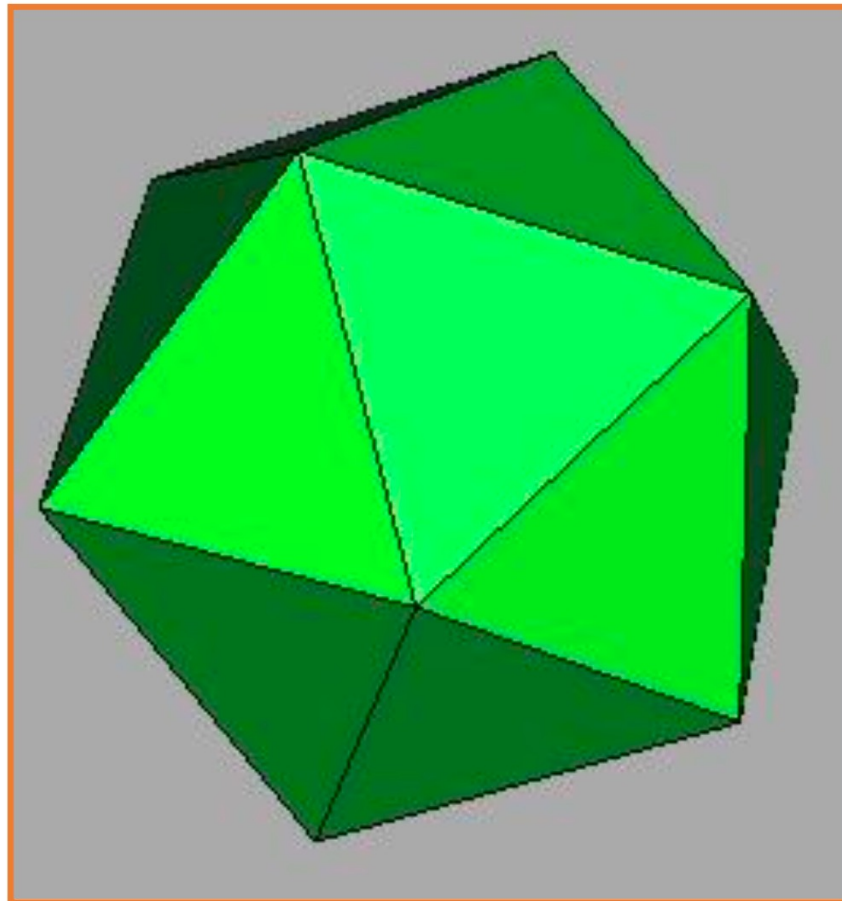


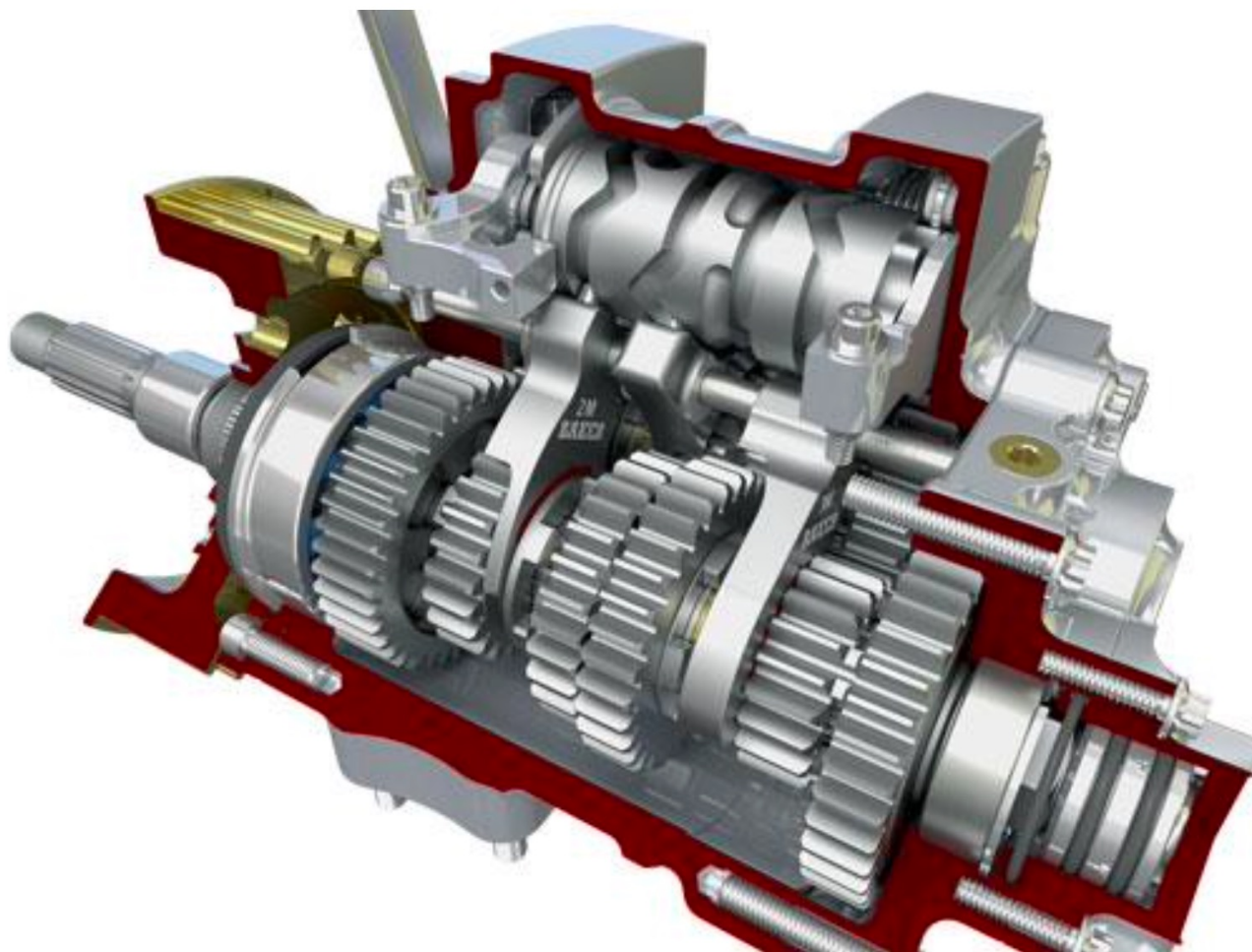
Εισαγωγή

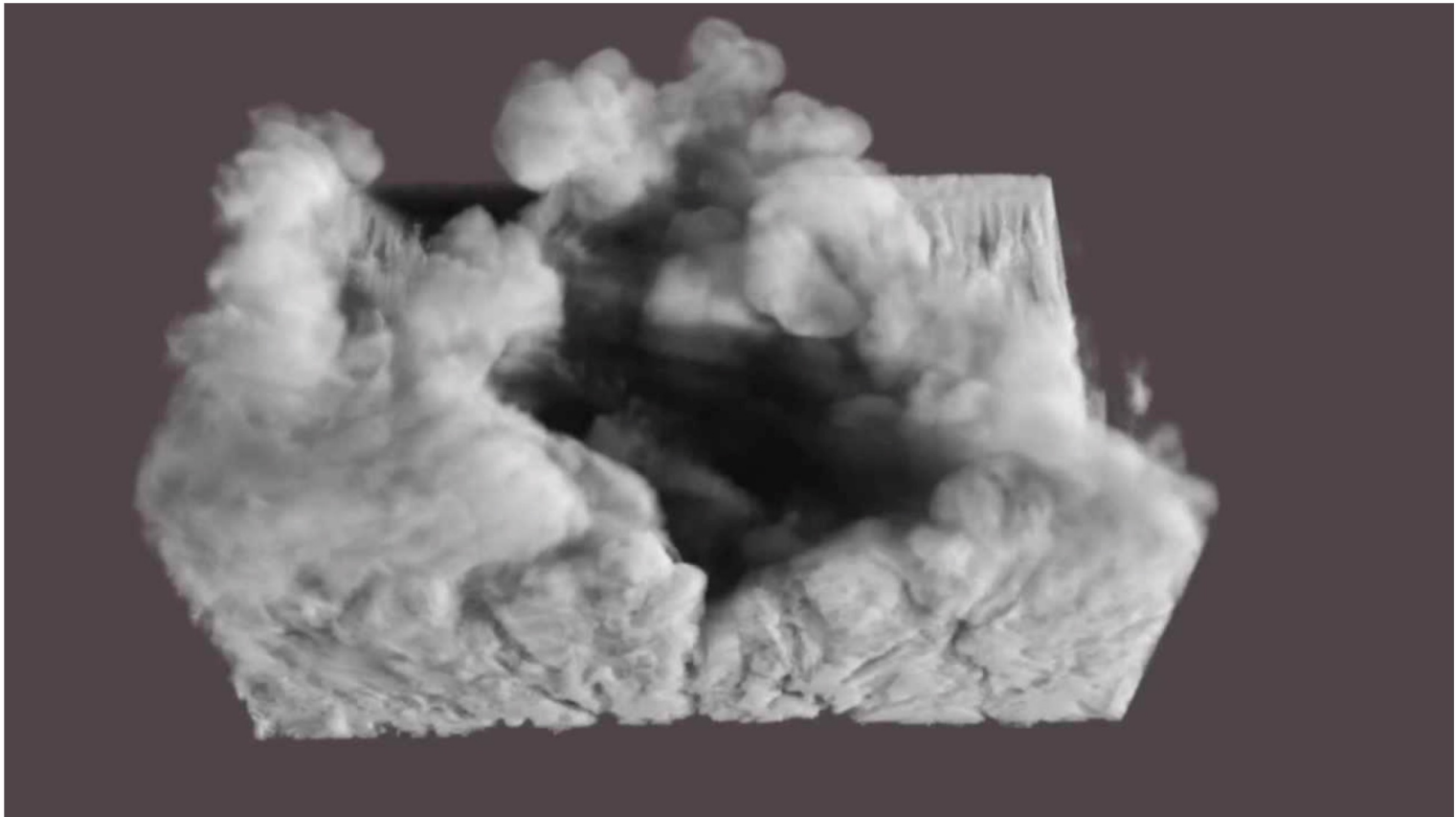
- Οι 3Δ εικόνες στα Γραφικά αποτελούνται από διάφορα σχήματα & δομές:
 - Γεωμετρικά σχήματα (π.χ. σφαίρες)
 - Μαθηματικές επιφάνειες (π.χ. τμήματα επιφάνειας NURBS)
 - Τυχαίες επιφάνειες, όχι μαθηματικά ορισμένες (π.χ. επιφάνεια ψηφιοποιημένου αντικειμένου)
 - Αντικείμενα όγκου, όπου η εσωτερική δομή του αντικειμένου είναι εξίσου σημαντική με την επιφάνεια του (π.χ. ανθρώπινο όργανο)
 - Ακανόνιστα αντικείμενα (π.χ. καπνός)
- *Μοντέλα:* Προσεγγιστικές αναπαραστάσεις αντικειμένων, κατασκευασμένα ώστε να διατηρούν πολλές από τις ιδιότητες του αντικειμένου
- Τα μοντέλα επιδέχονται την επεξεργασία που απαιτούν οι αλγόριθμοι γραφικών

Εισαγωγή (2)

- *Πολυγωνικά Μοντέλα*: Συνηθέστερη αναπαράσταση επιφανειών
- Η πληροφορία που περιέχεται στα μοντέλα αυξάνεται συνεχώς
- Βασικές εφαρμογές στα Γραφικά συχνά απαιτούν μοντέλα με λιγότερη πληροφορία
- *Απλοποίηση Μοντέλου*: Μειώνει την ποσότητα πληροφορίας που περιέχει ένα μοντέλο, χωρίς να θυσιάζει σημαντικά την ποιότητα της αναπαράστασης







Ανασκόπηση Μοντέλων

- 2 κύριες κατηγορίες μοντέλων:
 - Αναπαράσταση Επιφάνειας (ή αναπαράσταση συνόρου (*b-rep*))
 - Αναπαριστά μόνο την επιφάνεια αντικειμένου
 - Αναπαράσταση Όγκου (ή υποδιαίρεση χώρου)
 - Αναπαριστά όλο τον όγκο που καταλαμβάνει ένα αντικείμενο
- Η αναπαράσταση επιφάνειες χρησιμοποιείται συχνότερα διότι:
 - Πολλά αντικείμενα δεν είναι κλειστά → αναπαράσταση όγκου μη εφαρμόσιμη
 - Τα περισσότερα αντικείμενα είναι αδιαφανή → εξοικονόμηση επεξεργασίας με την αναπαράσταση μόνο της επιφάνειας τους, που καθορίζει την εμφάνιση τους
- Η αναπαράσταση όγκου χρησιμοποιείται:
 - Για ημιδιαφανή αντικείμενα
 - Για αντικείμενα με ενδιαφέρουσα εσωτερική δομή
 - Σαν βοηθητική δομή σε γενικούς αλγόριθμους γραφικών

Ανασκόπηση Μοντέλων (2)

- Κάποια μοντέλα δεν κατηγοριοποιούνται εύκολα σε μια από τις παραπάνω κατηγορίες:
 - *Μοντέλα κατασκευαστικής στερεάς γεωμετρίας (CSG)*: αναπαριστούν ένα αντικείμενο συνδυάζοντας στοιχειώδη γεωμετρικά σχήματα
 - *Άμορφα αντικείμενα & φαινόμενα*: μοντελοποιούνται σαν νέφη σημείων ή σαν συνοθύλευμα στοιχειωδών επιφανειών ή όγκων
- Τα Μοντέλα Επιφάνειας κατηγοριοποιούνται:
 - Σε αυτά που έχουν κάποια μαθηματική περιγραφή όπως:
 - Στοιχειώδη Γεωμετρικά Σχήματα
 - Επιφάνειες NURBS
 - Επιφάνειες Υποδιαίρεσης
 - Γενικές Παραμετρικές Επιφάνειες
 - Και σε αυτά που δεν έχουν κάποια μαθηματική περιγραφή:
 - Αποτελούνται από ένα σύνολο σημείων & ένα σύνολο πολυγώνων που κατασκευάζονται από τα σημεία του συνόλου σαν κορυφές → *πολυγωνικά μοντέλα*

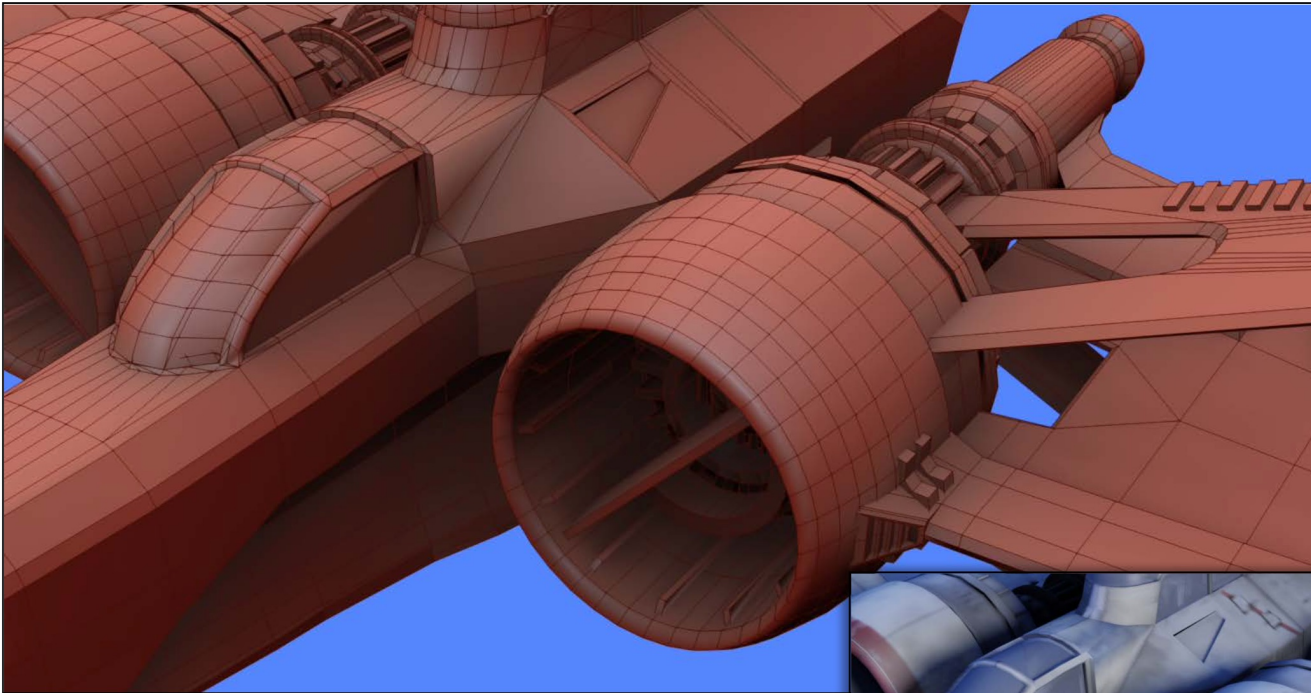
Ανασκόπηση Μοντέλων (3)

- Συγκρίνοντας τις δυο μορφές των μοντέλων επιφάνειας:
 - Τα μαθηματικά μοντέλα:
 - Είναι συνήθως ακριβείς αναπαραστάσεις αντικειμένων
 - Επιτρέπουν ακριβείς υπολογισμούς για το αντικείμενο (π.χ. κανονικό διάνυσμα)
 - Περιορίζονται σε ορισμένα είδη αντικειμένων
 - Δεν μπορούν να περιγράψουν τυχαία σχήματα
 - Τα πολυγωνικά μοντέλα:
 - Είναι προσεγγίσεις των αρχικών αντικειμένων
 - Αν χρησιμοποιούνται αρκετές κορυφές, είναι αρκετά ακριβείς προσεγγίσεις
 - Είναι πιο γενικά
 - Ακόμα και η μαθηματικές αναπαραστάσεις συνήθως κατασκευάζονται σε μια ‘διακριτή’ μορφή πολυγωνικών μοντέλων

Ανασκόπηση Μοντέλων (4)

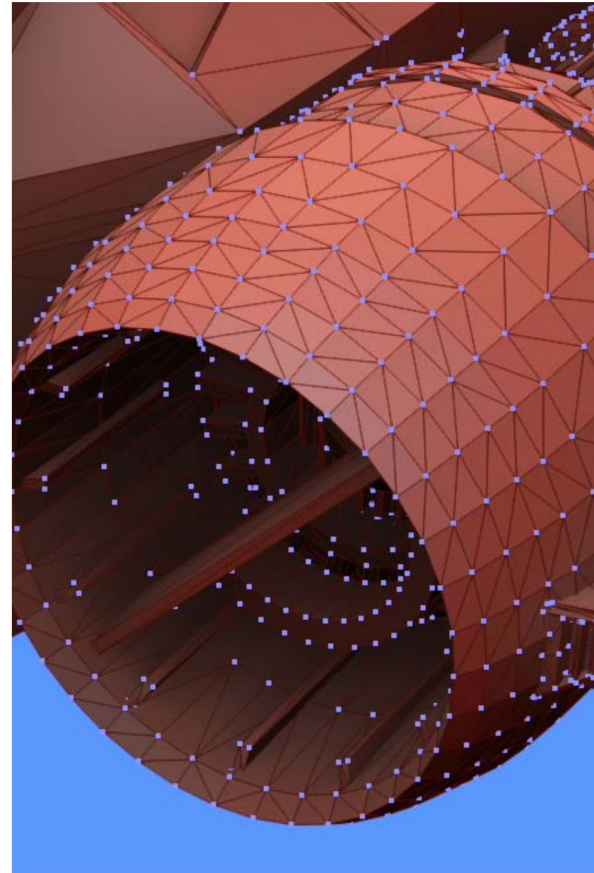
- Τα πολύγωνα μπορεί να αποτελούνται από οποιοδήποτε πλήθος κορυφών.
Στην πράξη:
 - Τετράπλευρα
 - Τρίγωνα
- Πολυγωνικά Μοντέλα Τετραπλεύρων:
 - Παράγονται κατά τον σχεδιασμό παραμετρικών επιφανειών
 - Δυστυχώς, ένα τετράπλευρο στις 3Δ δεν είναι απαραίτητα επίπεδο:
 - Περιορίζεται το σχήμα και η ευελιξία του μοντέλου
 - Δυσκολότεροι υπολογισμοί
- Πολυγωνικά Μοντέλα Τριγώνων:
 - Ένα τρίγωνο είναι πάντα επίπεδο
 - Κάθε πολύγωνο τριγωνοποιείται εύκολα → τριγωνικό μοντέλο παράγεται από οποιοδήποτε πολυγωνικό μοντέλο → τριγωνικά μοντέλα (ή τριγωνικά πλέγματα) χρησιμοποιούνται σχεδόν πάντα σε εφαρμογές που περιλαμβάνουν πολυγωνικά μοντέλα

Mesh – Πολυγωνικό μοντέλο



Triangle Meshes

- The most common type of polygonal mesh representation
- Very convenient to use in real-time rendering!

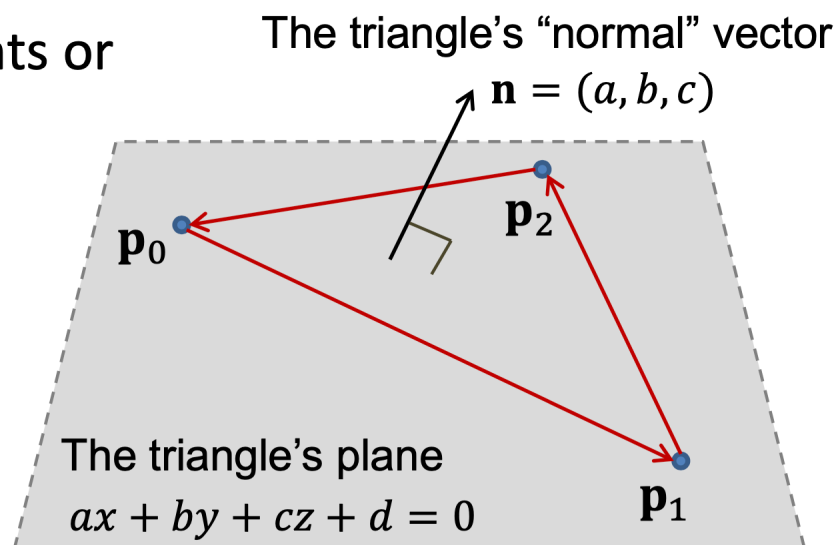


Συστατικά των Meshes

- What are their data?
 - Flat elements approximating both curved and flat surfaces
- Where are they?
 - Vertices of triangles (points on the plane or in space)
- What is their shape?
 - Connectivity among vertices defines the structure of the mesh
- How do they look?
 - Material and shading attributes per vertex – interpolated / predicted inside each triangle

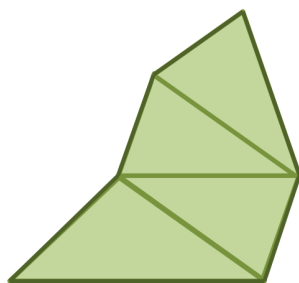
Το τρίγωνο

- A set of three (ordered) vertices (points in space)
- Connectivity:
 - Implied by order of points or
 - Given explicitly



Χρήσιμες ιδιότητες

- Minimal: The most elementary surface shape (3 points define an oriented plane)
- Always convex!
 - This is not true for generalized n -gons
 - A very useful property in calculations: its boundary coincides with its convex hull
- Any other polygon can be decomposed into a set of triangles!



Γραμμικός συνδυασμός στο επίπεδο

- Any point on the plane of the triangle can be uniquely written as a linear combination of the three triangle vertices. In fact:
- For any number k of vectors and scalar coefficients:

$$\mathbf{q} = \sum_{i=1}^k a_i \mathbf{v}_i \quad \text{is their linear combination}$$

Αναπαράσταση του εσωτερικού του τριγώνου

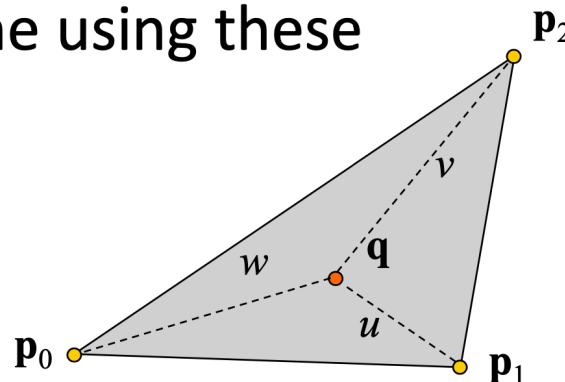
- Given the three triangle vertices (corners), any point inside the triangle (both in 2D and 3D) can be expressed as an affine combination of them
- This is an important property that translates to:
 - We only need the three corners of a triangle to fully describe its interior
- Relates to the topic of “affine transformations”

Βαρυκεντρικές συντεταγμένες

- To interpolate parameters across a triangle we need to find the set of (unique) parameters u, v, w that define $\mathbf{q}$ as the linear combination of all 3 vertices

$$\mathbf{q} = w\mathbf{p}_0 + u\mathbf{p}_1 + v\mathbf{p}_2$$

- All vertex attributes can be linearly interpolated at arbitrary locations on the plane using these **barycentric coordinates**



Βαρυκεντρικές συντεταγμένες

- Barycentric coordinates are extremely useful in triangle rendering, as they are used for:
 - **Interpolating** triangle properties for arbitrary points inside the triangle
 - Performing **point containment tests** for various primitive intersection tests

Βαρυκεντρικές συντεταγμένες

- They form a parametric space of:

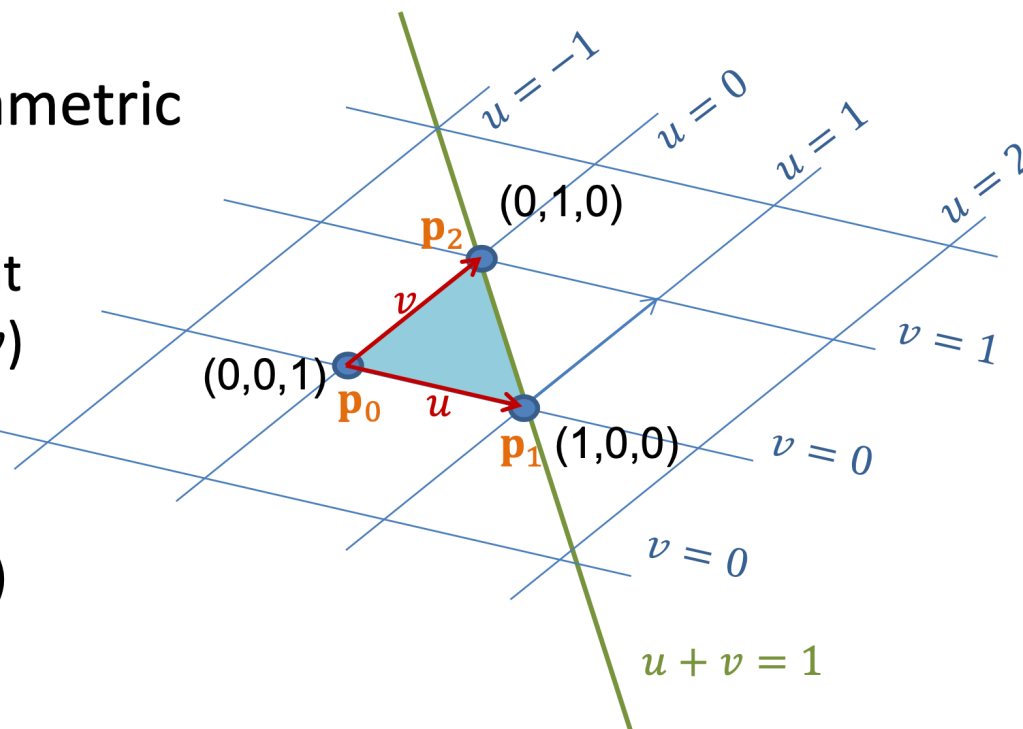
- two independent parameters (u, v)
- One dependent parameter

$$(w = 1 - u - v)$$

- All points with:

$$u > 0 \text{ AND } v > 0 \text{ AND } w > 0$$

Lie **inside** the triangle

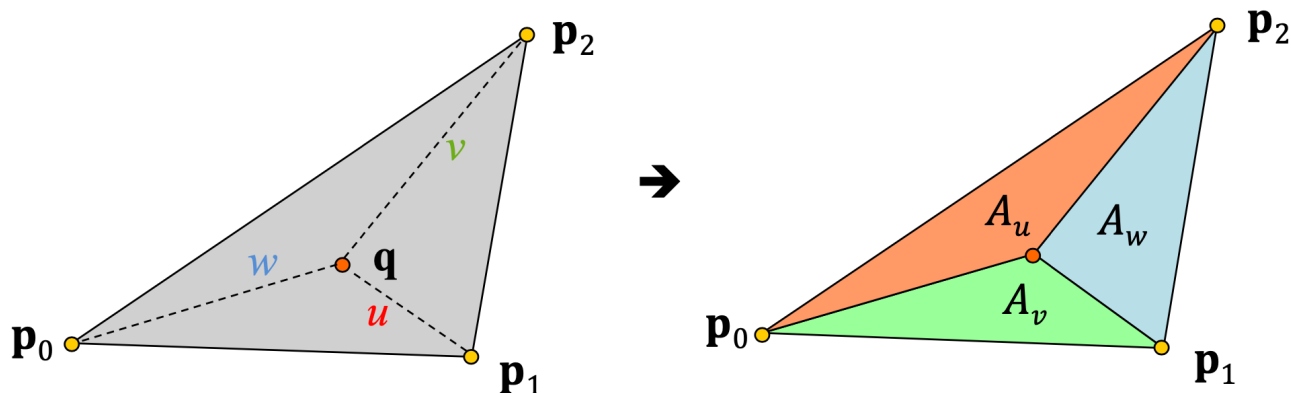


Βαρυκεντρικές συντεταγμένες

- Geometric properties:

- Barycentric coordinates equal the ratio of triangle areas formed by the opposite side and the query point against the total triangle area
- Can be exploited to compute them:

$$w = 1 - u - v, \quad u = \frac{A_u}{A_{total}}, \quad v = \frac{A_v}{A_{total}}$$

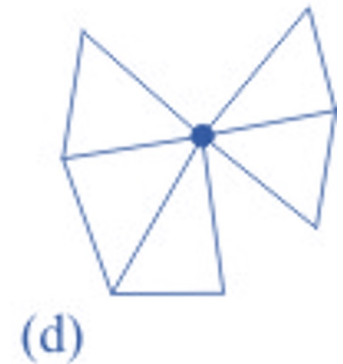
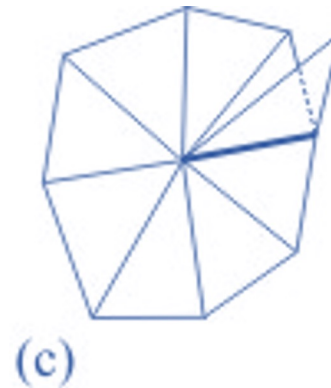
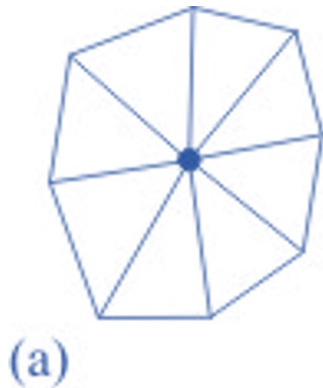


Ιδιότητες Πολυγωνικών Μοντέλων

- Μοντέλο επιφάνειας *πολλαπλότητας* 2Δ (ή απλά *πολλαπλότητα*): Κάθε σημείο της επιφάνειας έχει γειτνίαση ομοιομορφική με έναν ανοικτό δίσκο (εσωτερικό κύκλου)
 - Αν και η επιφάνεια υπάρχει στον 3Δ χώρο, όταν εξεταστεί σε μια μικρή περιοχή γύρω από ένα σημείο είναι τοπολογικά επίπεδη
- **Σε μια επιφάνεια-πολλαπλότητα:**
 - Κάθε ακμή μοιράζεται μεταξύ ακριβώς 2 εδρών
 - Γύρω από κάθε κορυφή υπάρχει ένας κλειστός βρόγχος εδρών
- **Επιφάνεια *πολλαπλότητα με σύνορο*:** Κάθε σημείο της επιφάνειας έχει γειτνίαση ομοιομορφική με έναν μισό δίσκο
- Σε μια επιφάνεια *πολλαπλότητα με σύνορο*:
 - Κάποιες ακμές (συνοριακές) ανήκουν ακριβώς σε μια έδρα
 - Γύρω από κάποιες κορυφές (συνοριακές) ο βρόγχος εδρών είναι ανοικτός
- Συνήθως, μια 3Δ επιφάνεια που δεν είναι *πολλαπλότητα με σύνορο* είναι μια *κλειστή επιφάνεια*

Ιδιότητες Πολυγωνικών Μοντέλων (2)

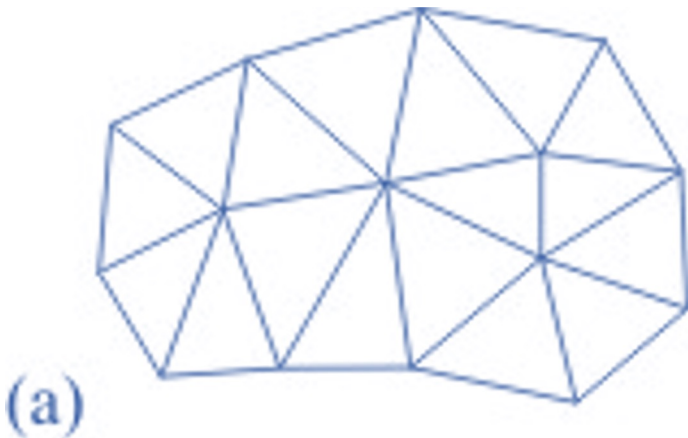
- (a) Τμήμα επιφάνειας πολλαπλότητας
- (b) Συνοριακή κορυφή επιφάνειας πολλαπλότητας με σύνορο
- (c) Ακμή που δεν ανήκει σε επιφάνεια πολλαπλότητας
- (d) Μη συνοριακή κορυφή που δεν ανήκει σε επιφάνεια πολλαπλότητας



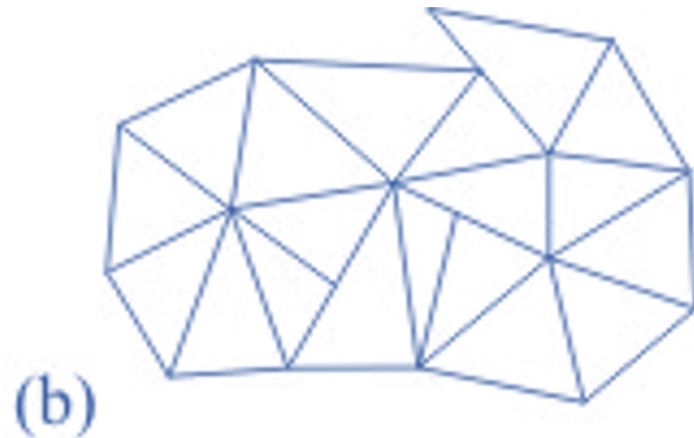
Ιδιότητες Πολυγωνικών Μοντέλων (3)

- Ένα μοντέλο επιφάνειας είναι ένα **απλοειδές σύμπλοκο** αν τα πολύγωνα που το αποτελούν εφάπτονται μόνο κατά μήκος των ακμών τους, και οι ακμές του μοντέλου τέμνονται μόνο στα άκρα τους

- (a) Απλοειδές σύμπλοκο
σύμπλοκο
(τριγωνικό πλέγμα)

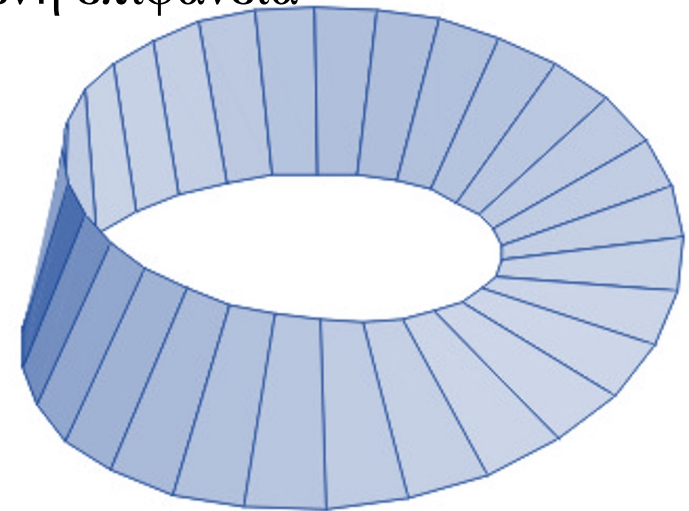


- (b) Μη απλοειδές
(τριγωνικό πλέγμα)



Ιδιότητες Πολυγωνικών Μοντέλων (4)

- Προσανατολισμένη επιφάνεια: Επιφάνεια που έχει 2 “όψεις”
 - Όπως μια κόλλα χαρτιού
- Οι περισσότερες επιφάνειες είναι προσανατολισμένες
- Στις κλειστές προσανατολισμένες επιφάνειες το ‘εσωτερικό’ & ‘εξωτερικό’ τμήμα της επιφάνειας είναι σαφώς διακριτά
- Κατά σύμβαση, το κανονικό διάνυσμα μιας κλειστής προσανατολισμένης επιφάνειας δείχνει προς την ‘έξω’ μεριά
- Η ταινία Moebius είναι μια μη προσανατολισμένη επιφάνεια



Ιδιότητες Πολυγωνικών Μοντέλων (5)

- Τα κλειστά μοντέλα πολλαπλότητας που είναι ομοιομορφικά ως προς τη σφαίρα, ικανοποιούν τον τύπο του Euler:

$$V - E + F = 2$$

όπου:

V: # κορυφών

E: # ακμών του μοντέλου

F: # εδρών

Ιδιότητες Πολυγωνικών Μοντέλων (6)

- Για ένα κλειστό τριγωνικό μοντέλο ο παραπάνω τύπος υποδηλώνει:
 - Ότι το πλήθος των τριγώνων του μοντέλου είναι σχεδόν 2-πλάσιο του πλήθους των κορυφών
 - Ότι ο μέσος αριθμός τριγώνων γύρω από μια κορυφή είναι 6
- Ο τύπος του Euler γενικεύεται για μοντέλα που δεν είναι πολλαπλότητες:

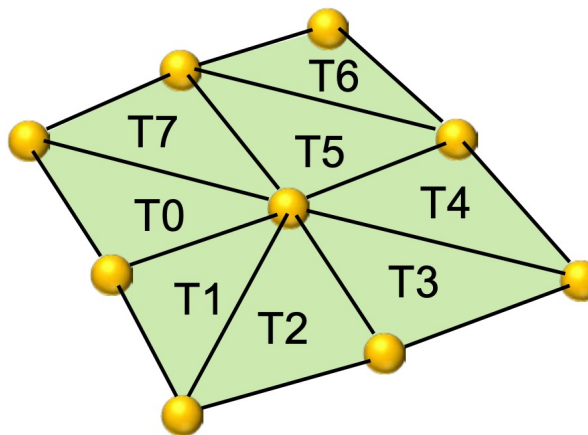
$$V - E + F = 2 - 2G$$

όπου G είναι το γένος (genus) του μοντέλου

- Το γένος ενός μοντέλου μπορεί να θεωρηθεί το πλήθος των διαμπερών οπών του μοντέλου:
 - Ο τόρος (torus) έχει γένος 1
 - Ο διπλός τόρος έχει γένος 2

Αναπαράσταση και αποθήκευση Mesh

- (Triangle) meshes are collections of triangles
- Can be:
 - A set of disjoint, independent triangles (a triangle “soup”)
 - A set of vertices “shared” among triangles, given a set of “connections”:



Triangle data

```
class Triangle
{
    float vertex[3][3];          // 3 vertices X 3 coords
    float vertex_normal[3][3];  // 3 vertices X 3 coords
    float vertex_color[3][3];   // 3 vertices X 3 components
    float plane_normal [3];     // xyz
    class Material * p_mat;     // pointer to material data
};
```



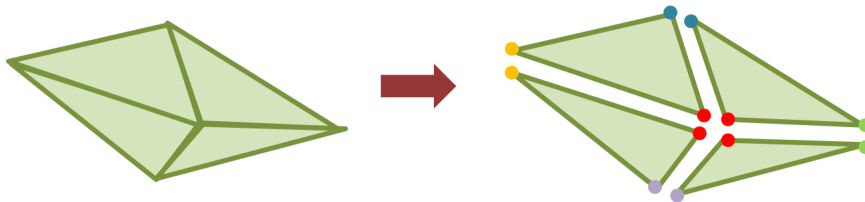
Per vertex attributes

Shared material: many triangles use the same material,
so the triangle only points to this single material instance

Can I do better?

Triangle data

- Can I do better?
 - Observe that many vertices are repeated in the triangle data:



- We can save significant storage and bandwidth (this will become important later – see real-time graphics) if we:
 - Keep per vertex attribute data separately
 - Index the vertex attributes

Δομές Δεδομένων για Πολυγωνικά Μοντέλα

- Πολλές δομές δεδομένων έχουν προταθεί για την αναπαράσταση πολυγωνικών μοντέλων. Διαφέρουν:
 - Στον τύπο των πολυγωνικών μοντέλων που μπορούν να αναπαραστήσουν
 - Στην ποσότητα & τύπο πληροφορίας που φυλάσσουν άμεσα για το μοντέλο
 - Στις λοιπές πληροφορίες που μπορούν να παράγουν έμμεσα για το μοντέλο
- Χρήσιμες πληροφορίες για τα Γραφικά:
 - *Τοπολογικές πληροφορίες*: αν το μοντέλο είναι πολλαπλότητα, είναι κλειστό, έχει όριο ή έχει τρύπες
 - *Πληροφορίες γειτνίασης*: γειτονικές έδρες δοθέντων ακμών και εδρών, γειτονικές ακμές και έδρες γύρω από δοθείσα κορυφή, σύνορο ενός ανοικτού μοντέλου
 - *Χαρακτηριστικά συνημμένα στο μοντέλο*: κανονικά διανύσματα, χρώματα, ιδιότητες υλικού, συντεταγμένες υφής

Άμεση λίστα ακμών /εδρών

- Οι βασικότερες δομές δεδομένων που χρησιμοποιούνται:
 - Άμεση λίστα ακμών
 - Άμεση λίστα εδρών } περιέχουν για κάθε ακμή/έδρα του μοντέλου τις συντεταγμένες των αντίστοιχων κορυφών

Παράδειγμα:

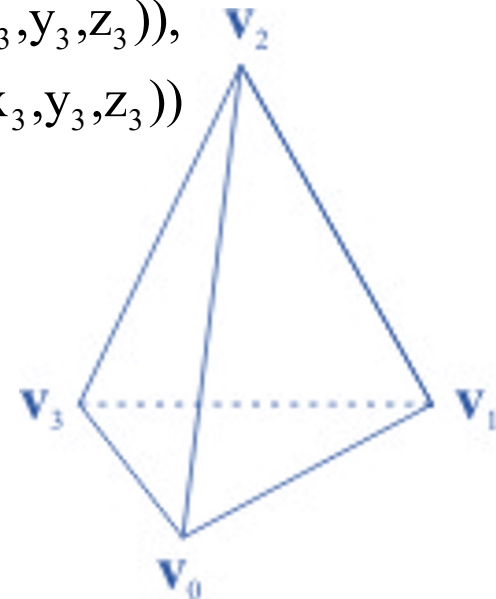
- Για το τετράεδρο της εικόνας ισχύει:
 - Λίστα ακμών:
 $e_0 = ((x_0, y_0, z_0), (x_1, y_1, z_1)), e_3 = ((x_1, y_1, z_1), (x_2, y_2, z_2)),$
 $e_1 = ((x_0, y_0, z_0), (x_2, y_2, z_2)), e_4 = ((x_1, y_1, z_1), (x_3, y_3, z_3)),$
 $e_2 = ((x_0, y_0, z_0), (x_3, y_3, z_3)), e_5 = ((x_2, y_2, z_2), (x_3, y_3, z_3))$
 - Λίστα εδρών:

$$f_0 = ((x_3, y_3, z_3), (x_2, y_2, z_2), (x_1, y_1, z_1)),$$

$$f_1 = ((x_2, y_2, z_2), (x_3, y_3, z_3), (x_0, y_0, z_0)),$$

$$f_2 = ((x_1, y_1, z_1), (x_0, y_0, z_0), (x_3, y_3, z_3)),$$

$$f_3 = ((x_0, y_0, z_0), (x_1, y_1, z_1), (x_2, y_2, z_2))$$

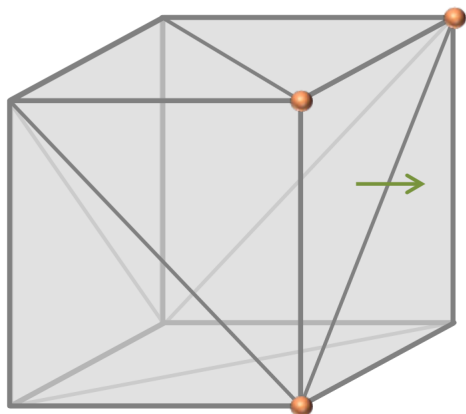


Δεικτοδοτημένη λίστα εδρών

- Αρκετά από τα μειονεκτήματα των προηγούμενων δομών επιλύονται από την δεικτοδοτημένη λίστα εδρών:
 - Περιέχει λίστα κορυφών & λίστα εδρών του μοντέλου
 - Οι κορυφές των εδρών δίνονται σαν αναφορές στη λίστα κορυφών
- Παράδειγμα: το προηγούμενο τετράεδρο αναπαρίσταται ως:
$$\begin{aligned} \mathbf{v}_0 &= (x_0, y_0, z_0), & \mathbf{f}_0 &= (\mathbf{v}_3, \mathbf{v}_2, \mathbf{v}_1), \\ \mathbf{v}_1 &= (x_1, y_1, z_1), & \mathbf{f}_1 &= (\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_0), \\ \mathbf{v}_2 &= (x_2, y_2, z_2), & \mathbf{f}_2 &= (\mathbf{v}_1, \mathbf{v}_0, \mathbf{v}_3), \\ \mathbf{v}_3 &= (x_3, y_3, z_3), & \mathbf{f}_3 &= (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2) \end{aligned}$$
- Για την αναπαράσταση προσανατολισμένων μοντέλων με χρήση της δεικτοδοτημένης λίστας εδρών, συνηθίζεται να ταξινομούνται οι κορυφές των εδρών είτε με την φορά των δεικτών του ρολογιού είτε αντίστροφα → ευκολότεροι υπολογισμοί

Triangle data – Attribute buffers

- Now we can have **separate attribute buffers** of different size:



12 triangles, indexing:



8 vertices



6 normals



1 color



Unindexed vs Indexed data

- Memory savings from this indexing (cube example):
 - Cost for per vertex data in unindexed cube triangles:
27 floats (4 bytes each) X 12 triangles = **1296 bytes**
 - Cost for per vertex data in indexed cube triangles:
9X12 integer indices (4 bytes each) = 432 bytes
8 vertices X 3 floats = 96 bytes
6 normals X 3 floats = 72 bytes
1 color X 3 floats = 12 bytes
total: **612 bytes**

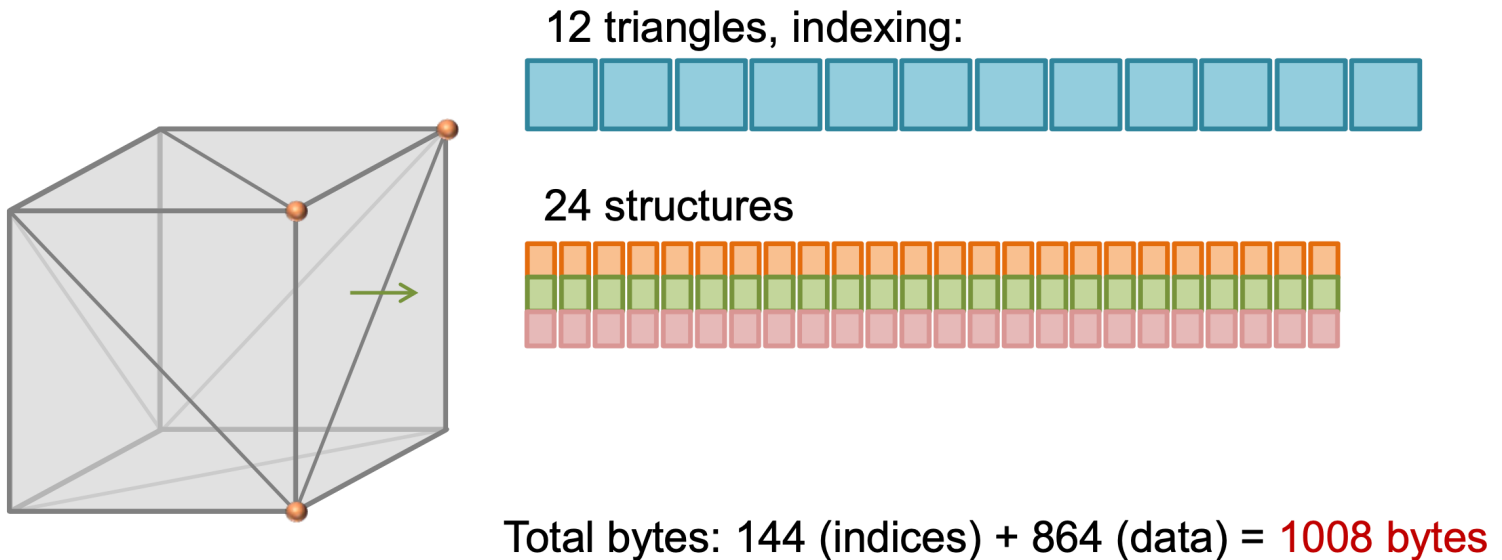
Attributes structures

- In this example, we waste a lot of memory in indexing
- Alternatively, we could use a per vertex structure and index a single array of *vertex objects*:

```
class Triangle
{
    unsigned int index[3];    // 3 vertices X 1 structure index
    float plane_normal [3];  // xyz
    class Material * p_mat;   // pointer to material data
};
```

Attributes structures

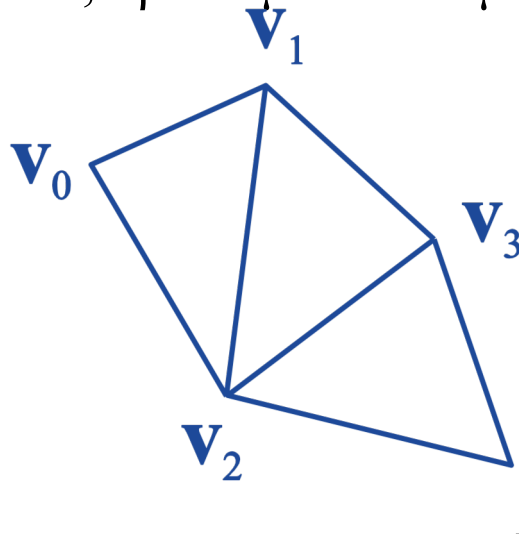
- Now we have a single **attribute buffer**:



This is larger than the separate data buffer size!!! Then why use it...???

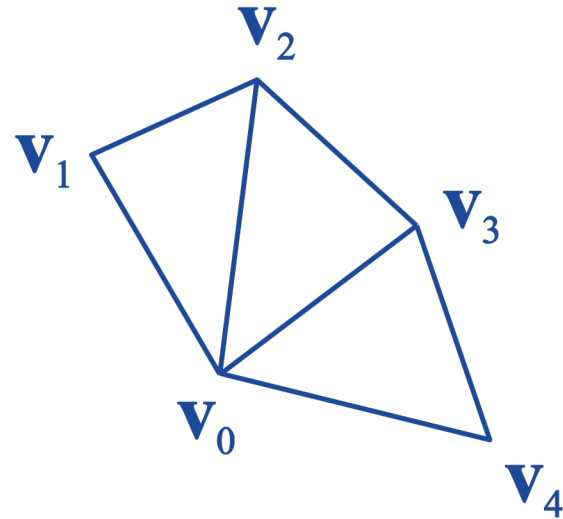
Δομές Δεδομένων για Πολυγωνικά Μοντέλα (6)

- Ειδικά για τριγωνικά μοντέλα, τα γειτονικά τρίγωνα μπορούμε να τα χειριστούμε πιο αποδοτικά σαν *ταινία τριγώνων* ή *βεντάλια τριγώνων*, προκειμένου να μειωθούν οι επαναλήψεις δεδομένων



Ταινία τριγώνων

Βεντάλια τριγώνων

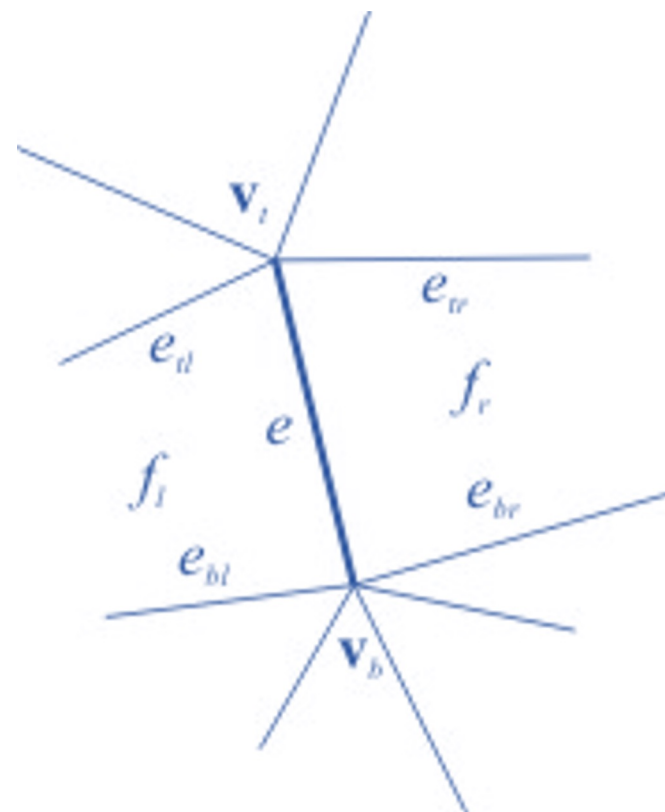


$(v_0, v_1, v_2, v_3, v_4)$

$(v_0, v_1, v_2, v_3, v_4)$

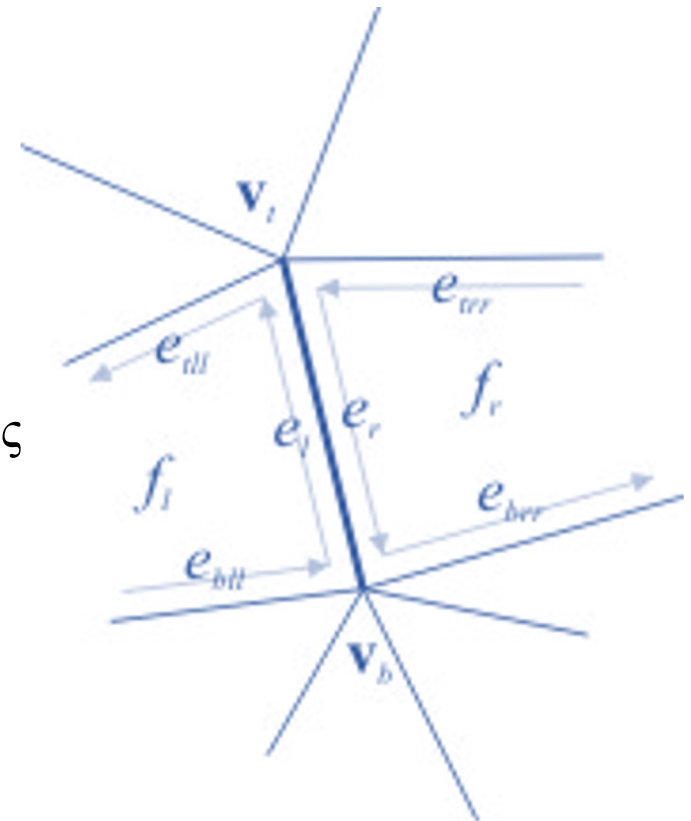
Δομές Δεδομένων για Πολυγωνικά Μοντέλα (8)

- Η δομή *winged-edge*, είναι μια τέτοια δομή:
 - Κεντρικός κόμβος πληροφορίας είναι η ακμή
 - Κάθε ακμή αποθηκεύει αναφορές:
 - Στις 2 κορυφές της
 - Στις 2 γειτονικές της έδρες
 - Στις 4 γειτονικές της ακμές



Δομές Δεδομένων για Πολυγωνικά Μοντέλα (9)

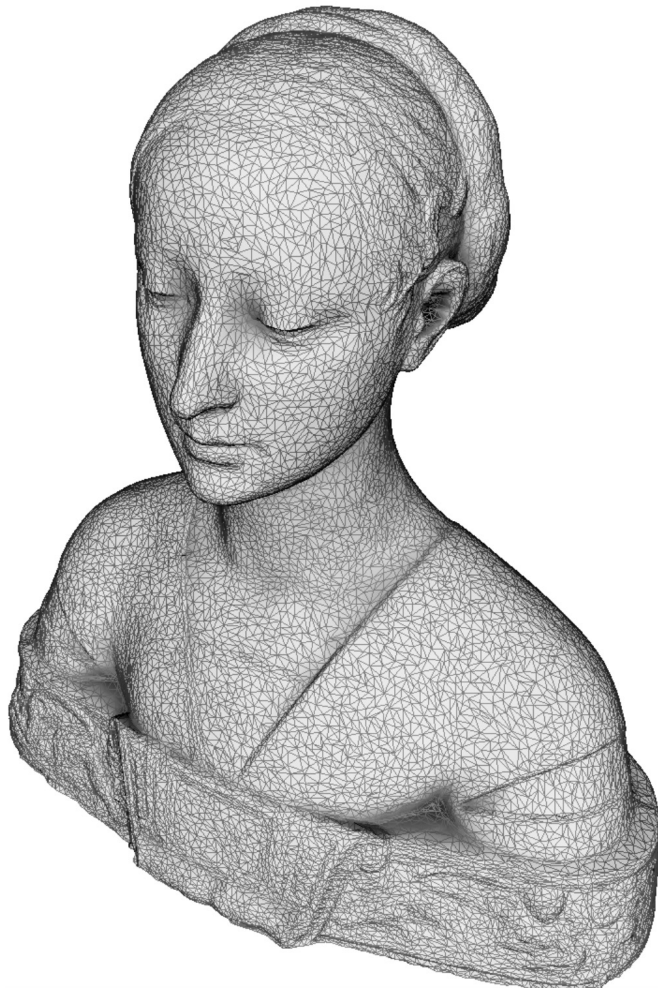
- Η δομή ημι-ακμής είναι παρόμοια με την δομή winged-edge, αλλά χρησιμοποιεί προσανατολισμένες ακμές:
 - Κάθε ακμή διαχωρίζεται σε 2 ημι-ακμές
 - Κάθε ημι-ακμή αποθηκεύει μια αναφορά:
 - Στην κορυφή που αρχίζει & την κορυφή που τελειώνει
 - Στη γειτονική έδρα της
 - Στις 2 γειτονικές της ημι-ακμές κατά μήκος της γειτονικής της έδρας
 - Στην αντίθετή της ημι-ακμή
- Η δομή ημι-ακμής είναι πιο αποδοτική από την δομή winged-edge για αρκετές ερωτήσεις γειτνίασης



Δομές Δεδομένων για Πολυγωνικά Μοντέλα (10)

- Η δομή *Quad-edge* είναι παρόμοια με τις παραπάνω:
 - Είναι πιο πολύπλοκη
 - Χρησιμοποιείται για τον αποδοτικό υπολογισμό ερωτήσεων γειτνίασης
 - Μπορεί να αναπαραστήσει ταυτόχρονα ένα μοντέλο πολλαπλότητας & το *δυϊκό* του
 - Το *δυϊκό* μοντέλο κατασκευάζεται περιστρέφοντας τις ακμές κατά 90° , και αντικαθιστώντας τις κορυφές με έδρες & αντίστροφα
- Π.χ. Το *δυϊκό* ενός τετραέδρου είναι ένα τετράεδρο
- Το *δυϊκό* ενός κύβου είναι ένα οκτάεδρο & αντίστροφα
- Χρήσιμη στην *υπολογιστική γεωμετρία*, την αλγοριθμική δηλαδή μελέτη γεωμετρικών προβλημάτων

A more typical indexed Mesh



- Using unindexed triangles (positions+normals):
3,596,688 bytes
- Using separate attribute buffers:
1,867,560 bytes
- Using a single vertex object buffer:
1,268,112 bytes

Απλοποίηση Πολυγωνικού Μοντέλου

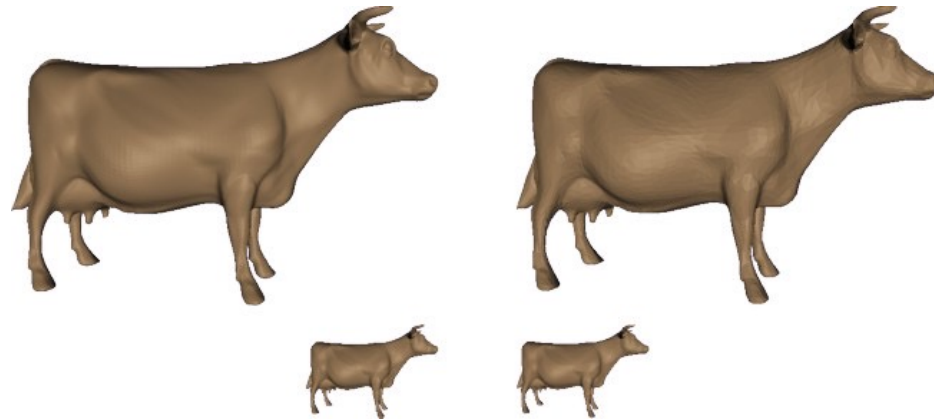
- Τα πολυγωνικά μοντέλα συχνά δημιουργούνται αυτόματα με:
 - Σχεδίαση των επιφανειών, που είναι ορισμένες με μαθηματικό τρόπο
 - 3Δ σάρωση αληθινών αντικειμένων
- Η αύξηση της δύναμης των υπολογιστών και τα πλεονεκτήματα της 3Δ σάρωσης οδηγούν σε μοντέλα με μεγάλο αριθμό κορυφών και εδρών
- Το έργο *Digital Michelangelo*, π.χ., χρησιμοποίησε σαρωτές προηγμένης τεχνολογίας για να σαρώσει και ανακατασκευάσει κάποια από τα γλυπτά του Μιχαήλ Άγγελου
 - Αποτέλεσμα: εκατομμύρια τρίγωνα (1/4 mm πλέγμα σάρωσης)
 - Μερικά gigabytes δεδομένα
 - Το μέγεθος της πληροφορίας είναι δύσκολο στην επεξεργασία
 - Αυτή η λεπτομέρεια είναι χρήσιμη μόνο σε συγκεκριμένες εφαρμογές

Απλοποίηση Πολυγωνικού Μοντέλου (2)

- Οι εφαρμογές των γραφικών μπορούν να ωφεληθούν από πολλαπλές αναλύσεις (επίπεδα λεπτομέρειας (LODs)) του μοντέλου που μπορούν να χρησιμοποιηθούν σε διαφορετικές συνθήκες θέασης
 - Όταν η προβολή του μοντέλου είναι μικρή, μόνο ένα μικρό τμήμα λεπτομέρειας είναι ορατό
- Είναι επίσης ωφέλιμο να μεταβάλλεται η λεπτομέρεια σε διαφορετικές περιοχές του ίδιου μοντέλου
 - Συνεπίπεδα τρίγωνα συγχωνεύονται σε λιγότερα και μεγαλύτερα τρίγωνα
 - Περιοχές κοντά στον παρατηρητή χρειάζονται περισσότερη λεπτομέρεια από εκείνες που βρίσκονται μακριά
- Τα επίπεδα λεπτομέρειας και η επιλεκτική εκλέπτυνση είναι κατάλληλα για αλληλεπιδραστικές εφαρμογές

Απλοποίηση Πολυγωνικού Μοντέλου (3)

- Διαφορετικά επίπεδα λεπτομέρειας και μεγέθη: (5000 –αριστερά- vs. 1000 –δεξιά- τρίγωνα)



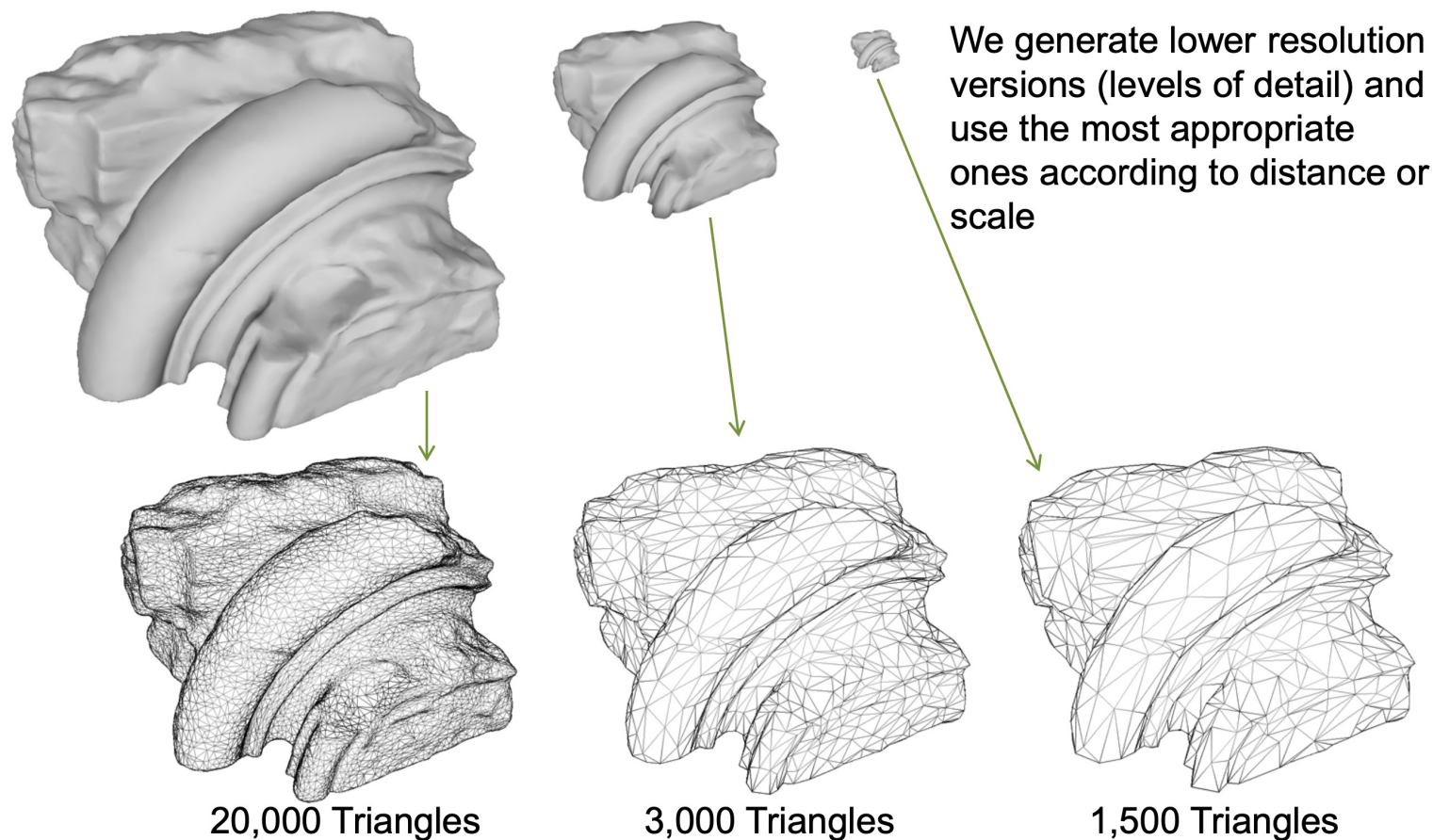
- Έχουν αναπτυχθεί αρκετές τεχνικές απλοποίησης μοντέλων
- Μειώνουν τον αριθμό των εδρών ενός πολυγωνικού μοντέλου, ενώ παράλληλα προσπαθούν να διατηρούν την εμφάνιση και δομή του αρχικού μοντέλου
- Οι σχετικές εφαρμογές, συνήθως έχουν το αρχικό μοντέλο σε διαφορετικά επίπεδα λεπτομέρειας και δυναμικά επιλέγουν την κατάλληλη

Level of Detail (LoD)

- For distant/small versions of objects, detail is lost, i.e. cannot be sufficiently sampled and displayed
- There is no point in attempting to display rich detail if we are not going to see it! → waste of bandwidth and processing power



Level of Detail (LoD)



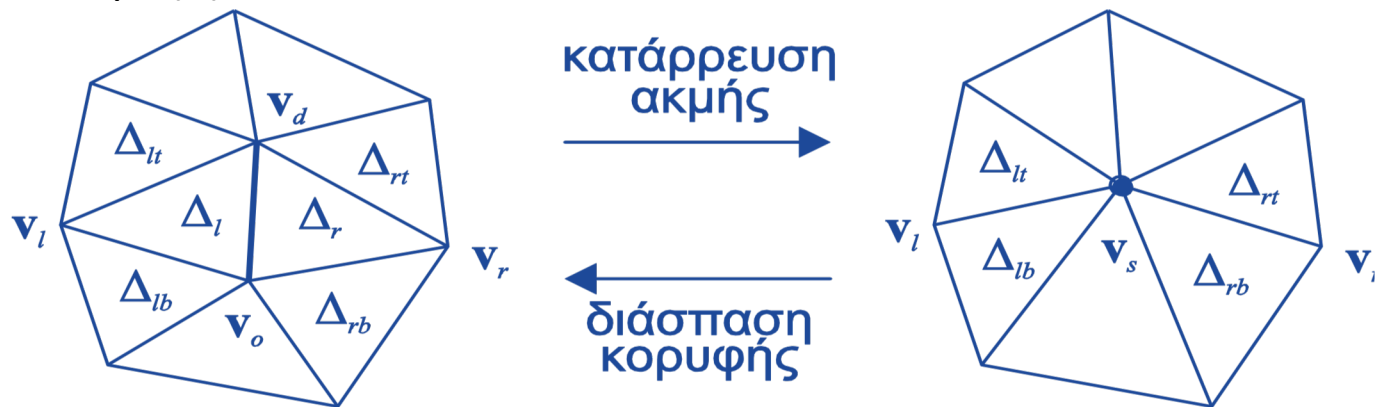
Απλοποίηση Πολυγωνικού Μοντέλου (4)

- Οι αλγόριθμοι απλοποίησης:
 - Αντιμετωπίζουν ευκολότερα κλειστά πλέγματα που είναι πολλαπλότητες
 - Χειρίζονται συχνά και το σύνορο των μη-κλειστών μοντέλων

Απλοποίηση με Διαδοχικές Καταρρεύσεις Ακμών

- Κατάρρευση Ακμής (edge collapse):

- Τοπική πράξη σε τριγωνικό πλέγμα
- Αφαιρεί μια ακμή του μοντέλου και 2 γειτονικά της τρίγωνα με την κατάρρευση μιας ακμής σε μία κορυφή



- Χρησιμοποιώντας κατάρρευση ακμών, είναι εύκολος ο υπολογισμός της απόστασης μεταξύ του απλοποιημένου και του αρχικού μοντέλου → διαφέρουν στις έδρες γύρω από την ακμή που κατέρρευσε
- Υποθέτουμε μοντέλα που είναι πολλαπλότητες, αλλά υπάρχουν και παραλλαγές για μη πολλαπλότητες

Why Different Representations?

Efficiency for different tasks

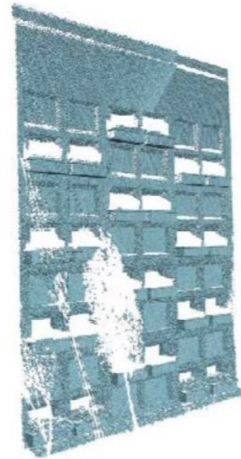
- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation

→ Data structures determine algorithms

Why Different Representations?

Efficiency for different tasks

- Acquisition
 - Range Scanning
- Rendering
- Analysis
- Manipulation
- Animation



Live Body Scan
Data acquired in 0.01 seconds



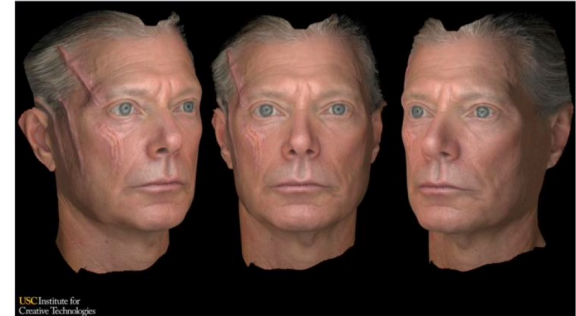
Why Different Representations?

Efficiency for different tasks

- Acquisition
 - Computer Vision
- Rendering
- Analysis
- Manipulation
- Animation



Indiana University

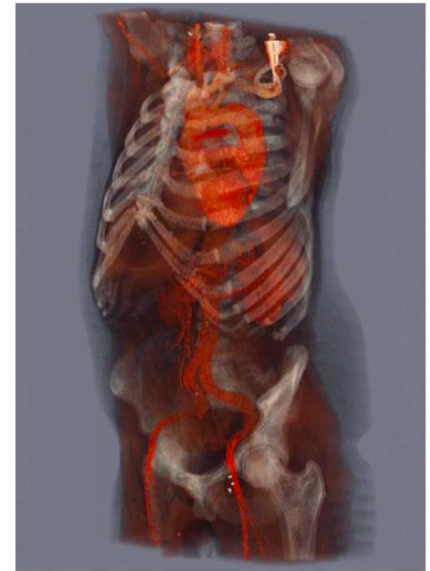
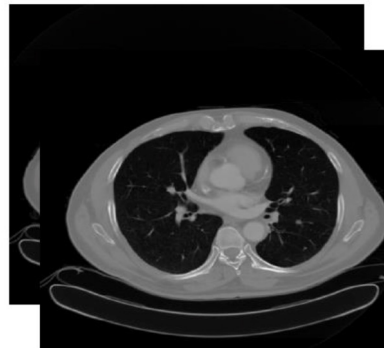


© 2012 CBS Interactive

Why Different Representations?

Efficiency for different tasks

- Acquisition
 - Tomography
- Rendering
- Analysis
- Manipulation
- Animation



DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

- Acquisition
- **Rendering**
 - Intersection
- Analysis
- Manipulation
- Animation

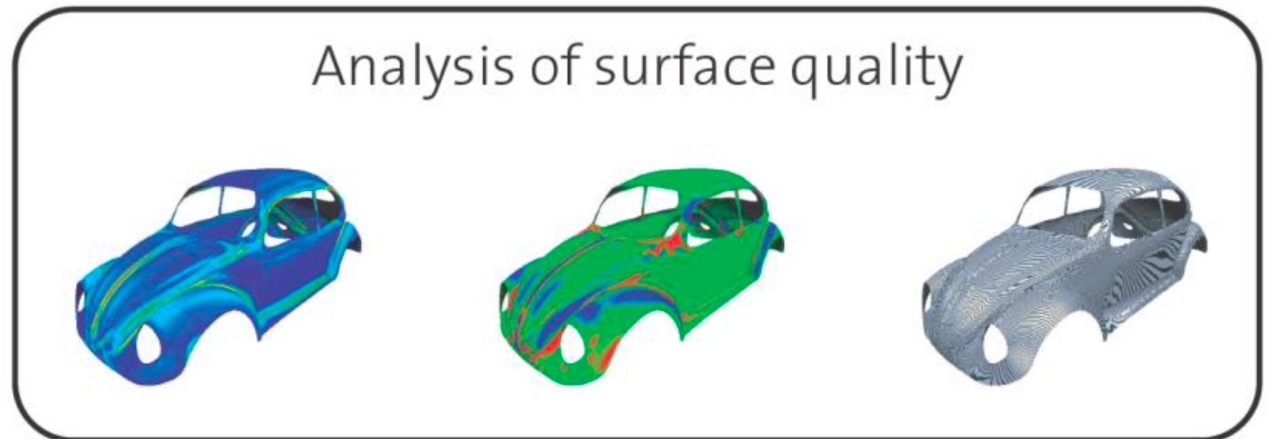


Autodesk

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- **Analysis**
 - Curvature, smoothness
- Manipulation
- Animation

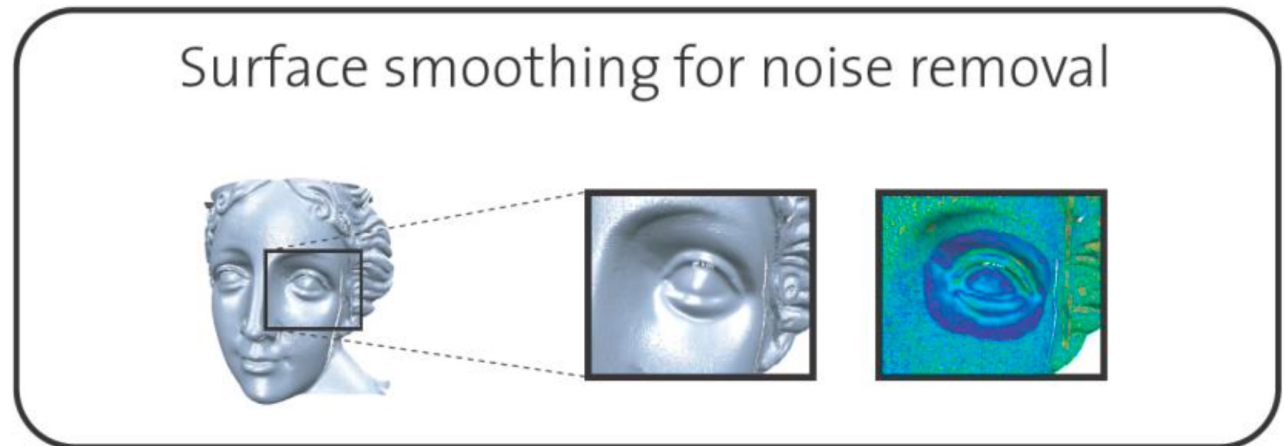


DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- **Analysis**
 - **Fairing**
- Manipulation
- Animation



DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

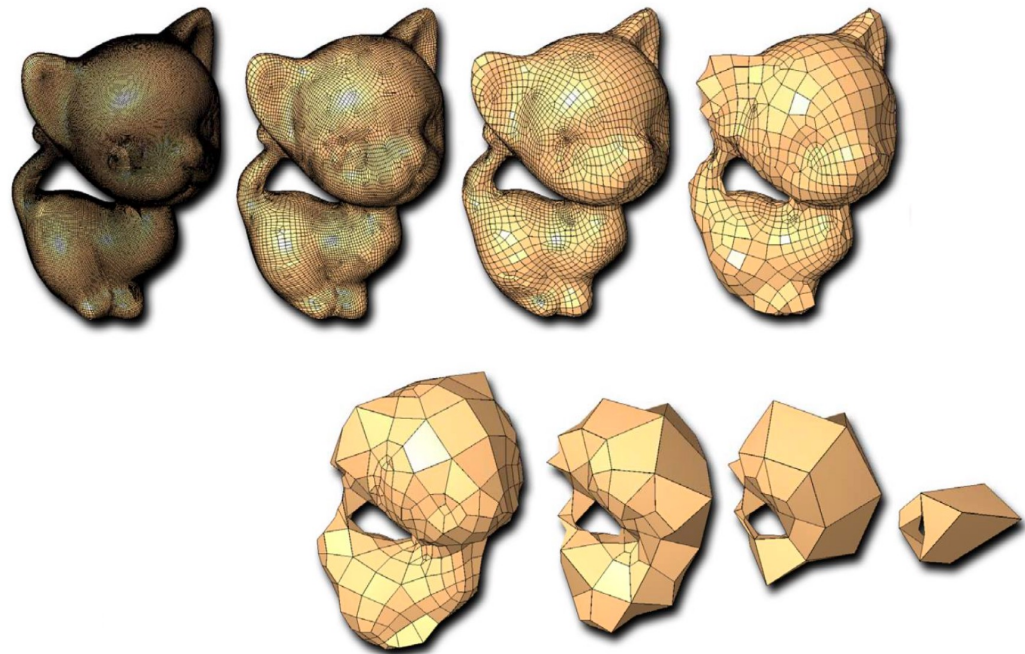
- Acquisition
- Rendering
- Analysis
 - Texture mapping
- Manipulation
- Animation



Why Different Representations?

Efficiency for different tasks

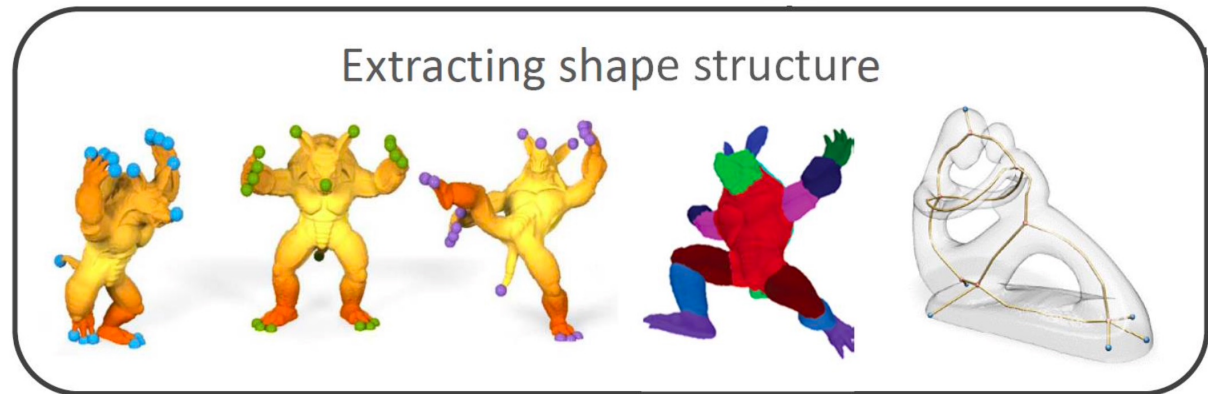
- Acquisition
- Rendering
- **Analysis**
 - Reduction
- Manipulation
- Animation



Why Different Representations?

Efficiency for different tasks

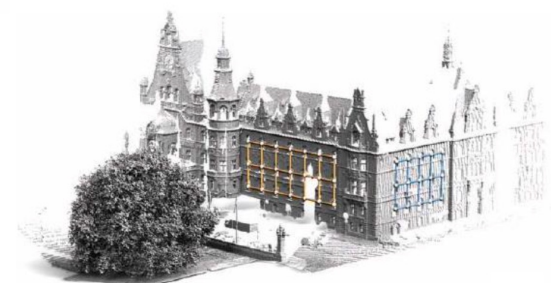
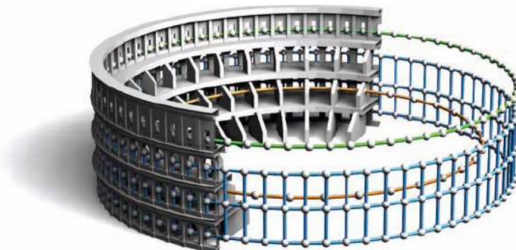
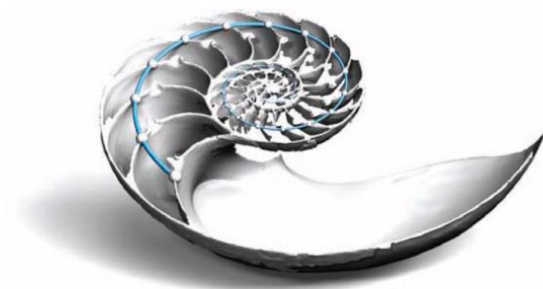
- Acquisition
- Rendering
- **Analysis**
 - **Structure**
- Manipulation
- Animation



Why Different Representations?

Efficiency for different tasks

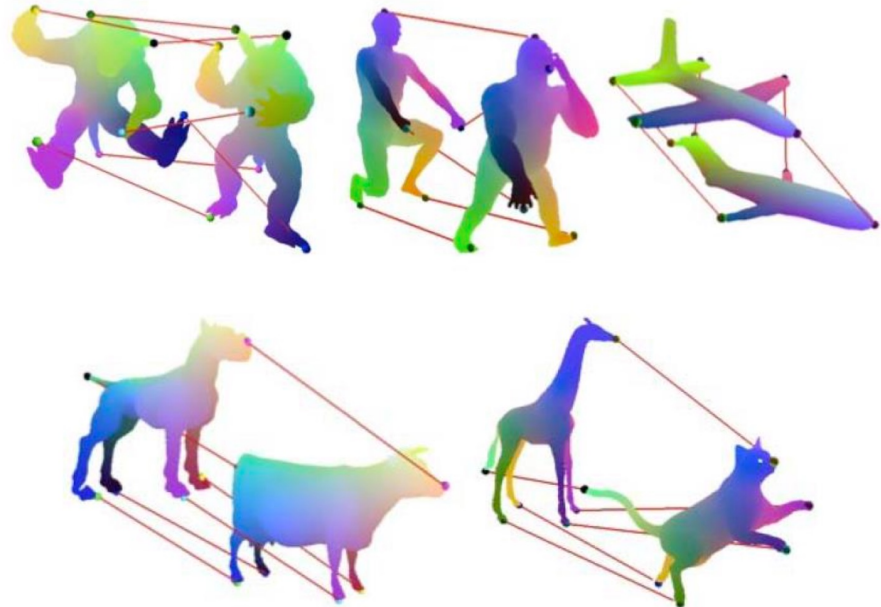
- Acquisition
- Rendering
- Analysis
 - Symmetry detection
- Manipulation
- Animation



Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- **Analysis**
 - Correspondence
- Manipulation
- Animation



Why Different Representations?

Efficiency for different tasks

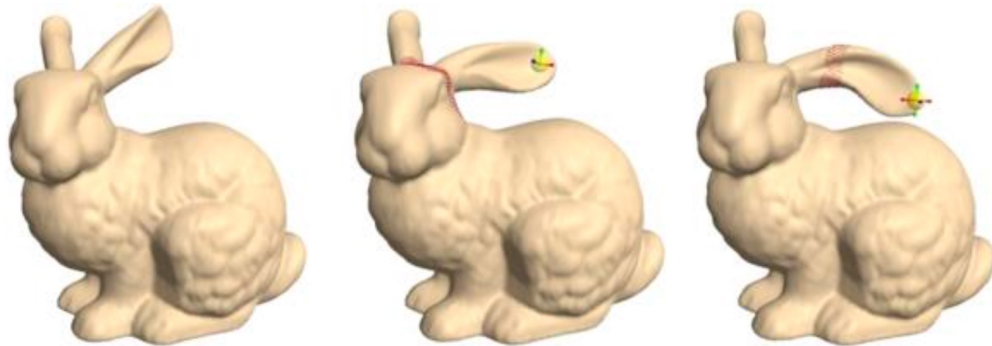
- Acquisition
- Rendering
- Analysis
 - Segmentation
- Manipulation
- Animation



Why Different Representations?

Efficiency for different tasks

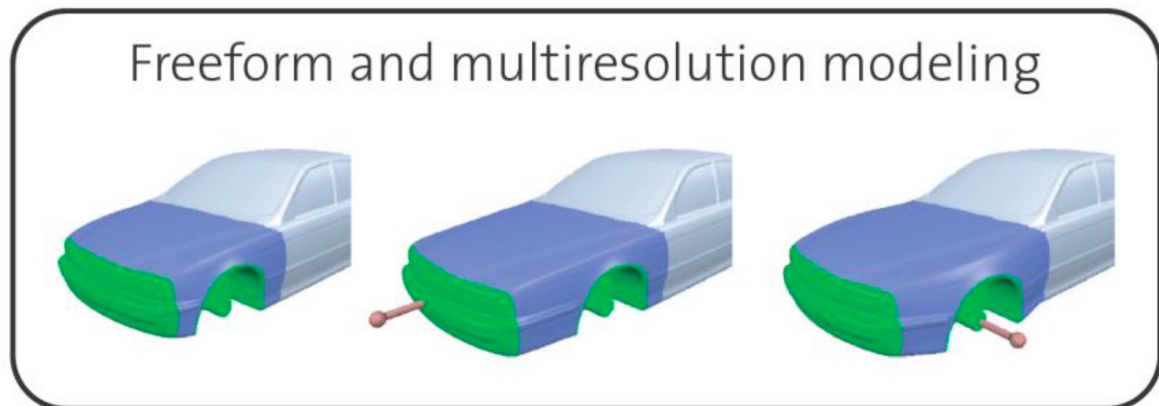
- Acquisition
- Rendering
- Analysis
- **Manipulation**
 - Deformation
- Animation



Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- **Manipulation**
 - Deformation
- Animation

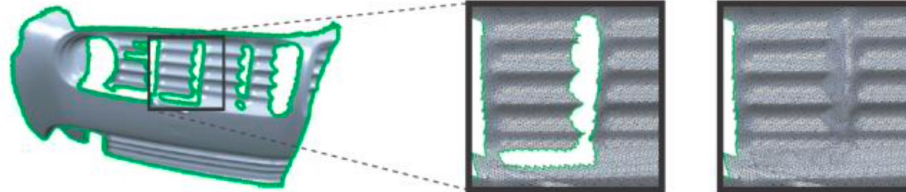


Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- **Manipulation**
 - Healing
- Animation

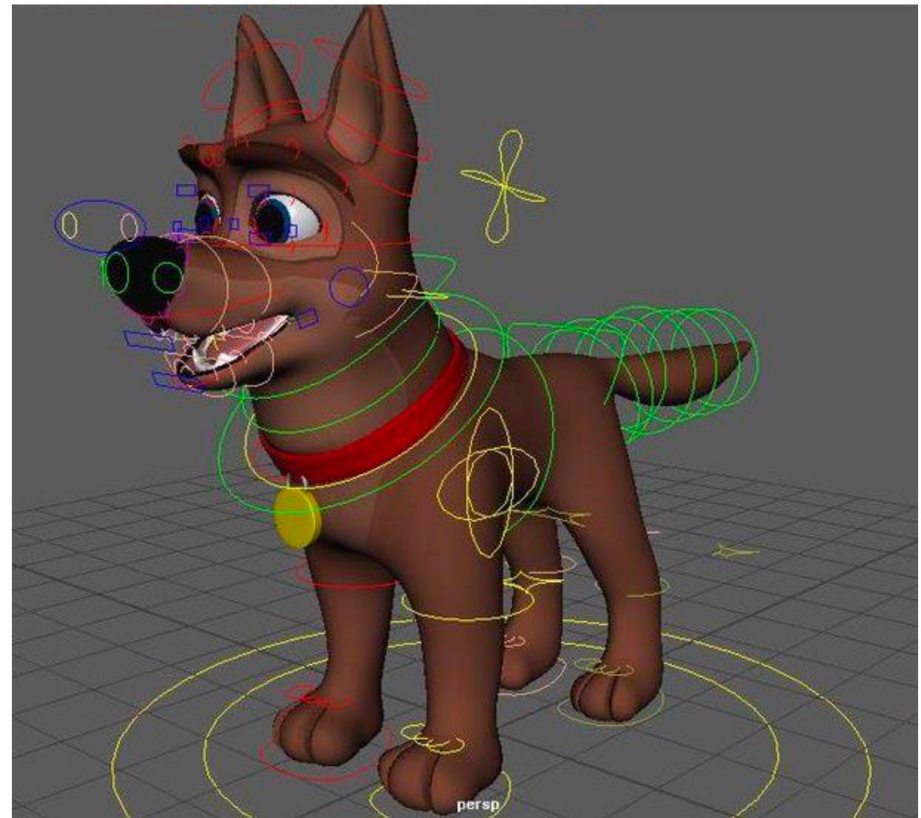
Removal of topological and geometrical errors



Why Different Representations?

Efficiency for different tasks

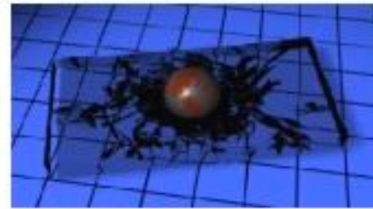
- Acquisition
- Rendering
- Analysis
- Manipulation
- **Animation**
 - Rigging



Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- **Animation**
 - Simulation



Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- **Animation**
 - **Fabrication**



Scene creation

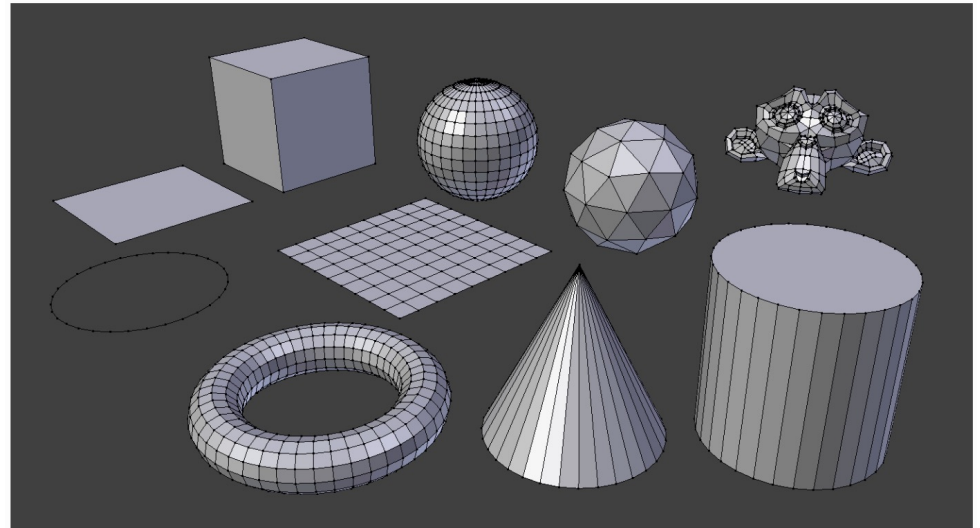
The creation of a 3D scene needs at least three key components:

- Models,
- materials and
- lights.

Modeling is simply the art and science of creating a surface that either mimics the shape of a real-world object or expresses your imagination of abstract objects.

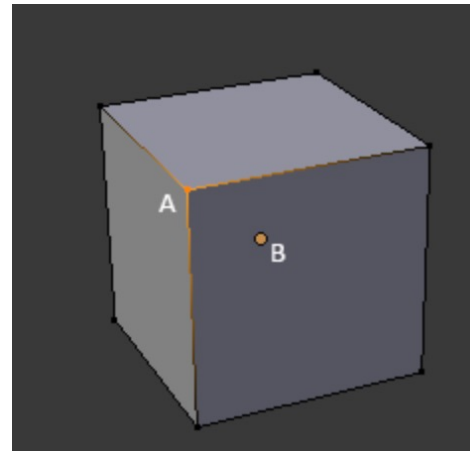
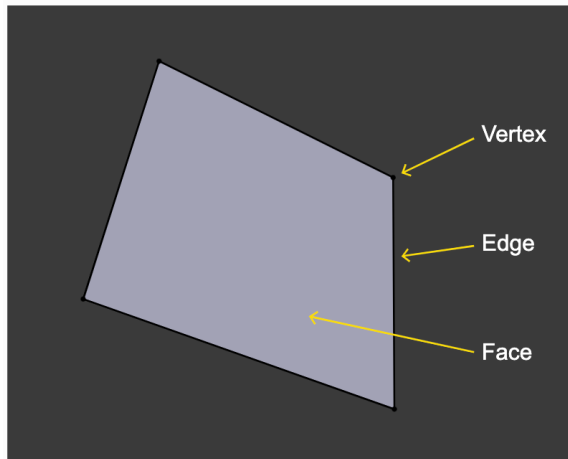
Mesh modeling

- Mesh Modeling typically begins with a **Mesh Primitive** shape (e.g. circle, cube, cylinder...).
- From there you might begin editing to create a larger, more complex shape.
- Plane, Cube, Circle, UV Sphere, Isosphere, Cylinder, Cone, torus, Grid



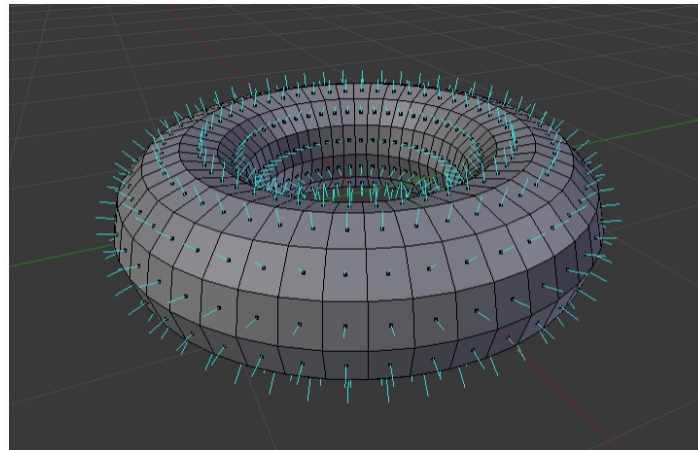
Structure

Meshes are built using three basic structures:
vertices, edges and faces.



Normals

- In geometry, a normal is a direction or line that is perpendicular to something a triangle or surface
- each blue line represents the normal for a face on the torus. The lines are each perpendicular to the face on which they lie



Topology

- Edge loops
- Face loops

