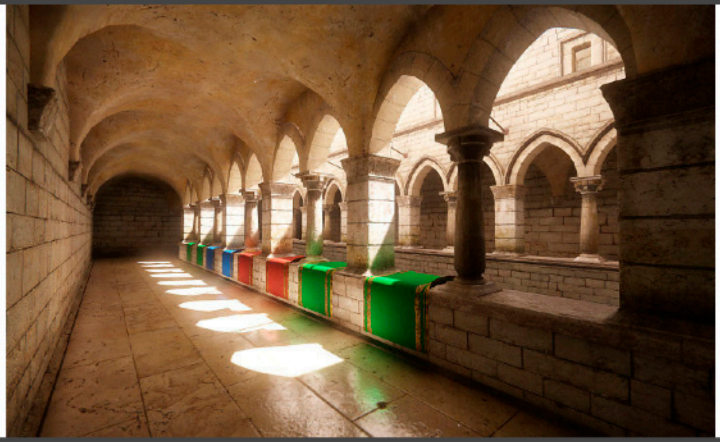
Real-Time High Quality Rendering

GAMES202, Lingqi Yan, UC Santa Barbara

Lecture 14:
A Glimpse of Industrial Solutions

Announcements

- GAMES101 resubmission has started
    - http://smartchair.org/GAMES101-Spring2021
- GAMES202 homework 4 & 5 will be released soon
- Course certification with my signature
    - Will be sent out in electronic version after all the resubmissions
    - Sign up for “Certification Request” (like a homework)

- Today: the last lecture of GAMES202!

Last Lectures

- Real-Time Ray Tracing (RTRT)
  - Basic idea
  - Temporal
    - Motion vector
    - Temporal accumulation / filtering
    - Temporal failures
  - Spatial
    - Implementing a spatial filter
    - Joint bilateral filtering
    - Outlier removal

Today

- Finishing up: specific filtering solutions for RTRT
  - Spatiotemporal Variance-Guided Filtering (SVGF)
  - Recurrent AutoEncoder (RAE)
- Practical Industrial solutions
  - Anti-aliasing
  - Super sampling and DLSS
  - Cascaded / multi-resolution solutions
  - /tiled/deferred shading，particles，engines

Specific Filtering Approaches for RTRT

SVGF — Basic Idea

- Spatiotemporal Variance-Guided Filtering [Schied et al.]
  - Very similar to the basic spatio-temporal denoising scheme
  - But with some additional variance analysis and tricks

[Spatiotemporal Variance-Guided Filtering]

SVGF — Joint Bilateral Filtering

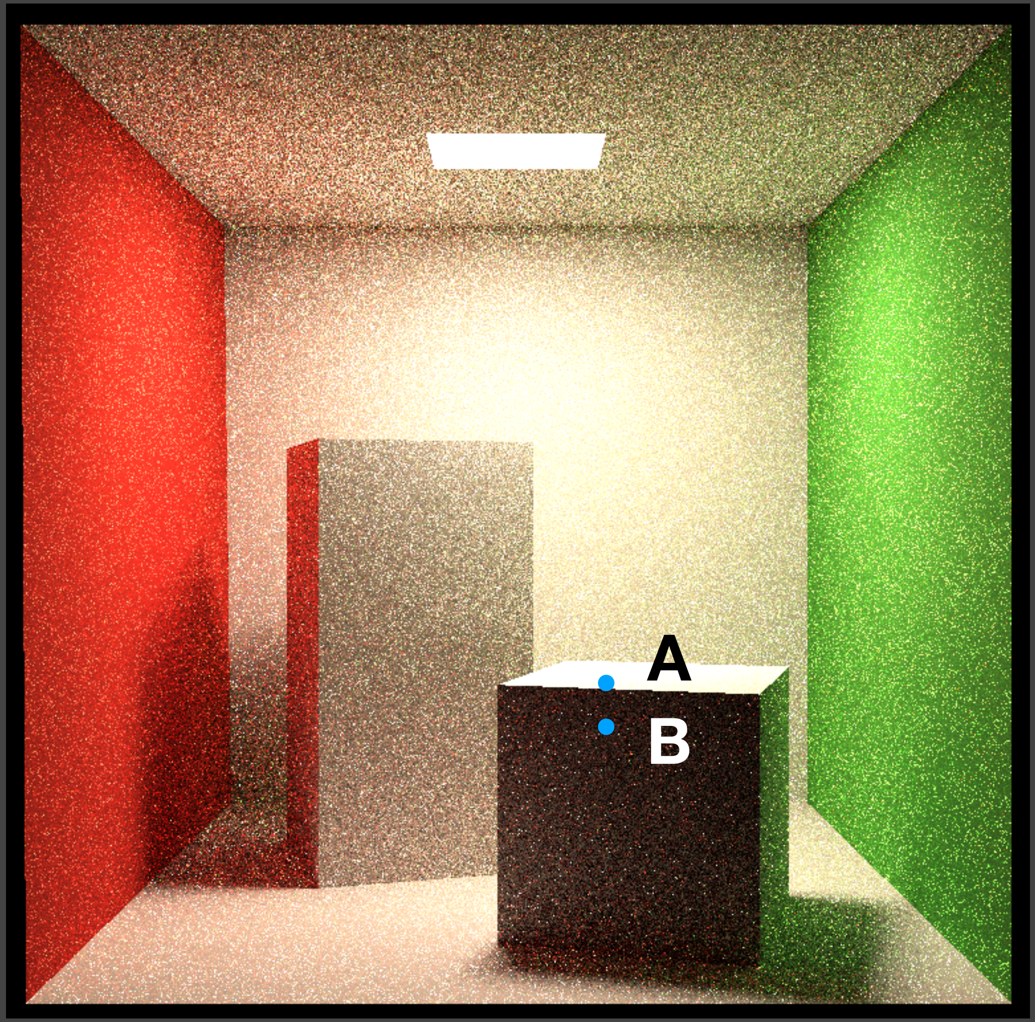
- 3 factors to guide filtering
  - **Depth**

$$w_z = \exp\left(-\frac{|z(p) - z(q)|}{\sigma_z |\nabla z(p) \cdot (p - q)| + \epsilon}\right)$$

  - Understanding:
    - A and B are on the same plane, of similar color, so they should contribute to each other
    - But the depth between A and B are very different!
    - Therefore, it is preferred to use the depth difference **w.r.t. the tangent plane**

SVGF — Joint Bilateral Filtering

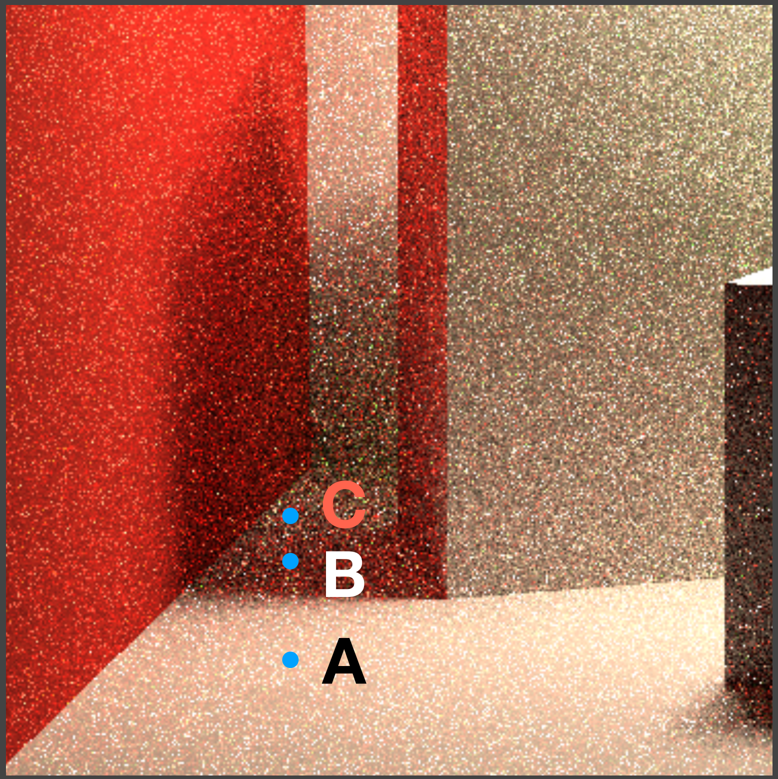
- 3 factors to guide filtering
  - **Normal**
    $w_n = \max(0, n(p) \cdot n(q))^{\sigma_n}$
  - Recall, does not have to be a Gaussian
  - Note: in case normal maps exist, use macro normals



SVGF — Joint Bilateral Filtering

- 3 factors to guide filtering
  - **Luminance** (grayscale color value)

$$w_l = \exp\left(-\frac{|l_i(p) - l_i(q)|}{\sigma_l\sqrt{g_{3\times3}(\mathrm{Var}(l_i(p)))} + \epsilon}\right)$$



- Variance
  - Calculate spatially in 7x7
  - Also averaged over time using motion vectors
  - Take another 3x3 spatial filter before use

SVGF — Results

SVGF — Results

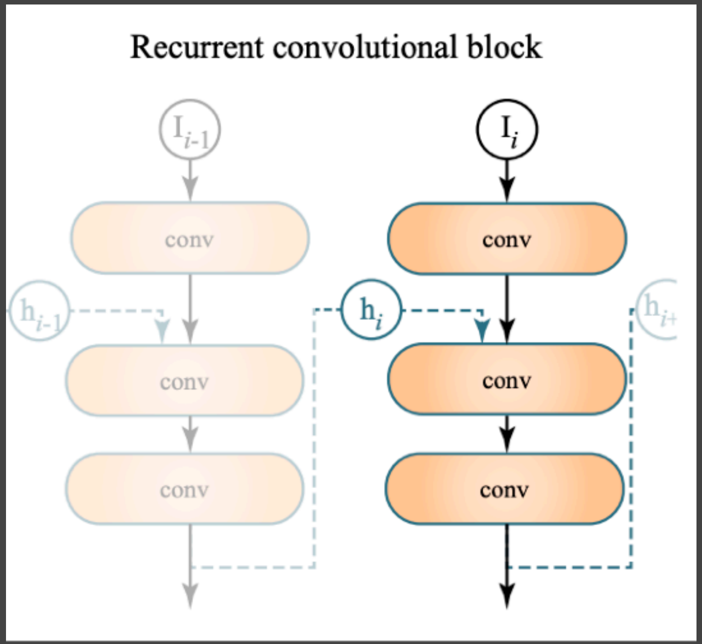
SVGF — Failure Cases

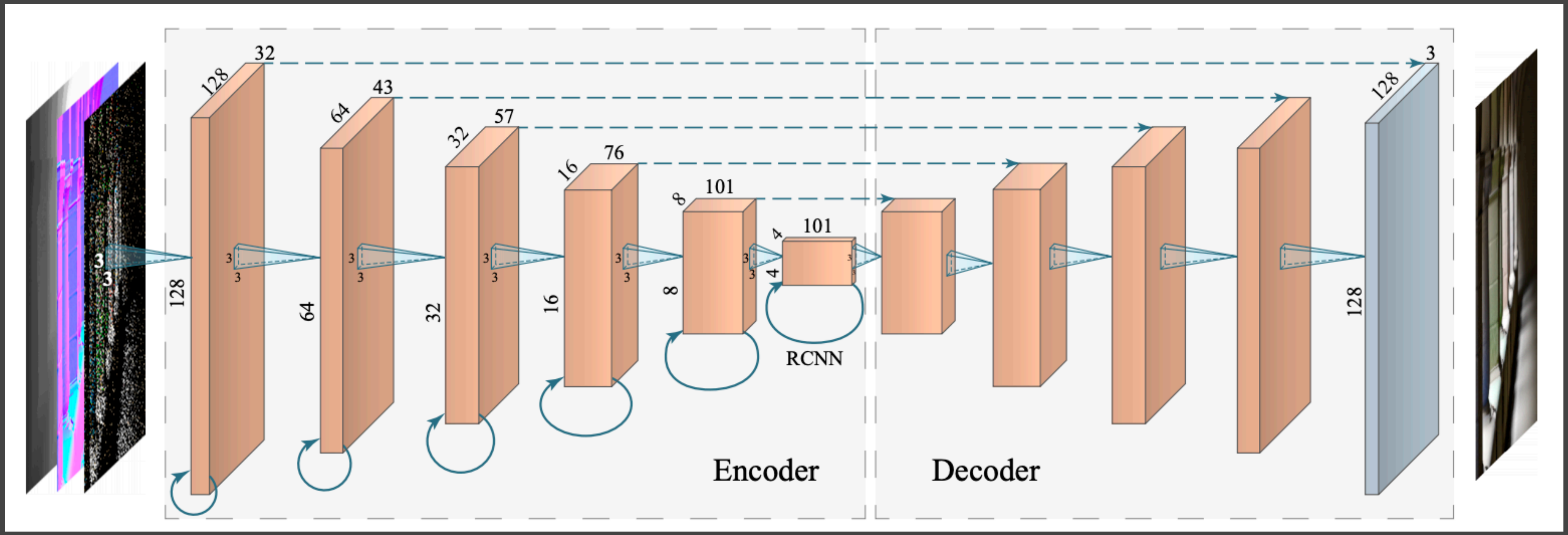
RAE — Basic Idea

- Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent denoising AutoEncoder [Chaitanya et al.]
  - A post-processing network that does denoising (noisy -> clean)
  - With the help of G-buffers
  - The network automatically performs temporal accumulation
- Key architecture design
  - AutoEncoder (or U-Net) structure
  - Recurrent convolutional block

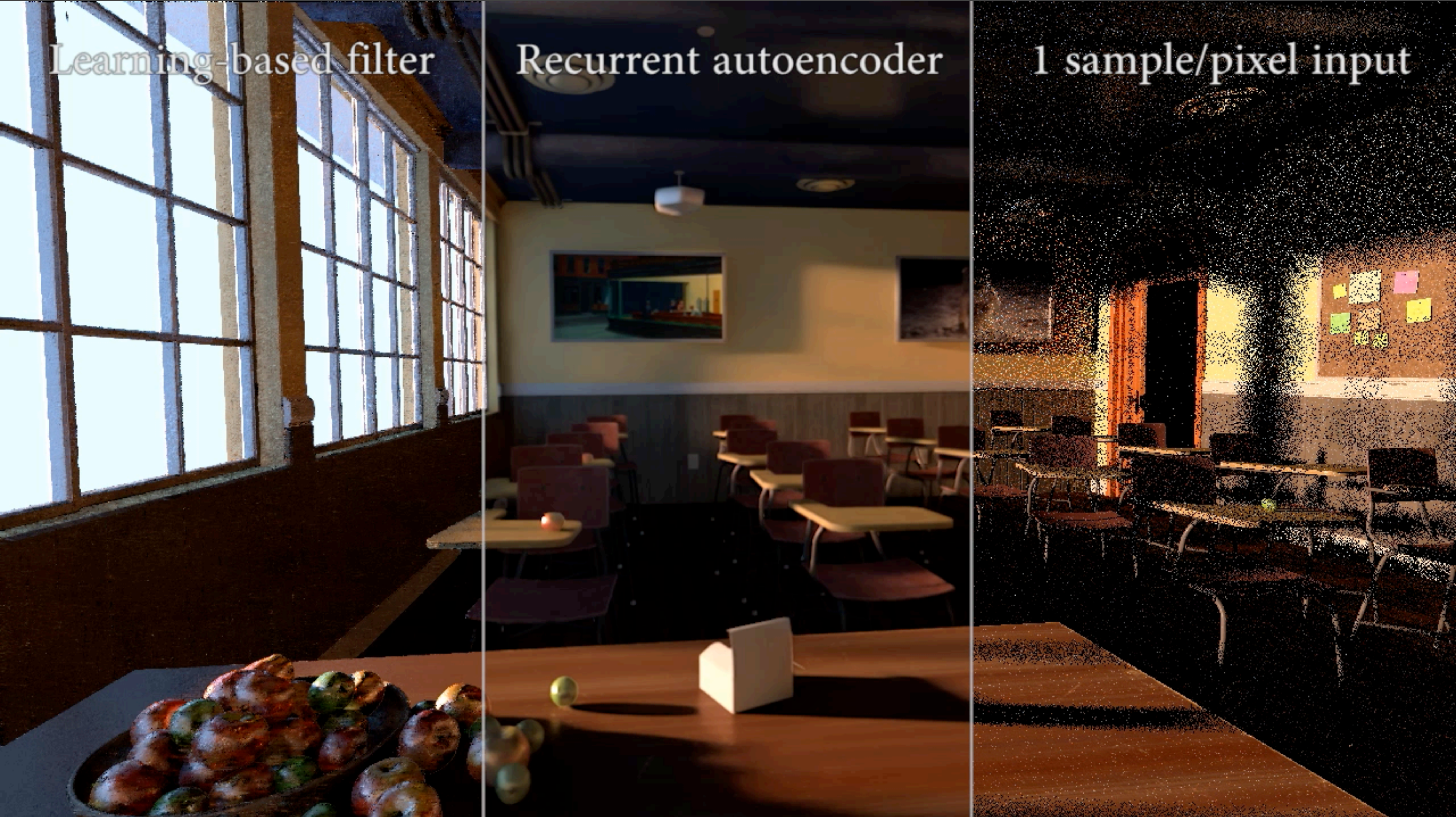
RAE — Architecture

- AutoEncoder
  - Skip / residual connections for faster and better training
- Recurrent block
  - Accumulates (and gradually forgets) information from previous frames





RAE — Results


Learning-based filter
Recurrent autoencoder
1 sample/pixel input

RAE — Results

Comparison

	Quality	Artifact	Performance	Explanability	Where did the paper go
SVGF	Clean	Ghosting	Fast	Yes	HPG
RAE (when first invented)	Overblur	Ghosting	Slow	No	SIGGRAPH

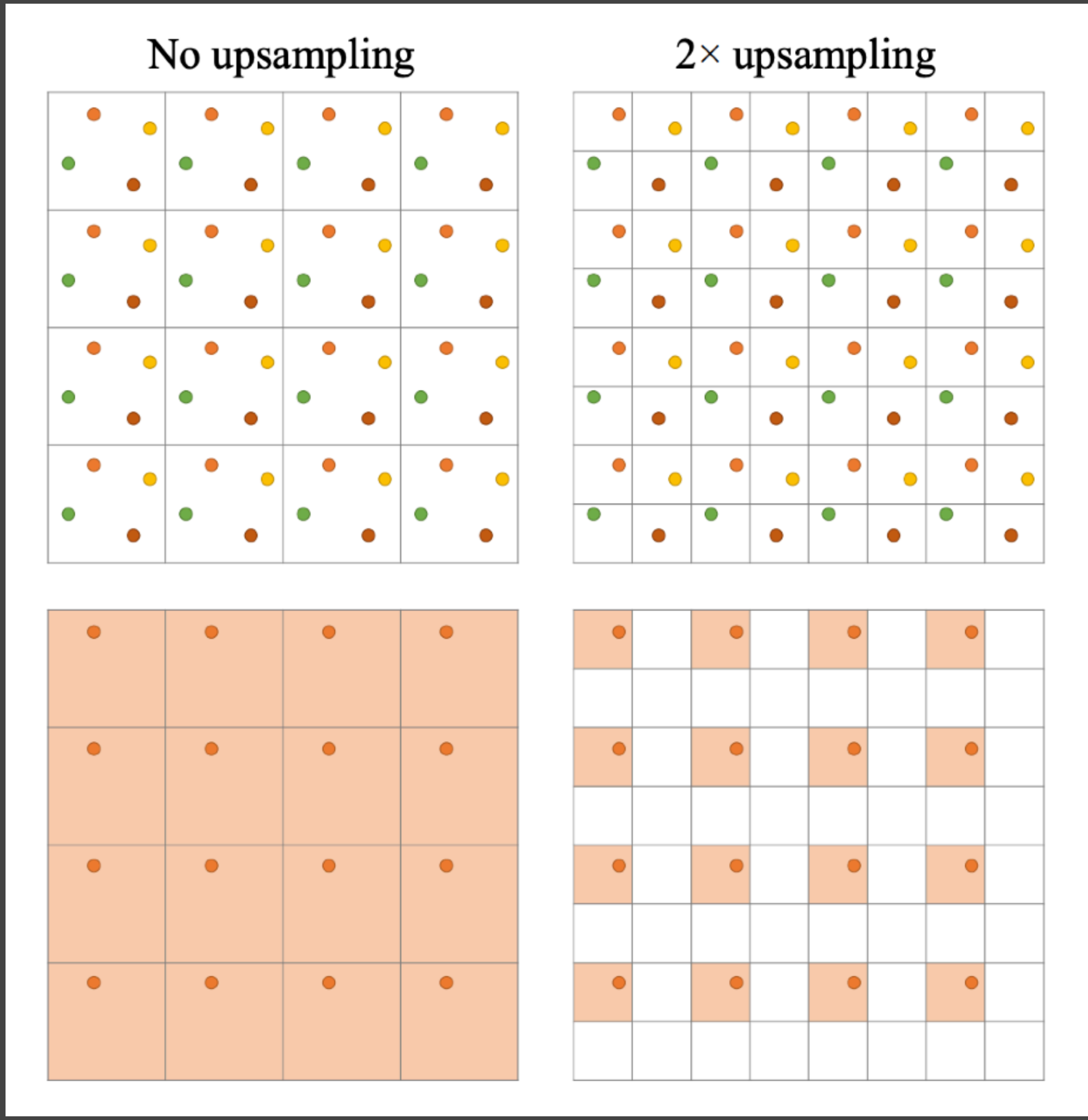
Questions?

Practical Industrial solutions

(Still, from the scientific perspective)

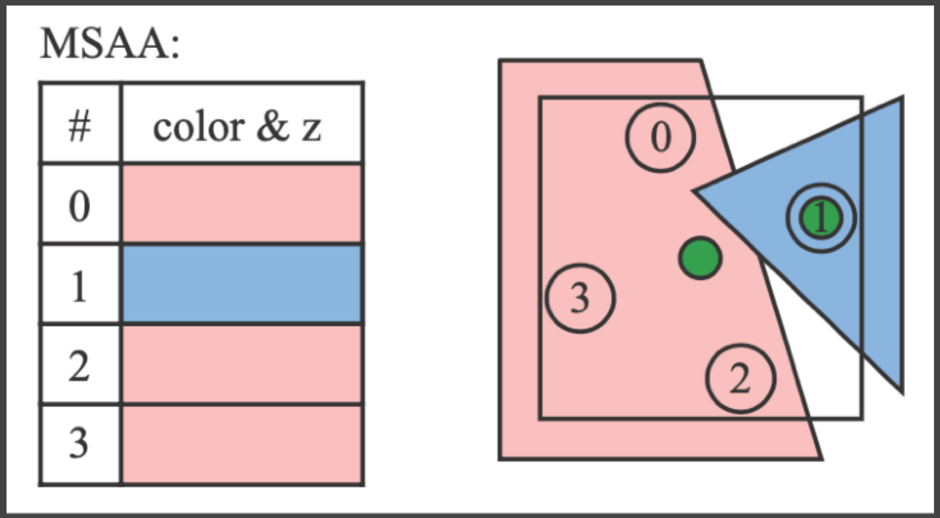
Temporal Anti-Aliasing (TAA)

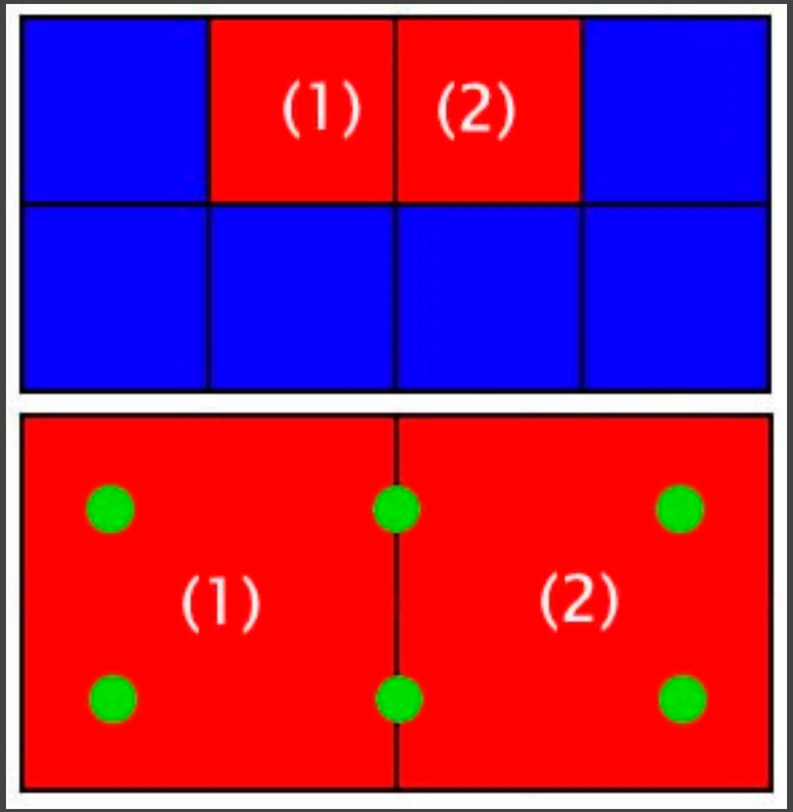
- Recall: why aliasing?
  - Not enough samples per pixel during rasterization
  - Therefore, the ultimate solution is to use more samples
- Temporal Anti-Aliasing
  - Distributing / reuse samples across frames (time)
  - Almost exactly the same as in RTRT



Notes on Anti-Aliasing

- Additional note 1
  - MSAA (Multisample) vs SSAA (Supersampling)
- SSAA is straightforward
  - Rendering at a larger resolution, then downsample
  - The ultimate solution, but costly
- MSAA: an improvement on performance
  - The same primitive is shaded only once
  - Reuse samples across pixels

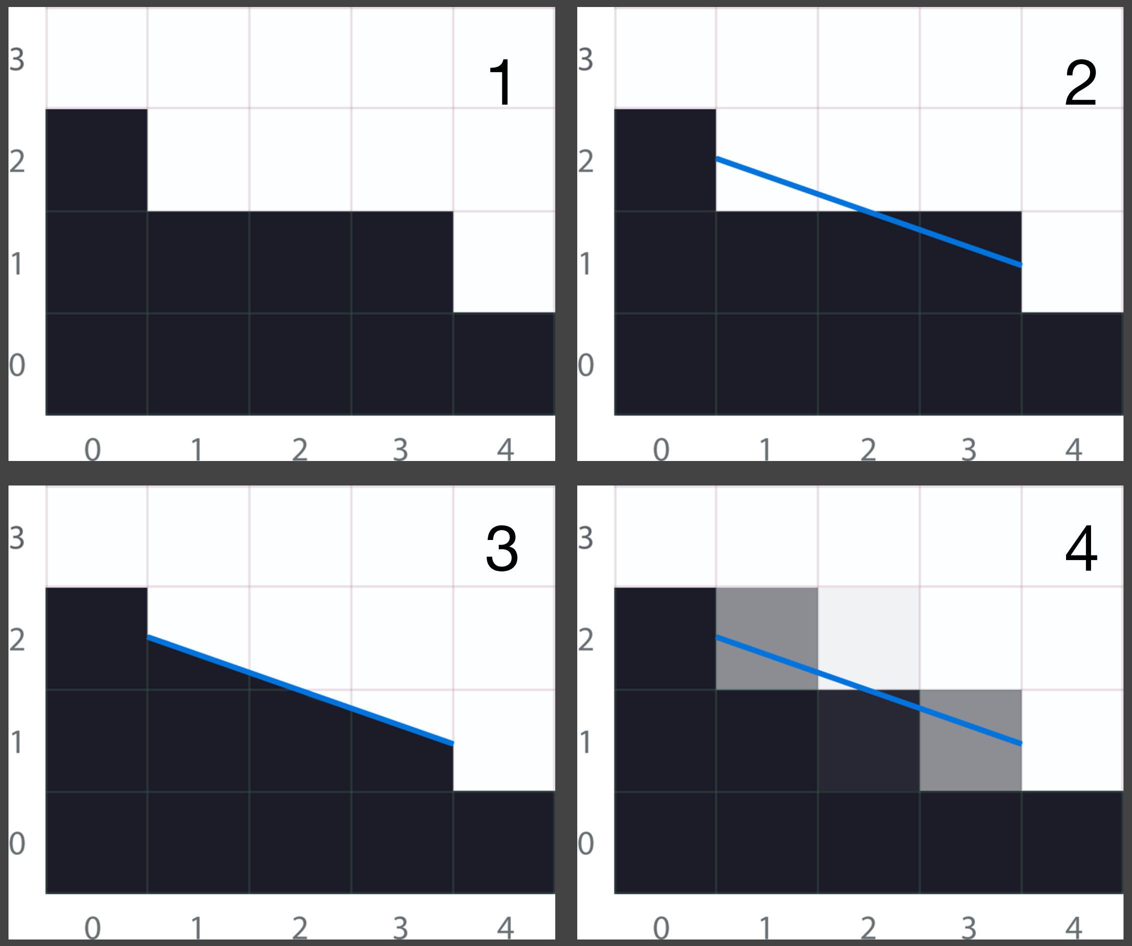




https://www.sapphirenation.net/anti-aliasing-comparison-performance-quality

Notes on Anti-Aliasing

- Additional note 2
  - State of the art image based anti-aliasing solution
  - SMAA (Enhanced subpixel morphological AA)
  - History: FXAA -> MLAA (Morphological AA) -> SMAA
- Additional note 3
  - G-buffers should never be anti-aliased!



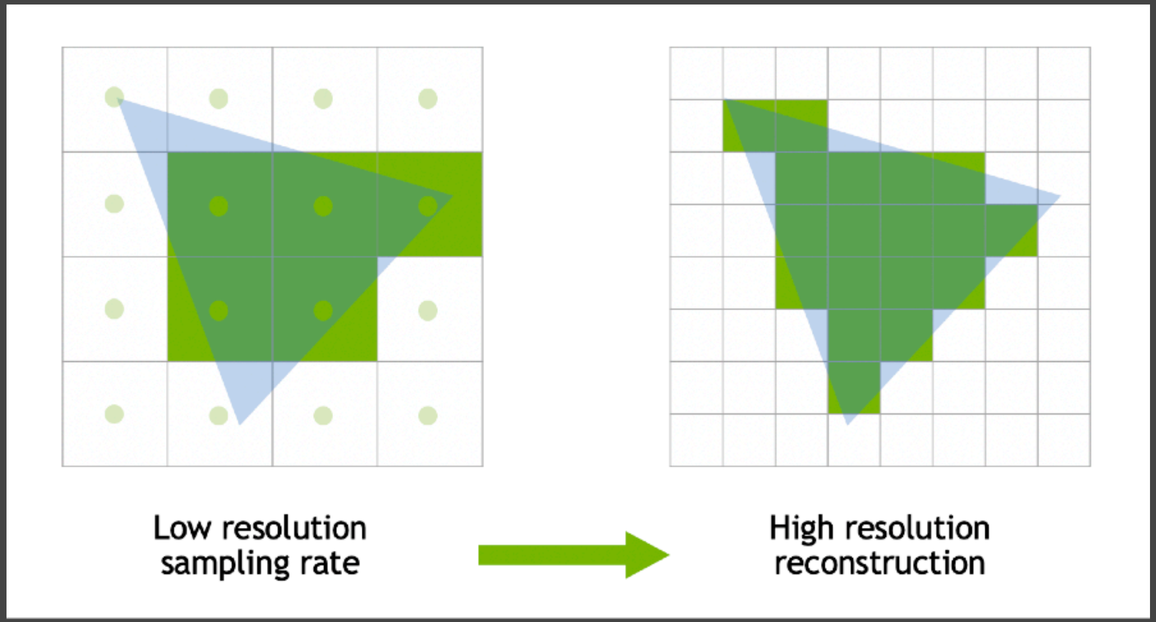
http://www.iryoku.com/smaa/

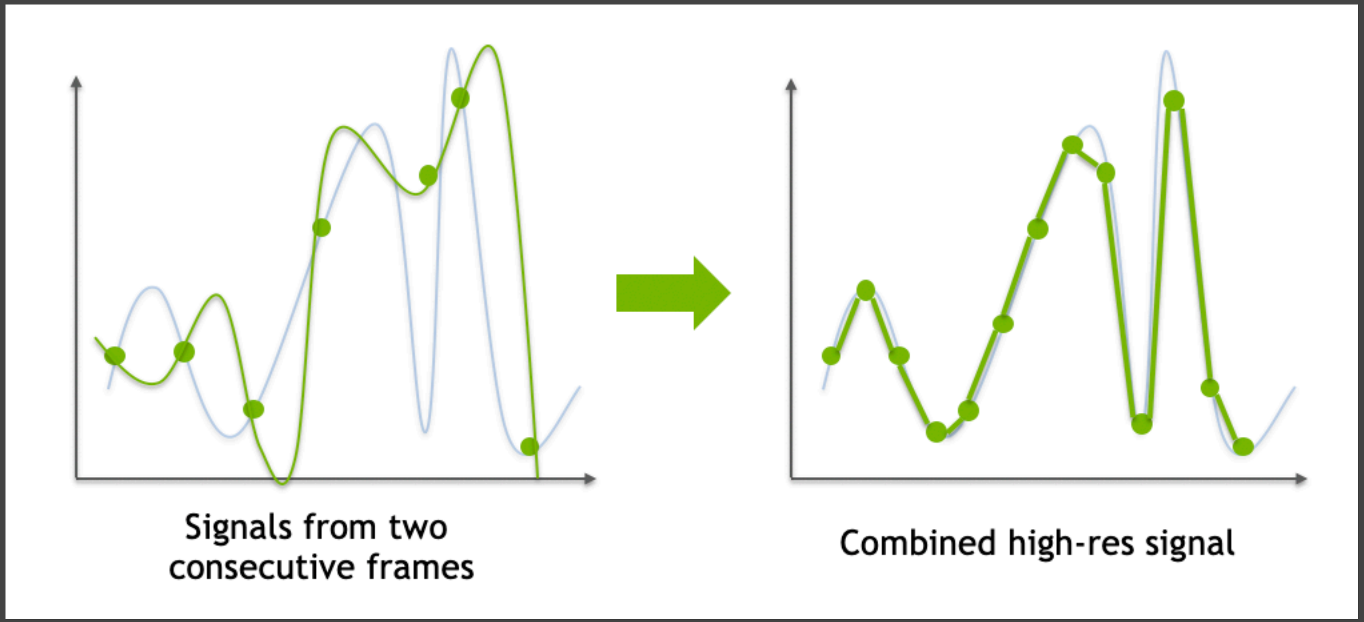
Temporal Super Resolution

- Super resolution (or super sampling)
  - Literal understanding: increasing resolution
  - Source 1 (DLSS 1.0): out of nowhere / completely guessed
  - Source 2 (DLSS 2.0): from temporal information
- Key idea of Deep Learning Super Sampling (DLSS) 2.0
  - Yet another TAA-like application
  - Temporally reuse samples to increase resolution

DLSS 2.0

- Main problem
  - Upon temporal failure, clamping is no longer an option
  - Because we need a clear value for each smaller pixel
  - Therefore, key is how to use temporal info smarter than clamping





DLSS 2.0

DLSS 2.0

DLSS 2.0

DLSS 2.0

- An importance practical issue
  - If DLSS itself runs at 30ms per frame, it's dead already
  - Network inference performance optimization (classified)
- Counterpart of DLSS
  - By AMD: FidelityFX Super Resolution
  - By Facebook: Neural Supersampling for Real-time Rendering [Xiao et al.]
- Any future work?
  - Also classified
  - But wish me good luck in SIGGRAPH Asia 2021

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?

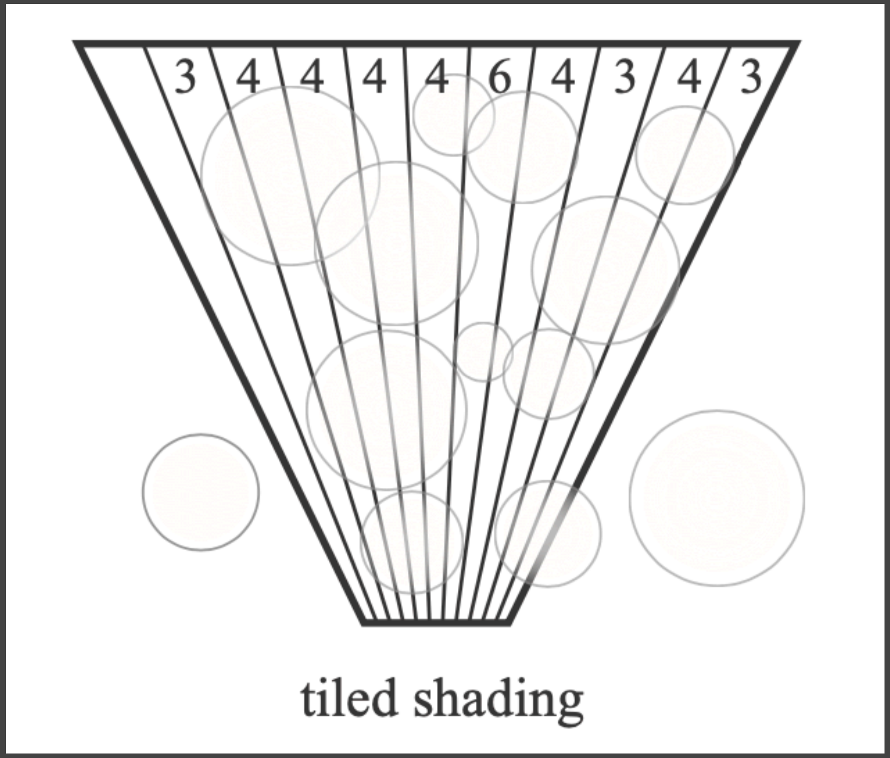
Deferred Shading

- 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

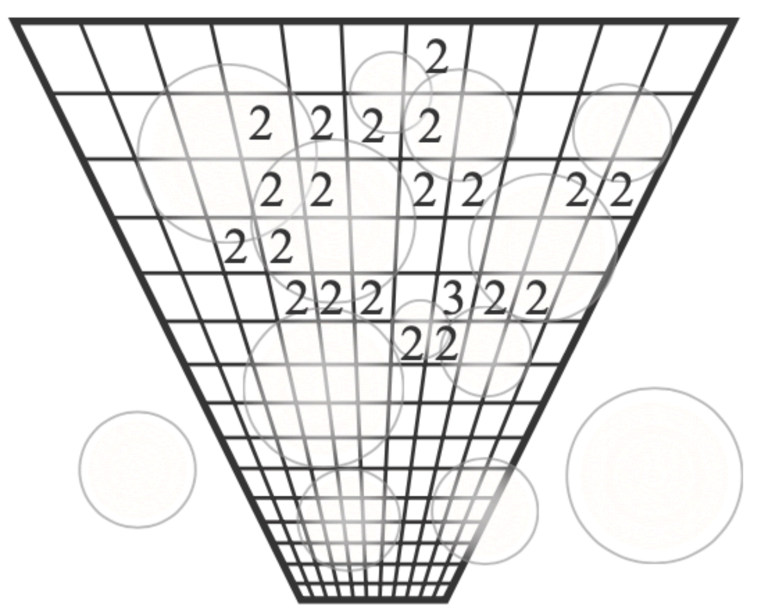
- Improvement: tiled shading
  - Subdivide the screen into tiles of e.g. 32x32 then shade each

- 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)



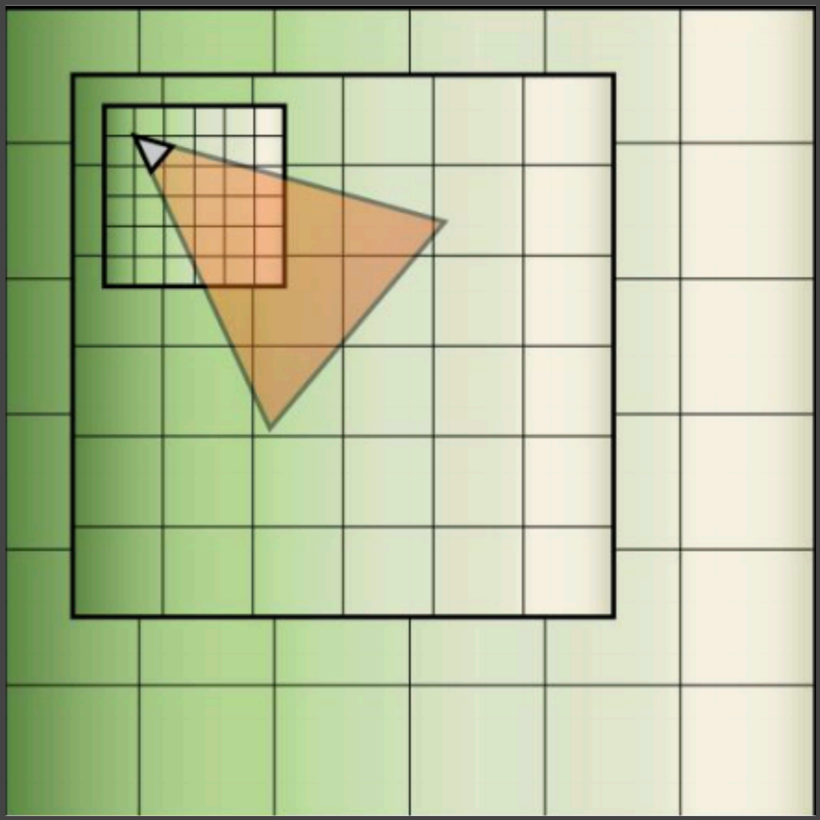
clustered shading

Level of Detail Solutions

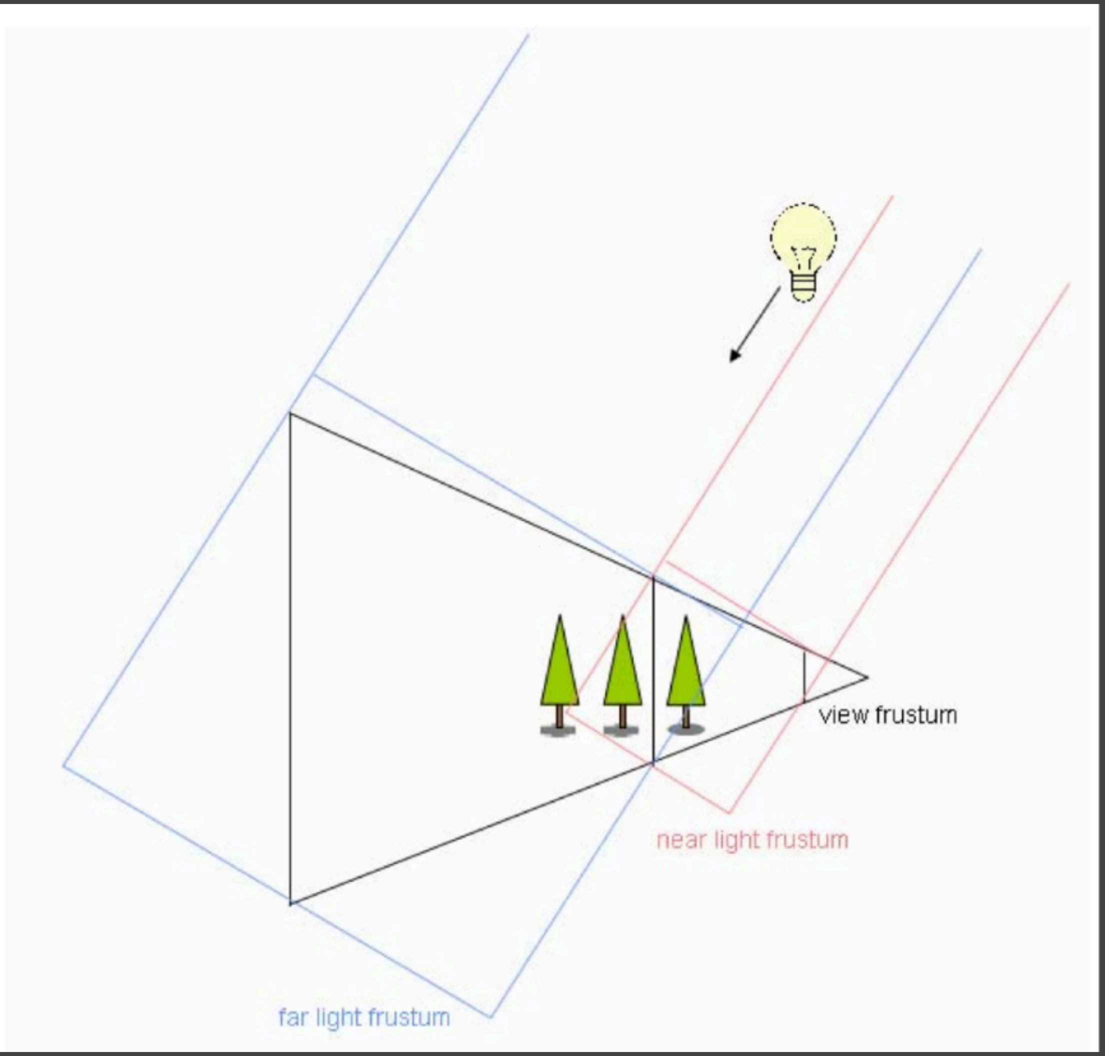
- Level of Detail (LoD) is very important
  - Recall: texture MIPMAP-ing
  - Choosing the right level of detail to use can save computation
- The use of multiple levels of detail
  - Often called “cascaded” by the RTR industry

Level of Detail Solutions

- Example
  - Cascaded shadow maps
  - Cascaded LPV



[Anton Kaplanyan, Light Propagation Volumes in CryEngine 3]



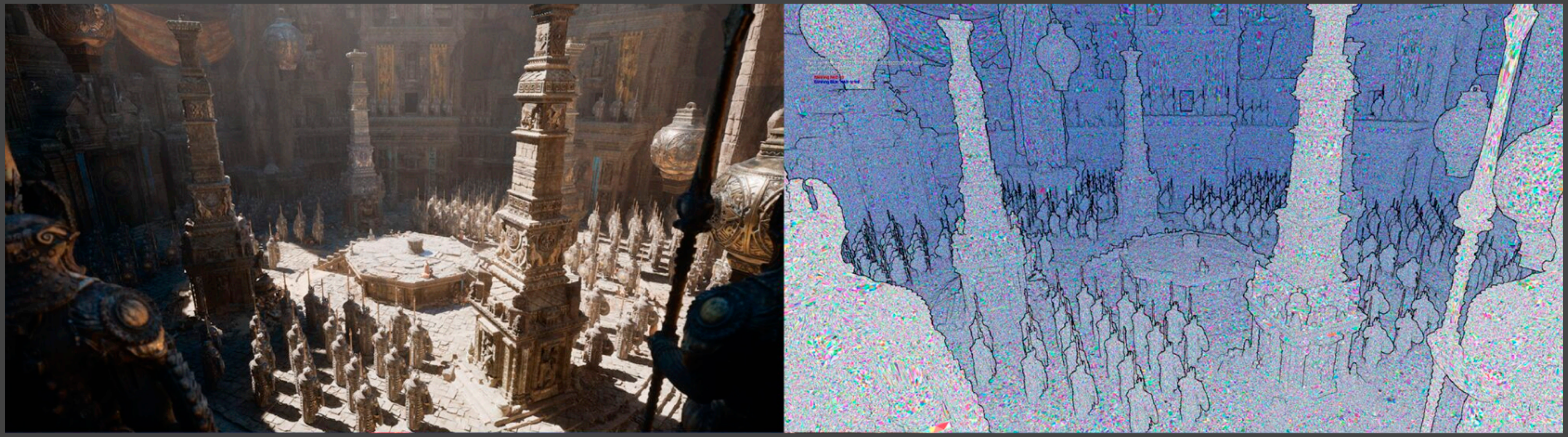
[Dimitrov et al., Cascaded Shadow Maps]

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!

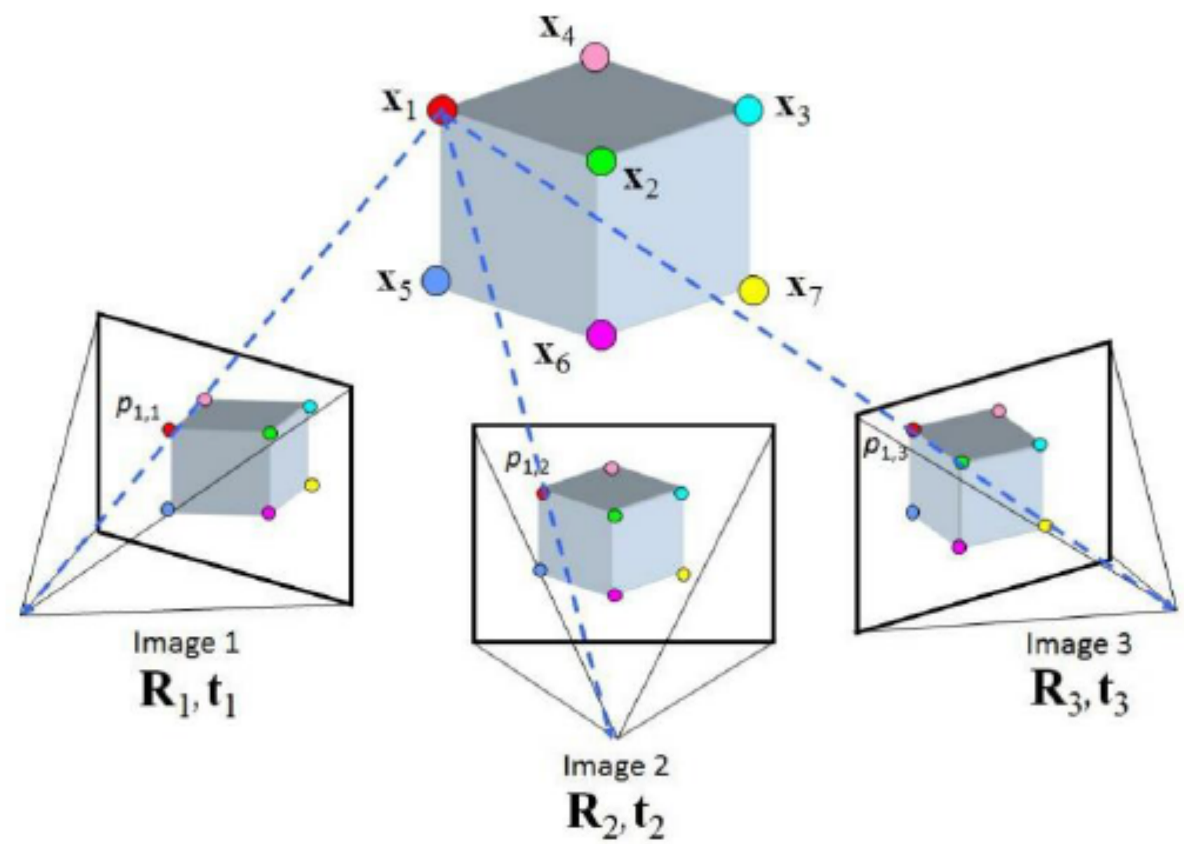
Advertisements

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

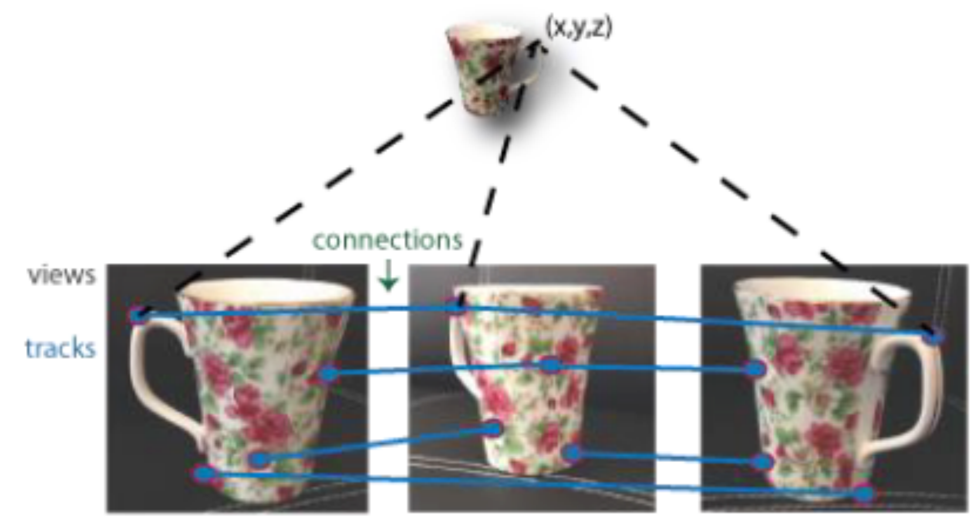
Topic I: 3D Reconstruction

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

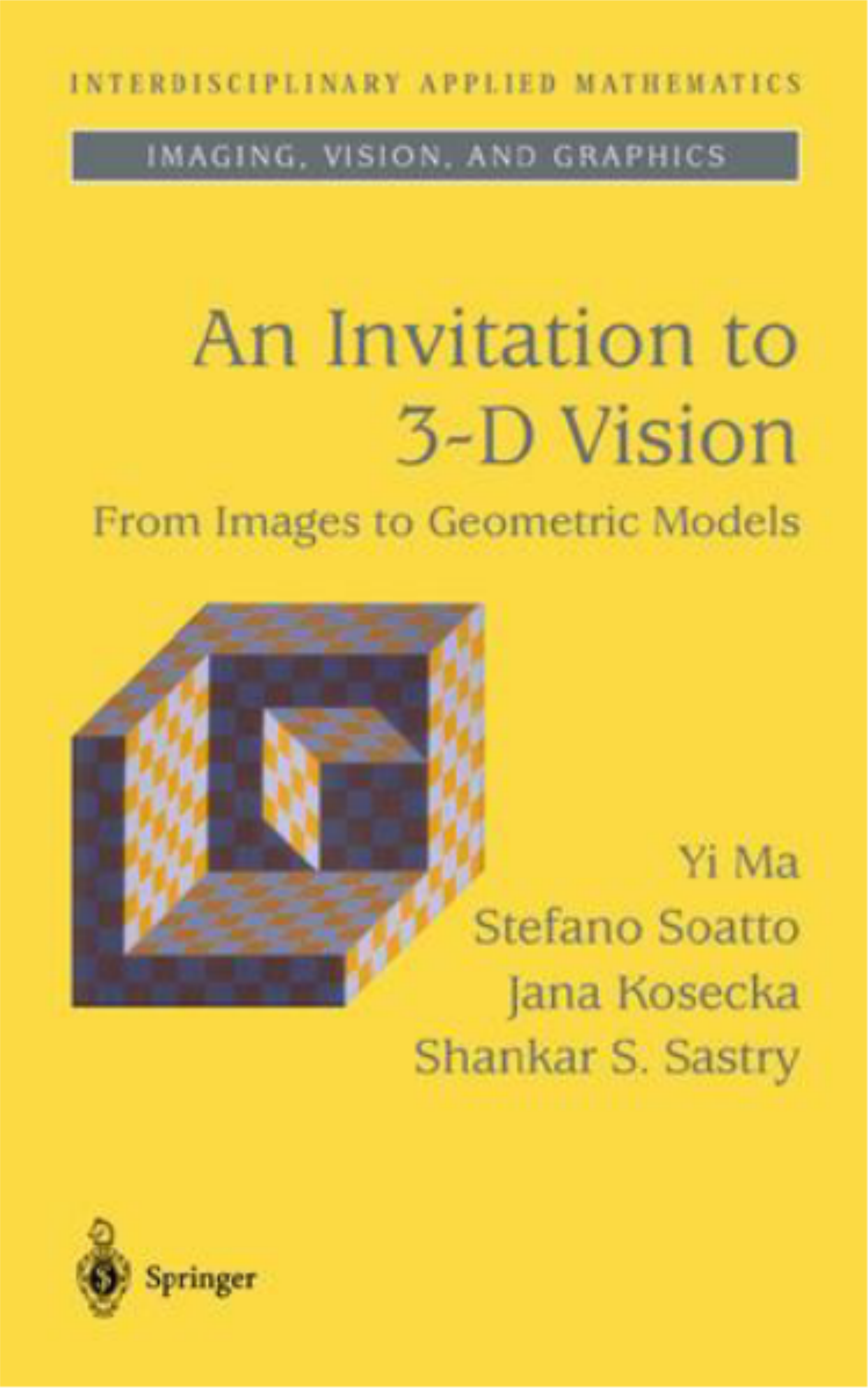


Topic I: 3D Reconstruction

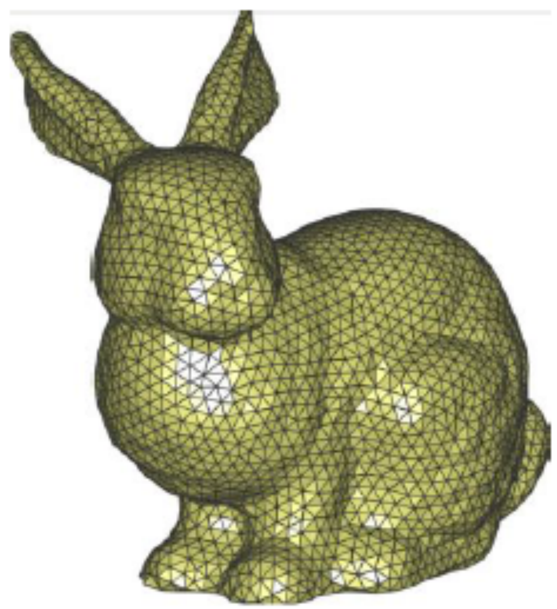
- Algorithms
  - Feature extraction
  - Feature correspondences
  - Relative camera pose
  - Structure-from-motion
  - Multiview stereo
  - Bundle adjustment
  - ICP



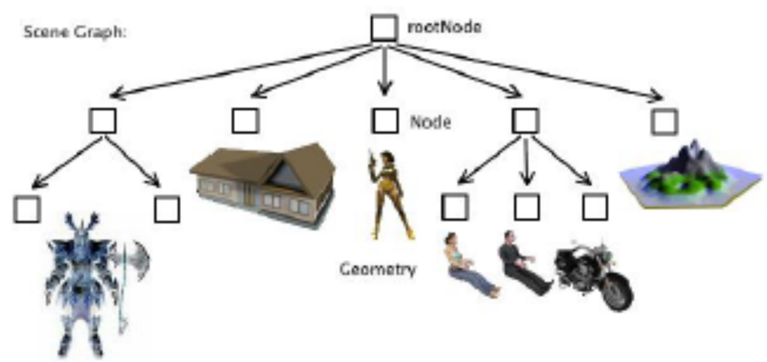
Textbook


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

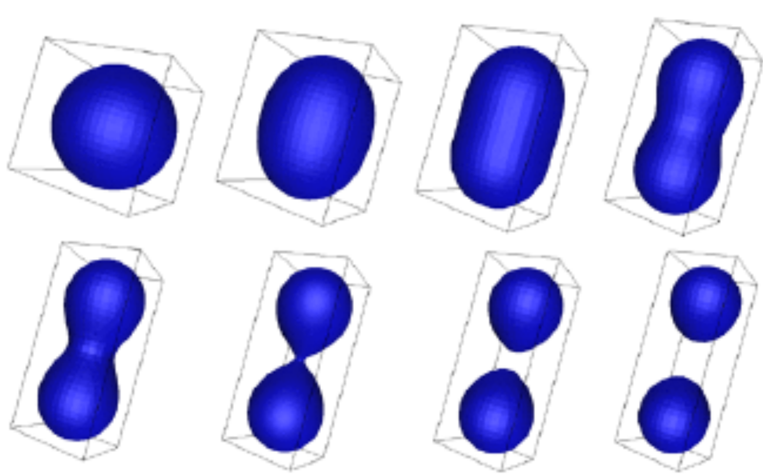
Topic II: How to represent 3D Data



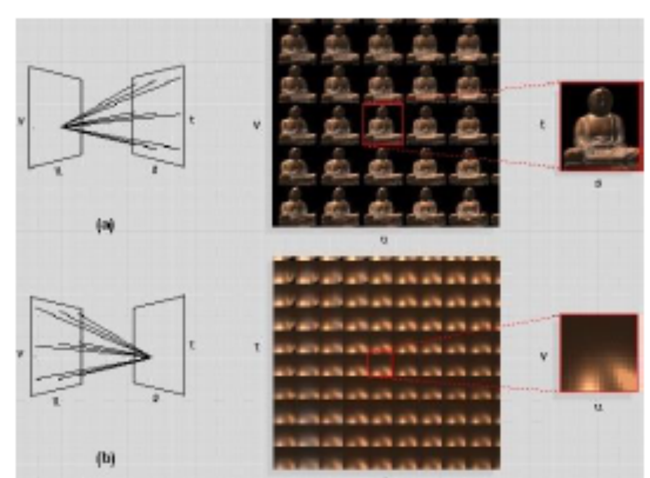
Triangular mesh



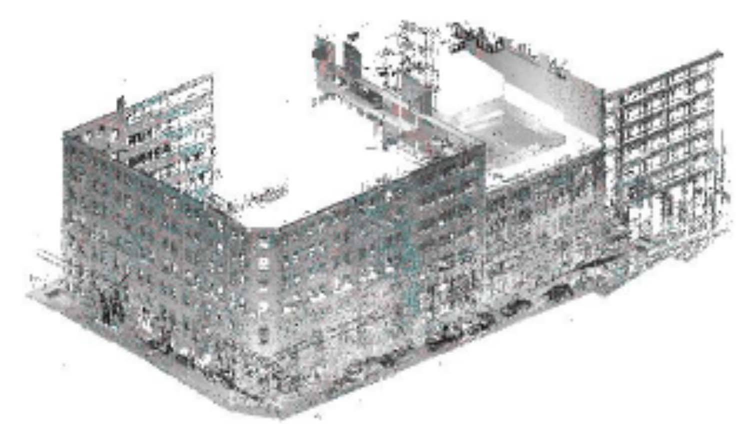
Part-based models



Implicit surface



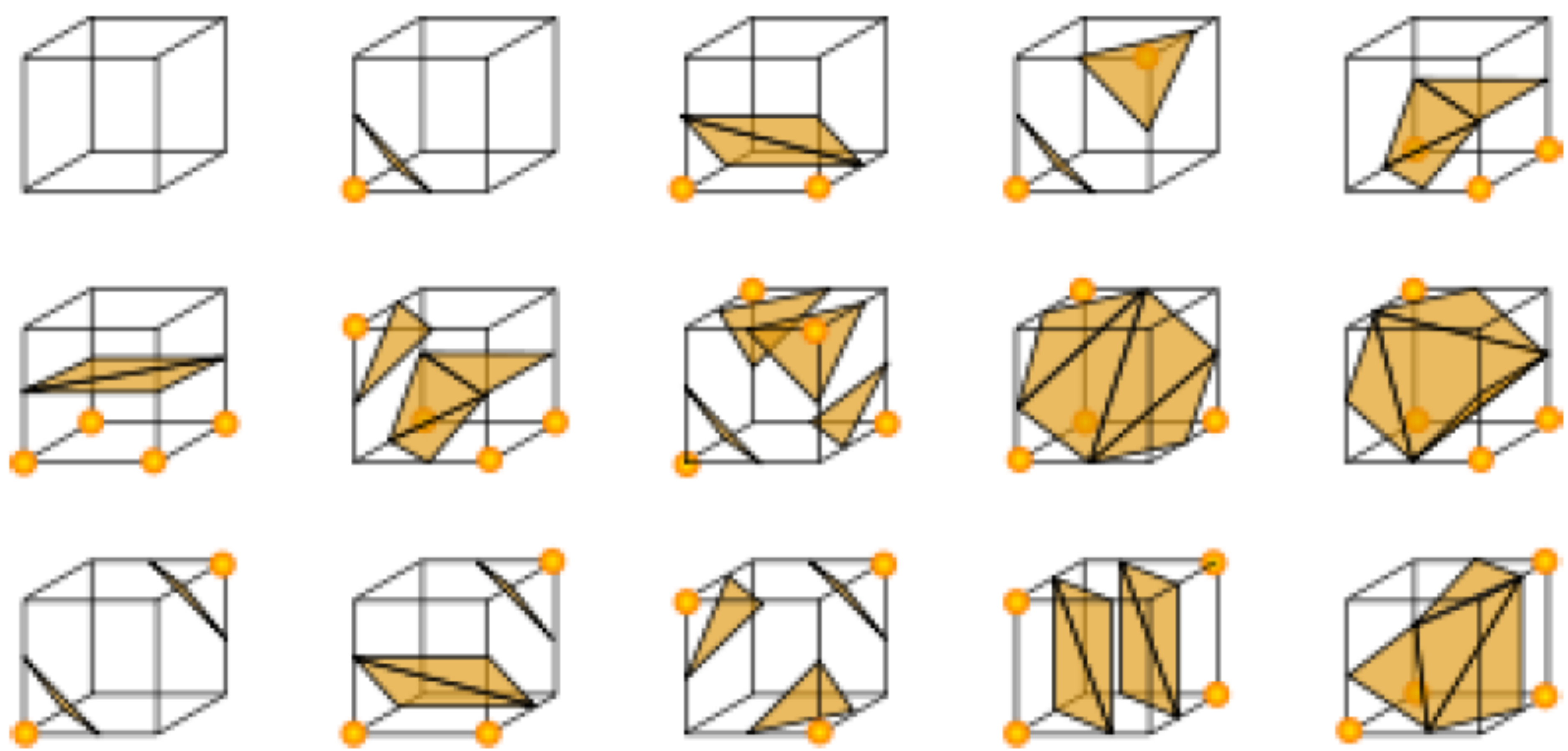
Light Field Representation



Point cloud

Conversion between different representations

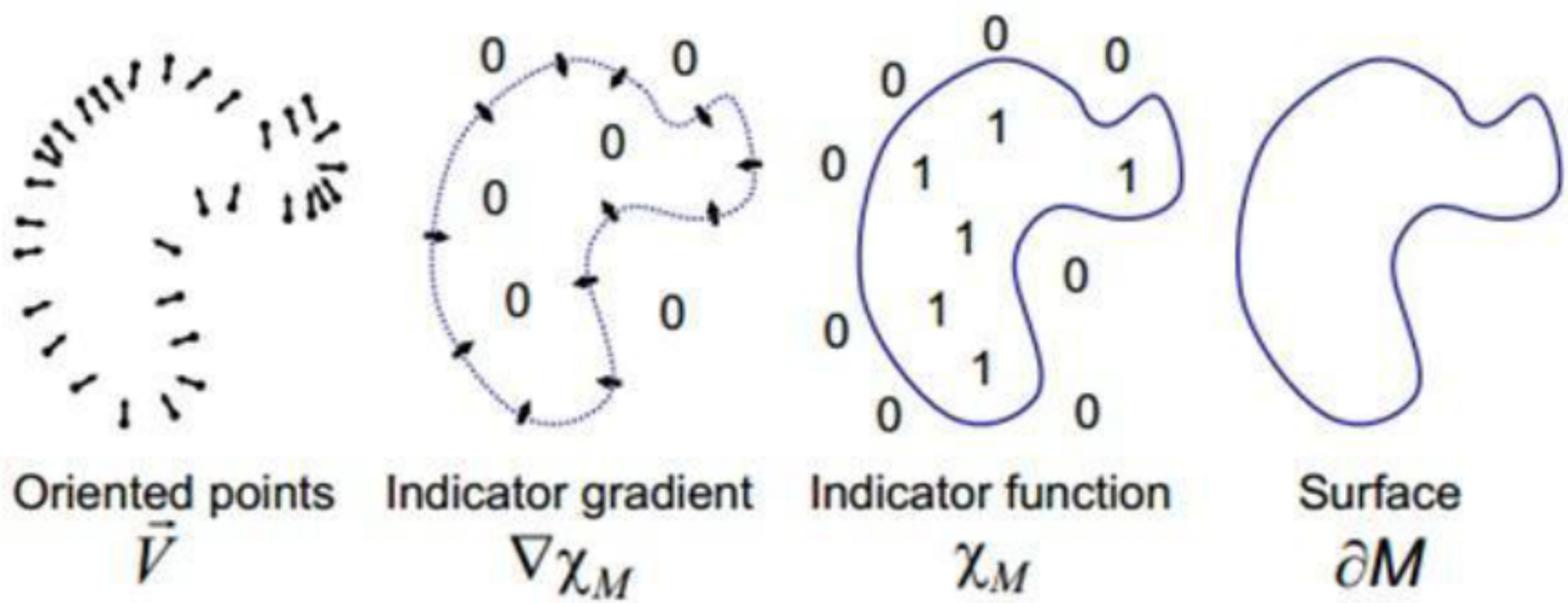
- Implicit -> mesh (Marching Cube)



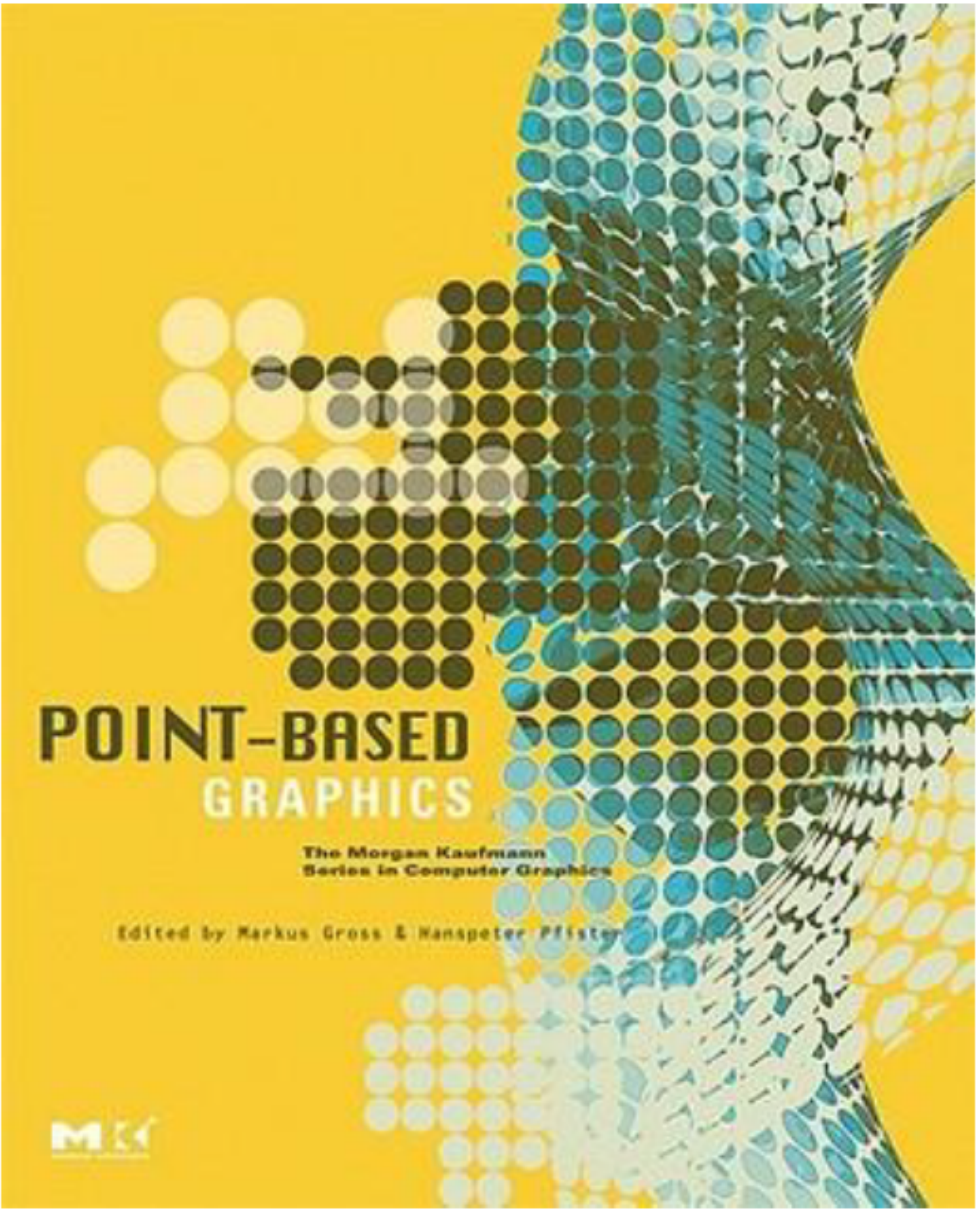
Conversion between different representations

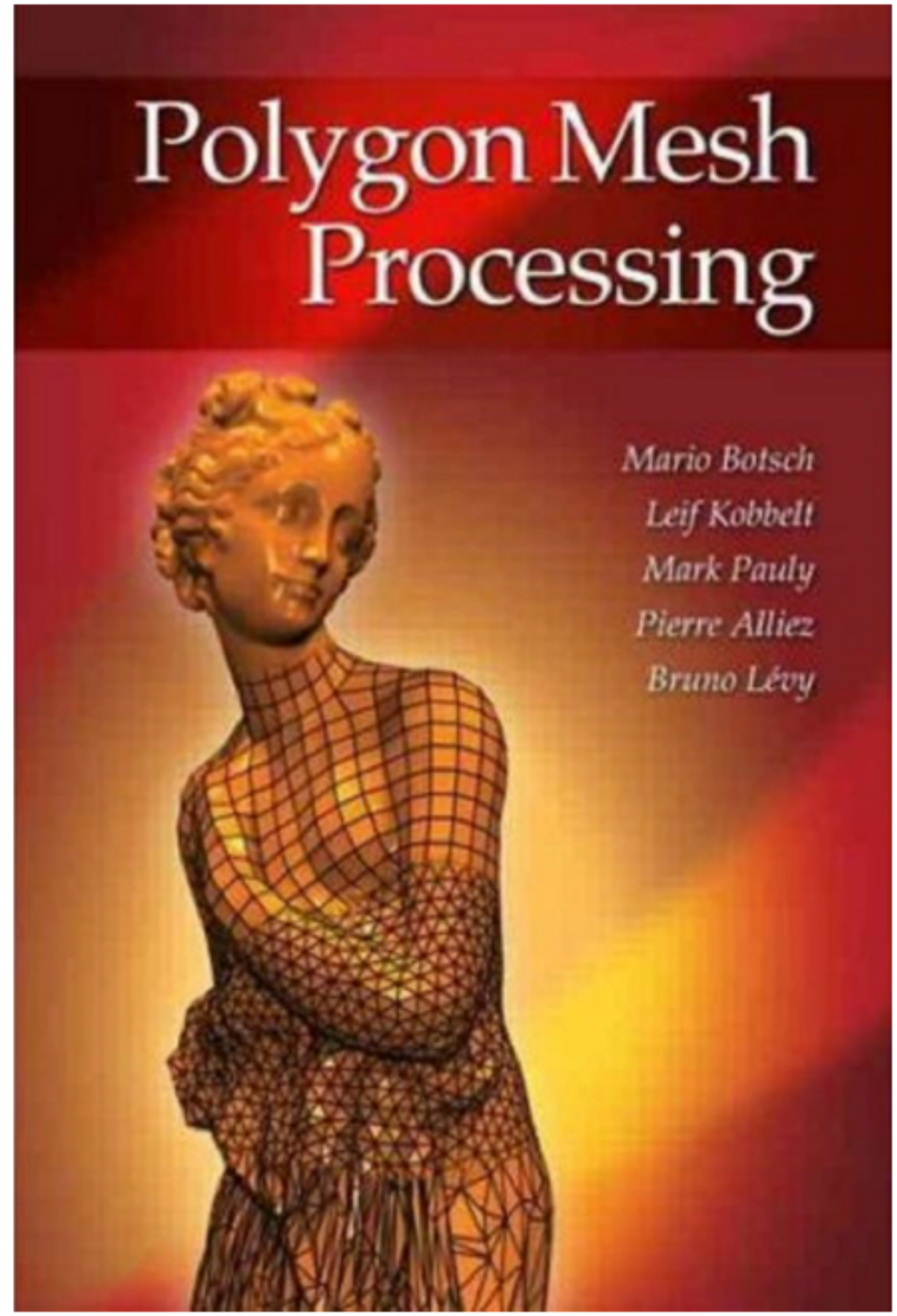
- Pointcloud -> Implicit -> Mesh

[Kazhdan et al. 06]



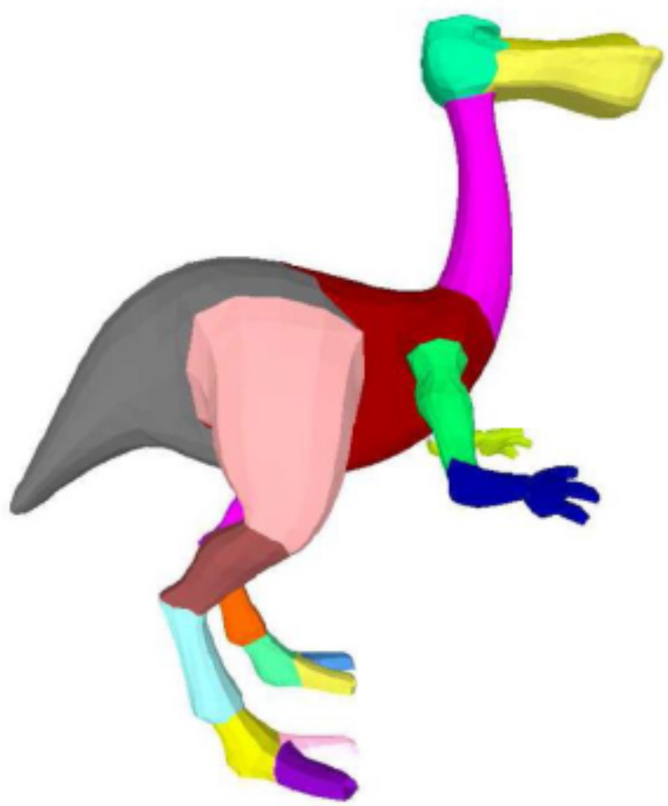
Two recommended books





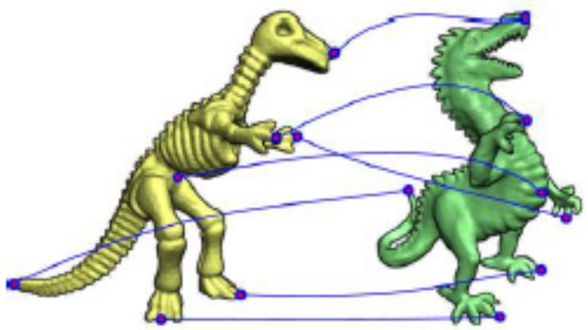
Topic III: How to understand 3D Data

- 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）

所有资料（视频/PPT）云端保存，总观看 100+ 万人次

- 在线交流微信群：16个群（7900+人）

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!