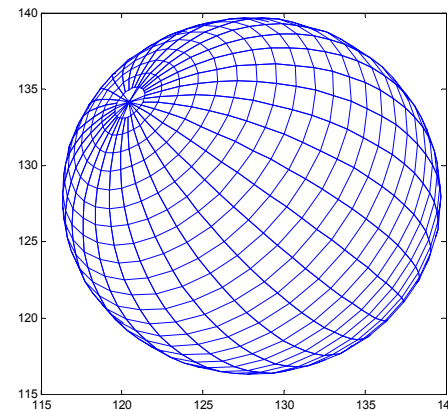
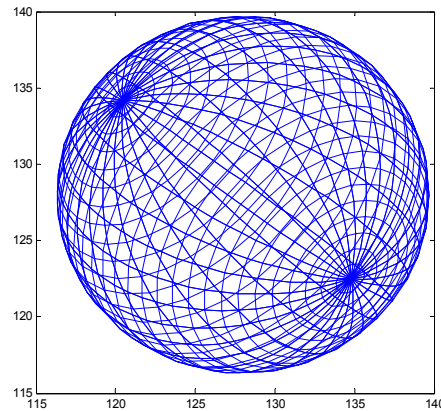
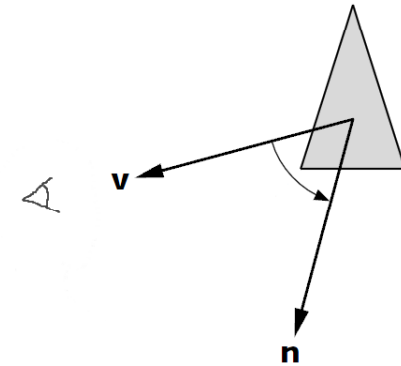
Κ. Δελήμπασης

Προβολή του αντικειμένου: Βασικές αρχές

- Ο απλούστερος αλγόριθμος προβολής:
- Επιλέγεται επίπεδο προβολής και το είδος (κεντρική, παράλληλη κλπ) και υπολογίζεται ο πίνακας προβολής P .
- Ορίζονται οι διαστάσεις του παραθύρου απεικόνισης $(0,0)$, (x_{\min}, y_{\min})
- Για όλα τα σημεία του αντικειμένου (αποθηκευμένα στο vertices):
 - Προβάλλονται τα σημεία και υπολογίζεται το \min και \max της προβολής x_{\min} , y_{\min} , x_{\max} , y_{\max}
 - Υπολογίζεται ο μετασχηματισμός παρατήρησης MP που απεικονίζει το x_{\min} , y_{\min} , x_{\max} , y_{\max} \rightarrow παράθυρο απεικόνισης
- Για κάθε πολύγωνο (face):
 - Προβάλλονται τα σημεία του πολυγώνου (εφαρμογή του Πίνακα P)
 - Συνδέονται τα σημεία του, με τη σειρά που ορίζονται
 - Εφαρμόζεται ο MP

Απλός αλγόριθμος απόκρυψης ακμών για κυρτά αντικείμενα

- Για κάθε ένα πολύγωνο της επιφάνειας
 - Υπολογίζεται το κάθετο διάνυσμα N του επιπέδου
 - Υπολογίζεται το πρόσημο του εσωτερικού γινομένου του διανύσματος προβολής P με το N
 - Αν $P \cdot N > 0 \rightarrow$ πολύγωνο ορατό, αλλιώς πολύγωνο αόρατο



Αλγόριθμοι απόκρυψης ακμών

- Έστω ακμή του αντικειμένου που αποτελείται από 2 σημεία $P_1(x_1, y_1, z_1)$ και $P_2(x_2, y_2, z_2)$
- Χρησιμοποιώντας τον πίνακα προβολής Π , υπολογίζουμε τις προβολές των P_1 και P_2 , $P_1' = \Pi \cdot P_1$, $P_2' = \Pi \cdot P_2$ και την παραμετρική εξίσωση της προβολής $\pi = P_1' + \mu(P_2' - P_1')$, $\mu \in [0, 1]$.
- Υπολογίζουμε την παράμετρο του σημείου τομής μ_i της προβολής k με τις προβολές όλων των άλλων ακμών k_i . (Αν $\mu_i > 0$ και < 1 , τότε οι προβολές των ακμών τέμνονται).
- Διατάσσονται τα μ_i με αύξουσα σειρά.
- Τα διατεταγμένα μ_i ορίζουν τμήματα της ακμής τα οποία είτε είναι πλήρως ορατά ή αόρατα.

- Ελέγχουμε αν κάθε τμήμα της ακμής είναι ορατό, ελέγχοντας το μέσο του $\sigma = (\mu_i + \mu_{i+1})/2$.
- Οι συντεταγμένες της προβολής αυτής είναι $m' = P_1' + \sigma(P_2' - P_1')$. Οι συντεταγμένες του σημείου m που χρησιμοποιείται για την προβολή, δεν μπορούν να υπολογιστούν με χρήση του αντίστροφου του πίνακα προβολής Π διότι αυτός είναι ιδιάζων.
- Το m είναι η τομή της ευθείας προβολής $m = \nu m'$ και της παραμετρικής εξίσωσης της ακμής $m = P_1 + \mu(P_2 - P_1)$.

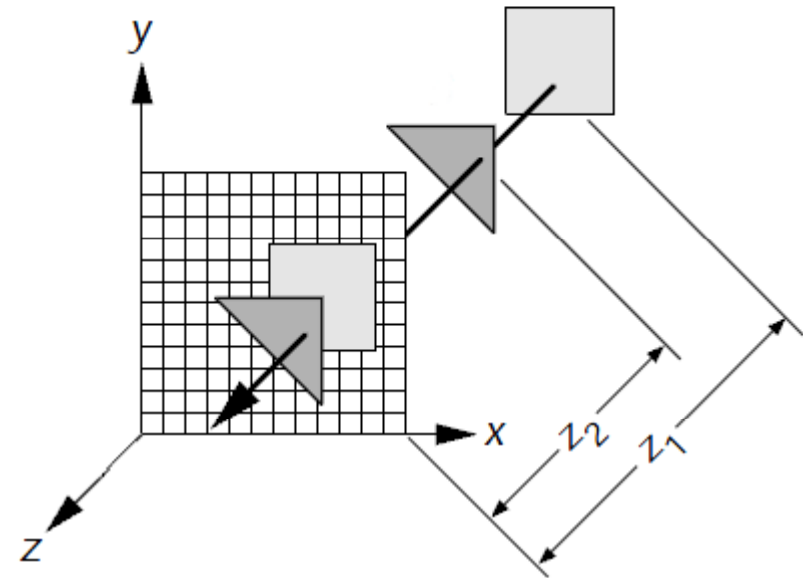
- Αφού υπολογιστεί η τιμή της παραμέτρου μ (άρα και το σημείο m):
 - Εντοπίζονται όλα τα πολύγωνα του αντικειμένου που δεν περιέχουν την ακμή k
 - Υπολογίζεται η εξίσωση του επιπέδου του πολυγώνου
 - Υπολογίζεται το σημείο τομής της ευθείας που συνδέει το σημείο προβολής με το επίπεδο του πολυγώνου και άρα η παράμετρος w του σημείου τομής
 - Αν $w > 1 \rightarrow$ το επίπεδο βρίσκεται πίσω από το σημείο $m \rightarrow$ δεν μπορεί να κρύβει την ακμή k
 - Αν $0 < w < 1 \rightarrow$ το επίπεδο βρίσκεται ανάμεσα στο σημείο m και το σημείο προβολής \rightarrow είναι δυνατό να κρύβει την ακμή k . Τότε:

- Προβάλλουμε όλα τα σημεία του πολυγώνου (πίνακας Π)
- Ελέγχουμε αν το σημείο m' βρίσκεται μέσα στην προβολή του πολυγώνου. Αν ναι, τότε όλο το τμήμα της προβολής $P_1' P_2'$ της ακμής που ορίζεται από τις παραμέτρους μ_i, μ_{i+1} είναι κρυμμένο, αλλιώς είναι ορατό.

Αλγόριθμος απόκρυψης επιφανειών: Z-buffer

- Ο εν λόγω αλγόριθμος αποκρύπτει επιφάνειες αντί για ακμές και σήμερα αποτελεί ίσως τον πιο δημοφιλή (παλιότερα δεν χρησιμοποιείτο λόγω υψηλών για την εποχή απαιτήσεων μνήμης)

- Για κάθε pixel του frame buffer (μνήμη οθόνης) τηρείται μία ταξινομημένη λίστα των πολυγώνων που προβάλλονται σε αυτό, καθώς και της χρωματικής τους τιμής (όπως προκύπτει μετά την εφαρμογή του μοντέλου φωτισμού).



- Κάθε φορά που βρίσκεται πολύγωνο πιο κοντά σε κάποιο pixel, τότε το χρώμα του και η απόσταση του από αυτό (Z συντεταγμένη) ενημερώνουν τις ιδιότητες του pixel.

- Ο ψευδοκώδικας του αλγορίθμου ακολουθεί

Κ. Δελημπασης

```

// Αρχικοποίηση
for all pixels p in frame_buffer
    z_buffer(p)=max;
    frame_buffer(p)=c;
end
for all polygons π in faces
    προβολή πολυγώνου π στο επίπεδο
    for all scan lines (γραμμή σάρωσης)
        for all pixels p στη γραμμή σάρωσης
            Υπολογισμός του Z του σημείου τομής της γραμμής
            σάρωσης με το π (διπλή γραμμική παρεμβολή)
            if z<z_buffer(p) then
                z_buffer(p)=z;
                frame_buffer(p)=color(π);
            end
        end
    end
end
end
end

```

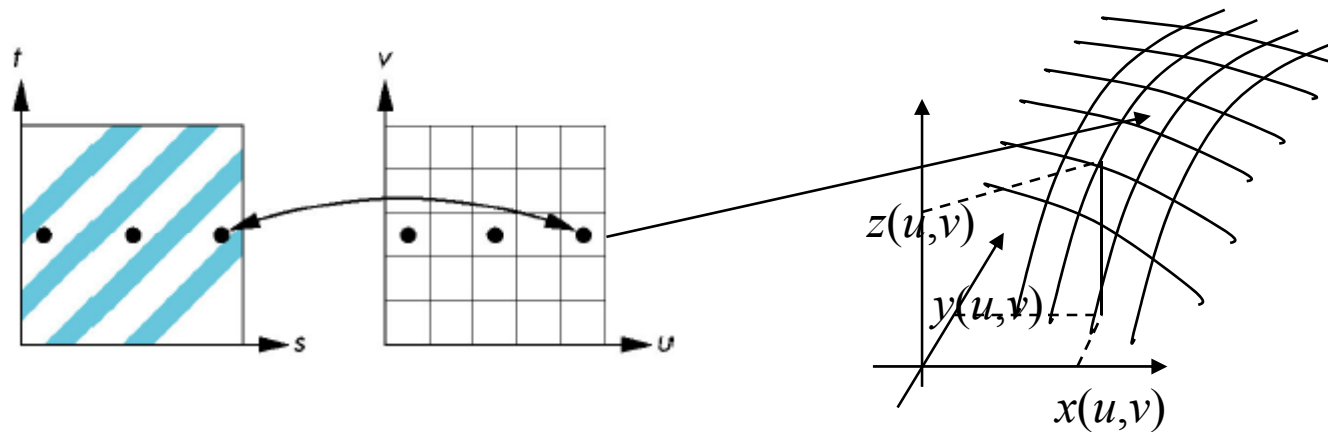
- Ο z-buffer έχει πολυπλοκότητα $O(NP)$, P αριθμός pixel/πολύγωνο και N αριθμός πολυγώνων. Το γινόμενο NP είναι σχεδόν ανεξάρτητο από την πολυπλοκότητα της σκηνής που απεικονίζεται.
- Ο z-buffer χειρίζεται με δυσκολία επιφάνειες με βαθμό διαφάνειας.

Απεικόνιση υφής – Texture mapping

- Η δισδιάστατη υφή είναι δισδιάστατη ψηφιακή εικόνα που βρίσκεται στη μνήμη σαν πίνακας
- Θεώρηση ως συνεχούς εικόνας $T(s,t)$, όπου s,t συντεταγμένες υφής
- Πρέπει να απεικονίσουμε την $T(s,t)$ στην επιφάνεια (x,y,z) .

Άπευθείας απεικόνιση του παραμετρικού χώρου της εικόνας υφής στον παραμετρικό χώρο της επιφάνειας

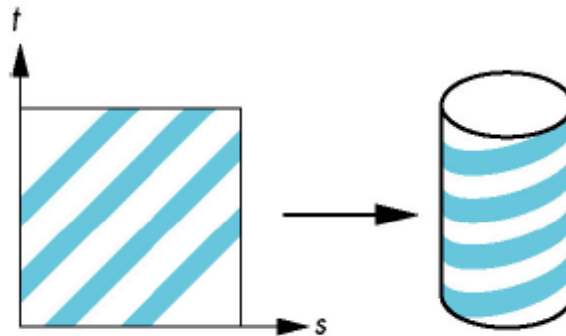
- Ο παραμετρικός χώρος (t,s) της εικόνας υφής απεικονίζεται στον παραμετρικό χώρο της επιφάνειας (u,v) , μέσω μίας συνάρτησης T (συνήθως γραμμικής)
- Στην γραμμική περίπτωση $u=a_{11}s+a_{12}t+a_{13}$, $v=a_{21}s+a_{22}t+a_{23}$



Απεικόνιση υφής μέσω ενδιάμεσης επιφάνειας

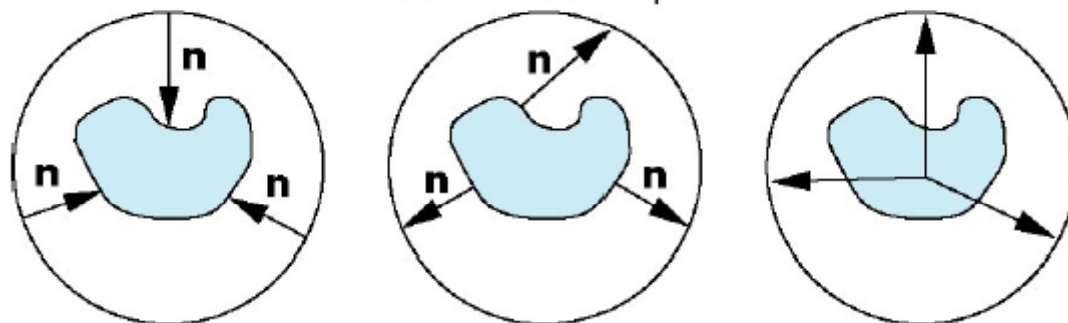
- Η μέθοδος αυτή χρησιμοποιεί μία ενδιάμεση επιφάνεια η οποία πρέπει να είναι τοπολογικά όμοια με την επιφάνεια στην οποία θα γίνει τελικά η απεικόνιση υφής.
- Σε δεύτερο βήμα απεικονίζεται η ενδιάμεση επιφάνεια στην τελική επιφάνεια

- Προβολή σε ενδιάμεσο κύλινδρο
 - Αν $0 < s, t < 1$ οι παράμετροι του χώρου υφής και $0 < u, v < 1$ οι παράμετροι του της ενδιάμεσης επιφάνειας
 - $x = r \cos(2\pi u)$, $y = r \sin(2\pi u)$, $z = v/h$
 - Απεικόνιση: $u = s$, $t = v$
 - Απεικόνιση χωρίς παραμόρφωση για επιμήκη αντικείμενα



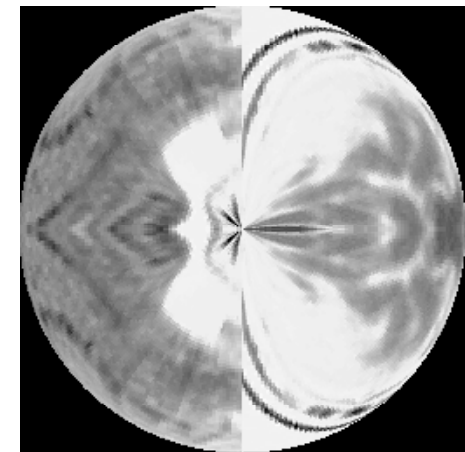
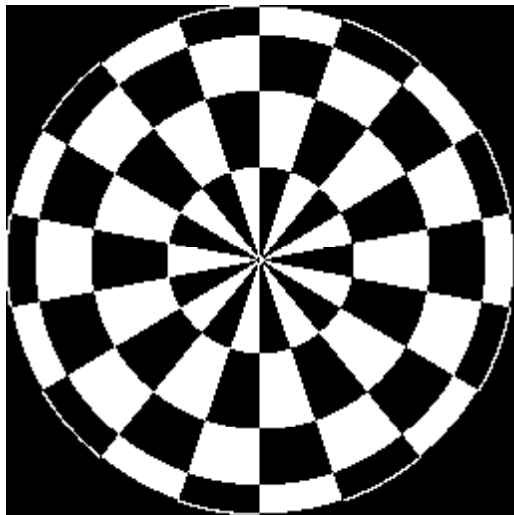
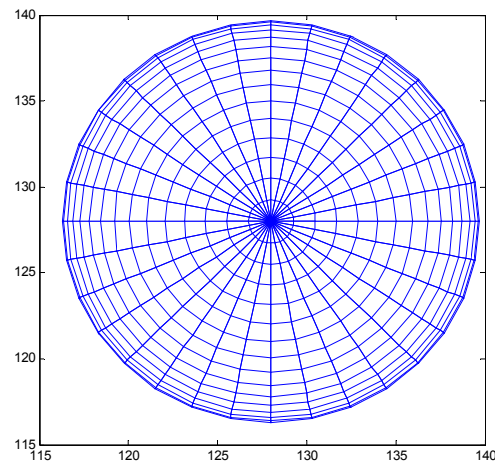
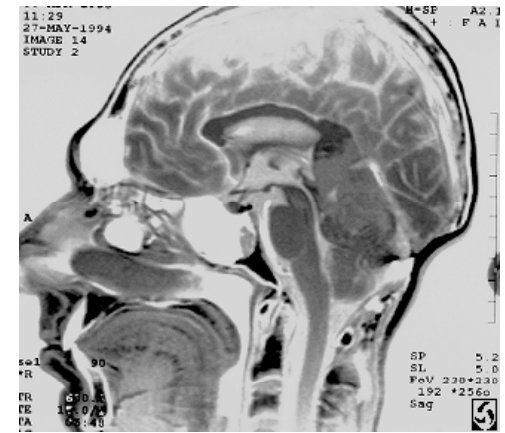
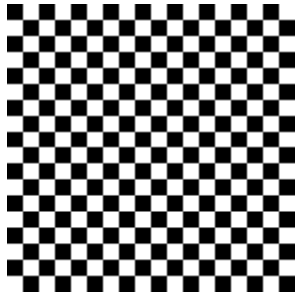
Προβολή από την ενδιάμεση επιφάνεια στην τελική επιφάνεια

- Τρεις διαφορετικές προσεγγίσεις:
 - Σύμφωνα με το κάθετο διάνυσμα της ενδιάμεση επιφάνειας
 - Σύμφωνα με το κάθετο διάνυσμα της τελικής επιφάνειας
 - Ως προς σταθερό σημείο προβολής



Απεικόνιση υφής – Texture mapping

- Για κάθε pixel (i,j) του παραθύρου απεικόνισης
 - Εφαρμόζουμε τον αντίστροφο MP
 - Εντοπίζεται το σημείο τομής της ευθείας προβολής στο τρέχον Pixel με το αντικείμενο της ενδιάμεσης επιφάνειας
 - Υπολογίζεται το ζεύγος παραμέτρων (u,v) του σημείου τομής
 - Υπολογίζεται το pixel της εικόνας που αντιστοιχεί στο (u,v) και απεικονίζεται η τιμή της στο (i,j) .



Κ. Δελήμασης

Επιφάνειες Bezier

- Ορίζονται $(m+1) \times (n+1)$ σημεία ελέγχου –control points- P_{ij} .
Χρησιμοποιούμε τη βάση συναρτήσεων των καμπύλων Bezier.

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_{m,i}(u) B_{n,j}(v) P_{ij}$$

$$B_{m,i}(u) = \frac{m!}{i!(m-i)!} u^i (1-u)^{m-i}$$

$$B_{n,j}(v) = \frac{n!}{j!(n-j)!} v^j (1-v)^{n-j}$$

- Έστω ο αρχικός ορισμός της καμπύλης Bezier με βαθμό m , παράμετρο t και $m+1$ σημεία ελέγχου: $P_i^0, i=0\dots m$.

$$P^m(t) = \sum_{i=0}^m B_i^m(t) P_i^0$$

- Ας θεωρήσουμε ότι τα κάθε ένα από τα σημεία ελέγχου P_i^0 αποτελούν σημεία που παράγονται από άλλη καμπύλη Bezier, με βαθμό n , παράμετρο u και $n+1$ σημεία ελέγχου: $P_{ij}^0, j=0\dots n$.

$$P_i^0 = P^n(u) = \sum_{j=0}^n B_j^n(u) P_{ij}^0$$

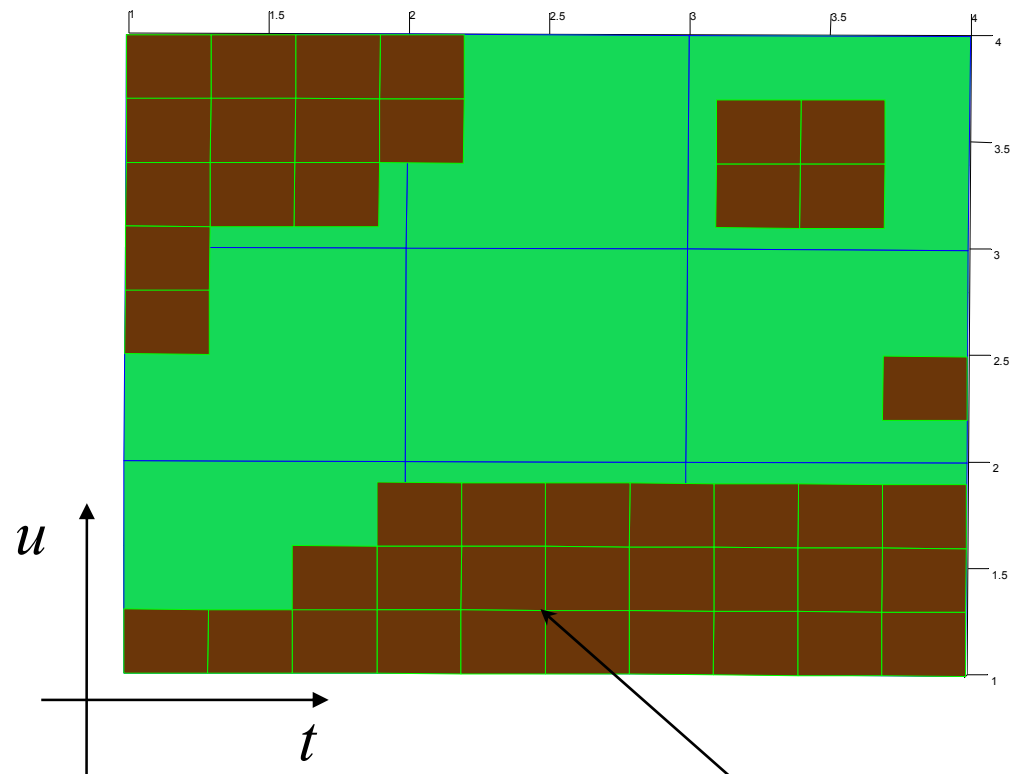
- Συνδυάζουμε τα παραπάνω:

$$P^{m,n}(t,u) = \sum_{i=0}^m B_i^m(t) P_i^0 = \sum_{i=0}^m B_i^m(t) \left(\sum_{j=0}^n B_j^n(u) P_{ij}^0 \right) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(t) B_j^n(u) P_{ij}^0$$

Αλγόριθμος De Casteljau για επιφάνειες Bezier

- Έστω $n \times m$ P_{ij}^0 σημεία ελέγχου της επιφάνειας
- for $i=1$ to n
 - $P_i = \text{Bezier}(P_{ij}^0, u)$ // κατασκευάζουμε το σημείο που αντιστοιχεί σε καμπύλη Bezier με σημεία ελέγχου την γραμμή i και παράμετρο u
- end;
- $P_{uv} = \text{Bezier}(P_i, t)$ // χρησιμοποιούμε τα P_i ως σημεία ελέγχου για την κατασκευή του σημείου P_{uv}

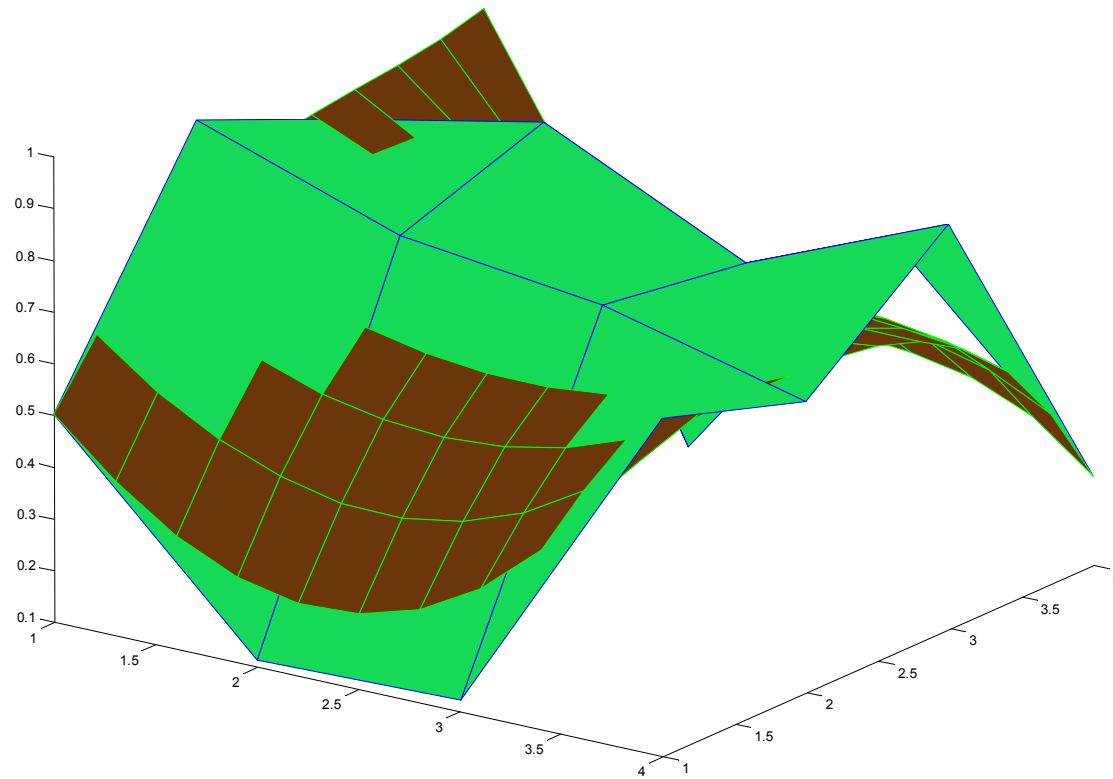
Παράδειγμα επιφάνειας Bezier υπολογισμένης σε 11x11
σημεία, με 4x4 σημεία ελέγχου



$$f(u,v)$$

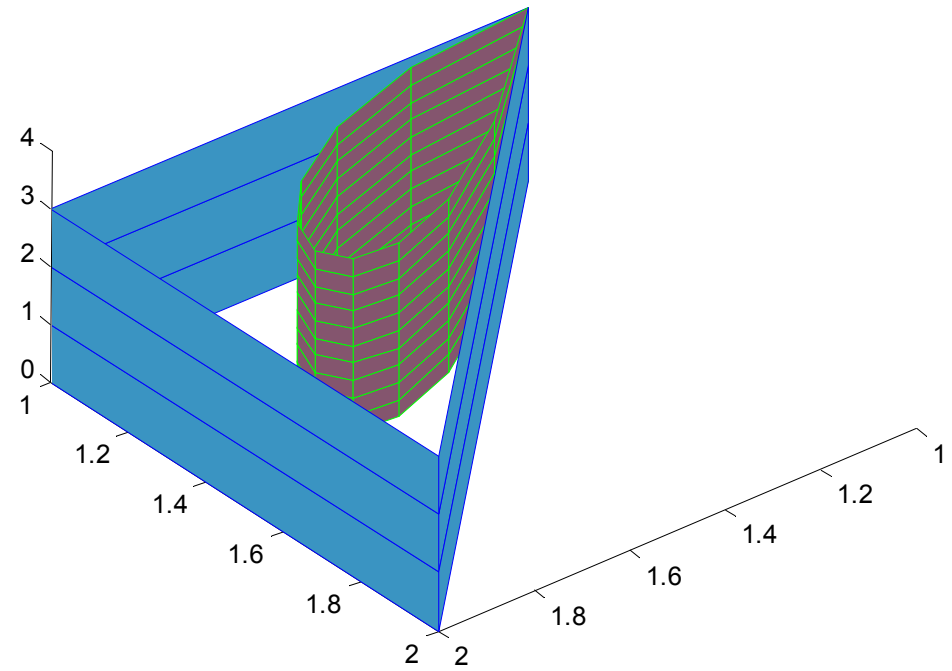
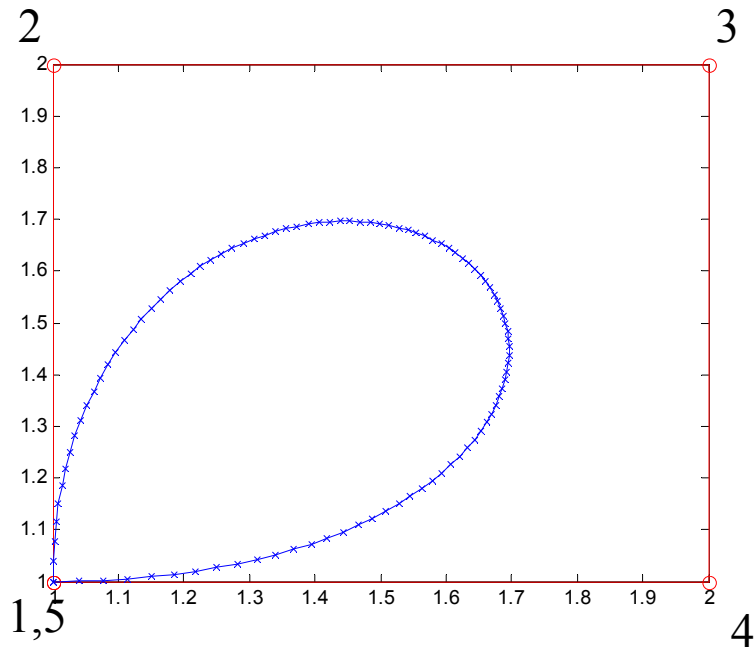
Κ. Δελήμπασης

Παράδειγμα επιφάνειας Bezier υπολογισμένης σε 11x11
σημεία, με 4x4 σημεία ελέγχου



Κ. Δελήμασης

Bezier 2D \rightarrow 3D

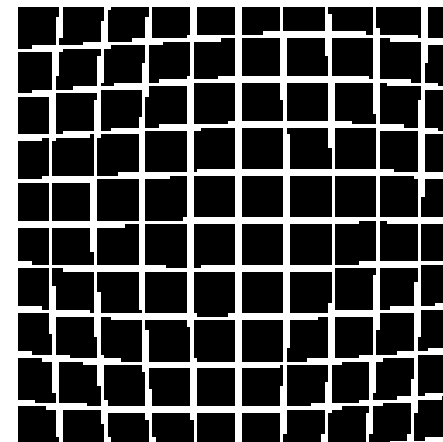
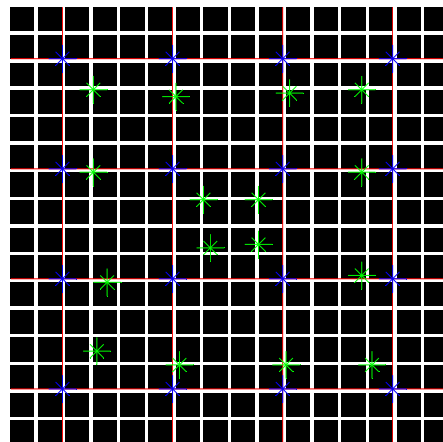
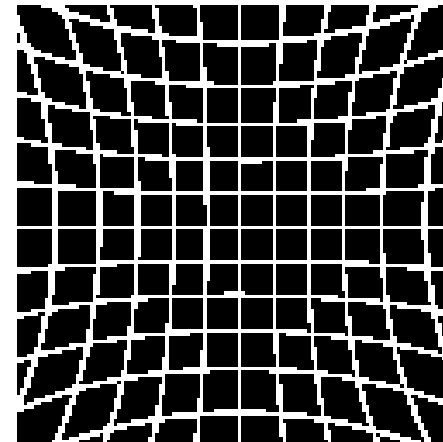
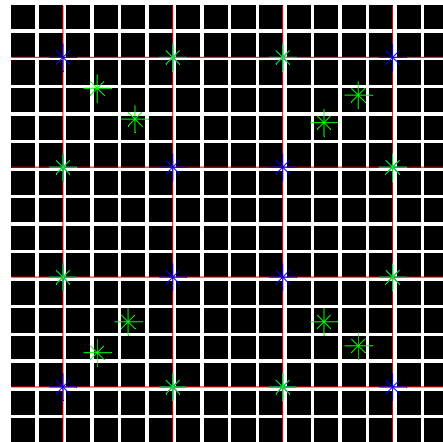


Κ. Δελήμπασης

Εφαρμογή των επιφανειών Bezier και B-Spline για παραμόρφωση εικόνων

- Εστω εικόνα I , N γραμμών M στηλών. Είναι δυνατό να χρησιμοποιηθούν οι αλγόριθμοι παραγωγής επιφανειών για παραμόρφωση της εικόνας:
 - Ορίζεται ένα σύνολο n_{ctrl} σημείων ελέγχου, διασκορπισμένα στην επιφάνεια της εικόνας
 - Ορίζεται η αντιστοιχία μεταξύ του χώρου της εικόνας και του χώρου των παραμέτρων τα επιφάνειας μέσω μίας γραμμικής συνάρτησης:
$$f : (x, y) \in [1, N] \times [1, M] \rightarrow (u, v) \in [0, 1] \times [0, 1]$$
 - Εφαρμόζεται ο αλγόριθμος De Castelau στις διαστάσεις X, Y
 - Εκτελείται ο ακόλουθος ψευδοκώδικας:

- For each pixel \mathbf{p} in I
 - $(u,v)=f(\mathbf{p})$
 - $\mathbf{p}'=\text{De Castelau}(\mathbf{p})$
 - $I1(\mathbf{p}')=I(\mathbf{p})$
- Ο παραπάνω ψευδοκώδικας παρουσιάζει φαινόμενα οπών στην παραγόμενη εικόνα $I1$, όπως είδαμε στην απλή περιστροφή εικόνας.
- Ο αντίστροφος πίνακας του μετασχηματισμού δεν υπολογίζεται εν γένει. Η απλούστερη λύση είναι η αντιστροφή των σημείων ελέγχου, και η εναλλαγή της $I1$ με την I .

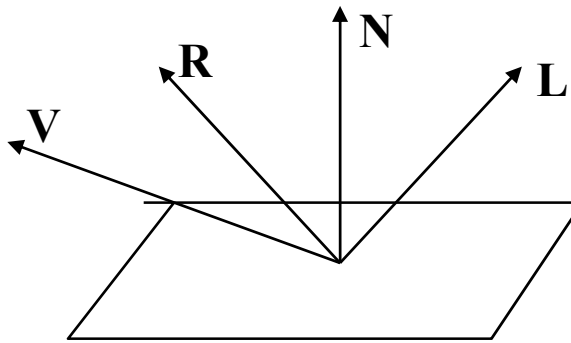


- Παραδείγματα παραμόρφωσης εικόνας με επιφάνειες Bezier.
- *: ομογενής κατανομή σημείων ελέγχου επιφάνειας
- *: τελικά σημεία ελέγχου επιφάνειας

Κ. Δελήμπασης

Μοντέλο φωτισμού

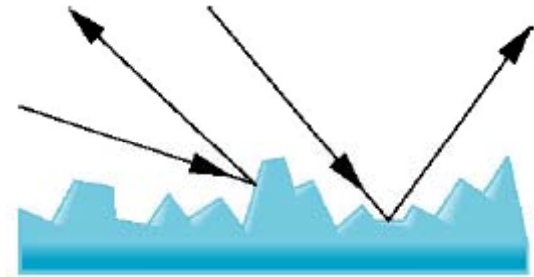
- Ο τρόπος υπολογισμού του χρώματος μίας επιφάνειας, βάσει του προσανατολισμού της σε σχέση με το διάνυσμα προσπίπτοντος φωτός L και το διάνυσμα παρατήρησης του παρατηρητή V .
- **Μοντέλο ανάκλασης:** αποτελεί τμήμα του μοντέλου φωτισμού
- **Μοντέλο ανάκλασης Phong:** αποτελείται από 3 συνιστώσες:
 - Διάχυτη ανάκλαση: εξαρτάται από L , N
 - Κατευθυνόμενη ανάκλαση: εξαρτάται από R , V
 - Έμμεσος φωτισμός (ambient light)



Κ. Δελήμπασης

Διάχυτη ανάκλαση – Diffuse Reflection

- Επιφάνειες διάχυτης ανάκλασης ανακλούν το φως ομοιόμορφα προς κάθε κατεύθυνση, λόγω ανωμαλιών
- Τέλεια διαχυτικές επιφάνειες: σύμφωνα με το Νόμος Lambert το R_d είναι ανάλογο του $\cos\theta$, βλέπουμε την κάθετη συνιστώσα του προσπίπτοντος φωτός
- Η θέση του θεατή δεν παίζει ρόλο
- k_d : ποσοστό διαχυτικού φωτός που ανακλάται (το υπόλοιπο απορροφάται).
- Είναι δυνατή η προσθήκη εξασθένησης με την απόσταση d από τον παρατηρητή.
- Μόνο με διάχυτη ανάκλαση και ανάκλαση περιβάλλοντος φωτός τα αντικείμενα φαίνονται μουντά, χωρίς φωτεινές κηλίδες (highlights) με χρώμα αυτό της φωτεινής πηγής.

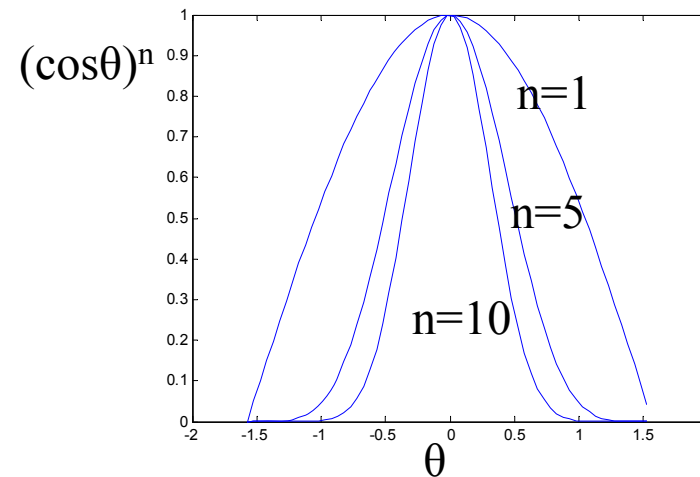


$$I_d = \frac{I_i}{d + d_0} (k_d \vec{L} \cdot \vec{N})$$

Κατευθυντική ανάκλαση - Specular Reflection

- Οι επιφάνειες κατευθυντικής ανάκλασης είναι λείες.
- Το φως από κατευθυντική ανάκλαση που βλέπει ο παρατηρητής εξαρτάται από τη γωνία φ μεταξύ της κατεύθυνσης παρατήρησης V και της κατεύθυνσης τέλει ανάκλασης R .
- N : συντελεστής γυαλάδας
 - Μικρό: ανάκλαση ορατή από μεγάλο εύρος γωνιών
 - Μεσαίο (<100) απλωμένες γυαλιστερές κηλίδες)
 - Μεγάλο: μεταλλικές επιφάνειες
 - Άπειρο: καθρέπτες

$$I_s = \frac{I_i}{d + d_0} \left(k_s (\vec{R} \cdot \vec{V})^n \right)$$



Παράδειγμα κατευθυντικής ανάκλασης



$n=3$

$n=5$

$n=10$

$n=27$

$n=100$

- Οι 3 συνιστώσες του **Μοντέλου ανάκλασης Phong** συνδυάζονται για τον υπολογισμό της τιμής φωτισμού

$$I = I_a + I_d + I_s$$

$$I = I_a k_a + \frac{I_i}{d + d_0} \left(k_d \vec{L} \cdot \vec{N} + k_s (\vec{R} \cdot \vec{V})^n \right)$$

Υπολογισμός του κάθετου διανύσματος N

- Αν η επιφάνεια εκφράζεται αναλυτικά (πχ παραμετρική, bezier, spline κλπ), τότε το N υπολογίζεται βάσει των παραγώγων της
- Αν η επιφάνεια αποτελείται από τρίγωνα / πολύγωνα, τότε το N υπολογίζεται:
 - Ως εξωτερικό γινόμενο

$$\vec{N} = (p_{i+1} - p_i) \times (p_i - p_{i-1})$$

- Από την εξίσωση του επιπέδου που διέρχεται από το πολύγωνο (ακόμα και αν τα σημεία του πολυγώνου δεν είναι συνεπίπεδα).

$$a = \sum_{i=1}^n (y_i - y_{i+1})(z_i + z_{i+1})$$

$$b = \sum_{i=1}^n (z_i - z_{i+1})(x_i + x_{i+1})$$

$$c = \sum_{i=1}^n (x_i - x_{i+1})(y_i + y_{i+1})$$

Καθιστούμε το N μοναδιαίο

Κ. Δελήμπασης

- Αν έχουμε 3 σημεία στο χώρο, τότε το κάθετο διάνυσμα προκύπτει από την εξίσωση του επιπέδου που διέρχεται από αυτά:

$$\begin{aligned} ax_1 + by_1 + cz_1 &= -1 \\ ax_2 + by_2 + cz_2 &= -1 \\ ax_3 + by_3 + cz_3 &= -1 \end{aligned} \quad \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

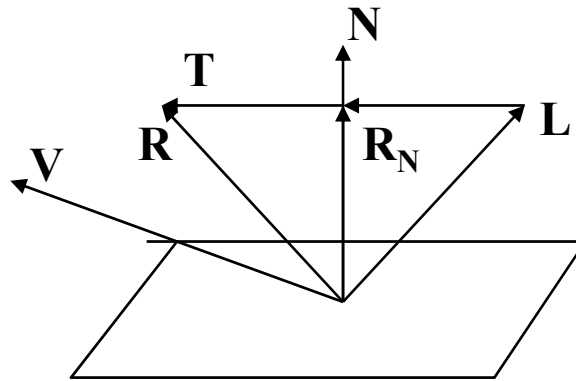
Υπολογισμός του διανύσματος ανάκλασης R

- Το R υπολογίζεται σε κάθε κορυφή ή σημείο της επιφάνειας \rightarrow έχει σημασία η ταχύτητα.

$$R_N = N(L \cdot N)$$

$$L + T = R_N \rightarrow T = R_N - L$$

$$R = R_N + T = 2N(L \cdot N) - L$$

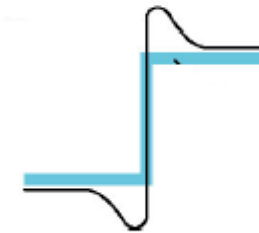


- Εναλλακτικός υπολογισμός:
 - Ταυτίζουμε το \mathbf{N} με το \mathbf{Z} βάσει μετασχηματισμού A
 - Υπολογίζουμε το νέο $\mathbf{L}' = A\mathbf{L}$
 - $\mathbf{R} = A^{-1}(-L'_x, -L'_y, Z'_z)$

Αλγόριθμος σταθερού φωτισμού – flat shading

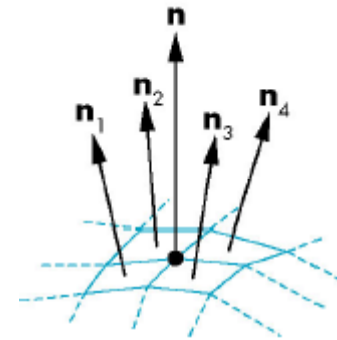
- Συνιστώσες: υπολογίζονται μία φορά για κάθε τρίγωνο
- Υπολογίζεται ένα μοναδικό χρώμα για όλο το πολύγωνο (βάσει μοντέλου Phong).
- Ισοδυναμεί με σταθερά N , V , L
 - Διάχυτη ανάκλαση (εξαρτάται από L , N): NAI
 - Κατευθυνόμενη ανάκλαση (εξαρτάται από R , V): OXI
 - Έμμεσος φωτισμός (ambient light): NAI
 - Συνάρτηση από απόσταση από παρατηρητή: OXI

- Απογοητευτικά αποτελέσματα αν χρησιμοποιούμε πολύγωνα για να μοντελοποιήσουμε ομαλές επιφάνειες.
- Γειτονικά πολύγωνα με διαφορετικό χρώμα \rightarrow οπτικά υπερτονίζεται η διαφορά χρώματος στο όριο τους (Φαινόμενο Mach band)
- Καλά αποτελέσματα πχ σε τριγωνοποιημένες επιφάνειες (Marching Cubes)
- Ταχύτατη υλοποίηση

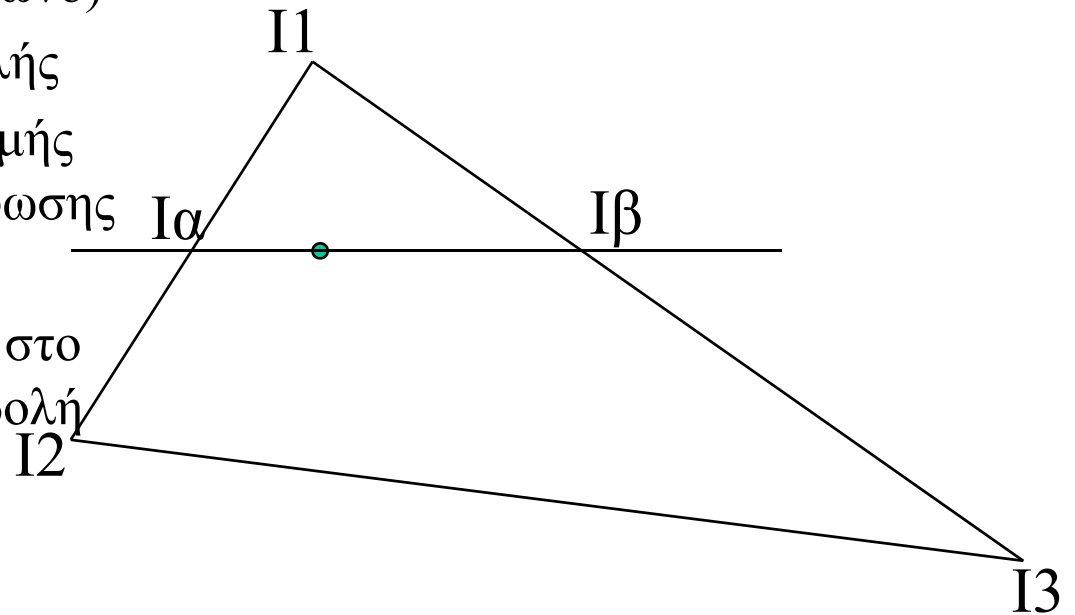


Αλγόριθμος φωτισμού Gouraud

- Χρησιμοποιείται το μοντέλο ανάκλασης του flat shading
- Υπολογίζεται το N στις κορυφές των τριγώνων και βάσει αυτών η φωτεινότητα. Ο υπολογισμός γίνεται σαν μέσος όρος των N των πολυγώνων που περιέχουν την κορυφή.
- Η φωτεινότητα στο εσωτερικό του τριγώνου / πολυγώνου δεν είναι σταθερή, αλλά υπολογίζεται για κάθε pixel με παρεμβολή (interpolation)
- Υλοποίηση με matlab: μετά τον υπολογισμό της φωτεινότητας σε κάθε κορυφή, η παρεμβολή μπορεί να γίνει αυτόματα

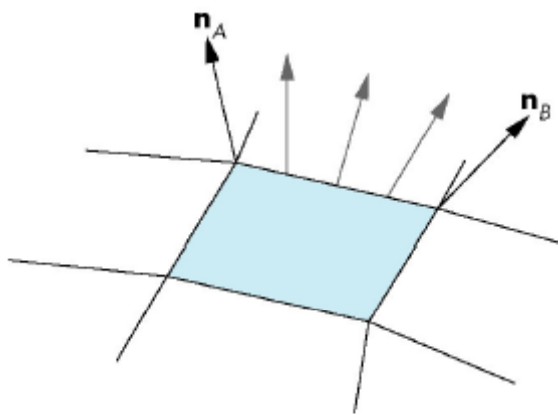


- Προβάλλουμε το πολύγωνο (τρίγωνο)
- Για κάθε pixel εντός της προβολής
 - Υπολογίζουμε τα σημεία τομής της οριζόντιας γραμμής σάρωσης με τις πλευρές του
 - Υπολογίζουμε την ένταση I στο σημείο με γραμμική παρεμβολή

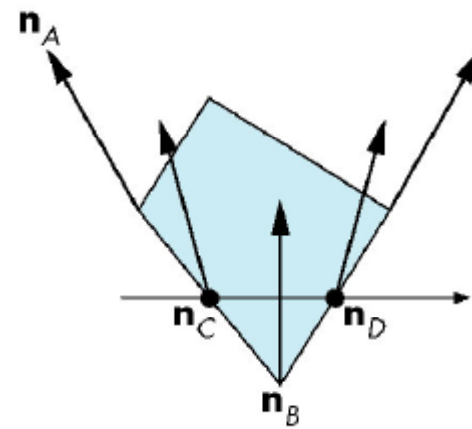


Αλγόριθμος φωτισμού Phong

- Μέσος όρος για τον υπολογισμό των διανυσμάτων στις κορυφές
- Παρεμβολή διανυσμάτων σε ακμές και στη συνέχεια στο εσωτερικό
- Ένα κάθετο διάνυσμα για κάθε σημείο του πολυγώνου.
- Υπολογισμός της εξίσωσης φωτισμού σε κάθε σημείο ανεξάρτητα
- Καλύτερη σκίαση αλλά εξαιρετικά χρονοβόρα.
- Δεν παρέχεται άμεσα από την OpenGL
- Γίνεται συνήθως off-line



Παρεμβολή των N κατά μήκος της ακμής



Παρεμβολή των N κατά μήκος της γραμμής σάρωσης

Παράδειγμα

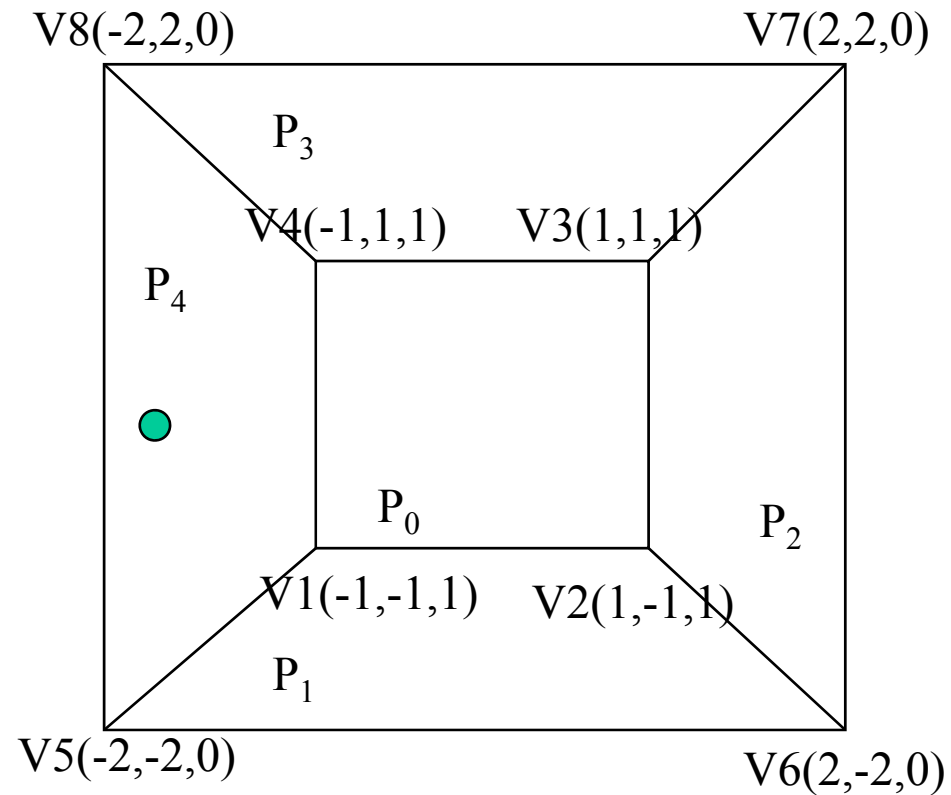
- Υπολογίστε την ένταση του φωτισμού στο σημείο Γ.

$$\mathbf{L} = (0,0,1), \mathbf{V} = (-1,-1,1)$$

$$d = 0, d_0 = 1$$

$$I_a = 1, I_1 = 1$$

$$n = 3, k_s = 0.7, k_d = k_a = 0.25$$



Υπολογισμός με τον αλγόριθμο σταθερού φωτισμού

- Υπολογίζουμε την εξίσωση του επιπέδου $V_1V_4V_8V_5$ λύνοντας το σύστημα των εξισώσεων:

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_6 & y_6 & z_6 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

- $N=[a,b,c]=[-2, 0, 2]$
- Μοναδοποίηση N : $N=[-1/\sqrt{2}, 0, 1/\sqrt{2}]$
- Ομοίως $N=[-1/\sqrt{2}, 0, 1/\sqrt{2}]$

$$N = \frac{(V_4 - V_1) \times (V_8 - V_4)}{|(V_4 - V_1) \times (V_8 - V_4)|}$$

$$L \cdot N = (0, 0, 1) \left(-\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}} \right) = \frac{1}{\sqrt{2}} = 0.5774$$

$$R = 2N(LN) - L = 2 \left(-\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}} \right) \left(\frac{1}{\sqrt{2}} \right) - (0, 0, 1) = (-1, 0, 0)$$

$$R \cdot V = (0, 0, 1) \left(-\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}} \right)$$

$$I = I_a k_a + \frac{I_i}{d + d_0} \left(k_d \vec{L} \cdot \vec{N} + k_s (\vec{R} \cdot \vec{V})^n \right)$$

Κ. Δελήμπασης

Υπολογισμός με τον αλγόριθμο φωτισμού Gouraud

- Υπολογίζουμε τα κάθετα διανύσματα στα επίπεδα P_0, P_1, P_3, P_4 .

$$N_{P_0} = (0, 0, 1)$$

$$N_{P_1} = \frac{(V_5V_6) \times (V_6V_2)}{|(V_5V_6) \times (V_6V_2)|} = (0, -4, 4) \frac{1}{\sqrt{32}} = (0, -0.7071, 0.7071)$$

$$N_{P_3} = \frac{(V_3V_7) \times (V_7V_8)}{|(V_3V_7) \times (V_7V_8)|} = (0, 4, 4) \frac{1}{\sqrt{32}} = (0, 0.7071, 0.7071)$$

$$N_{P_4} = \frac{(V_5V_1) \times (V_1V_4)}{|(V_5V_1) \times (V_1V_4)|} = (-2, 0, 2) \frac{1}{\sqrt{8}} = (-0.5774, 0, 0.5774)$$

- Υπολογίζουμε τα κάθετα διανύσματα στα 4 σημεία που περικλείουν το σημείο Γ

$$\mathbf{N}_{V_1} = \mathbf{N}_{P_0} + \mathbf{N}_{P_1} + \mathbf{N}_{P_4} = (0, 0, 1) + (0, -0.7071, 0.7071) + (-0.7071, 0, 0.7071) = (-0.7071 \quad -0.7071 \quad 2.4142) \rightarrow (-0.2706 \quad -0.2706 \quad 0.9239)$$

$$\mathbf{N}_{V_4} = \mathbf{N}_{P_0} + \mathbf{N}_{P_3} + \mathbf{N}_{P_4} = (0, 0, 1) + (0, 0.7071, 0.7071) + (-0.7071, 0, 0.7071) = (-0.7071 \quad 0.7071 \quad 2.4142) \rightarrow (-0.2706 \quad 0.2706 \quad 0.9239)$$

$$\mathbf{N}_{V_8} = \mathbf{N}_{P_3} + \mathbf{N}_{P_4} = (0, 0.7071, 0.7071) + (-0.7071, 0, 0.7071) = (-0.7071 \quad 0.7071 \quad 1.4142) \rightarrow (-0.4082 \quad 0.4082 \quad 0.8165)$$

$$\mathbf{N}_{V_5} = \mathbf{N}_{P_1} + \mathbf{N}_{P_4} = (0, -0.7071, 0.7071) + (-0.7071, 0, 0.7071) = (-0.7071 \quad -0.7071 \quad 1.4142) \rightarrow (-0.4082 \quad -0.4082 \quad 0.8165)$$

- Υπολογίζουμε τα διανύσματα ανάκλασης στα 4 σημεία που περικλείουν το σημείο Γ

$$\mathbf{R}_{V1} = 2\mathbf{N}_{V1}(\mathbf{LN}_{V1}) - \mathbf{L} = (0 \quad 0 \quad 1)$$

$$\mathbf{R}_{V4} = 2\mathbf{N}_{V4}(\mathbf{LN}_{V4}) - \mathbf{L} = (-0.5000 \quad 0.5000 \quad 0.7071)$$

$$\mathbf{R}_{V5} = 2\mathbf{N}_{V5}(\mathbf{LN}_{V5}) - \mathbf{L} = (-0.6667 \quad -0.6667 \quad 0.3333)$$

$$\mathbf{R}_{V8} = 2\mathbf{N}_{V8}(\mathbf{LN}_{V8}) - \mathbf{L} = (-0.6667 \quad 0.6667 \quad 0.3333)$$

- Υπολογίζουμε την ένταση στα 4 σημεία που περικλείουν το σημείο Γ

$$I_{V1} = 3.4013$$

$$I_{V4} = 3.1512$$

$$I_{V5} = 3.4772$$

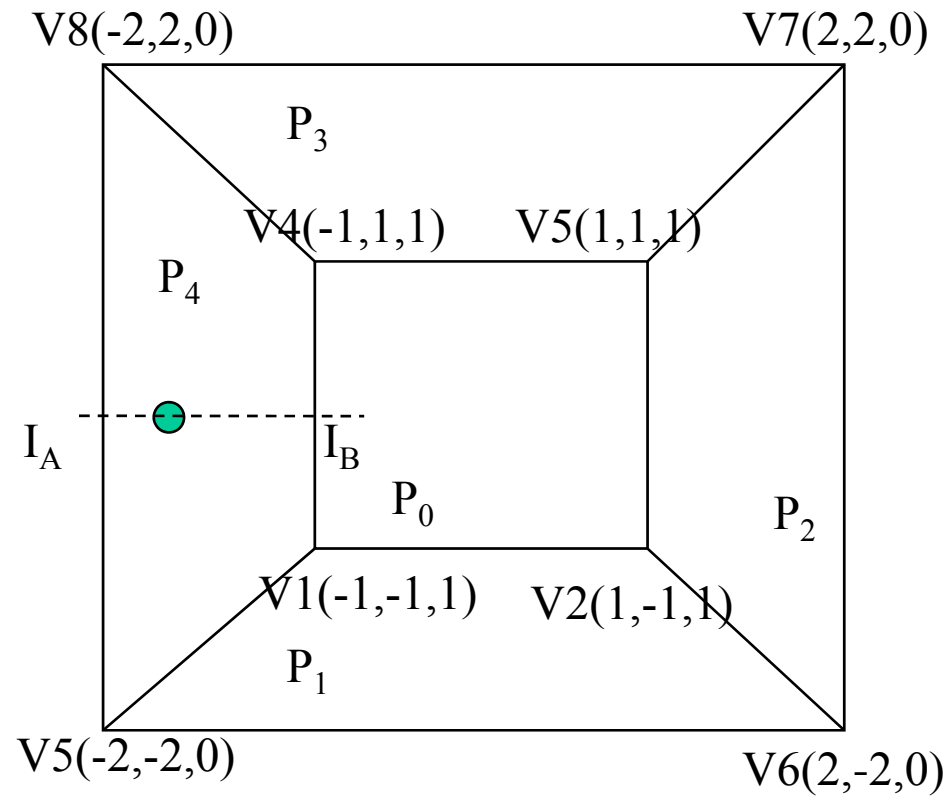
$$I_{V8} = 2.8585$$

- κηξιοξ

$$I_A = \frac{y_8 - y_C}{y_8 - y_5} I_5 + \frac{y_C - y_5}{y_8 - y_5} I_8$$

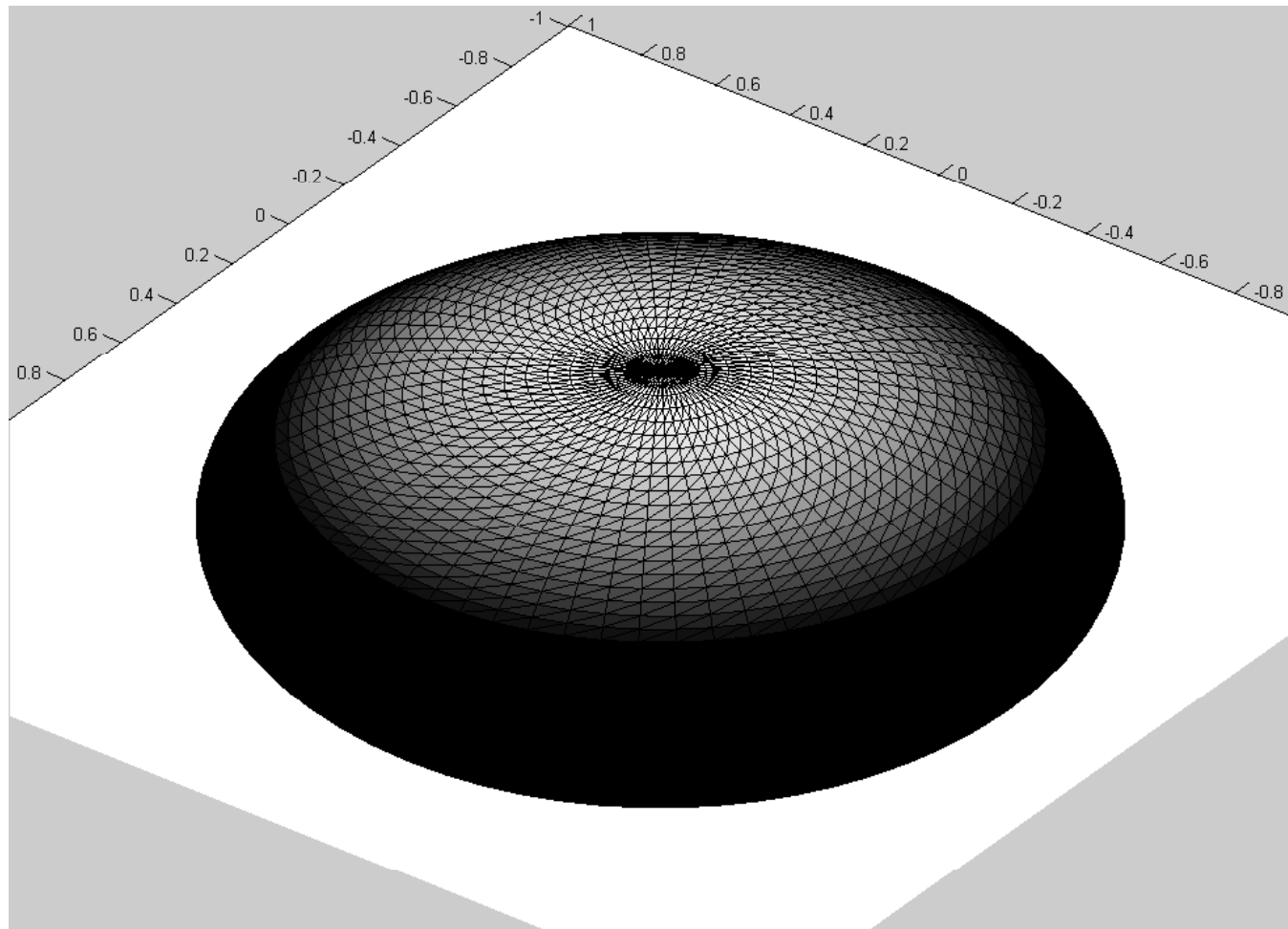
$$I_B = \frac{y_8 - y_C}{y_8 - y_5} I_1 + \frac{y_C - y_5}{y_8 - y_5} I_3$$

$$I_C = \frac{x_B - x_C}{x_B - x_A} I_B + \frac{x_C - x_A}{x_B - x_A} I_A$$

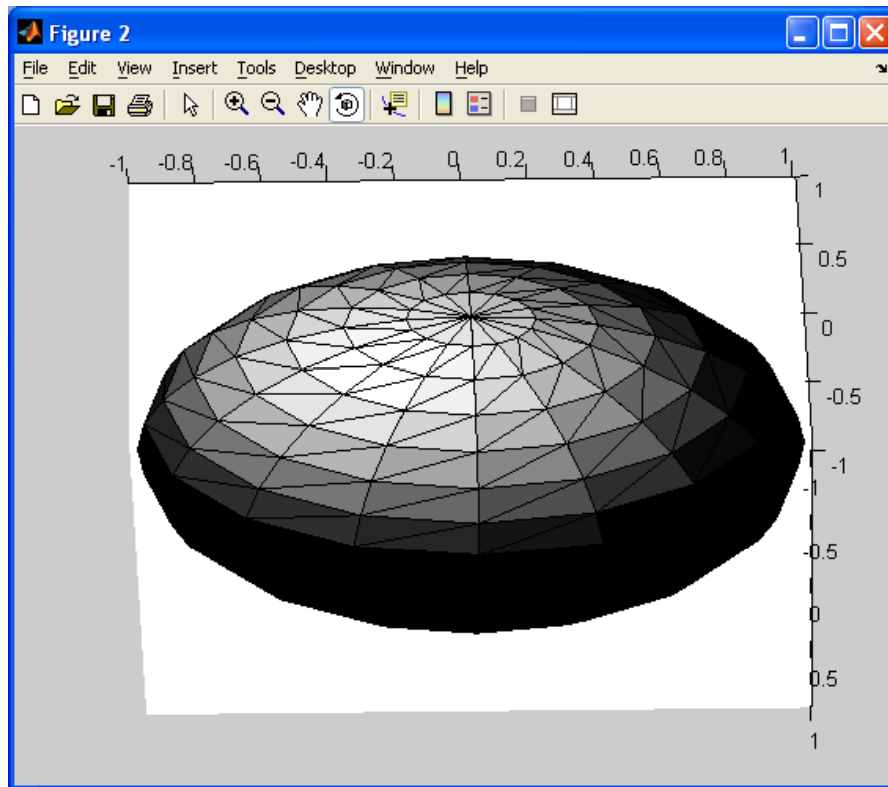


Κ. Δελήμπασης

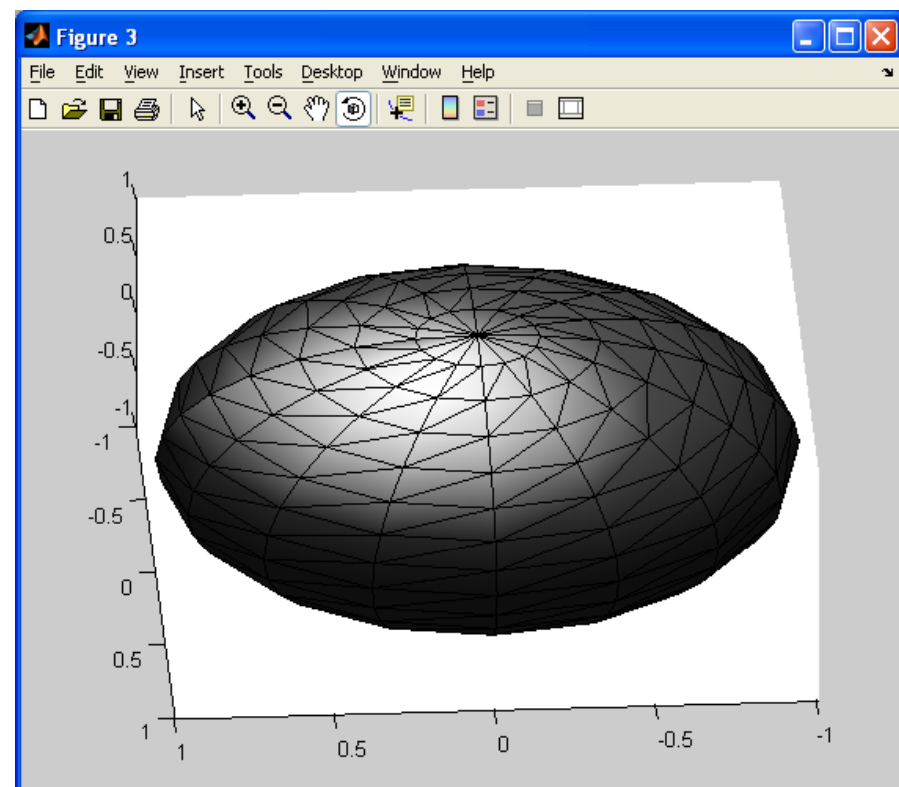
Παράδειγμα: Μοντέλο φωτισμού Phong και
παραμετροποιημένη σφαιρική επιφάνεια. Τρίγωνα
με flat shading



Κ. Δελήμασης

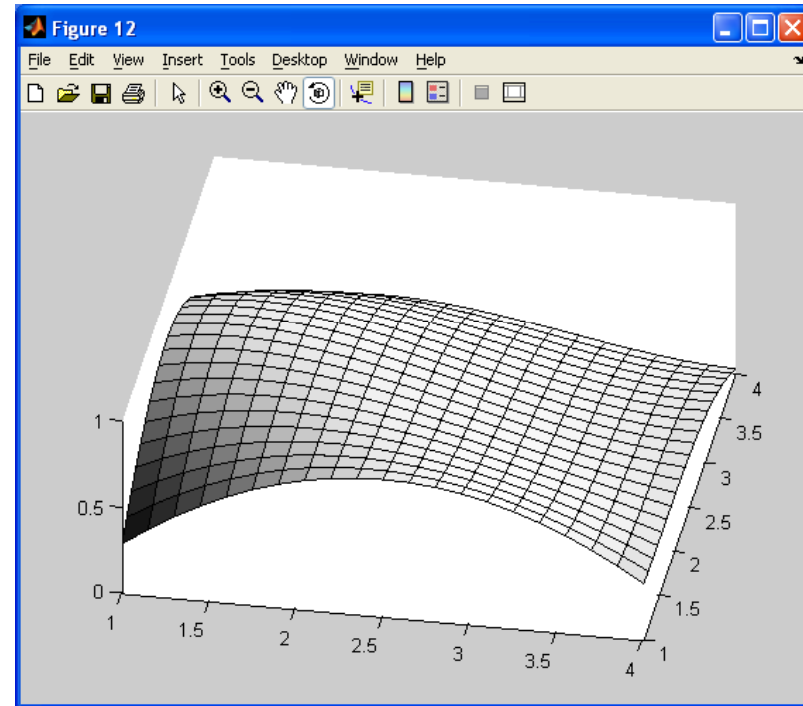
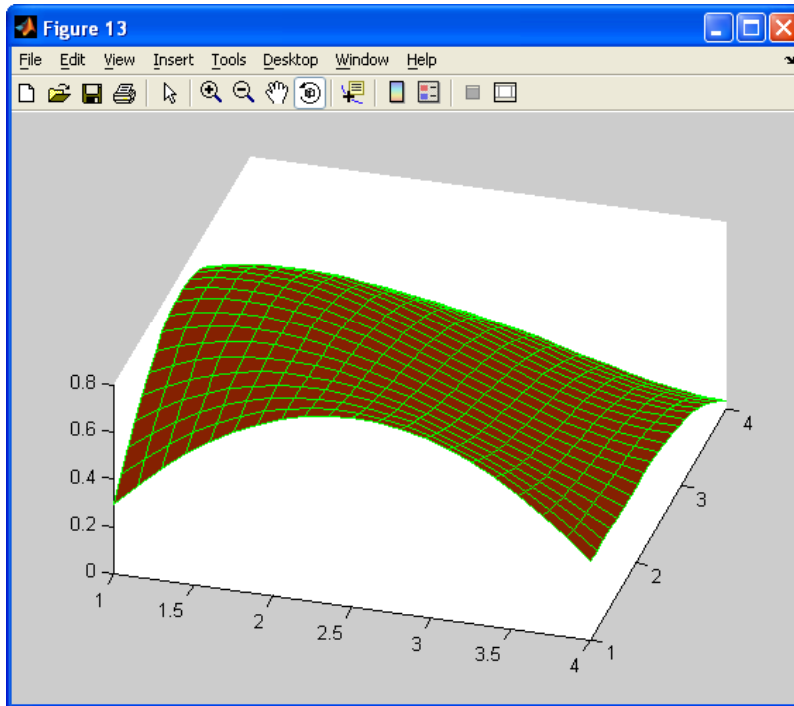


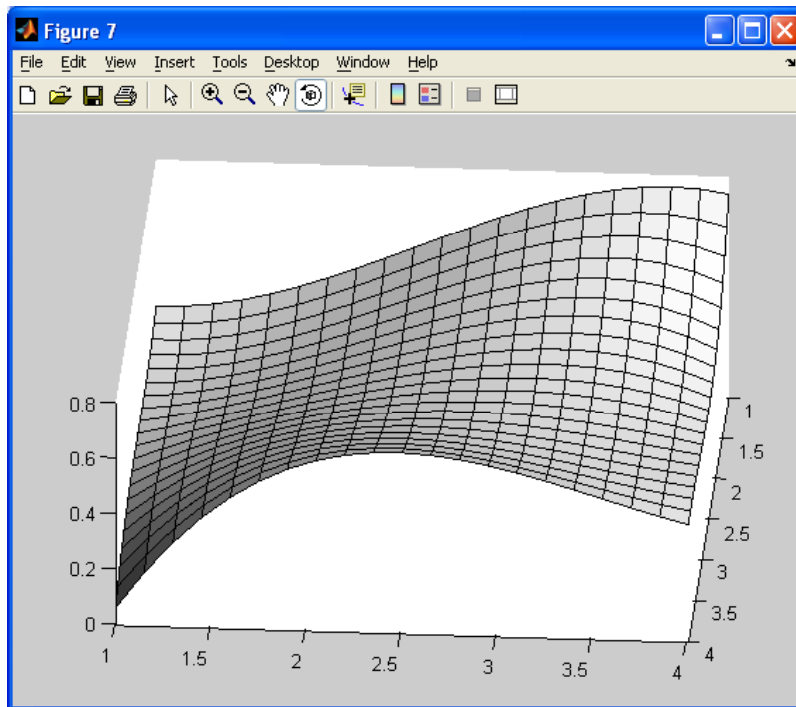
Τρίγωνα με σταθερό (flat) shading



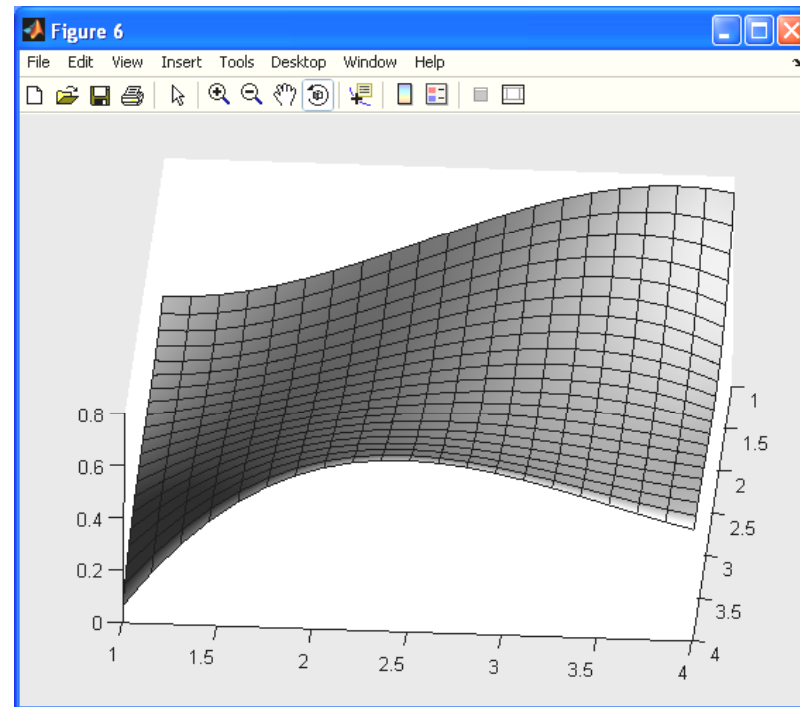
Τρίγωνα με interpolated shading (Gouraud)

Παράδειγμα: Μοντέλο φωτισμού Phong και επιφάνεια Bezier - Τετράγωνα με flat shading





Τετράγωνα με flat shading

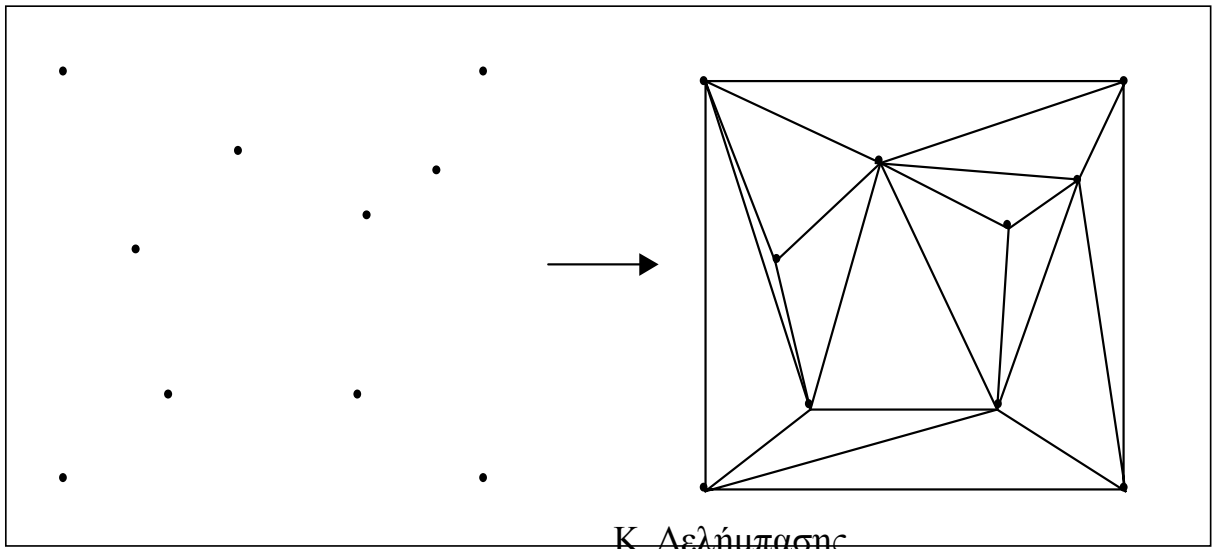
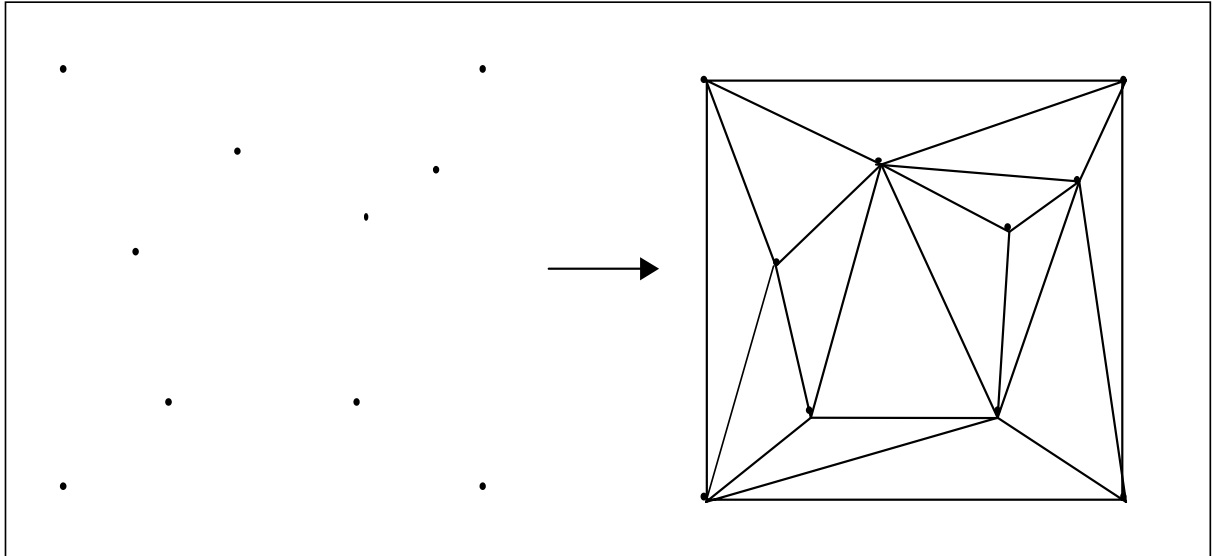


Τετράγωνα με interpolated shading

Κ. Δελημπασα

Τριγωνισμός σημείων στο επίπεδο

- Με δεδομένο ένα σύνολο σημείων ο τριγωνισμός παράγει ένα σύνολο πλευρών οι οποίες δημιουργούν τρίγωνα με συγκεκριμένες ιδιότητες
- Τριγωνισμός Delauney:
 - Τα τρίγωνα δεν τέμνονται
 - Κάθε σημείο δεν εμπεριέχεται στον περιγεγραμμένο κύκλο κανενός άλλου τριγώνου
 - Η ελάχιστη γωνία κάθε τριγώνου μεγιστοποιείται



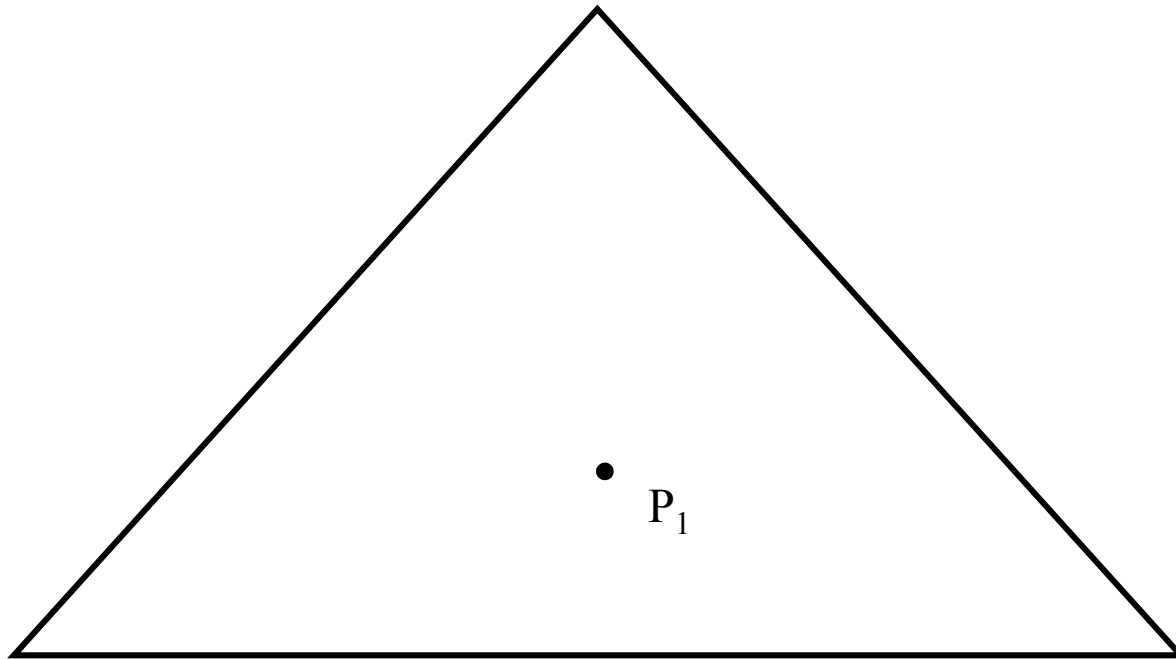
Κ. Δελημπασης

Αυξητικός αλγόριθμος του Watson (1981)

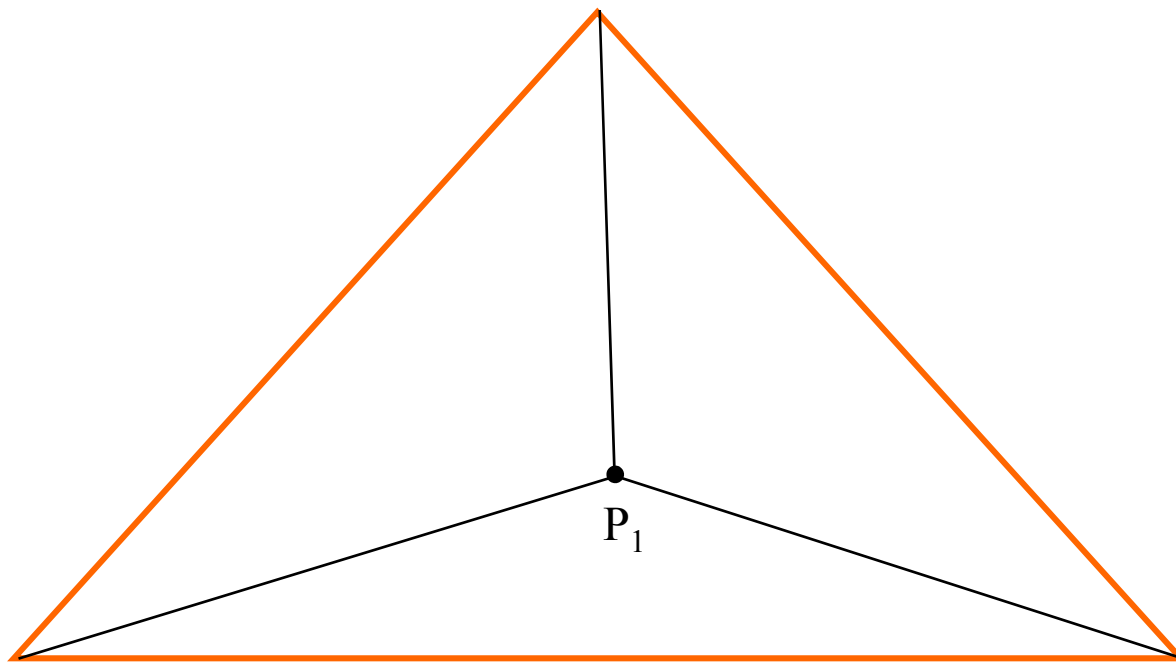
- Δημιουργούμε ένα τρίγωνο που περικλείει τα σημεία προς τριγωνοποίηση
- Συνδέουμε το P_1 με τις κορυφές του εξωτερικού τριγώνου
- Για όλα τα επόμενα σημεία P_i :
 - Εντοπίζονται τα τρίγωνα T_i των οποίων ο περιγεγραμμένος κύκλος περιέχει το σημείο P_i
 - Εντοπίζονται οι πλευρές P_i των τριγώνων T_i που εμφανίζονται μόνο μία φορά
 - Διαγράφονται τα τρίγωνα T_i
 - Δημιουργούνται τα τρίγωνα με κορυφή το P_i και βάση τα P_i
- Διαγράφουμε όλα τα τρίγωνα στα οποία μετέχουν τα σημεία (κορυφές) του αρχικού περικλείοντος τριγώνου

Παράδειγμα του αυξητικού αλγόριθμου του Watson

- Εστω τα 3 σημεία με συντεταγμένες τέτοιες ώστε να περοκλείου όλα τα σημεία προς τριγωνοποίηση. Προβάλλουμε το 1 από τα σημεία προς τριγωνοποίηση.

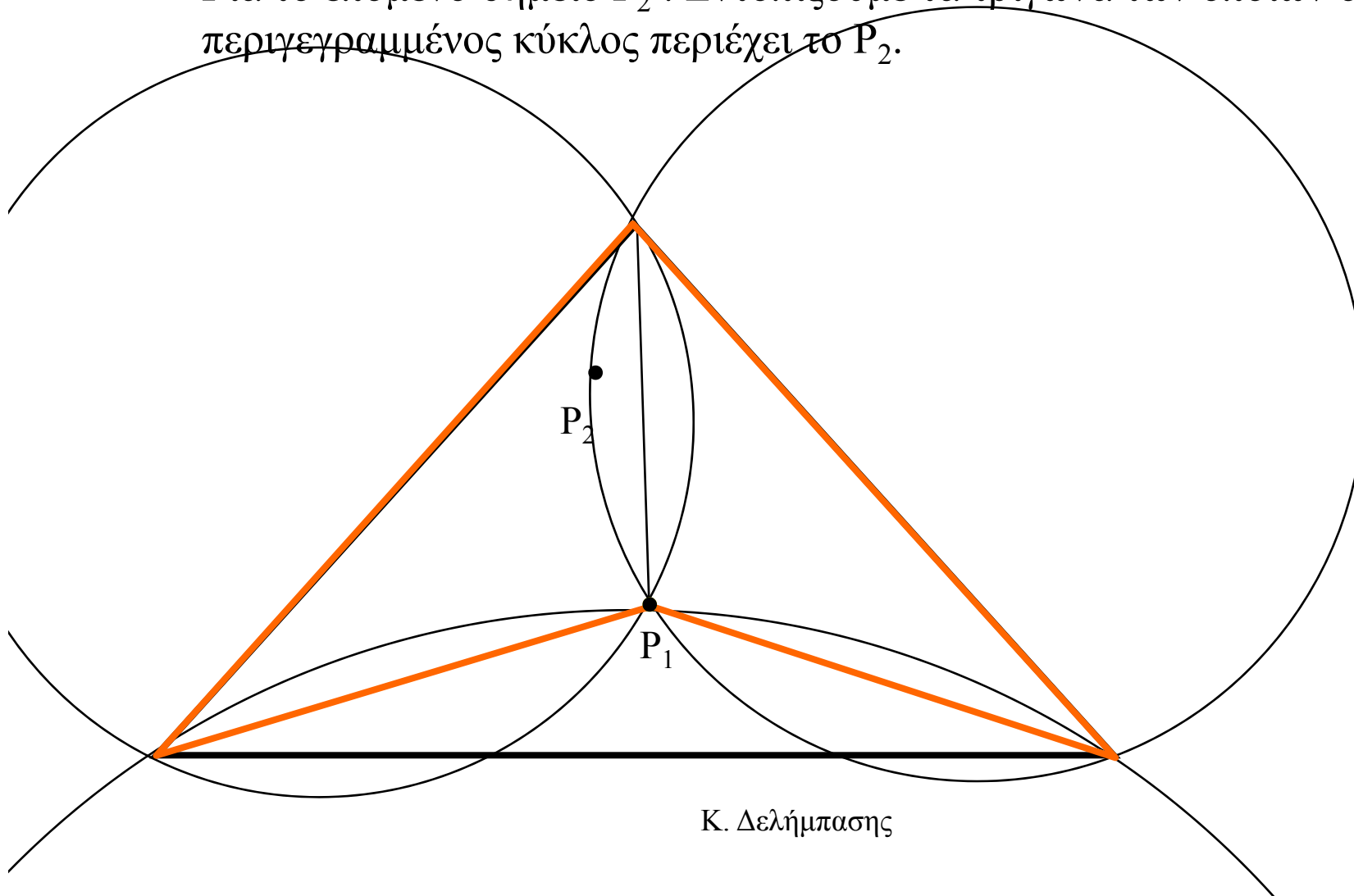


- Συνδέουμε το P_1 με τα 3 σημεία δημιουργώντας 3 τρίγωνα.

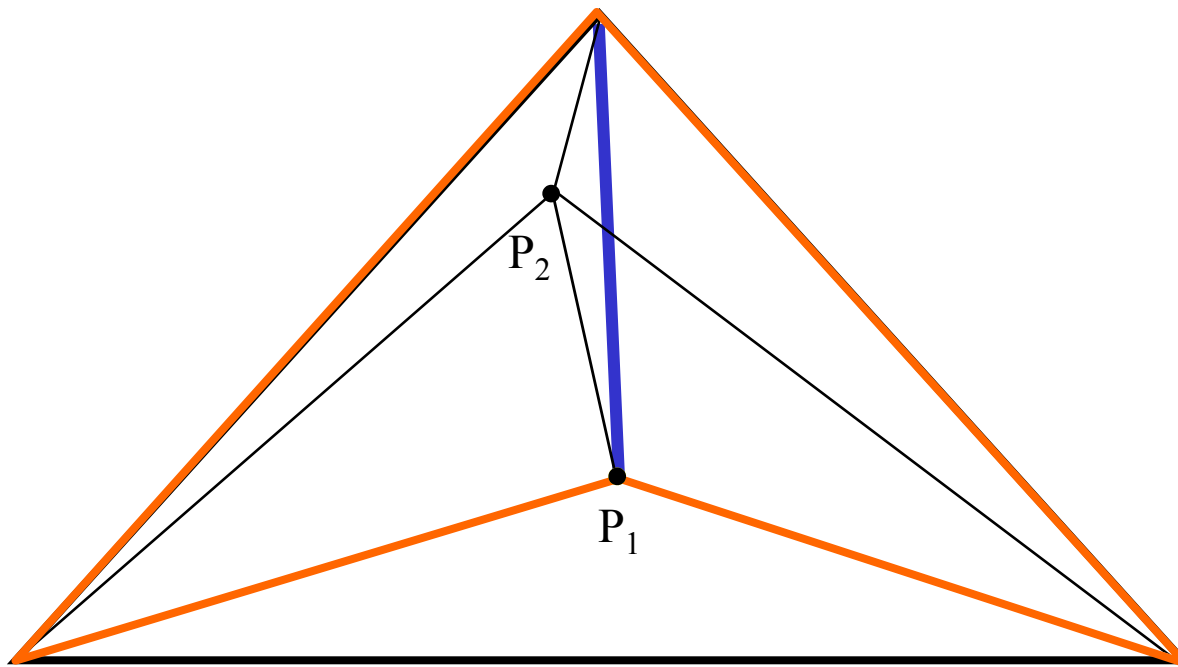


Κ. Δελήμπασης

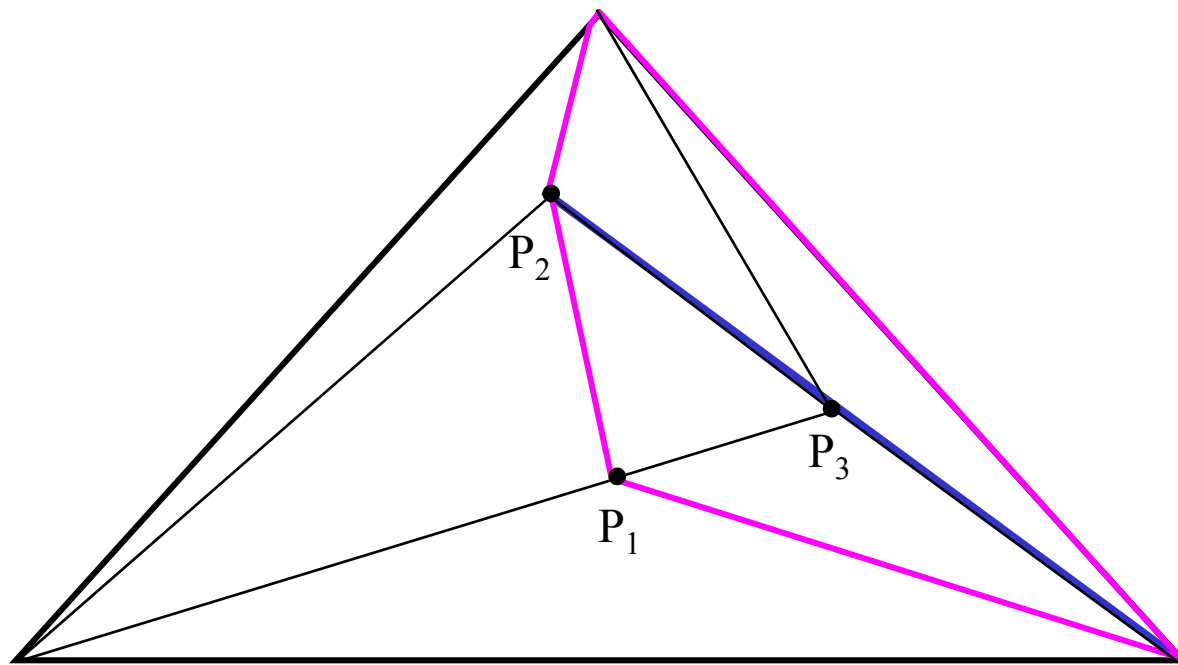
- Για το επόμενο σημείο P_2 : Εντοπίζουμε τα τρίγωνα των οποίων ο περιγεγραμμένος κύκλος περιέχει το P_2 .



- Διαγράφουμε τα τρίγωνα αυτά και δημιουργούμε νέα συνδέοντας το P_2 με όλα τα σημεία των διαγραμμένων τριγώνων.

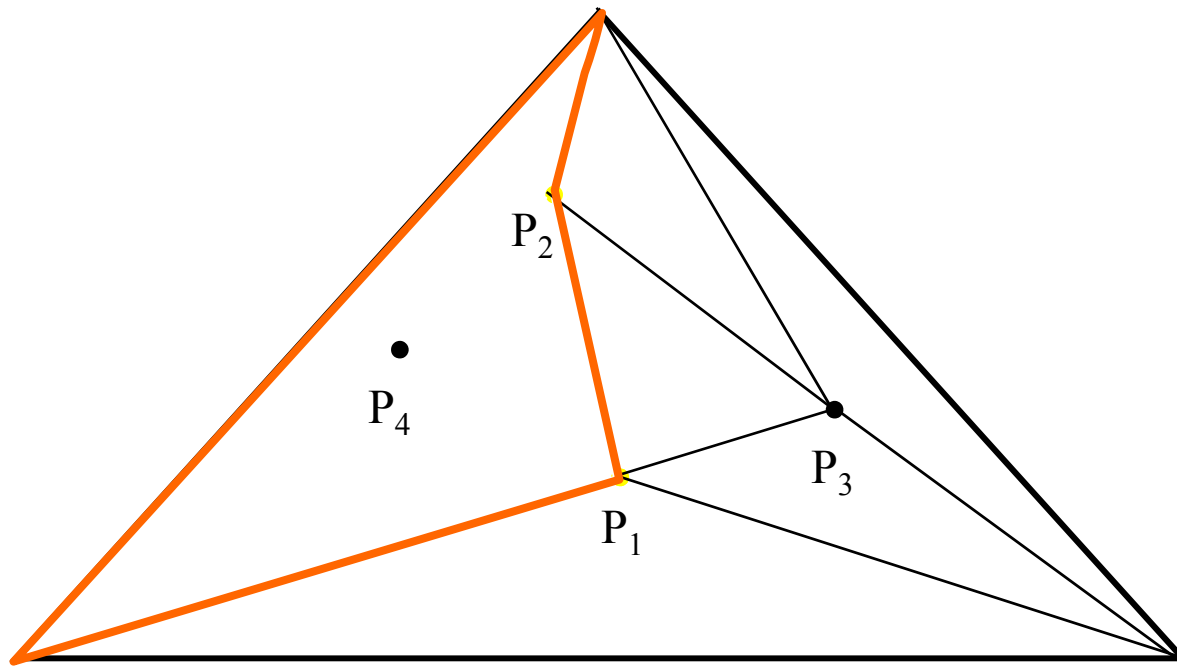


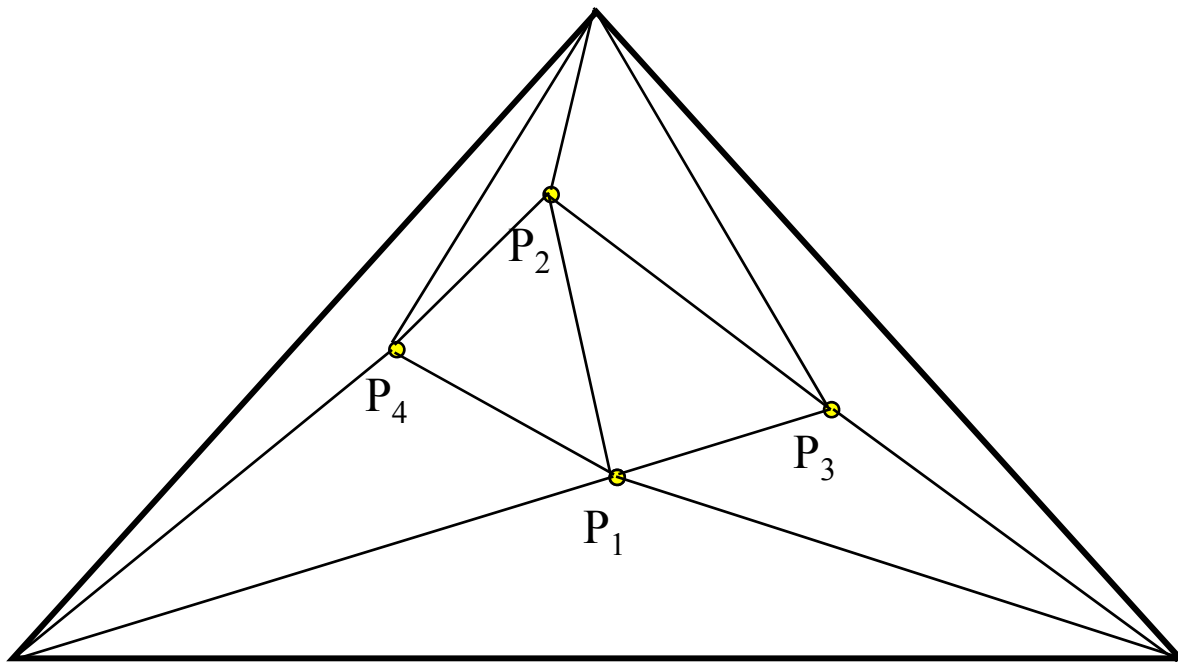
- Για το επόμενο σημείο P_3 : Εντοπίζουμε τα τρίγωνα των οποίων ο περιγεγραμμένος κύκλος περιέχει το P_3 .



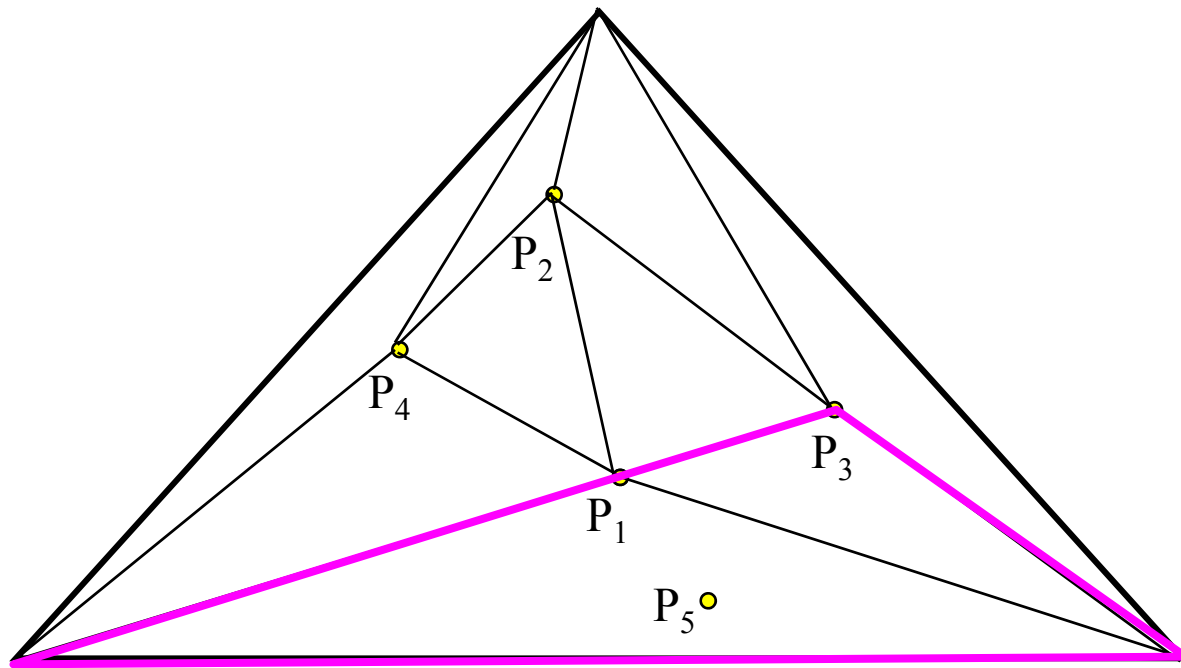
Κ. Δελήμπασης

- Για το επόμενο σημείο P_3 : Εντοπίζουμε τα τρίγωνα των οποίων ο περιγεγραμμένος κύκλος περιέχει το P_3 .

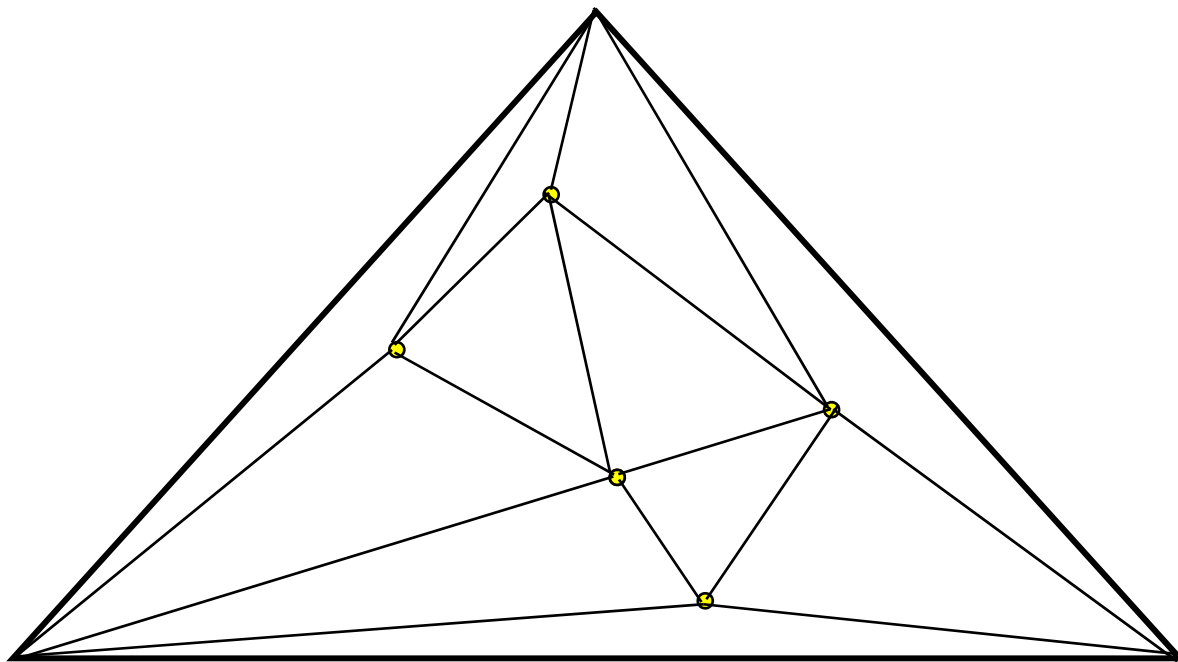




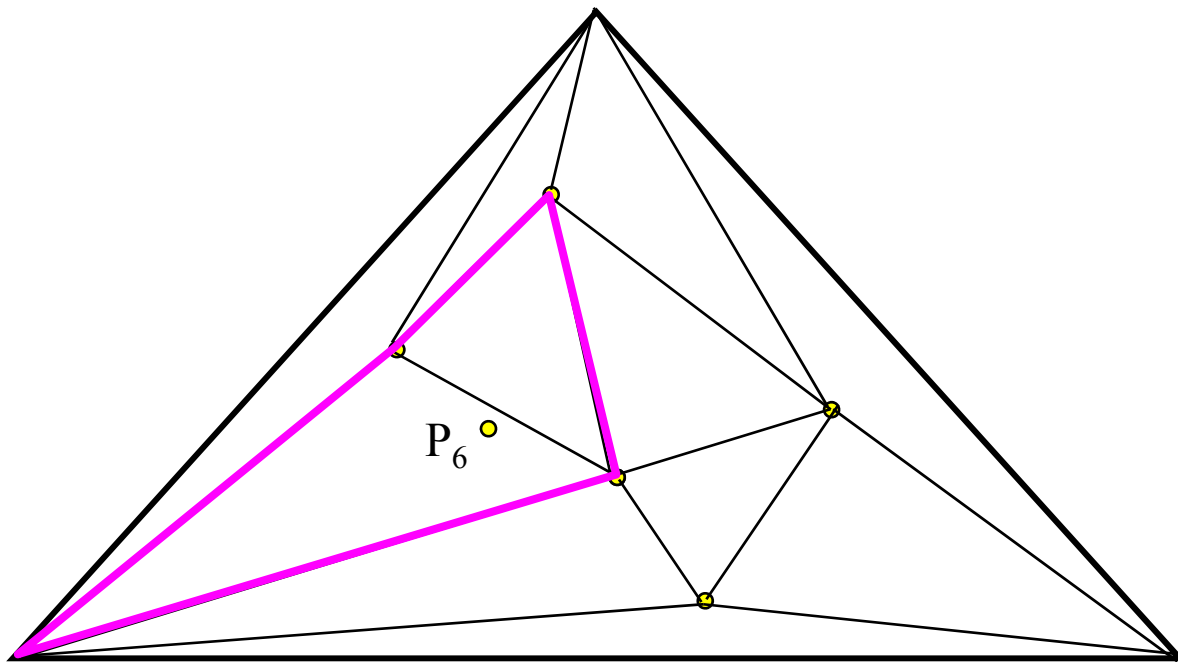
Κ. Δελήμπασης



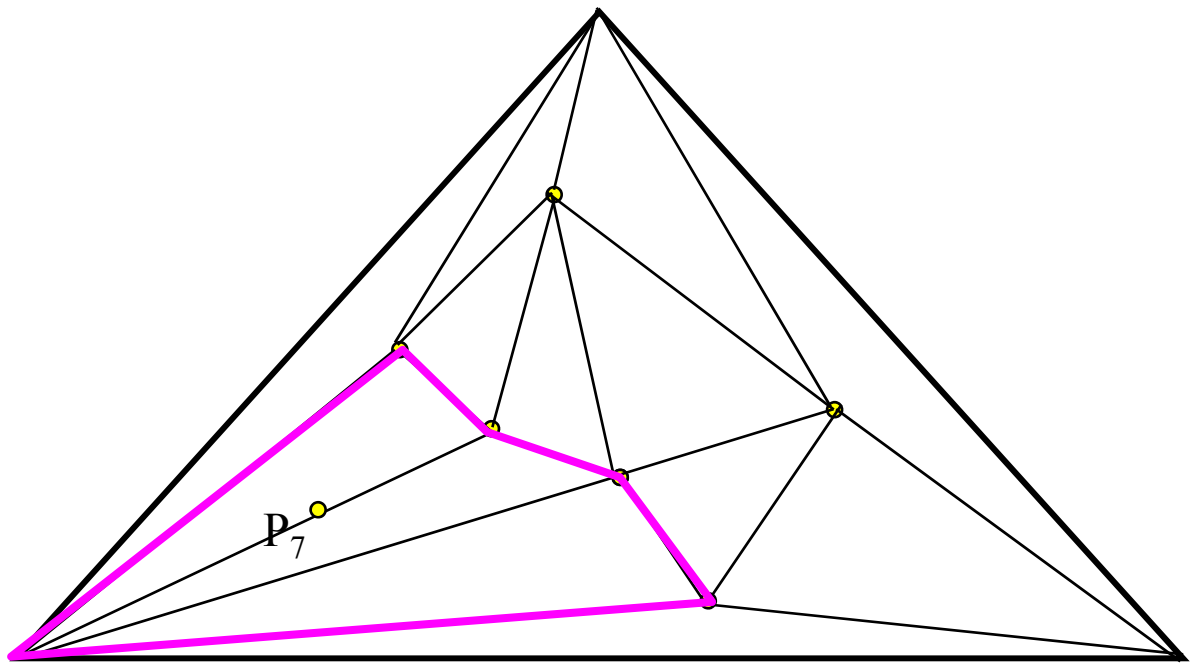
Κ. Δελήμπασης



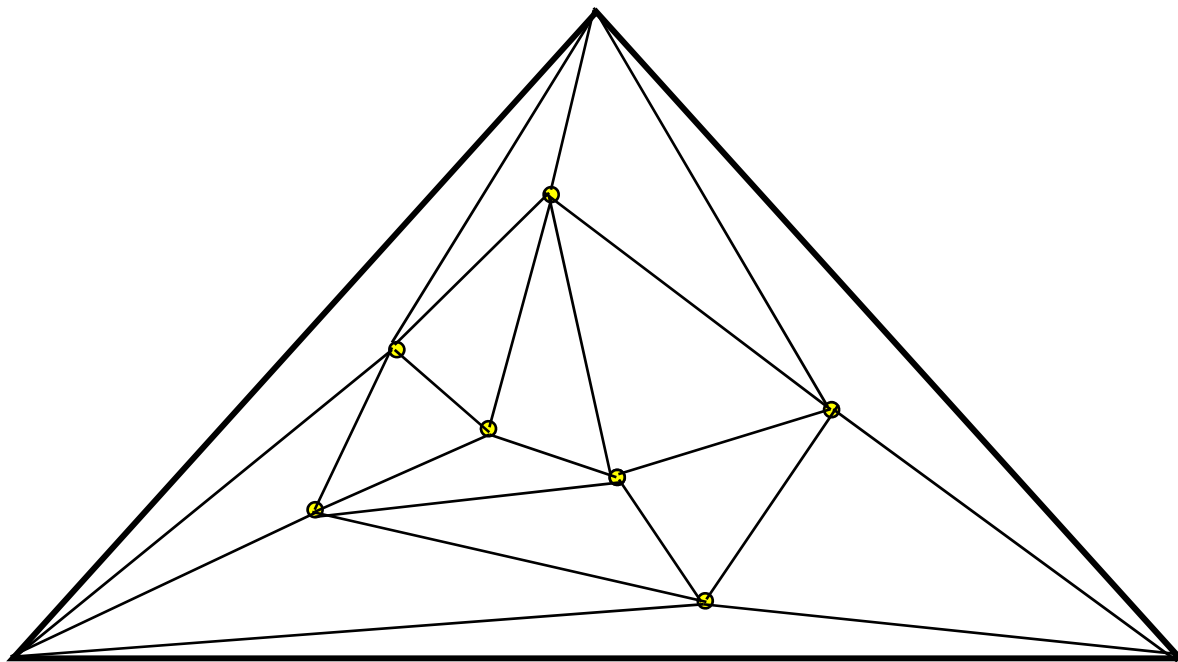
Κ. Δελήμπασης



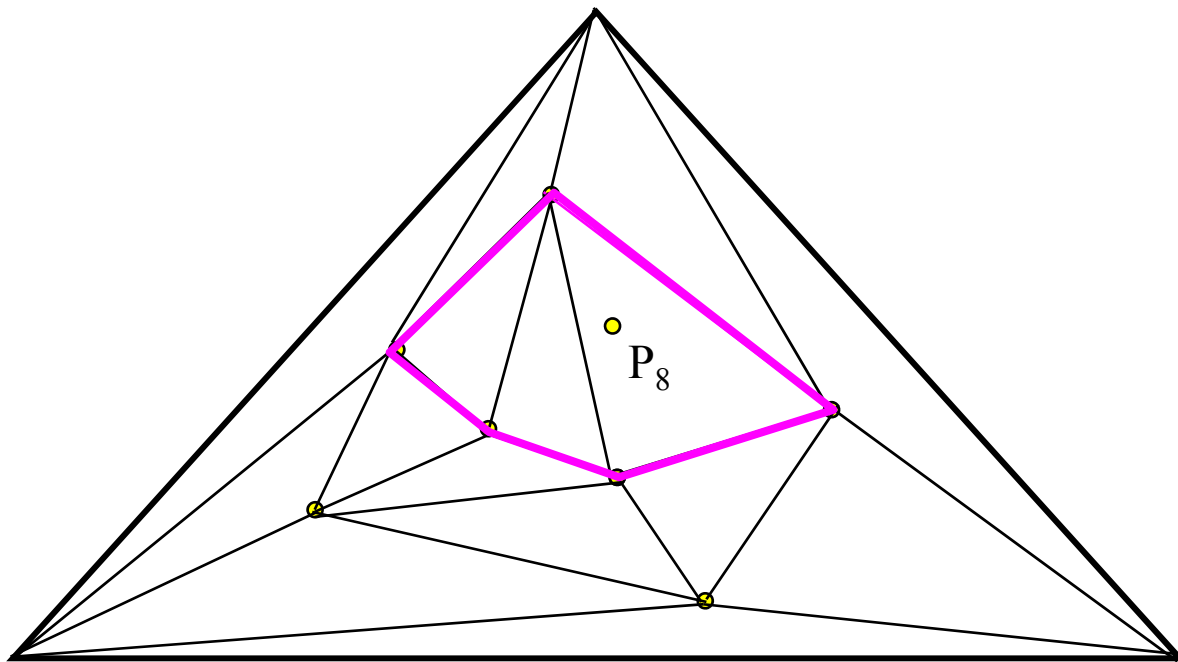
Κ. Δελήμπασης



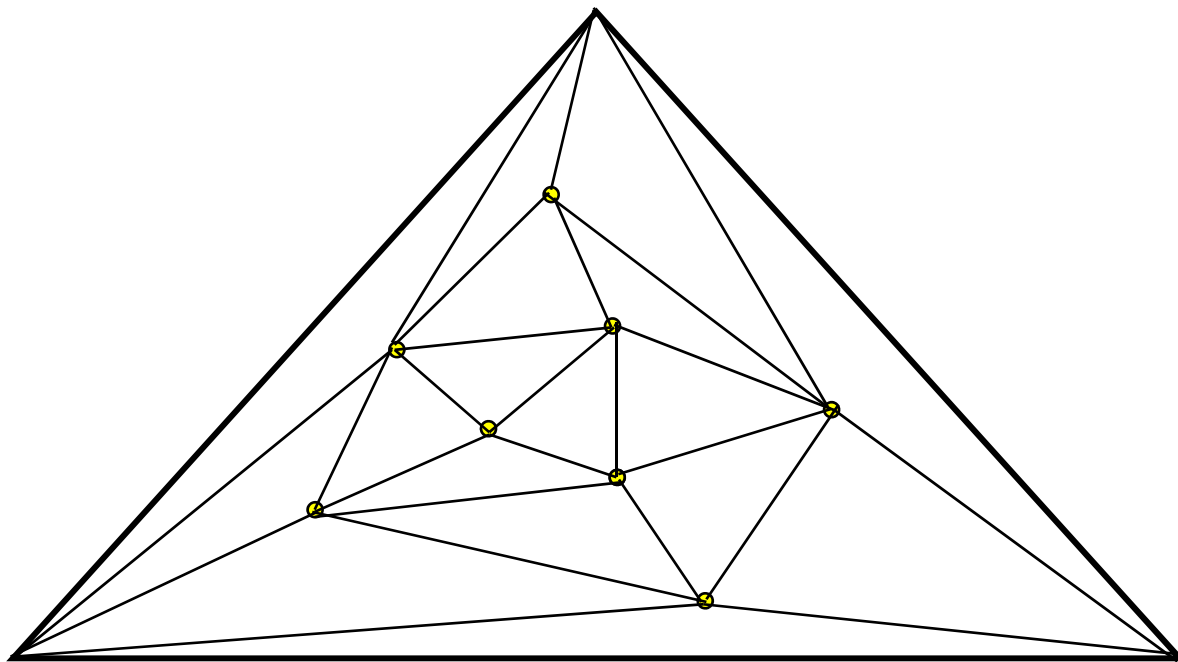
Κ. Δελήμασης



Κ. Δελήμασης

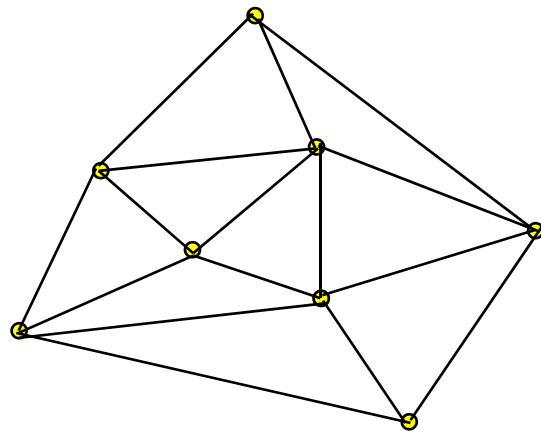


Κ. Δελήμασης



Κ. Δελήμασης

- Διαγράφουμε τα βοηθητικά σημεία του περικλείοντος τριγώνου και όλες τις συνδέσεις τους



Κ. Δελήμασης

Τριγωνισμός σημείων στο χώρο: ο Αλγόριθμος του βηματίζοντος κύβου (Marching Cubes - MC)

- Ο αλγόριθμος MC τριγωνοποιεί επιφάνειες οι οποίες ορίζονται σε τρισδιάστατες (3D) εικόνες.
- Οι επιφάνειες ορίζονται με δύο τρόπους:
 - Είτε η αρχική εικόνα τμηματοποιείται και παράγεται μία δυαδική εικόνα η οποία έχει σε κάθε voxel τιμή 0 αν αυτό ανήκει στο υπόβαθρο και τιμή 1 (ή 255) αν αυτό ανήκει στο αντικείμενο ενδιαφέροντος
 - Είτε η επιφάνεια ορίζεται στην αρχική εικόνα (με τις «gray level» τιμές των voxel) μέσω της τιμής της T_0 : θεωρούμε ότι η επιφάνεια περνά μόνο από voxel με τιμή ίση με T_0 . Η επιφάνεια αυτή καλείται ισοεπιφάνεια
- Ο ψευδοκώδικας του αλγόριθμου ακολουθεί

```

FOR each image voxel
  a cube of length 1 is placed on eight adjacent voxels of
  the image
  FOR each of the cube's edge{
    IF (the one of the node voxels has value greater than
        or equal to t AND the other voxel has value
        less than t)

    THEN

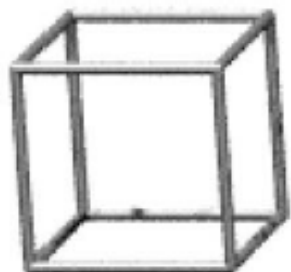
      {calculate the position of a point on the
      cube's
      edge that belongs to the isosurface, using
      linear interpolation}
  }
FOR each of the predefined cube configurations{
  FOR each of the eight possible rotations{
    FOR the configuration's complement{
      {compare the produced cube configuration
      of the above calculated isopoints to the
      set of predefined cube configurations and
      produce the corresponding triangles}
    }
  }
}

```

Βασικές αρχές του αλγόριθμου MC

- Ένας κύβος με ακμή ίση με 1 voxel ολισθαίνει («βηματίζει») στην 3D εικόνα, έτσι ώστε οι κορυφές του να ακουμπούν σε 8 γειτονικά voxel της εικόνας.
- Κάθε ένα από τα 8 voxel αναπαρίσταται σαν σφαίρα στα σχήματα που ακολουθούν αν το voxel έχει τιμή $>T_0$.
- Ανάλογα με τη διάταξη του κύβου (συνδυασμό των 8 voxel των κορυφών του κύβου με τιμή $>T_0$) καθορίζονται τα στοιχειώδη τρίγωνα / πολύγωνα
- Ο MC βασίζεται σε 15 διαφορετικές περιπτώσεις, που προκύπτουν μετά από την αρχή της συμπληρωματικότητας και περιστροφές της διάταξης στο χώρο

Οι 15 διαφορετικές διατάξεις του MC και τα
αντίστοιχα στοιχειώδη τρίγωνα



Case 0



Case 1



Case 2



Case 3



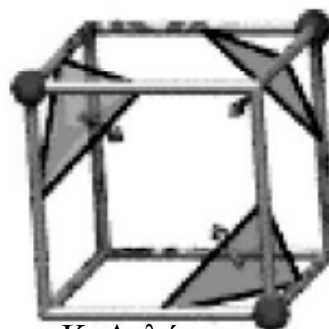
Case 4



Case 5



Case 6



Κ. Δελήμασης
Case 7



Case 8



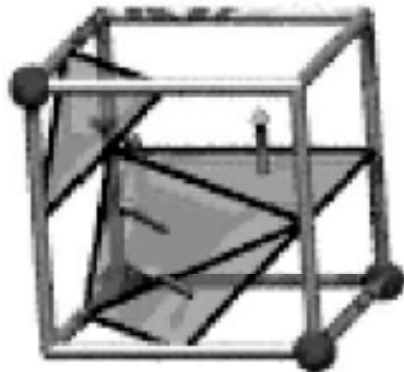
Case 9



Case 10



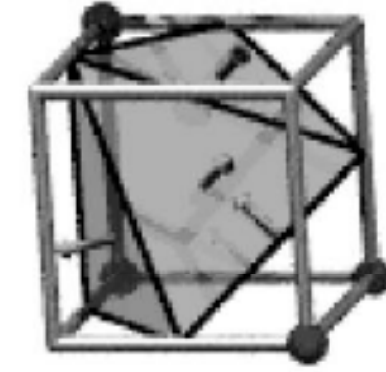
Case 11



Case 12

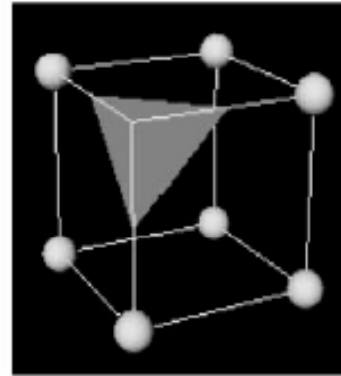
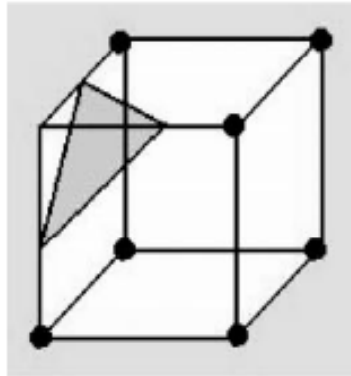


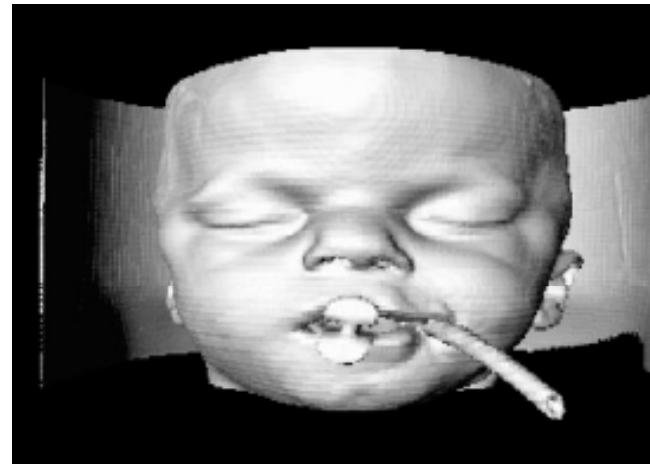
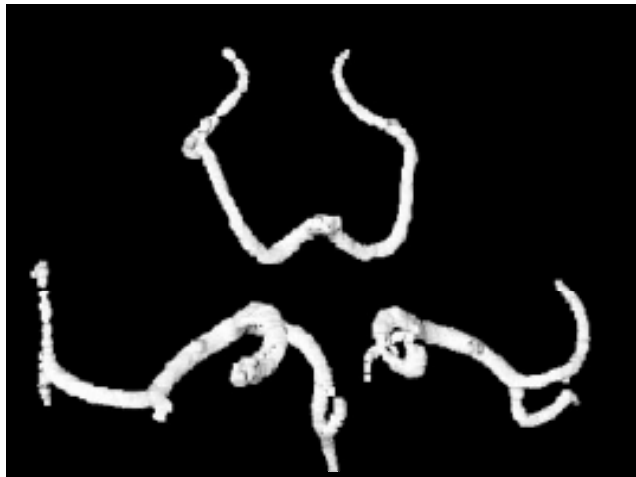
Case 13



Case 14

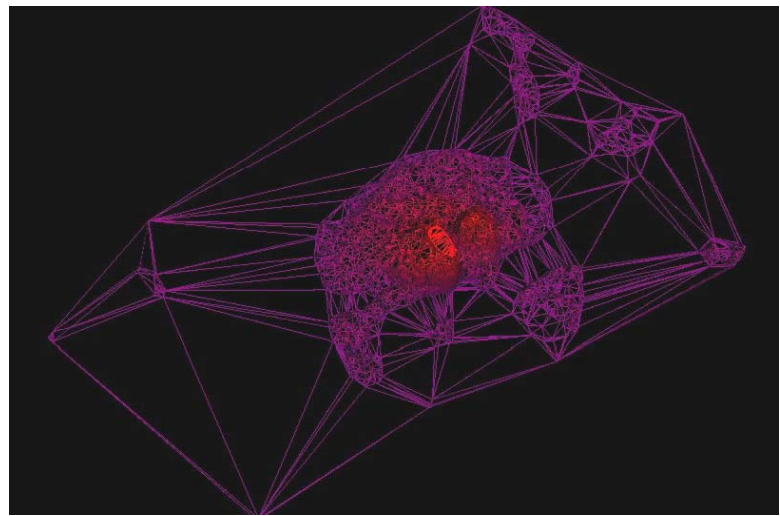
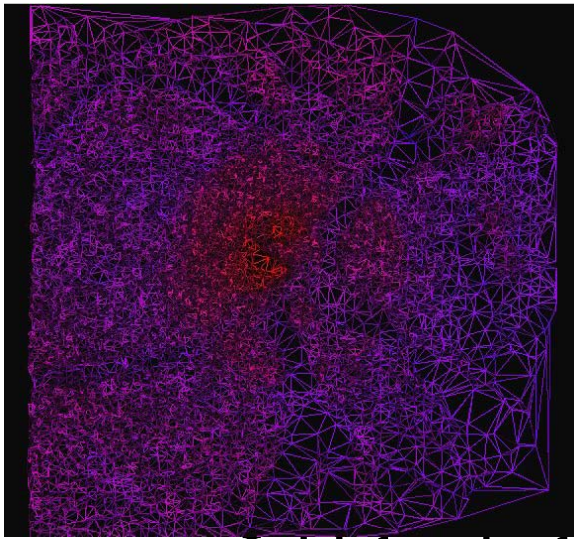
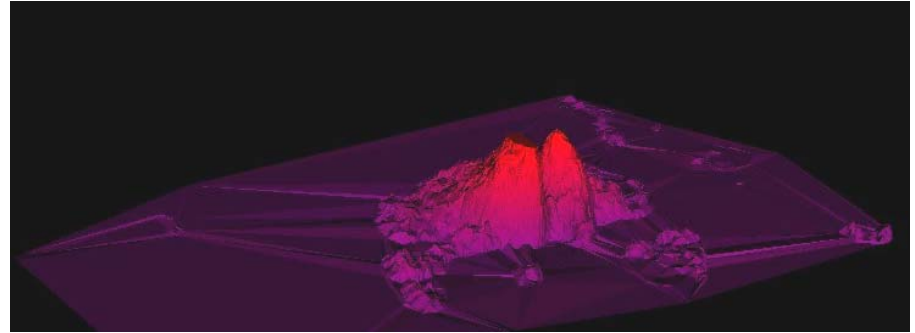
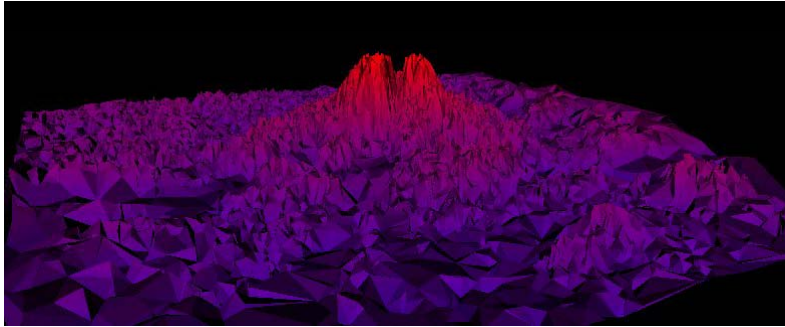
- Η αρχή της συμπληρωματικότητας στον ΜC: συμπληρωματικές διατάξεις του κύβου παράγουν την ίδια τριγωνοποίηση





Αποτελέσματα εφαρμογής του MC με δεδομένα CT και MRI.
Οι τιμές κατωφλίωσης της ισοεπιφάνειας επιλέχθησαν ώστε να
απεικονίζεται αγγεία και δέρμα.

Κ. Δελήμπασης



Initial retinal surface with low decimation effect