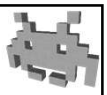
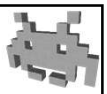
Course Plan



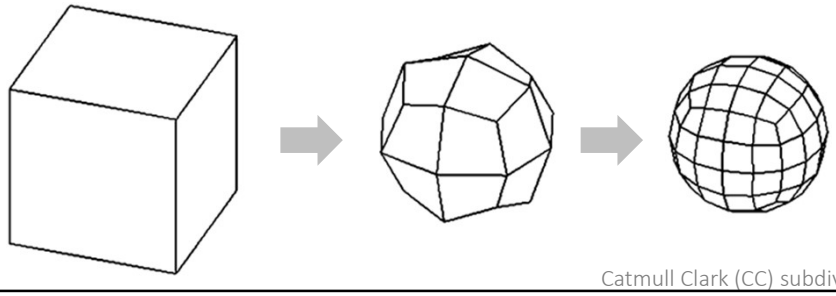
lec. 1: Introduction
lec. 2: Mathematics for 3D Games
lec. 3: Scene Graph
lec. 4: Game 3D Physics +
lec. 5: Game Particle Systems
lec. 6: Game 3D Models
lec. 7: Game Textures
lec. 8: Game 3D Animations
lec. 9: Game 3D Audio
lec. 10: Networking for 3D Games
lec. 11: Artificial Intelligence for 3D Games
lec. 12: Game 3D Rendering Techniques

68

3D mesh authoring techniques: subdivision surfaces



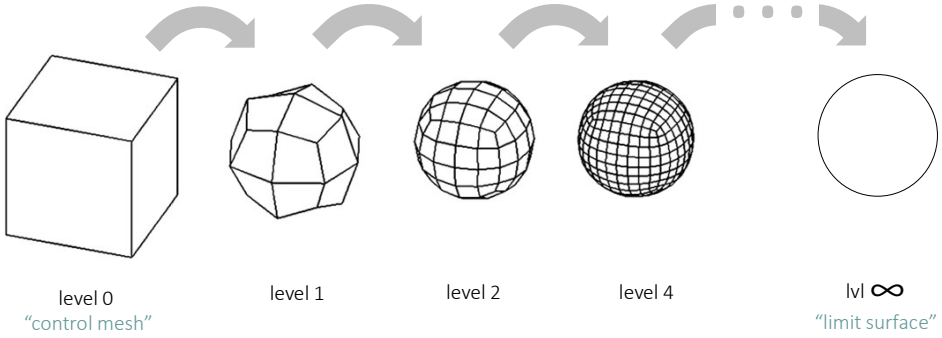
- Subdivision step:
  an algorithm that operates on a mesh
  and obtains a higher resolution, smoother mesh
- Can be iterated



Catmull Clark (CC) subdivision

69

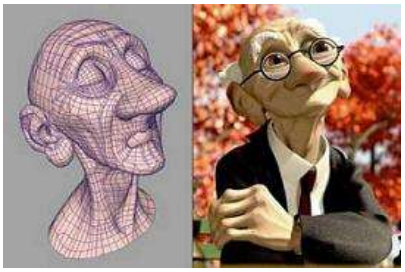
Example: with Catmull-Clark scheme



70

3D mesh authoring techniques: subdivision surfaces

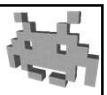
- Many subdivision algorithms (schemas) exists
  - each with its own properties
- Produces clean, regular meshes
- Excellent for smooth, curved, organic looking objects



famously pioneered by movie industry (not games):

71

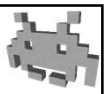
Subdivision surfaces as a tool…



- …to encode smooth surfaces
  - Idea: we encode the control mesh to represent the limit surface
  - use in games: rendering (now, rare – but popular around 2015)
    1. keep control mesh in GPU ram
    2. let 1-3 subdivision steps happen during rendering
- …to author 3D meshes
  - idea: alternate (low-poly) editing and subdivisions steps
  - at first steps: edit global shape
  - at last steps: edit minute details
  - use in games: during asset creation, by artists

72

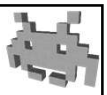
Subdivision surfaced as way to define (curved) surfaced



- Modeler creates a low-poly mesh, the "control mesh"
  - control mesh: piecewise linear (i.e. flat) surface
- The control mesh is subdivided (in theory ∞ times) and a "limit surface" is obtained
  - limit surface: curved & smooth surface
- The control mesh is a representation of the limit surface
  - note: the subdivision steps are only performed on the fly, during rendering
  - the more step are done, the better the limit surface is approximated

73

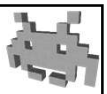
Subdivision surfaces as a mesh authoring tool

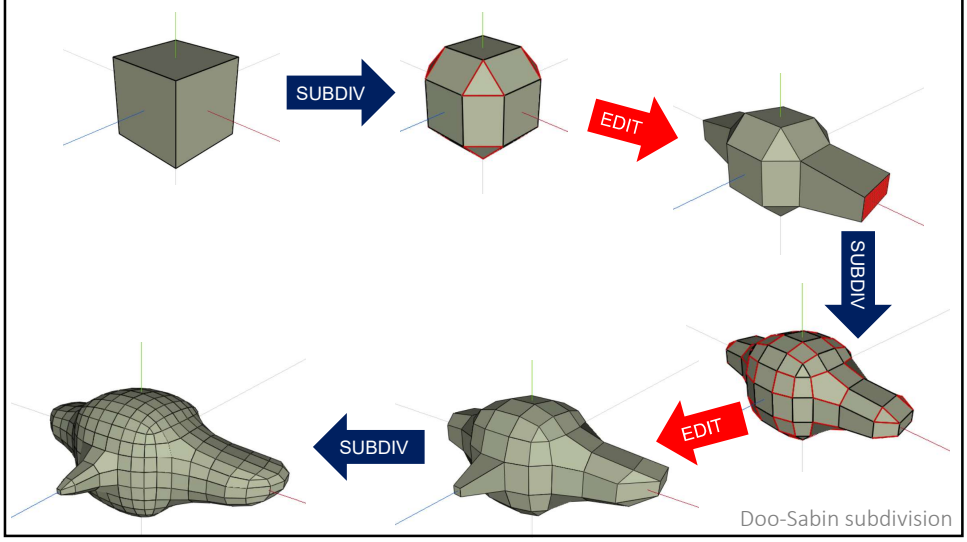


1. Create a coarse mesh with a very approx. shape
   - e.g. by low-poly modelling
2. Apply subdivision step
   - a higher resolution model
3. Re-edit results
   - Retouch all the smaller parts
4. Goto 2, until good final result

74

Subdivision surfaces as a mesh authoring tool (example)

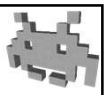




Doo-Sabin subdivision

76

Example of subdivision schema: “Butterfly” (for tri-meshes)

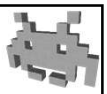


- It a 1-to-4 schema
  in a subdivision step, each triangle is split into 4
  by adding one vertex in each original edge

- Which coordinates have to be assigned to the new vertex?
  Every subdivision scheme has its own formula. For instance…

78

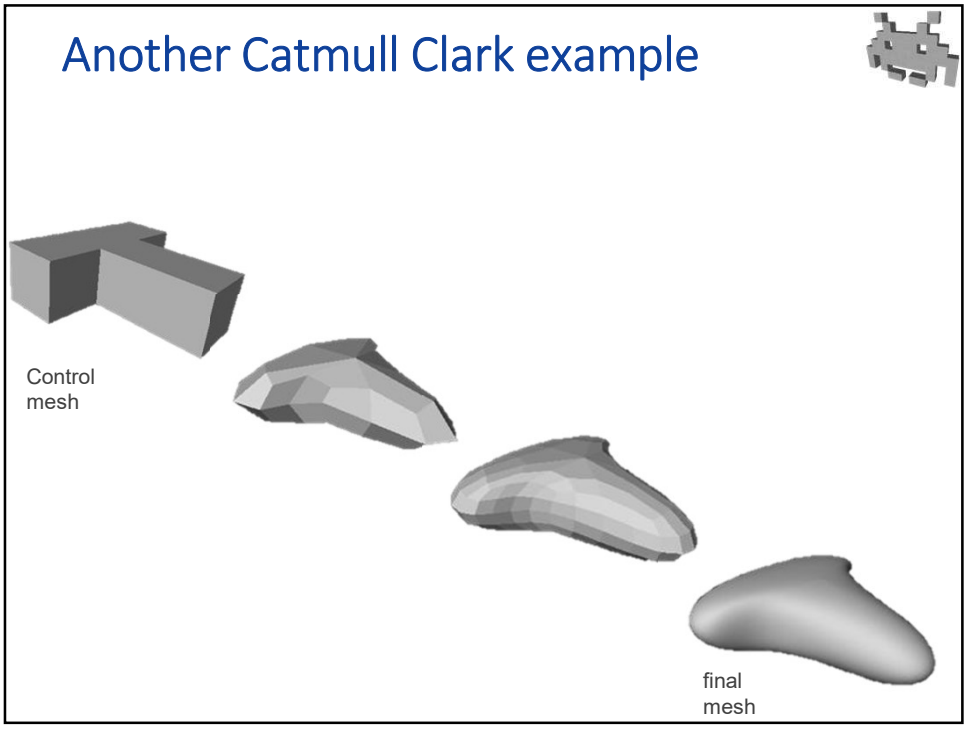
Subdivision surfaces in general



- A step typically increases resolution by a factor **x4**
- The geometry of the subidvided mesh (xyz pos of its vertices) is computed according to a formula of the pos of their neighbors.
  - In some schemas (called interpolative), the old vertices are kept at the same positions
  - In other schemas (called approximative), old vertices are kept but moved into a new position
  - In other schemas (called dual) older vertices aren’t kept
- Most created vertices are *regular*

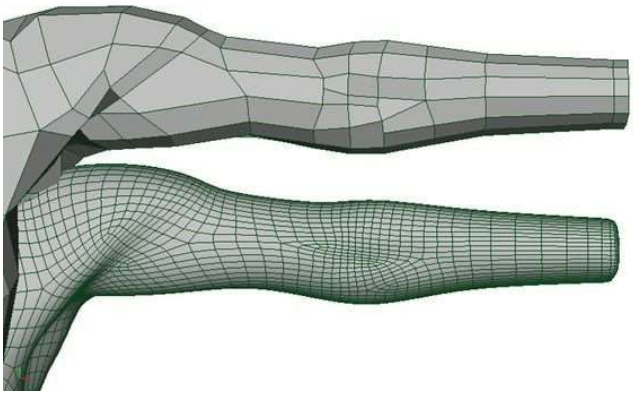
80

Another Catmull Clark example



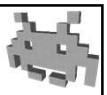
81

Another Catmull Clark example



82

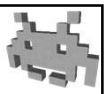
Commonest subdivision schemas



- Doo-Sabin
  - can operate on any polygonal mesh
  - produces polygonal meshes
  - dual scheme
- Loop
  - 1-to-4 scheme for triangle meshes (only)
  - appoximative schema
  - nicely smooth limit surface (C2)
- Butterfly
  - 1-to-4 scheme for triangle meshes (only)
  - interpolative scheme
  - not-so-smooth limit surface (only C1)

84

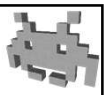
Commonest subdivision schemas



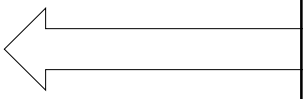
- Catmull-Clark
  - can operate on any polygonal mesh
  - produces quad-meshes
  - approximative scheme
  - nicely smooth limit surface (C2)
  - traditionally, movie-industry favourite
  - a recent trend in games: use it for mesh rendering
    - store control mesh in GPU ram,
    - subdivide mesh on card on the fly during rendering (as much as required)
    - this is possible thanks to closed formula approximation of limit surface

85

3D Mesh authoring: approaches

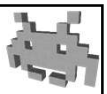


- Popular 3D modeling approaches:
  - Direct low-poly modelling
    - e.g. with wings3D
  - Subdivision surfaces
    - e.g. with blender
  - Digital sculpting
    - e.g. with Z-brush, (or Sculptris Alpha)



87

Digital Sculpting



mouse (or stylus)

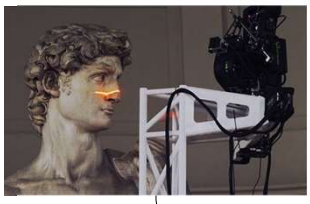
=

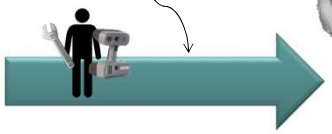
chisel

88

3D models: other sources

- 3D scanning
  - Technologies for obtaining 3D digital models from real objects





3D scanning
(e.g. laser scanning)

89

3D models: other sources

Sculptor (real)

Reale model

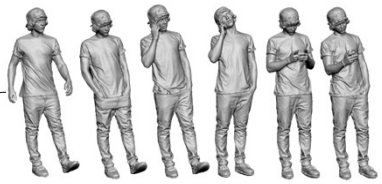
3D scanning

Hi res model

90

3D models: other sources

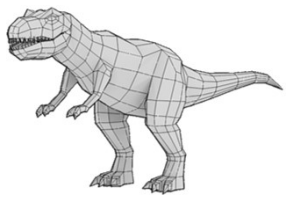
- 3D scanning
  - A.k.a. *automatic 3D model acquisition*
  - Lot of different technologies
    - Laser scanners
    - Time of flight
    - Structured light (kinect)
    - …
  - Different characteristics
    - Results quality
      - Noise / resolution
    - Automatism
    - Invasiveness
      - Markers? Powder?
    - Real time? (kinect)
    - Price
    - Max object dimension
      - (full body scanner?)



91

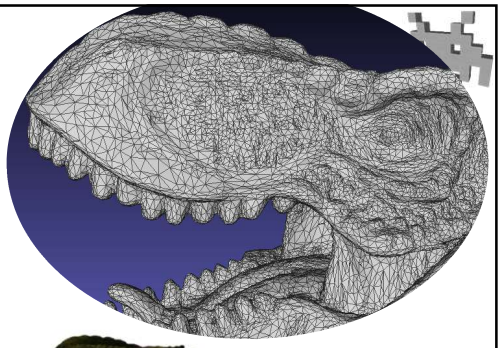
3D models sources: comparison

PERFECT for games!
(much easier to: animate, re-edit, uvmap, …)



manually edited
low-poly mesh
(2K triangles)

VS



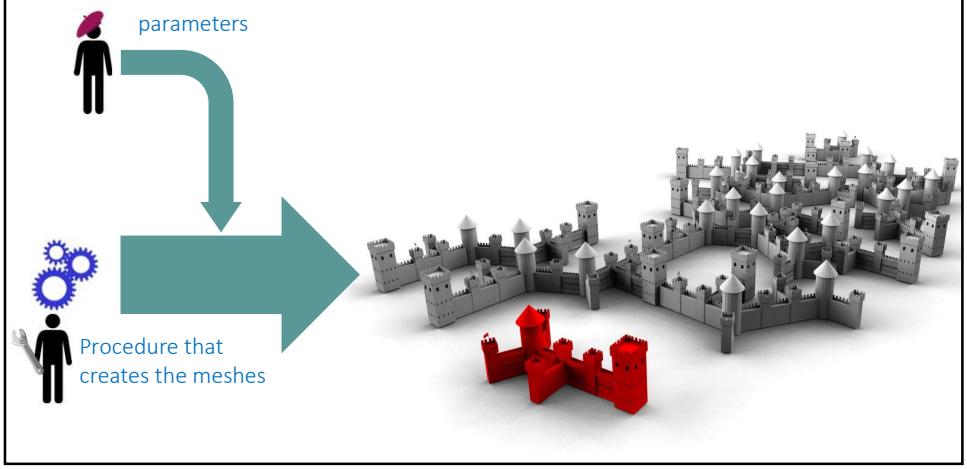
Dino, scanned by artec3d

scanned & cleaned
hi res mes
(30K triangles)

(sculpted meshes are similar)

92

Another source for 3D models: procedural modelling



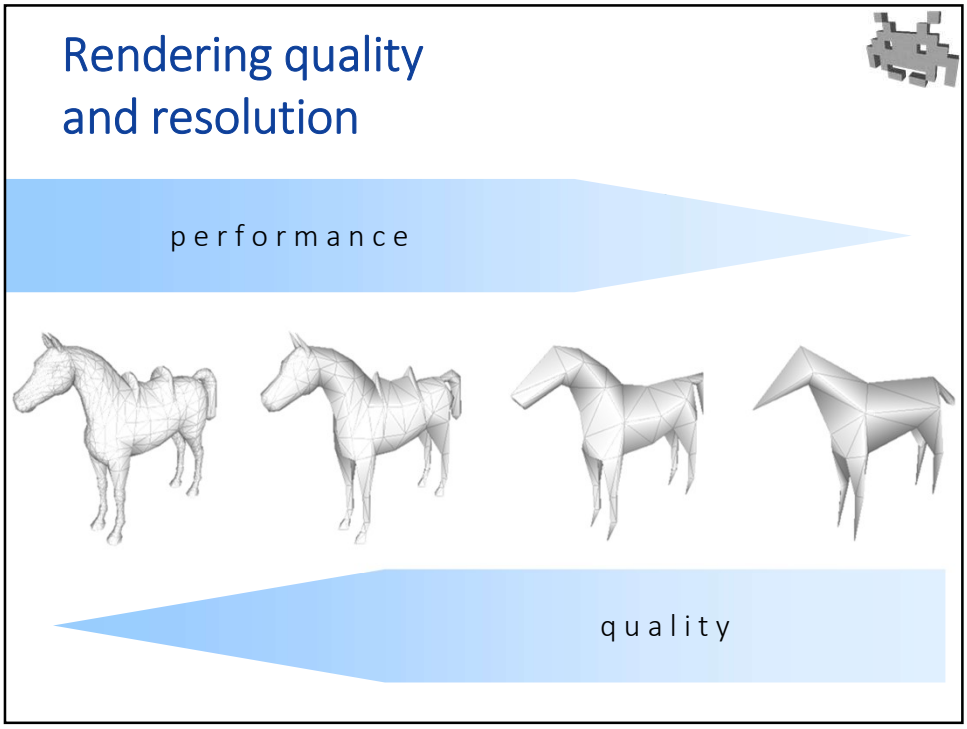
93

Again about mesh resolution

- all costs: linear on the triangles number
  - in memory (disk, CPU RAM, GPU RAM)
  - in time (rendering, loading, etc)
- (and, linear # of vert. with # triangles)
  - (*rule of thumb*: K verts → 2K tris)
- reminder: possible adaptive resolution
  - higher-res in some parts
  - lower-res in others

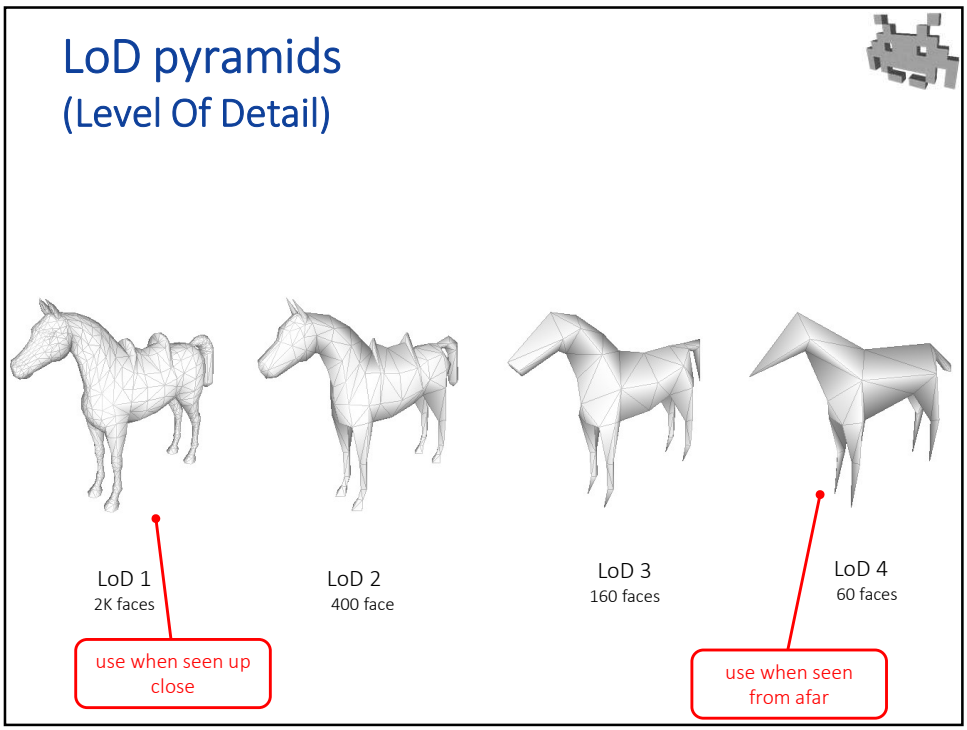
94

Rendering quality and resolution



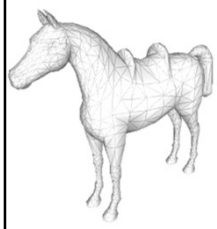
95

LoD pyramids (Level Of Detail)



96

LoD pyramids (Level Of Detail)



97

LoD pyramids (Level Of Detail)

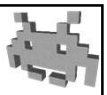
- Goal:
  - decrease the geometry budget (total number of vertices)
  - ideal: size of triangles in screen space (in pixel): constant
    - (if importance / complexity is the same)
- Task: determining the level to use (dynamically, at runtime)
  - depending on observer distance
  - and/or, depending on rendering workload
    - e.g.: rendering is lagging ⇒ decrease LoD
  - this is task of the rendering engine)

computed from scene graph (how?)

- Task: LOD creation or "LOD-ding" (during asset creation)
  - starting from LOD-0 (higher-res)
  - manual, or automatic *(see later on),* or assisted (mixed)
    - often manual
  - note: sometimes "LoD 0" is used only in special cases
    - e.g. during a cut-scenes

98

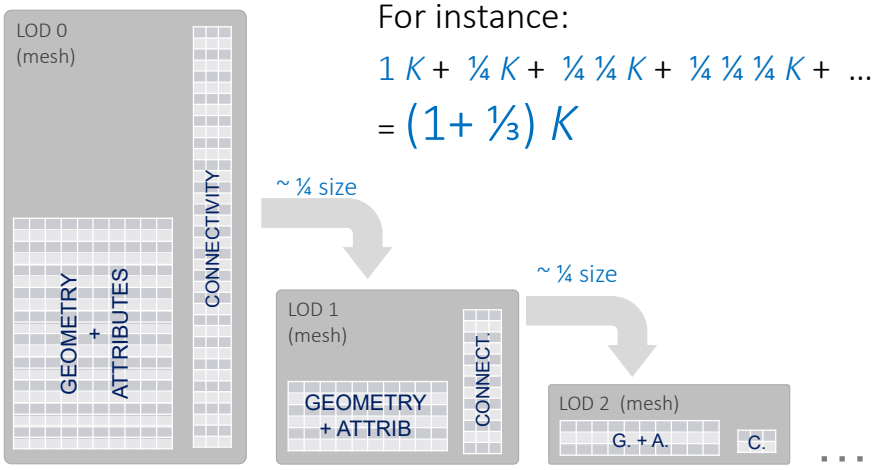
LoD pyramids (Level Of Detail)



Total memory usage: limited
For instance:

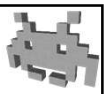
$1\ K + \frac{1}{4}\ K + \frac{1}{4}\frac{1}{4}\ K + \frac{1}{4}\frac{1}{4}\frac{1}{4}\ K + \ldots$

$= (1 + \frac{1}{3})\ K$

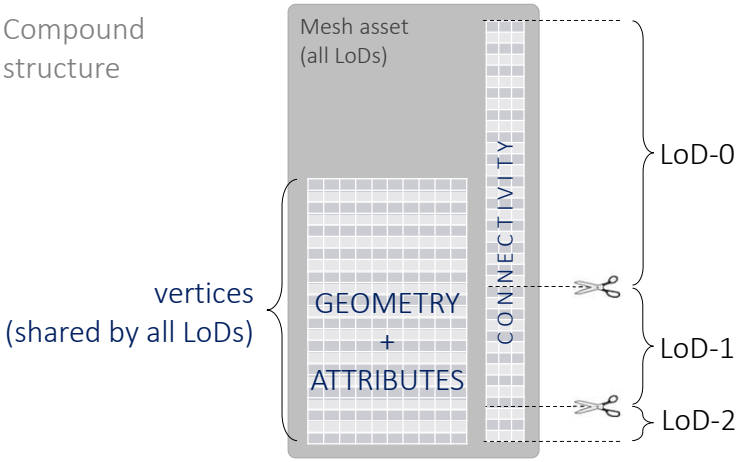


99

LoD pyramids (Level Of Detail)

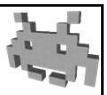


Compound structure

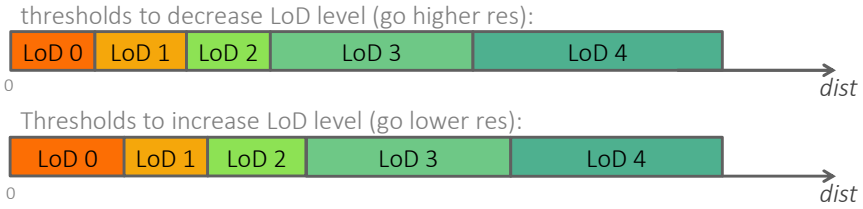


100

LoD pyramids (Level Of Detail)

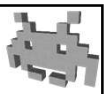


- Popping artefact: ⚠ beware
  - to attenuate: instead of a fixed thresholds:
    - when current = LoD 2:
      if (dist > 11m) → switch to LoD3
    - when current = LoD 3:
      if (dist < 9m) → switch to LoD2

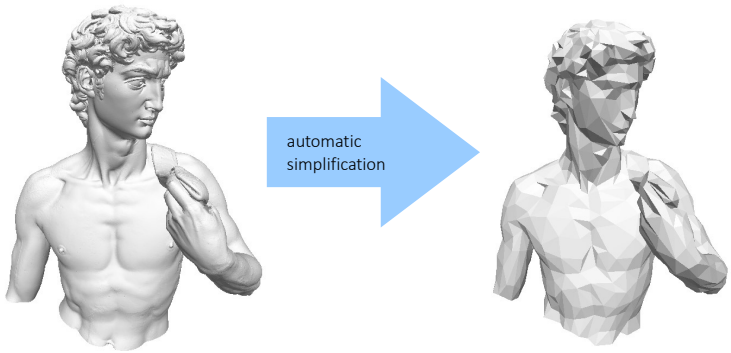


101

Poly-reduction (aka Mesh "simplification" / "decimation")



- parameters:
  - a maximum error
  - or number of faces objective

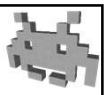


Original mesh
500K triangles

Simplified mesh
2K triangles

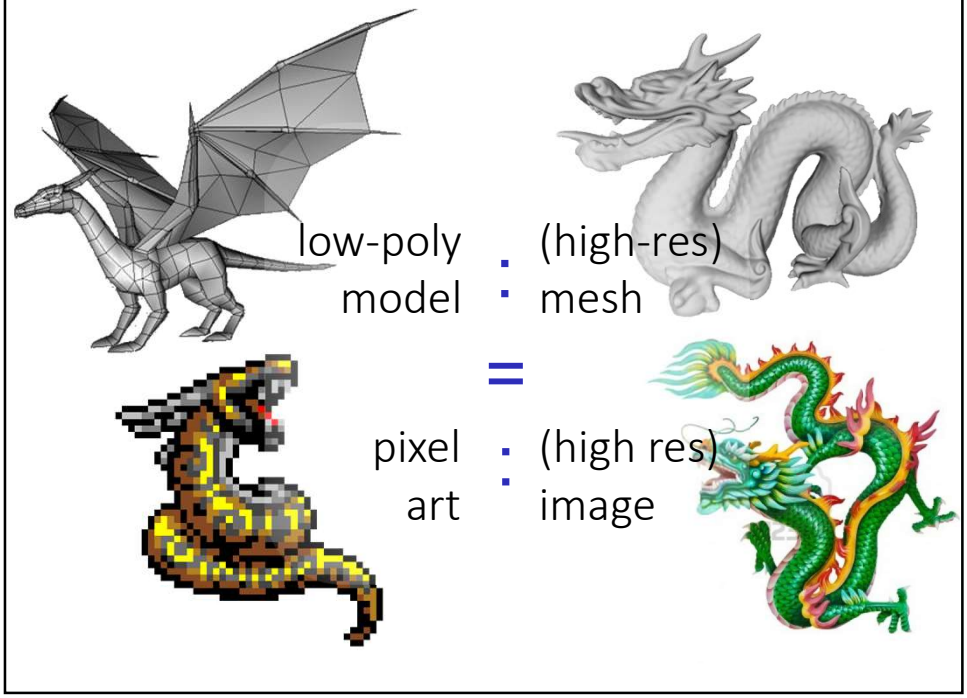
102

Poly-reduction (called mesh simplification)



- Different approaches are studied in Geometry Processing.
  - Adaptive or not
    - use more triangles where needed (ex. not in flat parts)
    - or not
  - Maximum error introduced:
    - measured and/or limited
    - or not
  - Topology:
    - kept
    - or not
  - Streamable
    - Possible
    - or not

103



104