

Το Σύστημα Απόδοσης Γραφικών



Πριν ξεκινήσουμε: Voxel Game Engines

- Βασίζονται σε voxels (όγκους) και όχι σε πολύγωνα (επιφάνειες).
- Open Source: [Luanti](#)

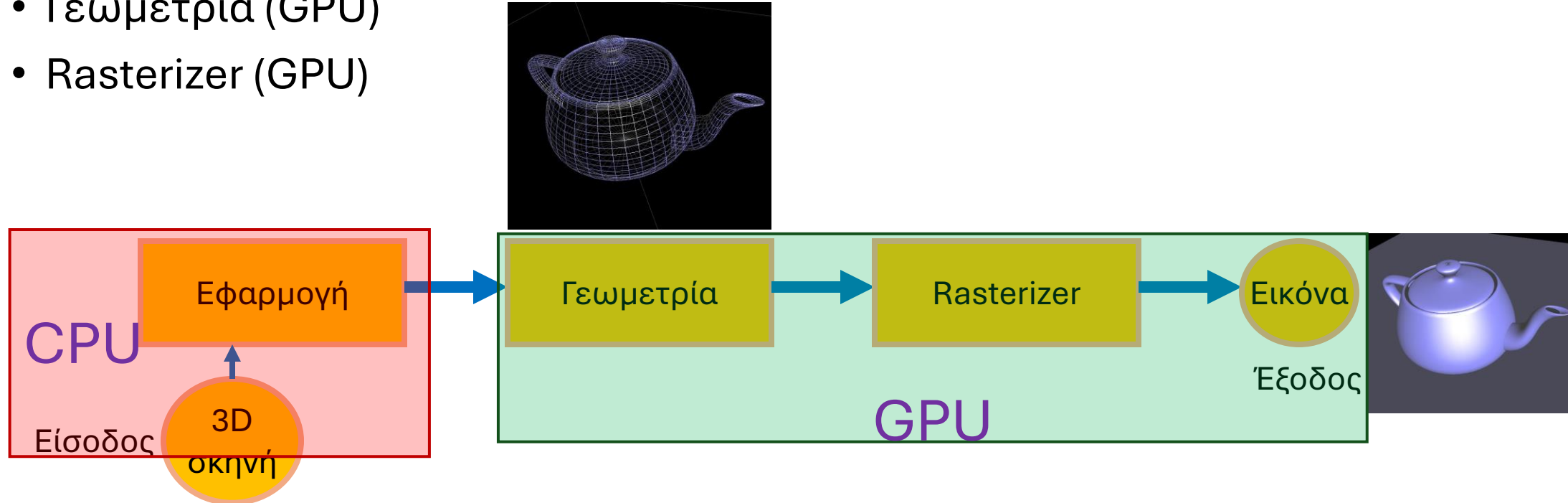


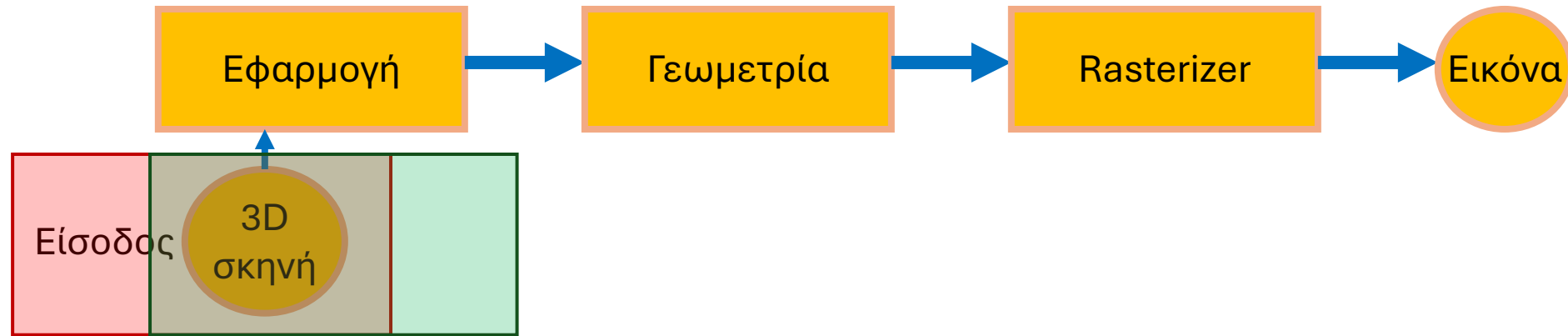
Σύγκριση:

- + διαδικασιακή παραγωγή στοιχείων – κόσμων, περιβάλλοντα που καταστρέφονται εύκολα, gameplay τύπου Minecraft
- Ρεαλισμός και μνήμη

Η Ροή Εργασίας για Απόδοση Γραφικών

- Δημιουργεί εικόνες από 3d σκηνές
- Τρία εννοιολογικά στάδια:
 - Εφαρμογή (το παιχνίδι (ή ό,τι άλλο) που εκτελείται στη CPU)
 - Γεωμετρία (GPU)
 - Rasterizer (GPU)



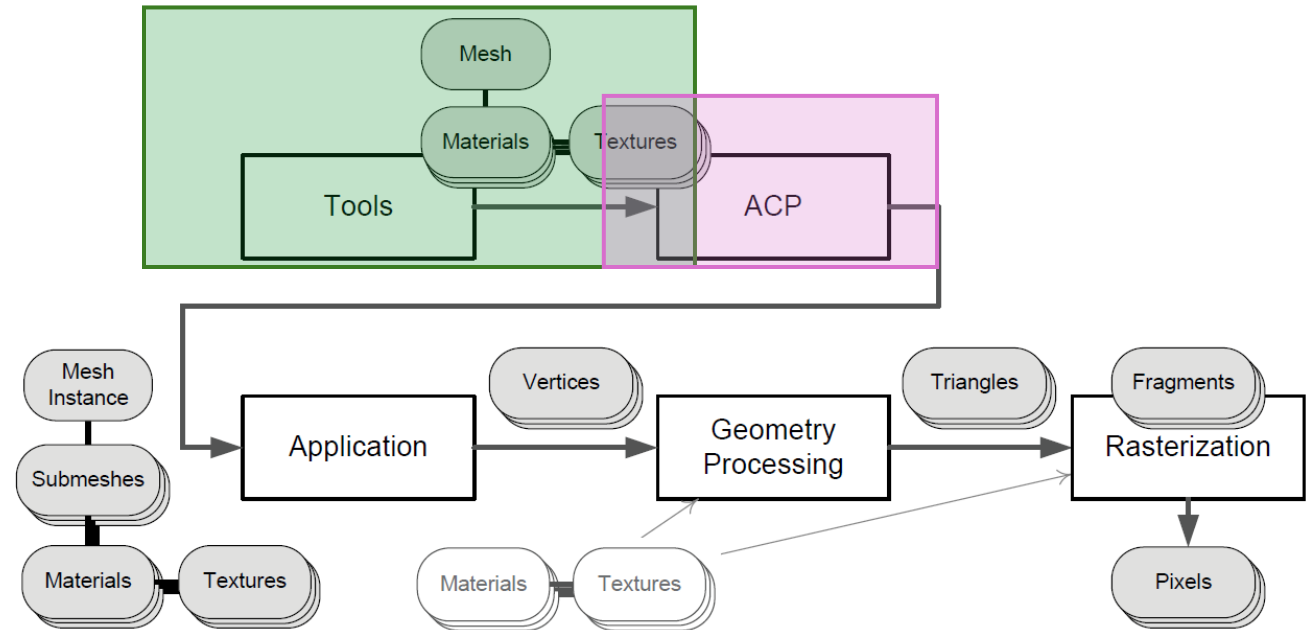


Πριν το Στάδιο της Εφαρμογής

Offline

Εργαλεία – ACP (Asset Conditioning Pipeline)

- Ψηφιακή δημιουργία περιεχομένου (Maya, Blender, Photoshop, Krita, ...)
- ACP
 - Μετατροπή σε μορφή συμβατή με αυτή της μηχανής
 - Κατασκευή δομών δεδομένων για το γράφημα σκηνής
 - Ακριβοί υπολογισμοί φωτισμού (στατικός φωτισμός)
- Μπορεί να χρησιμοποιείται η GPU για απαιτητικούς παραλληλοποιήσιμους υπολογισμούς

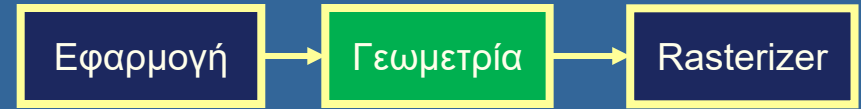


Στάδιο Εφαρμογής



- Εκτελείται στη CPU
 - Ο προγραμματιστής είναι υπεύθυνος για ό,τι συμβαίνει εδώ
- Παραδείγματα:
 - Ανίχνευση Σύγκρουσης
 - Animation
 - Λογική Κάμερας, AI
- Σημαντική εργασία: στείλε τα γεωμετρικά στοιχεία (π.χ., τρίγωνα) στη GPU

Στάδιο Γεωμετρίας



- Γεωμετρικές πράξεις στα στοιχεία εισόδου (κορυφές – τρίγωνα)
- Επιτρέπει:
 - Μετακίνηση αντικειμένων
 - Μετακίνηση κάμερας
 - Υπολογισμός φωτισμού στις κορυφές των τριγώνων
 - Προβολή στο επίπεδο (από 3d σε 2d)
 - Αποκοπή (αποφυγή απόδοσης τριγώνων εκτός οθόνης και όχι μόνο)
 - Προβολή στην οθόνη

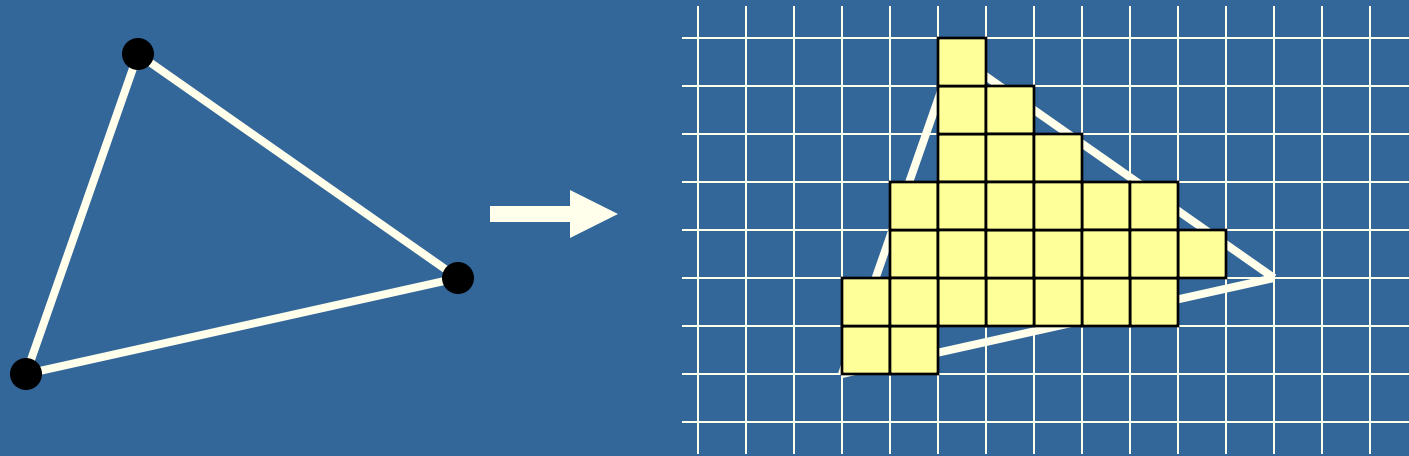
Το Στάδιο του Rasterizer

Εφαρμογή

Γεωμετρία

Rasterizer

- Πάρε την έξοδο από το στάδιο της γεωμετρίας και μετέτρεψε το σε pixels



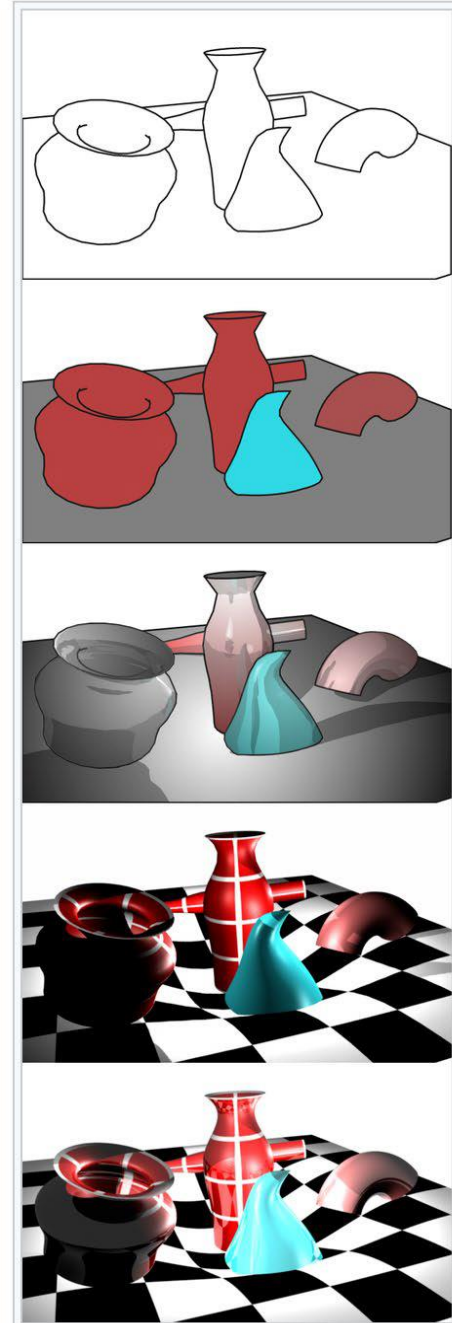
- Πρόσθεσε υφές και κάνε και άλλες πράξεις σε επίπεδο fragment/pixel
- Η ορατότητα επίσης μπορεί να καθοριστεί σε αυτό το σημείο: ταξινόμηση των τριγώνων κατά τη z διάσταση

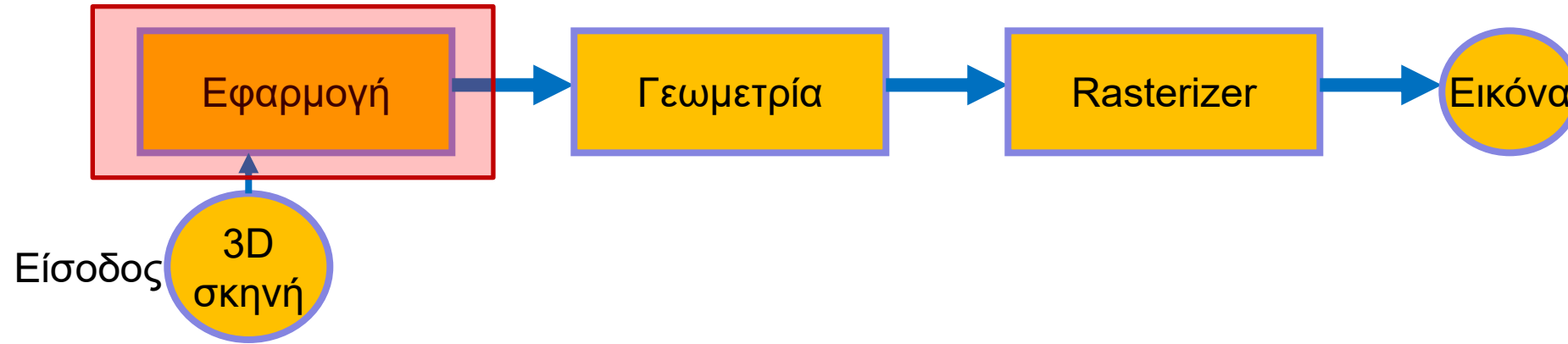
Απόδοση 3d Σκηνής

Βήματα απόδοσης:

- Περιγραφή εικονικής σκηνής (πως αναπαρίστανται οι 3d επιφάνειες)
- Τοποθέτηση εικονικής κάμερας (από που γίνεται η θέαση)
- Ορισμός πηγών φωτός (πως φωτίζονται τα αντικείμενα)
- Ορισμός οπτικών ιδιοτήτων επιφανειών (πως το φως αλληλοεπιδρά με τις επιφάνειες)
- Υπολογισμός χρώματος και φωτός κάθε pixel όπως φαίνεται από την κάμερα

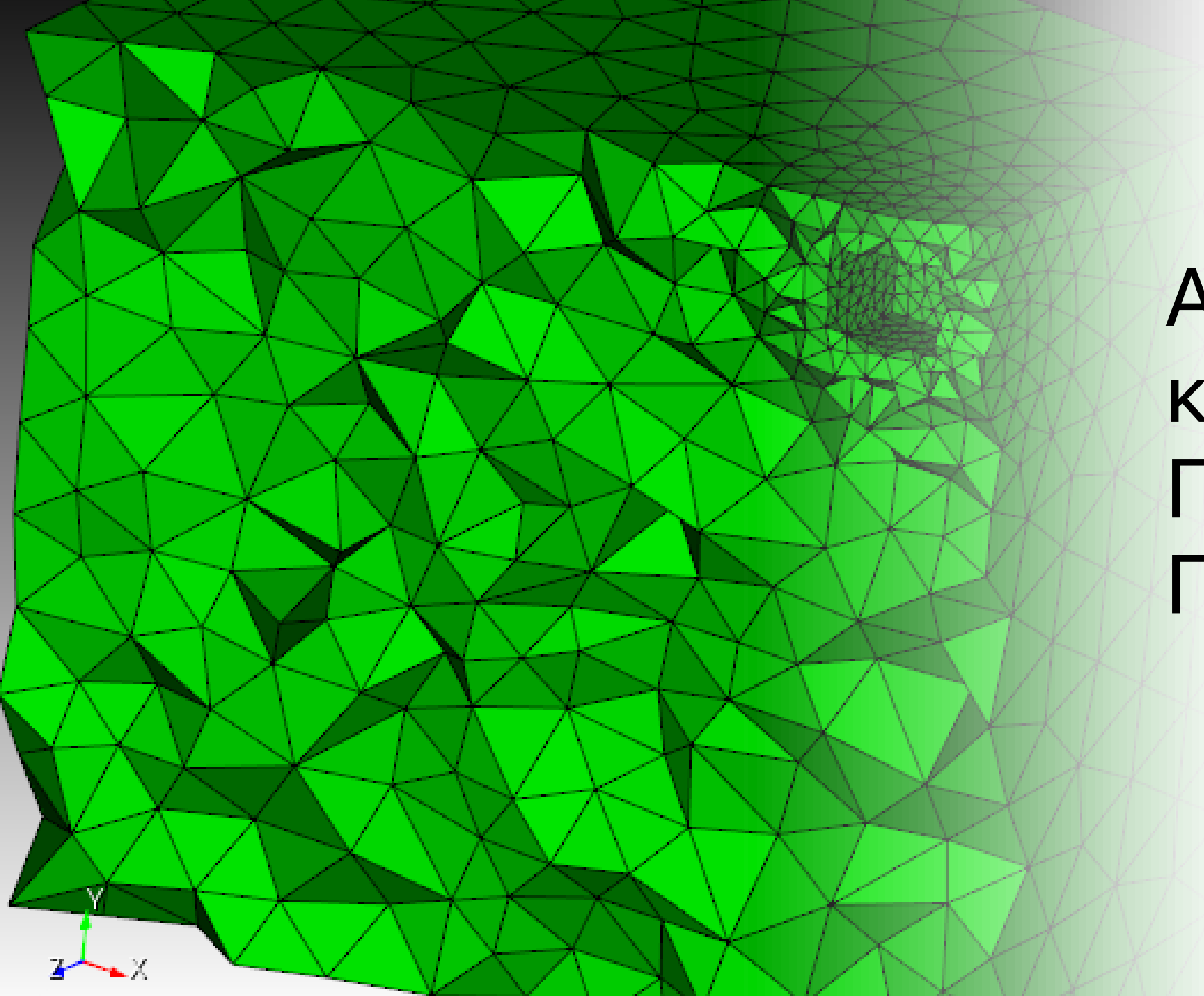
Διαφορετικές
τεχνικές απόδοσης
για την ίδια σκηνή





ΤΟ ΣΤΑΔΙΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Ορατότητα – Έλεγχος Παραμέτρων Απόδοσης – Αποστολή Γεωμετρίας στη GPU

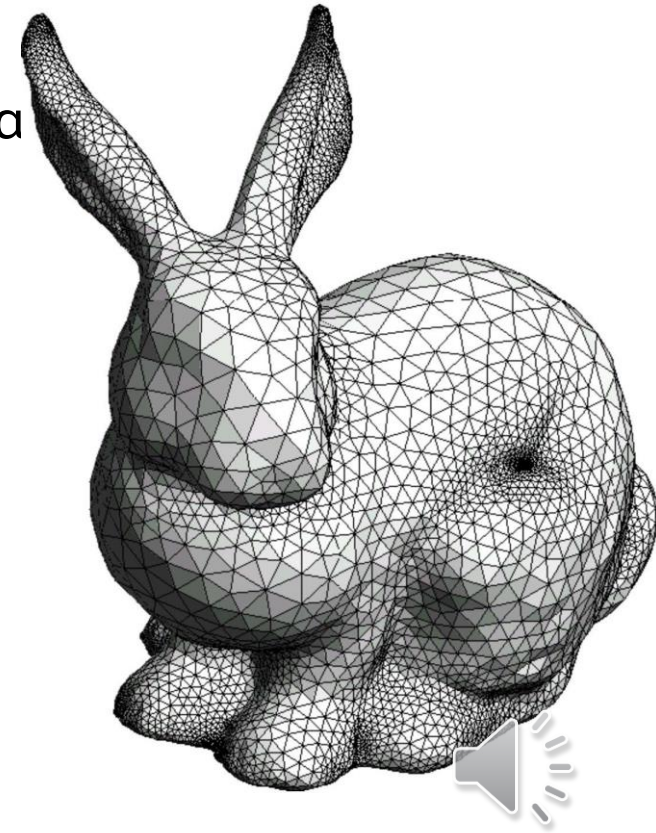


Αντικείμενα και Πολυγωνικά Πλέγματα

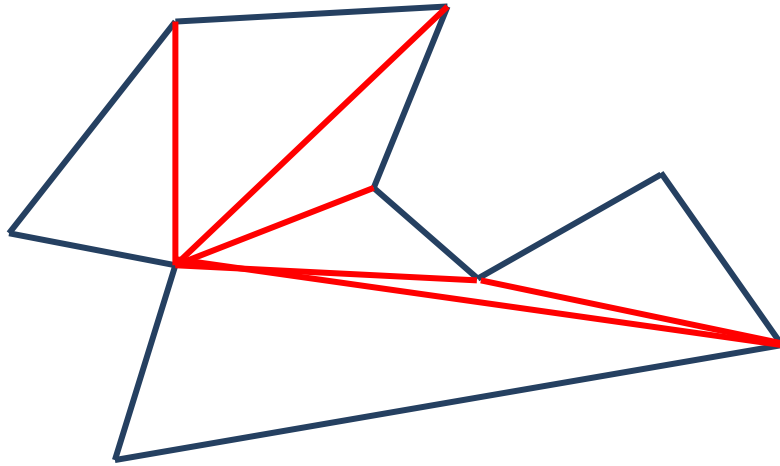
Πολυγωνικά Πλέγματα

- Οι επιφάνειες μοντελοποιούνται συνήθως χρησιμοποιώντας πολύγωνα (τετράπλευρα ή τρίγωνα)
- Οι σύγχρονες GPU έχουν σχεδιαστεί για να διαχειρίζονται και να αποδίδουν αποδοτικά τρίγωνα

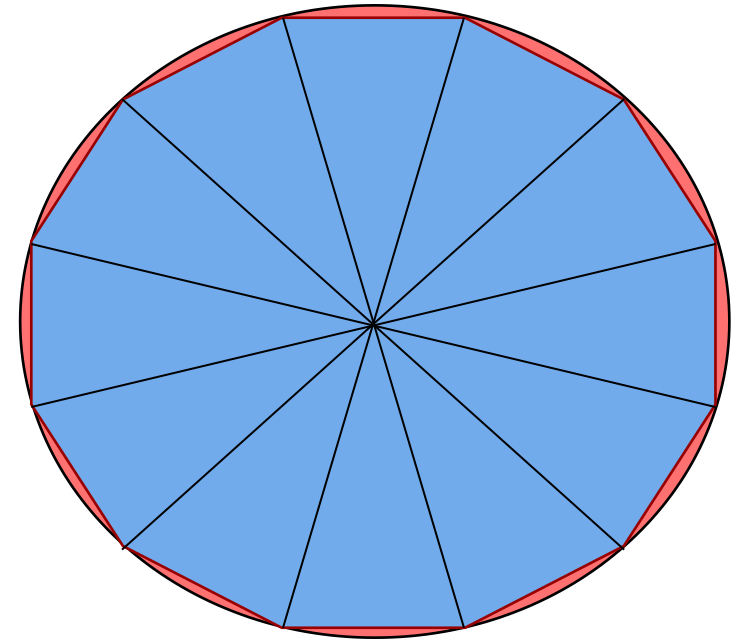
- + Το απλούστερο 2d σχήμα
- + Κάθε 2d πολύγωνο μπορεί να τριγωνοποιηθεί
- + Μπορεί εύκολα να αναπαραστήσει εξαιρετικά λεπτά χαρακτηριστικά επιφανειών
- Υψηλό κόστος σε μνήμη και επεξεργασία
- Δυσκολία στην αναπαράσταση καμπυλωτών επιφανειών (προσέγγιση)



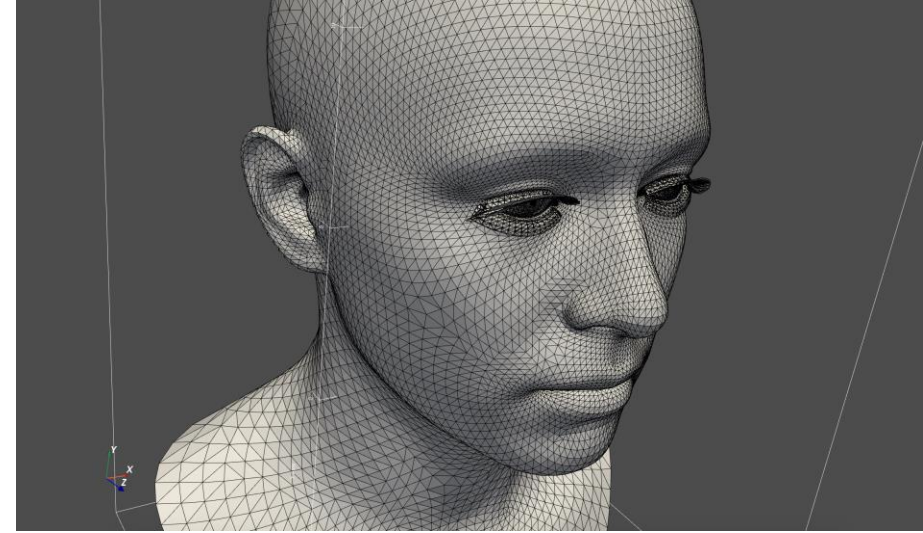
Τριγωνοποίηση Πολυγώνων



Κύκλων;;;



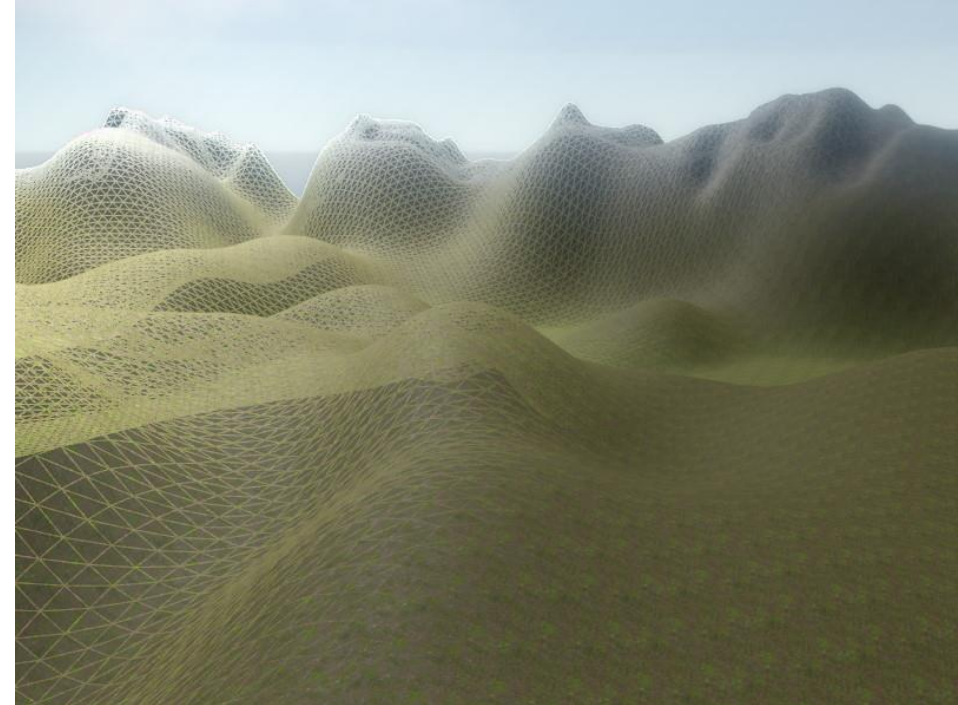
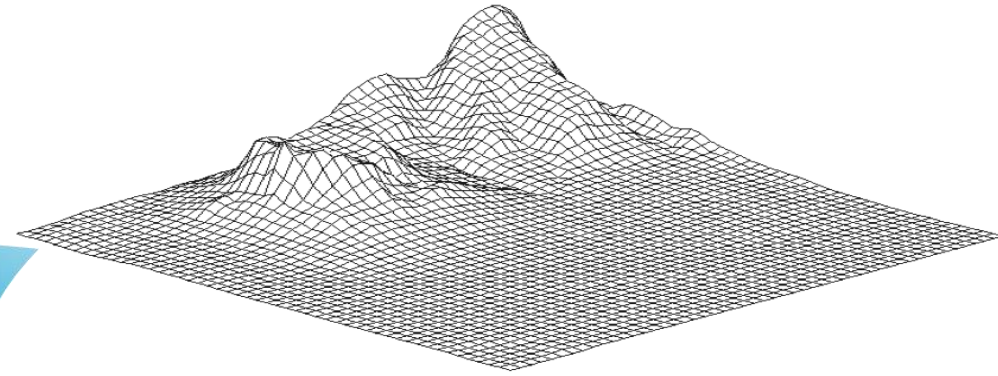
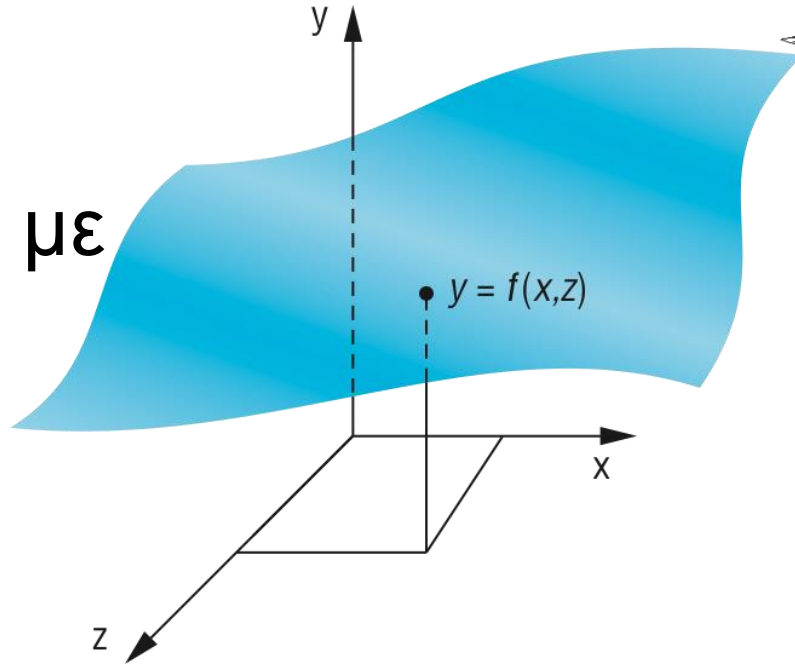
Τριγωνικά Πλέγματα



- Ένα πλέγμα είναι ένα δίκτυο τριγώνων.
- Συνδέονται μεταξύ τους μέσω κοινών κορυφών και άκρων για να σχηματίσουν μια ενιαία συνεχή επιφάνεια.
- Ένα πλέγμα είναι διαχειρίσιμο πιο αποτελεσματικά από μια συλλογή από τον ίδιο αριθμό άσχετων τριγώνων.
- Οι ελάχιστες πληροφορίες που απαιτούνται για ένα τριγωνικό πλέγμα είναι:
 - Ένα σύνολο τριγώνων (τριπλές κορυφές)
 - Οι θέσεις (σε 3d) των κορυφών τους
- Πολλές μορφές αρχείων/δομές δεδομένων αποθηκεύουν πρόσθετα δεδομένα στις κορυφές, τις ακμές ή τις όψεις
 - Αυτά τα δεδομένα υποστηρίζουν χαρτογράφηση υφής, σκίαση, κινούμενη εικόνα και άλλες λειτουργίες.

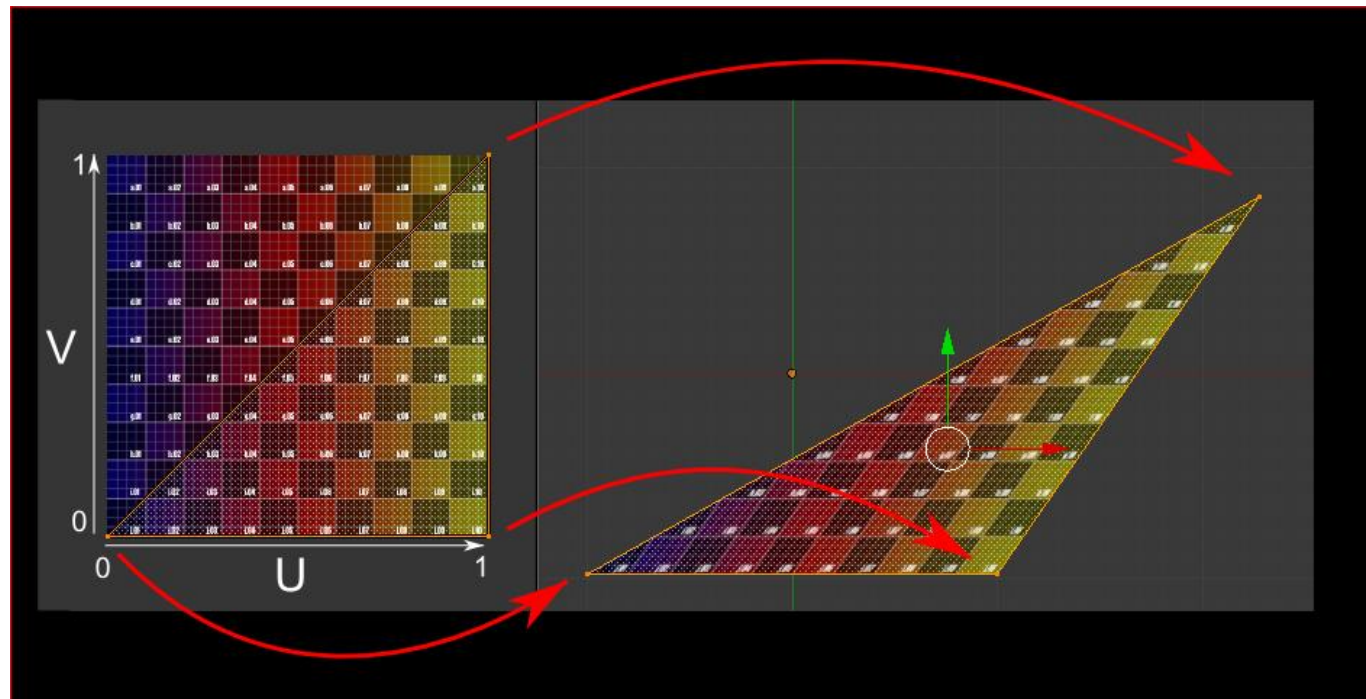
Παράδειγμα: Πεδία Ύψους

- Για κάθε (x, z) υπάρχει μοναδικό y
- Πίνακα από y τιμές με δειγματοληψία
- Απεικονίζονται ως τετράπλευρα ή τρίγωνα



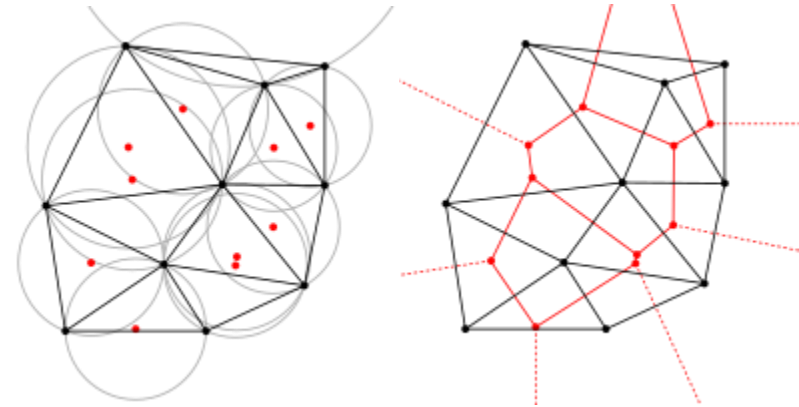
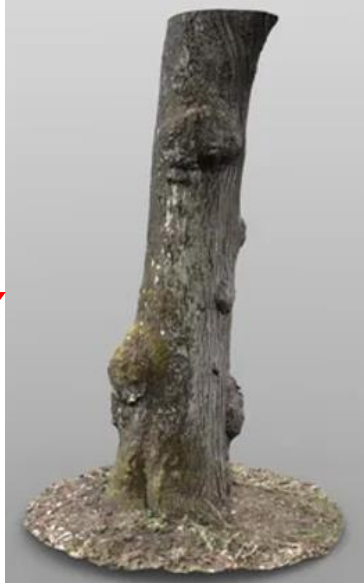
(u, v) -συντεταγμένες για Χαρτογράφηση

Κάθε κορυφή έχει (u, v) -συντεταγμένες που αντιστοιχούν σε μία εικόνα με την οποία θα ντυθούν τα αντίστοιχα τρίγωνα (χαρτογράφηση – texture mapping)

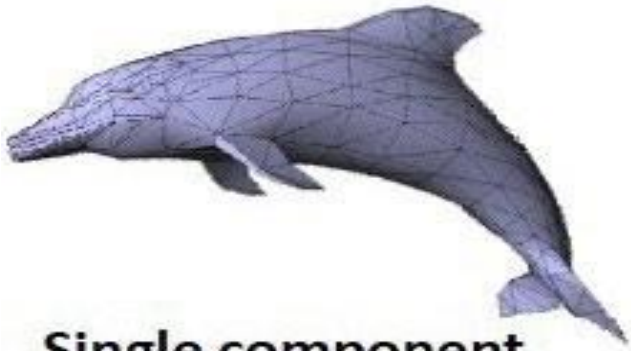


Θέματα που Αφορούν τα Πλέγματα

- Πώς κατασκευάζουμε ένα πλέγμα από ένα σύννεφο σημείων;
- Υπάρχουν τεχνολογίες που σαρώνουν αντικείμενα δημιουργώντας εκατομμύρια τέτοια σημεία (Laser συσκευές – εφαρμογές όπως το [RealityScan](#))
- Delaunay τριγωνοποίηση
- Απλοποίηση πλέγματος



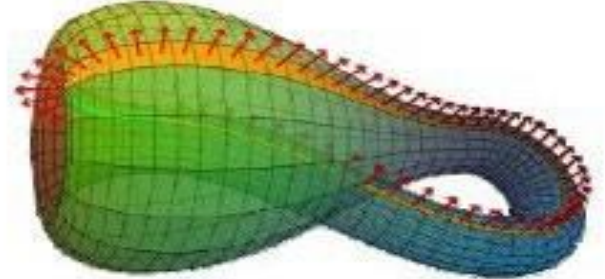
Πλέγματα Επιφανειών



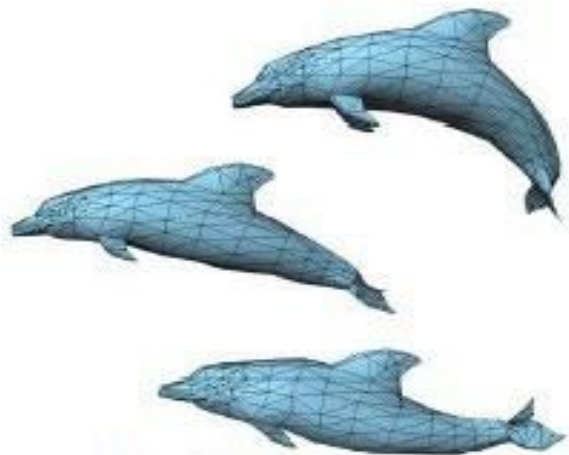
**Single component,
closed, triangular,
orientable manifold**



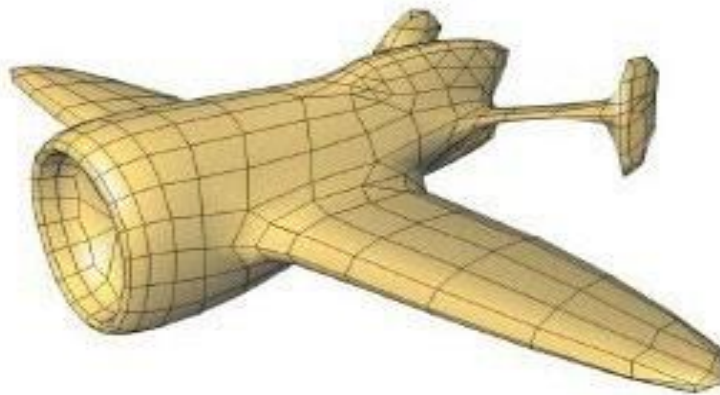
With boundaries



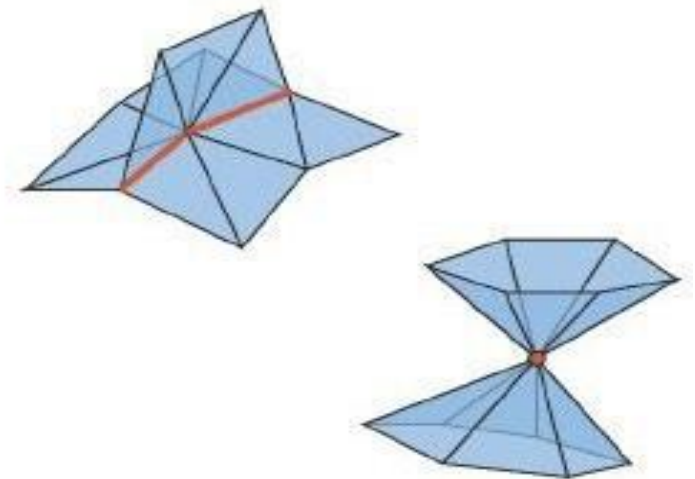
Not orientable



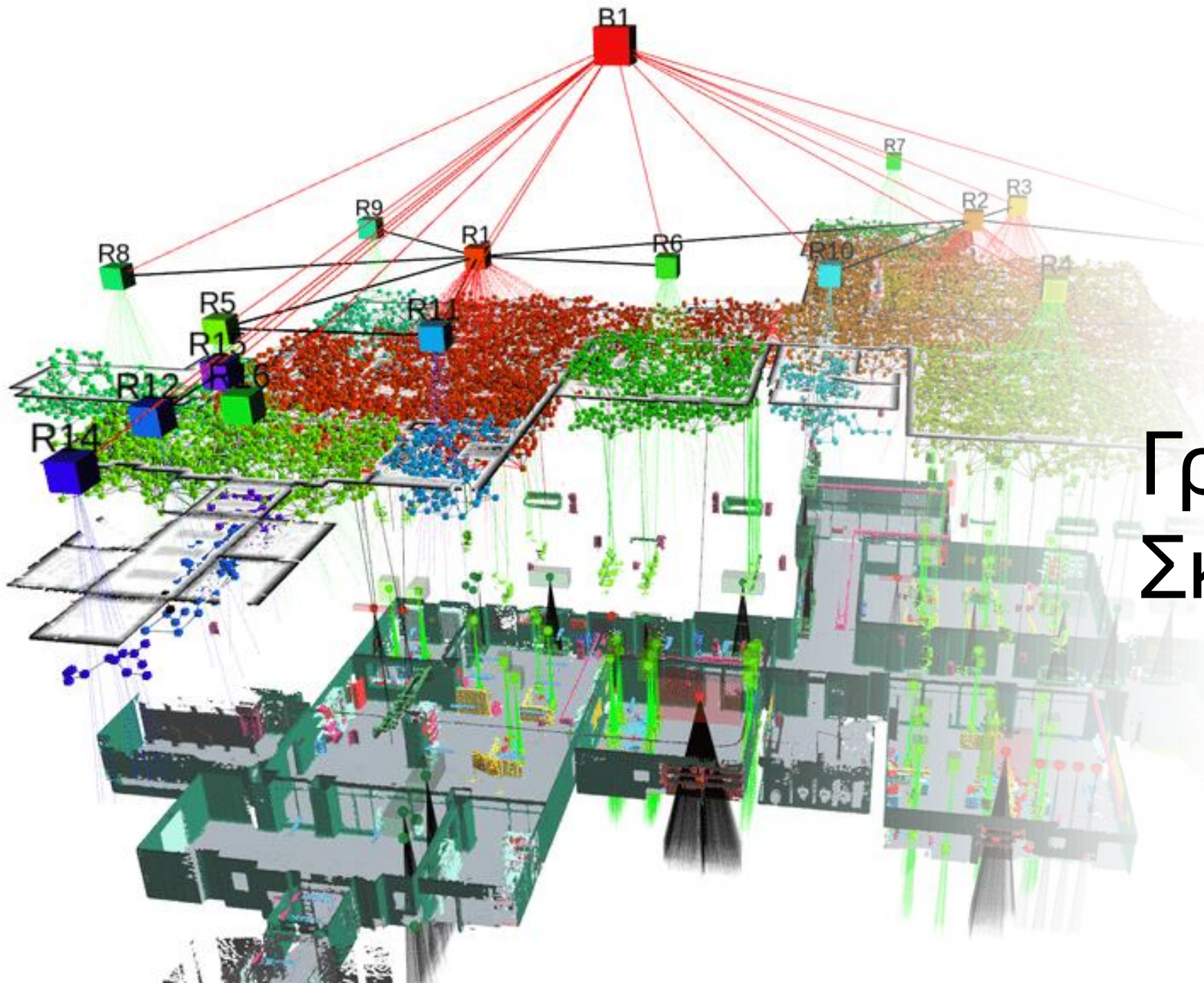
Multiple components



Not only triangles



Non manifold



Γράφημα
Σκηνής

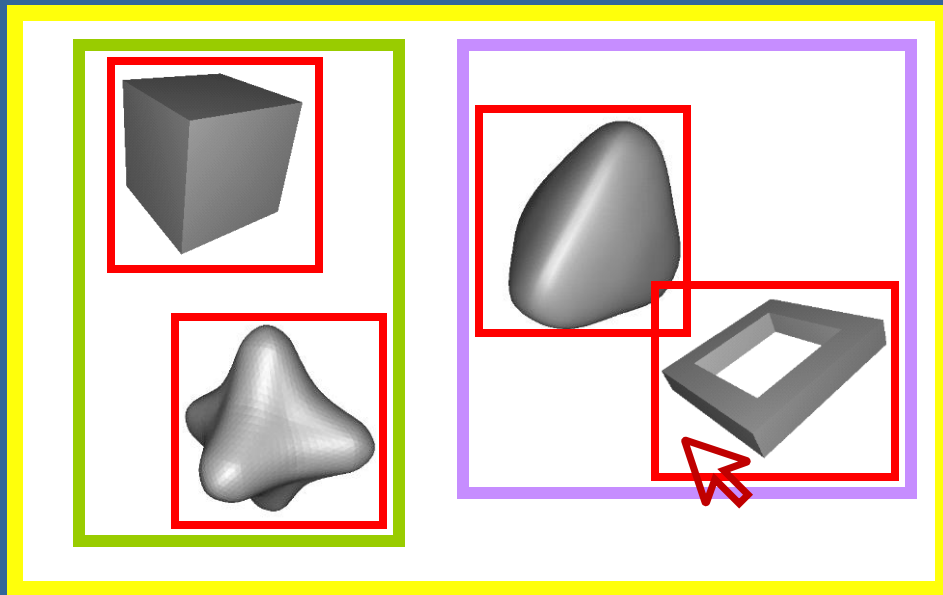
Χωρικές Δομές Δεδομένων

- Οργανώνουν τη γεωμετρία σε 2d ή 3d (ή παραπάνω)
- Ο στόχος είναι η ταχύτερη επεξεργασία
 - Ταχύτερη πραγματικού χρόνου απόδοση
 - Ταχύτερος έλεγχος τομής
 - Ταχύτερη ανίχνευση σύγκρουσης
 - Ταχύτερο ray tracing και καθολικός φωτισμός
- Χρησιμοποιούνται εκτενώς στα παιχνίδια για τα αντικείμενα της σκηνής αλλά και σε ταινίες (computer–animated)

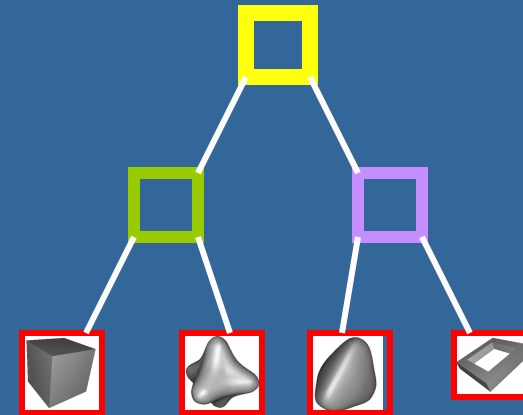
Γιατί;

- Οργανώνει τη γεωμετρία με ιεραρχικό τρόπο

Σε 2d



Δομή Δεδομένων



Πού έκανα κλικ;

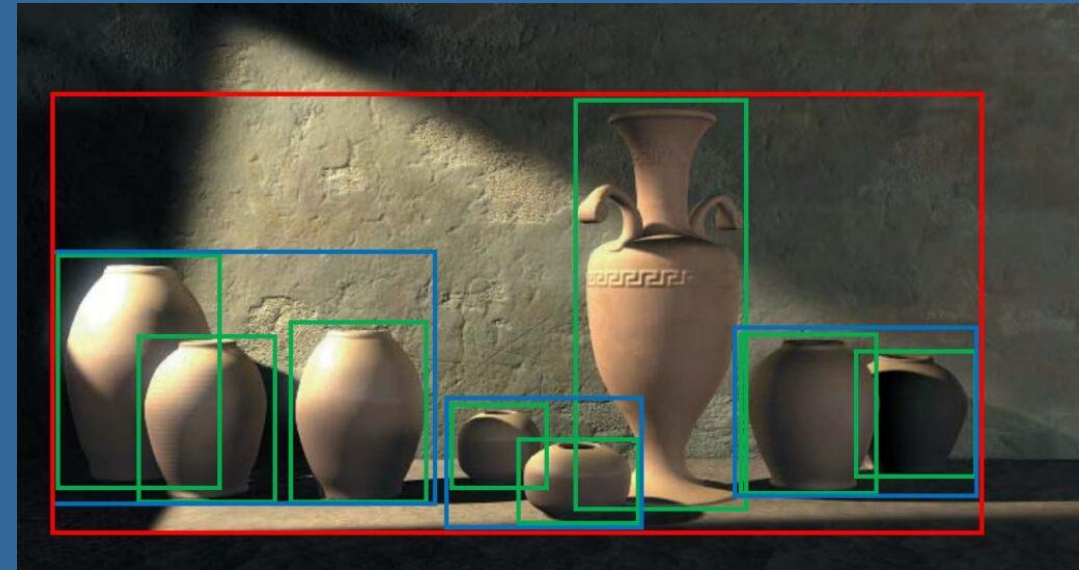
- 1) Έλεγχξε τη ρίζα
- 2) Διαπέρασε το δένδρο αναδρομικά
- 3) Τερμάτισε τη διαπέραση όταν βρεις τη θέση

Αναμένουμε $O(\log n)$ αντί για $O(n)$

Ιεραρχία Οριοθετημένου Όγκου

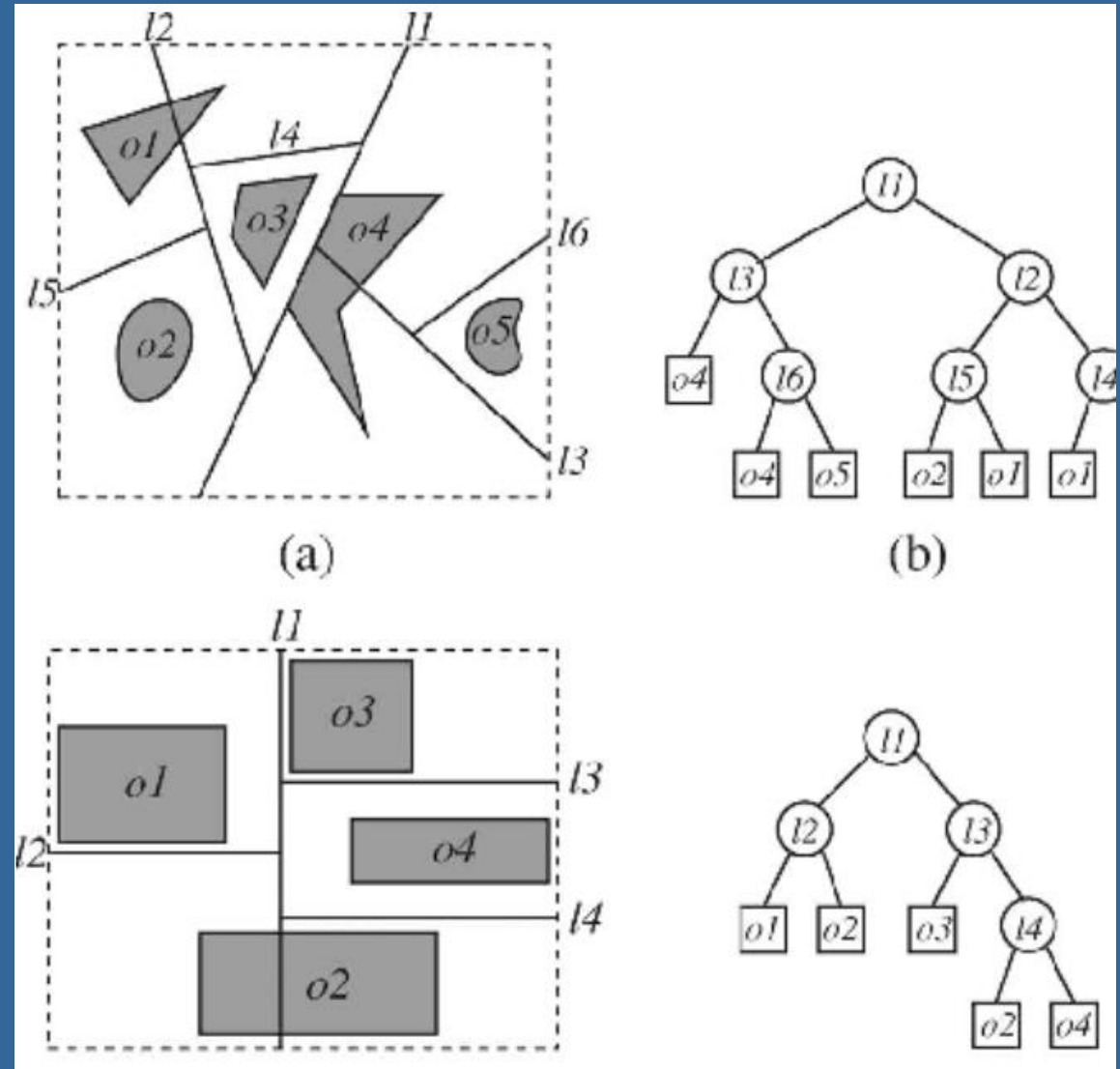
Bounding Volume Hierarchy – BVH

- Οι συνήθεις τύποι οριοθετημένων όγκων (BVs):
 - Σφαίρα
 - Κουτιά (AABB και OBB)
- Το BV δεν συνεισφέρει στην αποδιδόμενη εικόνα – απλά εσωκλείει ένα αντικείμενο
- Ένα k -αδικό δένδρο
 - Τα φύλλα έχουν τη γεωμετρία
 - Οι εσωτερικοί κόμβοι έχουν το πολύ k παιδιά
 - Οι εσωτερικοί κόμβοι έχουν BVs που εσωκλείουν όλη τη γεωμετρία στο υποδένδρο τους.



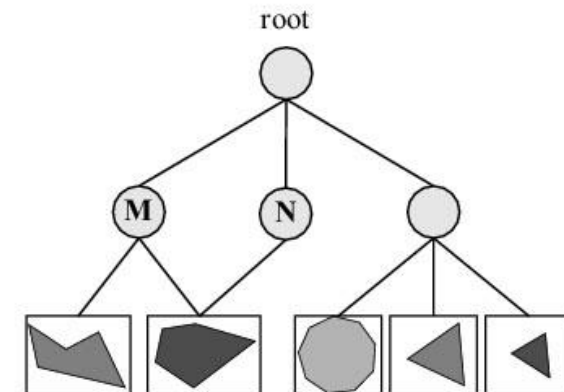
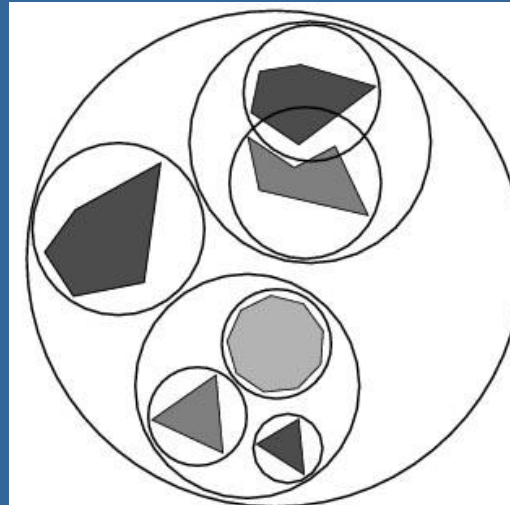
BSP Δένδρα (Binary Space Partitioning Trees)

- Ιδέα:
 - Διαμέριση του χώρου με ένα επίπεδο
 - Τοποθέτηση της γεωμετρίας στο υποχώρο που ανήκει
 - Αναδρομικά κάνε το ίδιο στους υποχώρους
- Δύο τύποι:
 - Ευθυγραμμισμένο με άξονα
 - Ευθυγραμμισμένο με το αντικείμενο

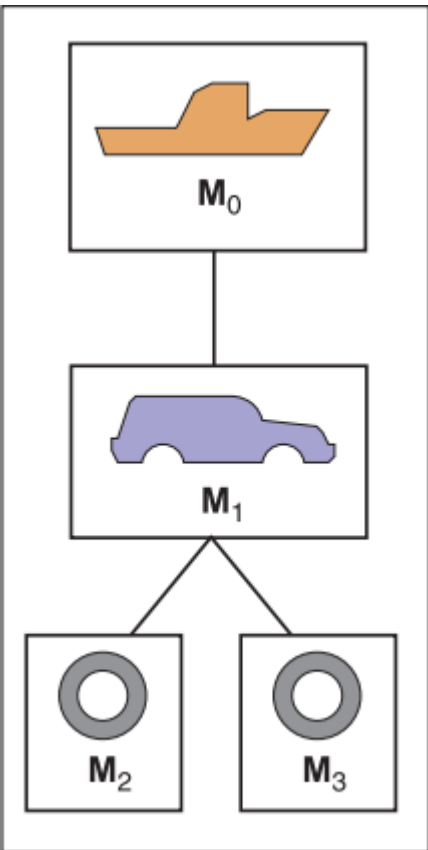


Γράφημα Σκηνής

- Η δομή BVH είναι η πλέον χρησιμοποιούμενη
 - Εύκολη στην κατανόηση
 - Εύκολη στην υλοποίηση
- Αποθηκεύει όμως μόνο γεωμετρία
 - Η απόδοση γραφικών δεν αφορά μόνο τη γεωμετρία
- Το γράφημα σκηνής είναι ένα εμπλουτισμένο BVH:
 - Φώτα
 - Υφές
 - Μετασχηματισμούς
 - Και άλλα...



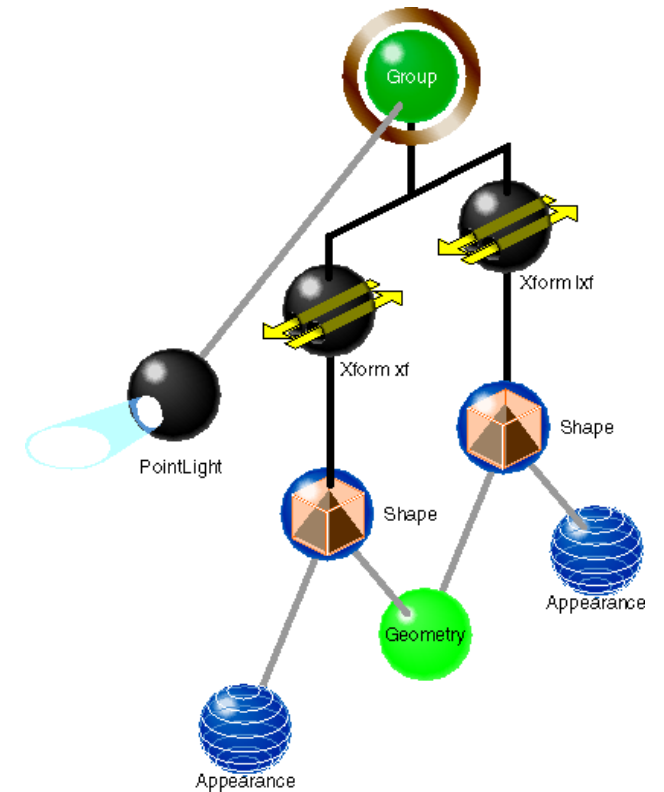
Γράφημα Σκηνής



- Το πλοίο μετασχηματίζεται από το μητρώο M_0
- Το σώμα του οχήματος μετασχηματίζεται από το μητρώο M_0M_1
- Η αριστερή ρόδα του οχήματος μετασχηματίζεται από το μητρώο $M_0M_1M_2$
- Η δεξιά ρόδα του οχήματος μετασχηματίζεται από το μητρώο $M_0M_1M_3$

Εκτός από γεωμετρική πληροφορία το γράφημα σκηνής μπορεί να έχει:

- Φώτα
 - Θέση
 - Χρώμα
 - Τύπος (π.χ., προβολέας ή ήλιος)
- Υλικά για αντικείμενα





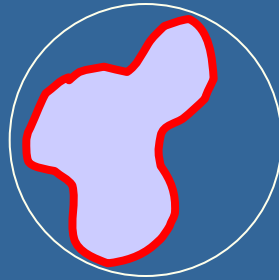
Αποκοπή Γεωμετρικών Αντικειμένων

*Δεν θέλουμε να επεξεργαζόμαστε δεδομένα που δεν θα
συμβάλουν στην τελική εικόνα.*

Τύποι Τεχνικών Αποκοπής (culling techniques)

Τα κόκκινα αντικείμενα δεν αποδίδονται.

view frustum



■ detail

backface



portal

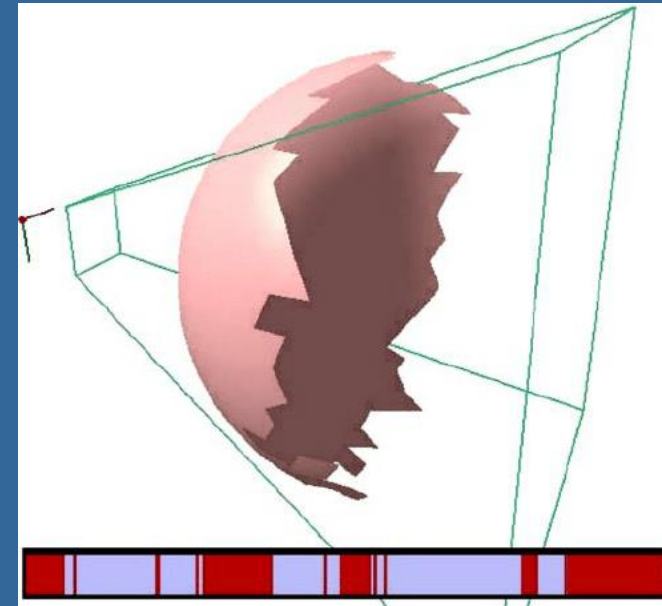


occlusion



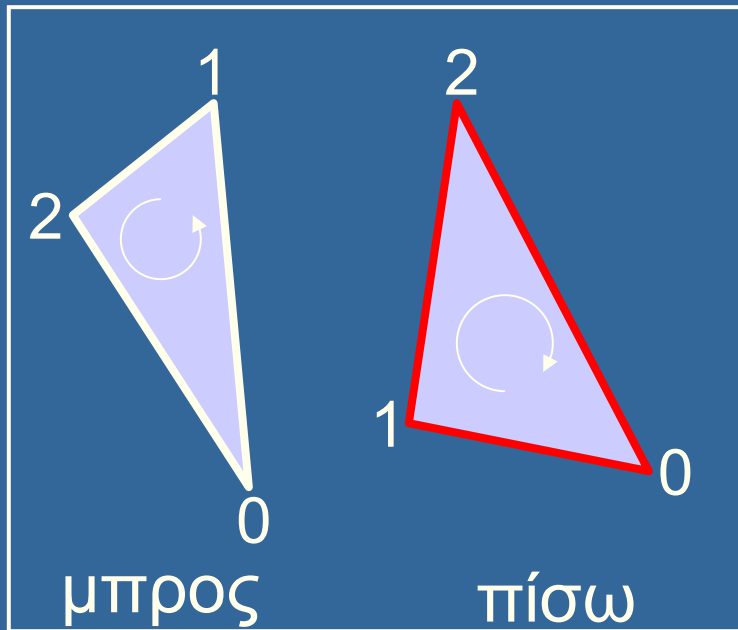
Αποκοπή Πίσω Όψης (backface culling – GPU)

- Απλή τεχνική για απόρριψη πολυγώνων που είναι προσανατολισμένα σε αντίθετη κατεύθυνση από αυτή του προσανατολισμού του θεατή
- Μπορεί να χρησιμοποιηθεί για:
 - Κλειστή επιφάνεια (π.χ., σφαίρα)
 - ή όταν ξέρουμε ότι οι πίσω όψεις δεν θα έπρεπε να φαίνονται ποτέ (π.χ., τοίχοι σε ένα δωμάτιο)
- Δύο προσεγγίσεις (συνήθως στο στάδιο της Γεωμετρίας ή του Rasterization):
 - Χώρος κάμερας
 - Χώρος οθόνης

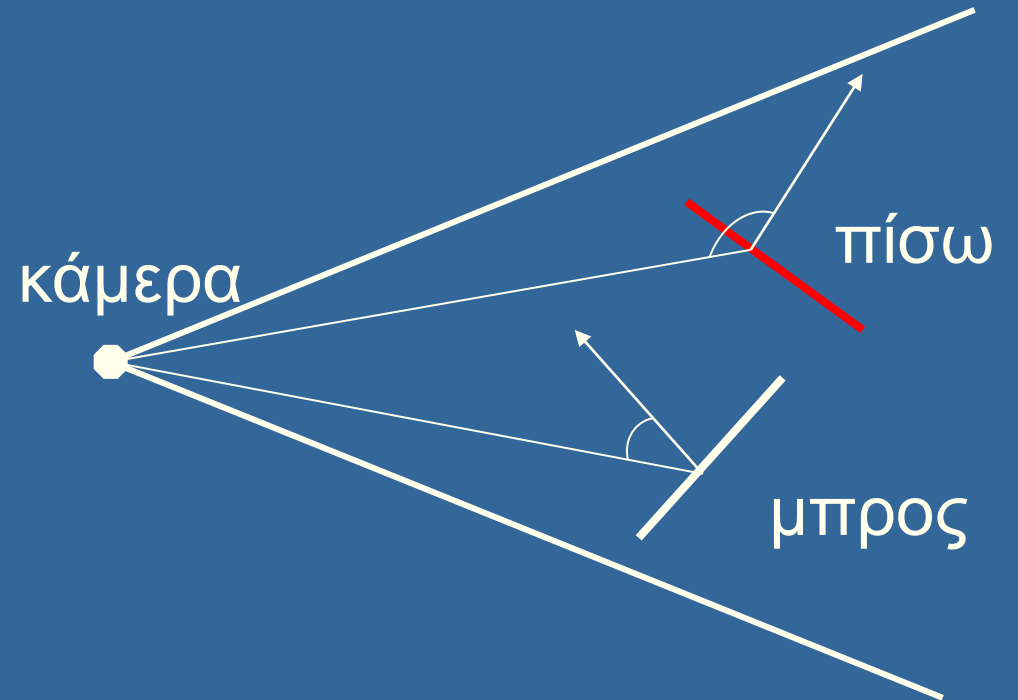


Πώς γίνεται;

- Δύο τρόποι ανάλογα με τον χώρο:



Χώρος οθόνης
(συνεπής κατεύθυνσης διαπέρασης)

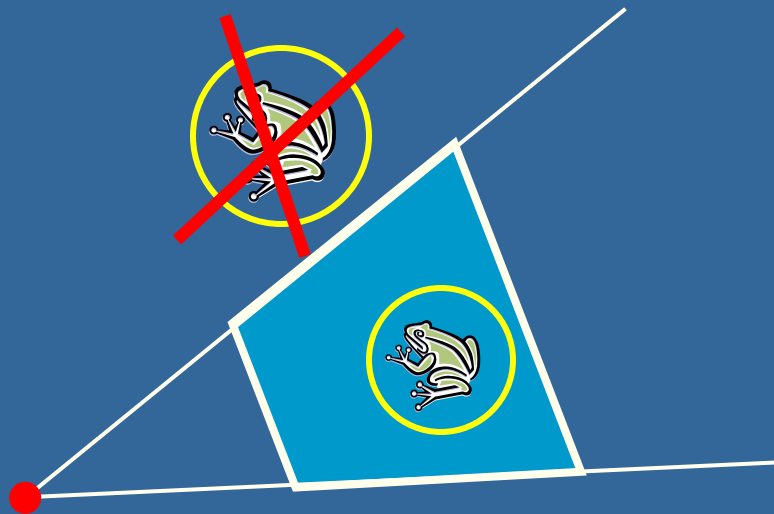


Χώρος κάμερας
(πάλι το εσωτερικό γινόμενο)

Αποκοπή Πεδίου Θέασης – CPU

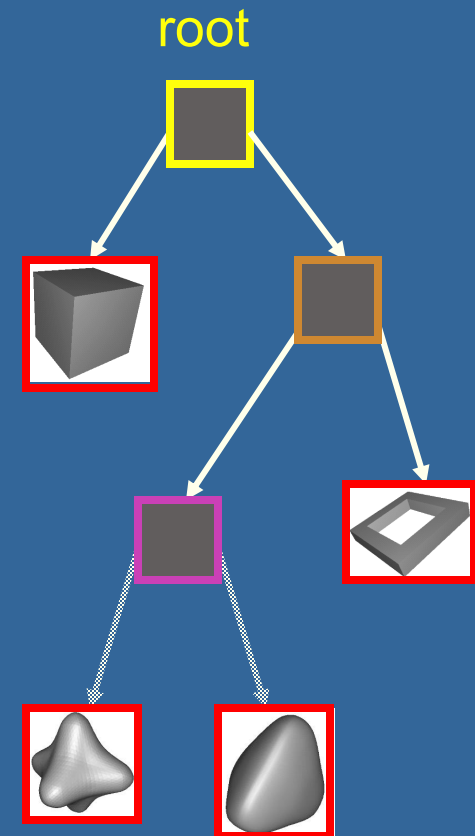
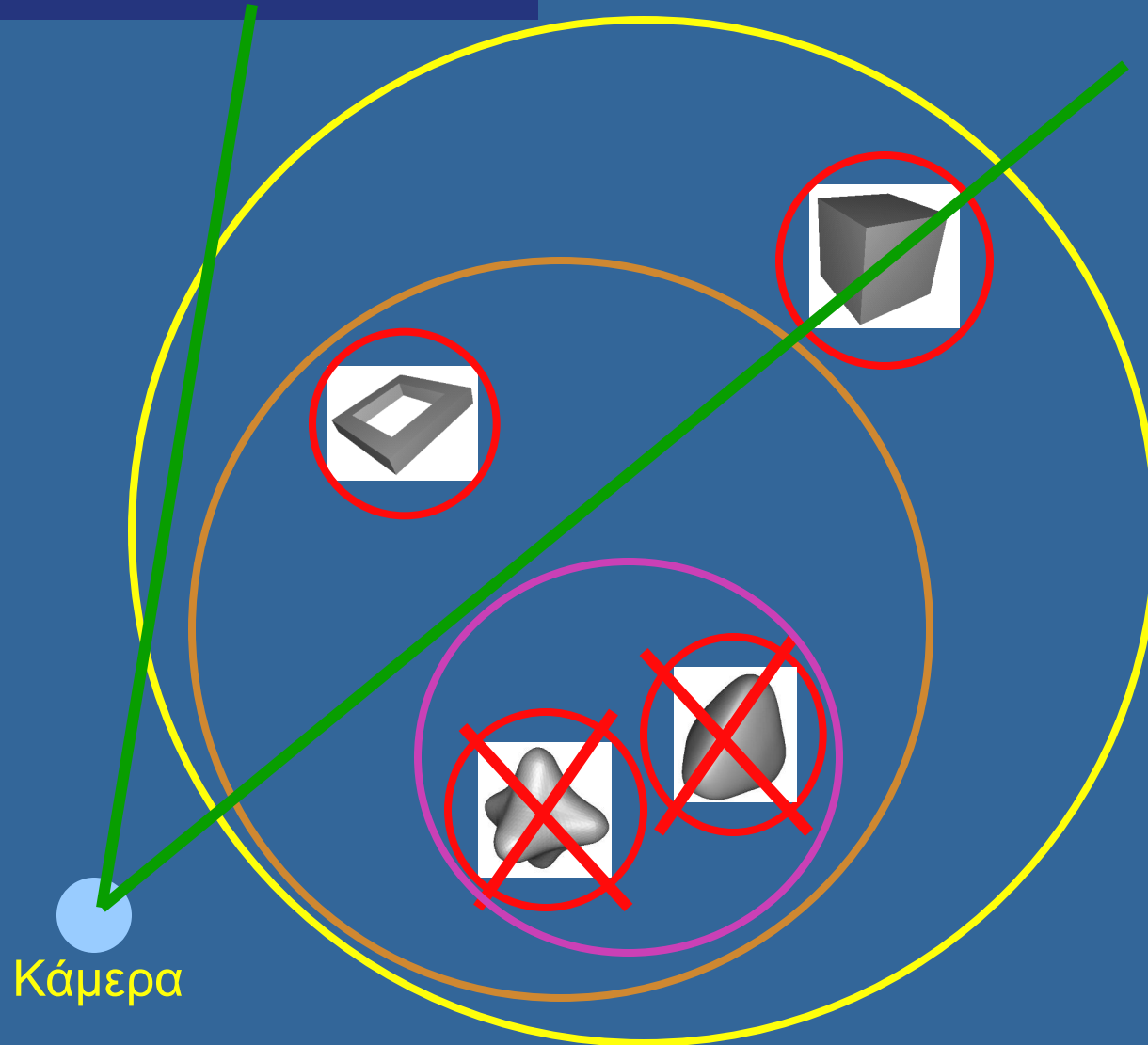
(View-Frustum Culling)

- Οριοθέτηση ενός «φυσικού» συνόλου στοιχείων με έναν απλό όγκο (π.χ., σφαίρα, κουτί)
- Αν το BV είναι εκτός του πεδίου θέασης, τότε όλα τα στοιχεία του BV δεν είναι ορατά



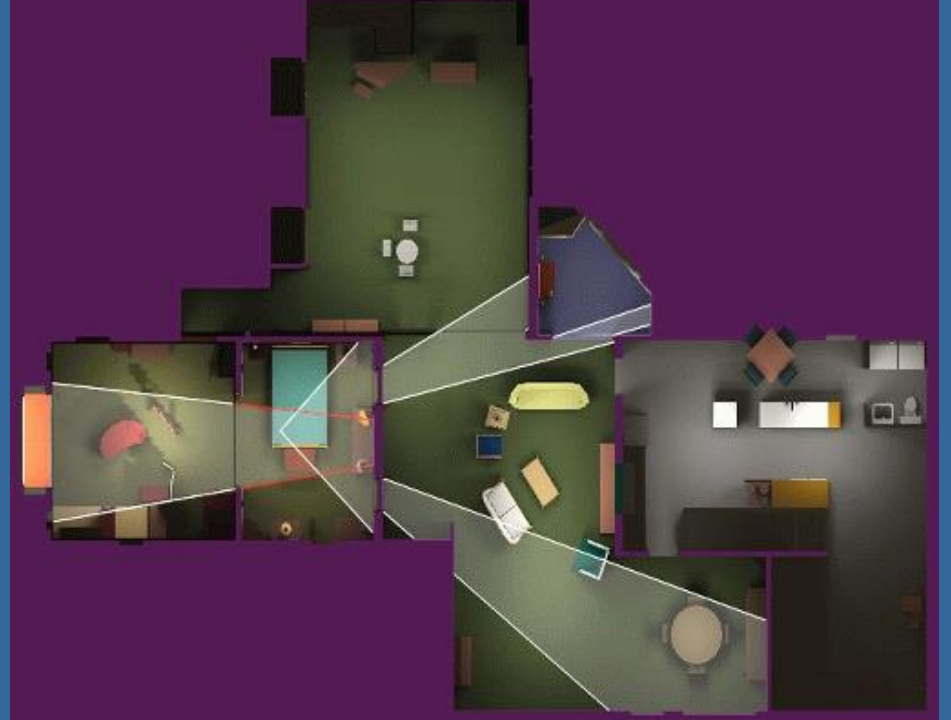
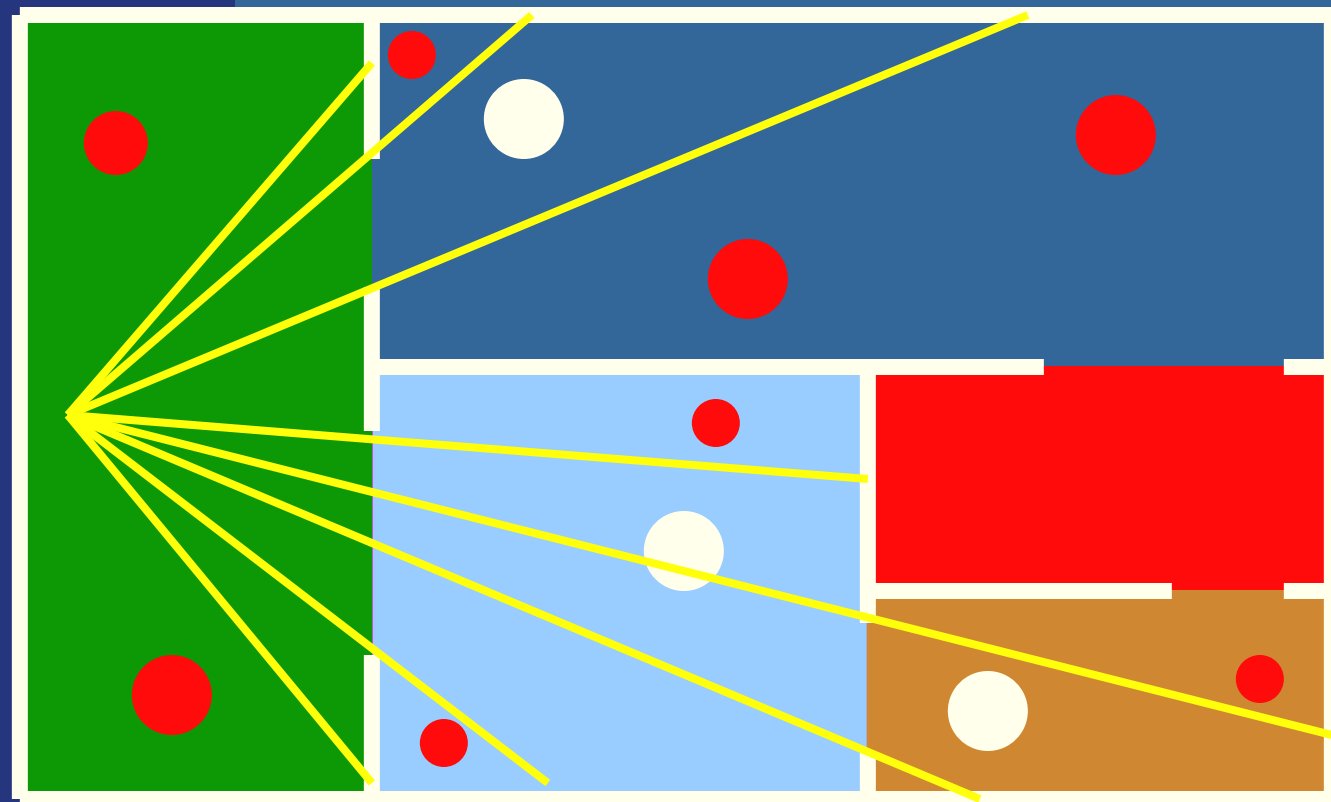
Χρήση BVH, BSP, ...

Παράδειγμα με BVH

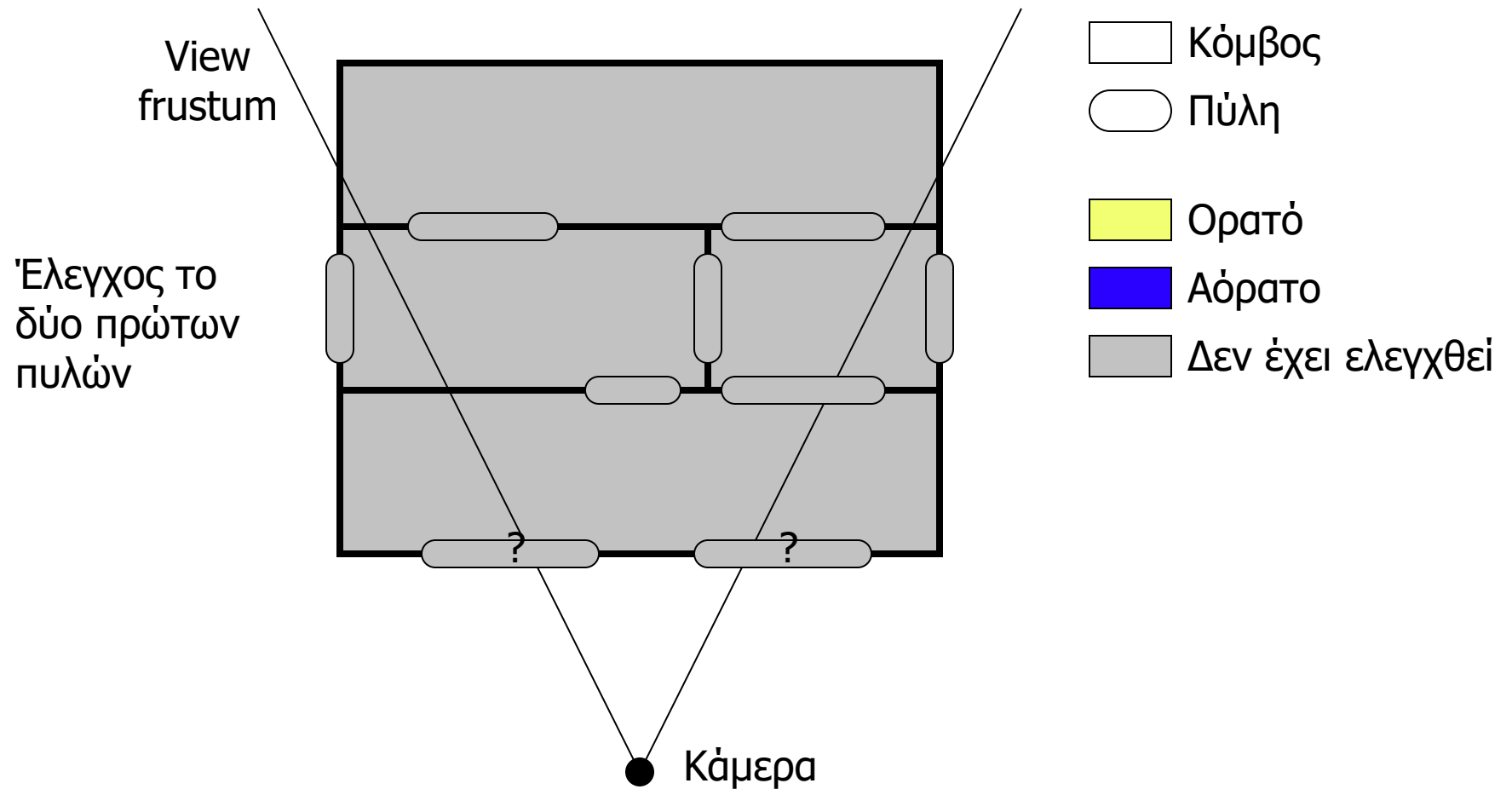


Αποκοπή Πύλης (Portal Culling – CPU

- Σε ένα κτίριο από πάνω
- Οι κύκλοι είναι αντικείμενα που μπορεί να αποδοθούν γραφικά

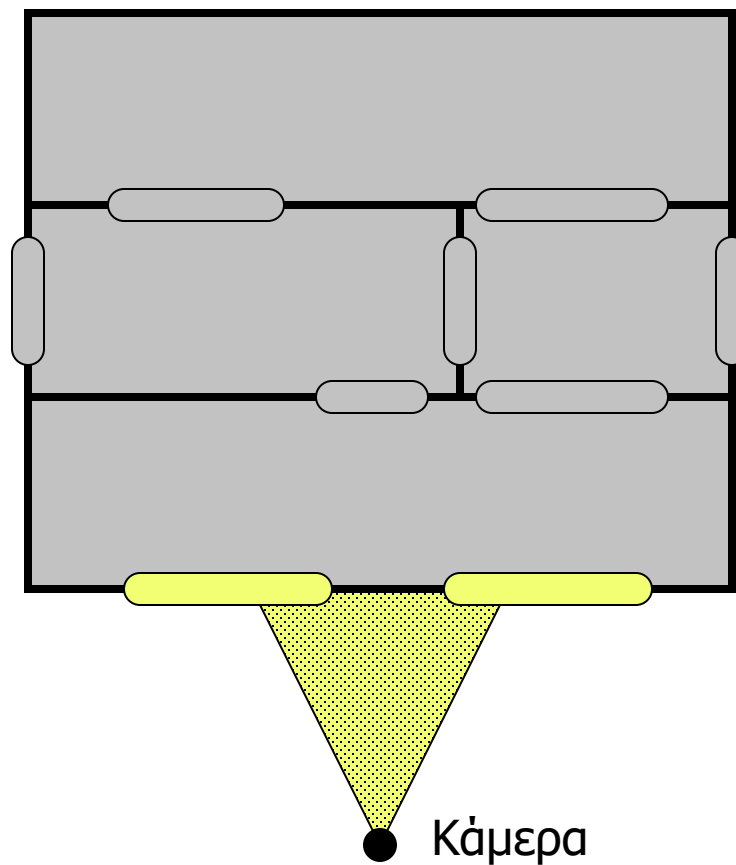


Διαμερισμός Όγκων – Πύλες



Διαμερισμός Όγκων – Πύλες

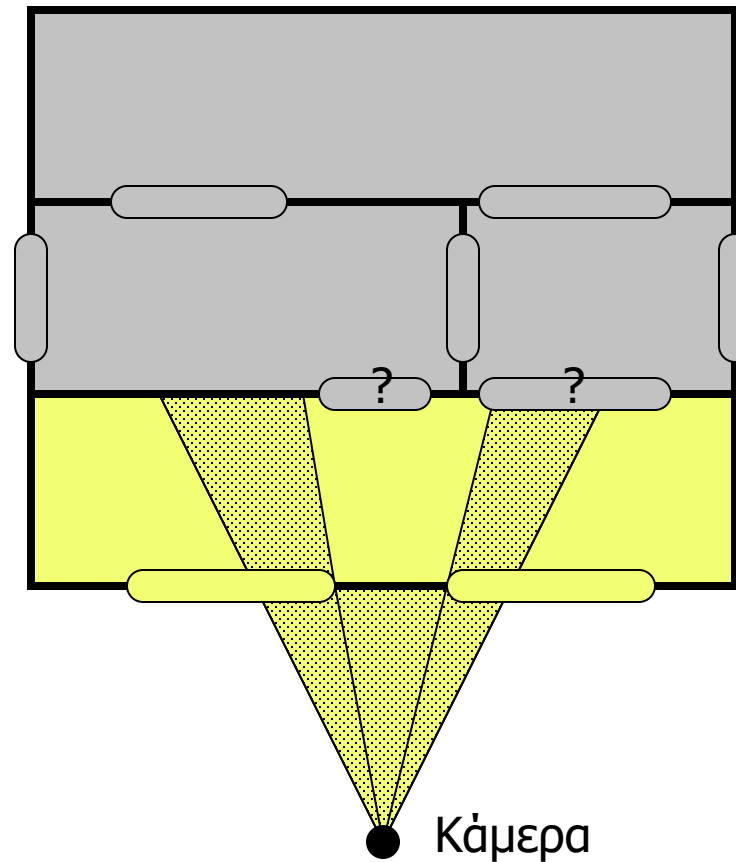
Και οι δύο κόμβοι
είναι ορατοί



- Κόμβος
- Πύλη
- Ορατό
- Αόρατο
- Δεν έχει ελεγχθεί

Διαμερισμός Όγκων – Πύλες

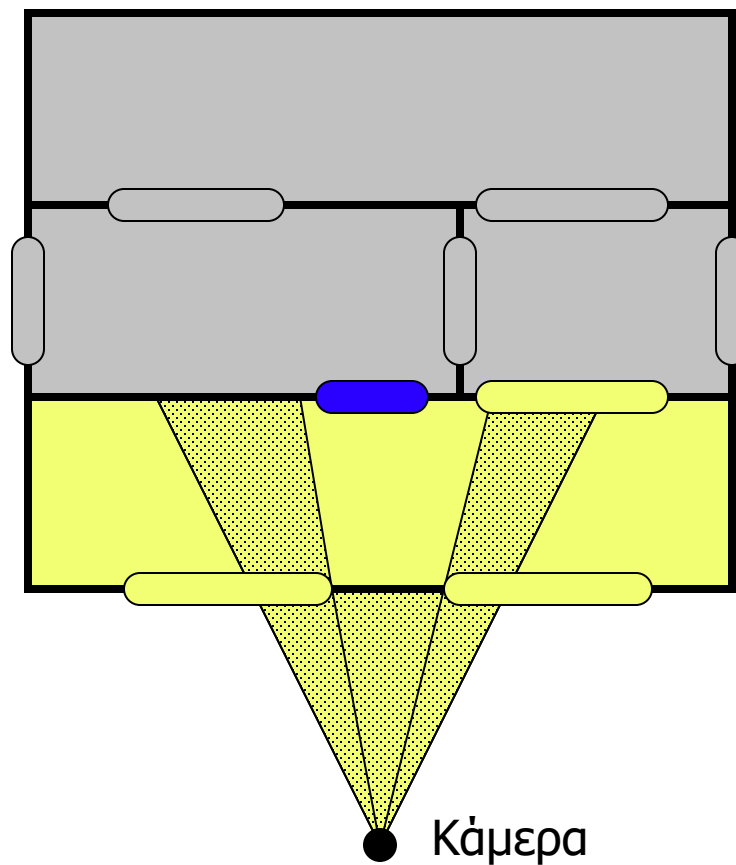
Μαρκάρισμα κόμβου ως
ορατός, έλεγχος όλων
των πυλών από αυτόν
τον κόμβο



- Κόμβος
- Πύλη
- Ορατό
- Αόρατο
- Δεν έχει ελεγχθεί

Διαμερισμός Όγκων – Πύλες

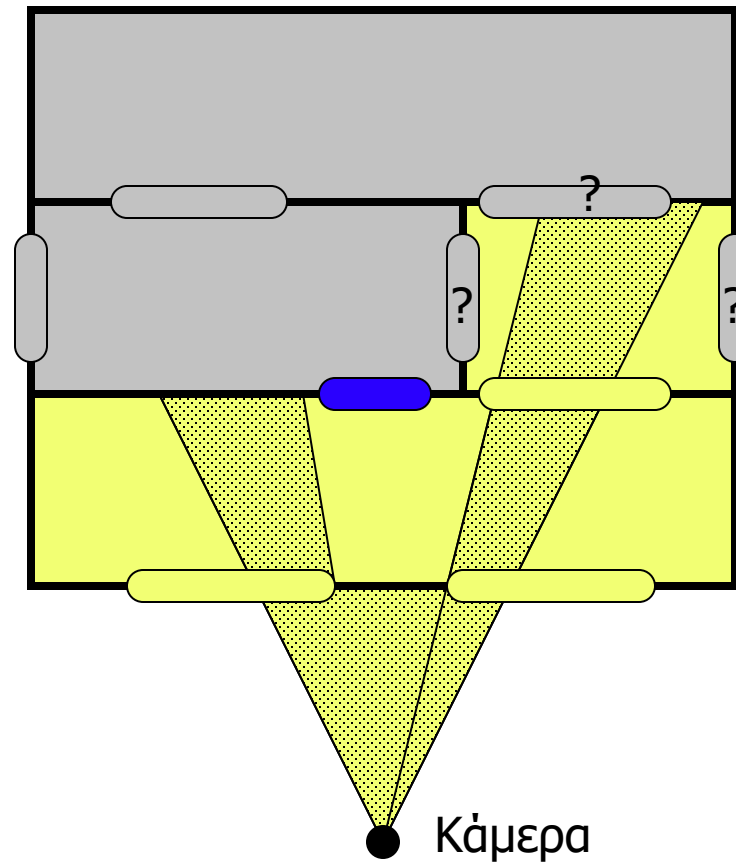
Μία πύλη ορατή
και μία αόρατη



- Κόμβος
- Πύλη
- Ορατό
- Αόρατο
- Δεν έχει ελεγχθεί

Διαμερισμός Όγκων – Πύλες

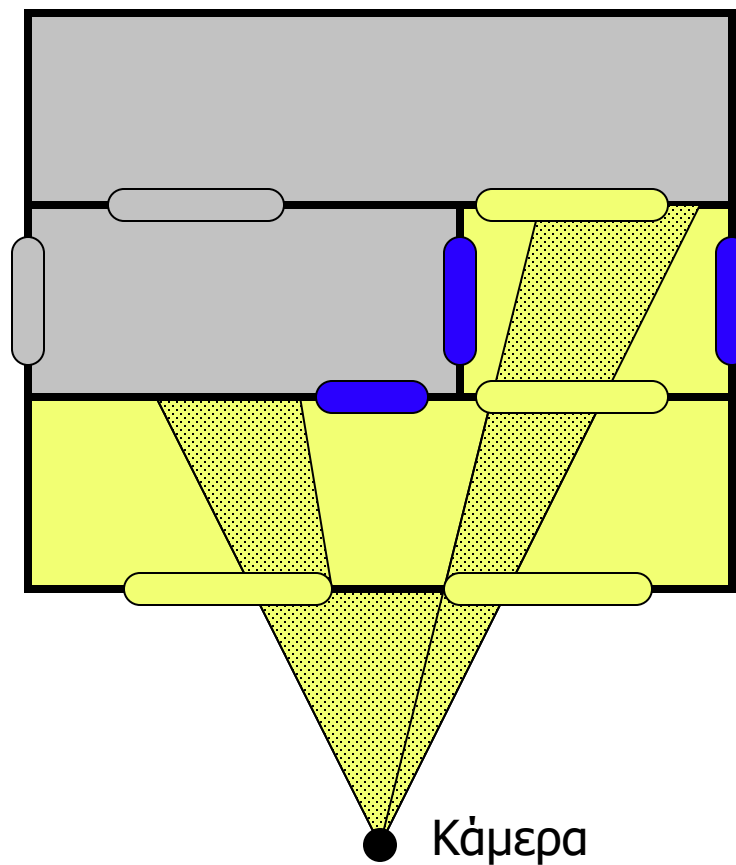
Μαρκάρισμα κόμβου ως ορατός. Δεν έγινε επίσκεψη στον άλλο κόμβο. Έλεγχος όλων των πυλών στον ορατό κόμβο



- Κόμβος
- Πύλη
- Ορατό
- Αόρατο
- Δεν έχει ελεγχθεί

Διαμερισμός Όγκων – Πύλες

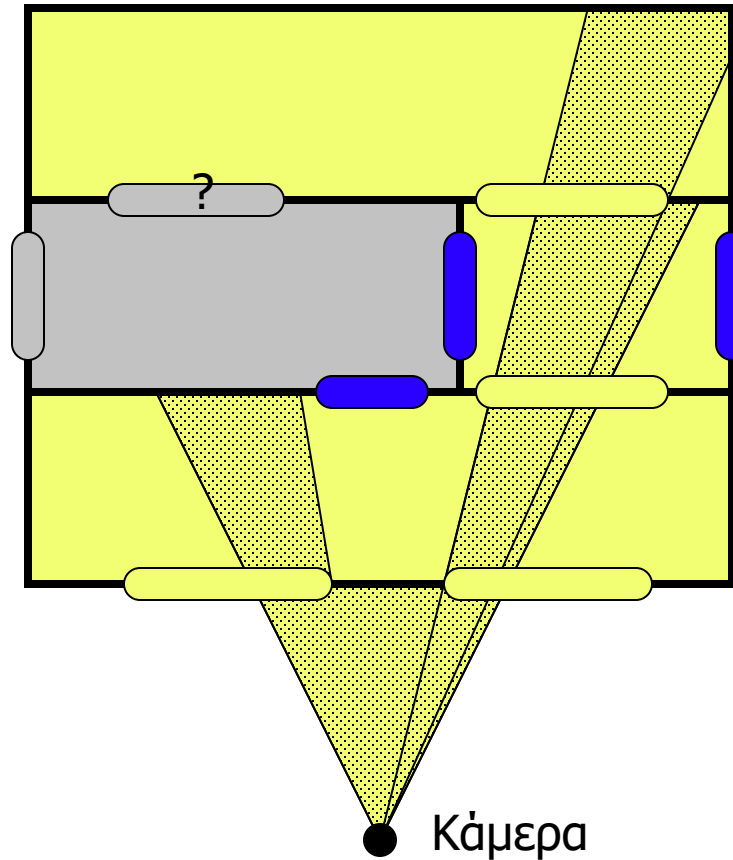
Μία ορατή και
δύο αόρατες



- Κόμβος
- Πύλη
- Ορατό
- Αόρατο
- Δεν έχει ελεγχθεί

Διαμερισμός Όγκων – Πύλες

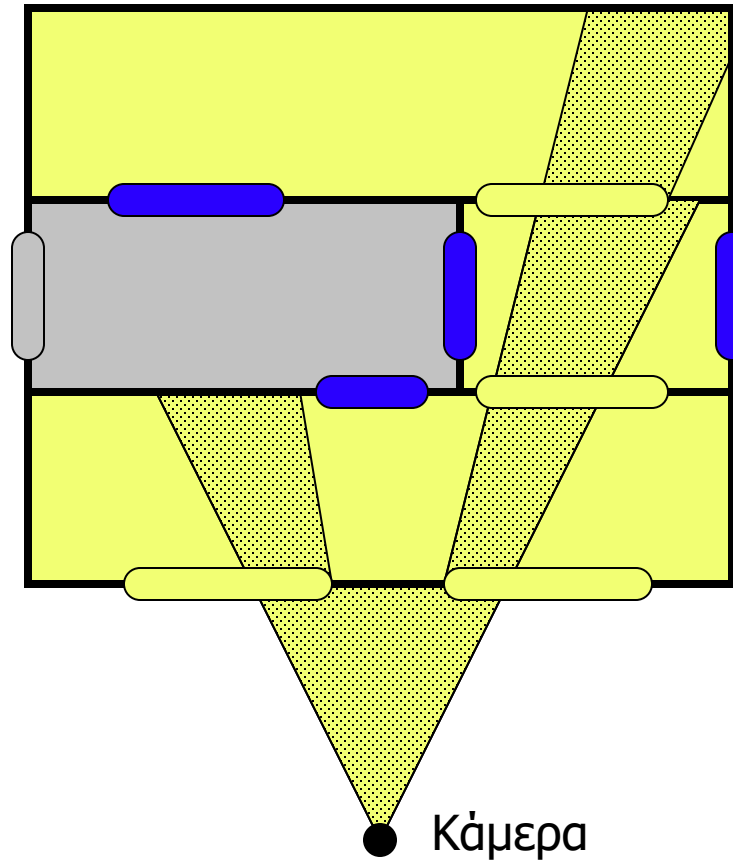
Μαρκάρισμα του κόμβου
ως ορατός και έλεγχος
των πυλών αυτού του
κόμβου



- Κόμβος
- Πύλη
- Ορατό
- Αόρατο
- Δεν έχει ελεγχθεί

Διαμερισμός Όγκων – Πύλες

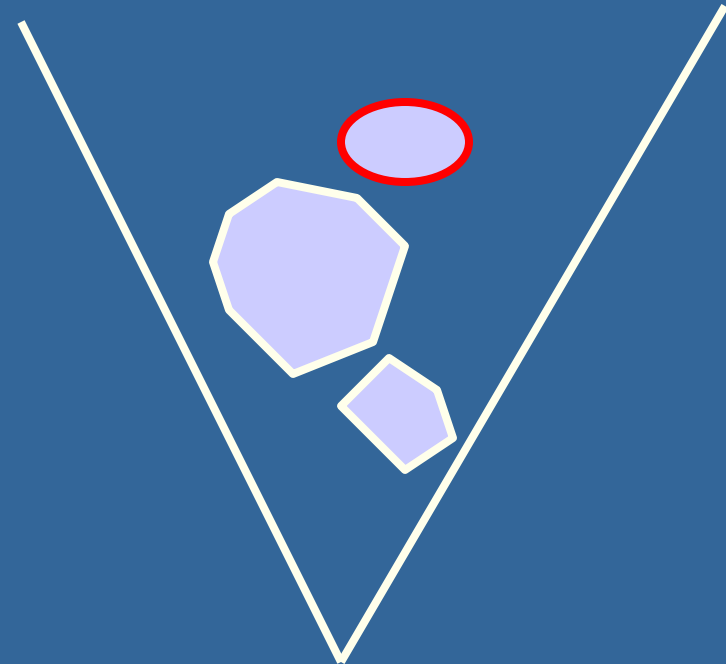
Ένας κόμβος αόρατος.
Κανένας κόμβος για
έλεγχο. Απόδοση σκηνής.



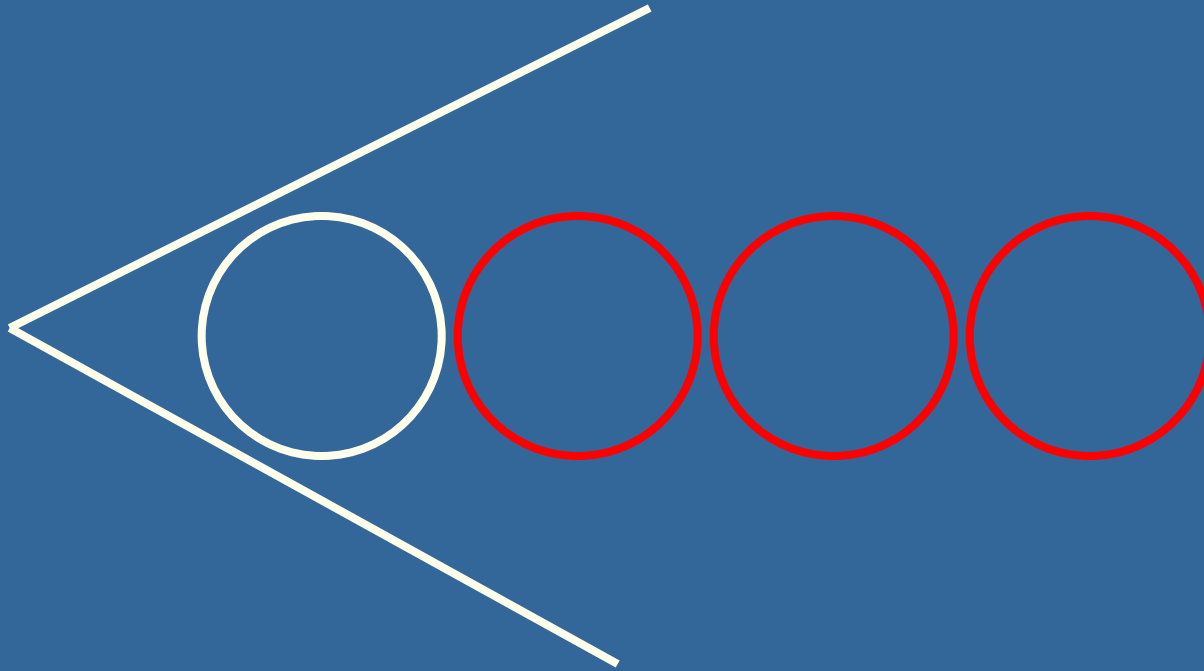
- Κόμβος
- Πύλη
- Ορατό
- Αόρατο
- Δεν έχει ελεγχθεί

Αποκοπή Απόκρυψης (Occlusion Culling) – GPU

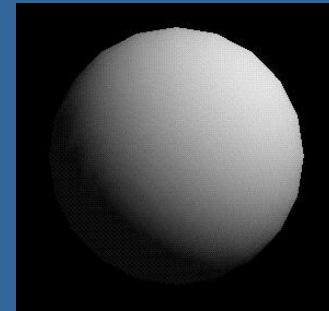
- Αντικείμενα που κείνται εξ ολοκλήρου «πίσω» από άλλα αντικείμενα θα πρέπει να αποκοπούν
- Δύσκολο πρόβλημα
- Αρκετή έρευνα



Παράδειγμα



Τελική Εικόνα



Η αποκοπή πύλης είναι ένας τύπος αποκοπής απόκρυψης

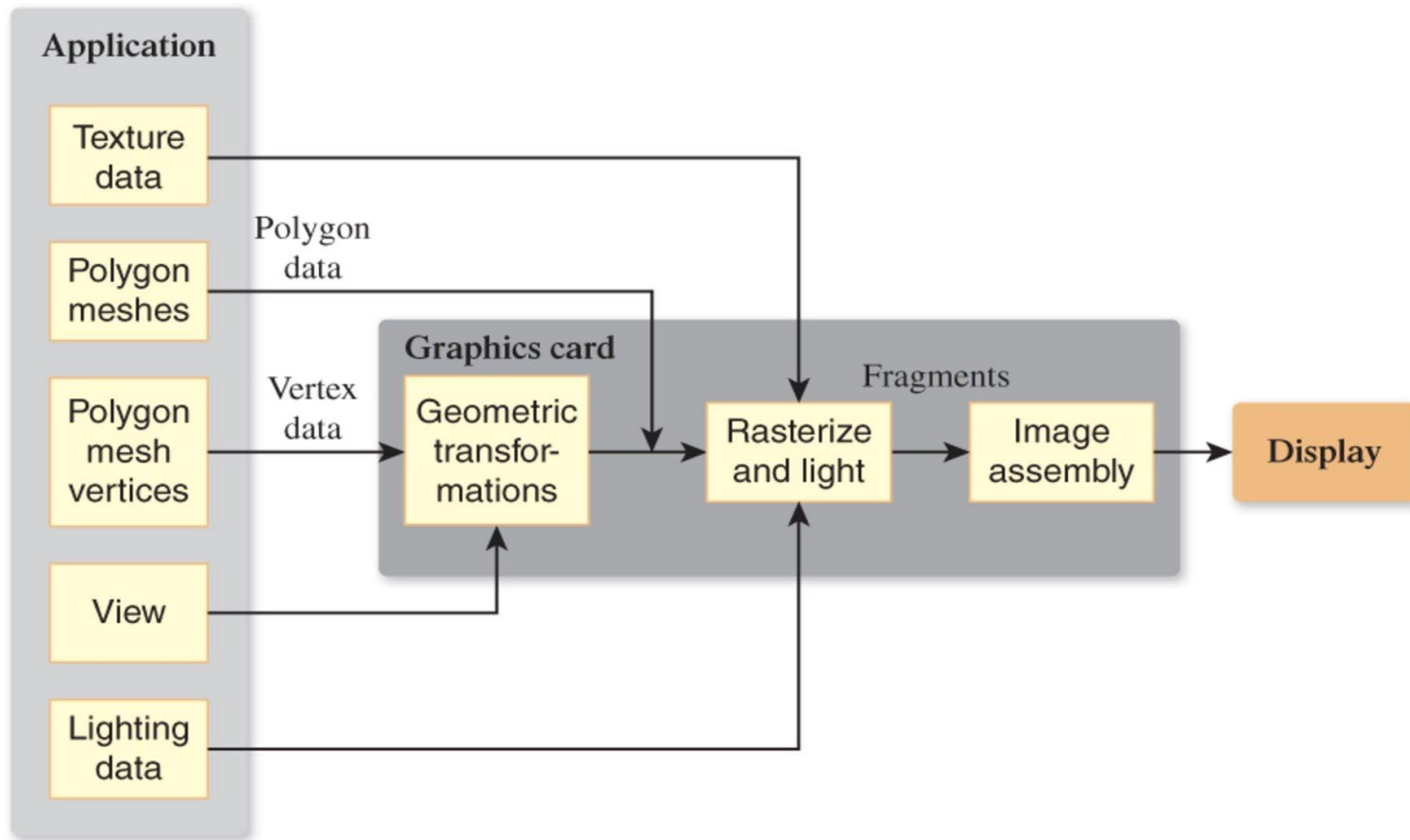
GEOMETRY DASH



Το Στάδιο της Γεωμετρίας

Vertex Shaders

Στη GPU



- Τα αντικείμενα της σκηνής είναι συνήθως με τη μορφή πολυγωνικών πλεγμάτων
- Το μεγαλύτερο μέρος της εργασίας για την απόδοση μιας εικόνας γίνεται στη GPU
- Ο κώδικας στη GPU θα έχει τουλάχιστον δύο μέρη:
 - **Vertex Shader**
 - **Fragment Shader**

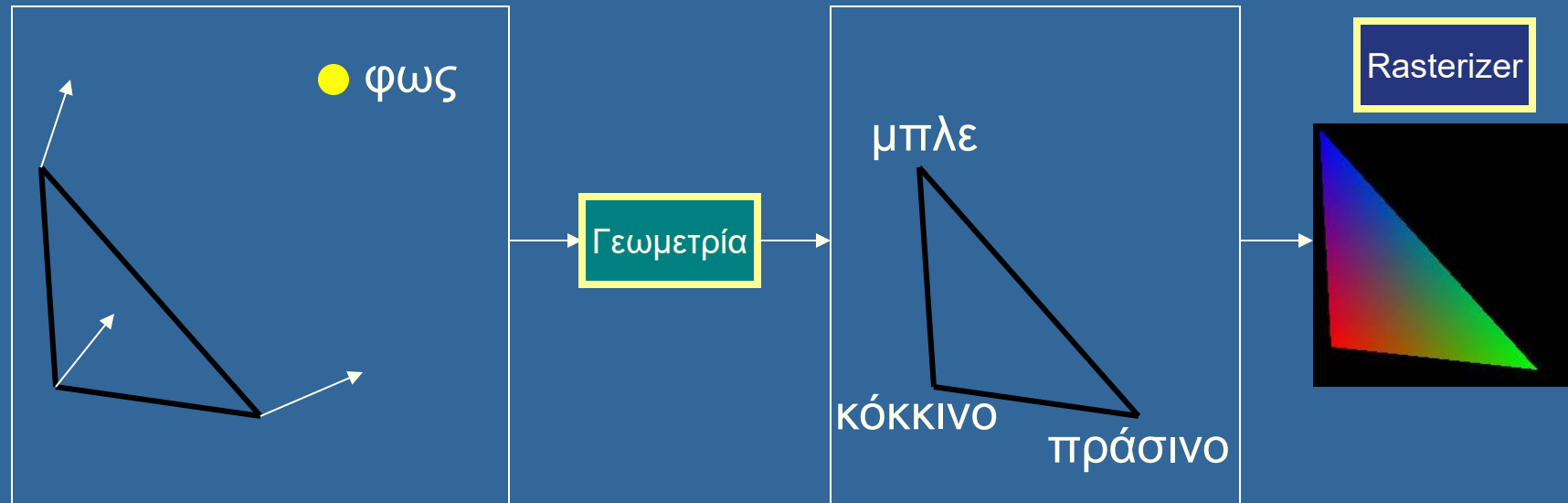
Το Στάδιο της Γεωμετρίας σε Λεπτομέρεια

Τα έχουμε δει ήδη:

- Μετασχηματισμοί αντικειμένων (μετακίνηση, κλιμάκωση, περιστροφή) από το χώρο του μοντέλου στο χώρο του κόσμου
- Μπορείτε να μετακινήσετε και την κάμερα με τον ίδιο τρόπο
- Προβολές
- Προβολή στην Οθόνη

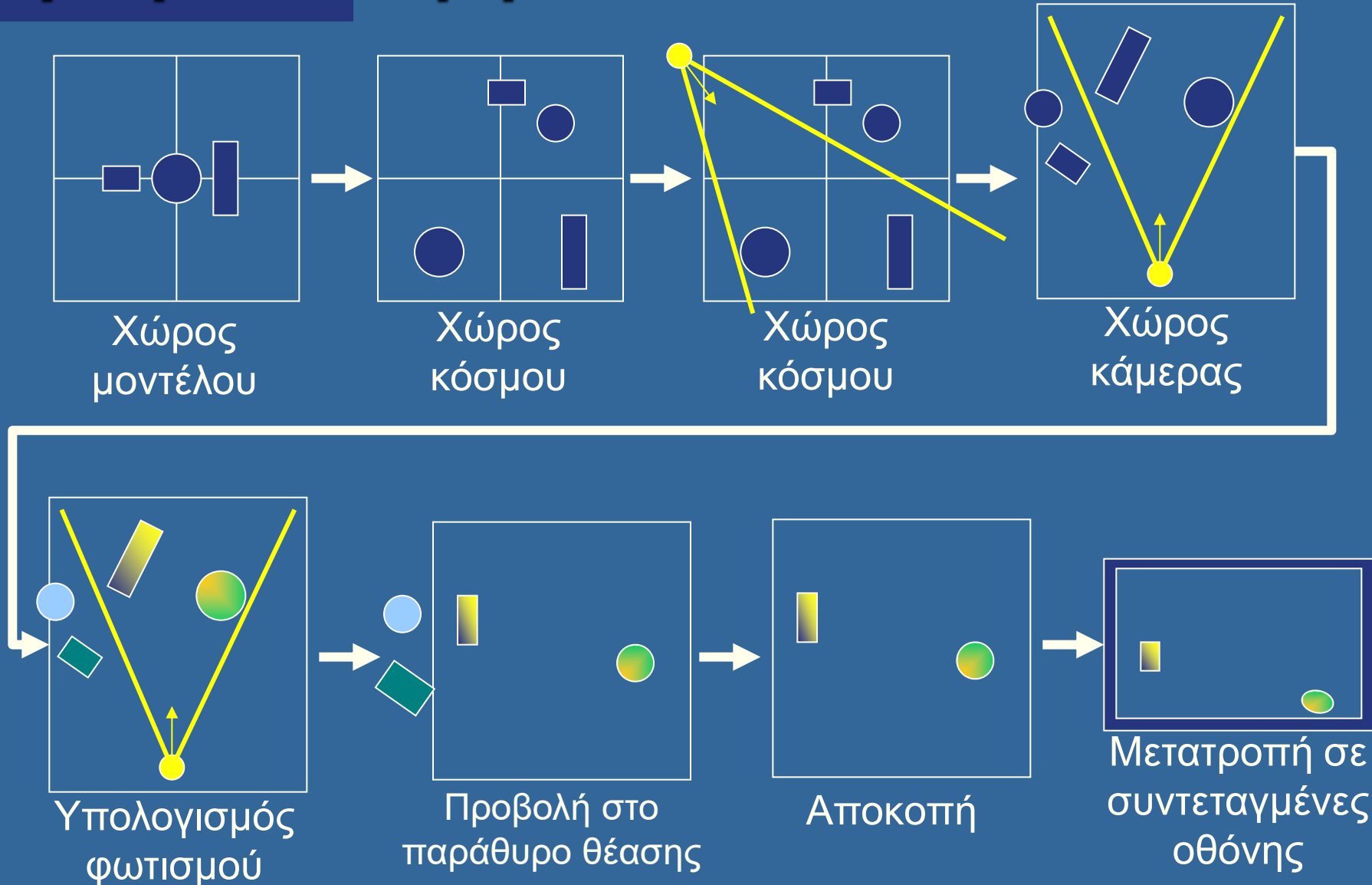
Φωτισμός

- Υπολογισμός φωτός στις κορυφές



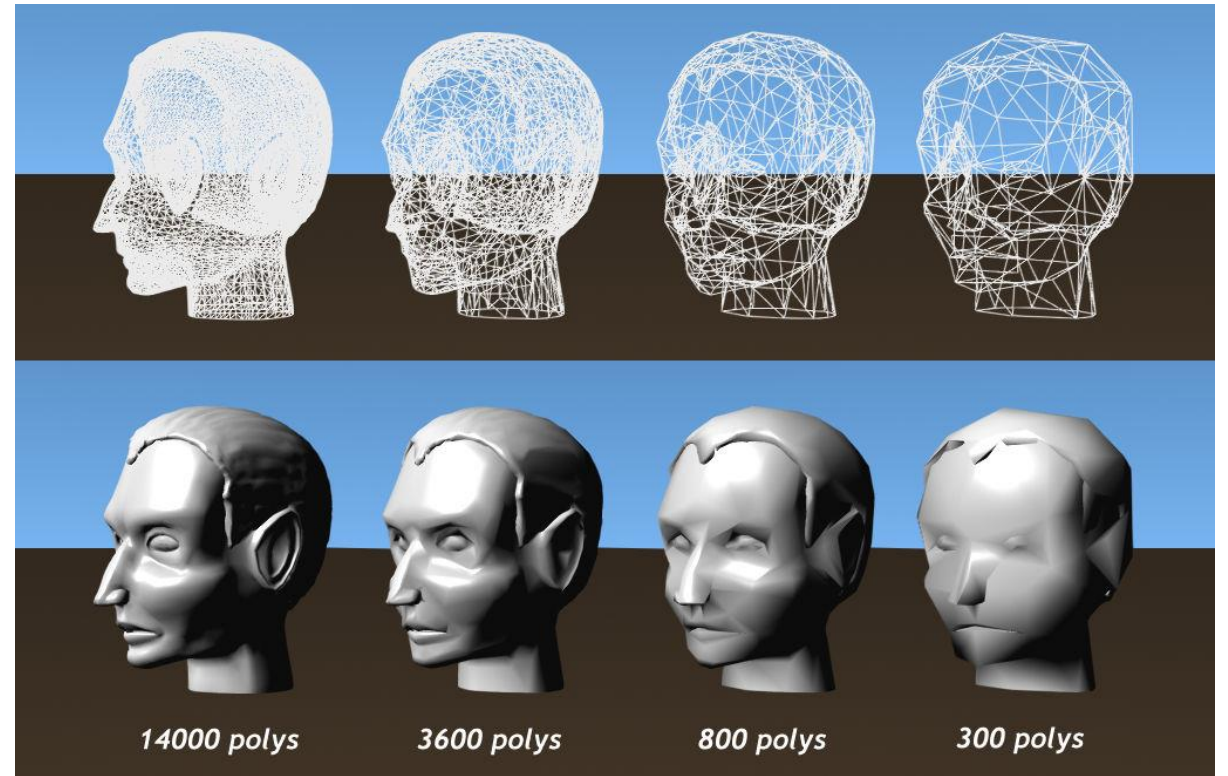
- Μίμηση της συμπεριφοράς του φωτός στη φύση
 - Δύσκολο και για αυτό χρησιμοποιούνται εμπειρικά μοντέλα, διάφορα κόλπα, και λίγο από πραγματική θεωρία

Γεωμετρία: Όλη η Ακολουθία



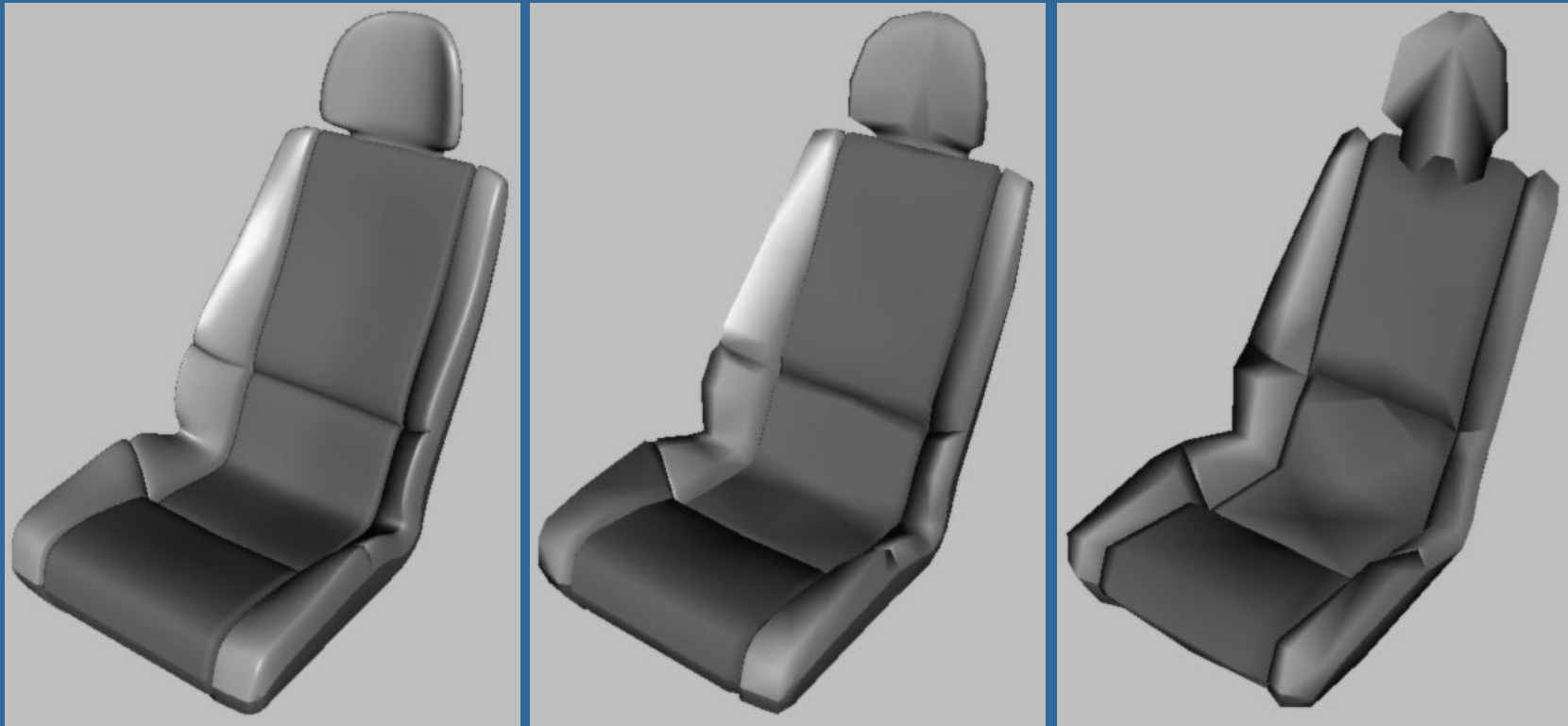
Απόδοση σε Διαφορετικά Επίπεδα Λεπτομέρειες

Level-of-Detail (LoD)
Rendering



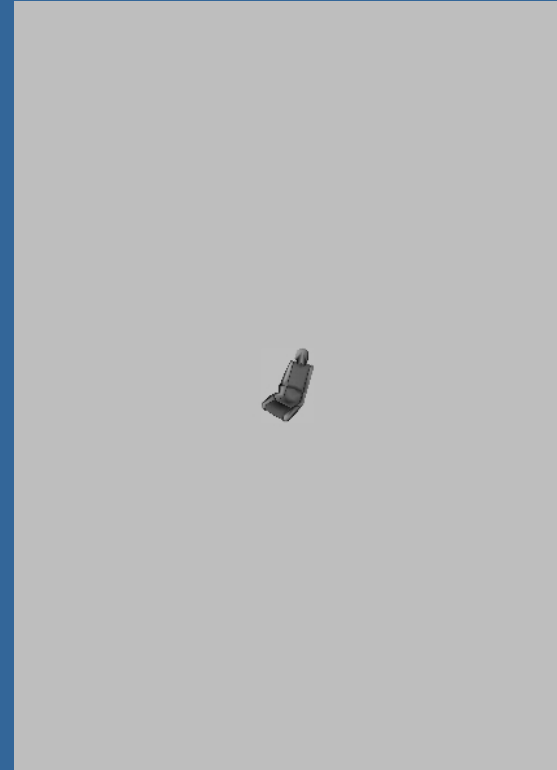
LoD

- Χρήση διαφορετικών επιπέδων λεπτομέρειας σε διαφορετικές αποστάσεις από τον θεατή
- Περισσότερα τρίγωνα κοντά στον θεατή



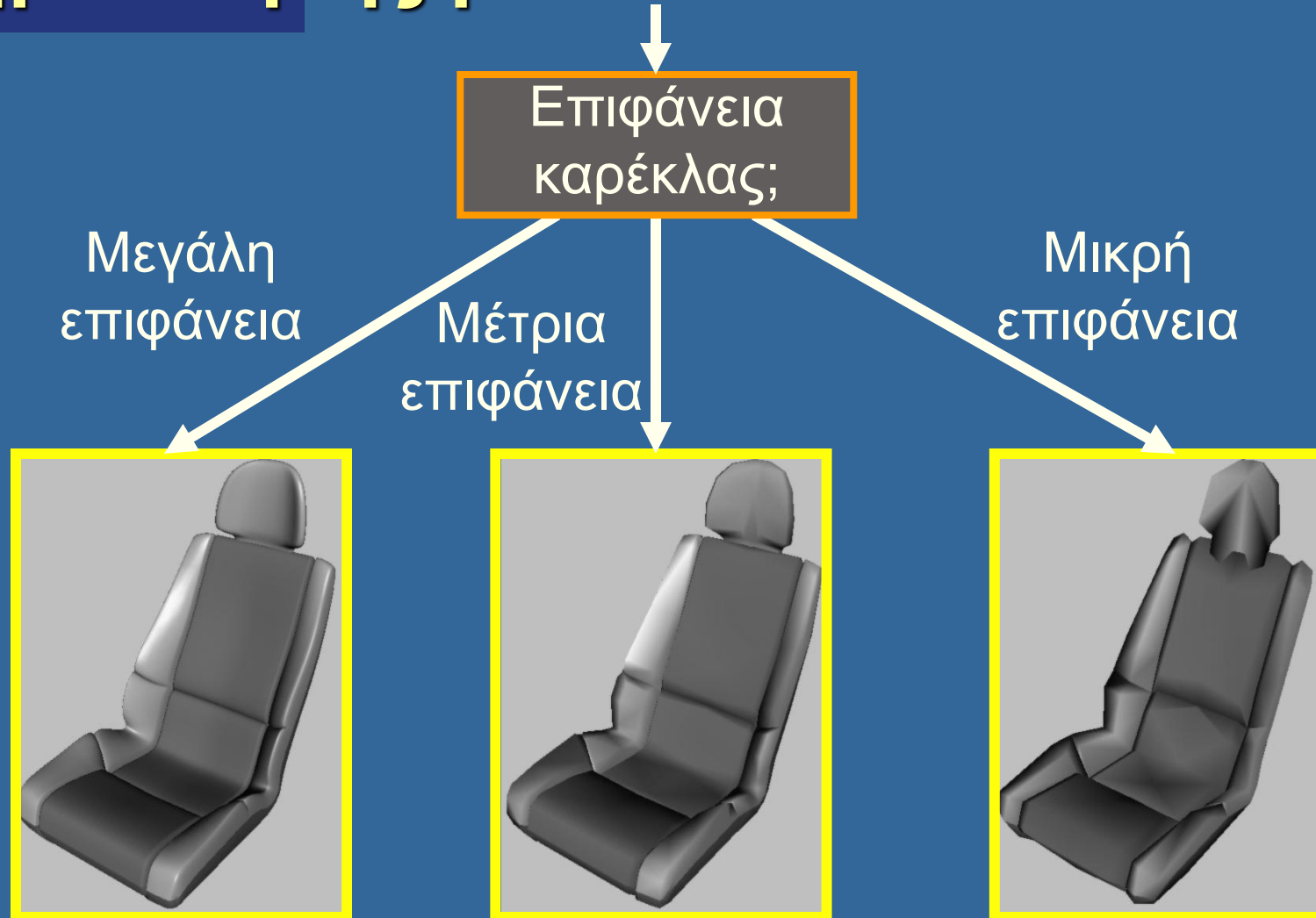
LoD Απόδοση Γραφικών

- Δεν έχει οπτική διαφορά (τουλάχιστον όχι μεγάλη) αλλά τεράστια διαφορά στην ταχύτητα



- Χρήση της επιφάνειας προβολής του BV για την επιλογή του μοντέλου με το κατάλληλο επίπεδο λεπτομέρειας

Γράφημα Σκηνής με LoDs



LoD Απόδοση για Απομακρυσμένα Αντικείμενα

- Όταν το αντικείμενο είναι μακριά, αντικατέστησέ το με ένα τετράγωνο κάποιου χρώματος
- Όταν το αντικείμενο είναι εξαιρετικά μακριά, δεν το επεξεργαζόμαστε (αποκοπή λεπτομέρειας – detail culling)!
- Χρήση της επιφάνειας προβολής για το πότε να γίνει αποκοπή

Unity

- Automatic LoD is a powerful Unity extension that allows you to quickly generate and manage multiple levels of detail for your 3D models.
- LoD levels are simplified meshes that are generated procedurally by the extension and help optimize your game especially on lower end platforms.
- Besides the generation process, the Automatic LoD component also takes care during the game of enabling the correct mesh for each object depending on its distance to the camera or its screen area covered. This way you can use meshes with lower polygon count when the object is far away or covers small screen space and higher polygon count for up-close views.



Unreal Engine 5

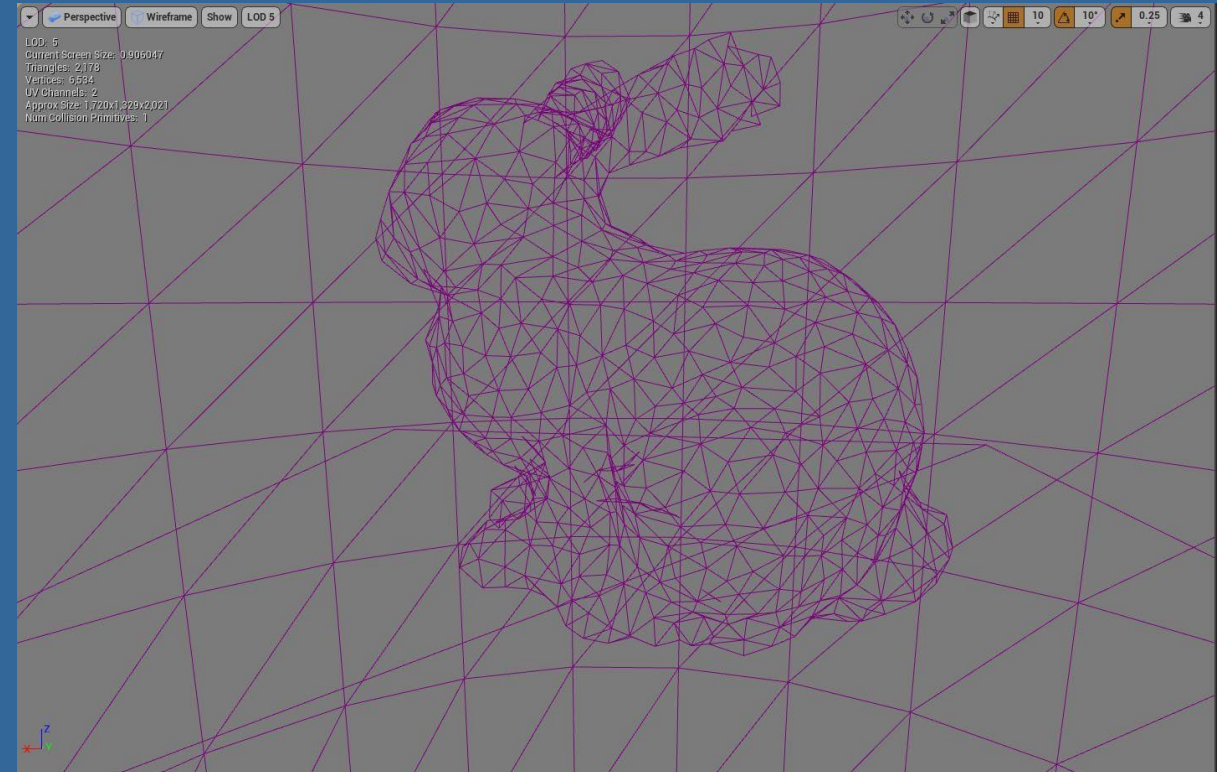
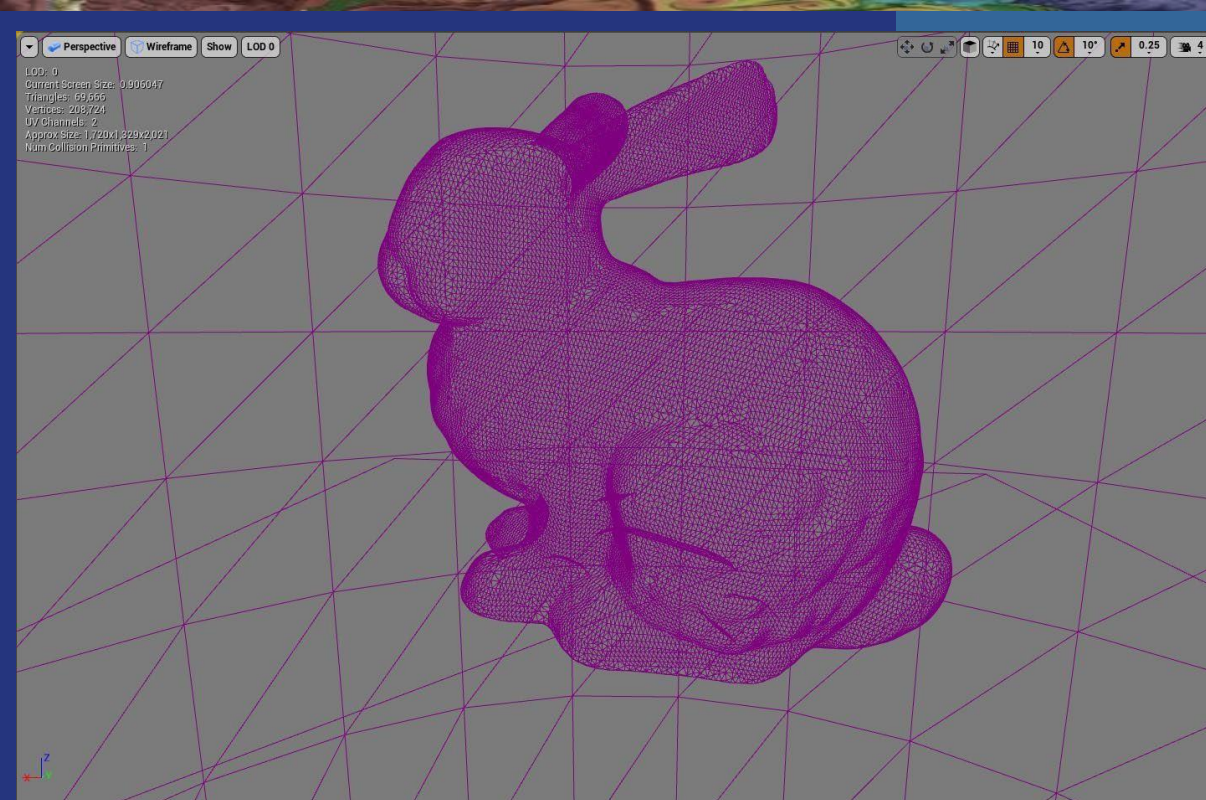
Static Mesh Automatic LOD Generation

How To use the Automatic LOD Generation system in UE5.



The Automatic LOD generation system allows you to automatically reduce the polygon count of your Static Meshes to create LODs with the Unreal Engine 5 (UE5) Editor. Automatic LOD generation uses what is called quadratic mesh simplification to help generate the LODs for Static Meshes. Quadratic mesh simplification works by calculating the amount of visual difference that collapsing an edge (by merging two vertices) would generate. It then picks the edge with the least amount of visual impact and collapses it. When this happens, the tool will pick the best place to put the newly merged vertex, removing any triangles that have also collapsed along with the edge. It will continue to collapse edges until it reaches the requested target number of triangles. In the following guide, we'll show you how-to setup and use the automatic LOD generation system in your UE5 projects.

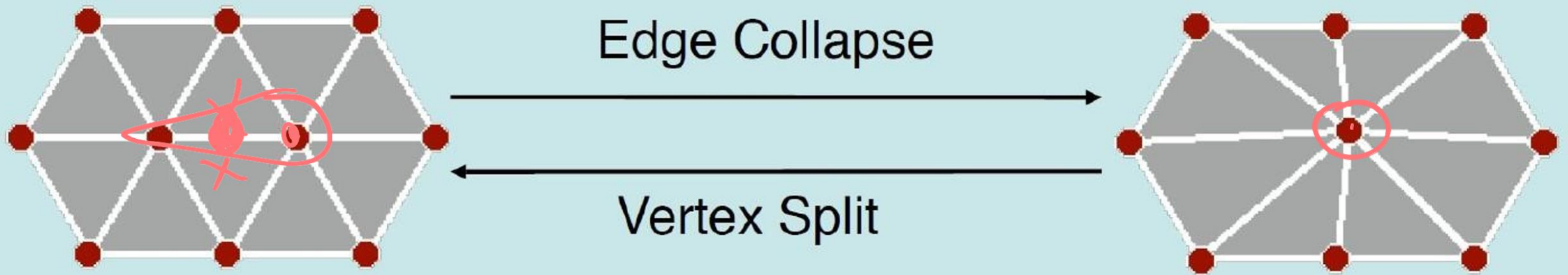
Σύγκριση: LoD0 έναντι LoD5 σε Unreal



Το LoD5 έχει περίπου 3% του πλήθους των τριγώνων του LoD0

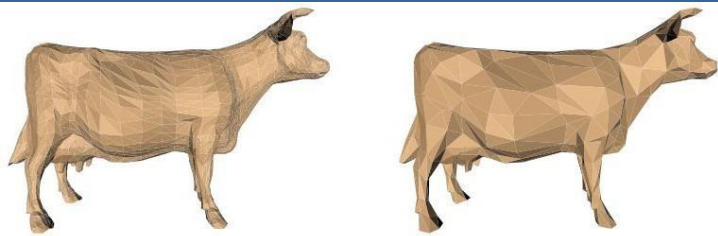
Πώς γίνεται το LoD αυτόματα;

Αυξητική Απλοποίηση: Συστολή Ακμών



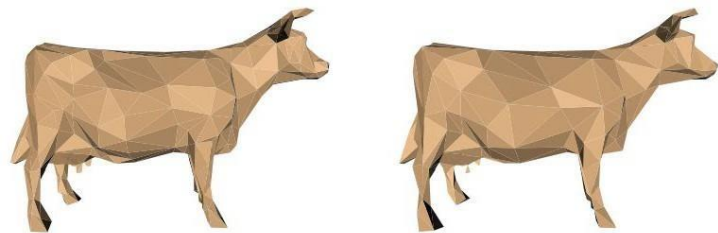
1. Συγχώνευση δύο γειτονικών τριγώνων
2. Ορισμός νέας θέσης της κορυφής

Άλλο Παράδειγμα



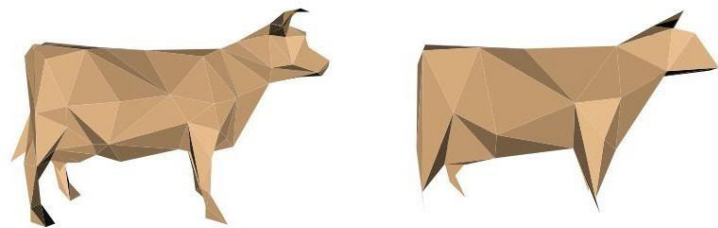
(a) 5804 faces

(b) 1000 faces



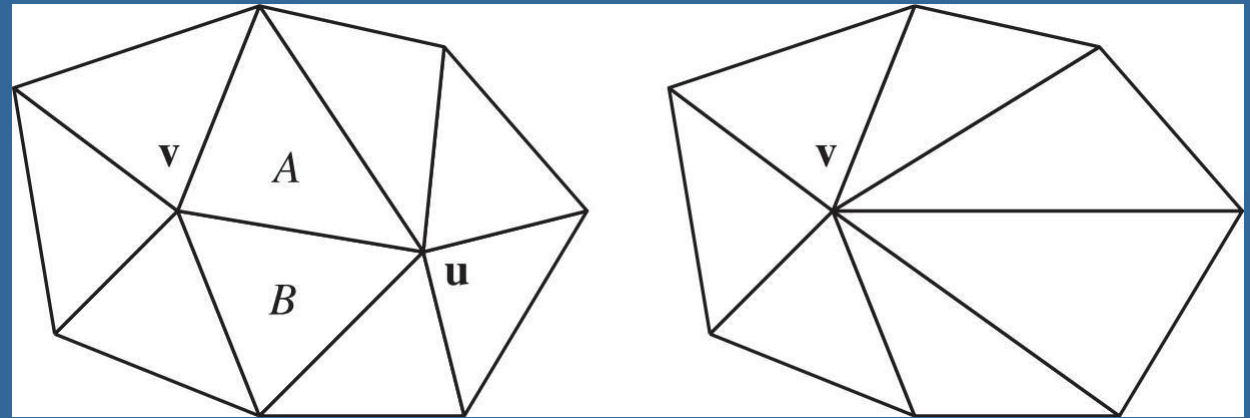
(c) 800 faces

(d) 500 faces



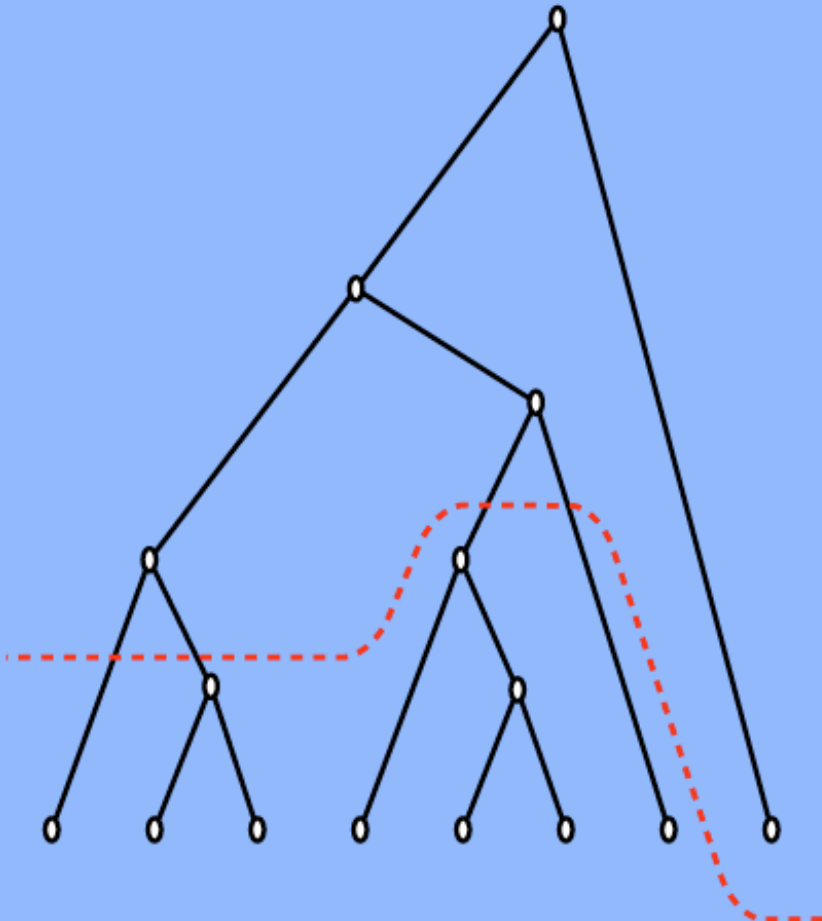
(e) 300 faces

(f) 100 faces



Συνεχές LoD με Ιεραρχίες Κορυφών

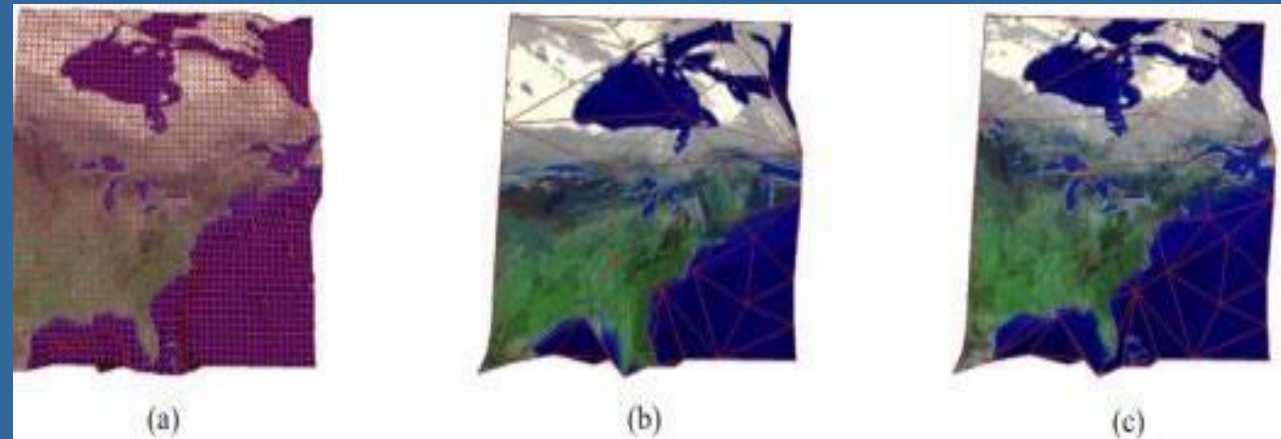
Μία από τις
κεντρικές ιδέες
του Nanite



- Κάθε αρχική κορυφή σε πλέγμα είναι ένα φύλλο σε αυτό το διάγραμμα
- Μια συστολή ακμής κάνει τις κορυφές αδέρφια δημιουργώντας έναν κοινό γονέα
- Το κόψιμο του δέντρου καθορίζει ένα σύνολο συστολών που εφαρμόζονται στο πλέγμα
- Θα μπορούσε να κάνει CLoD (Continuous LoD) χρησιμοποιώντας τη μέτρηση χώρου οθόνης για τον προσδιορισμό της αποκοπής
- ...για κάθε frame πρέπει να γίνεται προσαρμογή

Στην πράξη το LoD είναι αρκετά πιο περίπλοκο

- Το σφάλμα θα πρέπει να λαμβάνει υπόψη το χρώμα και τις υφές
- ...δεν θέλουμε να συγχωνεύουμε μη συνεχή τμήματα των υφών



Geometry & texture: A 3,872 face model (a) reduced to 53 faces without (b) and with (c) updating texture coordinates.

- Μερικές φορές κάποια χαρακτηριστικά πρέπει να παραμείνουν ή να κοπούν (απαιτεί είσοδο από τον καλλιτέχνη)



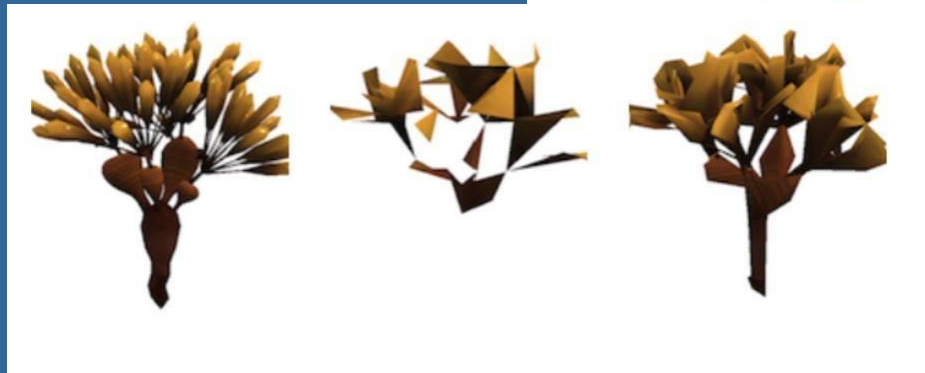
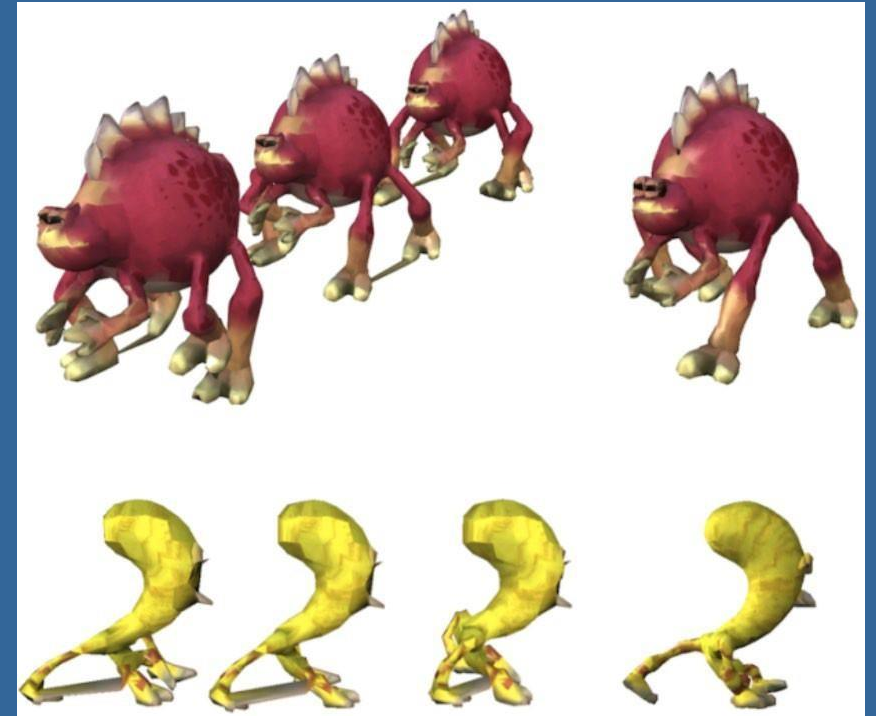
Χρησιμοποιείται σε Παιχνίδια

Rapid Simplification of Multi-Attribute Meshes

Andrew Willmott*
Maxis/Electronic Arts



Figure 1: From left: 160,000-triangle textured model; simplification by vertex clustering to 30,000 triangles in 40ms, causing artefacts at UV seams; artefacts eliminated by our algorithm in an additional 7ms; heavy simplification causing blockiness and dropouts in 13ms; improved by our extended algorithm in 3ms.



Spore (2008)



Developer(s)	Maxis
Publisher(s)	Electronic Arts
Designer(s)	Will Wright Alex Hutchinson Jenna Chalmers Chaim Gingold Stone Librande Soren Johnson
Programmer(s)	Andrew Willmott
Artist(s)	Michael A. Khoury
Composer(s)	Brian Eno Cliff Martinez
Platform(s)	Microsoft Windows ^[2] Mac OS X ^[2]
Release	September 5, 2008 ^{[1][2]}
Genre(s)	God game, life simulation, real-time strategy
Mode(s)	Single-player

Δυναμική αλλαγή του LoD των χαρακτήρων ανάλογα με την απόσταση ώστε να μην έχει υψηλές υπολογιστικές απαιτήσεις παρά το πλούσιο οπτικό περιεχόμενο και τους μηχανισμούς.

1. Διαδικαστική παραγωγή στοιχείων
2. Το LoD άλλαζε ανάλογα και με το gameplay
3. Το LoD εφαρμόστηκε και στα animations
4. LoD εφαρμόστηκε και στις υφές όπως και στους shaders
5. Αυτόματη ρύθμιση LoD ανάλογα τις επιδόσεις του συστήματος (πιο επιθετικό LoD όταν το σύστημα είχε χαμηλές επιδόσεις)

Abstract

We present a rapid simplification algorithm for meshes with multiple vertex attributes, targeted at rendering acceleration for real-time applications. Such meshes potentially feature normals, tangents, one or more texture coordinate sets, and animation information, such as blend weights and indices. Simplification algorithms in the literature typically focus on position-based meshes only, though extensions to handle surface attributes have been explored for those techniques based on iterative edge contraction. We show how to achieve the same goal for the faster class of algorithms based on vertex clustering, despite the comparative lack of connectivity information available. In particular, we show how to handle attribute discontinuities, preserve thin features, and avoid animation-unfriendly contractions, all issues which prevent the base algorithm from being used in a production situation.

Our application area is the generation of multiple levels of detail for player-created meshes at runtime, while the main game process continues to run. As such the robustness of the simplification algorithm employed is key; ours has been run successfully on many millions of such models, with no preprocessing required. The algorithm is of application anywhere rapid mesh simplification of standard textured and animated models is desired.



A scene from *The Last of Us: Remastered* (2014/™ SIE. Created and developed by Naughty Dog, PlayStation 4) rendered without textures.



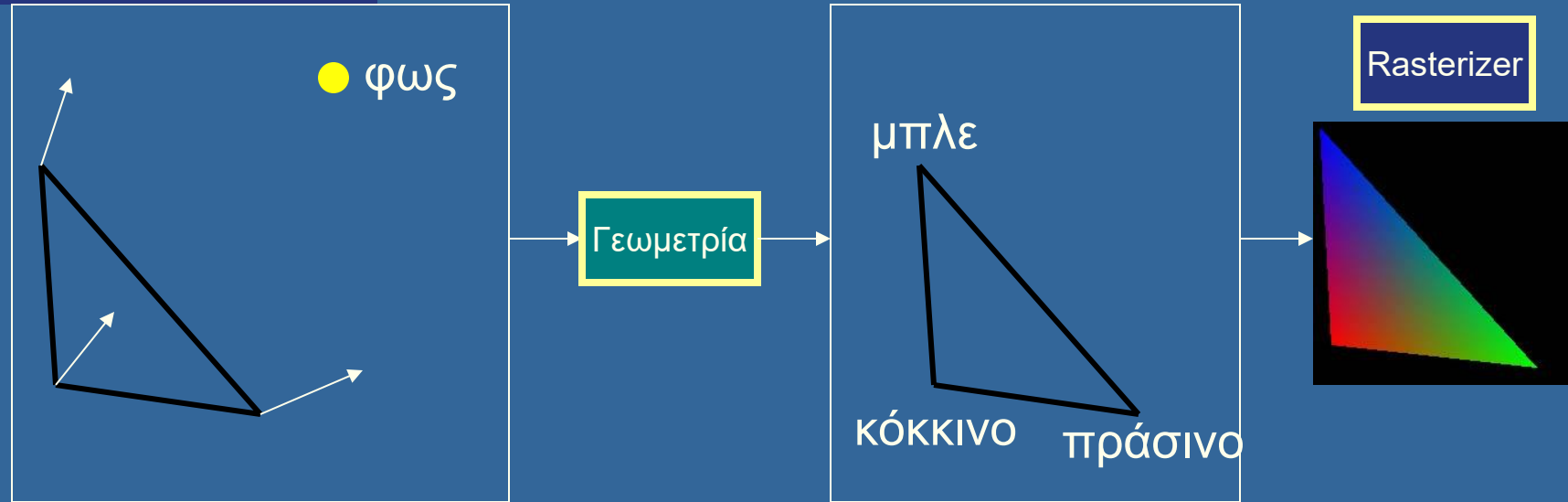
The same scene from *The Last of Us: Remastered* (© 2014/™ SIE. Created and developed by Naughty Dog, PlayStation 4) with only diffuse textures applied.



Scene from *The Last of Us: Remastered* (© 2014/™ SIE. Created and developed by Naughty Dog, PlayStation 4) with full lighting.

Φωτισμός

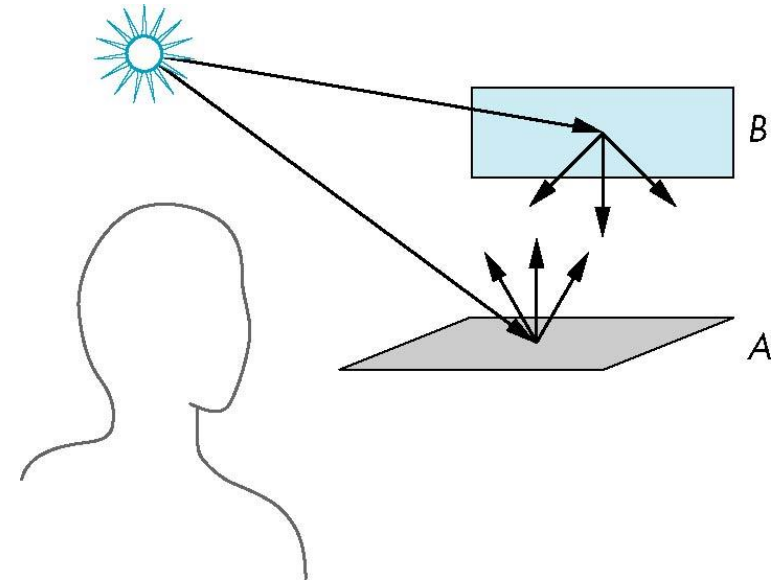
Υπολογισμός Φωτισμού



- Πώς υπολογίζεται ο φωτισμός;
- Θα μπορούσαμε να θέσουμε τα χρώματα στις κορυφές χειροκίνητα ☹️
- Για *περισσότερο* ρεαλισμό, υπολογίζουμε το φωτισμό από
 - Πηγές φωτός
 - Ιδιότητες υλικών (materials)
 - Γεωμετρικές σχέσεις αντικειμένων (θέση θεατή, προσανατολισμός επιφάνειας)

Σκέδαση Φωτός

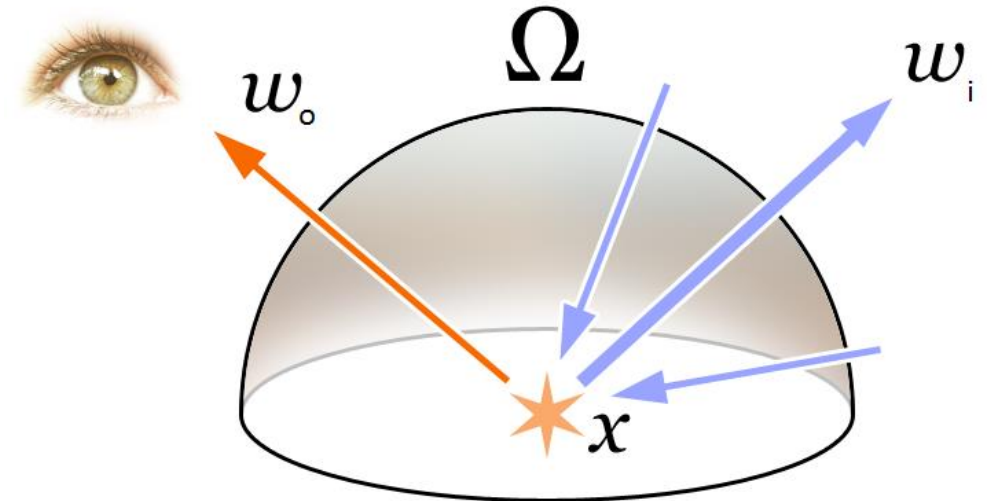
- Το φως χτυπά το σημείο **A**
 - Σκέδαση
 - Απορρόφηση
- Κάποιο διάσπαρτο φως (από το **A**) χτυπά το σημείο **B**
 - Σκέδαση
 - Απορρόφηση
- Κάποιο διάσπαρτο φως του **B** χτυπά το **A** και ούτω καθεξής ...



Η Εξίσωση Απόδοσης

(Rendering Equation)

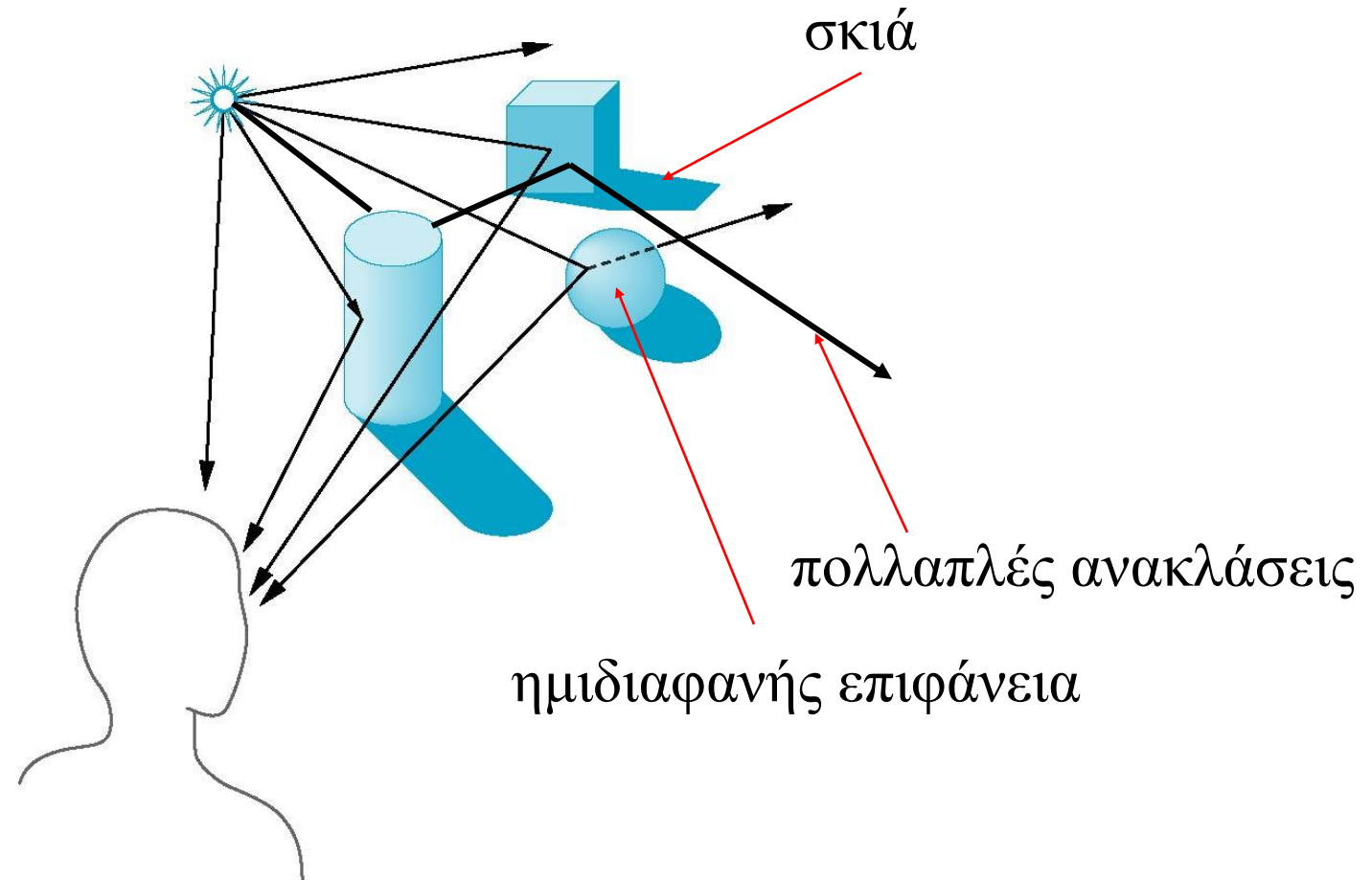
- Η άπειρη σκέδαση και απορρόφηση του φωτός περιγράφεται από την εξίσωση απόδοσης
 - Δεν μπορεί να λυθεί γενικά
 - Η τεχνική Ray-tracing είναι μια ειδική περίπτωση για τέλεια ανακλαστικές επιφάνειες
- Η εξίσωση απόδοσης είναι καθολική και περιλαμβάνει
 - Σκιές
 - Πολλαπλή σκέδαση μεταξύ αντικειμένων



$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + L_r(\mathbf{x}, \omega_o, \lambda, t)$$

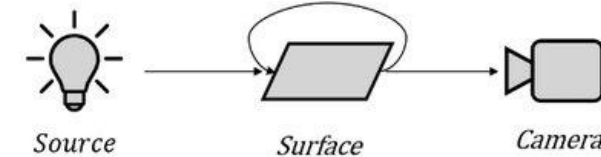
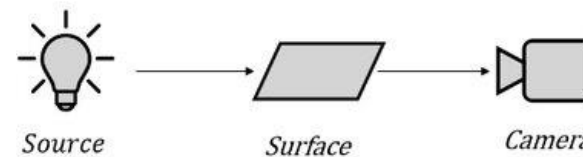
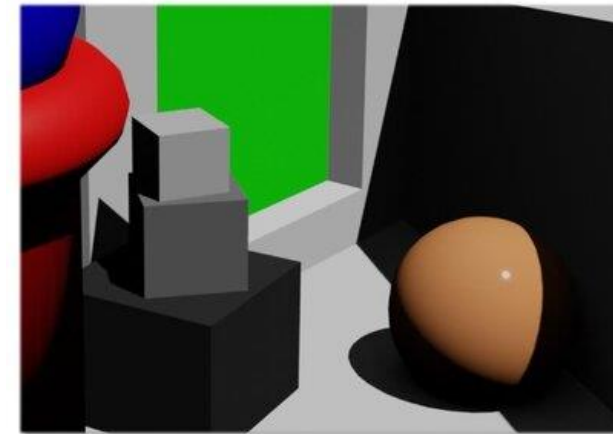
$$L_r(\mathbf{x}, \omega_o, \lambda, t) = \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{x}, \omega_i, \lambda, t) (\omega_i \cdot \mathbf{n}) d\omega_i$$

Καθολικές Επιδράσεις



Τοπικός και Καθολικός Φωτισμός

- Η σωστή σκίαση απαιτεί υπολογισμούς που περιλαμβάνουν όλα τα αντικείμενα και τις πηγές φωτός
 - Μη συμβατό με τη ροή εργασιών που σκιάζει κάθε πολύγωνο ανεξάρτητα (τοπική απόδοση)
- Ωστόσο, στα γραφικά υπολογιστών, ειδικά στα γραφικά σε πραγματικό χρόνο, είμαστε OK αν τα πράγματα «φαίνονται σωστά»
 - Υπάρχουν πολλές τεχνικές για την προσέγγιση των καθολικών επιδράσεων



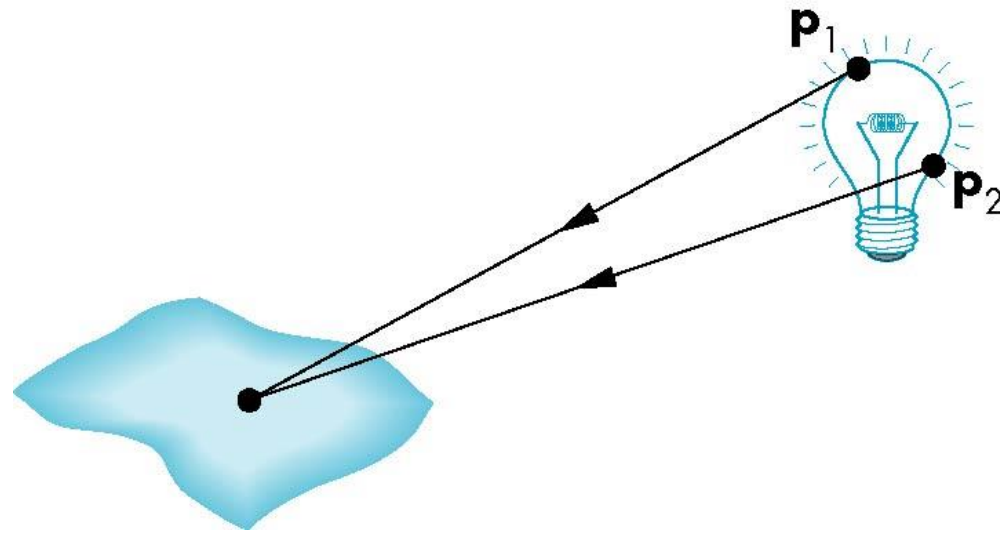
Αλληλοεπίδραση Φωτός και Υλικού

- Το φως που προσπίπτει σε ένα αντικείμενο απορροφάται μερικώς και σκεδάζεται μερικώς (ανακλάται)
- Η ποσότητα που ανακλάται καθορίζει το χρώμα και τη φωτεινότητα του αντικειμένου
 - Μια επιφάνεια εμφανίζεται κόκκινη κάτω από λευκό φως επειδή η κόκκινη συνιστώσα του φωτός αντανακλάται και το υπόλοιπο απορροφάται
- Το ανακλώμενο φως σκεδάζεται με τρόπο που εξαρτάται από την ομαλότητα και τον προσανατολισμό της επιφάνειας



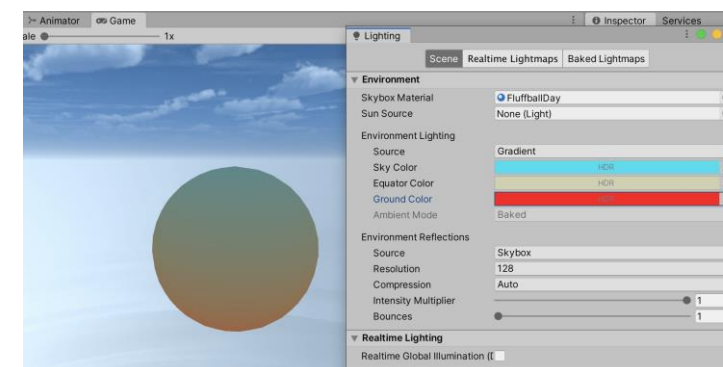
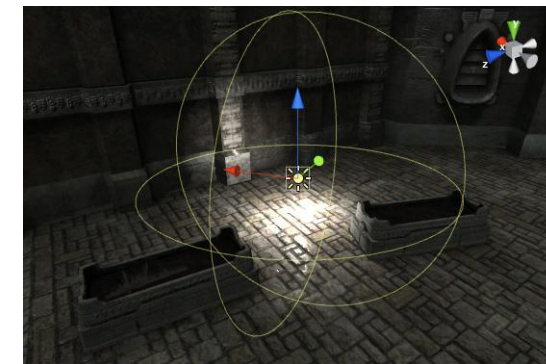
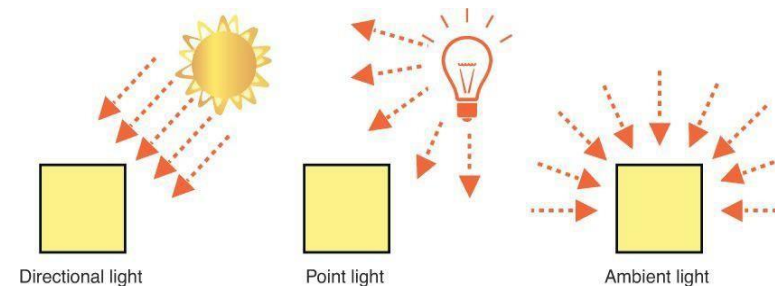
Πηγές Φωτός

Είναι δύσκολο να μοντελοποιήσουμε τις γενικές πηγές φωτός μιας και πρέπει να ενσωματώσουμε το φως που προέρχεται από όλα τα σημεία της πηγής



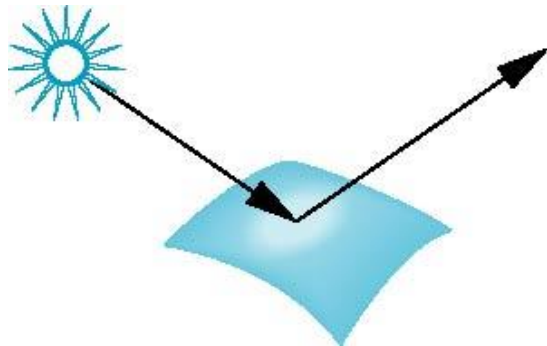
Απλές Πηγές Φωτός

- Σημειακή πηγή (point source)
 - Μοντελοποιείται με θέση και χρώμα
 - Μακρινή πηγή = άπειρη απόσταση (παράλληλες ακτίνες)
- Προβολέας (spotlight)
 - Περιορισμούς στο φως της σημειακής πηγής
- Φως περιβάλλοντος (ambient light)
 - Ίδια ποσότητα φωτός παντού στη σκηνή
 - Μπορεί να μοντελοποιήσει τη συμβολή πολλών πηγών και ανακλαστικών επιφανειών

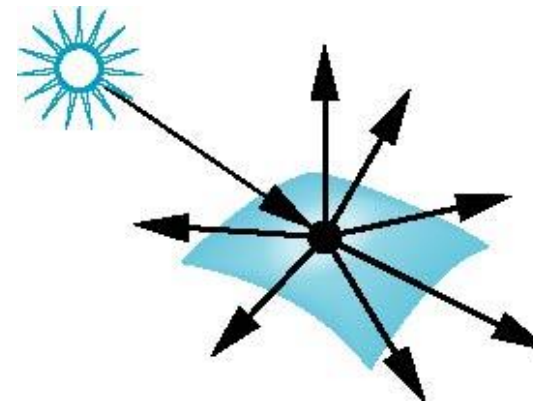


Τύποι Επιφανειών

- Όσο πιο λεία είναι μια επιφάνεια, τόσο περισσότερο ανακλώμενο φως συγκεντρώνεται προς την κατεύθυνση που ένας τέλειος καθρέφτης θα αντανακλούσε το φως
- Μια πολύ τραχιά επιφάνεια διασκορπίζει το φως προς όλες τις κατευθύνσεις



λεία επιφάνεια

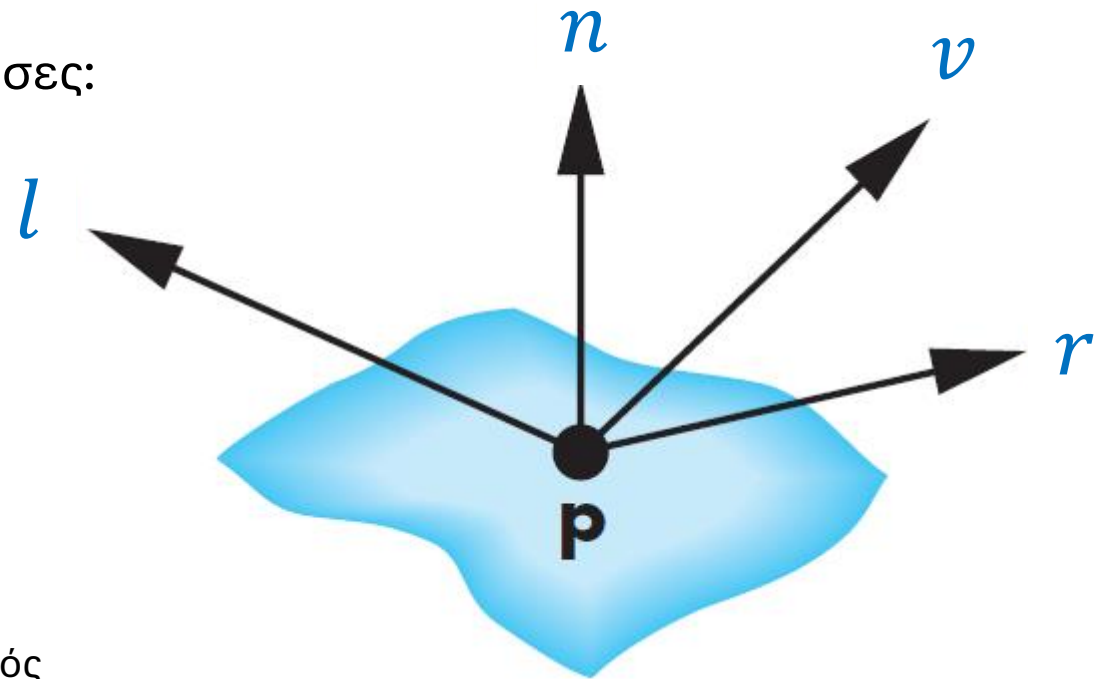


τραχιά επιφάνεια

Μοντέλο του Phong

- Ένα απλό μοντέλο που μπορεί να υπολογιστεί εύκολα – αν και απέχει πολύ από την πραγματικότητα της φυσικής του φωτός
- Το χρώμα σε ένα σημείο υπολογίζεται από τρεις συνιστώσες:
 - Διάχυτη ανάκλαση (Diffuse)
 - Κατοπτρική ανάκλαση (Specular)
 - Ανάκλαση από το περιβάλλον (Ambient)
- Χρησιμοποιεί τέσσερα διανύσματα
 - Προς το θεατή v
 - Προς τη πηγή φωτός (ή ένα σημείο της αν δεν είναι σημειακή) l
 - Κάθετο διάνυσμα n στο σημείο p
 - Το διάνυσμα r που δείχνει την τέλεια ανάκλαση από την πηγή φωτός

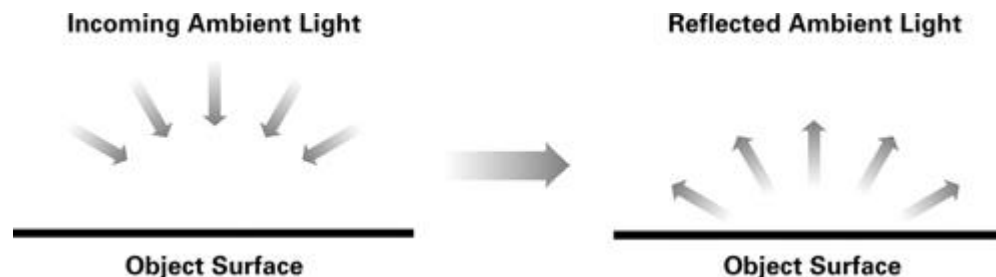
Κάθε πηγή μπορεί να έχει διαφορετικές συνιστώσες για διάχυτη, κατοπτρική και περιβαλλοντική ανάκλαση, μία για κάθε κανάλι χρώματος ($3 \times 3 = 9$)



Φως Περιβάλλοντος (φόντου)

(Ambient Light)

- Το φως που προκύπτει από τις αντανακλάσεις άλλων επιφανειών του περιβάλλοντος
- Ένα γενικό επίπεδο φωτεινότητας για μια σκηνή που είναι ανεξάρτητη από τις θέσεις φωτός ή τους προσανατολισμούς της επιφάνειας
- Δεν έχει κατεύθυνση
- Κάθε πηγή φωτός έχει μια συμβολή φωτός περιβάλλοντος L_a
- Για μια δεδομένη επιφάνεια, μπορούμε να καθορίσουμε πόσο φως περιβάλλοντος μπορεί να αντανακλά η επιφάνεια (χαρακτηριστικό του υλικού) χρησιμοποιώντας έναν συντελεστή ανάκλασης περιβάλλοντος: k_a ($0 \leq k_a \leq 1$)



Άρα μία επιφάνεια
αντανακλά $I_a = L_a \cdot k_a$
φως περιβάλλοντος

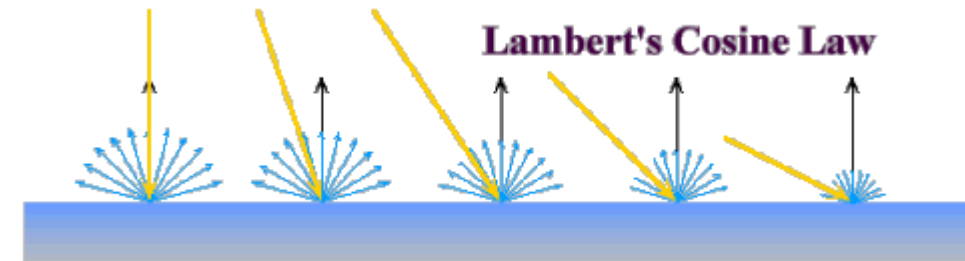
Διάχυτο Φως (Diffuse Light)

- Ο φωτισμός που λαμβάνει μια επιφάνεια από μια πηγή φωτός και αντανakλά εξίσου προς όλες τις κατευθύνσεις
- Αυτός ο τύπος ανάκλασης ονομάζεται Lambertian (αντίστοιχα Lambertian επιφάνειες)
- Η φωτεινότητα της επιφάνειας είναι ανεξάρτητη από τη θέση του παρατηρητή (καθώς το φως αντανakλάται εξίσου προς όλες τις κατευθύνσεις)

- Ο νόμος του Lambert: η ακτινοβολούμενη ενέργεια από ένα σημείο για μια δεδομένη πηγή φωτός είναι:

$$L_d \cdot \cos q$$

- L_d : η συνιστώσα διάχυσης της πηγής
- q : είναι η γωνία μεταξύ του κάθετου διανύσματος n και του διανύσματος της πηγής φωτός l



Η Συνιστώσα Διάχυσης

Ιδιότητα επιφάνειας υλικού: έστω ότι η επιφάνεια μπορεί να αντανακλά k_d ($0 \leq k_d \leq 1$) (συντελεστής ανάκλασης διάχυσης)

Ποσοστό φωτός διάχυσης:

$$I_d = k_d \cdot L_d \cdot \cos \theta$$



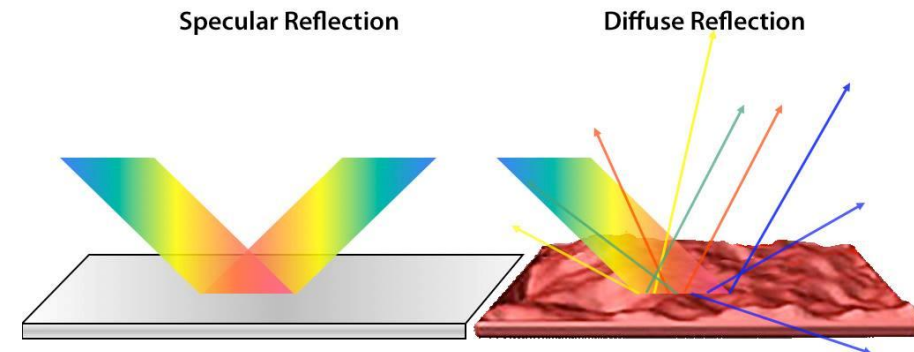
Φωτισμός σφαίρας από διαφορετικές γωνίες ($I_d + I_a$)

Κατοπτρικό Φως (Specular Light)

- Αυτά είναι τα φωτεινά σημεία σε αντικείμενα (όπως γυαλισμένο μέταλλο, μήλο ...)
- Το φως ανακλάται από την επιφάνεια άνισα προς όλες τις κατευθύνσεις.
- Το αποτέλεσμα της σχεδόν συνολικής ανάκλασης του προσπίπτοντος φωτός σε μια συγκεντρωμένη περιοχή γύρω από την κατοπτρική γωνία ανάκλασης

Ιδιότητα επιφάνειας υλικού: έστω ότι η επιφάνεια μπορεί να αντανακλά k_s ($0 \leq k_s \leq 1$), (συντελεστής κατοπτρικής ανάκλασης) ποσοστό φως κατοπτρικής ανάκλασης:

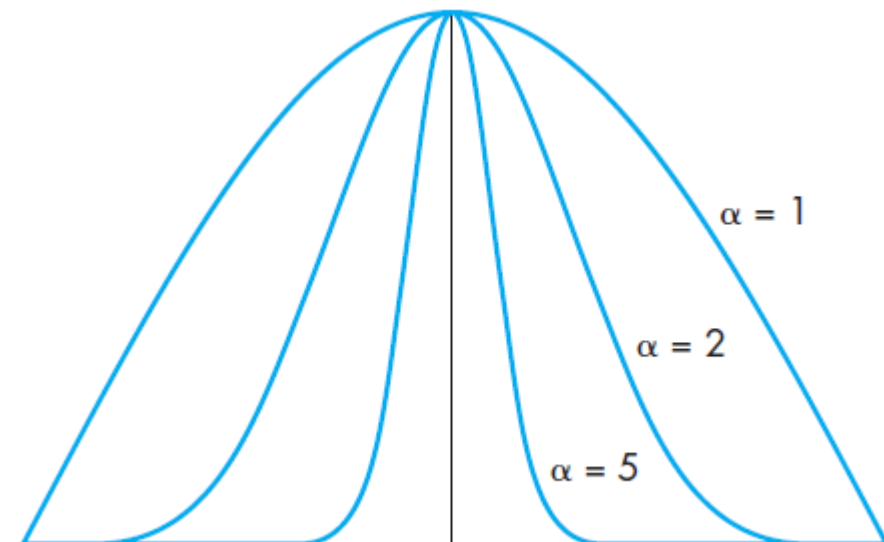
$$I_s = k_s \cdot L_s \cdot \cos^a \varphi$$



Συντελεστής Λάμψης

$$I_s = k_s \cdot L_s \cdot \cos^a \varphi$$

- Τιμές του a μεταξύ 100 και 200 αντιστοιχούν σε μέταλλα
- Τιμές του a μεταξύ του 5 και 10 αντιστοιχούν σε επιφάνειες που μοιάζουν πλαστικές
- Οι γυαλιστερές επιφάνειες αλλάζουν εμφάνιση όταν αλλάζει η οπτική γωνία
- Οι κατοπτρικές ιδιομορφίες προκαλούνται από μικροσκοπικά λείες επιφάνειες.
- Ένας καθρέφτης είναι ένας τέλειος κατοπτρικός ανακλαστήρας



Όλα Μαζί: Το Μοντέλο Phong

Ένα σημειακό φως:

$$I = I_a + I_d + I_s = k_a \cdot L_a + k_d \cdot L_d \cdot \cos \theta + k_s \cdot L_s \cdot \cos^a \varphi$$



Ambient



Diffuse



Specular



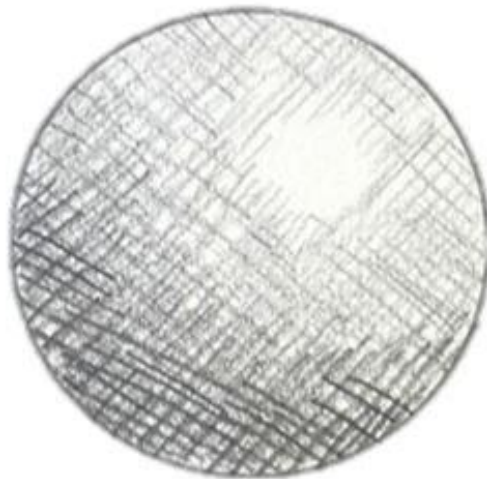
Προσθήκες

- Να εξαρτάται από την απόσταση (diffusion+specular):
$$\frac{1}{a+bt+ct^2}$$
- Περισσότερα φώτα: Απλά πρόσθεσε όλες τις συνεισφορές
- Διαφορετικοί τύποι φώτων
- Καθολικό φως περιβάλλοντος
- ...

Graphite Shading Techniques



Hatching



Crosshatching



Stumping



Stippling

Σκίαση – Βασικά Στοιχεία

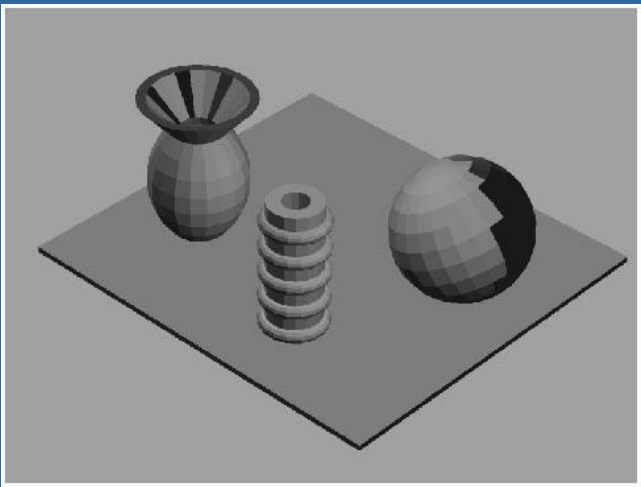
Πολλά εντάσσονται και στο
στάδιο του Rasterization

Φωτισμός και Σκίαση (Όχι σκιές ☺)

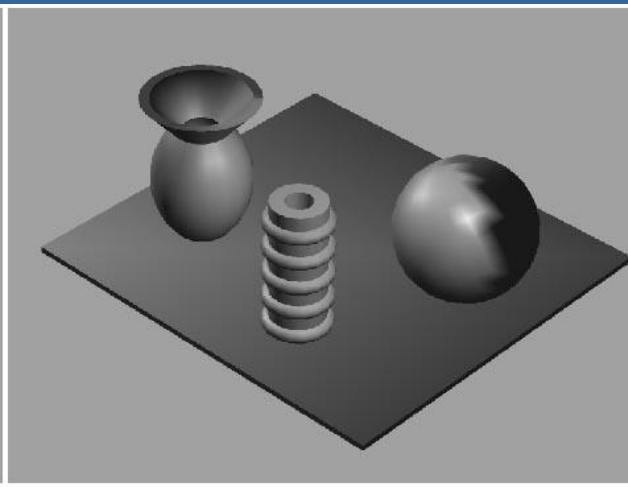
(Lighting and Shading)

- Φωτισμός: η αλληλεπίδραση μεταξύ φωτός και υλικών
- Σκίαση: εφάρμοσε φωτισμό στις κορυφές και καθόρισε το χρώμα των pixel από αυτές
- Τρεις τύποι κλασσικής σκίασης:

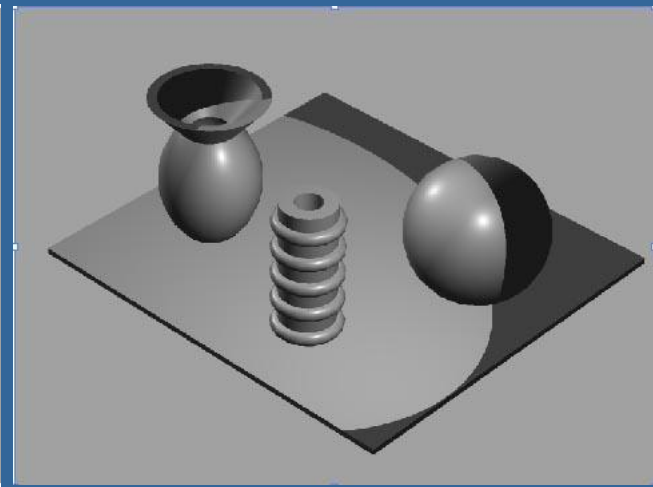
Επίπεδη



Gouraud



Phong



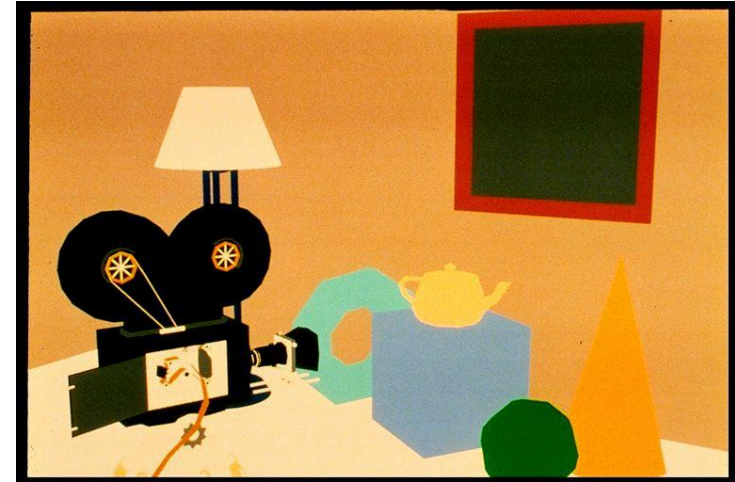
Μοντέλα Σκίασης για Πολύγωνα

- Σταθερή/Επίπεδη Σκίαση
 - Υπολογισμός φωτισμού σε κάποιο σημείο της επιφάνειας. Χρήση κάθετου διανύσματος της επιφάνειας του τριγώνου. Καλό για μακρινό φως και μακρινό θεατή ή εάν οι όψεις προσεγγίζουν καλά την επιφάνεια.
- Σκίαση ανά pixel
 - Υπολογισμός φωτισμού κάθε σημείου της επιφάνειας.
- Σκίαση με Παρεμβολή
 - Υπολογισμός φωτισμού στις κορυφές και παρεμβολή στα άλλα σημεία.

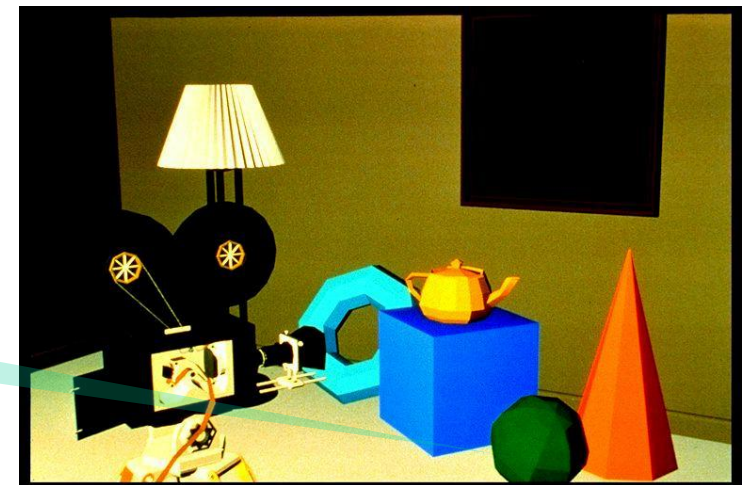
Επίπεδη Σκίαση



- Υπολογισμός φωτισμού σε ένα σημείο της επιφάνειας – η υπόλοιπη επιφάνεια έχει το ίδιο χρώμα παντού
- Δίνει καλά αποτελέσματα αν ισχύουν τα εξής:
 - Το αντικείμενο δεν είναι μια καμπύλη (λεία) επιφάνεια (π.χ., ένα πολύεδρο αντικείμενο)
 - Η πηγή φωτός είναι πολύ μακριά (άρα η γωνία του διανύσματος της πηγής με το κάθετο διάνυσμα της επιφάνειας δεν αλλάζει σημαντικά)
 - Το μάτι είναι πολύ μακριά (άρα η γωνία μεταξύ του θεατή και του ανακλώμενου φωτός δεν αλλάζει σημαντικά)
 - Η επιφάνεια είναι αρκετά μικρή (κοντά στο μέγεθος pixel)



Mach Band



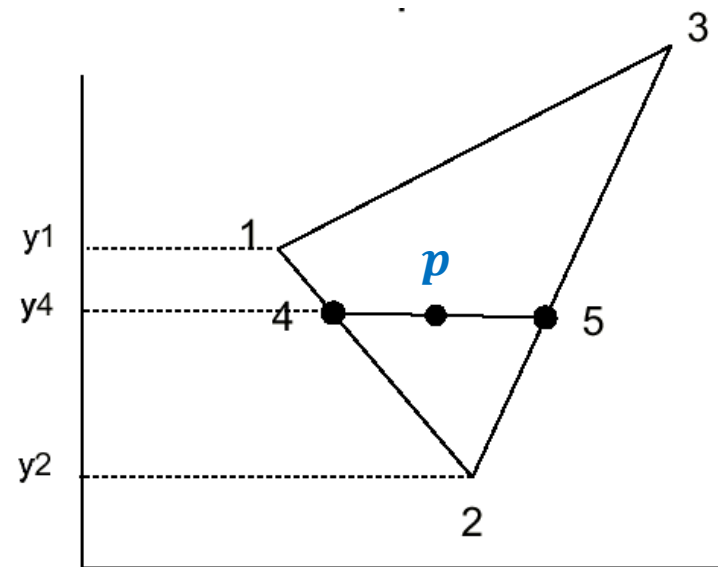
Λεία Σκίαση

- Χρειαζόμαστε κάθετες σε κάθε κορυφή
- Gouraud Σκίαση
 - Παρεμβολή χρώματος σε όλη την επιφάνεια του τριγώνου
 - Γρήγορο
- Phong Σκίαση
 - Παρεμβολή καθέτων διανυσμάτων στην επιφάνεια του τριγώνου
 - Πιο ακριβές αλλά πιο αργό

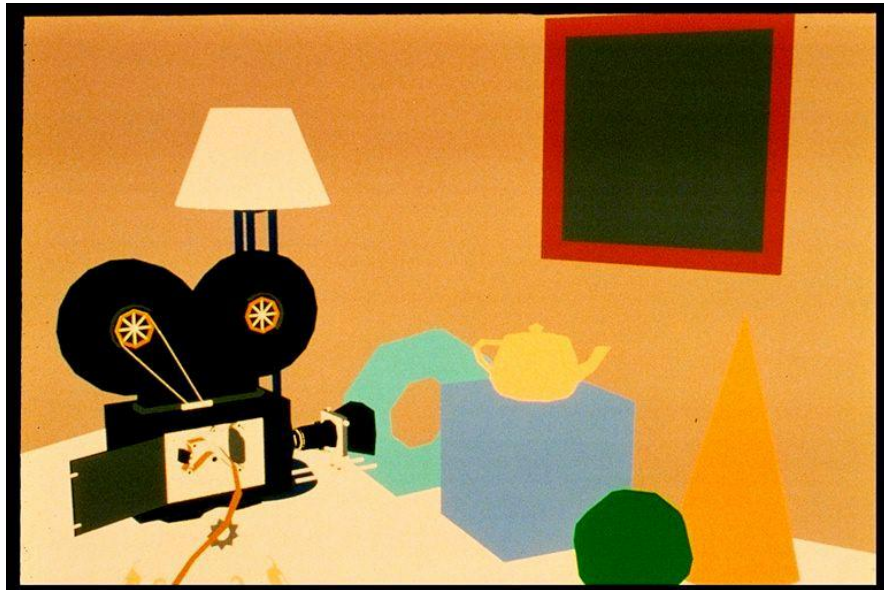
Gouraud Σκίαση: LERP σε Τρίγωνο

Για να προσδιορίσουμε την ένταση (χρώμα) του σημείου p στο τρίγωνο κάνουμε τα εξής:

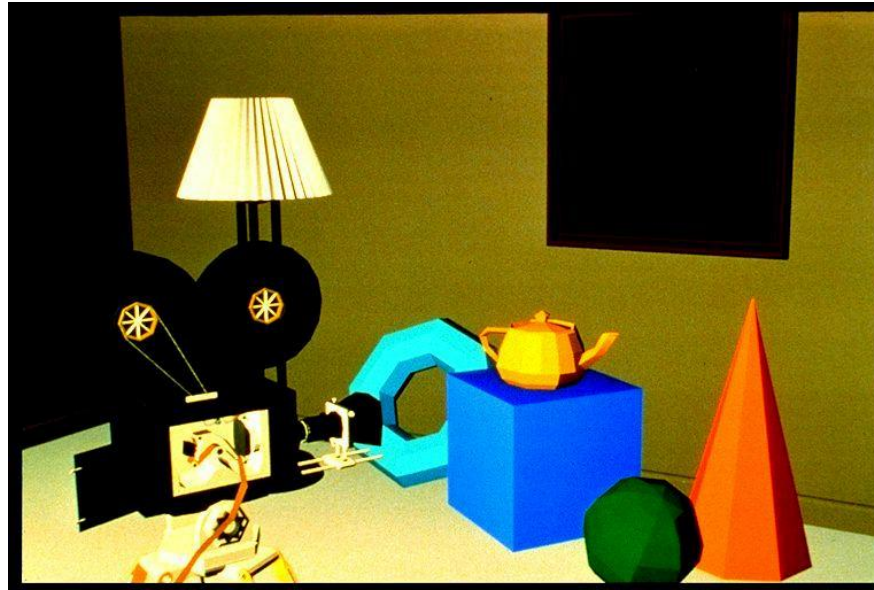
- Προσδιορίζουμε την ένταση του 4 παρεμβάλλοντας γραμμικά μεταξύ 1 και 2
- Προσδιορίζουμε την ένταση του 5 παρεμβάλλοντας γραμμικά μεταξύ 2 και 3
- Προσδιορίζουμε την ένταση του p με γραμμική παρεμβολή μεταξύ 4 και 5



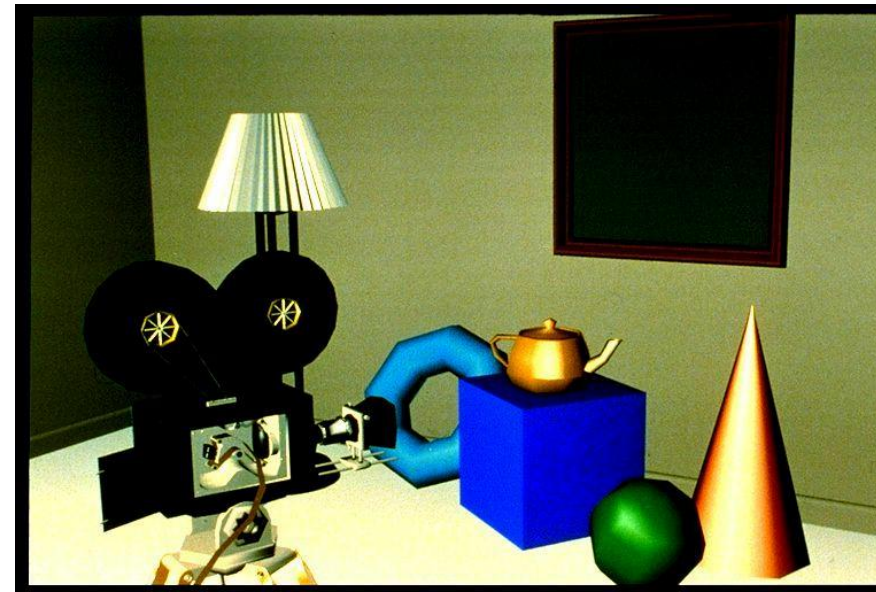
Σύγκριση



Χωρίς φωτισμό



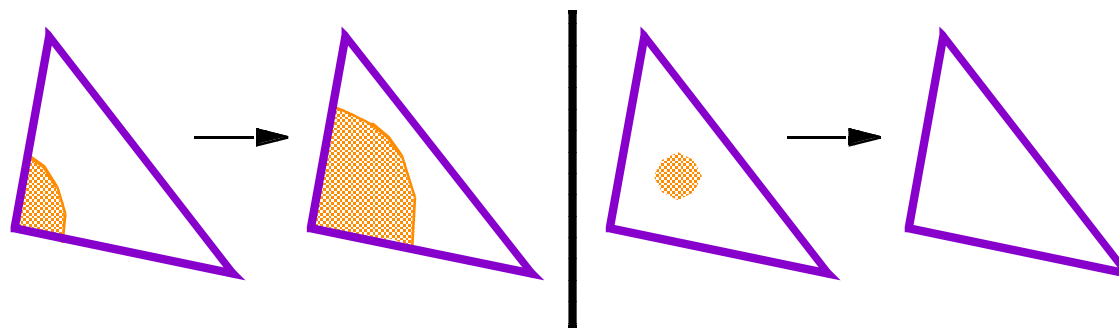
Επίπεδη Σκίαση



Gouraud Σκίαση

Phong Σκίαση

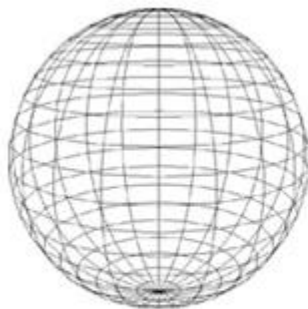
Η Gouraud σκίαση δεν χειρίζεται σωστά το κατοπτρικό φως, ιδιαίτερα όταν το α (συντελεστής λάμψης) είναι μεγάλος (μικρός σε έκταση τονισμός).



Λόγος: παρεμβολή στα χρώματα

Λύση: (Phong Σκίαση)

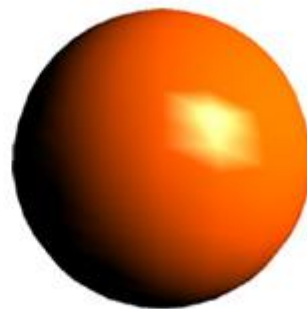
1. Υπολογισμός κάθετων διανυσμάτων στις κορυφές.
2. Παρεμβολή των **καθέτων** κατά μήκος των ακμών και των εσωτερικών σημείων.
3. Υπολογισμός φωτισμού ανά pixel.



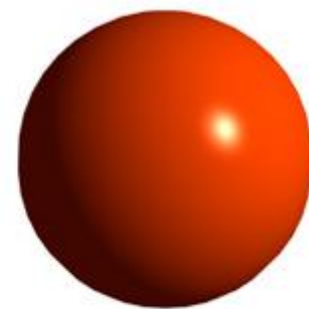
(a)



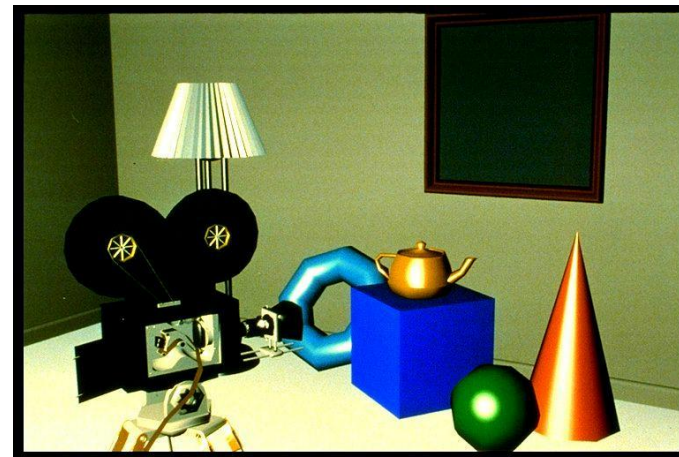
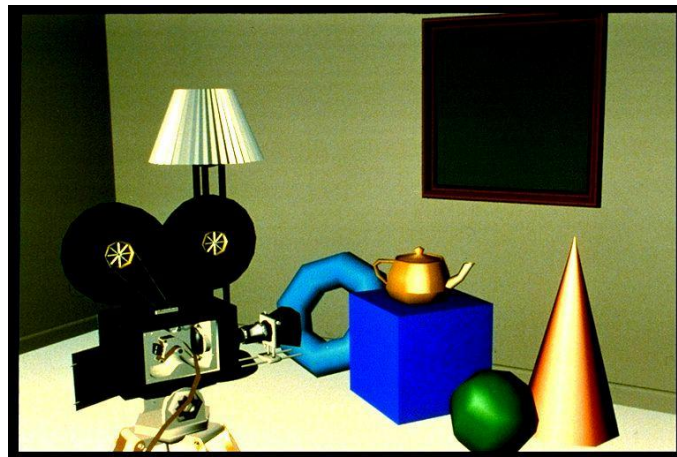
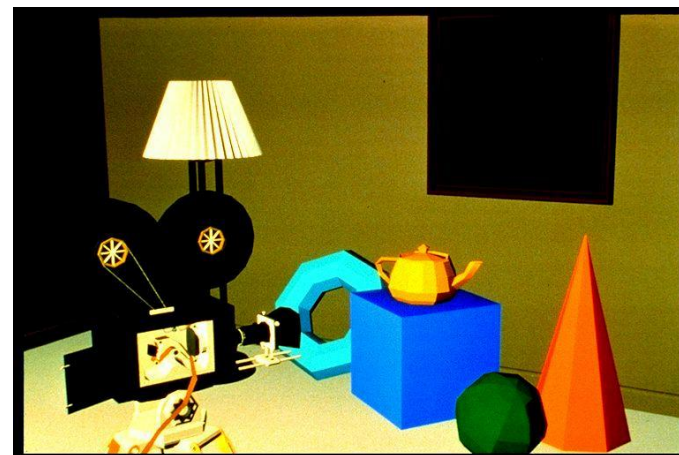
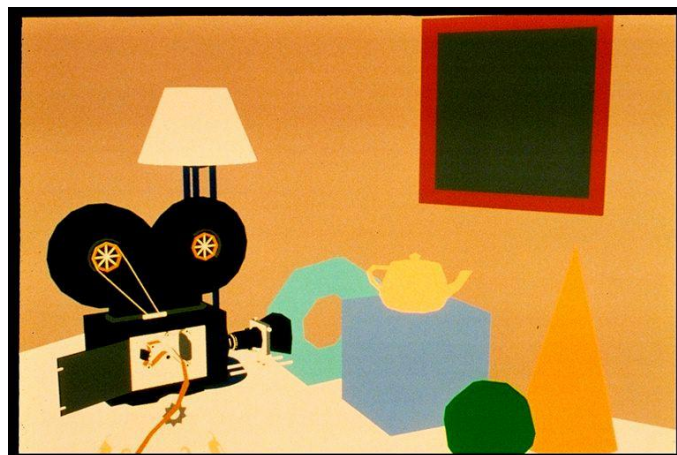
(b)



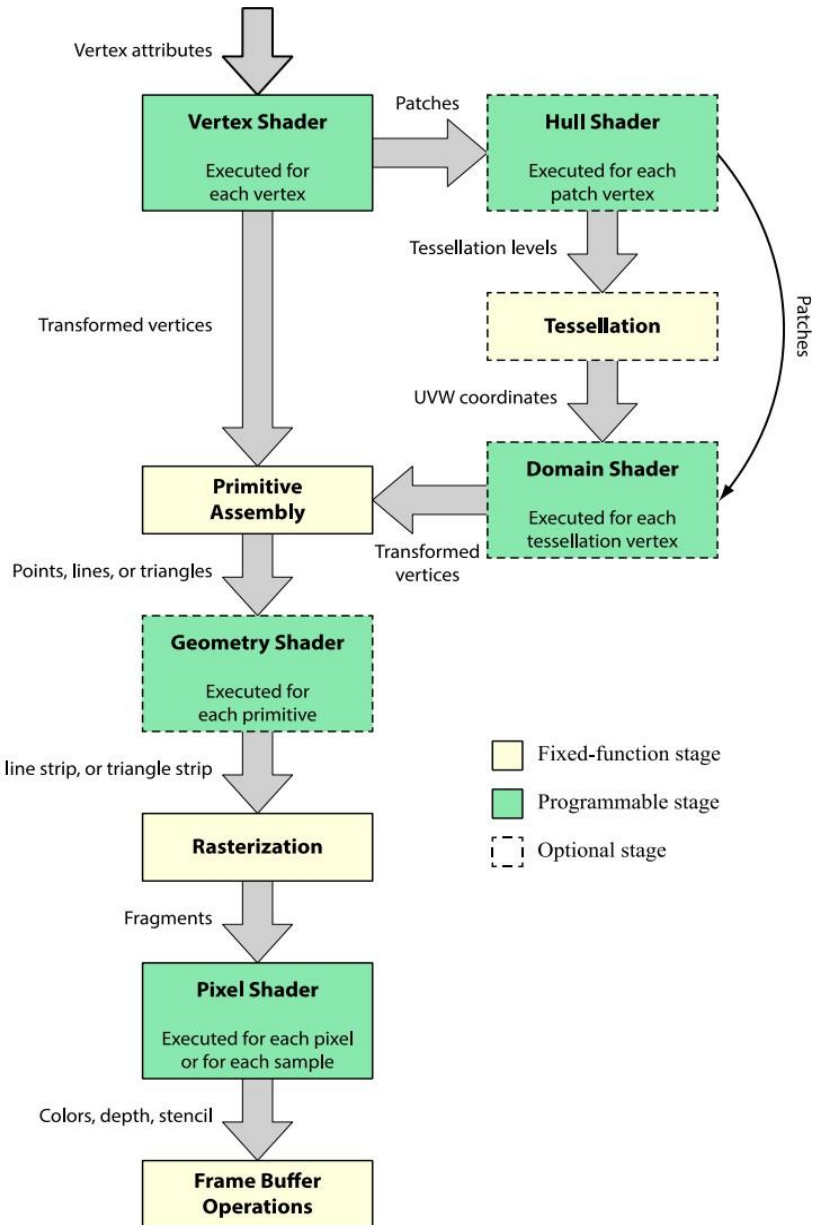
(c)



(d)

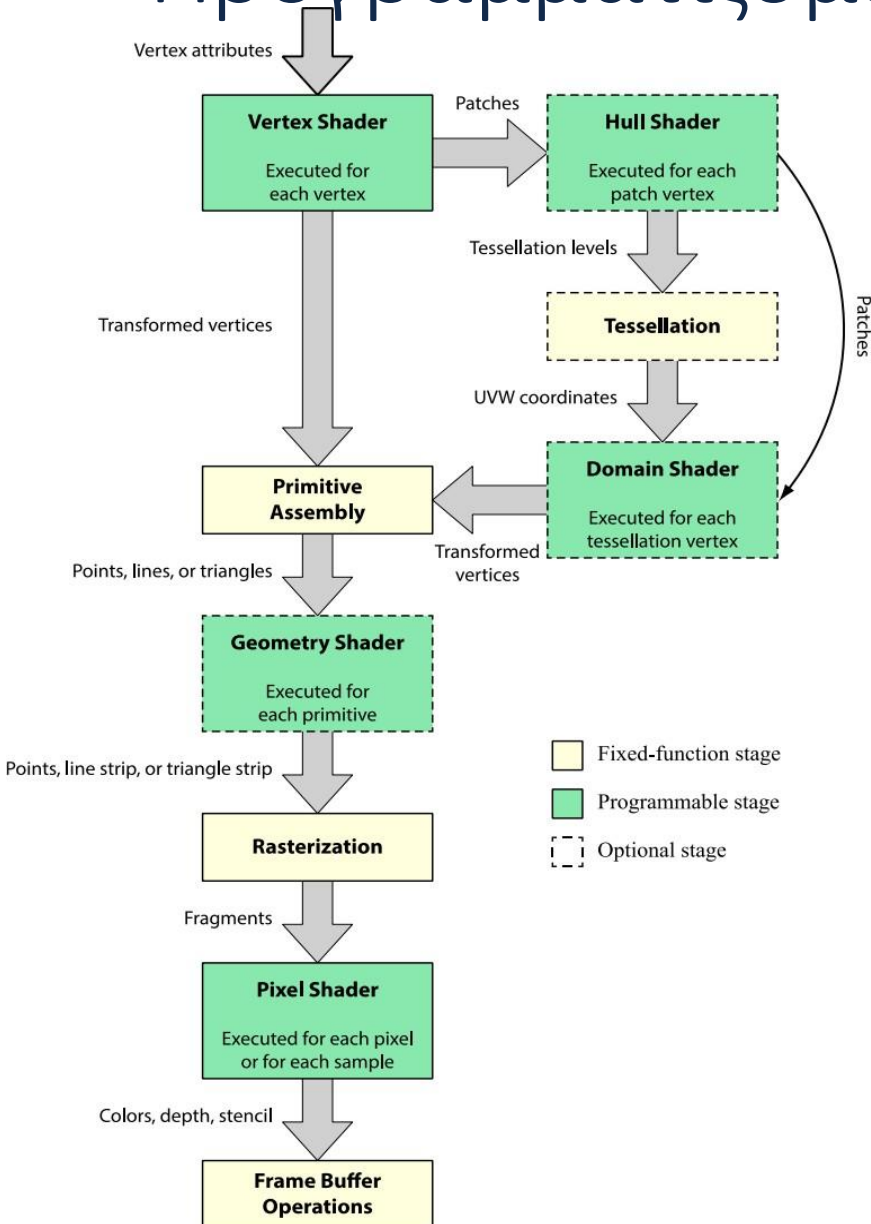


Προγραμματιζόμενοι Shaders



- Οι σύγχρονες GPU εισήγαγαν την ιδέα των shader προγραμμάτων
- Η λέξη shader έχει καταλήξει να σημαίνει οποιαδήποτε από μια σειρά προγραμματιζόμενων μονάδων GPU
- Η αρχιτεκτονική στα αριστερά είναι μόνο ένα παράδειγμα
- Δεν χρησιμοποιούνται όλοι αυτοί οι shader σε μηχανές παιχνιδιών
- Δεν υποστηρίζουν όλα τα API απόδοσης γραφικών όλους αυτούς τους τύπους shader
- Υπάρχουν και άλλοι τύποι shader
 - Tessellation Shaders
 - Mesh Shaders
 - Ray-Tracing Shaders
 - Compute Shaders
- Αφού οι shader είναι προγραμματιζόμενοι μπορούν να χρησιμοποιηθούν και για άλλους σκοπούς (GPGPU)

Προγραμματιζόμενοι Shaders



Shader Κορυφών

- Κάθε κορυφή στη γεωμετρία της σκηνής τίθεται προς επεξεργασία από ένα πρόγραμμα σκίασης κορυφής
- Συνήθως μόνο γεωμετρικοί μετασχηματισμοί που σχετίζονται με animation, προβολή, κ.λπ.
- Το πρόγραμμα εκτελείται πανομοιότυπα σε πολλούς πυρήνες κορυφών GPU

Pixel Shader (ή Fragment Shader)

- Εκτελείται για κάθε pixel (Rasterization)
- Συνήθως υπολογίζει ένα χρώμα για αυτό το pixel ← *αυτή είναι η πραγματική σκίαση*
- Το πρόγραμμα εκτελείται πανομοιότυπα σε πολλούς πυρήνες κορυφών GPU

Το Στάδιο της Ραστεροποίησης

Ραστεροποίηση

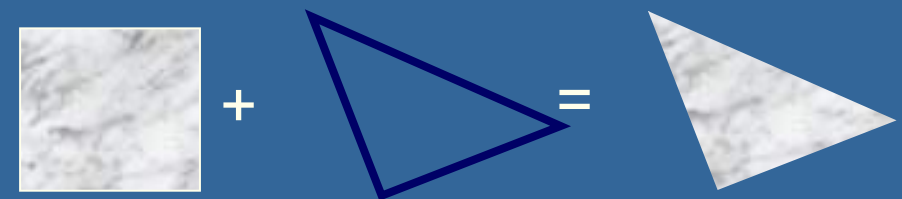
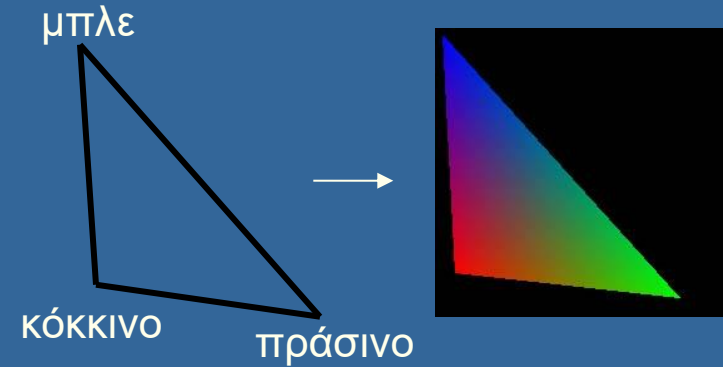


- Σάρωση και μετατροπή
 - Μάθετε ποια pixel βρίσκονται μέσα στο τρίγωνο
- Υφή
 - «Επικολλήστε» εικόνες σε τρίγωνα
- Παρεμβολή στο τρίγωνο
- Z-buffering
 - Βεβαιωθείτε ότι αυτό που είναι ορατό από την κάμερα εμφανίζεται
- Double buffering
- Antialiasing
- Και περισσότερα...

RASTERIZER: Σάρωση και Μετατροπή

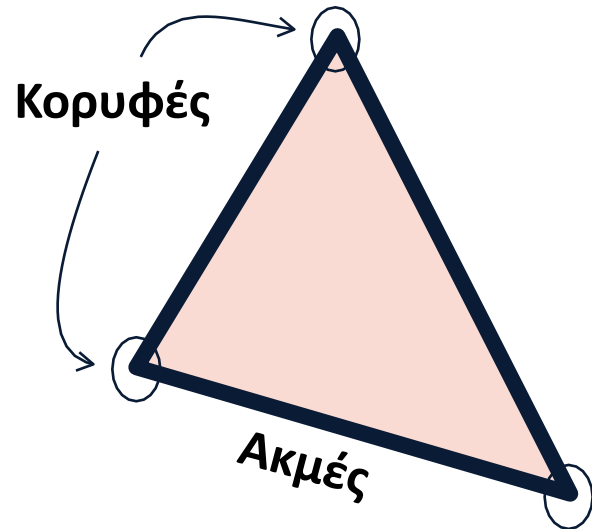
Είσοδος: Οι κορυφές τριγώνων από το γεωμετρικό στάδιο

- Ανιχνεύστε τα pixel μέσα στο τρίγωνο
 - Ή τα pixel σε μια γραμμή, ή σε ένα σημείο
- Εκτέλεση λειτουργιών ανά pixel:
 - Παρεμβολή
 - Υφή
 - Z-buffering
 - Και ό,τι άλλο χρειαστεί 😊



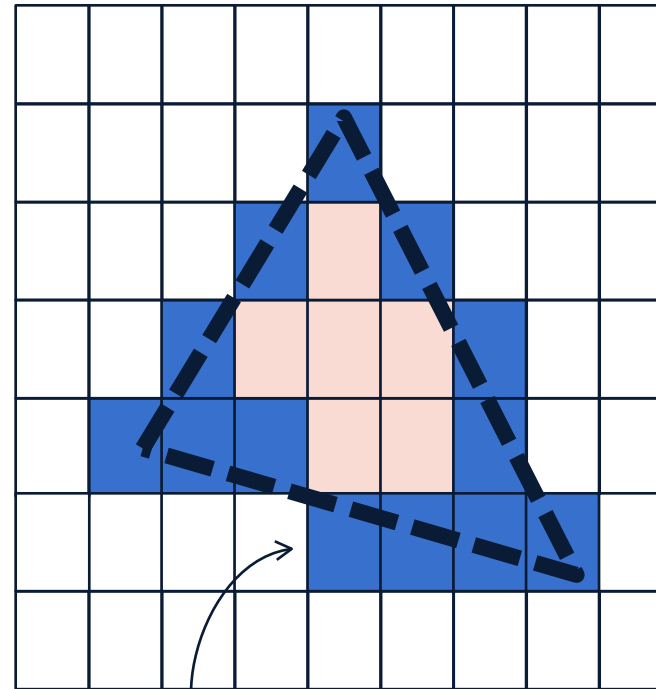
Rasterization

Θεμελιώδη γεωμετρικά στοιχεία



Δημιουργία μίας raster εικόνας
από μία διανυσματική περιγραφή

Pixels



Αλλοίωση – Aliasing

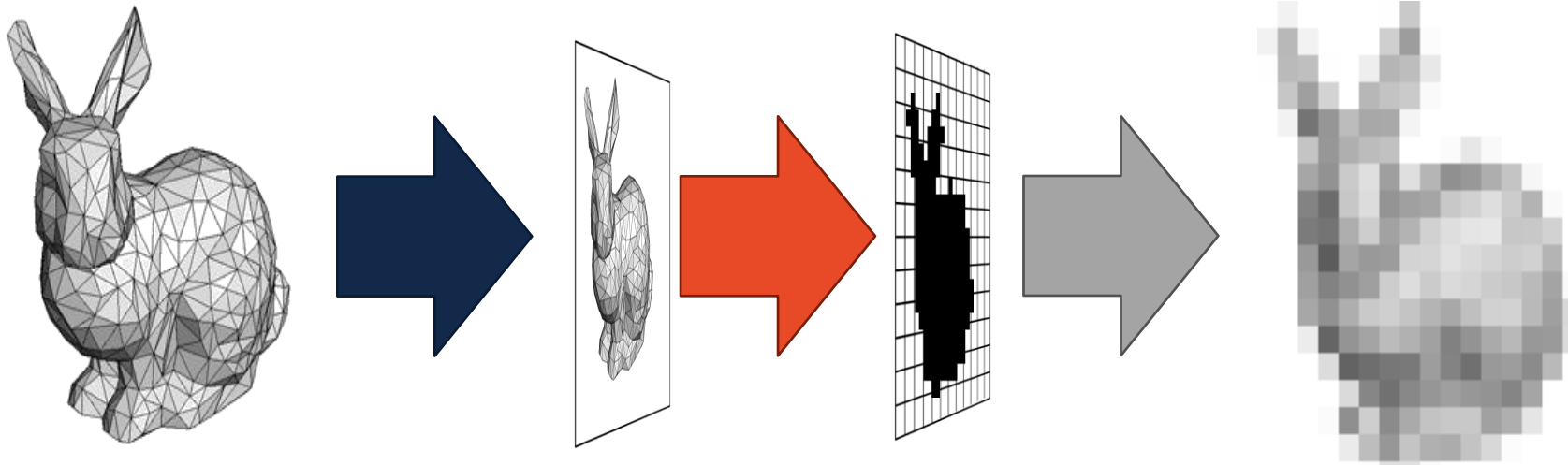


Ροή Εργασίας για 3d Γραφικά

Επεξεργασία
Κορυφών

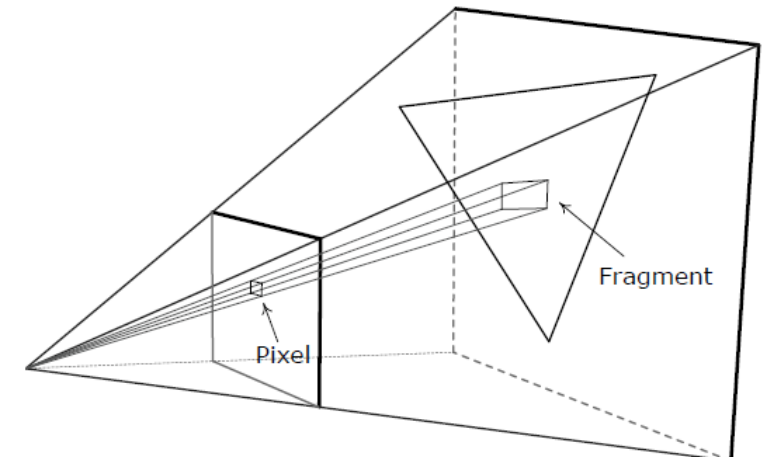
Rasterization

Επεξεργασία
Θραυσμάτων



Θραύσμα (Fragment)

- αντιπροσωπεύει μια μικρή περιοχή της επιφάνειας του τριγώνου που αντιστοιχεί σε ένα μόνο pixel στο οθόνη.
- στην περίπτωση της τεχνικής multisample antialiasing, ένα θραύσμα αντιστοιχεί σε ένα τμήμα ενός pixel
- Εφαρμόζονται δοκιμές ορατότητας



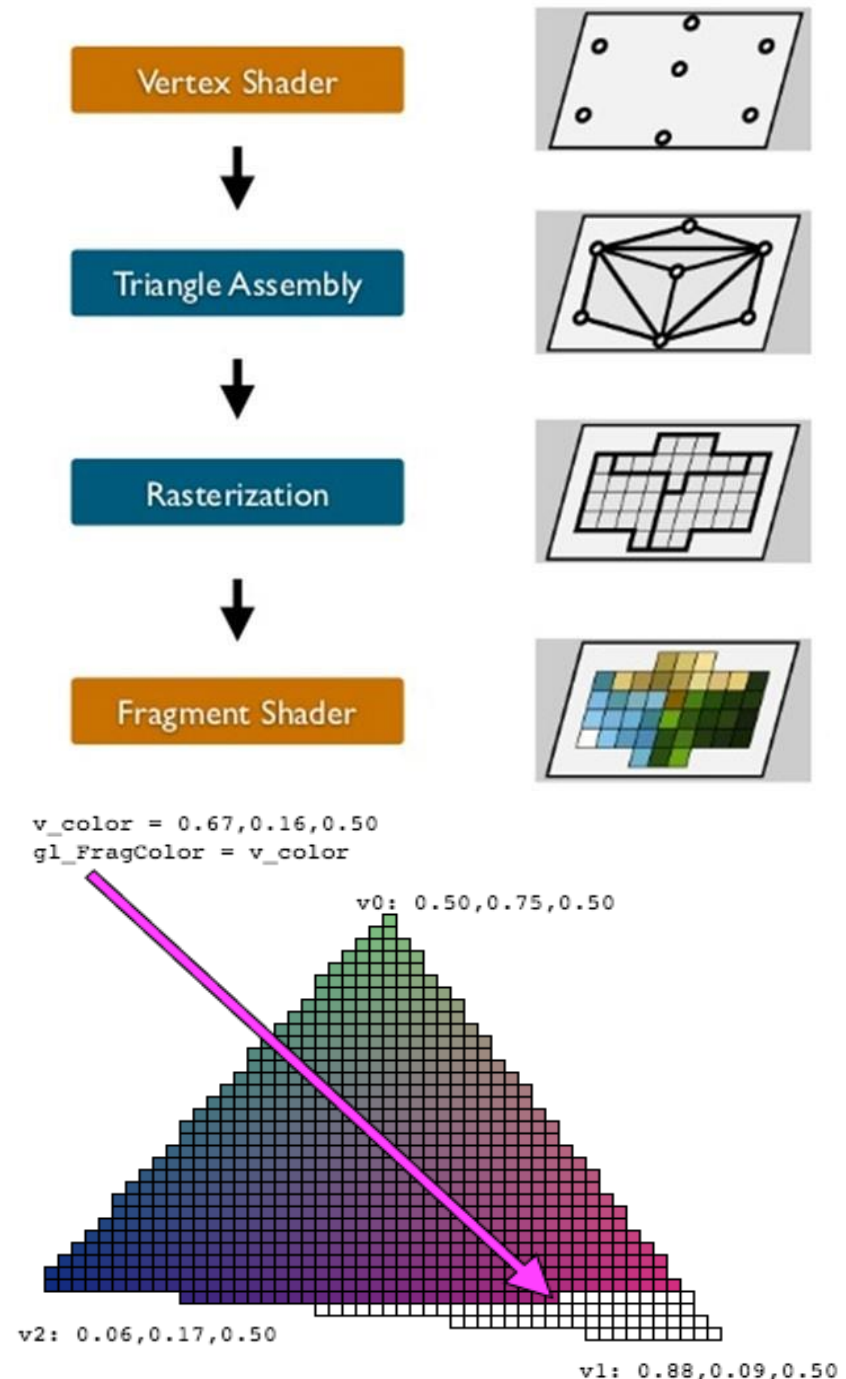
Rasterization

Παράγει ένα σύνολο θραυσμάτων για κάθε τρίγωνο

Τα θραύσματα είναι εν δυνάμει pixels

- Αποθηκεύονται στον frame buffer
- Ιδιότητες χρώματος και βάθους

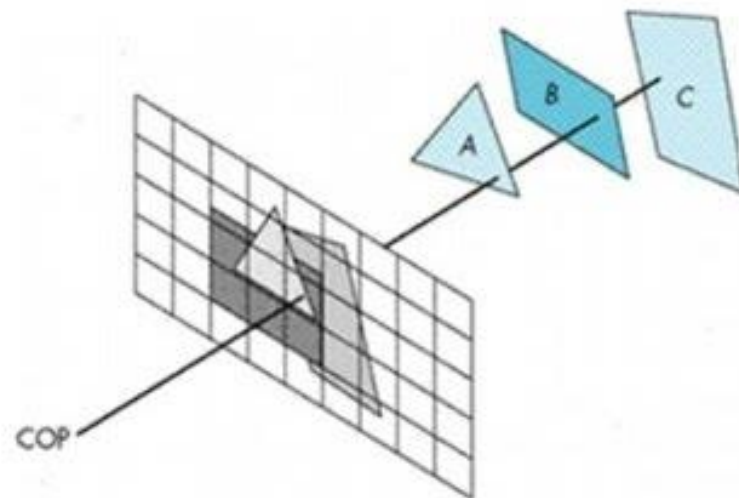
Εφαρμόζουμε παρεμβολή στα χαρακτηριστικά των κορυφών μεταξύ των θραυσμάτων



Πιο κοντά στην οθόνη...

Buffering

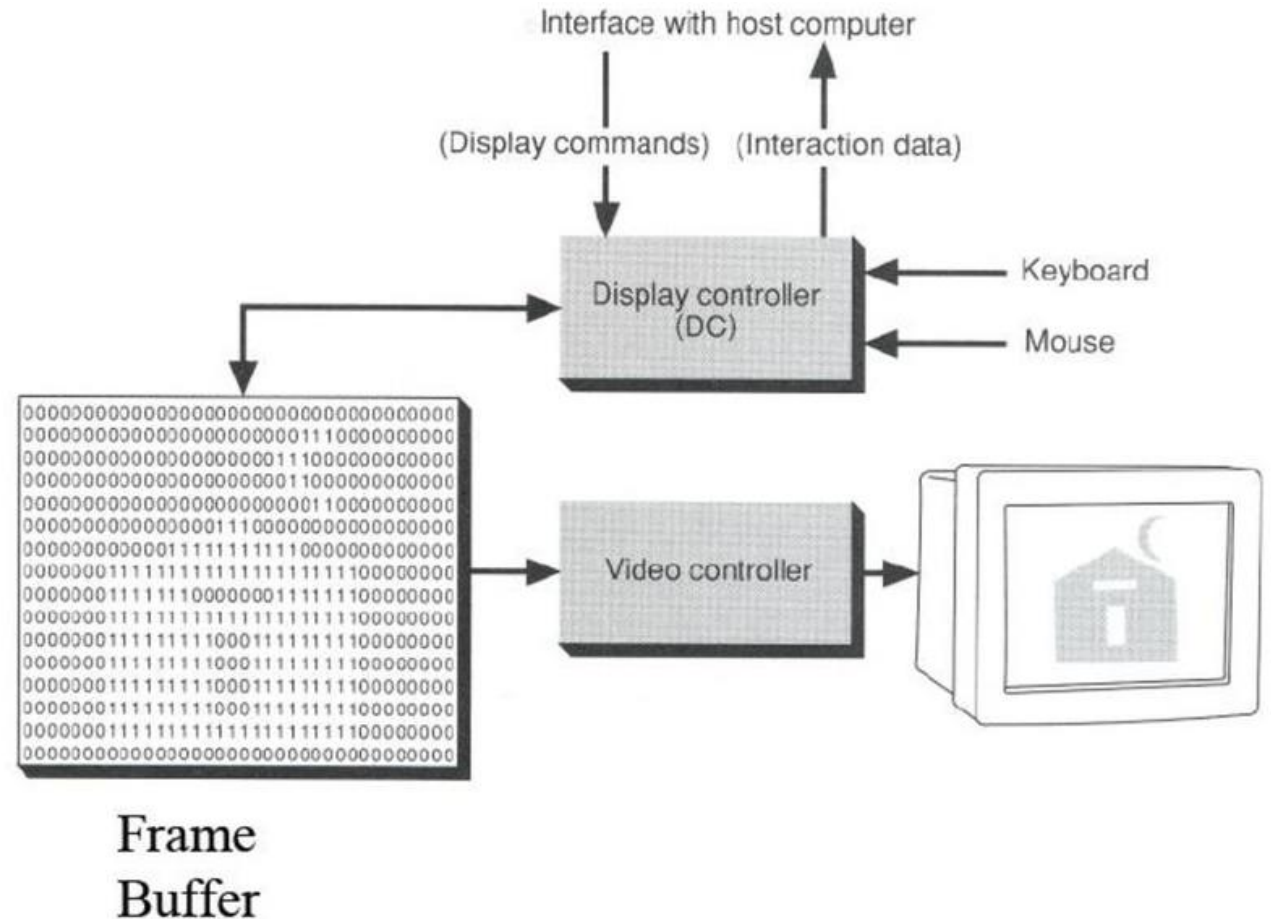
Z-BUFFER ALGORITHM



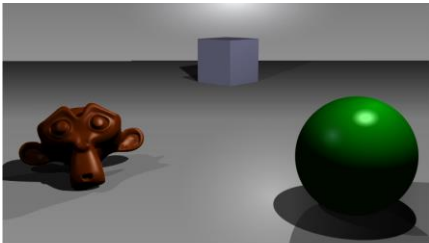
Frame Buffer

Η τελική εικόνα αποθηκεύεται στον Frame Buffer.

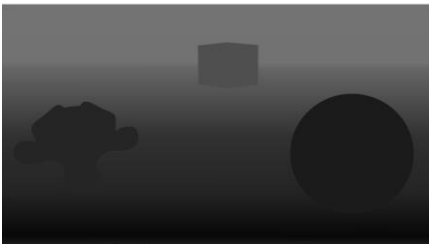
- Το υλικό απεικόνισης σαρώνει τον frame buffer για να απεικονίσει τα περιεχόμενά του.
- Συνήθως έχουμε τουλάχιστον δύο buffers: Ενώ ο ένας σαρώνεται ο άλλος χρησιμοποιείται για να προετοιμαστεί η επόμενη σκηνή – *double buffering*.
 - Μερικές μηχανές μπορεί να έχουν τρεις buffers (*triple buffering*)



Απόκρυψη (Occlusion) – Depth Buffer



A simple three-dimensional scene

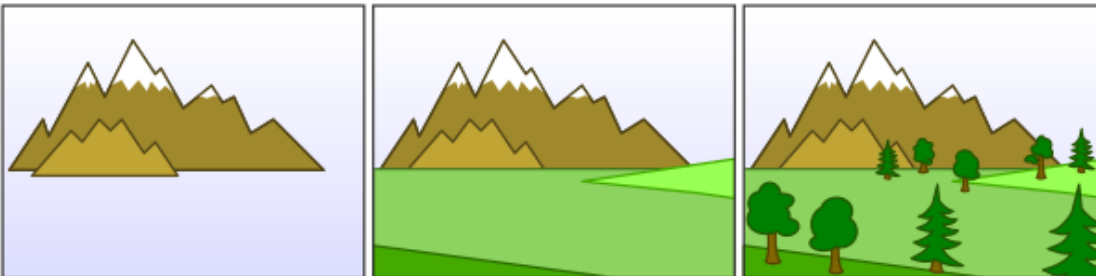


Z-buffer representation



Όταν δύο τρίγωνα επικαλύπτονται στο χώρο της οθόνης τότε θα πρέπει το τρίγωνο που είναι πιο κοντά στην κάμερα να ζωγραφιστεί

- Ο αλγόριθμος του ζωγράφου ☺ (painter's algorithm)
 - Ζωγράφισε τα τρίγωνα από πίσω προς τα εμπρός
 - Δεν δουλεύει πάντα
- **z-buffering:**
 - Κράτα ένα buffer με το βάθος (**z**) του θραύσματος για κάθε pixel που αντιστοιχεί στο frame buffer.
 - Έλεγχος όταν γράφεται ένα θραύσμα για το βάθος του
 - Μη-γραμμική απεικόνιση από τον εικονικό χώρο



ΧΑΡΤΟΓΡΆΦΗΣΗ

Texture, Bump/Normal,
Environment Mapping



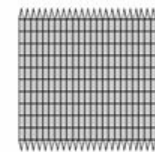
3-D Model



$$p = (x, y, z)$$



UV Map



$$p = (u, v)$$



Texture



Τα Όρια της Γεωμετρικής Μοντελοποίησης

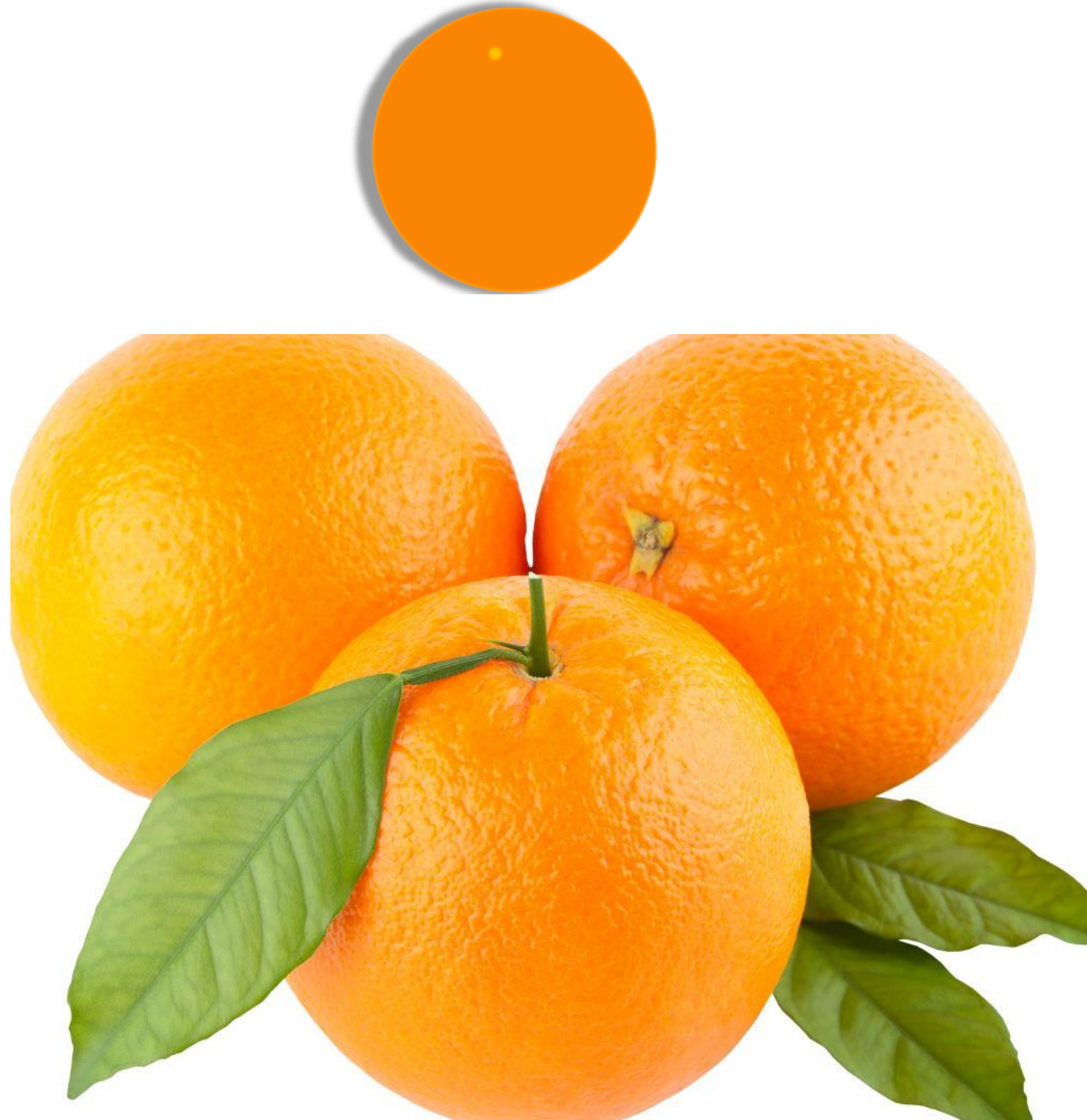
Παρόλου που οι σύγχρονες κάρτες γραφικών μπορούν να αποδώσουν δεκάδες εκατομμύρια πολύγωνα, δεν φτάνουν για φαινόμενα όπως:

- Σύννεφα
- Γρασίδι
- Έδαφος
- Δέρμα



Μοντελοποιώντας Πορτοκάλια

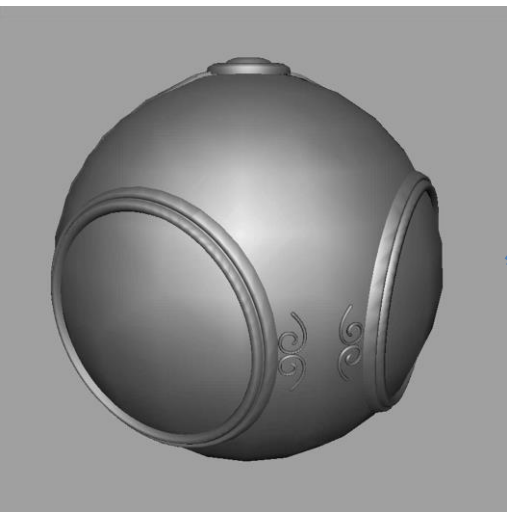
- Χρήση μίας σφαίρας σε πορτοκαλί χρώμα
 - Πολύ φτωχό αποτέλεσμα
- Αντί για σφαίρα χρήση πιο περίπλοκου σχήματος
 - Δεν περιέχει τα χαρακτηριστικά της επιφάνειας (ανομοιομορφίες)
 - Χρειάζεται πολλά πολύγωνα για να μοντελοποιηθούν αυτές οι ανομοιομορφίες
- Χρήση φωτογραφίας πραγματικού πορτοκαλιού και «επικόλληση» σε απλό γεωμετρικό μοντέλο
 - Αυτή η διαδικασία είναι γνωστή ως **χαρτογράφηση υφής (texture mapping)**
- Η επιφάνεια που προκύπτει θα είναι λεία
 - Απαιτείται αλλαγή τοπικού σχήματος
 - Χαρτογράφηση ανομοιομορφιών



Διάφοροι Τύποι Χαρτογράφησης

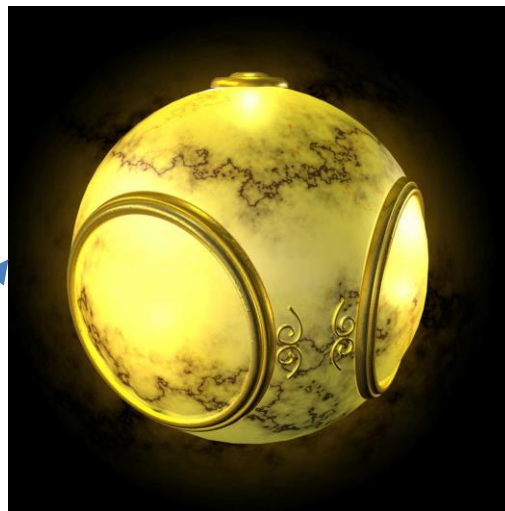
- Χαρτογράφηση Υφής
 - Χρήση εικόνων για να γεμίσουν οι επιφάνειες των πολυγώνων
- Χαρτογράφηση Περιβάλλοντος – Ανακλάσεων (environmental/reflection mapping)
 - Χρήση φωτογραφίας του περιβάλλοντος για υφή
 - Επιτρέπει τη προσομοίωση ανακλαστικών επιφανειών
- Χαρτογράφηση Ανομοιομορφιών (bump mapping)
 - Εξομοιώνει τα διαφορετικά κάθετα διανύσματα κατά τη διάρκεια της διεργασίας απόδοσης

Παράδειγμα

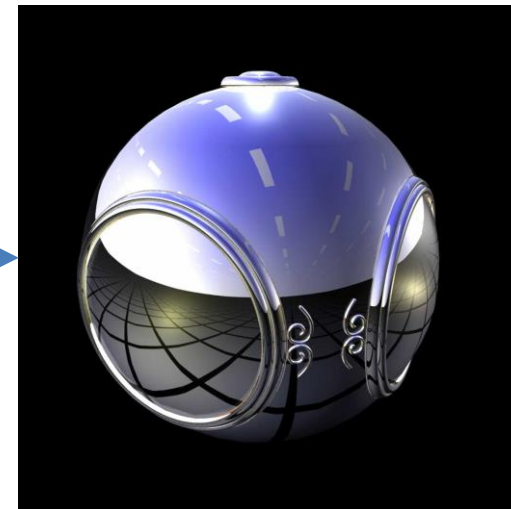


Γεωμετρικό Μοντέλο

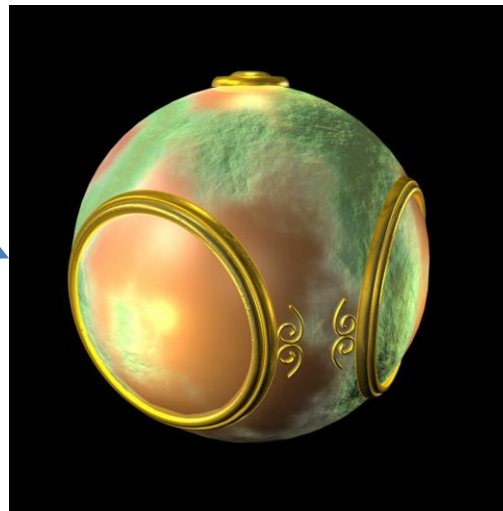
Χαρτογράφηση Υφής



Χαρτογράφηση Περιβάλλοντος

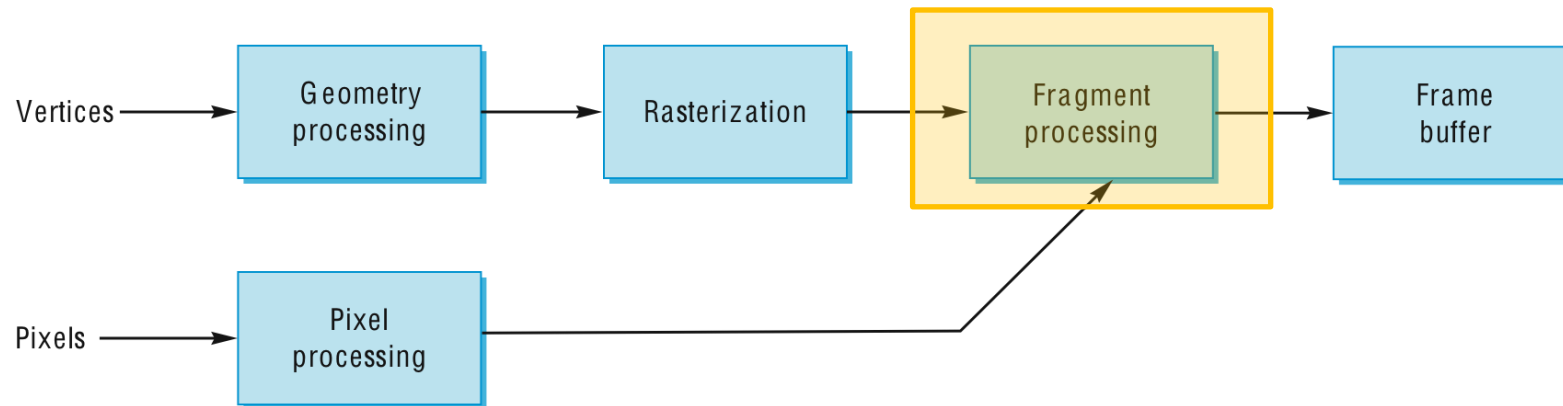


Χαρτογράφηση
Ανομοιομορφιών

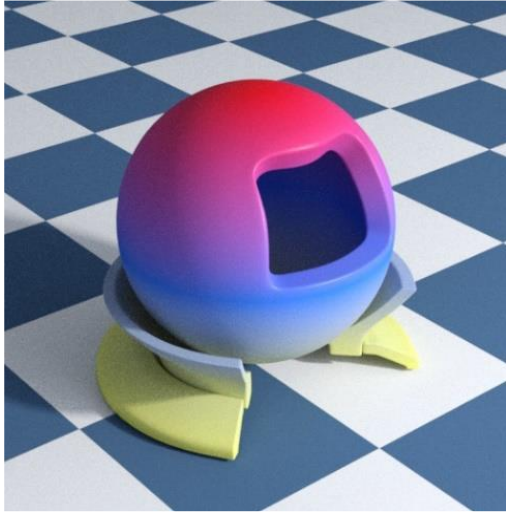


Πότε γίνεται η Χαρτογράφηση;

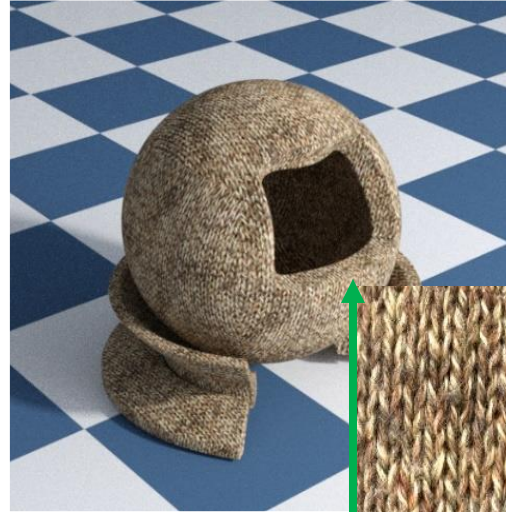
Αυτές οι τεχνικές υλοποιούνται κατά το τέλος της απόδοσης γραφικών
– Αποδοτικό λόγω αποκοπής πολυγώνων



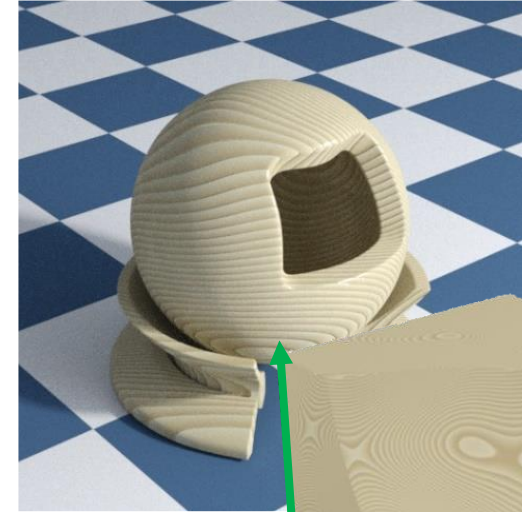
Χώρος Εικόνας Υφής

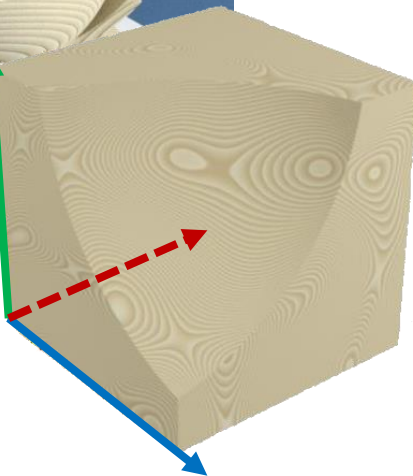


1d 



2d 



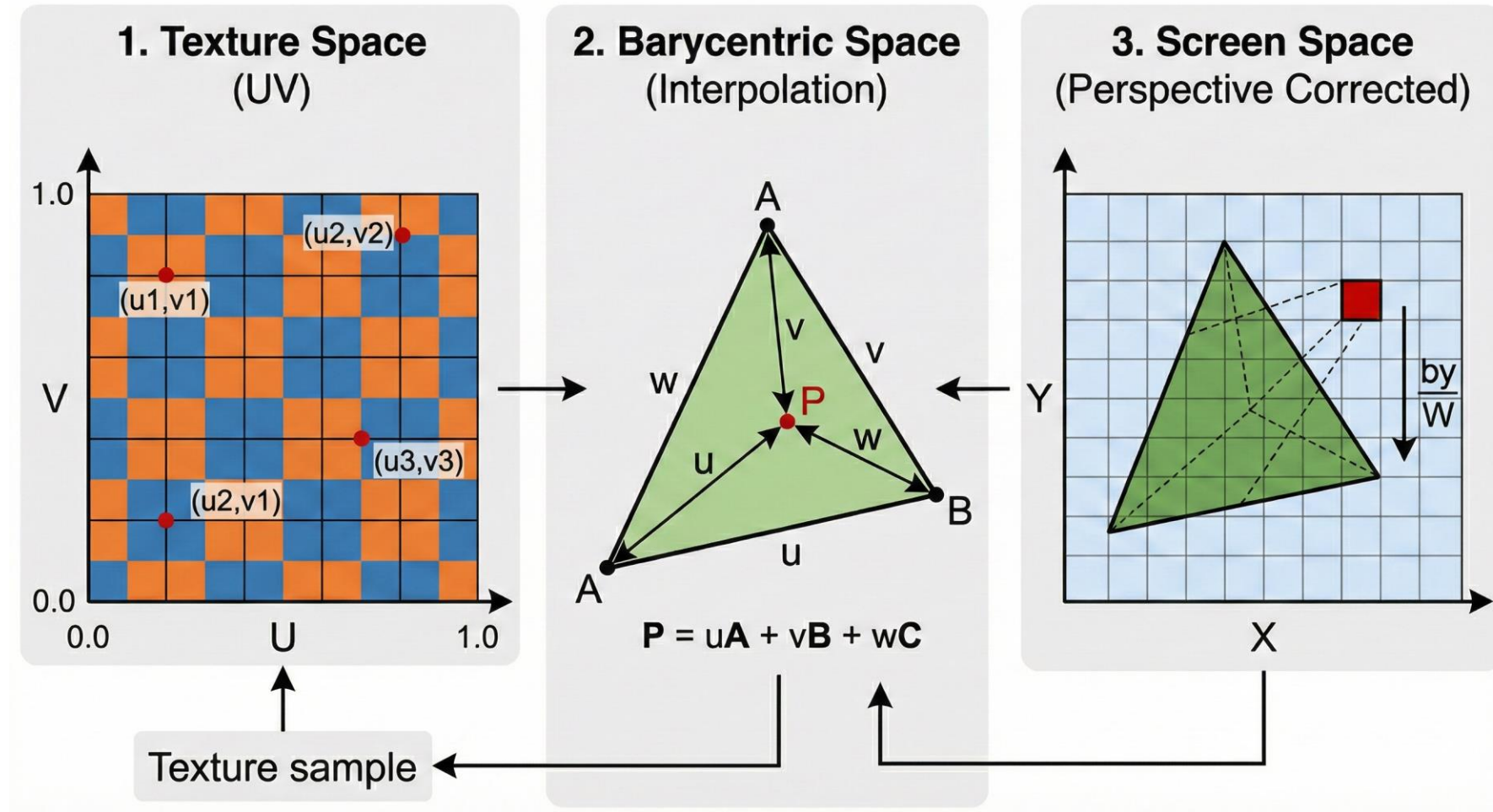
3d 

- Μία 1/2/3d εικόνα υφής ορίζεται σε έναν παραμετρικό χώρο υφής
- Ο παραμετρικός χώρος είναι συνήθως κανονικοποιημένος στο $[0,1]$.
 - Για παράδειγμα, σε 2d χρησιμοποιούμε δύο μεταβλητές u, v .

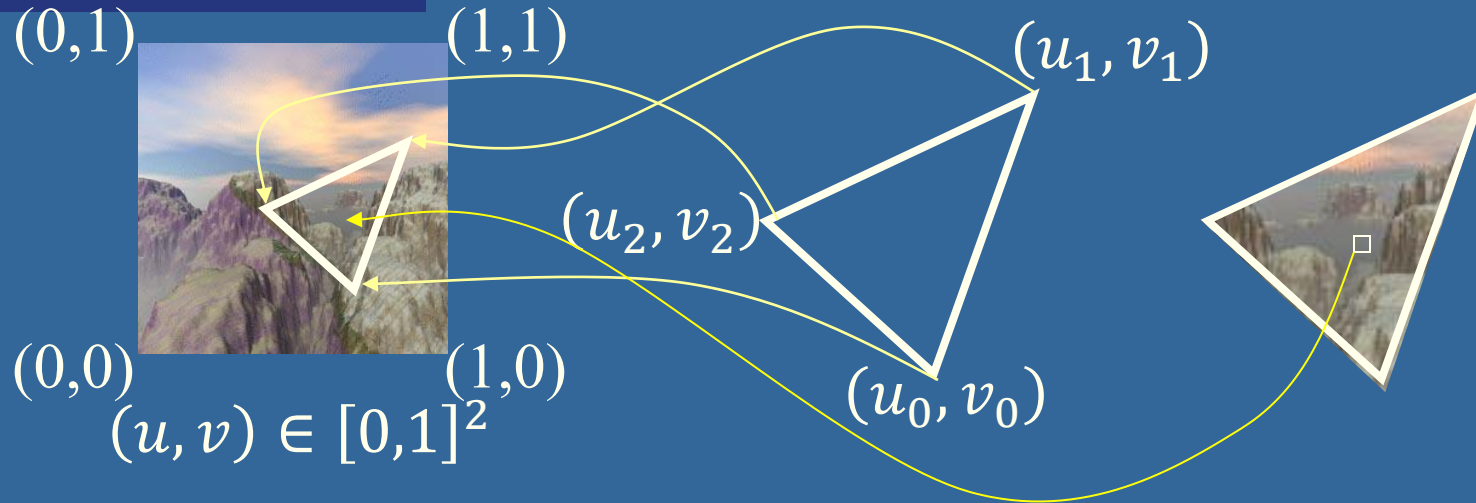
Πόσο Απλό είναι;

Η ιδέα είναι απλή: απλά αντιστοίχισε μια εικόνα σε μία επιφάνεια – αλλά εμπλέκονται 3 με 4 συστήματα συντεταγμένων

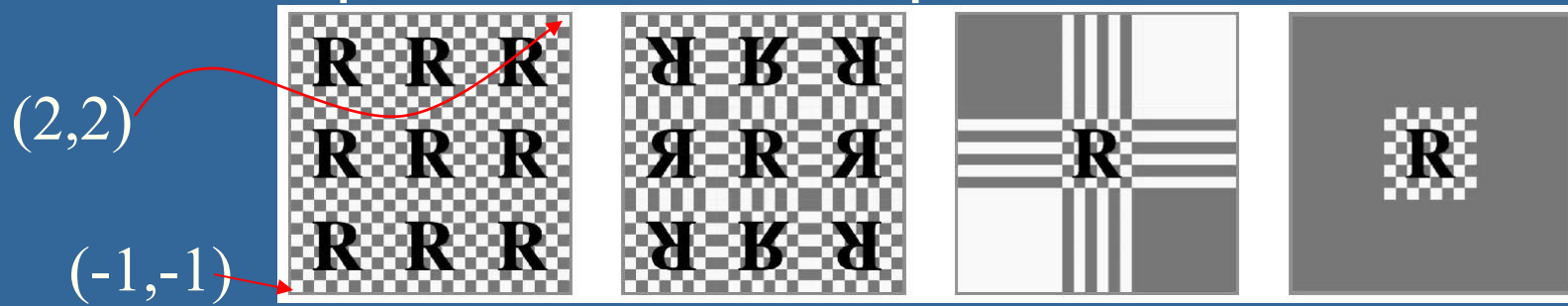
- Στα πλαίσια του μαθήματος δεν μας ενδιαφέρει πως γίνεται ακριβώς



Συντεταγμένες Υφής



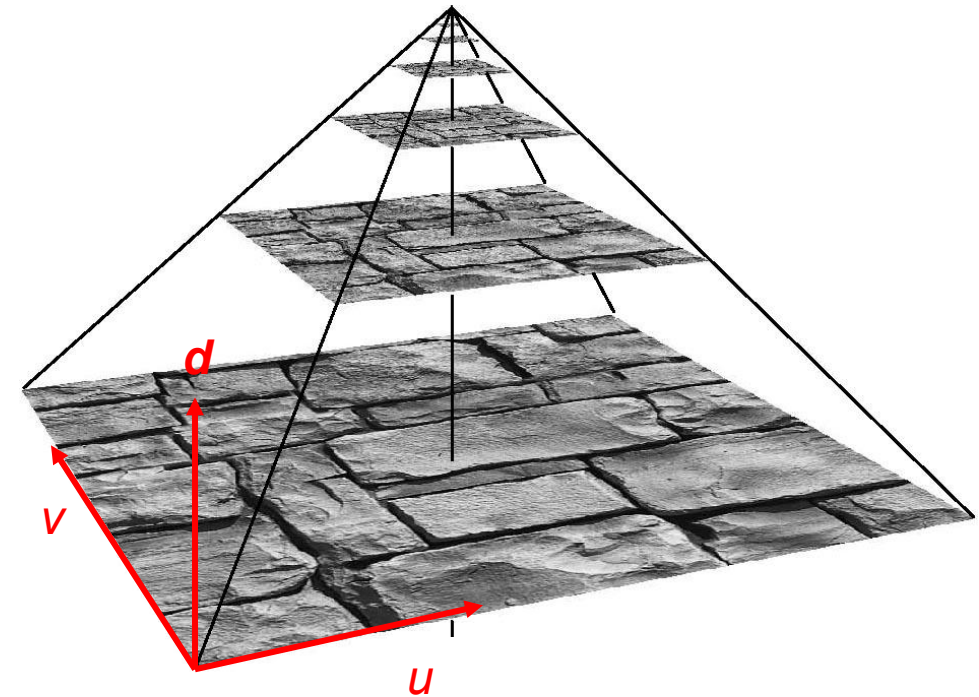
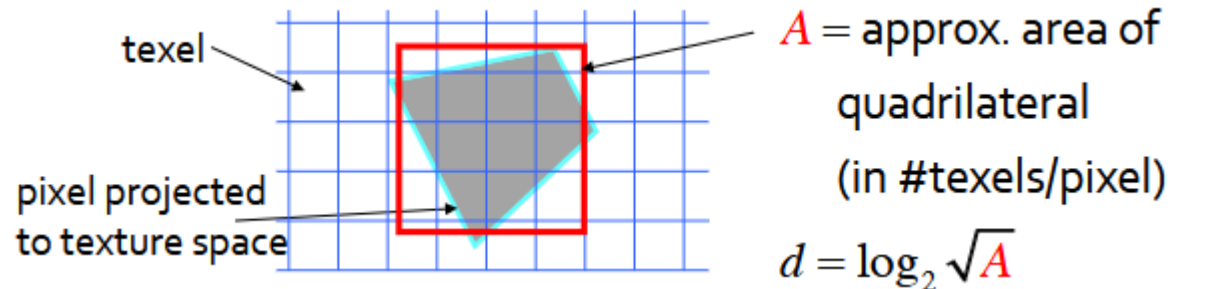
- Τί θα γίνει αν $(u, v) > 1$ ή $(u, v) < 0$;
- Επανάληψη υφών: χρήση μόνο κλασματικού μέρους, π.χ., $5.3 \rightarrow 0.3$
- Repeat, mirror, clamp, border:



Mipmapping: Μία Τεχνική για «LOD» σε Υφές

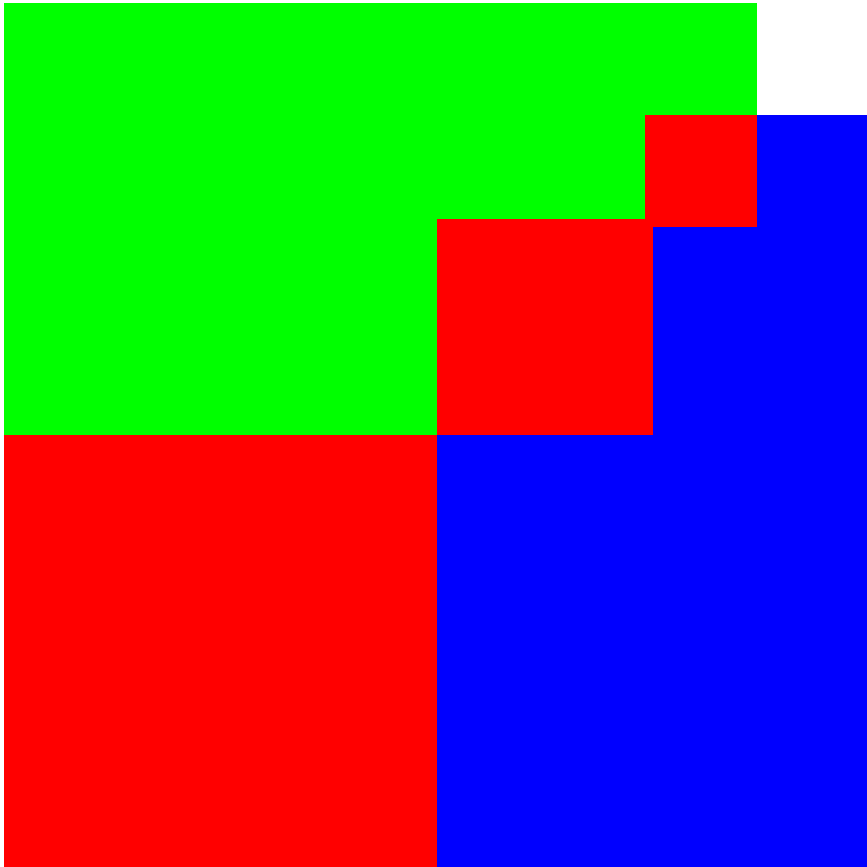
Πολλά texels αντιστοιχούν σε ένα pixel

- Πρέπει να πάρουμε μέσους όρους από αυτά τα texel για ένα pixel: **ακριβό**
- Προϋπολογισμός των υφών σε διαφορετικές αναλύσεις (δυνάμεις του 2).
- Υπολογισμός επιπέδου λεπτομέρειας από υπολογισμό texel ανά pixel

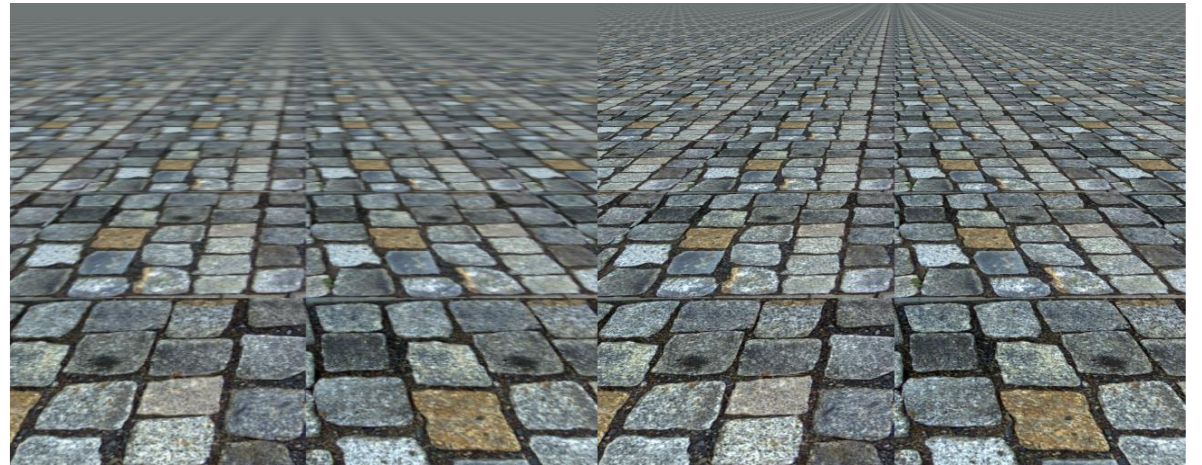


Mipmapping: Κάποια χαρακτηριστικά

- Χώρος: ~33% περισσότερος



- Καλύτερη απόδοση ιδιαίτερα για απομακρυσμένα αντικείμενα
- Μείωση αλλοιώσεων
- Θολότητα σε μακρινές υφές



Isotropic Filtering

Anisotropic Filtering

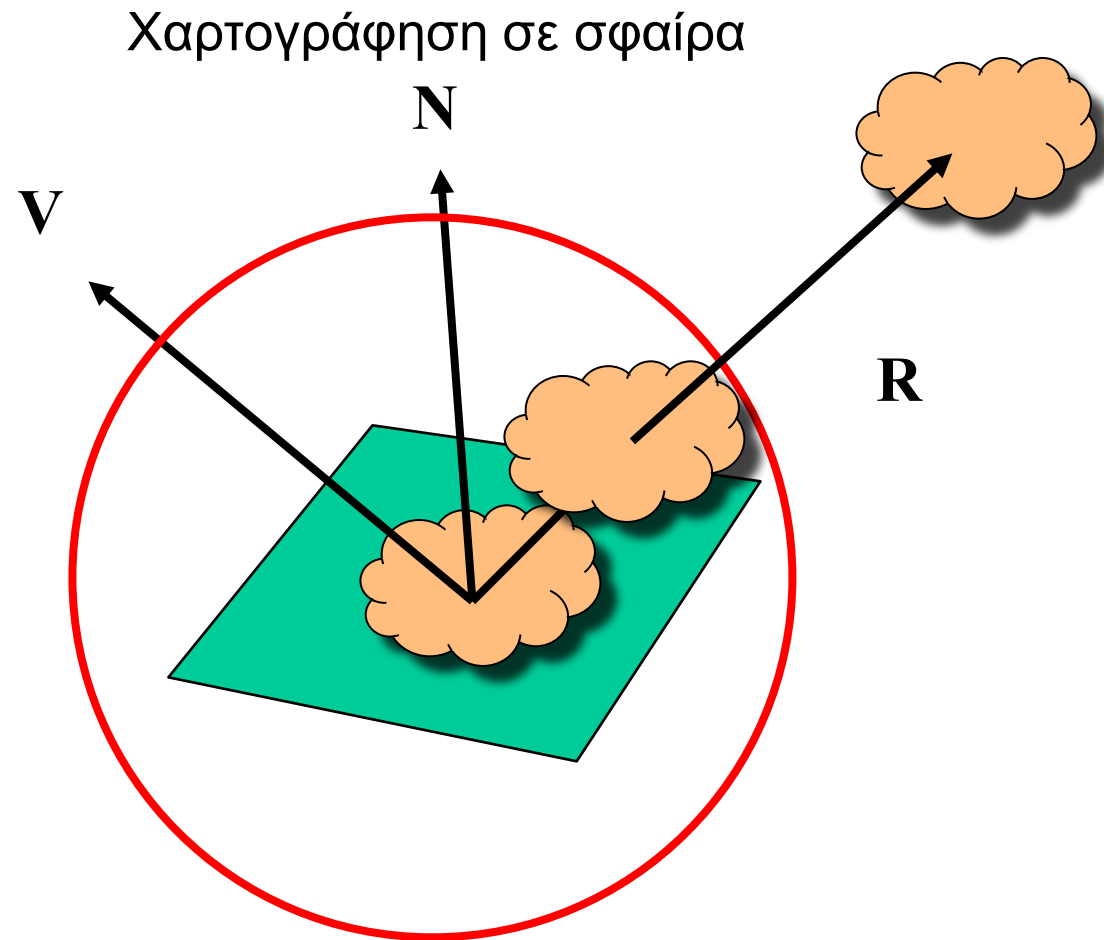
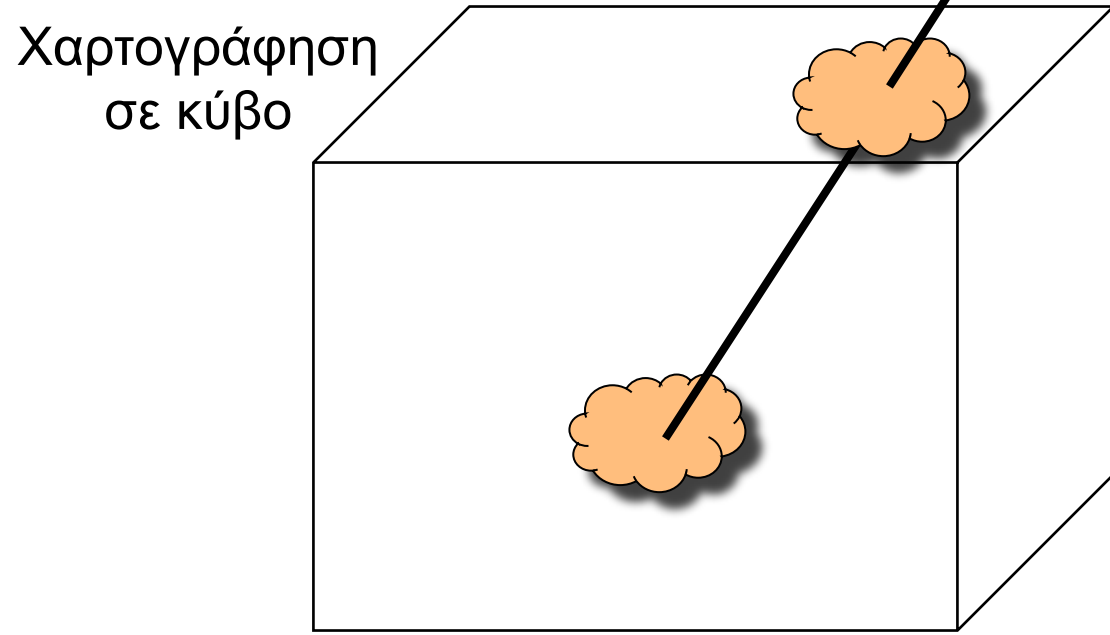
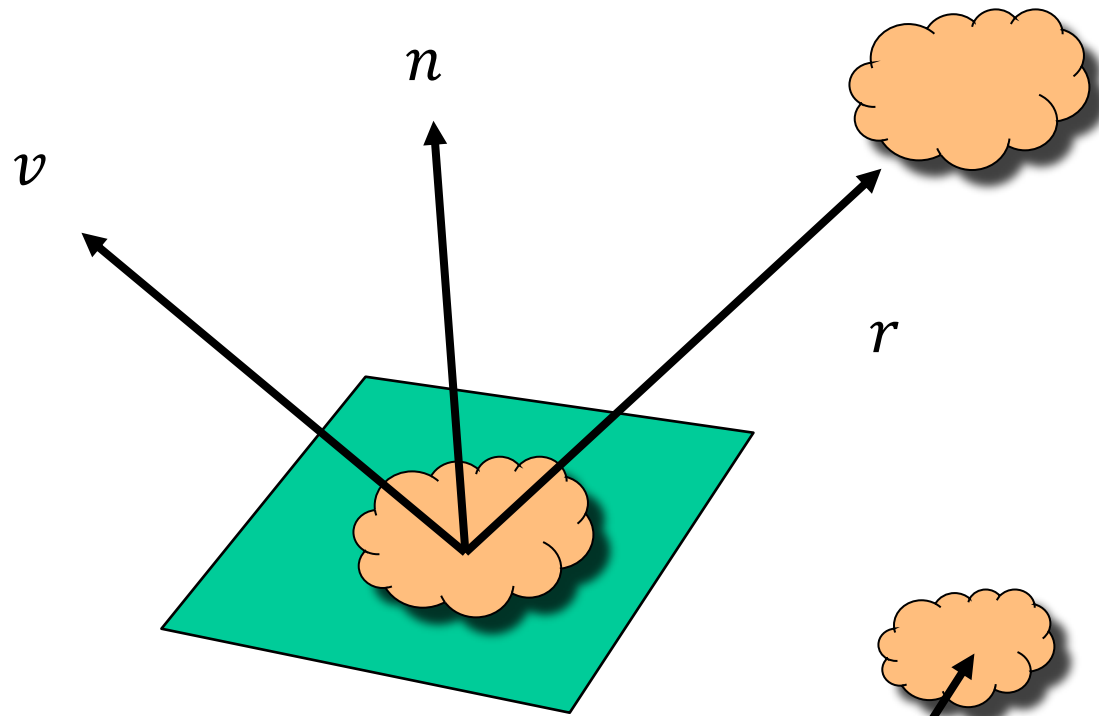
Χαρτογράφηση Περιβάλλοντος

Για επιφάνειες υψηλής ανακλαστικότητας χωρίς Ray-tracing που απαιτεί καθολικούς υπολογισμούς.

- Εικόνα του περιβάλλοντος μέσα από έναν ευρυγώνιο φακό
 - Μπορεί να είναι είτε μια πραγματική σαρωμένη εικόνα είτε μια συνθετική εικόνα
- Χρησιμοποιήστε αυτήν την υφή για να δημιουργήσετε έναν σφαιρικό χάρτη
- Εναλλακτικά μπορείτε να δημιουργήσετε ένα χάρτη κύβου

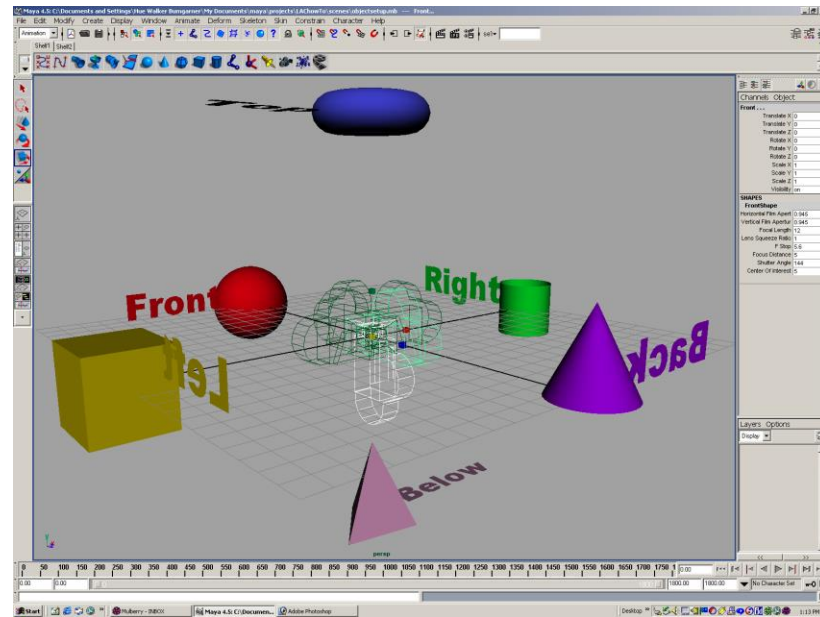
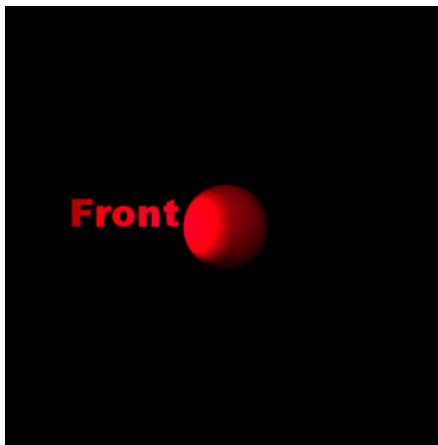


Ανάκλαση του Περιβάλλοντος

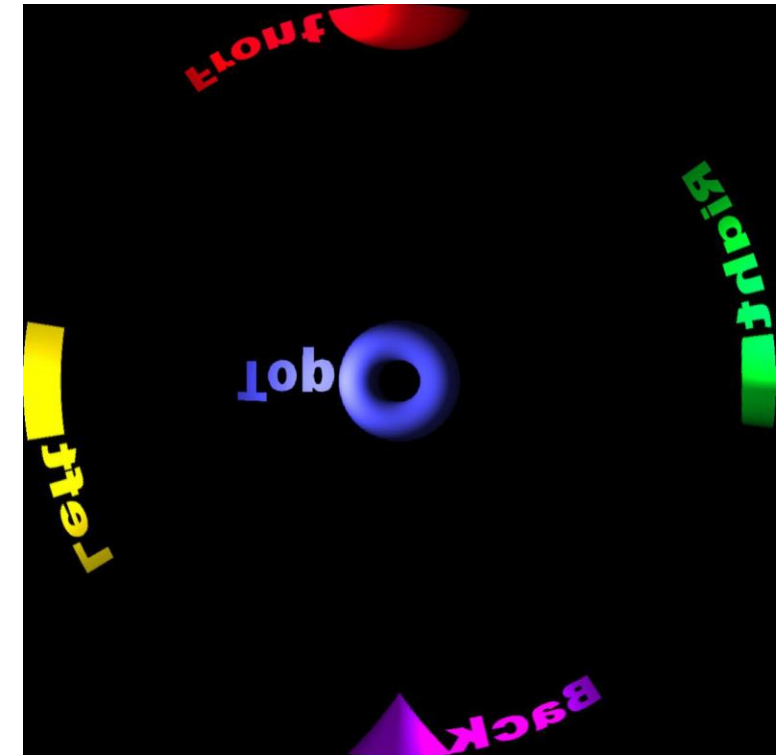
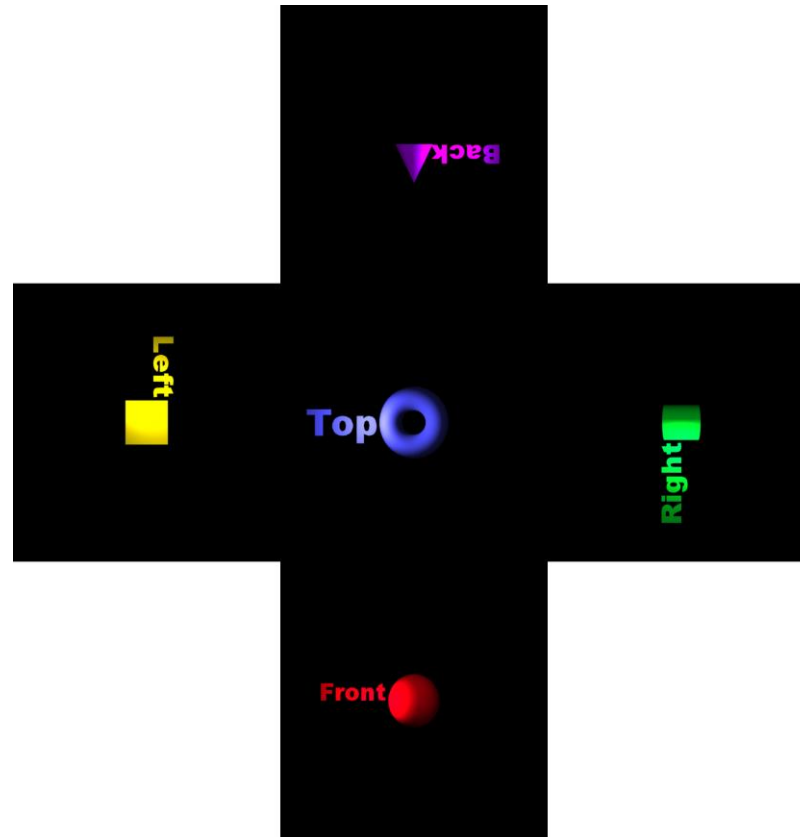
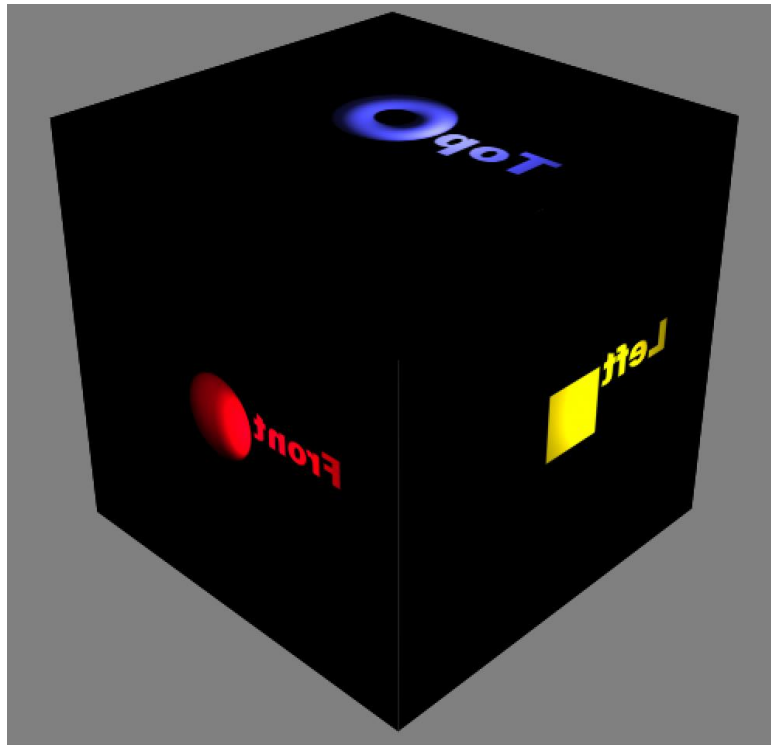


Δημιουργώντας ένα Χάρτη Κύβου

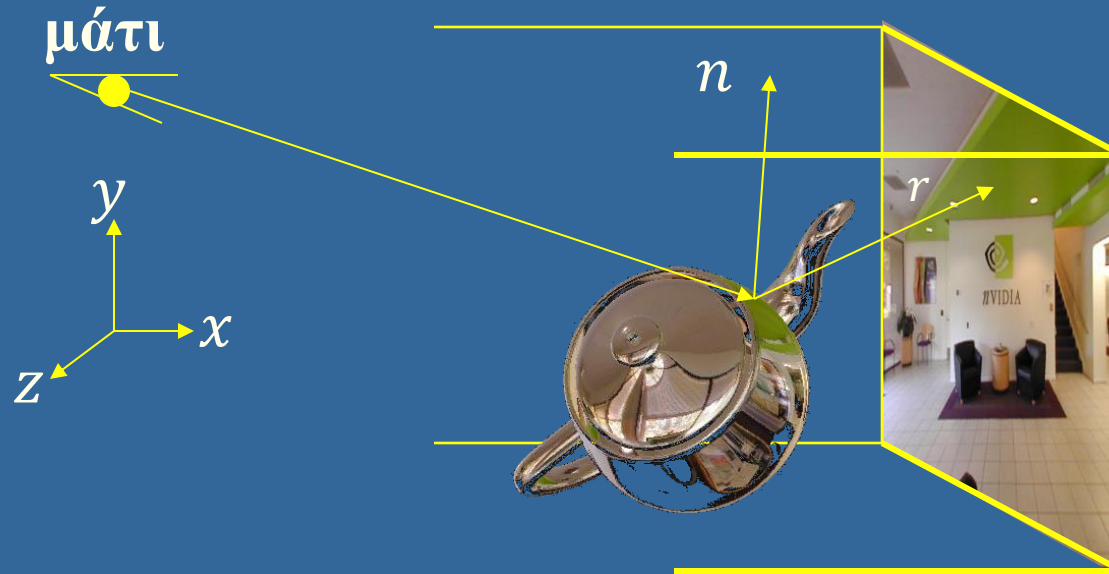
Χρήση 6 καμερών με 90 μοίρες γωνία θέασης η κάθε μία



Εικόνα Κύβου – Σφαίρας



Χαρτογράφηση σε Κύβο



- Απλά υπολόγισε το διάνυσμα ανάκλασης r
- Η μεγαλύτερη κατά απόλυτο τιμή των συντεταγμένων καθορίζουν την έδρα του κύβου.
 - $r = (5, -1, 2)$ δίνει την έδρα που βρίσκεται δεξιά στον άξονα x
- Διαιρώντας το r με το 5 δίνει $(u, v) = \left(-\frac{1}{5}, \frac{2}{5}\right)$

Σφαιρική Χαρτογράφηση



Σφαιρικός χάρτης
περιβάλλοντος

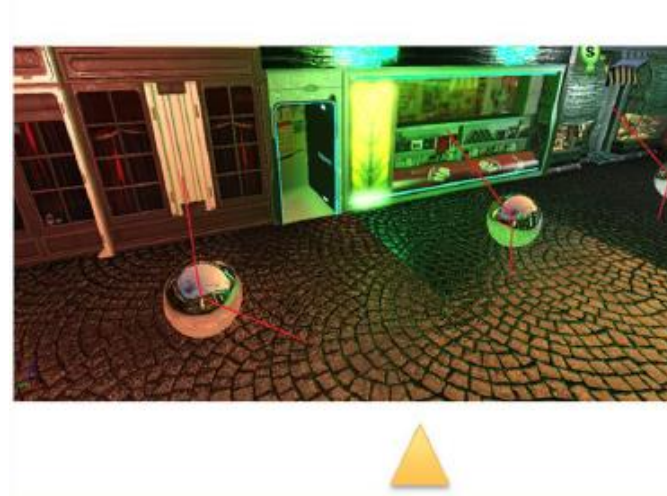
Χωρίς ανακλαστική υφή



Με ανακλαστική υφή



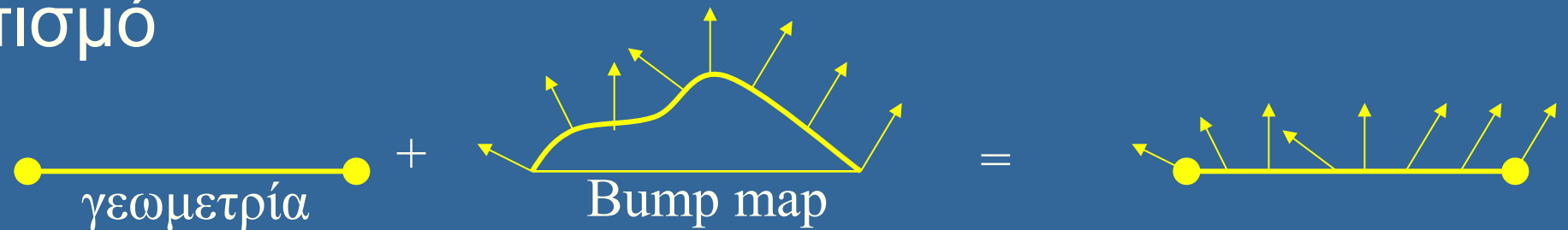
Προβλήματα & Λύσεις



- Πρέπει να υποθέσουμε ότι το περιβάλλον είναι πολύ μακριά από το αντικείμενο έτσι ώστε να μην παίζει ρόλο η γωνία θέασης
 - Το αντικείμενο δεν μπορεί να είναι κοίλο (δεν είναι δυνατές αυτοανακλάσεις)
 - Χωρίς αντανάκλασεις μεταξύ αντικειμένων
 - Χρειάζεστε έναν αντανάκλαστικό χάρτη για κάθε αντικείμενο
 - Χρειάζεστε νέο χάρτη εάν ο θεατής μετακινείται
-
- Χρήση ανιχνευτών ανάκλασης πραγματικού χρόνου για το δυναμικό κοντινό περιβάλλον
 - Συνδυασμός με ανακλάσεις στο χώρο οθόνης για τη κοντινή γεωμετρία
 - Χάρτες μικρότερης ανάλυσης για αντικείμενα χαμηλής κρισιμότητας
 - Συνδυασμός με χάρτες τραχύτητας για έλεγχο της ανακλαστικότητας με στόχο τον καλύτερο ρεαλισμό
 - Parallax-corrected χάρτες κύβων

Χαρτογράφηση Ανομοιομορφιών

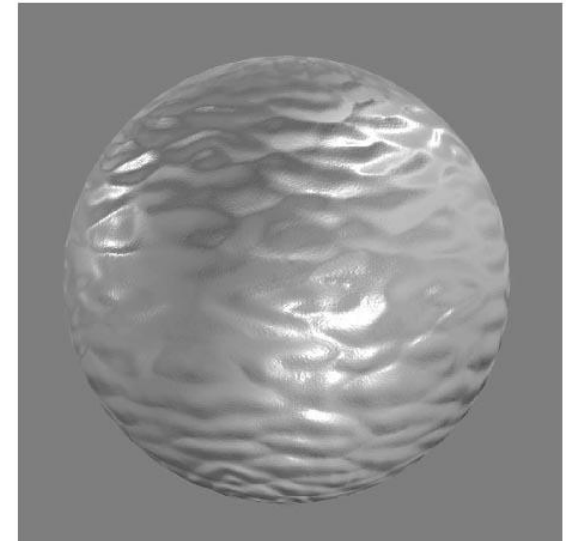
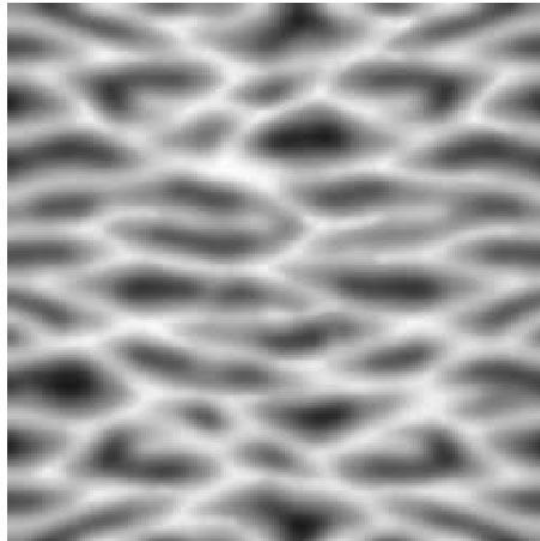
- Φθηνός τρόπος να προσομοιώνουμε ρυτίδες και λακκάκια στη γεωμετρία
 - Πολύ ακριβό να το κάνουμε γεωμετρικά
- Μία υφή αλλάζει το κάθετο διάνυσμα σε κάθε pixel και έπειτα χρησιμοποιεί αυτό το διάνυσμα για να υπολογίσει το φωτισμό



Αποθηκεύει ύψη: μπορεί να φτιάξει κάθετα διανύσματα

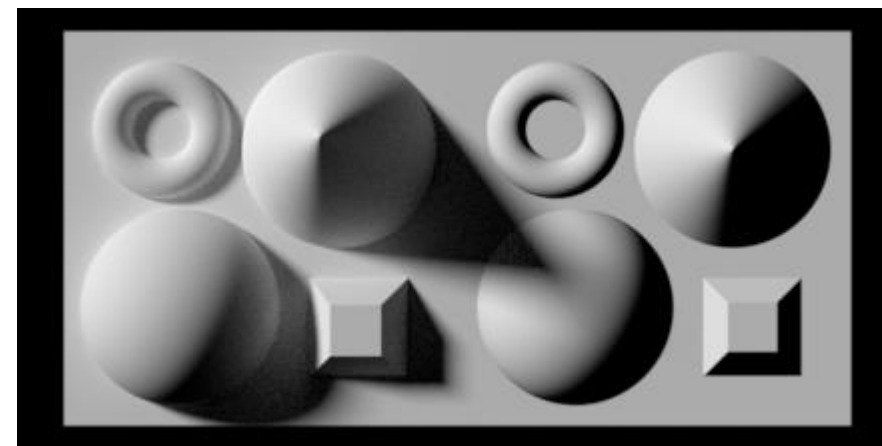
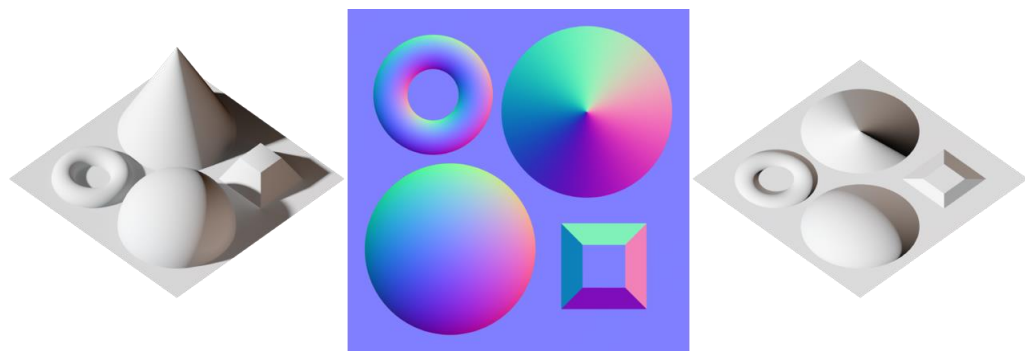
Χαρτογράφηση Ανομοιομορφιών

- Χρησιμοποιεί έναν **χάρτη υψής σε κλίμακα του γκρι** για την προσομοίωση διακυμάνσεων ύψους σε μια επιφάνεια.
 - Οι πιο σκοτεινές περιοχές αντιπροσωπεύουν χαμηλότερες περιοχές (εσοχές).
 - Οι πιο φωτεινές περιοχές αντιπροσωπεύουν υψηλότερες περιοχές (εξοχές).
- Αυτές οι πληροφορίες ύψους χρησιμοποιούνται για την τροποποίηση του τρόπου με τον οποίο το φως αλληλεπιδρά με την επιφάνεια, αλλάζοντας τα κάθετα διανύσματα της επιφάνειας κατά τους υπολογισμούς σκίασης.



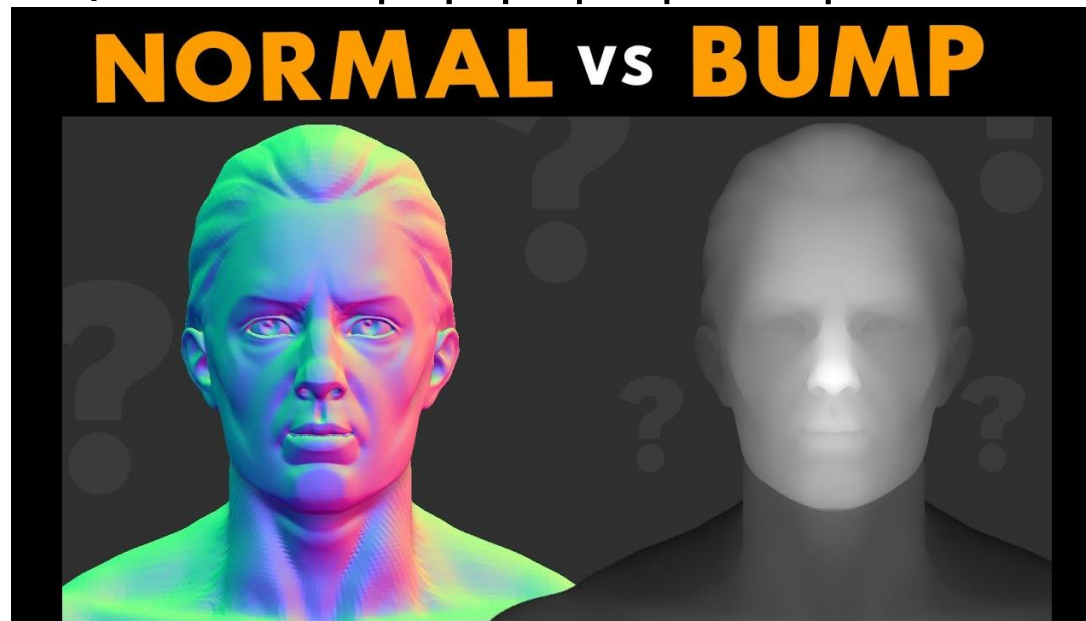
Χαρτογράφηση Καθετότητας

- Στη χαρτογράφηση ανομοιομορφίας βρίσκουμε το κάθετο διάνυσμα από το αποθηκευμένο υψόμετρο
- Η χαρτογράφηση καθετότητας (normal mapping) αποθηκεύει απευθείας το κάθετο διάνυσμα σε μία υφή

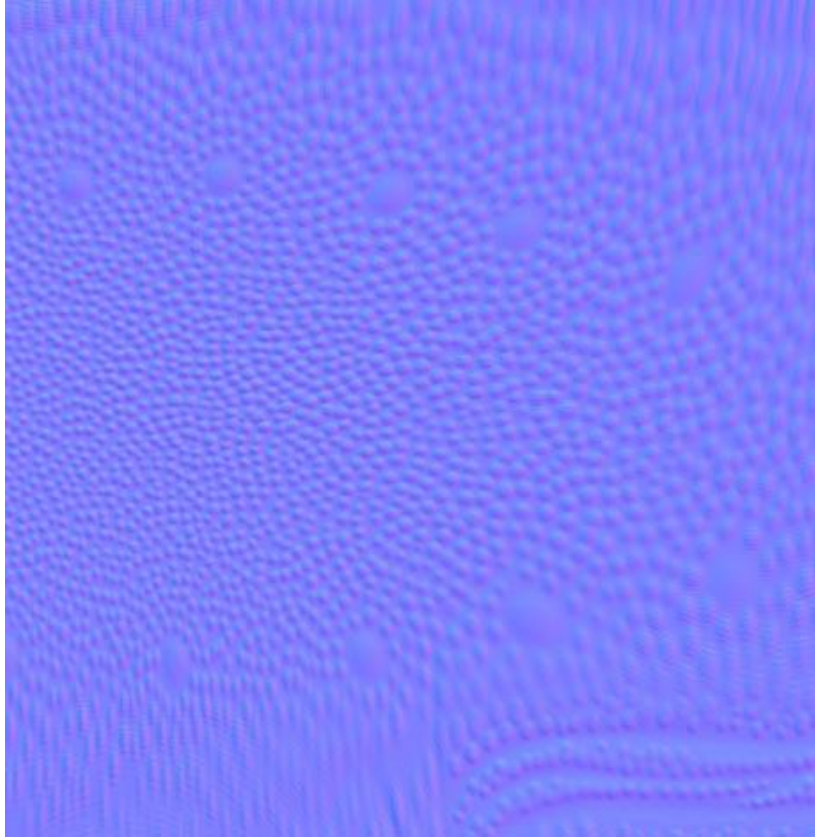


Χαρτογράφηση Καθετότητας

- Χρησιμοποιεί *έναν έγχρωμο χάρτη υφής* (συνχνά σε μορφή RGB) για να κωδικοποιήσει κανονικά διανύσματα επιφάνειας για κάθε σημείο της επιφάνειας.
 - Κάθε pixel στον κανονικό χάρτη αντιπροσωπεύει την κατεύθυνση που βλέπει η επιφάνεια σε αυτό το σημείο.
- Τα κανονικά διανύσματα χρησιμοποιούνται απευθείας στους υπολογισμούς σκίασης, επιτρέποντας πιο λεπτομερή εφέ φωτισμού.



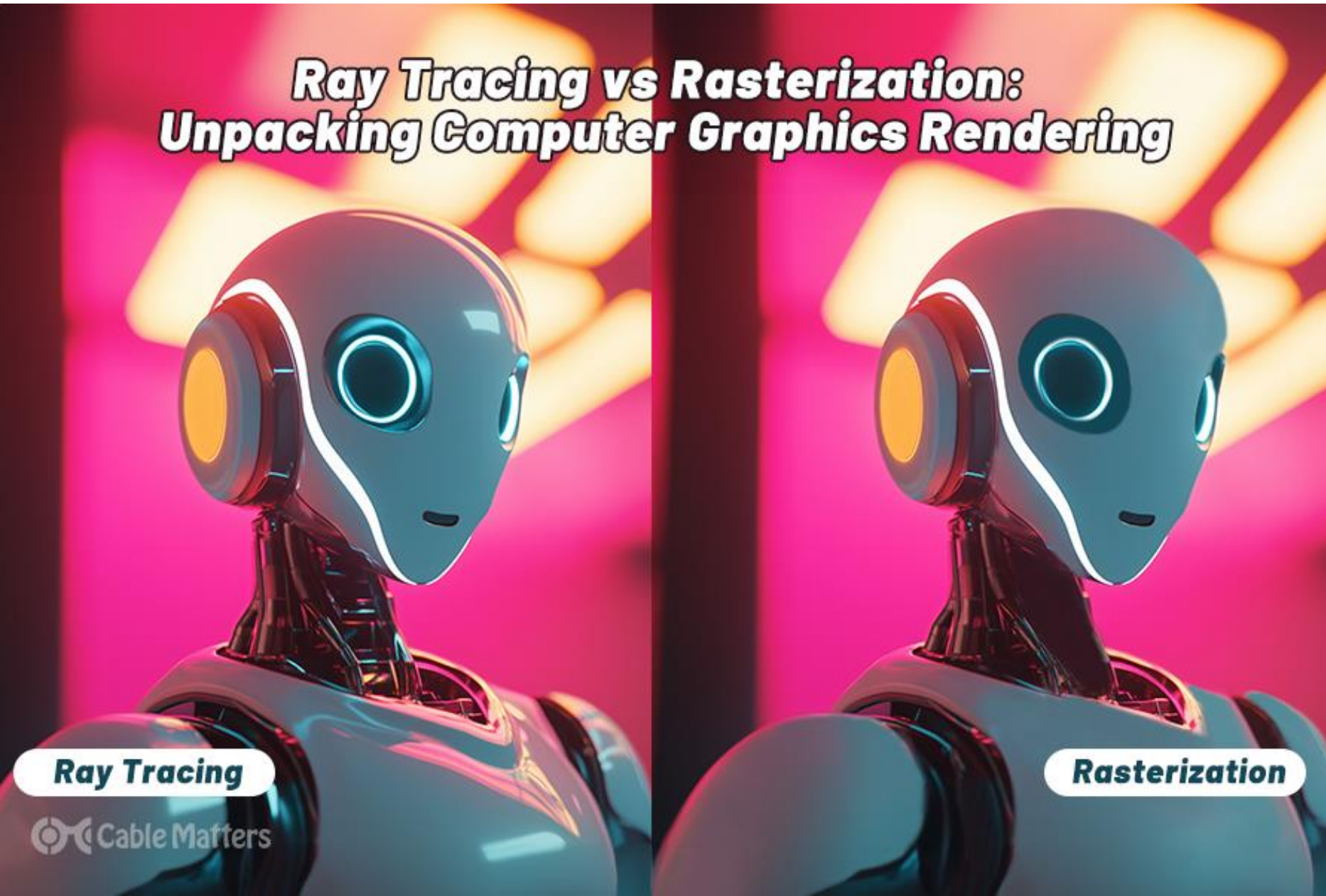
Παράδειγμα



ΆΛΛΕΣ ΡΟΈΣ ΑΠΌΔΟΣΗΣ ΓΡΑΦΙΚΩΝ

Προσεγγίσεις Απόδοσης Γραφικών

Ray Tracing vs Rasterization: Unpacking Computer Graphics Rendering



Ray Tracing

 Cable Matters

Rasterization

Η απόδοση γραφικών χρησιμοποιεί γενικά μία από τις δύο προσεγγίσεις

- Rasterization
- Ray tracing (το Lumen στο UE5 πάει προς τα εκεί – το HW επίσης πάει προς τα εκεί σε συνδυασμό με FSR 4 και DLSS 4)
- άλλες μέθοδοι όπως η radiosity αλλά για offline απόδοση γραφικών

Ραστεροποίηση έναντι Ray Tracing

Βρόχος Ραστεροποίησης

Για κάθε αντικείμενο

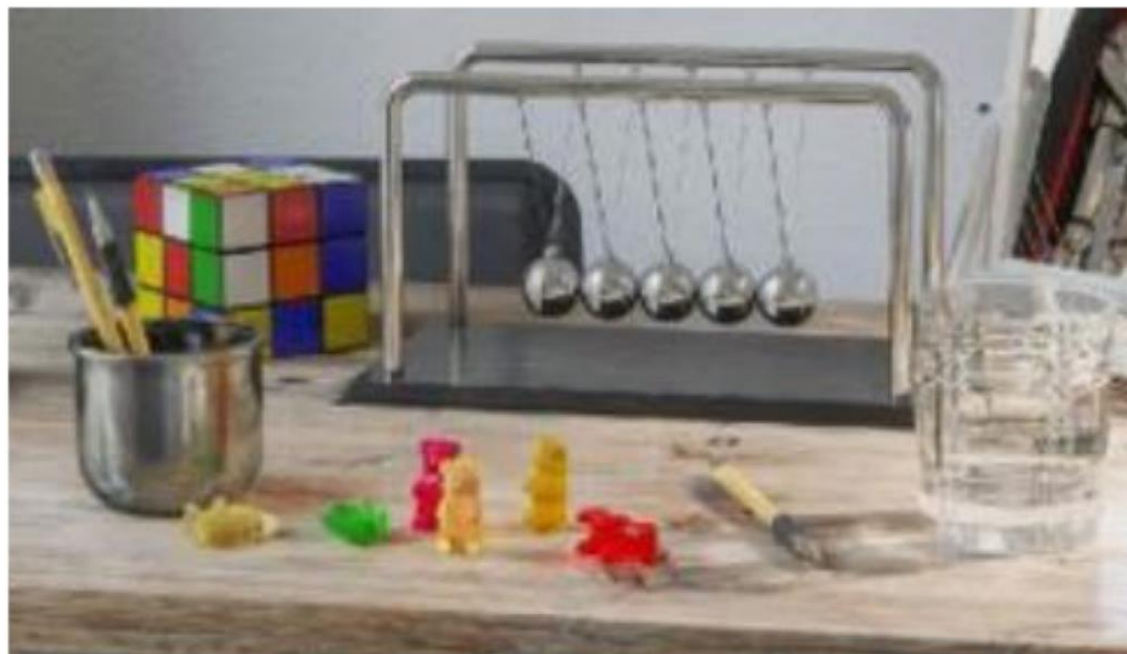
Για κάθε «κοντινό» pixel



Βρόχος Ray Tracing

Για κάθε pixel

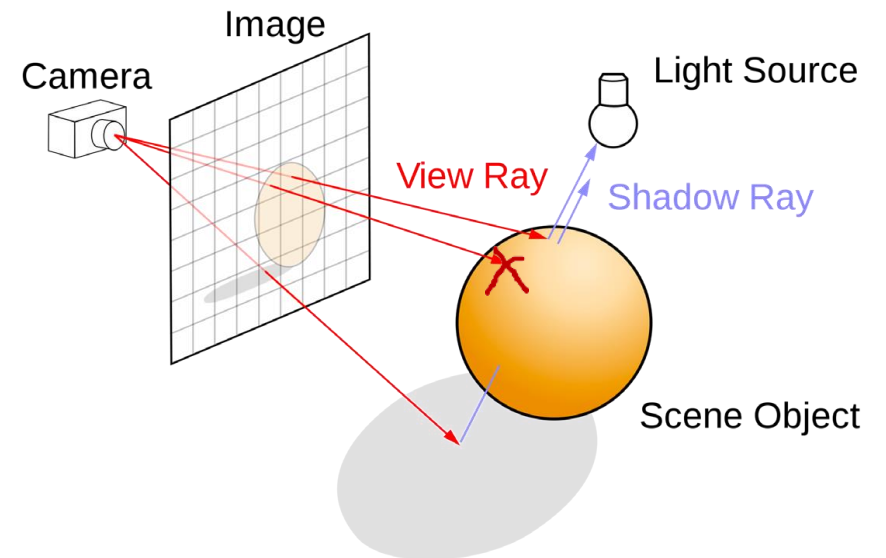
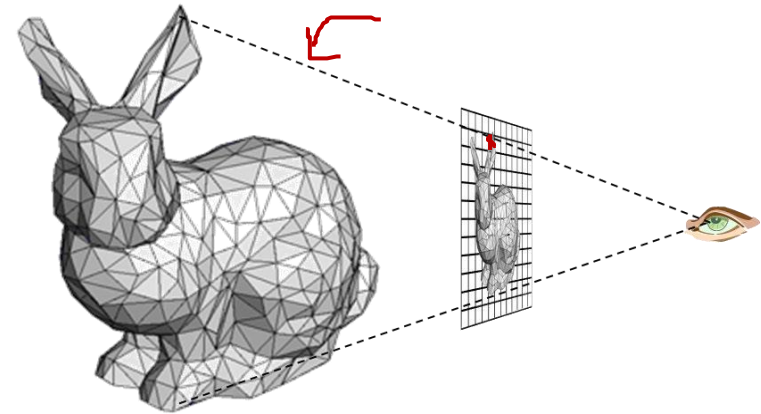
Για κάθε «κοντινό» αντικείμενο



Rasterization έναντι Ray Tracing

Υπεραπλουστεύοντας:

- Στο Rasterization, τα γεωμετρικά αντικείμενα προβάλλονται σε ένα επίπεδο και γίνεται ο υπολογισμός των στοιχείων που γεμίζουν.
- Στο Ray tracing, μοντελοποιούμε τη φυσική μεταφορά του φωτός στέλνοντας μία ακτίνα μέσω κάθε εικονοστοιχείου σε ένα επίπεδο εικόνας και βλέποντας τι χτυπά η ακτίνα στη σκηνή



Ray Tracing...

Αφού το Ray tracing φαίνεται να είναι πιο κοντά στη πραγματικότητα γιατί δεν χρησιμοποιείται ευρέως ακόμα;

- Εύκολο για μία απλή σκηνή με απλό φωτισμό
- Επί της αρχής μπορεί να δημιουργήσει καθολικά εφέ φωτισμού όπως σκιές και πολλαπλές ανακλάσεις αλλά είναι αργό ιδιαίτερα για παιχνίδια

ΟΧΙ ΠΙΑ (αν έχετε ισχυρή και σύγχρονη GPU)

