Deferred Shading

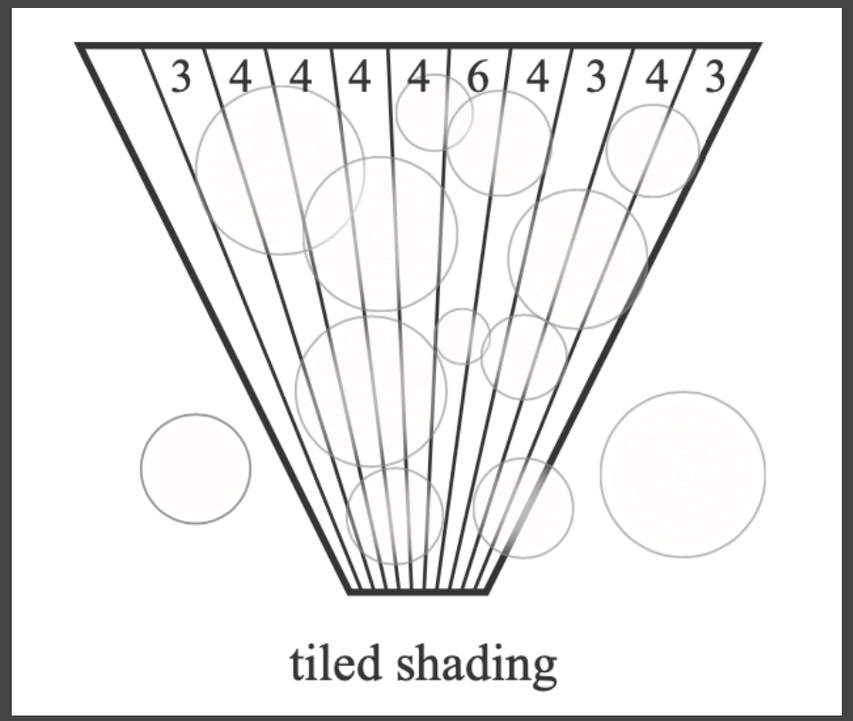
- Originally invented to save shading time
- Consider the rasterization process
  - Triangles -> fragments -> depth test -> shade -> pixel
  - Each fragment needs to be shaded (in what scenario?)
  - Complexity: O(#fragment * #light)
- Key observation
  - Most fragments will not be seen in the final image
  - Due to depth test / occlusion
  - Can we only shade those visible fragments?

Deferred Shading

- Modifying the rasterization process
  - Just rasterize the scene twice
  - Pass 1: no shading, just update the depth buffer
  - Pass 2 is the same (why does this guarantee shading visible frag. only?)
    - Implicitly, this is assuming **rasterizing the scene** is way faster than **shading all unseen fragments** (usually true)
    - Complexity: O(#fragment * #light) -> O(#vis. frag. * #light)
- Issue
  - Difficult to do anti-aliasing
  - But almost completely solved by TAA

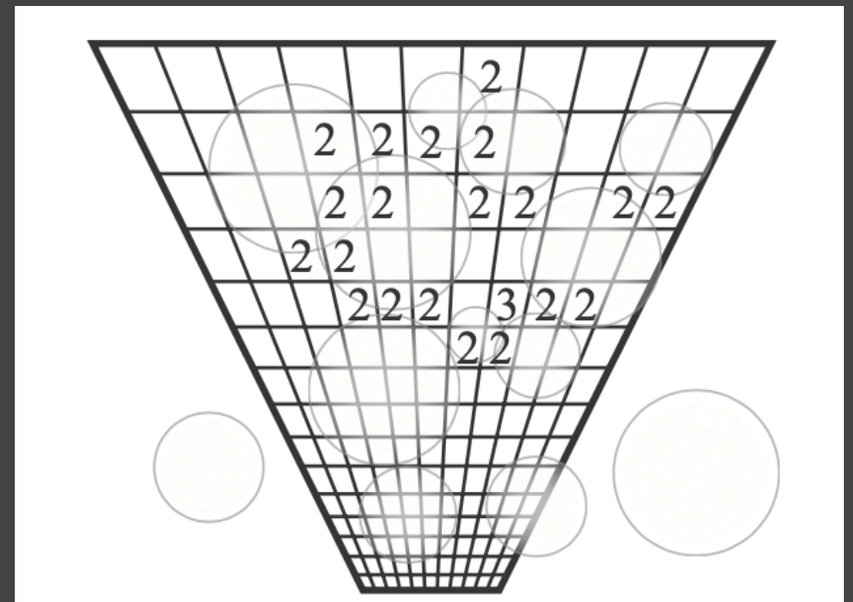
Tiled Shading

- Improvement: tiled shading
  - Subdivide the screen into tiles of e.g. 32x32 then shade each
- Key observation
  - Not all lights can illuminate a specific tile
  - Mostly due to the square falloff with distance (!)
  - Complexity: O(#vis. frag. * #light)
    -> O(#vis. frag. * avg #light per tile)



Clustered Shading

- Further improvement: clustered shading
  - Further subdivide each tile into different depth segments
  - Essentially subdividing the view frustum into a 3D grid
- Key observation
  - The depth range of each tile can be quite large
  - Therefore, a lot of lights may be identified to have potential to lit the tile
  - But some lights may only lit a small depth range
  - Complexity: O(#vis. frag. * avg #light per tile) -> O(#vis. frag. * avg #light per cluster)



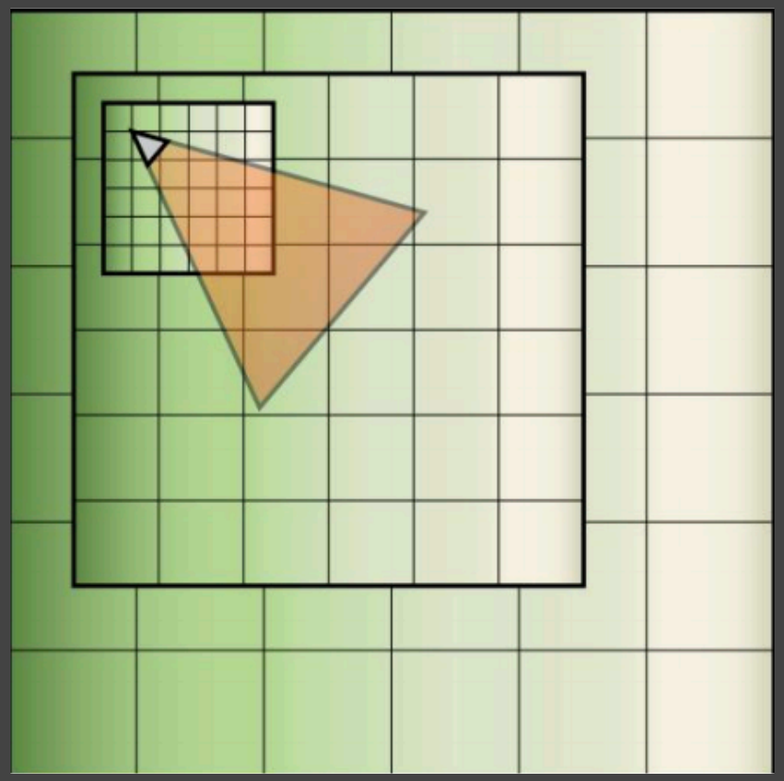
Level of Detail Solutions

- Level of Detail (LoD) is very important
  - Recall: texture MIPMAP-ing
  - Choosing the right level of detail to use can save computation

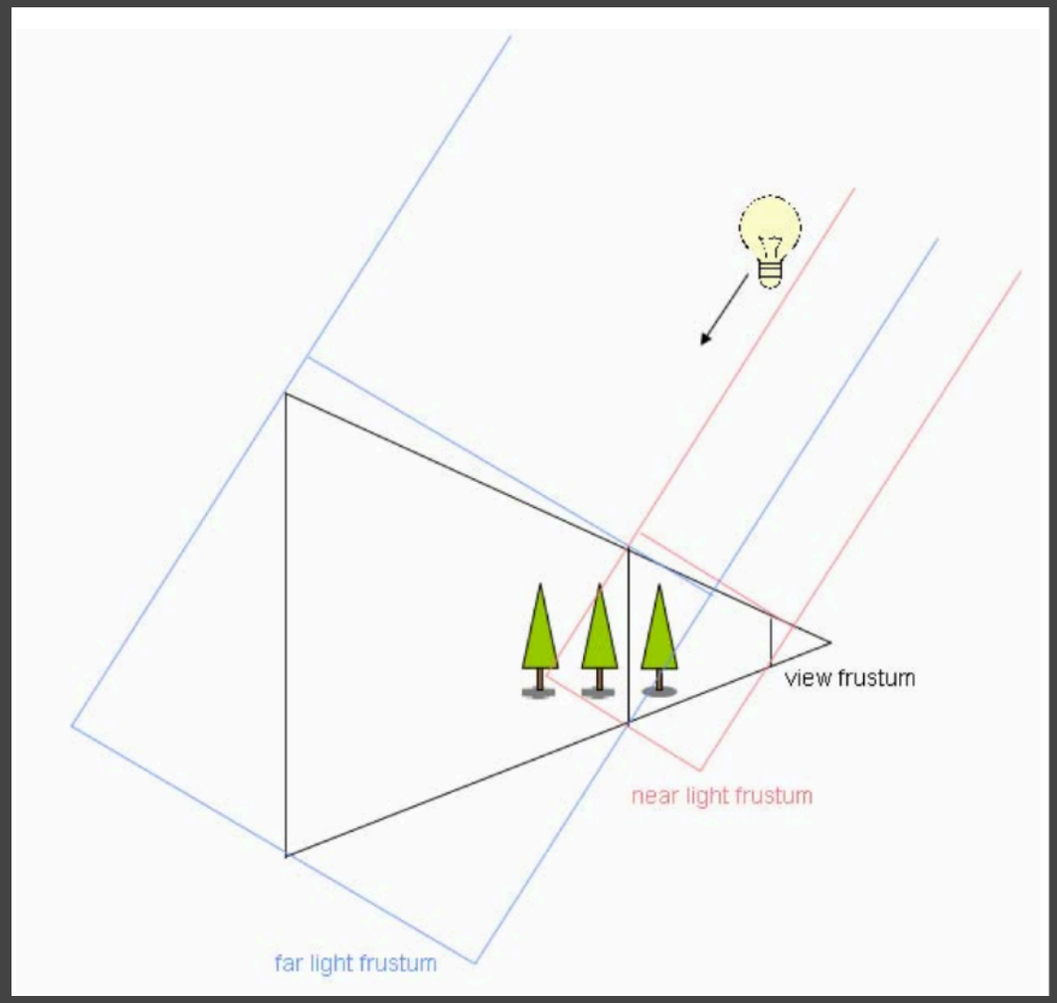
- The use of multiple levels of detail
  - Often called “cascaded” by the RTR industry

Level of Detail Solutions

- Example
  - Cascaded shadow maps
  - Cascaded LPV



[Anton Kaplanyan, Light Propagation Volumes in CryEngine 3]



[Dimitrov et al., Cascaded Shadow Maps]

Level of Detail Solutions

- Key challenge
  - Transition between different levels
  - Usually need some overlapping and blending near boundaries

- Another example: geometric LoD
  - Recall: pre-generating a set of simplified obj. with different #tri.
  - Based on the distance to the camera, choose the right object to show (or part of obj., s.t. no triangle will be larger than a pixel)
  - Popping artifacts? Leave it to TAA!
  - This is Nanite in UE5 (but of course, Nanite has way more)

Level of Detail Solutions

- FYI, some (strongly) technical difficulties
  - Different places with different levels, how about cracks?
  - Dynamically load and schedule different levels, how to make the best use of cache and bandwidth, etc.?
  - Representing geometry using triangles or geometry textures?
  - Clipping and culling for faster performance?
  - …

Global Illumination Solutions

- From this course, we can see that
  - Recall, when would screen space ray tracing (SSR) fail?
  - There is no single GI solution that is perfect for all cases, except for RTRT
  - But completely using RTRT is still too costly in the current generation
  - Therefore, the industry tends to use hybrid solutions

- For example, a possible solution to GI may include
  - SSR for a rough GI approximation (similar to our HW3)
  - Upon SSR failure, switching to more complex ray tracing
    - Either hardware (RTRT) or software (?)

Global Illumination Solutions

- Software ray tracing
  - HQ SDF for individual objects that are close-by
  - LQ SDF for the entire scene
  - RSM if there are strong directional / point lights
  - Probes that stores irradiance in a 3D grid (Dynamic Diffuse GI, or DDGI)
- Hardware ray tracing
  - Doesn’t have to use the original geometry, but low-poly proxies
  - Probes (RTXGI)
- The highlighted solutions are mixed to get Lumen in UE5

Summary: A Brief Q&A

- What is interesting?
  - Anything that requires thinking
  - Therefore, giving up thinking == committing suicide

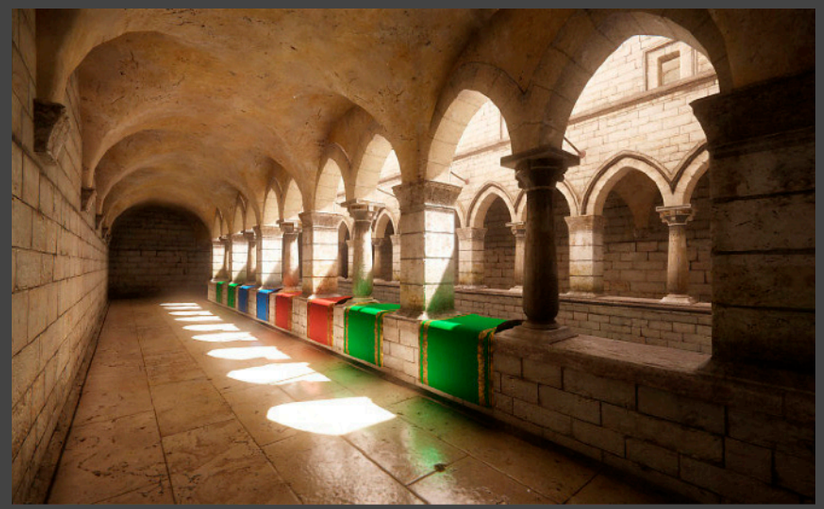
- Is implementation less important than theory?
  - NEVER. But engineering skills must be acquired in engineering.

- You don’t teach implementation, does it mean that you are not good at programming?
  - Dude, I was in Tsinghua’s ACM/ICPC team

Questions?

Congratulations!

Real-time shadows / env. lighting



Real-time global illumination

Real-time shading / materials

Real-time ray tracing

Congratulations!

- Yet still, a lot of uncovered topics
  - Texturing an SDF
  - Transparent material and order-independent transparency
  - Particle rendering
  - Post processing (depth of field, motion blur, etc.)
  - Random seed and blue noise
  - Foveated rendering
  - Probe based global illumination
  - ReSTIR, Neural Radiance Caching, etc.
  - Many-light theory and light cuts
  - Participating media, SSSSS
  - Hair appearance
  - …

Computer Graphics

is

AWESOME!

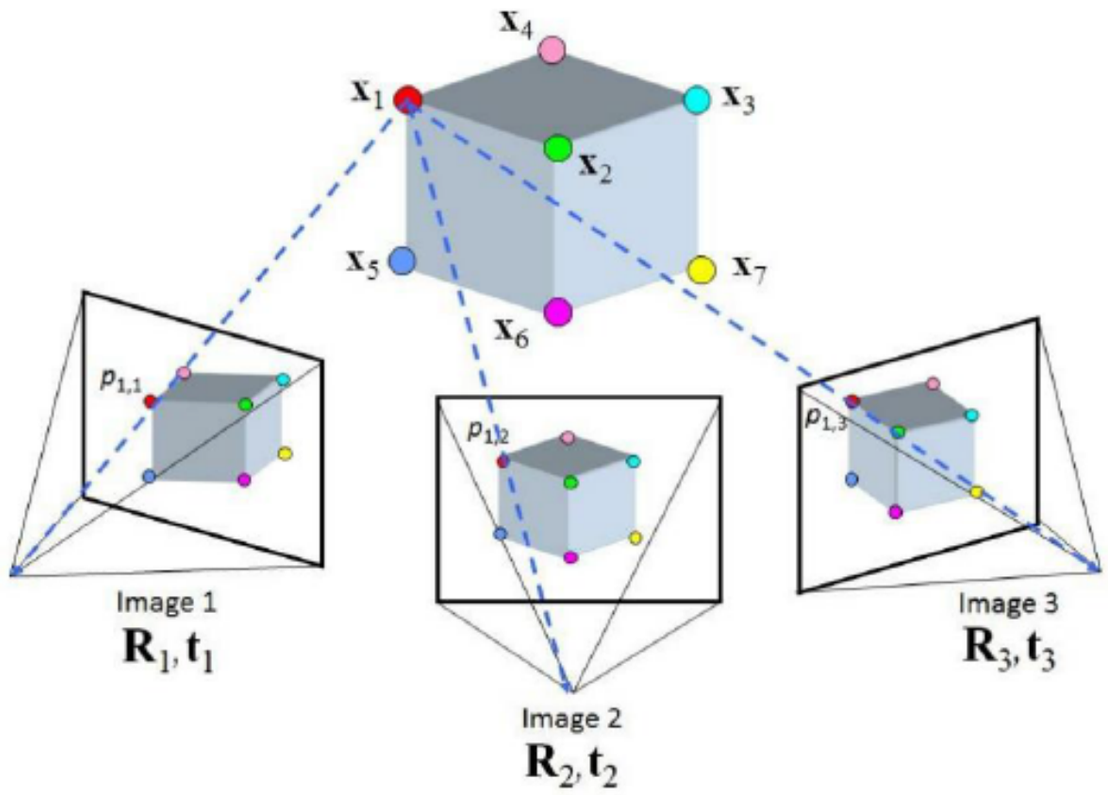
Advertisements

GAMES203

- 3D vision (reconstruction and more)
  - A combination of computer vision and computer graphics
- Staring July 2, 2021
- By Prof. Qixing Huang
  - From the University of Texas at Austin
- Let’s take a glance at this course
  - 3 different topics

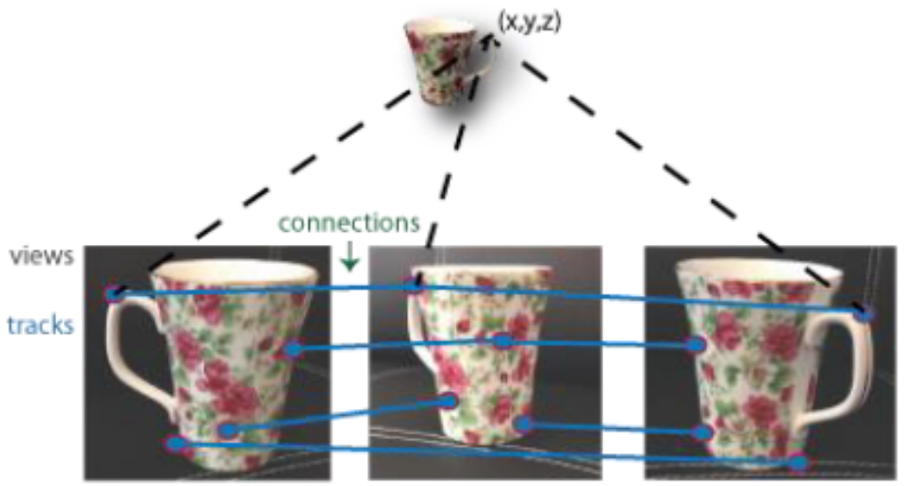
Topic I: 3D Reconstruction

- Geometry
  - Epipolar geometry
  - Fundamental matrix
  - Extrinsic/Intrinsic camera parameters
  - Camera calibration
  - Vanishing points
  - Homogeneous coordinates

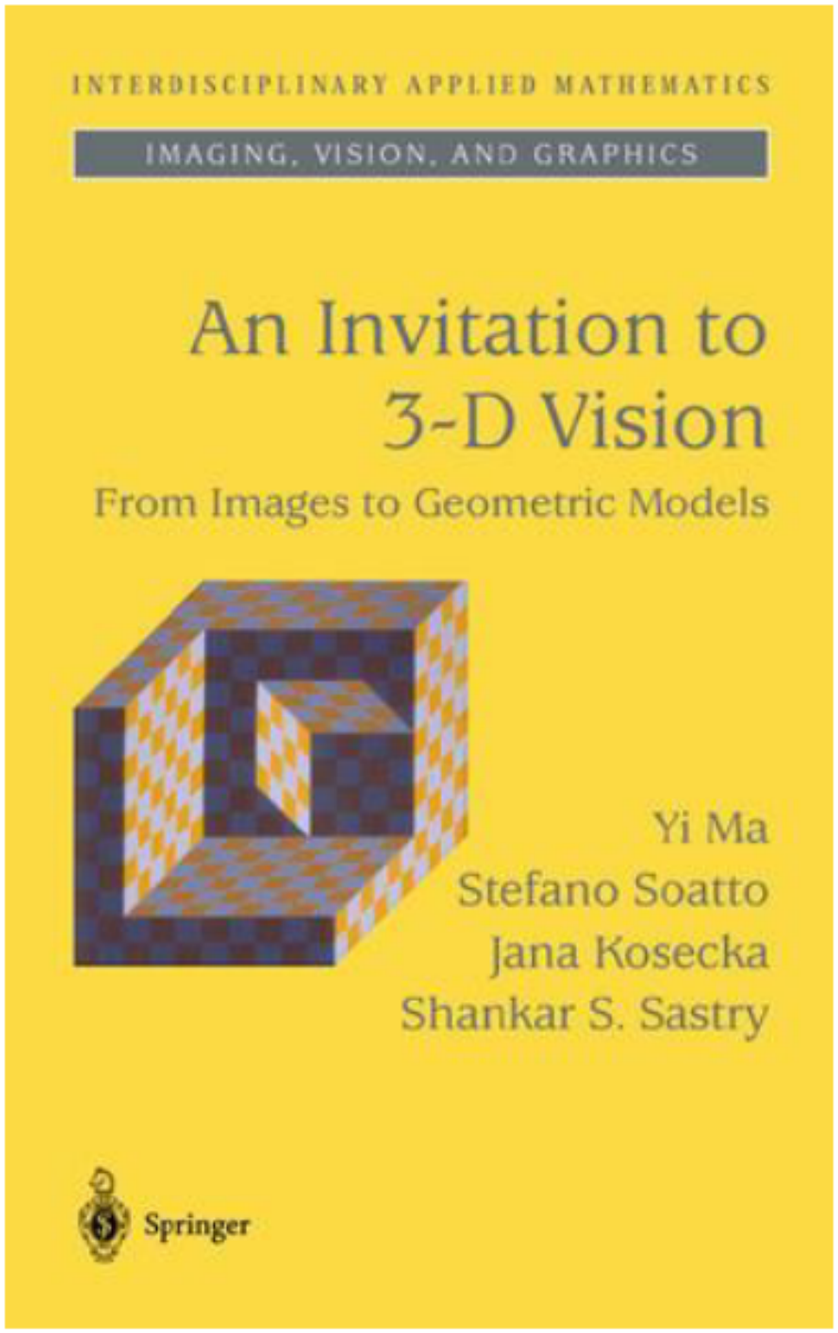


Topic I: 3D Reconstruction

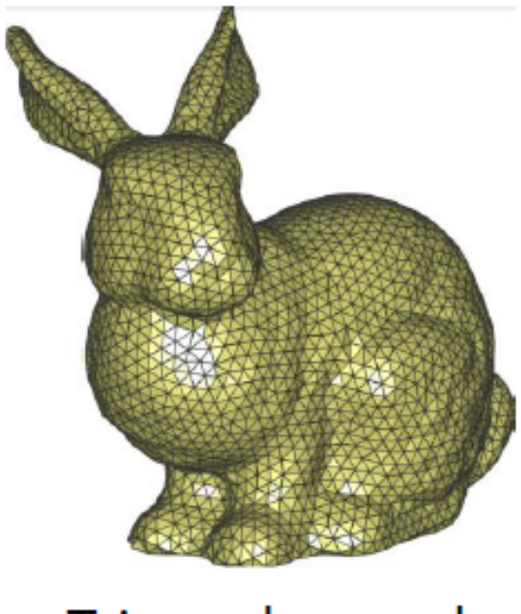
- Algorithms
  - Feature extraction
  - Feature correspondences
  - Relative camera pose
  - Structure-from-motion
  - Multiview stereo
  - Bundle adjustment
  - ICP



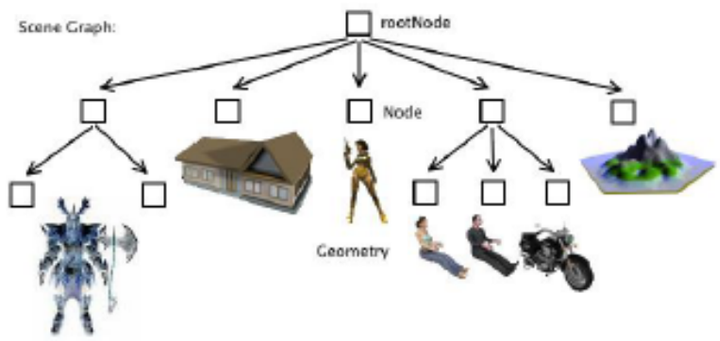
Textbook


INTERDISCIPLINARY APPLIED MATHEMATICS
IMAGING, VISION, AND GRAPHICS
An Invitation to
3-D Vision
From Images to Geometric Models
Yi Ma
Stefano Soatto
Jana Kosecka
Shankar S. Sastry
Springer

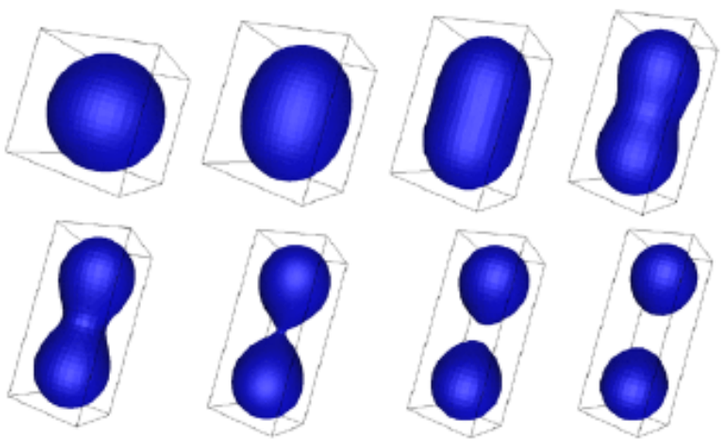
Topic II: How to represent 3D Data



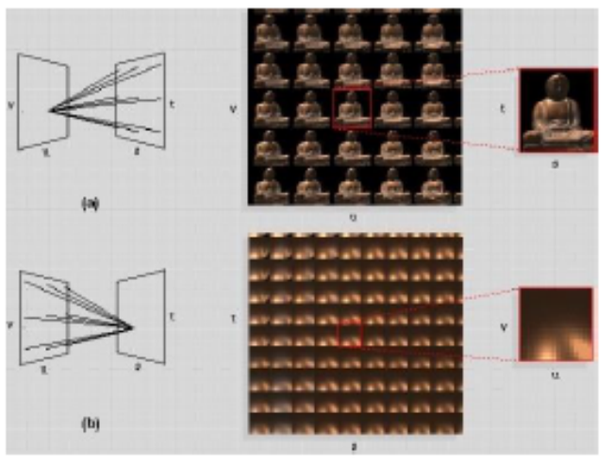
Triangular mesh



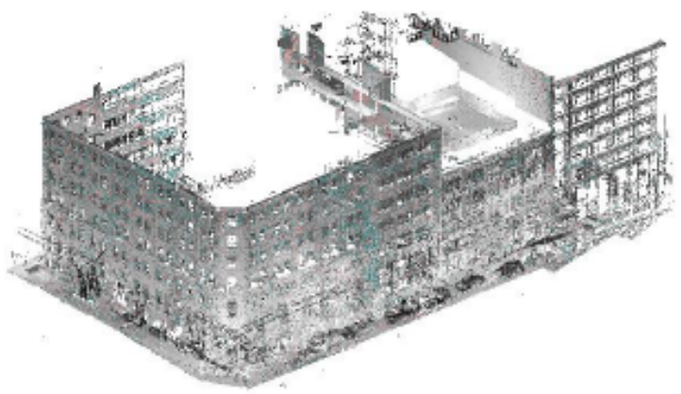
Part-based models



Implicit surface



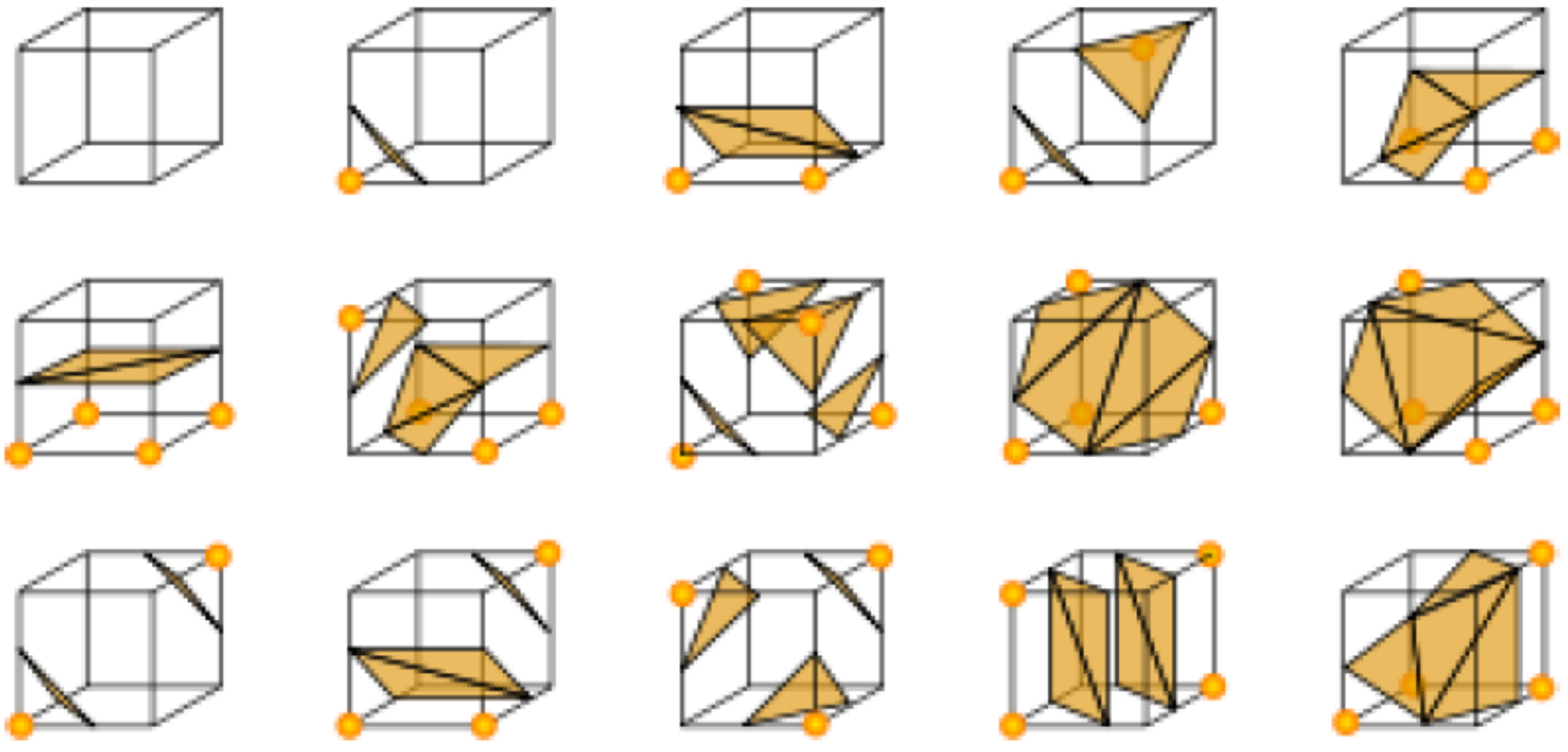
Light Field Representation



Point cloud

Conversion between different representations

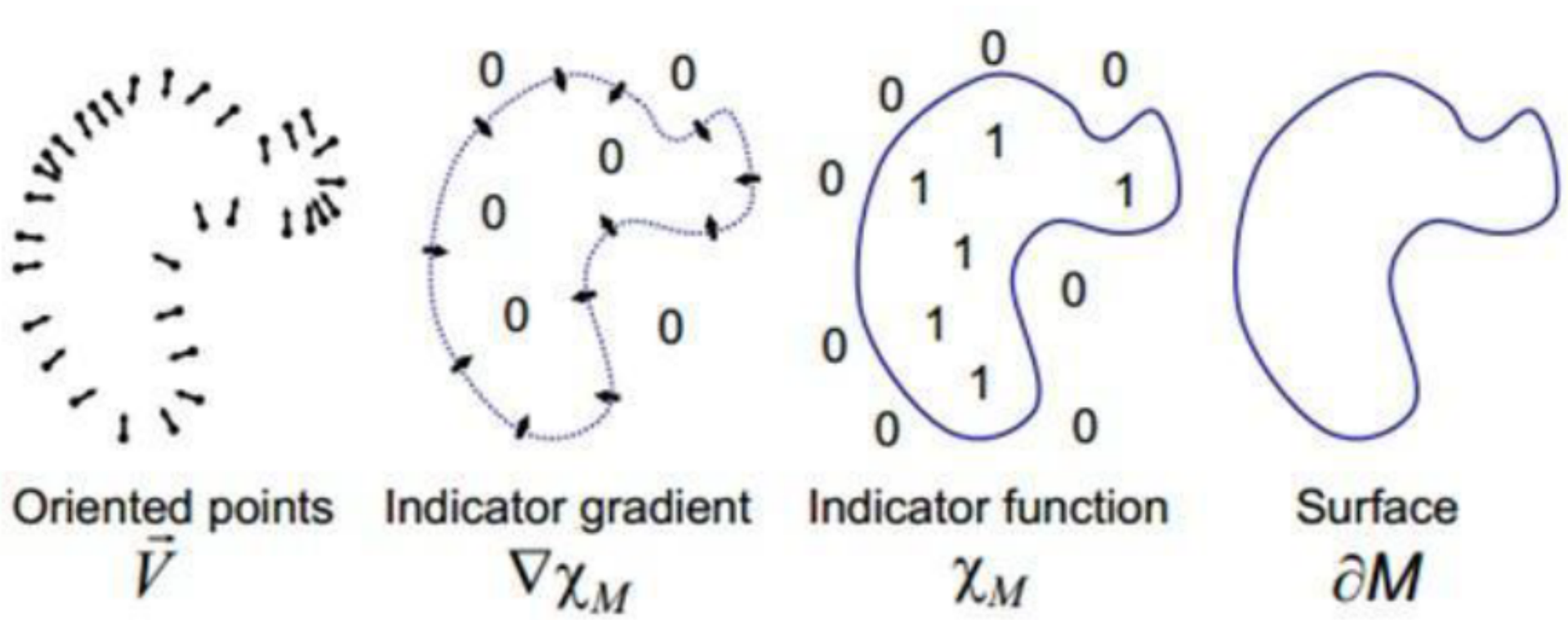
- Implicit -> mesh (Marching Cube)



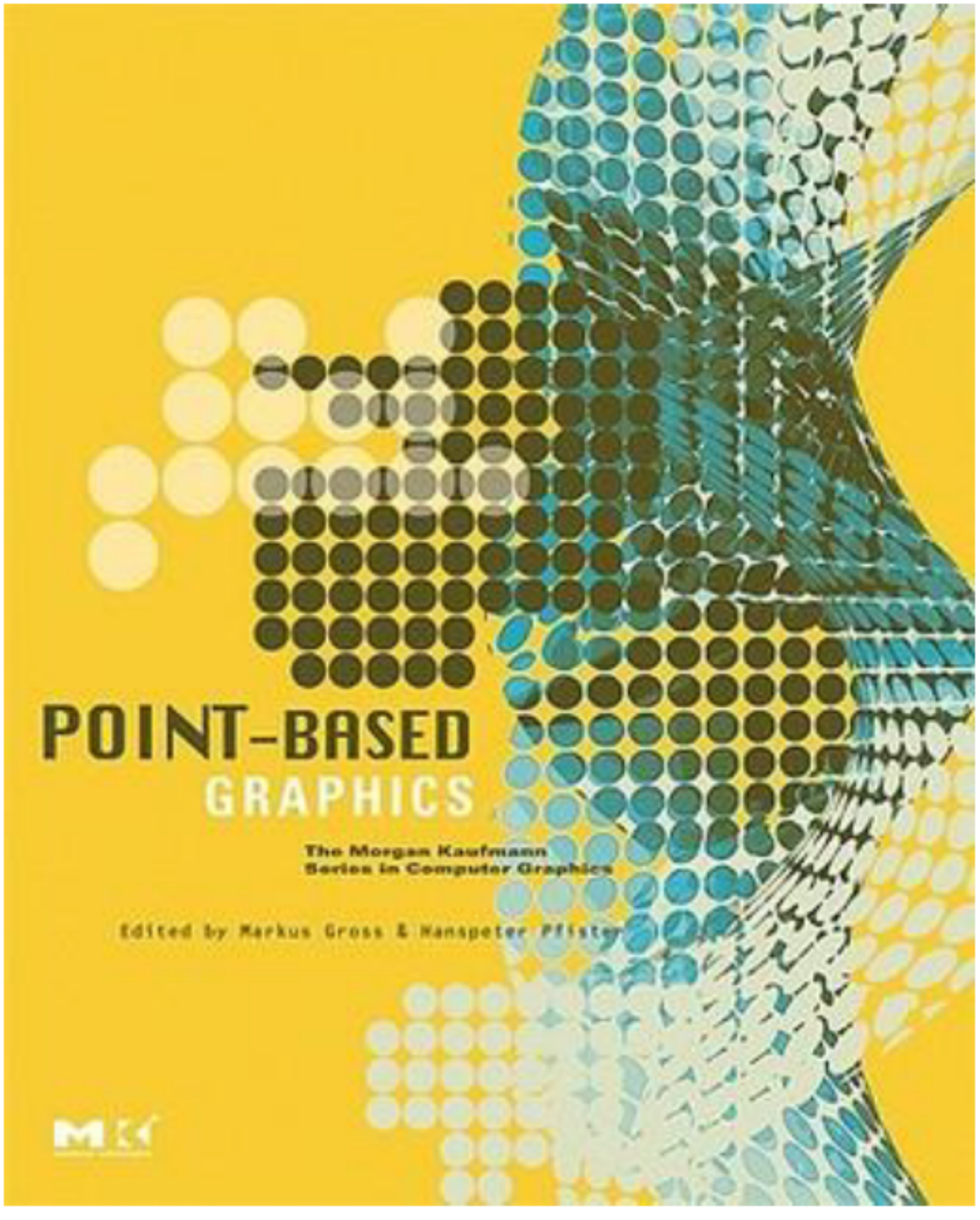
Conversion between different representations

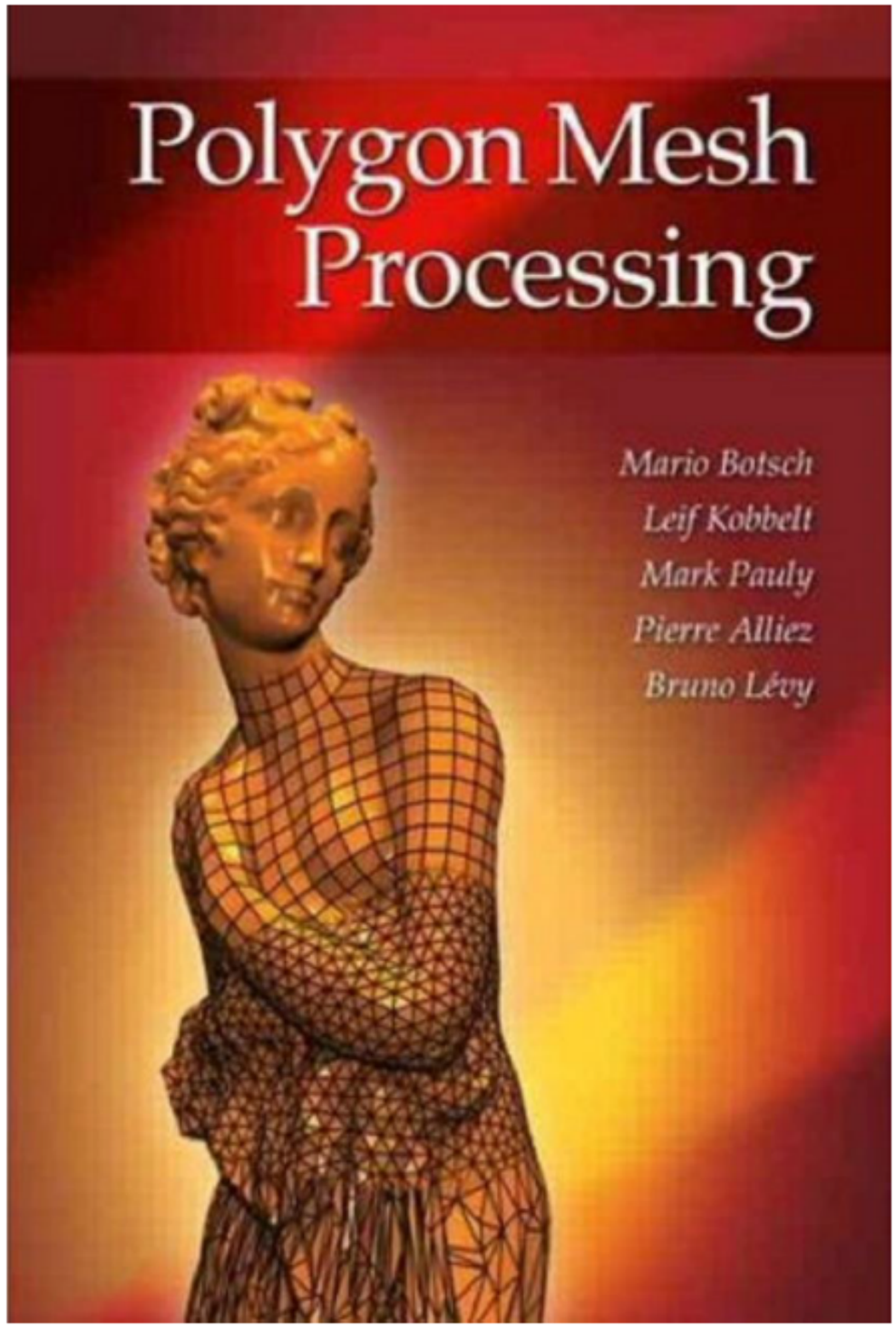
- Pointcloud -> Implicit -> Mesh

[Kazhdan et al. 06]



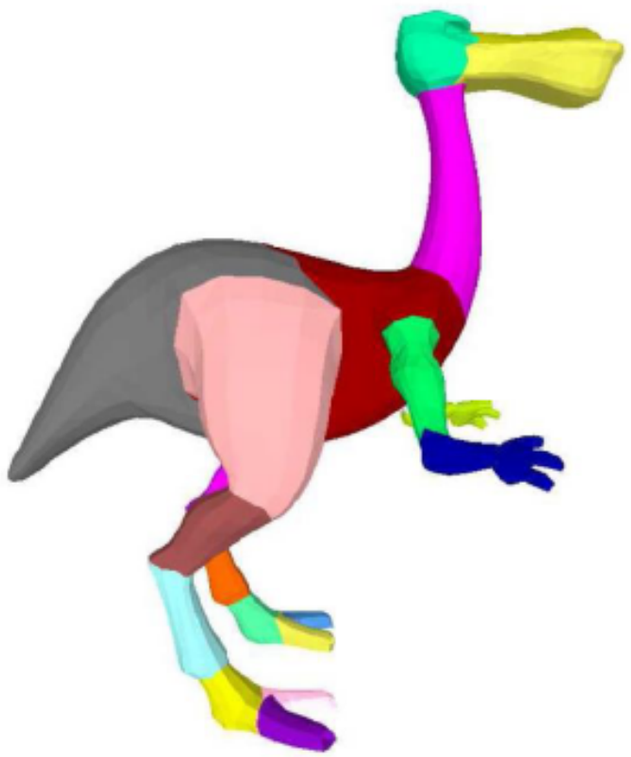
Two recommended books





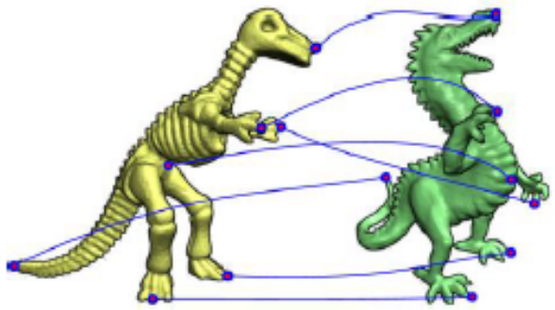
Topic III: How to understand 3D Data

- Design algorithms to extract semantic information from one or a collection of shapes



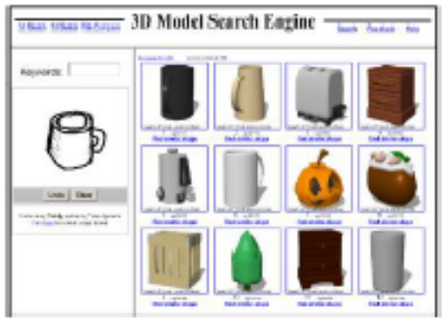
[Karz and Tal 03]

Segmentation



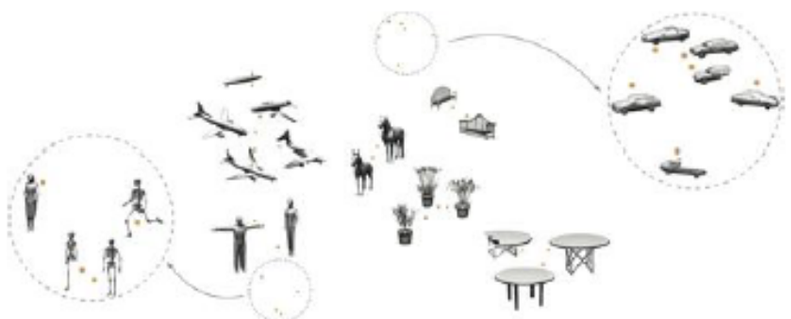
[van Kaick et al. 11]

Matching



[Funkhouser et al. 05]

Retrieval



[Mitra et al. 06]

Classification & Clustering

Another Rendering Course

- GAMES2XX
  - Unfortunately, GAMES3XX has been reserved for special topics
- Together with GAMES101 and GAMES202
  - A (hopefully helpful) computer graphics trilogy

Another Rendering Course

- GAMES2XX: Introduction to Offline Rendering / Advanced Image Synthesis
  - Part 1: Sampling and Light Transport
  - Part 2: Appearance Modeling
  - Part 3: State of the Art Research Topics

- Should be as easy / comfortable / enjoyable as this game 🐶🐶🐶

[Elden Ring, to appear]

GAMES: Graphics And Mixed Environment Symposium
图形学与混合现实在线平台

- 主页：**http://games-cn.org**
- 宗旨：图形学及相关领域交流的华人在线社区
- 在线直播活动：
  - 每周四晚8:00-9:30的在线报告（186期）
  - 专题：几何、绘制、模拟、视觉、可视化…
  - 课程：101（闫令琪）、201（胡渊鸣）、102（刘利刚）、202（闫令琪）
    - 已规划：203（黄其兴）、103（王华明）
- 在线交流微信群：16个群（7900+人）

所有资料（视频/PPT）云端保存，
总观看 100+ 万人次

加入微信群的方法：在微信中扫描右边的二维码，加games技术秘书为好友。然后回复“GAMES”即可获取群聊邀请。

Special Thanks to All of You!