COL781: Computer Graphics

39. Conclusion

Henrik Wann Jensen

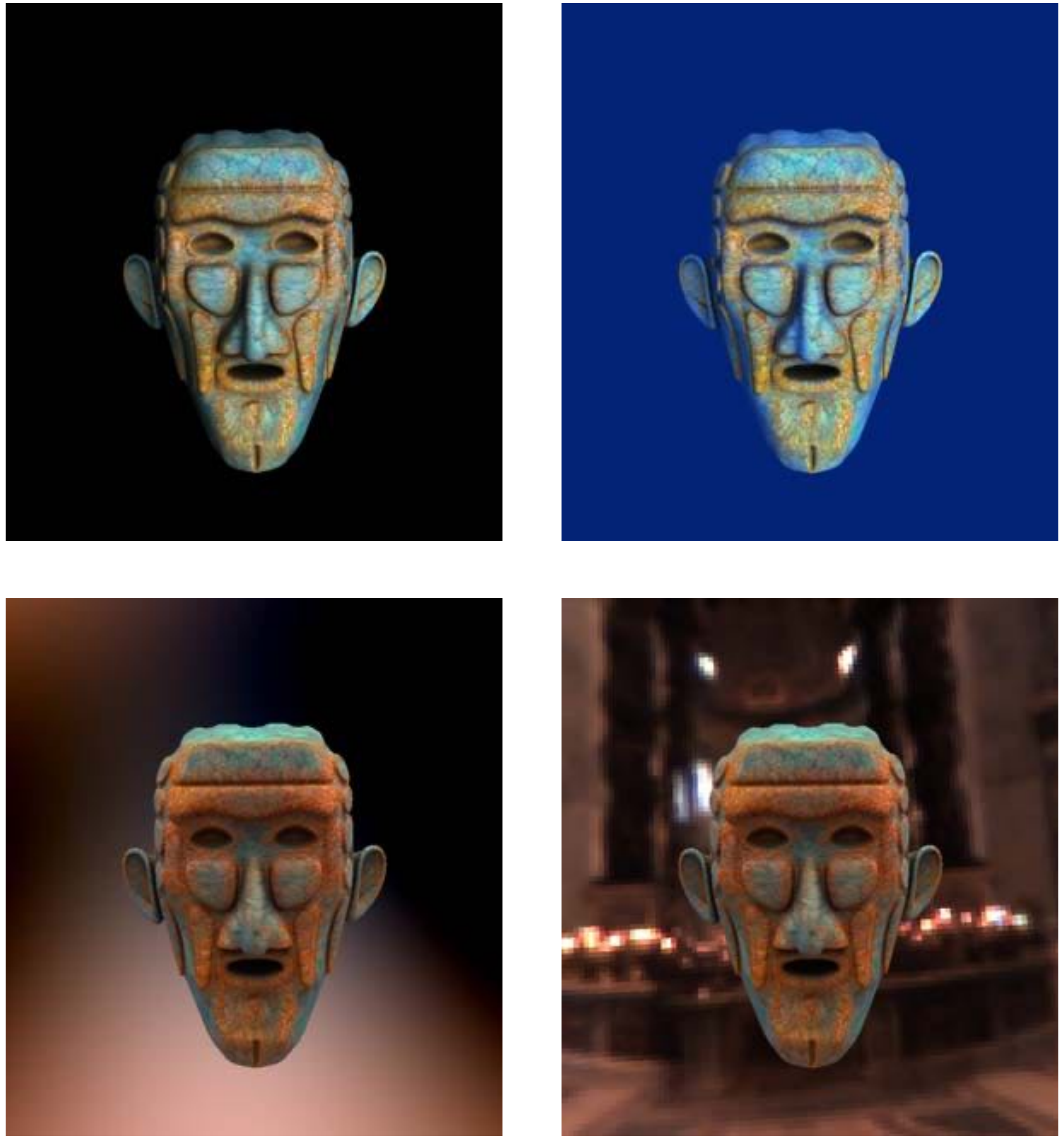
Precomputed radiance transfer

Fixed:

- Scene geometry
- Materials (let’s assume diffuse for simplicity)

Variable:

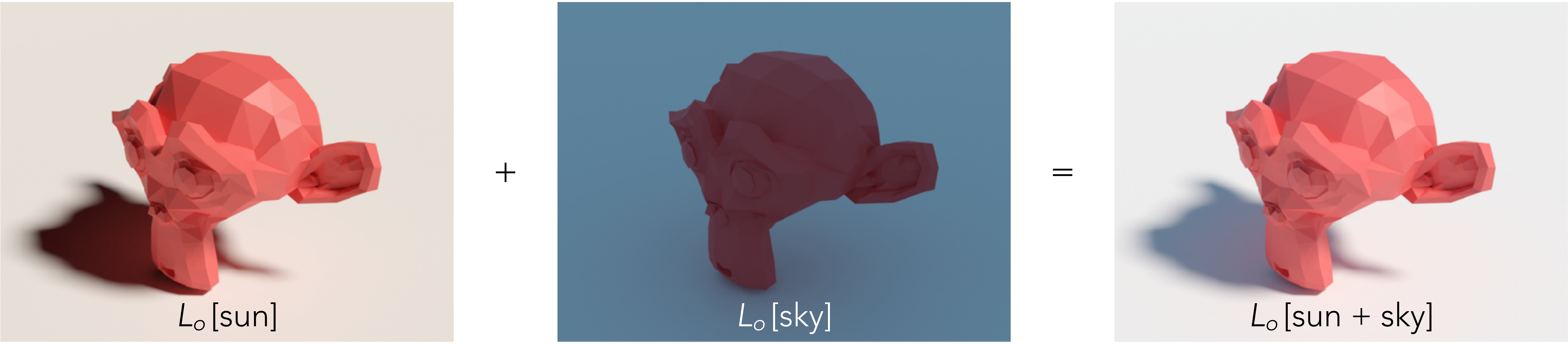
- Environment lighting



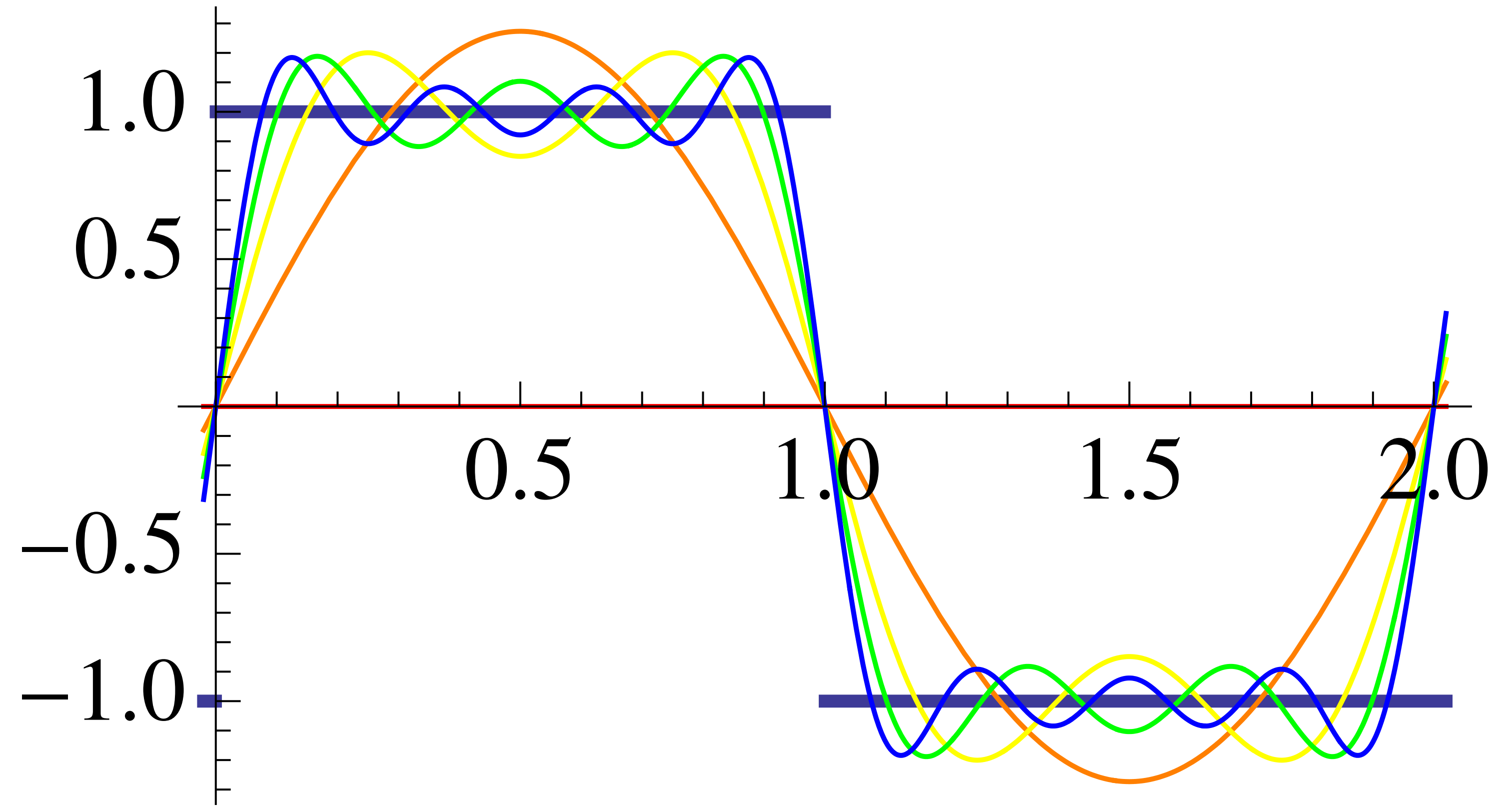
Key fact: scene appearance $L_o(\mathbf{p}, \boldsymbol{\omega})$ is linear in environment light distribution $L_{\text{env}}(\boldsymbol{\omega})$

- Choose a basis for environment lighting, $L_{\text{env}} = \ell_1 \, B_1 + \ell_2 \, B_2 + \cdots$
- For each lighting basis B_i, precompute radiance $L_o[B_i]$ at all points in scene
- At runtime, $L_o[L_{\text{env}}] = \ell_1 \, L_o[B_1] + \ell_2 \, L_o[B_2] + \cdots$

What's a good basis for environment lighting $L_{\text{env}}(\boldsymbol{\omega})$?



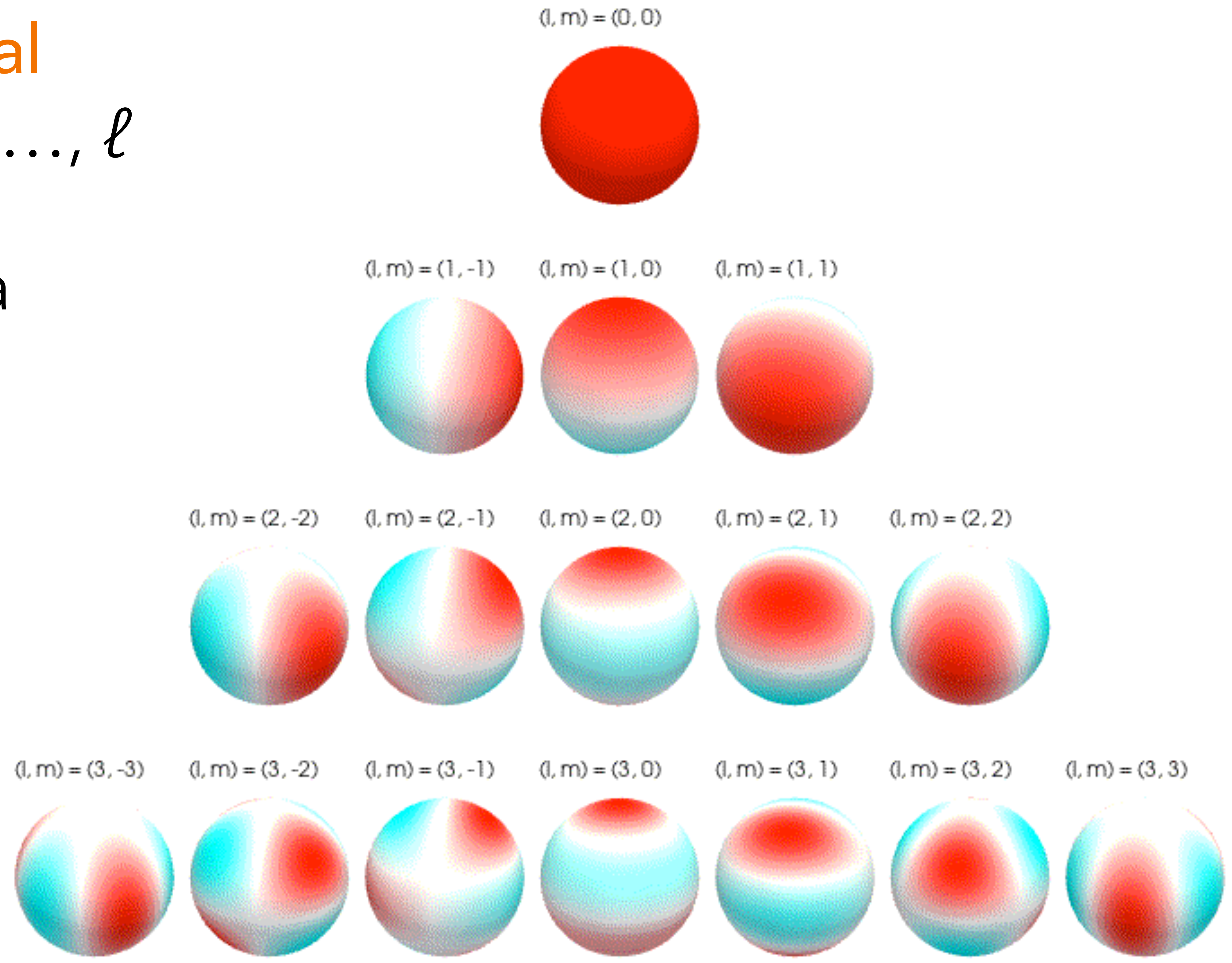
Fourier basis: Any function on a circle can be approximated by a linear combination of sinusoids $\cos(k\theta)$, $\sin(k\theta)$ for $k = 0, 1, 2, \ldots$



On a sphere, analogous basis functions are spherical harmonics $Y_{\ell m}(\boldsymbol{\omega})$ for $\ell = 0, 1, 2, \ldots,\ m = -\ell, -\ell + 1, \ldots, \ell$

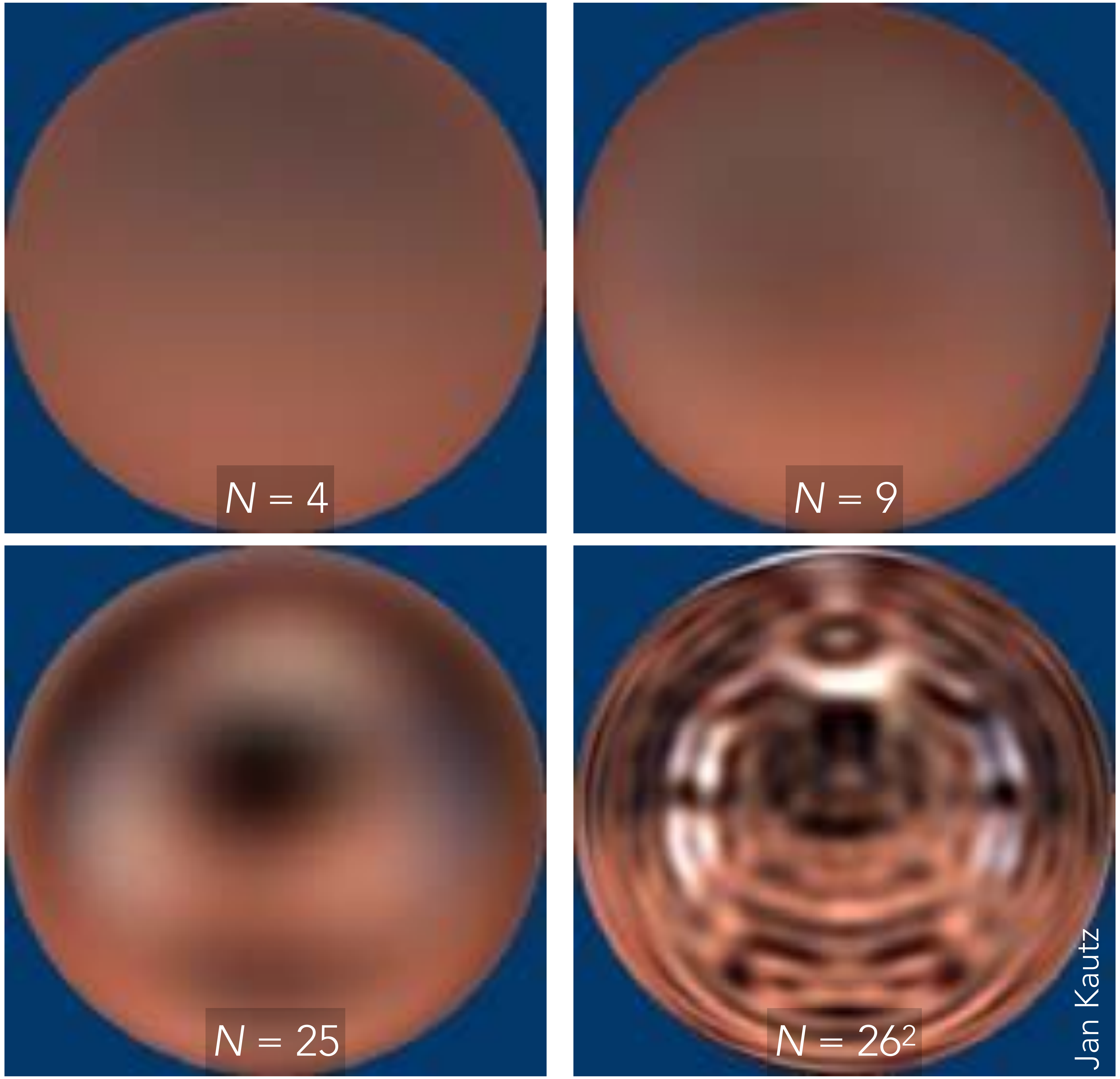
Any function on a sphere can be approximated as a linear combination $f(\boldsymbol{\omega}) \approx \sum c_{\ell m} Y_{\ell m}(\boldsymbol{\omega})$

- Only keep finite range of ℓ
  ⇒ low-frequency approximation



Approximating environment lighting with *N* spherical harmonics

$$L_{\text{env}}(\boldsymbol{\omega}) = \sum_{i=1}^{N} \ell_i B_i(\boldsymbol{\omega})$$



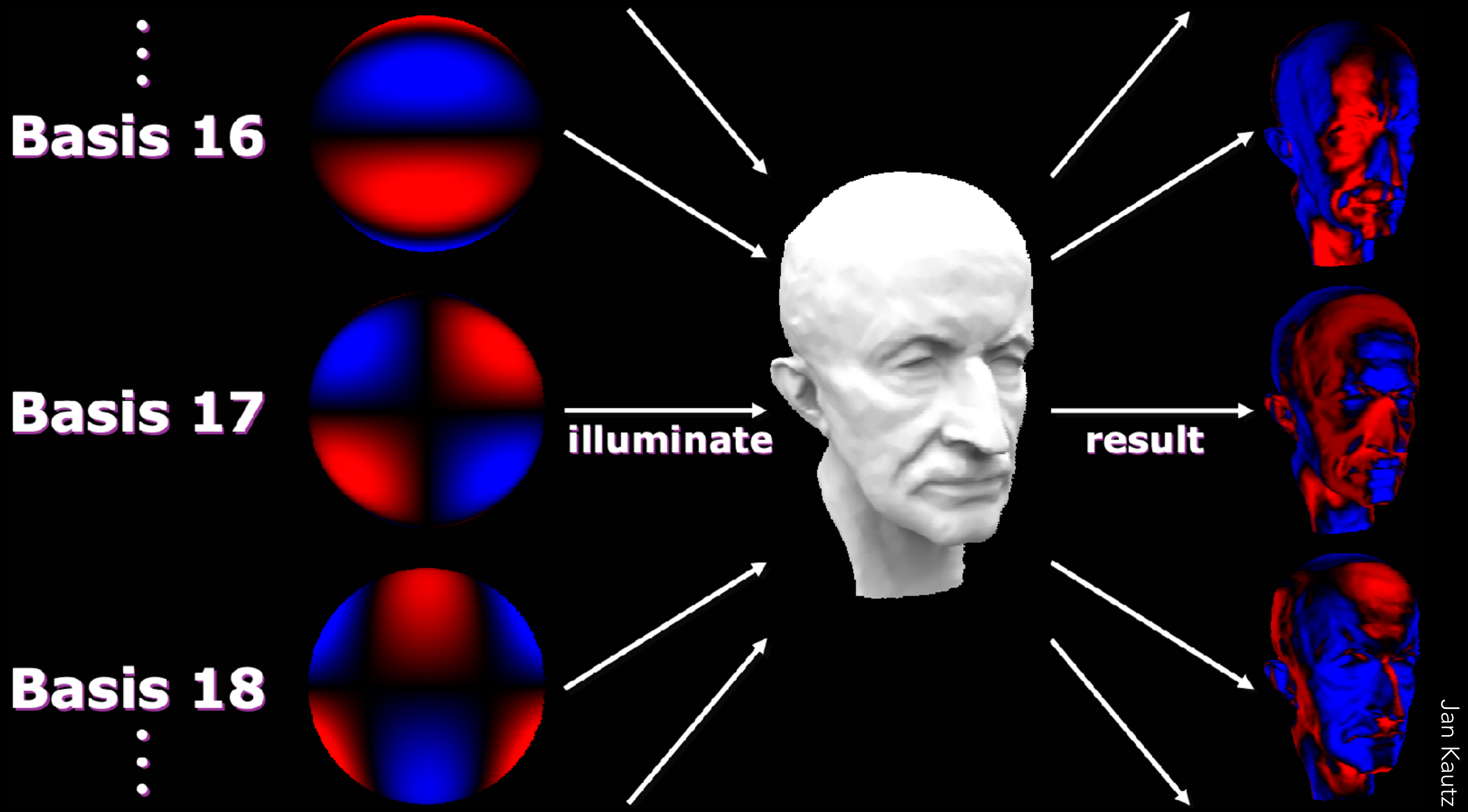
$$L_{\text{env}}(\boldsymbol{\omega}) = \sum_{i=1}^{N} \ell_i \, B_i(\boldsymbol{\omega})$$

Outgoing radiance at any point is linear in L_{env}, so…

$$L_o(\mathbf{p} \mid L_{\text{env}}) = \sum_{i=1}^{N} \ell_i \, L_o(\mathbf{p} \mid B_i)$$

- Precompute radiance in scene for each lighting basis B_i

Can run an arbitrarily complicated global illumination computation here!


Basis 16
Basis 17
Basis 18
illuminate
result
Jan Kautz

$$L_{\text{env}}(\boldsymbol{\omega}) = \sum_{i=1}^{N} \ell_i \, B_i(\boldsymbol{\omega})$$

Outgoing radiance at any point is linear in L_{env}, so…

$$L_o(\mathbf{p} \mid L_{\text{env}}) = \sum_{i=1}^{N} \ell_i \, L_o(\mathbf{p} \mid B_i)$$

Can run an arbitrarily complicated global illumination computation here!

- Precompute radiance in scene for each lighting basis B_i
- Store as transport vector $\mathbf{t}(\mathbf{p}) = (L_o(\mathbf{p} \mid B_1), L_o(\mathbf{p} \mid B_2), \ldots)$ at each point $\mathbf{p}$
- At run time, just a dot product: $L_o(\mathbf{p} \mid L_{\text{env}}) = \sum \ell_i \, t_i(\mathbf{p})$


Unshadowed
Shadowed
Interreflected

https://www.youtube.com/watch?v=2tLMcKklLS4

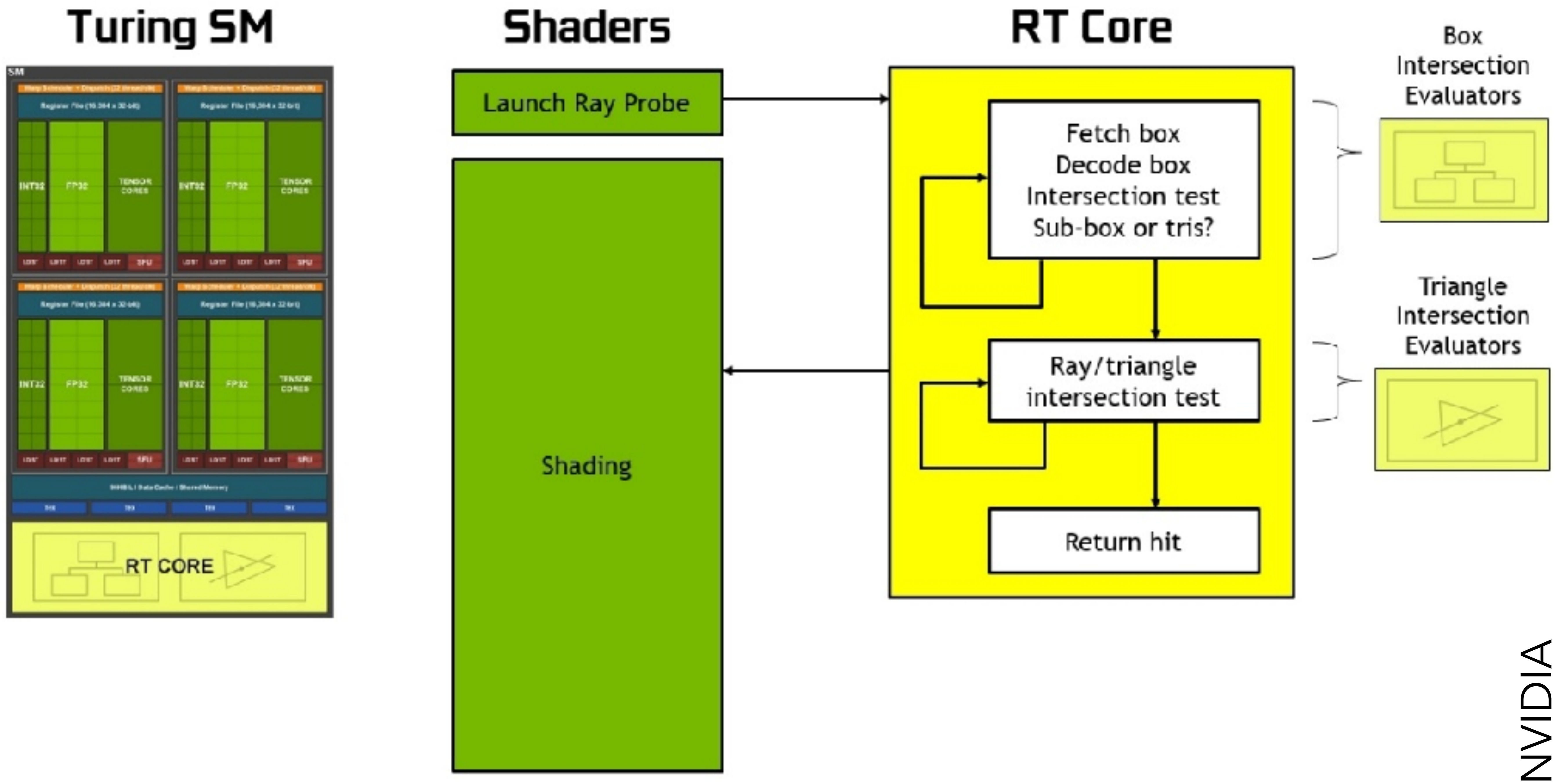
Sloan and Kautz 2002

Real-time ray tracing

Hardware support in recent graphics cards. But how does it work?

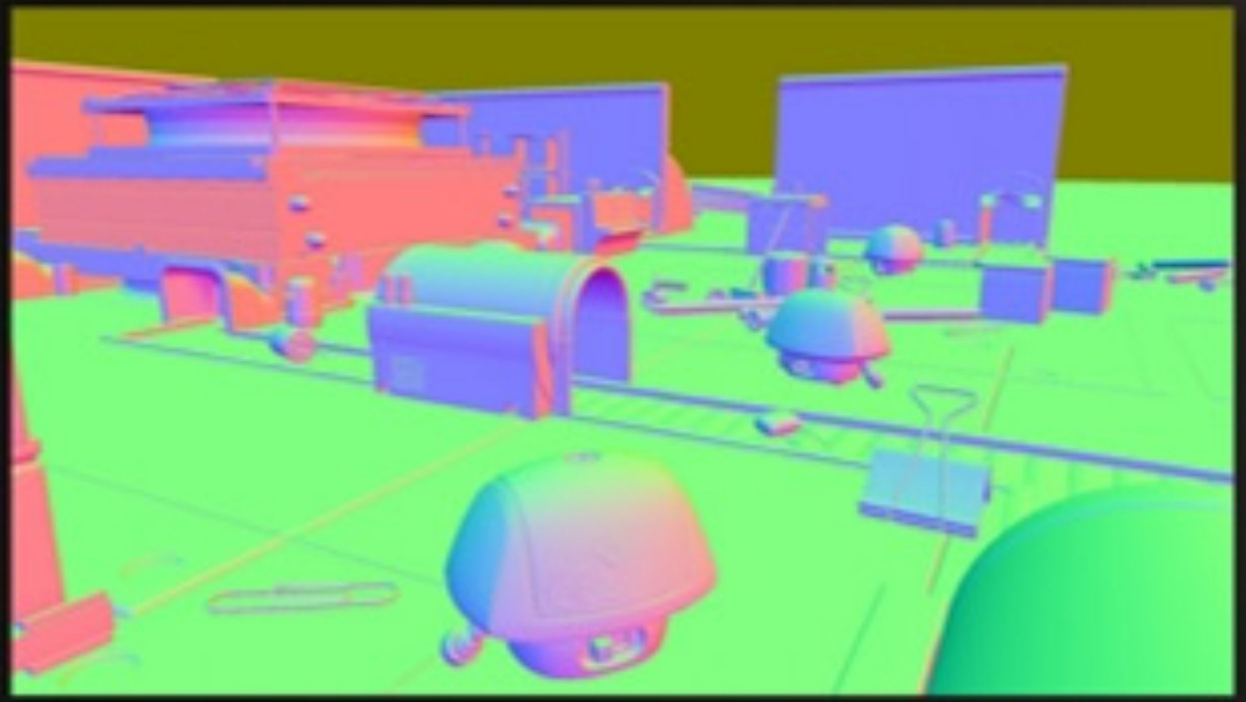
- Dedicated hardware cores for BVH traversal and ray-triangle intersection

- Ray tracing can be launched by shaders in rasterization pipeline: hybrid rendering

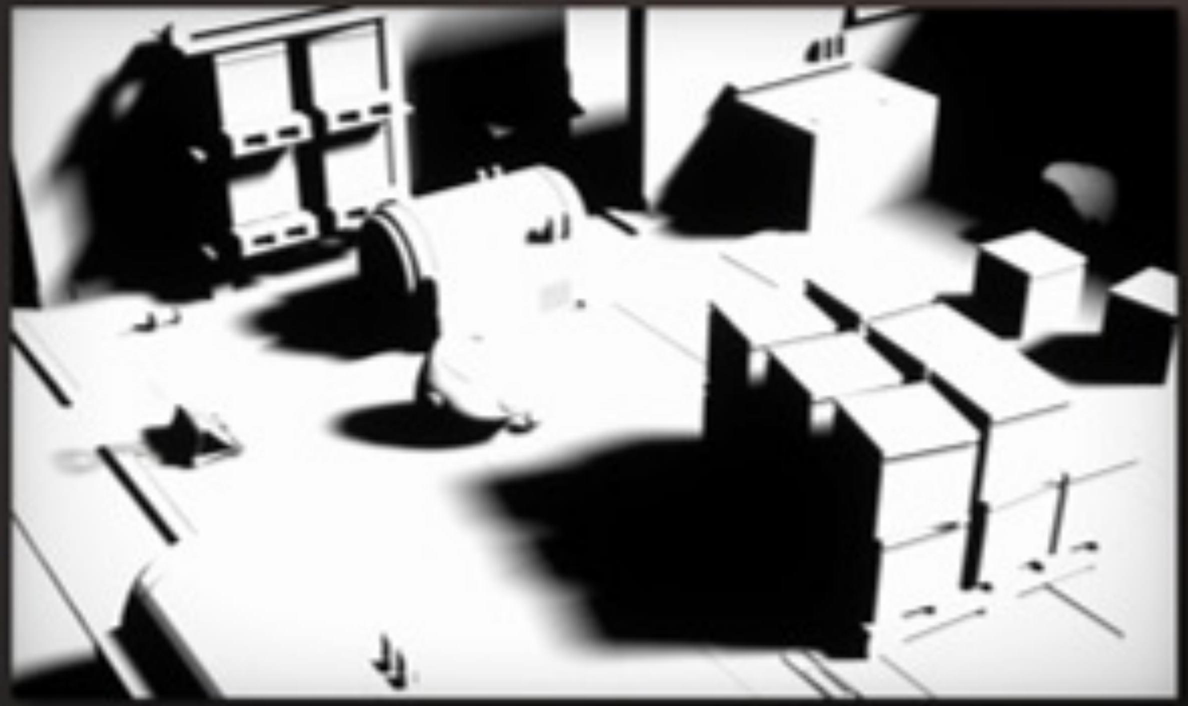


NVIDIA

Hybrid Rendering Pipeline

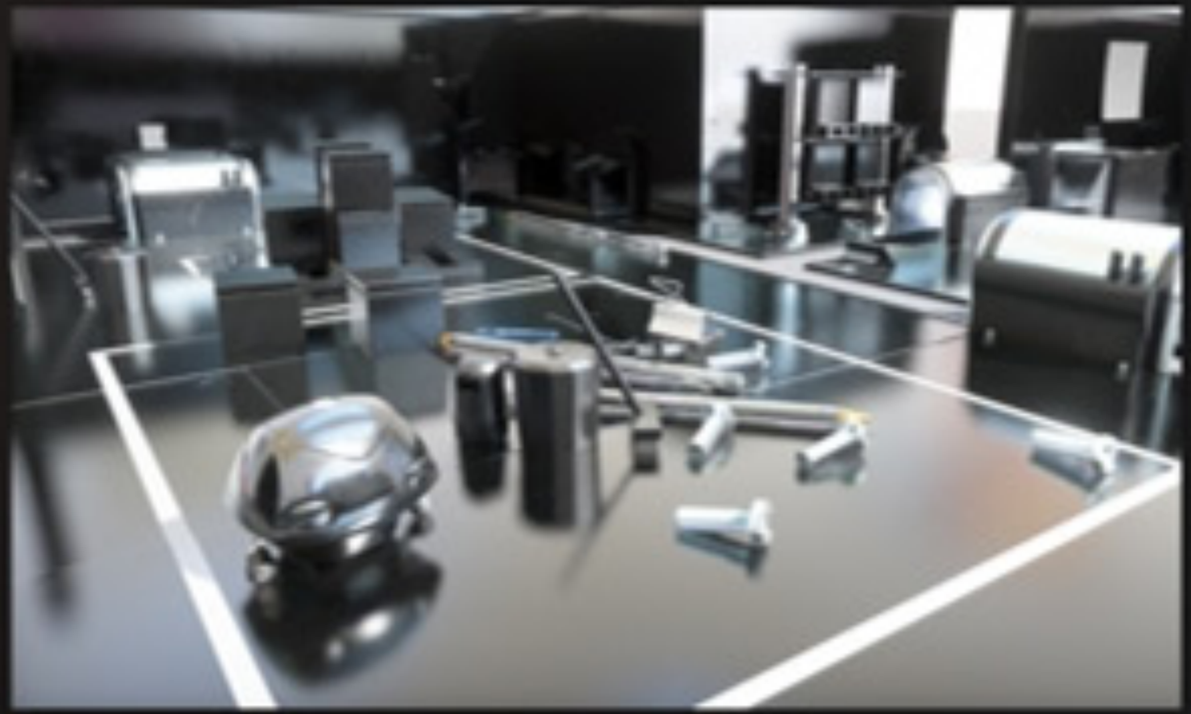


Deferred shading
(raster)



Direct shadows
(raytrace or raster)

Lighting
(compute + raytrace)



Reflections
(raytrace or compute)

Global Illumination
(compute and raytrace)

Ambient occlusion
(raytrace or compute)

Transparency & Translucency
(raytrace and compute)

Post processing
(compute)

For global illumination, usually:

- Path tracing with 1 sample per pixel
- Only 1 secondary ray (1-bounce indirect illumination)
- Lots of clever denoising!
    - Spatial (using nearby samples)
    - Temporal (using samples from previous frames)

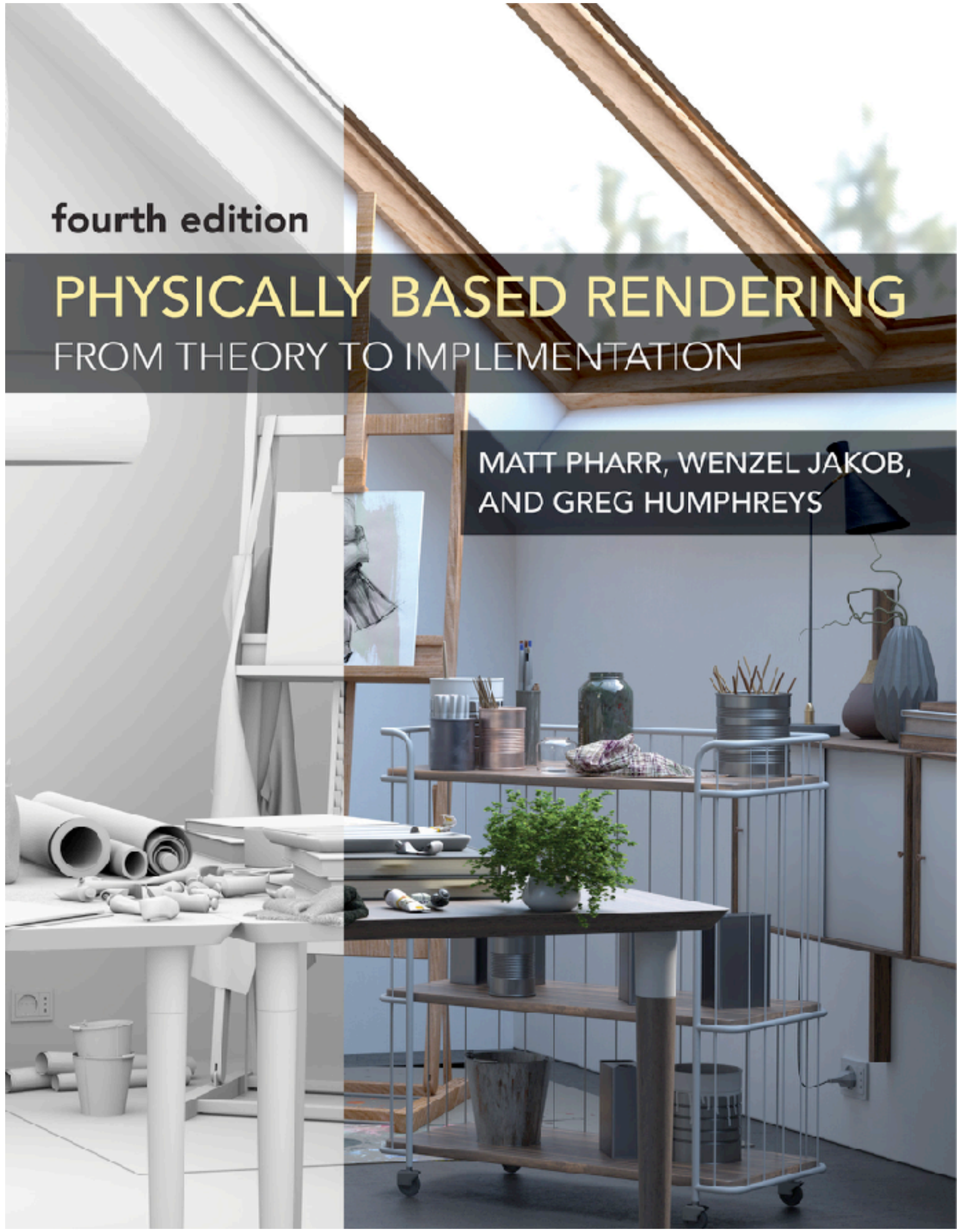
1spp input

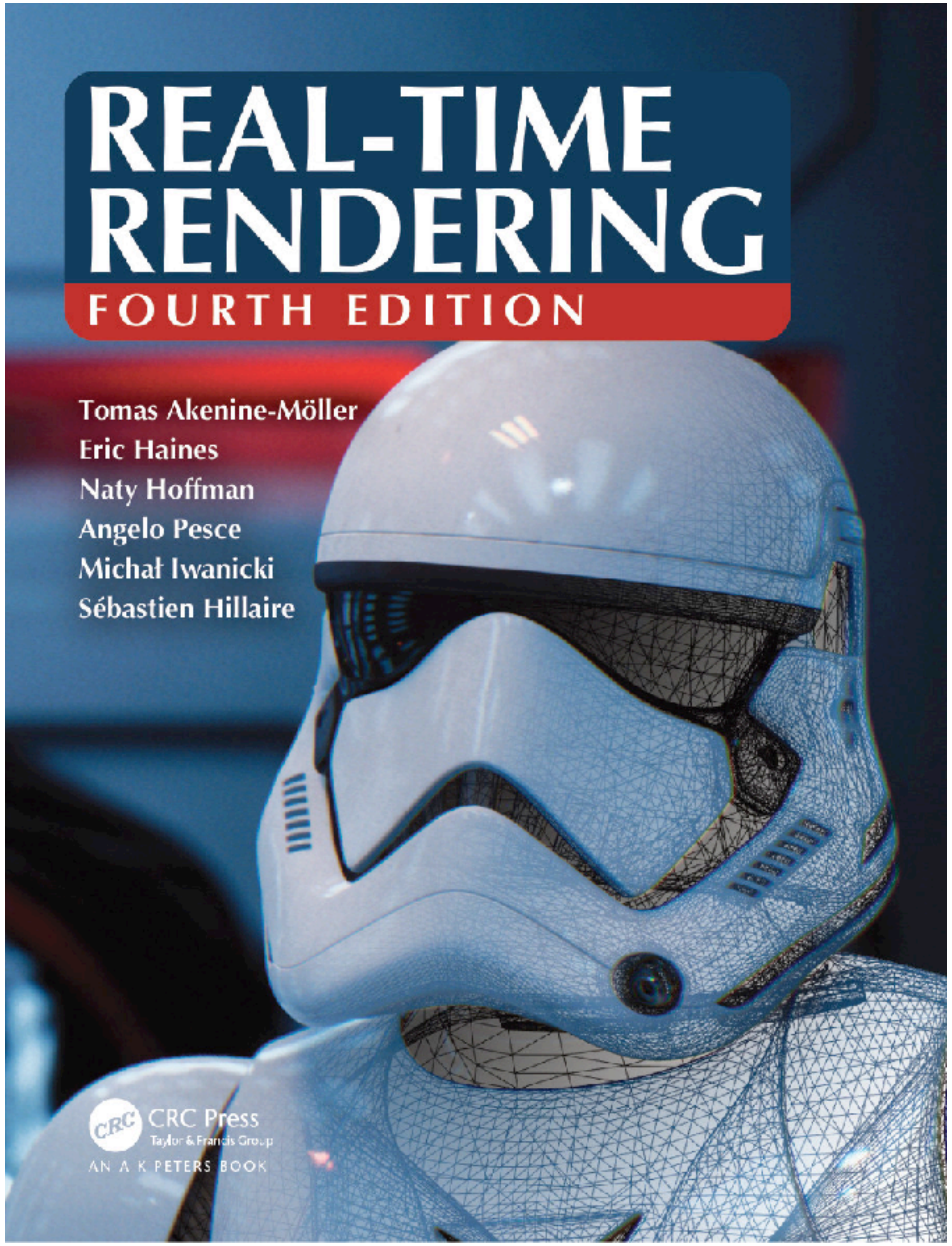
Denoised

Chaitanya et al. 2017

via Lingqi Yan

via Lingqi Yan


fourth edition
PHYSICALLY BASED RENDERING
FROM THEORY TO IMPLEMENTATION
MATT PHARR, WENZEL JAKOB,
AND GREG HUMPHREYS


REAL-TIME
RENDERING
FOURTH EDITION
Tomas Akenine-Möller
Eric Haines
Naty Hoffman
Angelo Pesce
Michał Iwanicki
Sébastien Hillaire
CRC Press
AN A K PETERS BOOK

Major exam format

Saturday, 4 May
10:30AM–12:30PM
LH 318

Syllabus is the entire course content, but more emphasis on latter part:

- 30% pre-minor topics
- 70% post-minor topics

You're allowed to bring a double-sided A4 size page of handwritten notes

Remaining evaluations

Assignment 4: Demos to be scheduled (during majors)

Participation: Based on attendance and Moodle Q&A

Course goals (from lecture 1)

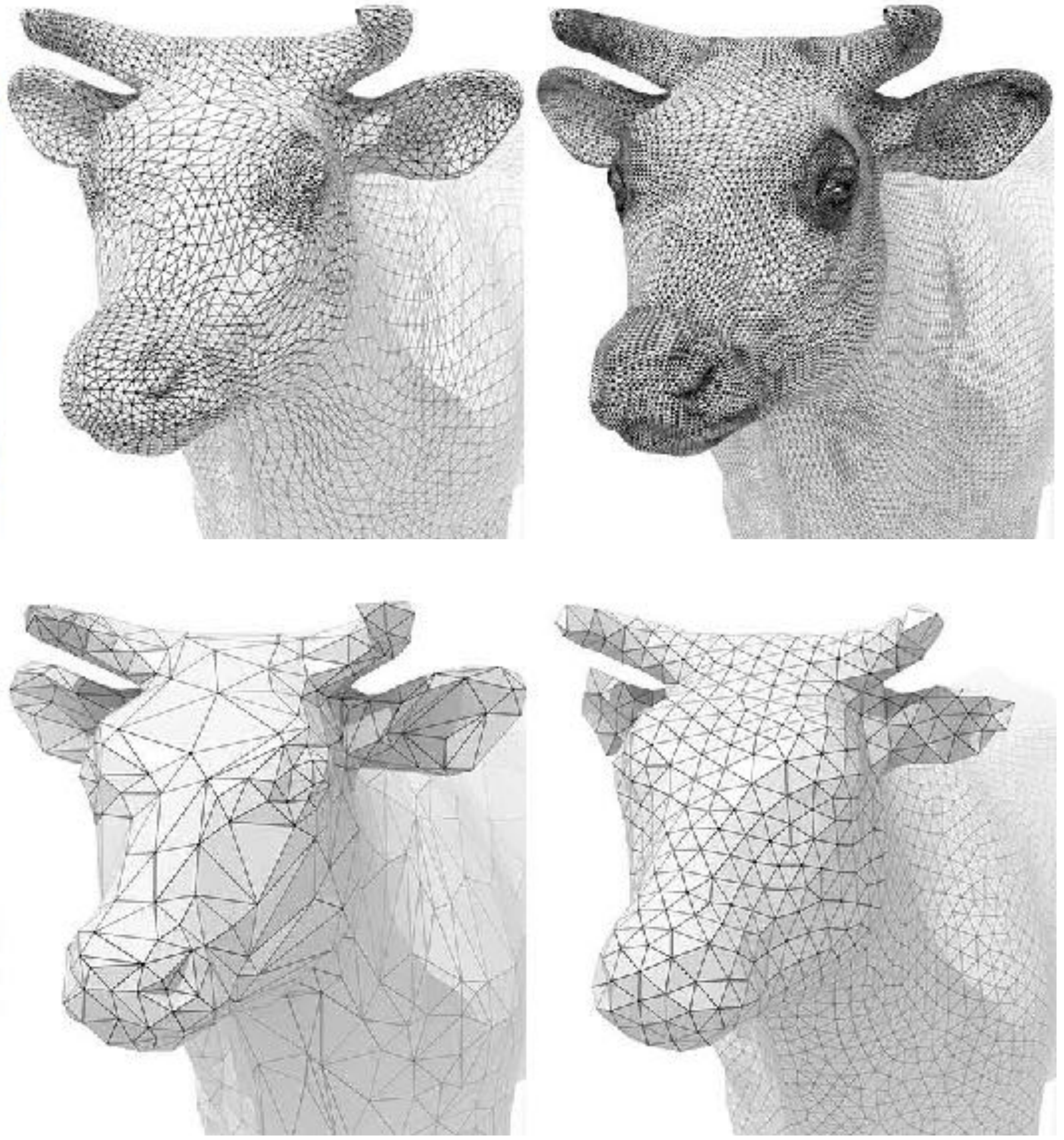
Scientific and mathematical foundations of graphics

- Physics of light and colour, materials, dynamics for animation, …
- Mathematics of curves and surfaces, perspective projection, sampling, …

Representations, algorithms, and systems

- Modelling geometry, images, transformations, …
- Mesh subdivision, ray tracing, time integration, …
- GPUs, hardware rendering pipeline, …

Course content

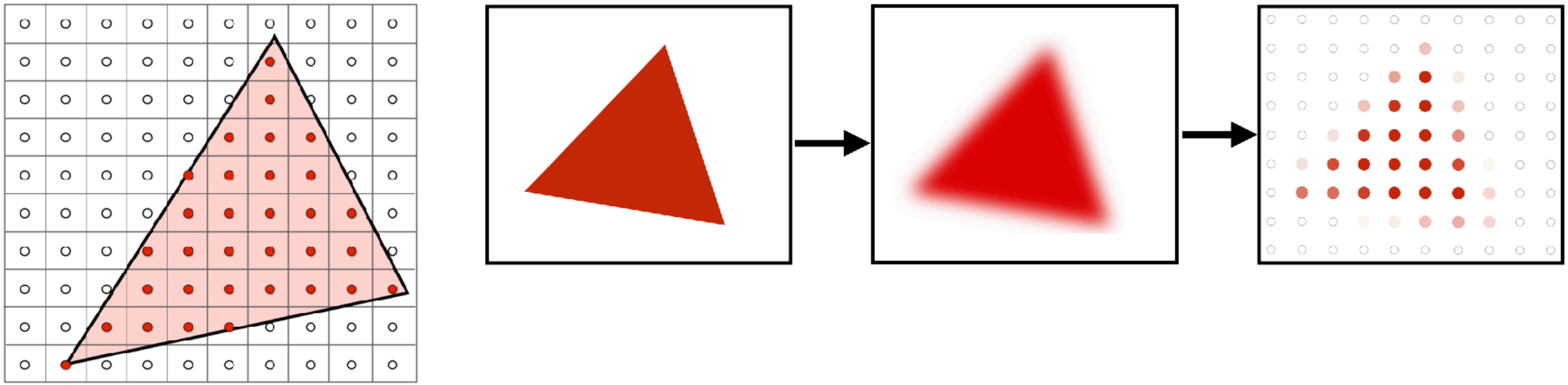


Modelling

Rendering

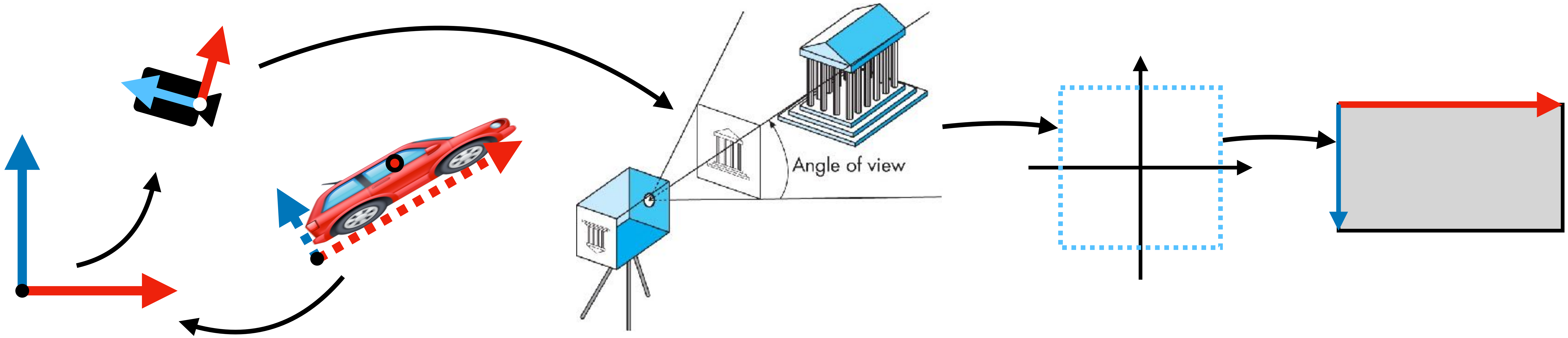
Animation

Rasterization, sampling, ...



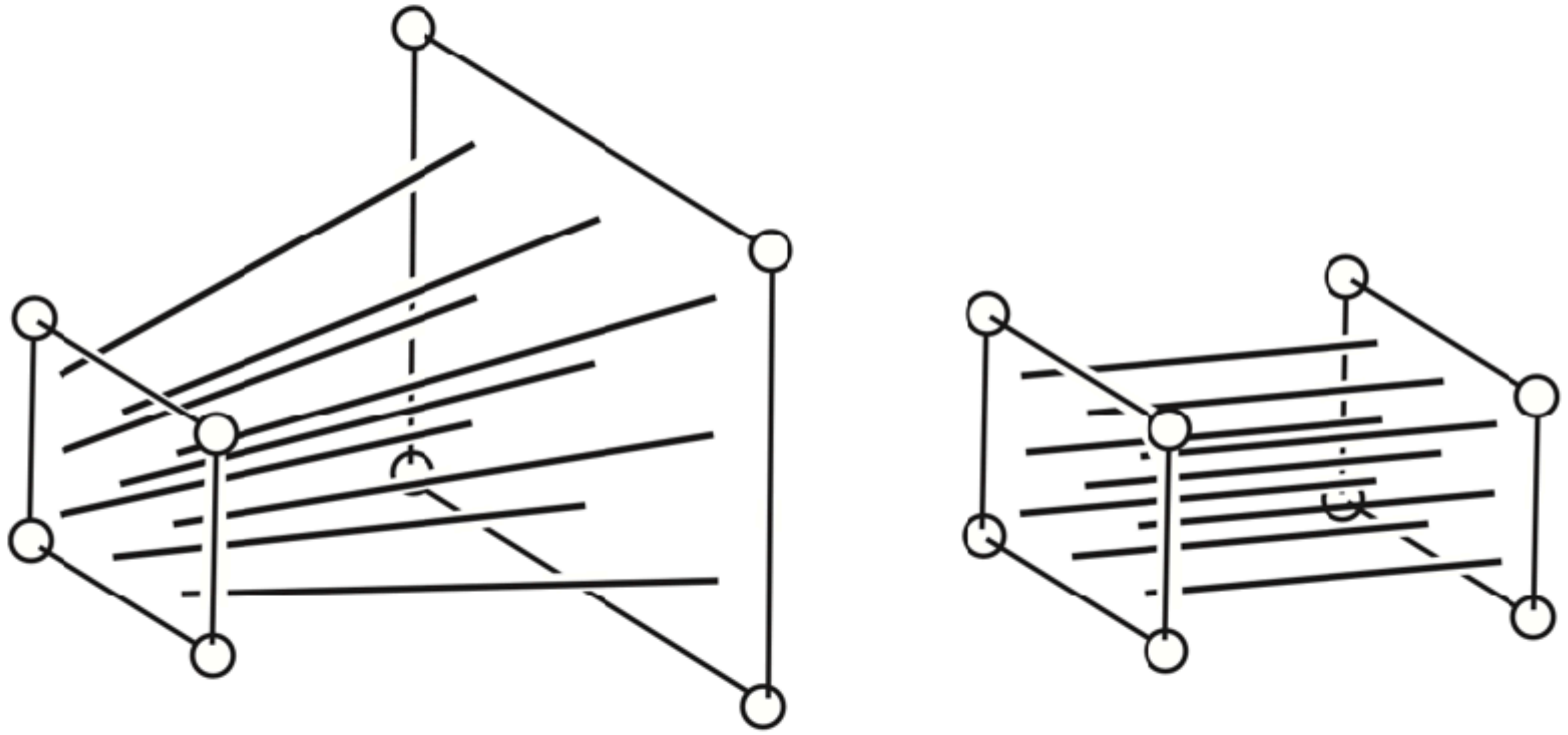
Vector vs. raster, point-in-triangle test, bounding boxes, supersampling, filtering, ...

Transformations: linear, affine, …



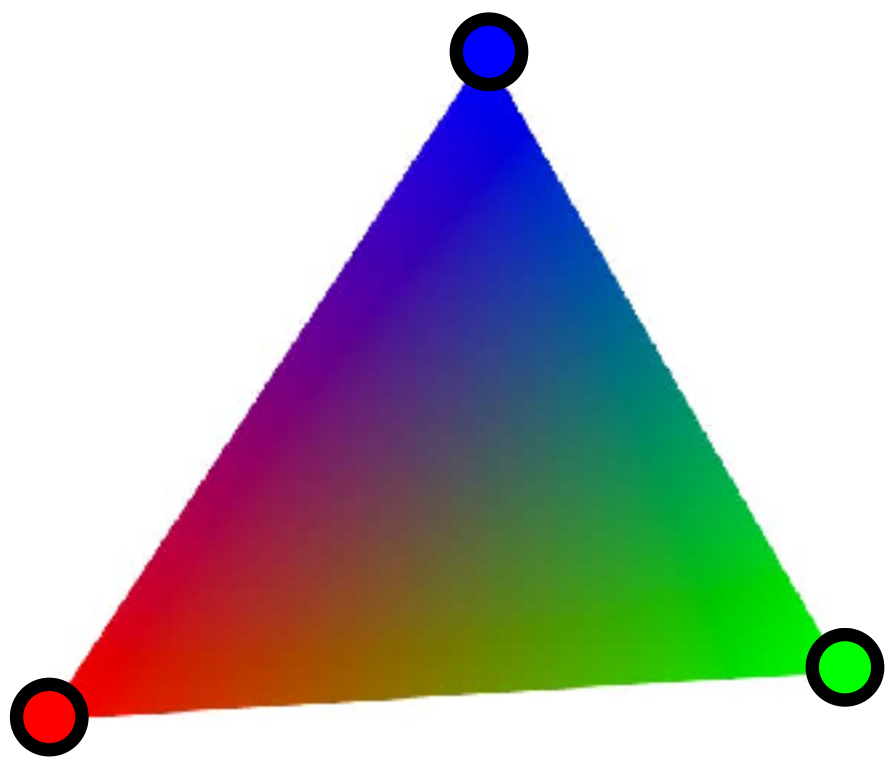
Matrices as linear transformations, coordinate systems, homogeneous coordinates, hierarchical transformations, transformation pipeline, …

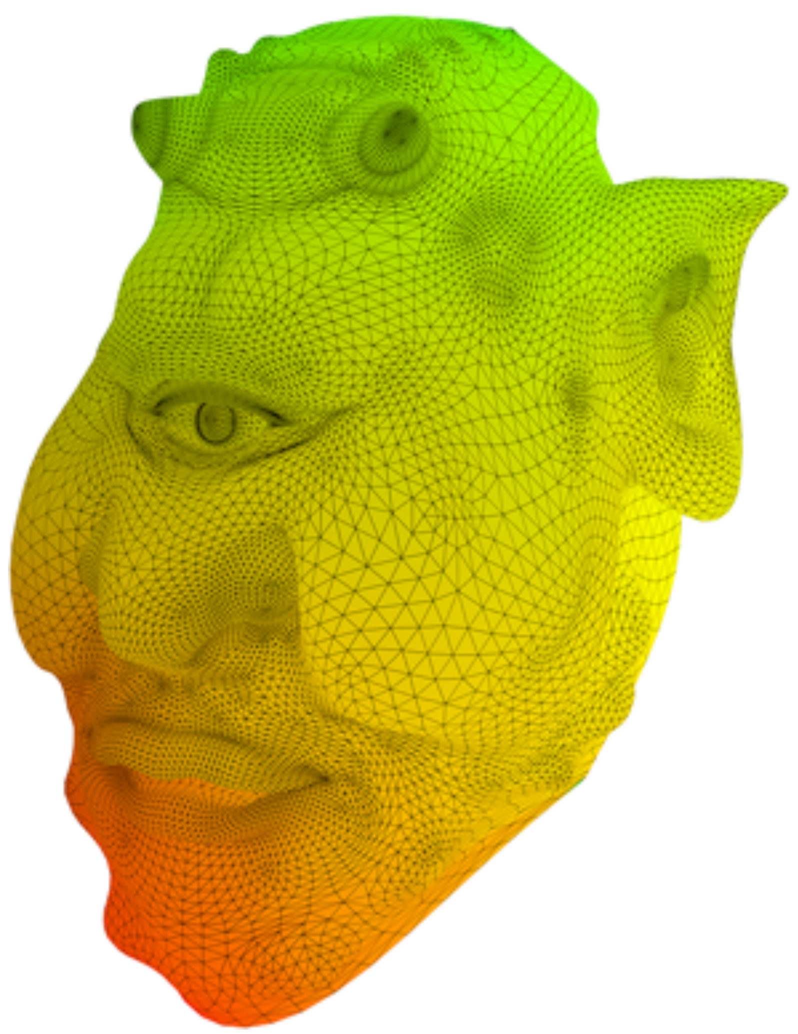
Perspective, visibility, …

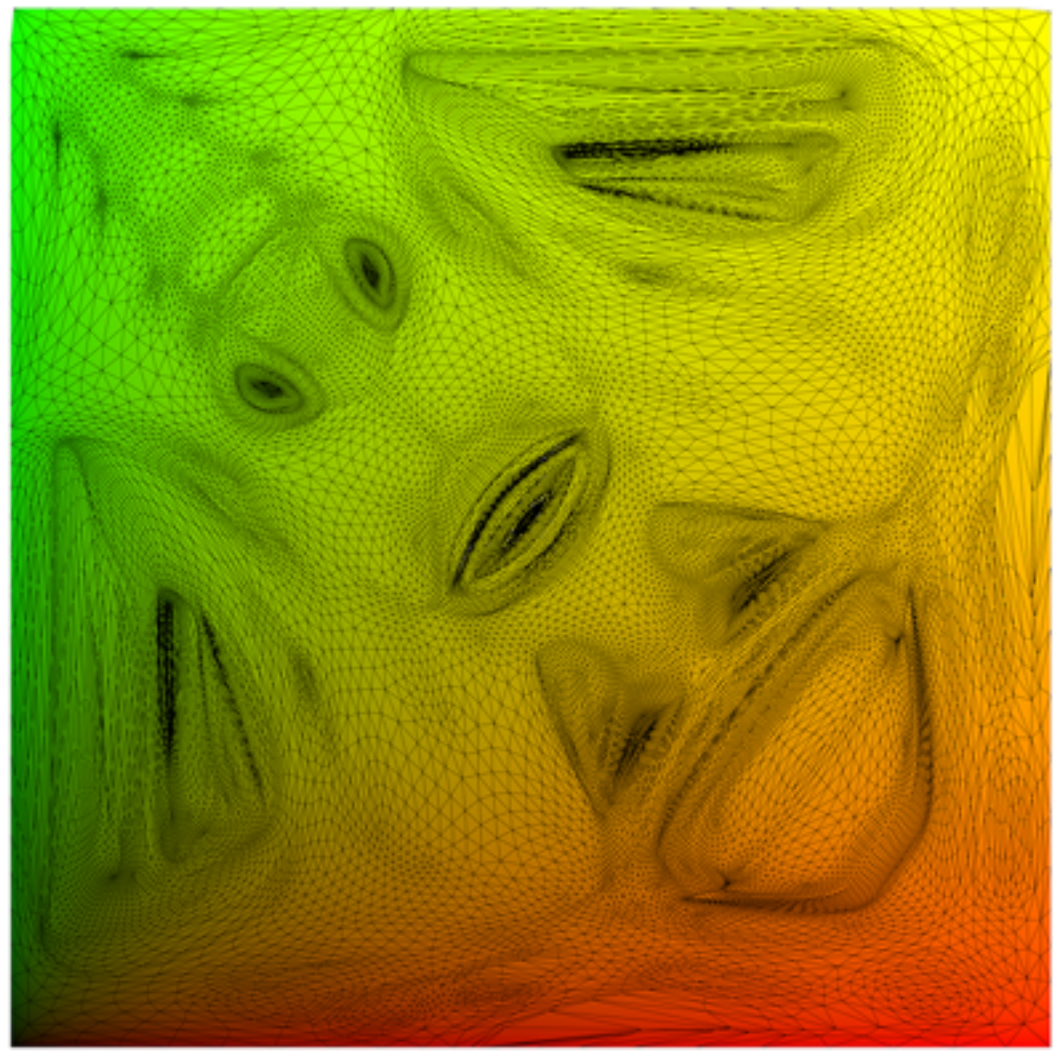


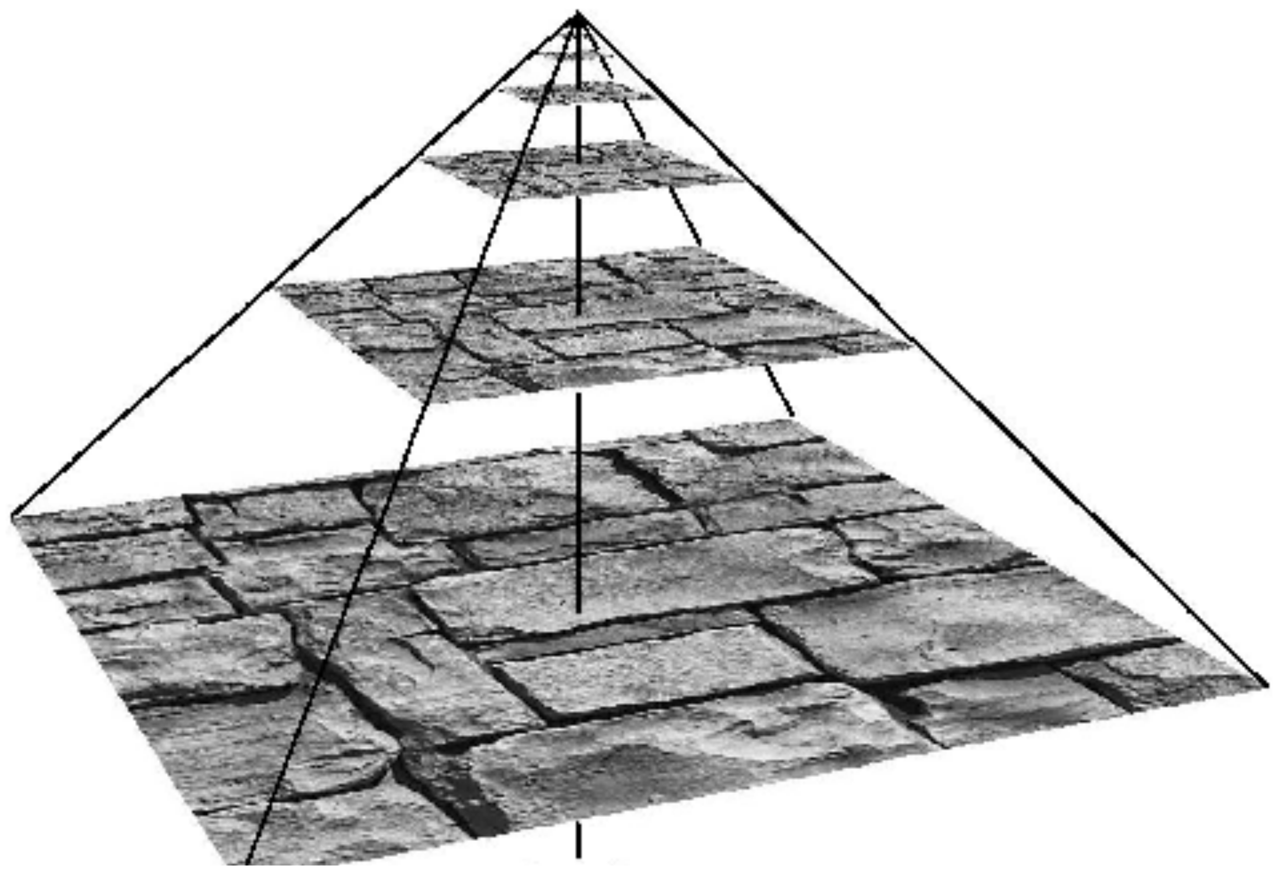
Perspective via homogeneous coordinates, the visibility problem, z-buffering, …

Interpolation, texture mapping, …



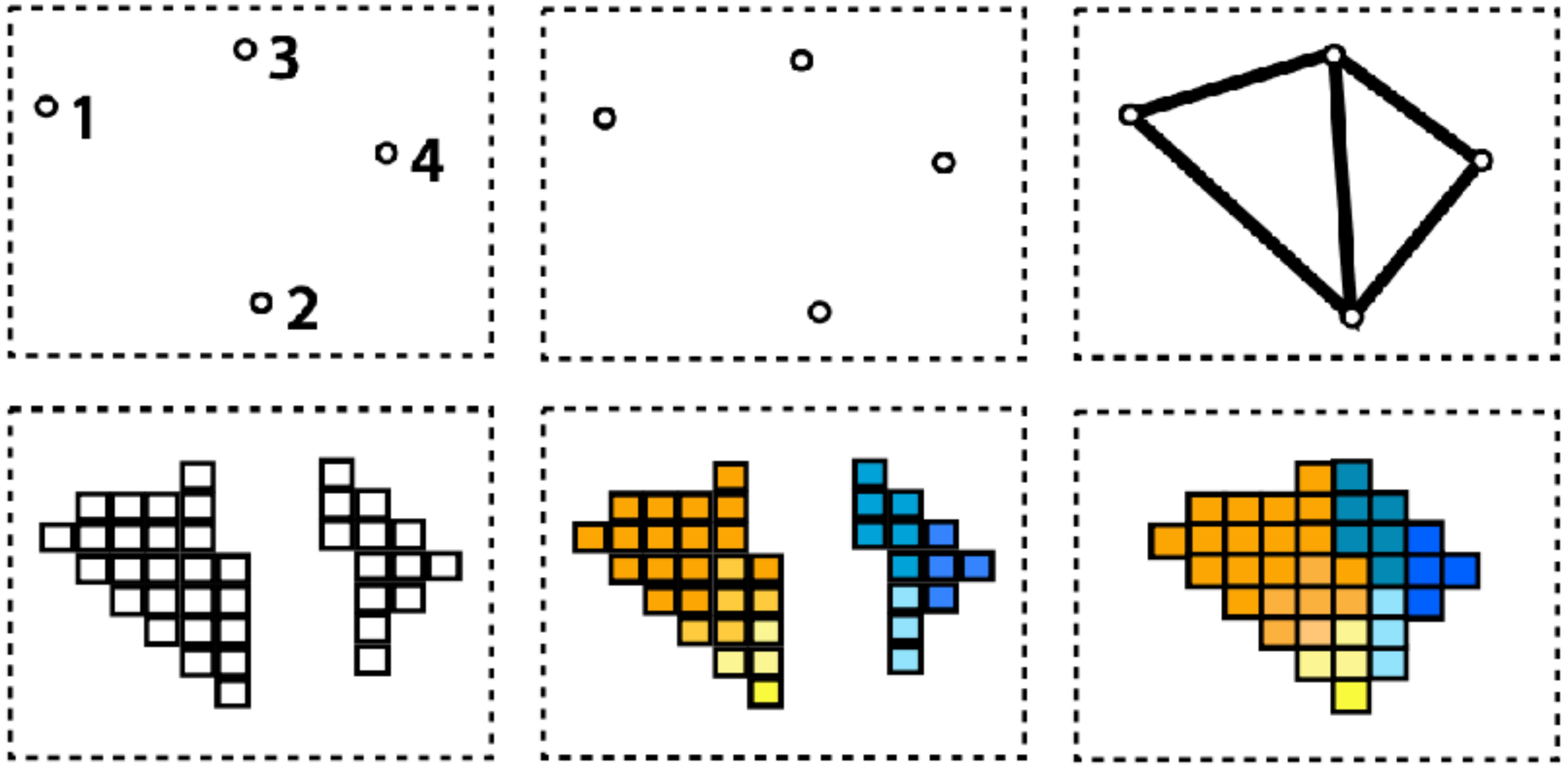


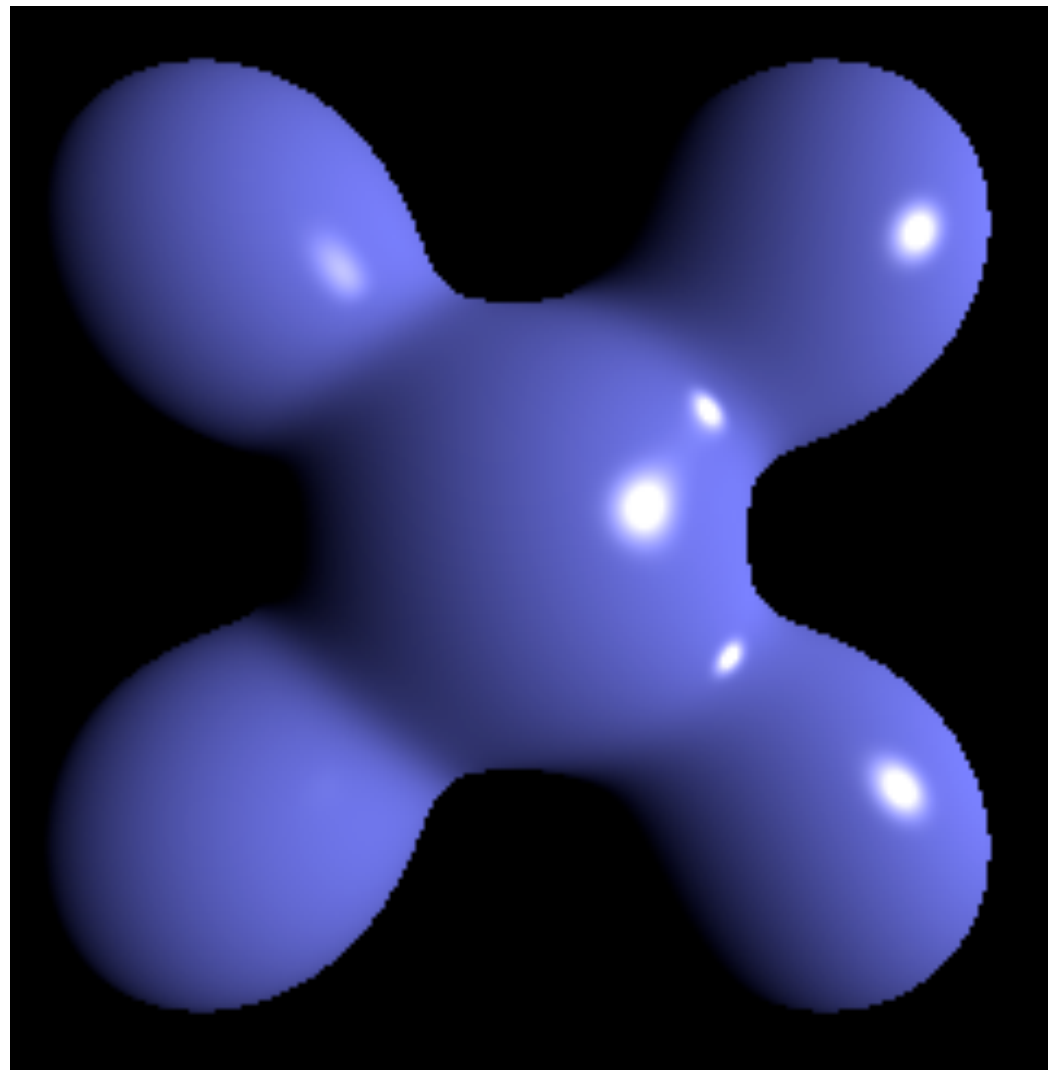




Barycentric coordinates, basis functions, parameterization via texture coordinates, bi- and trilinear interpolation, mipmaps for prefiltering, …

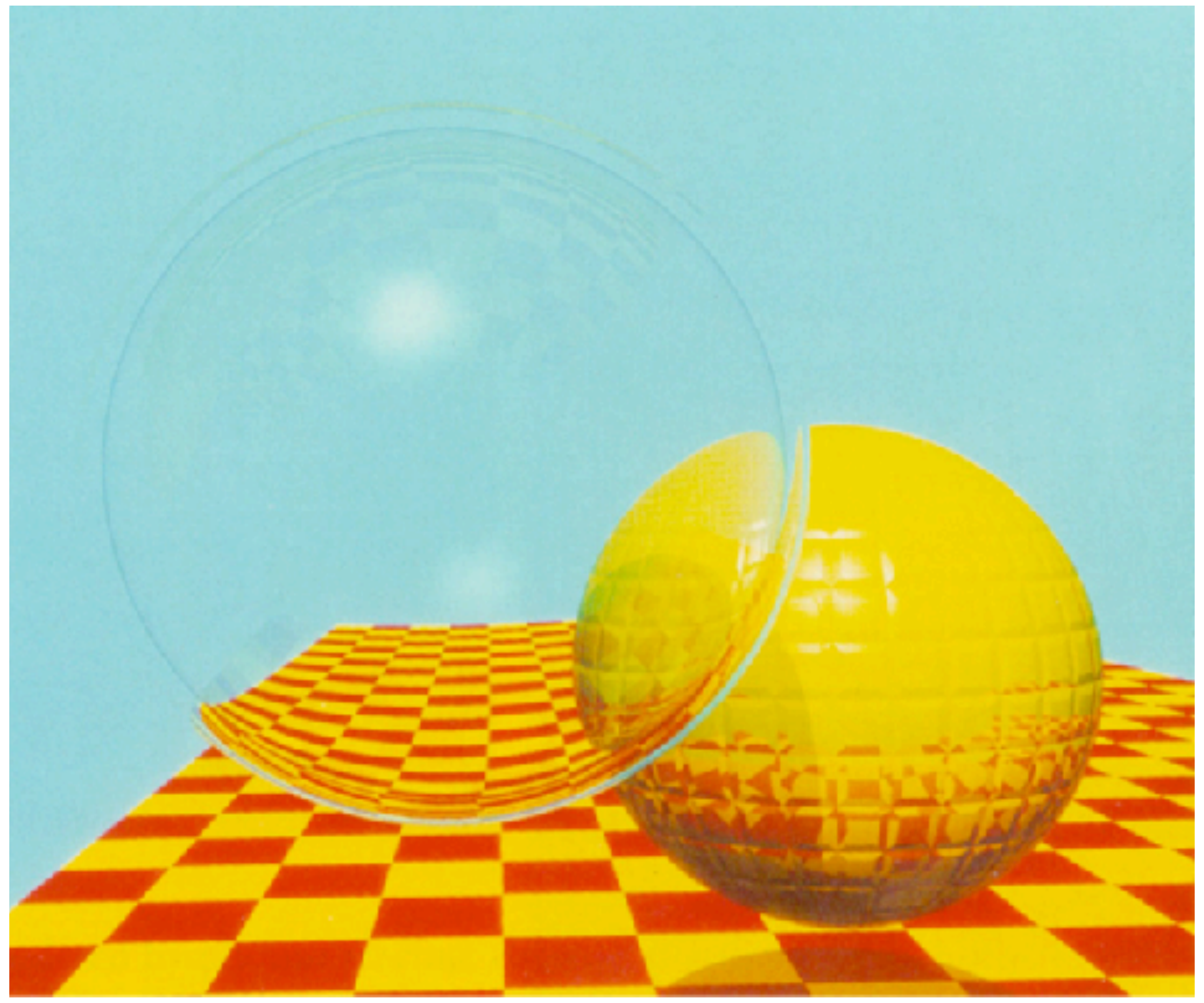
Rasterization pipeline, transparency, shading, …

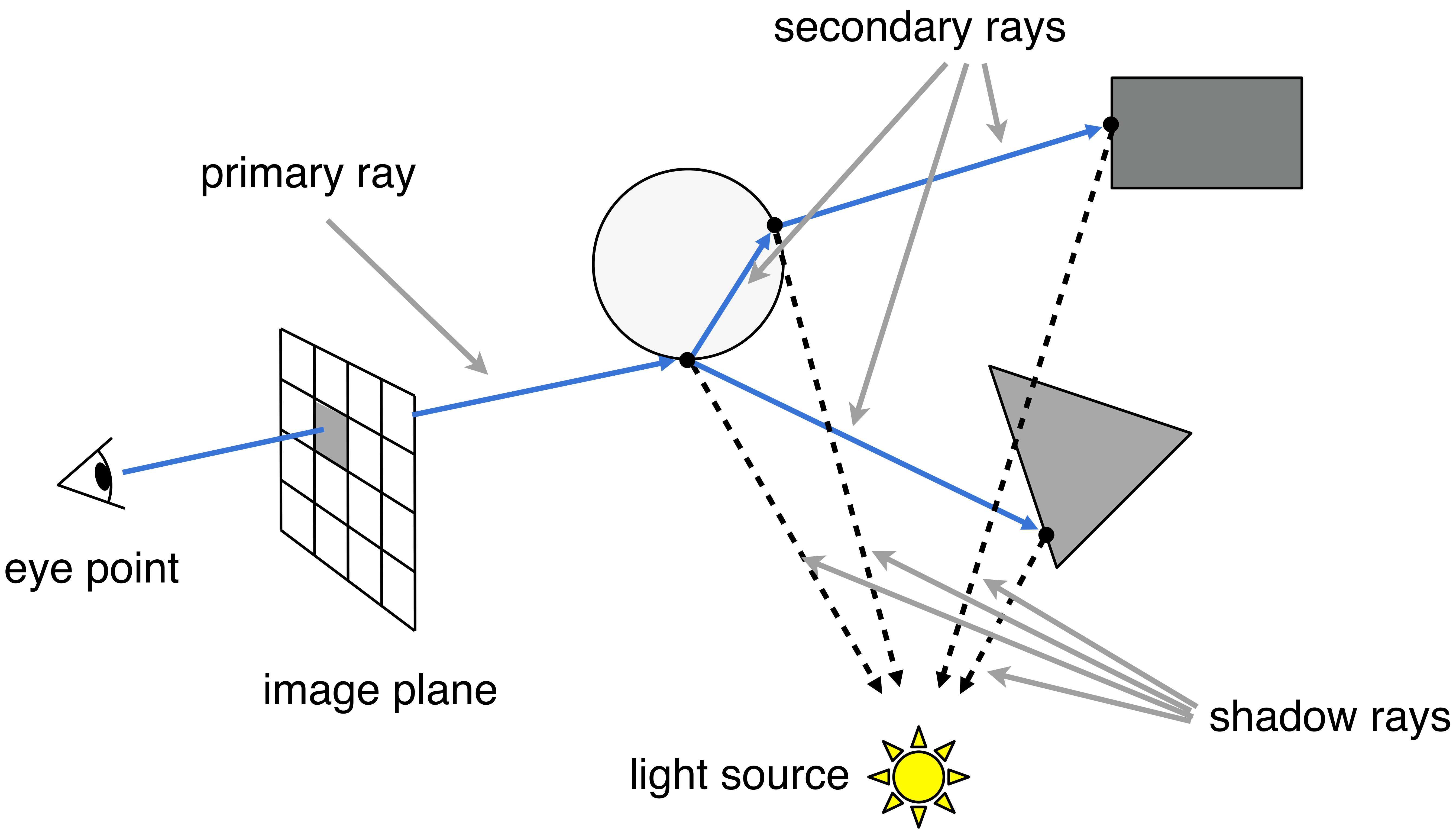




Programmable vertex and fragment processing, alpha compositing, Blinn-Phong reflectance model, …

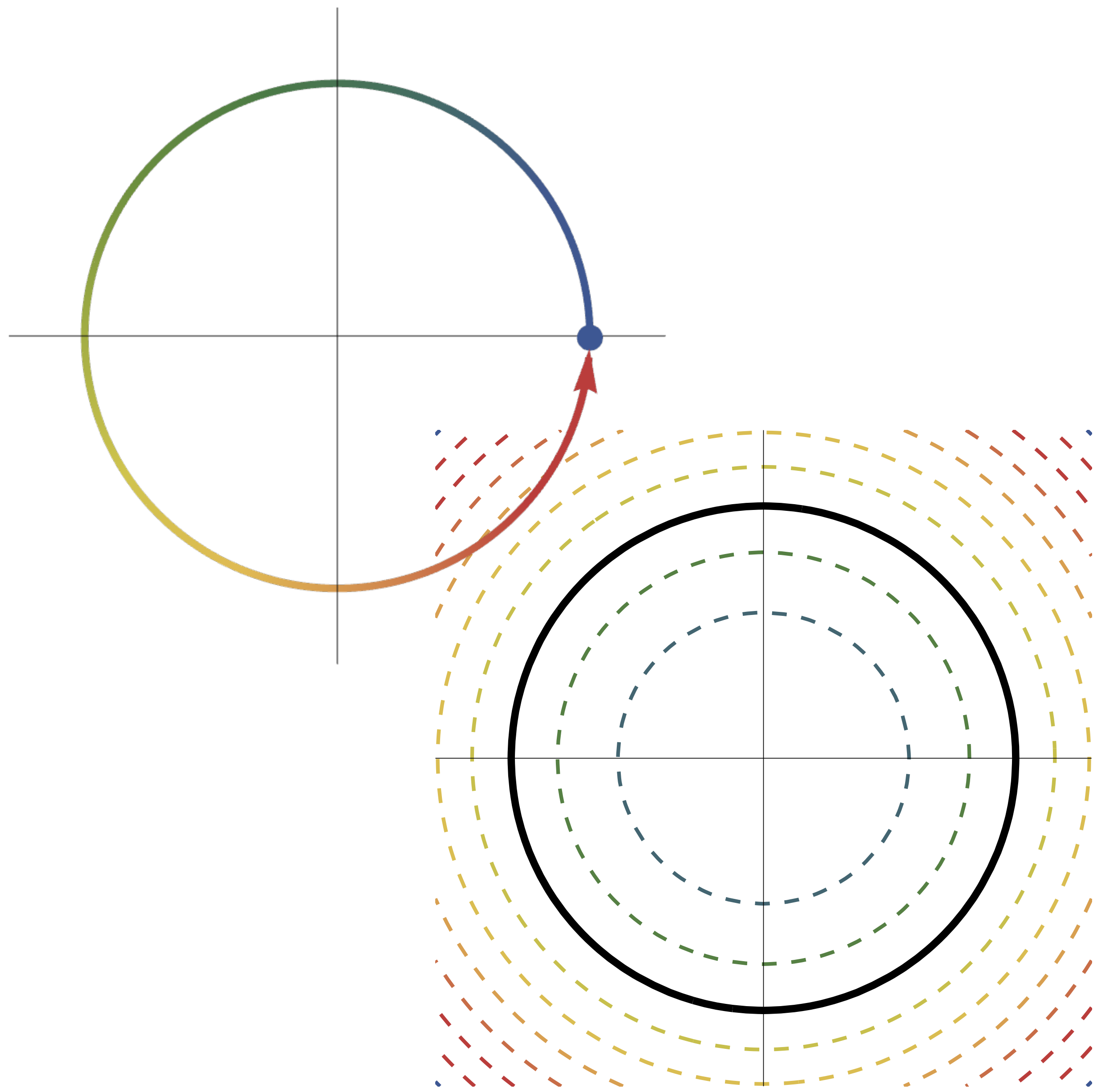
Ray tracing, …

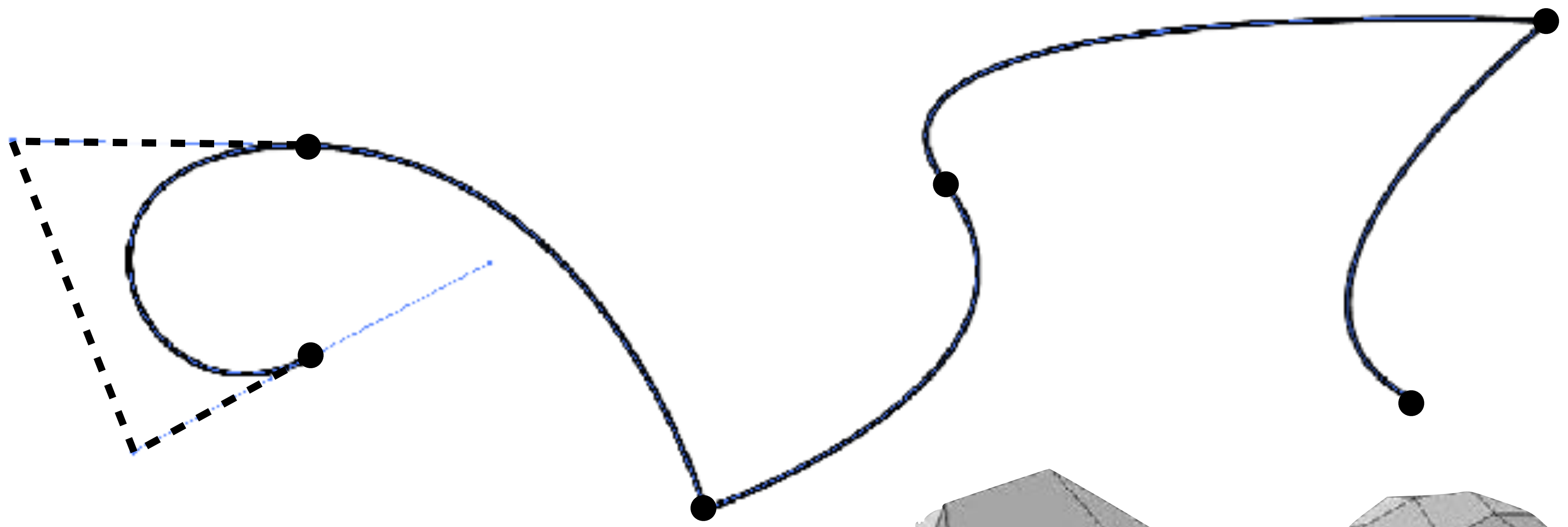


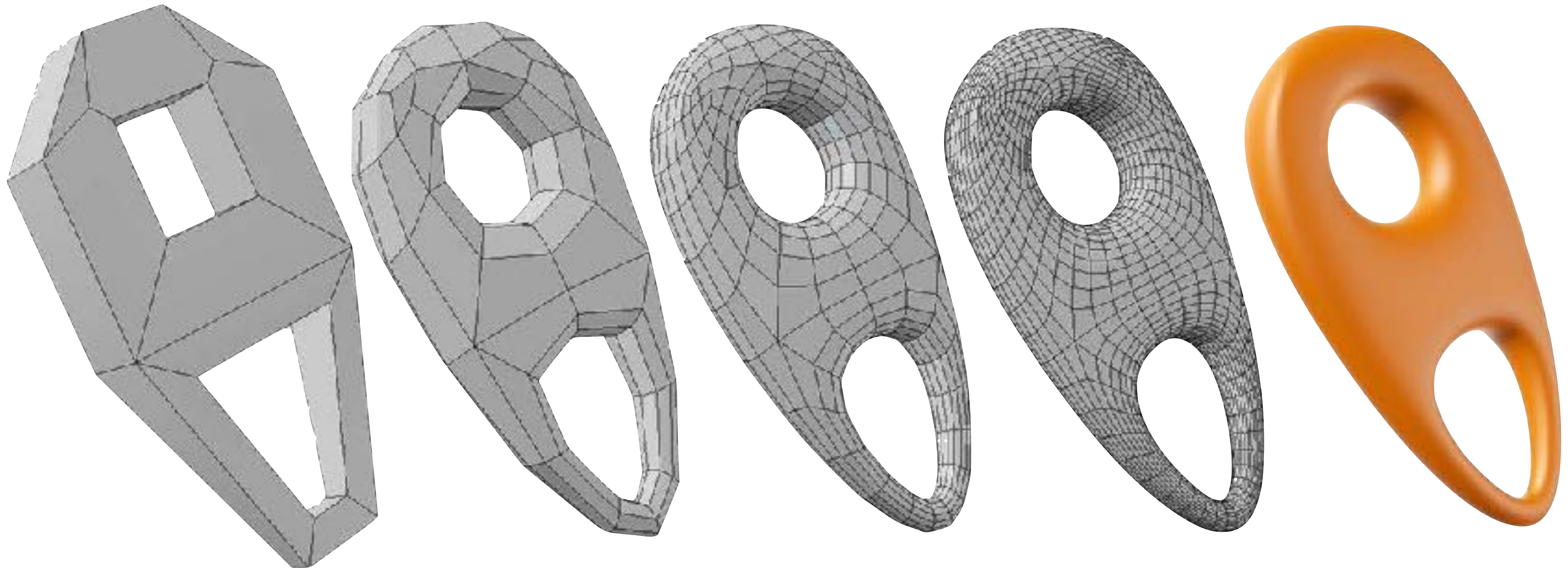


Ray-shape intersection, intersecting transformed shapes, shadow rays, recursive ray tracing, reflection and refraction, …

Modeling, Bézier splines, subdivision, …

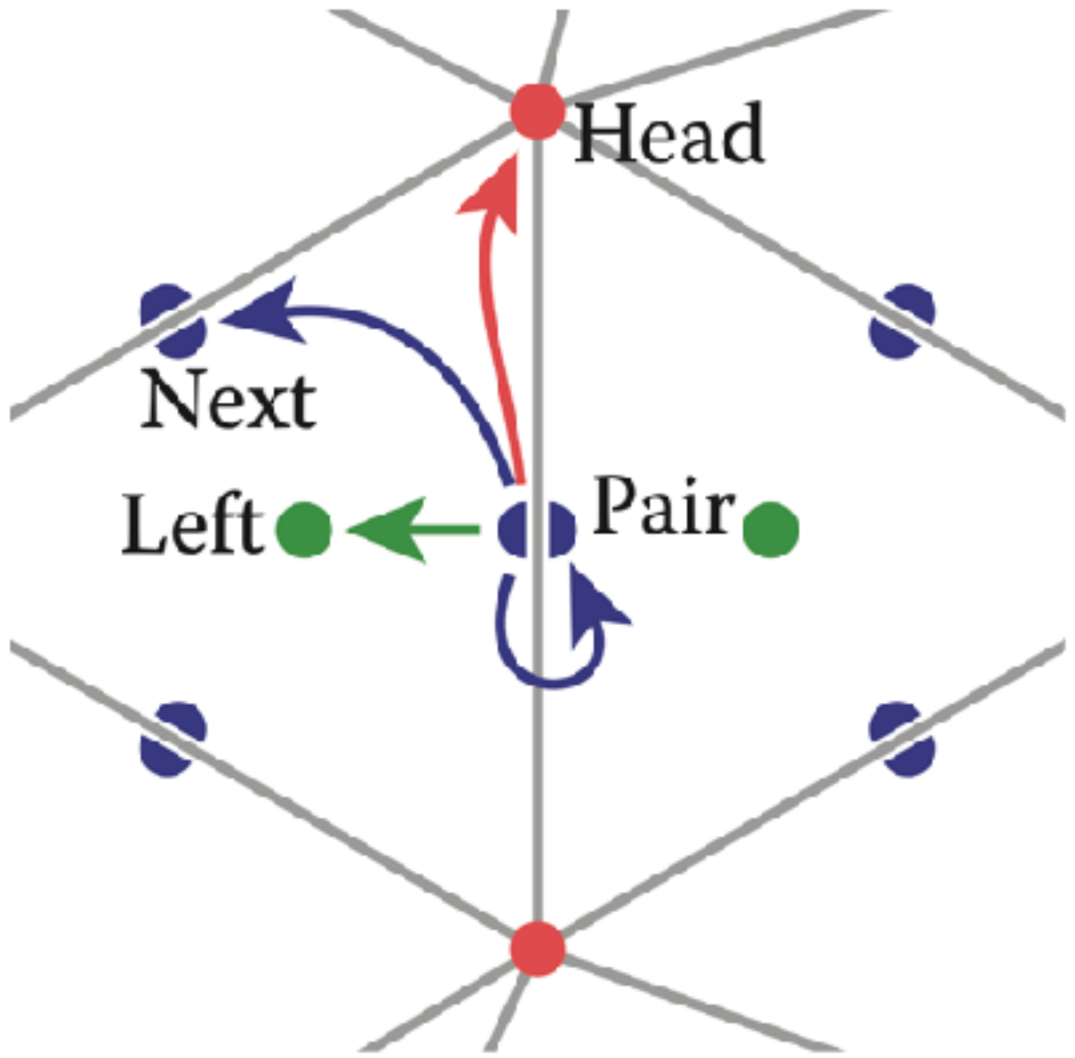


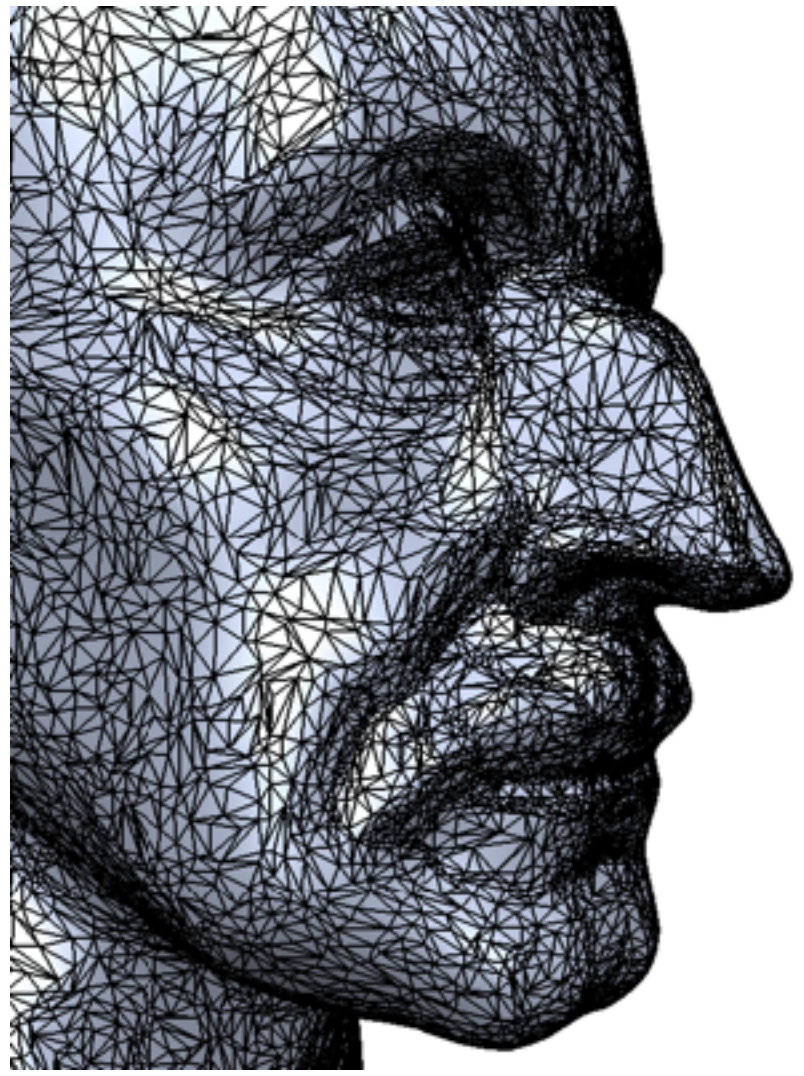


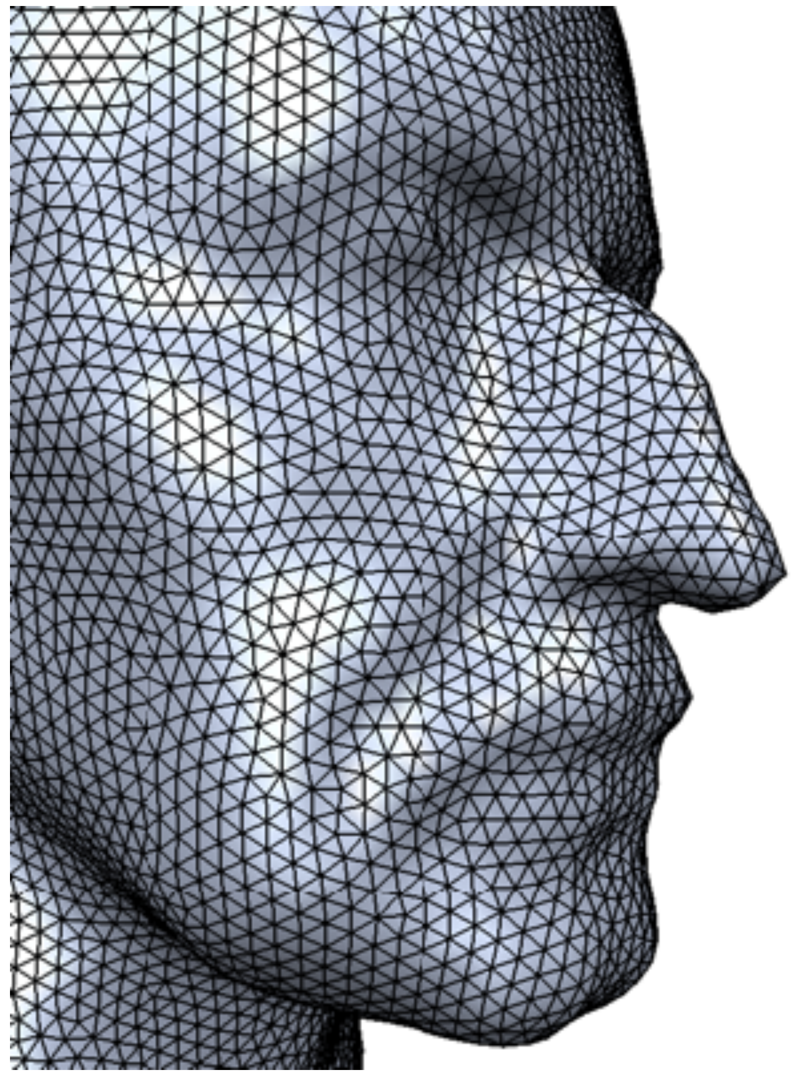


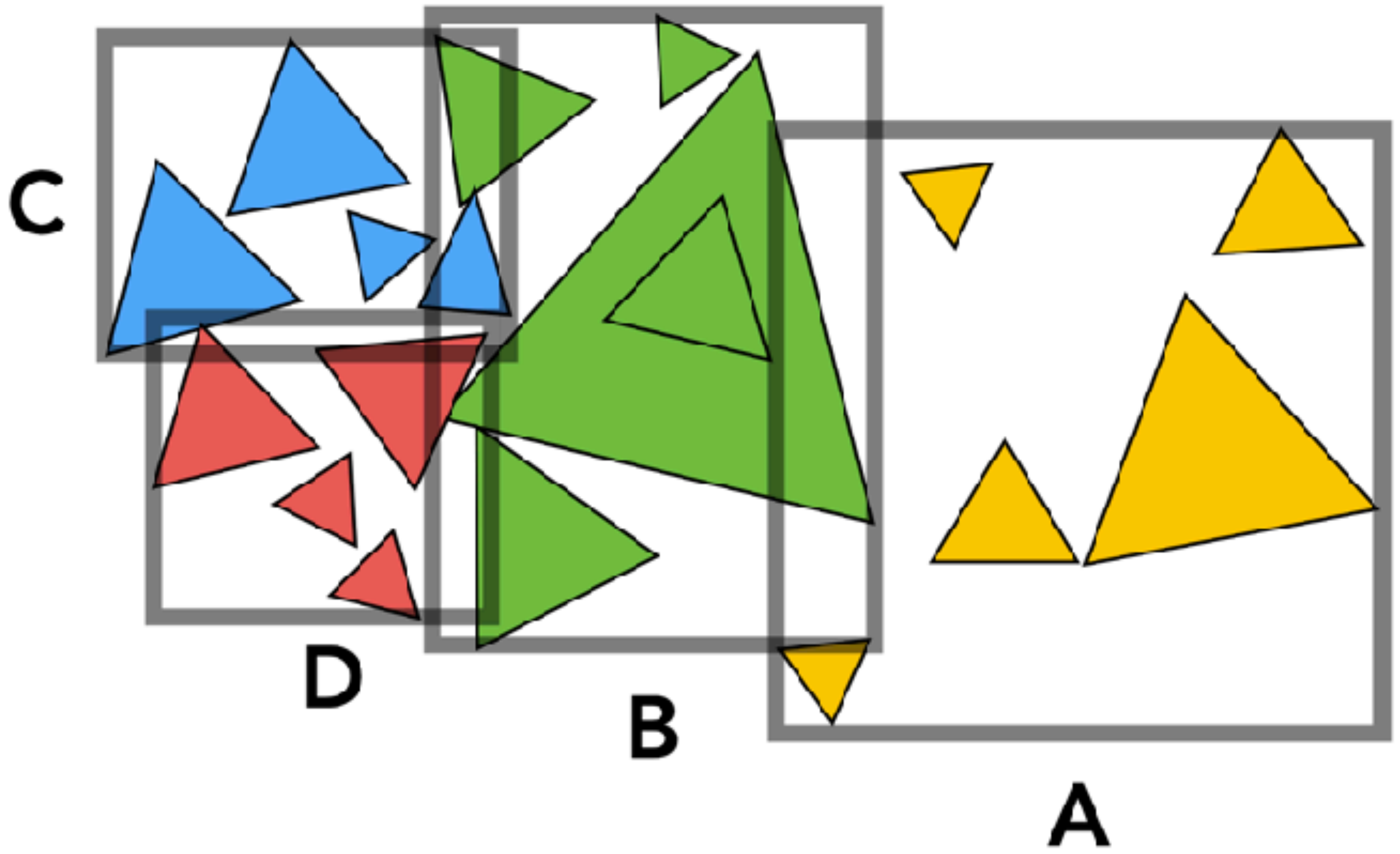
Explicit vs. implicit representations, splines, procedural vs. analytical forms, continuity, subdivision surfaces, …

Meshes, editing, spatial data structures…



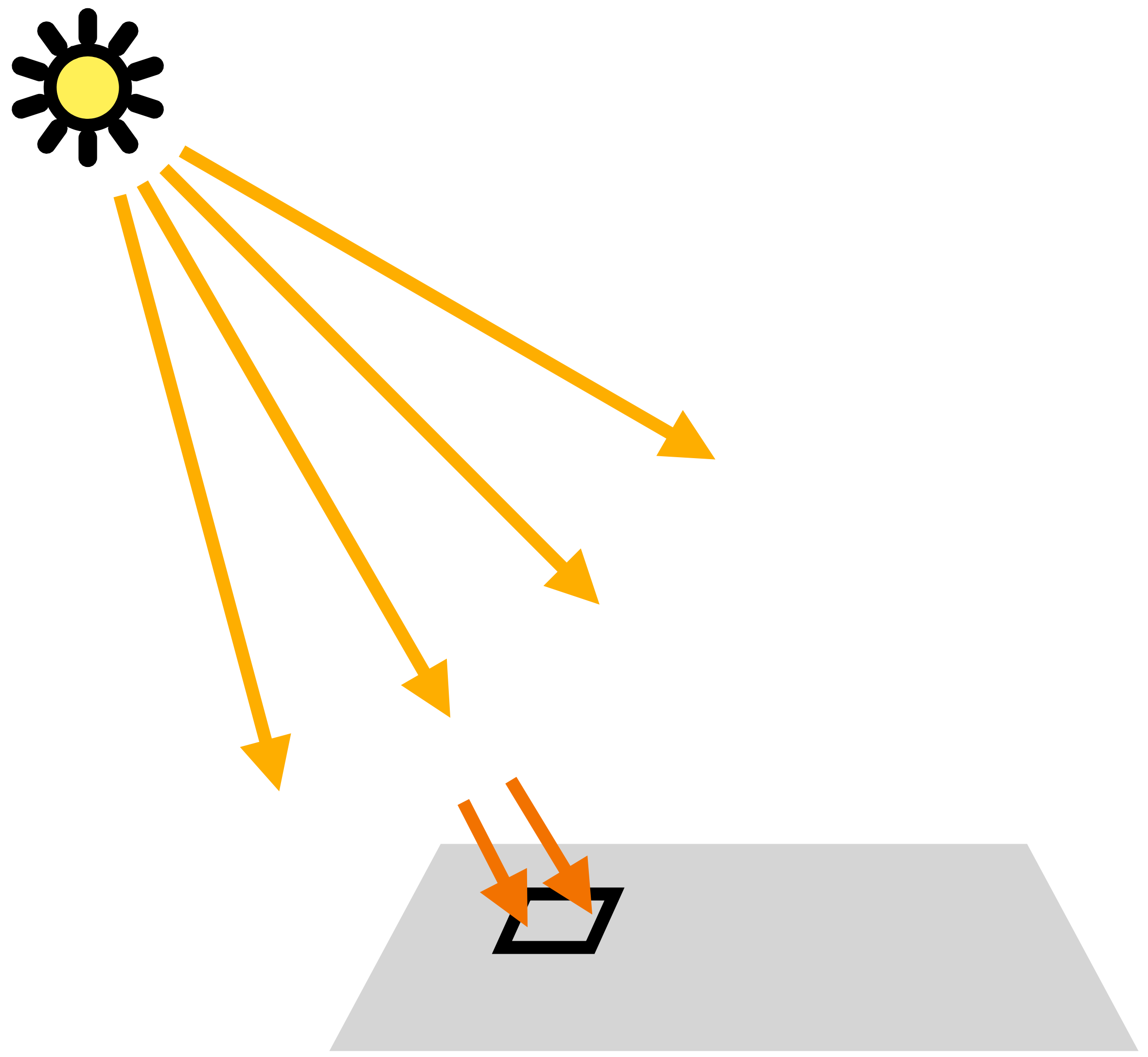


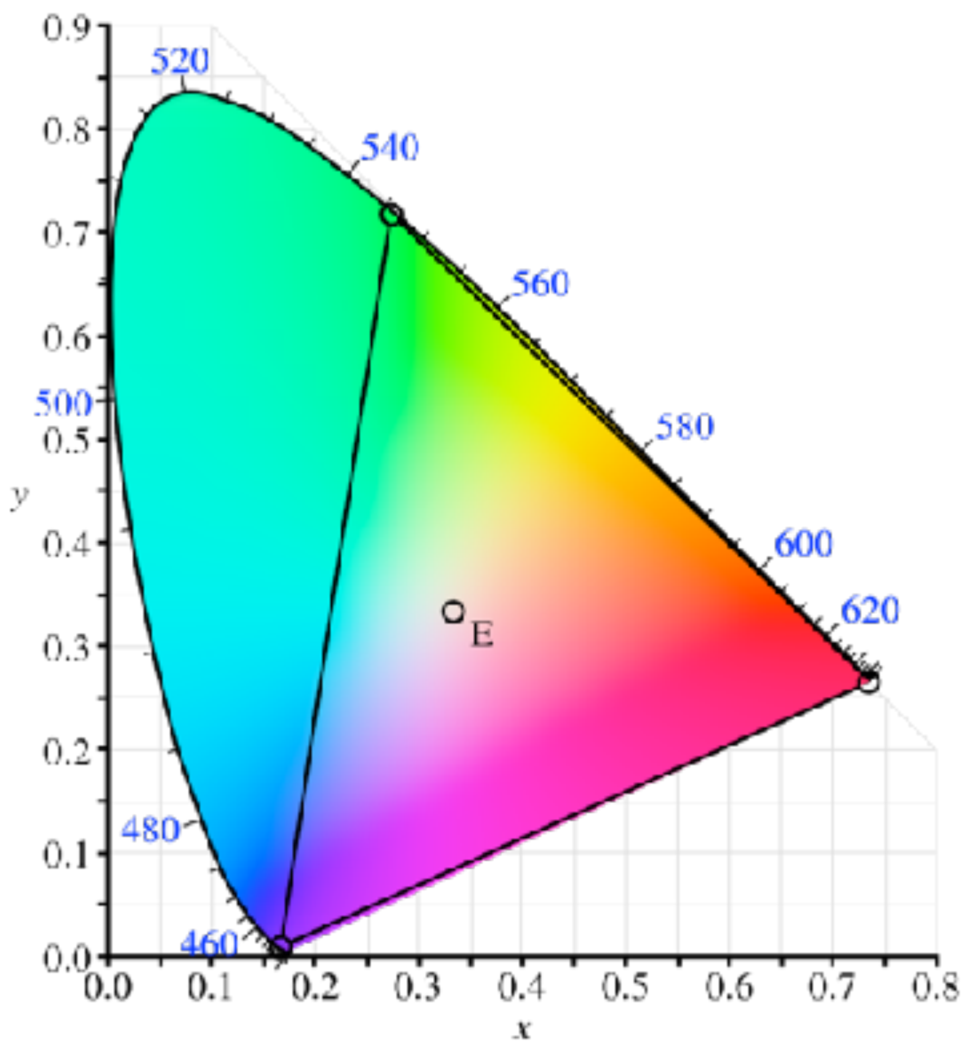




Manifoldness and orientation, connectivity vs. geometry, local operations, geometric queries, bounding volumes, space partitioning, recursive traversal, …

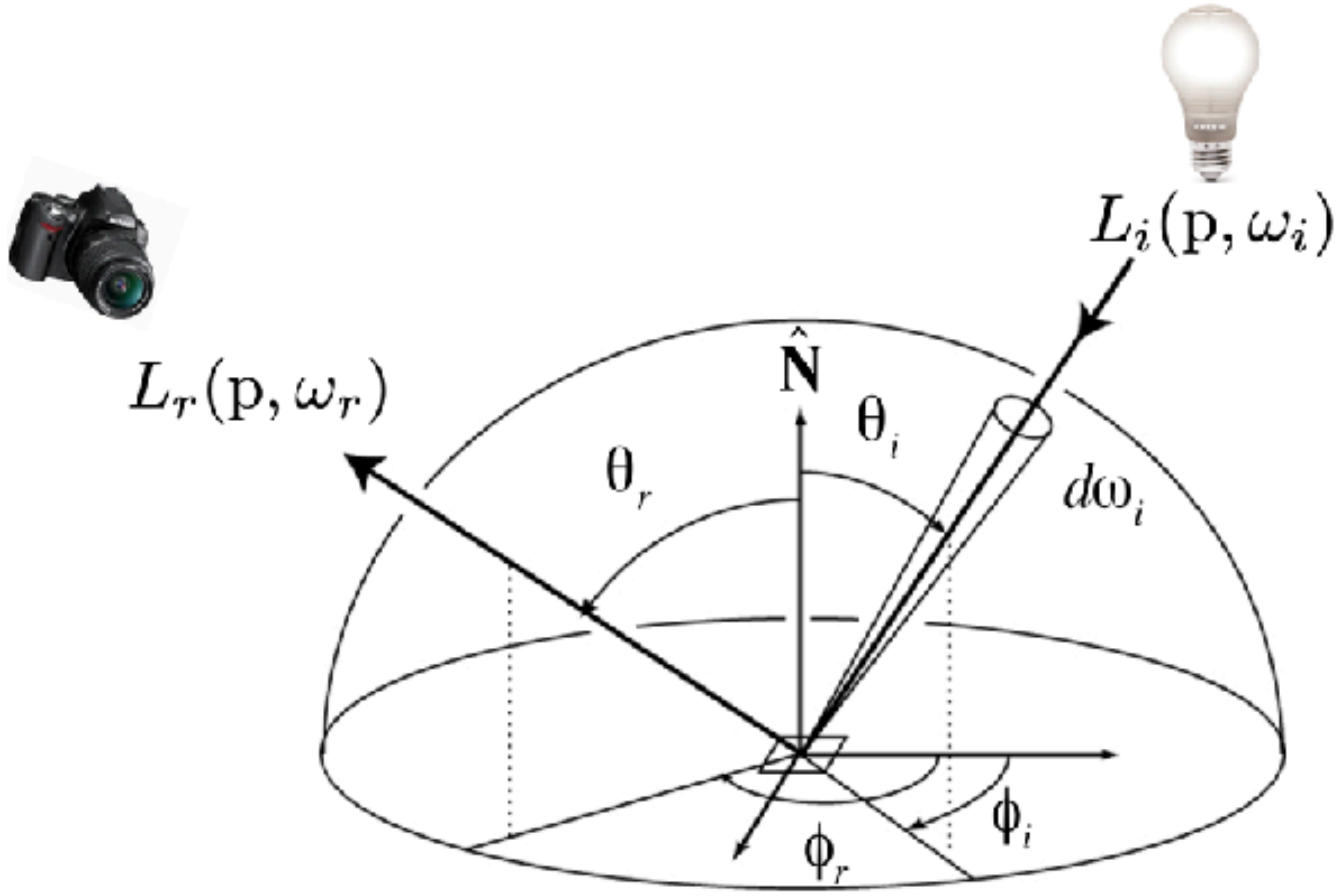
Radiometry, colour, materials, …

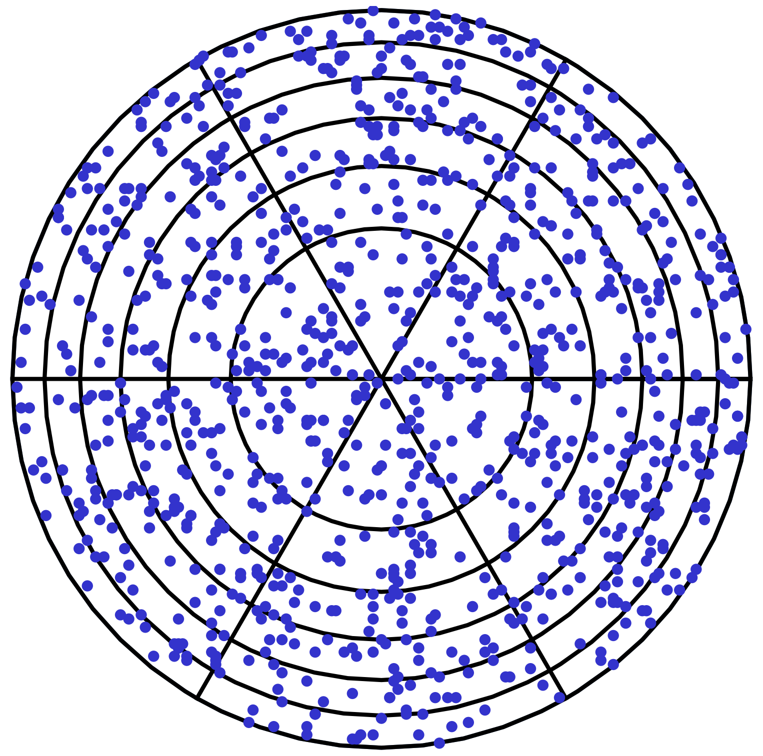


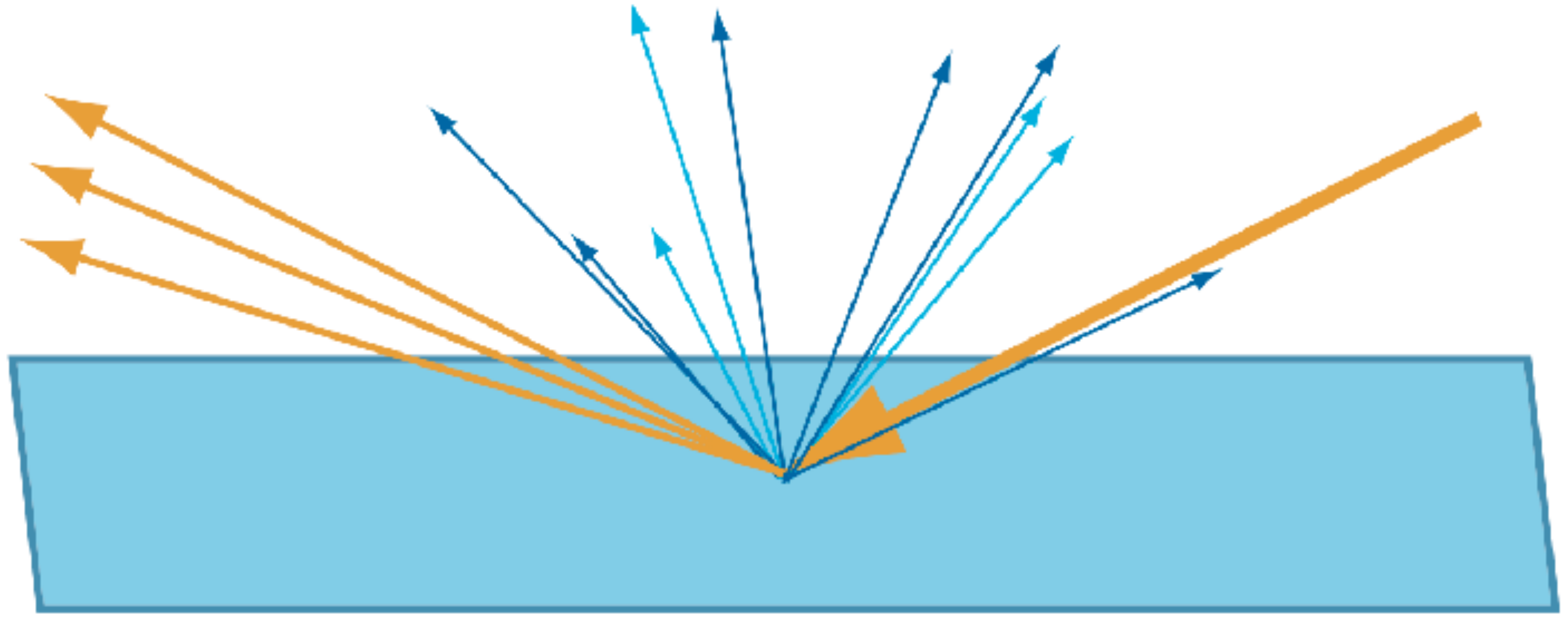


Radiant flux, irradiance vs. radiance, tristimulus values, gamma correction, BRDFs, microfacet models, Fresnel reflectance, …

The rendering equation, path tracing, …

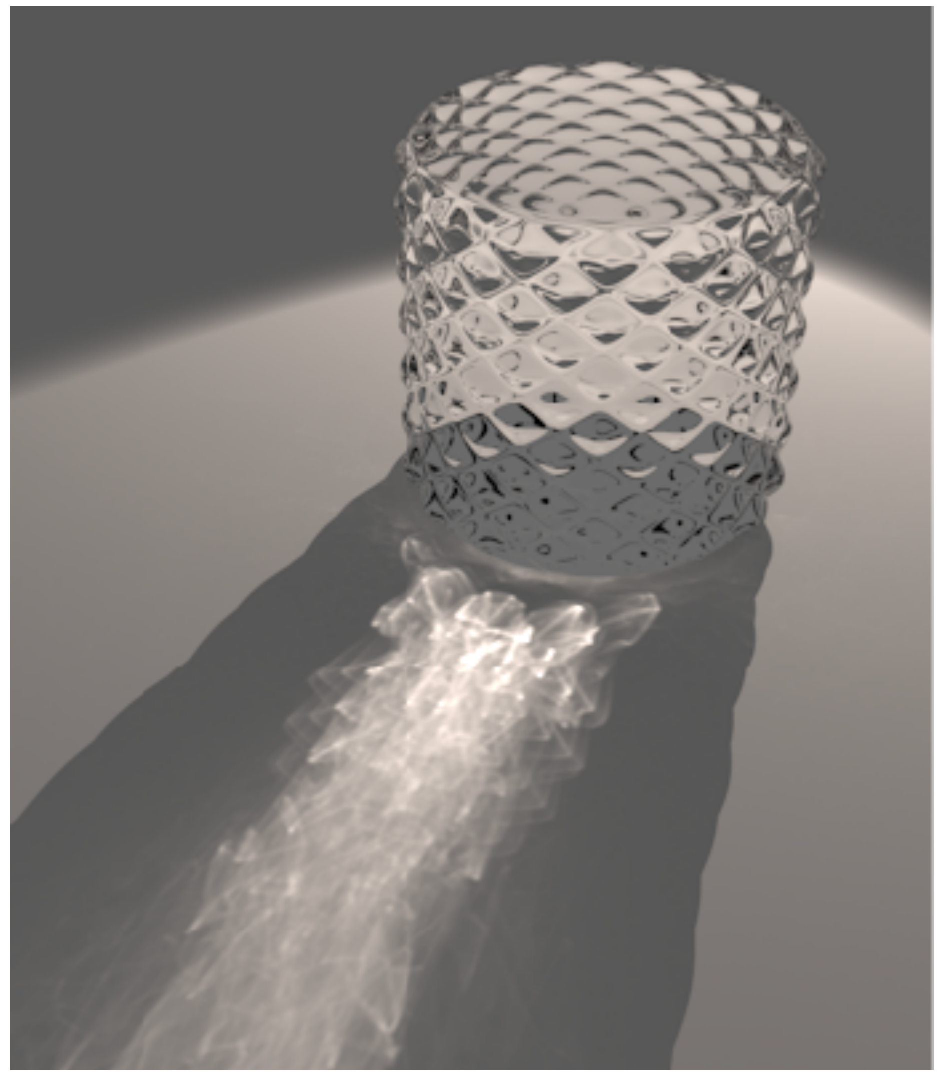
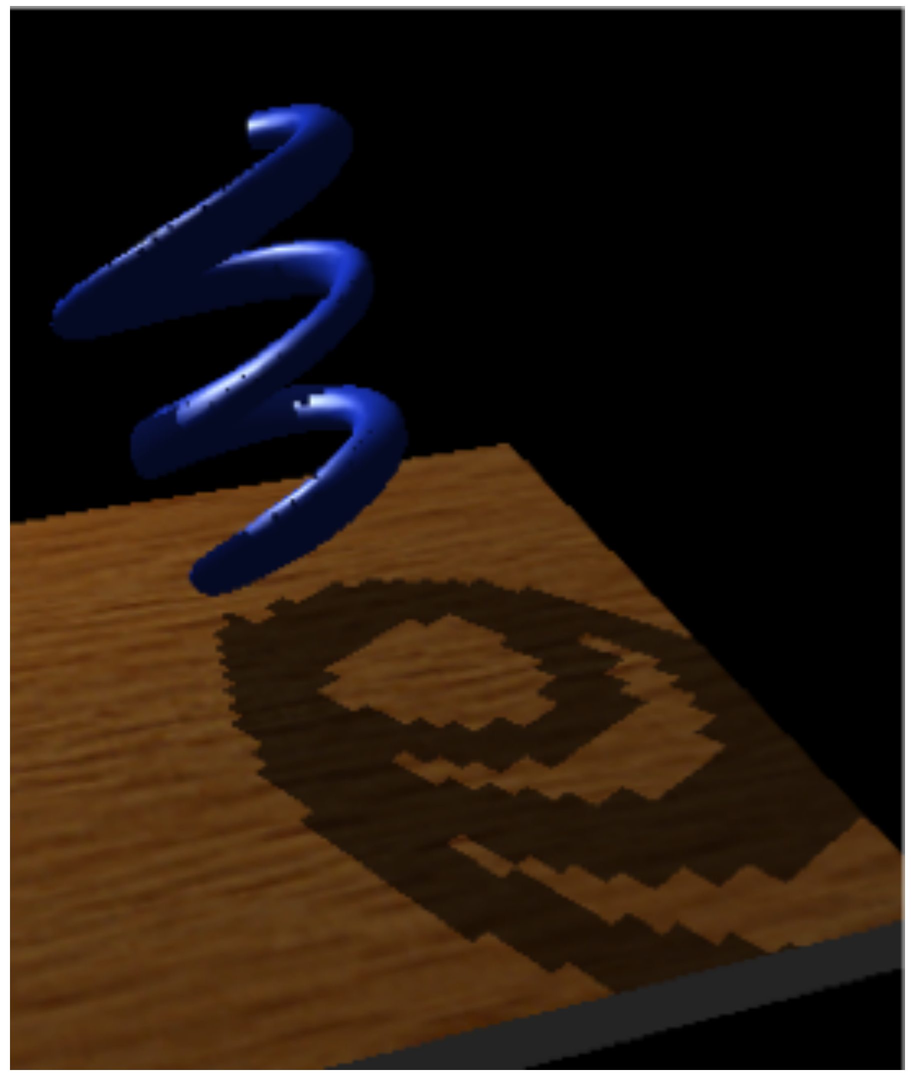






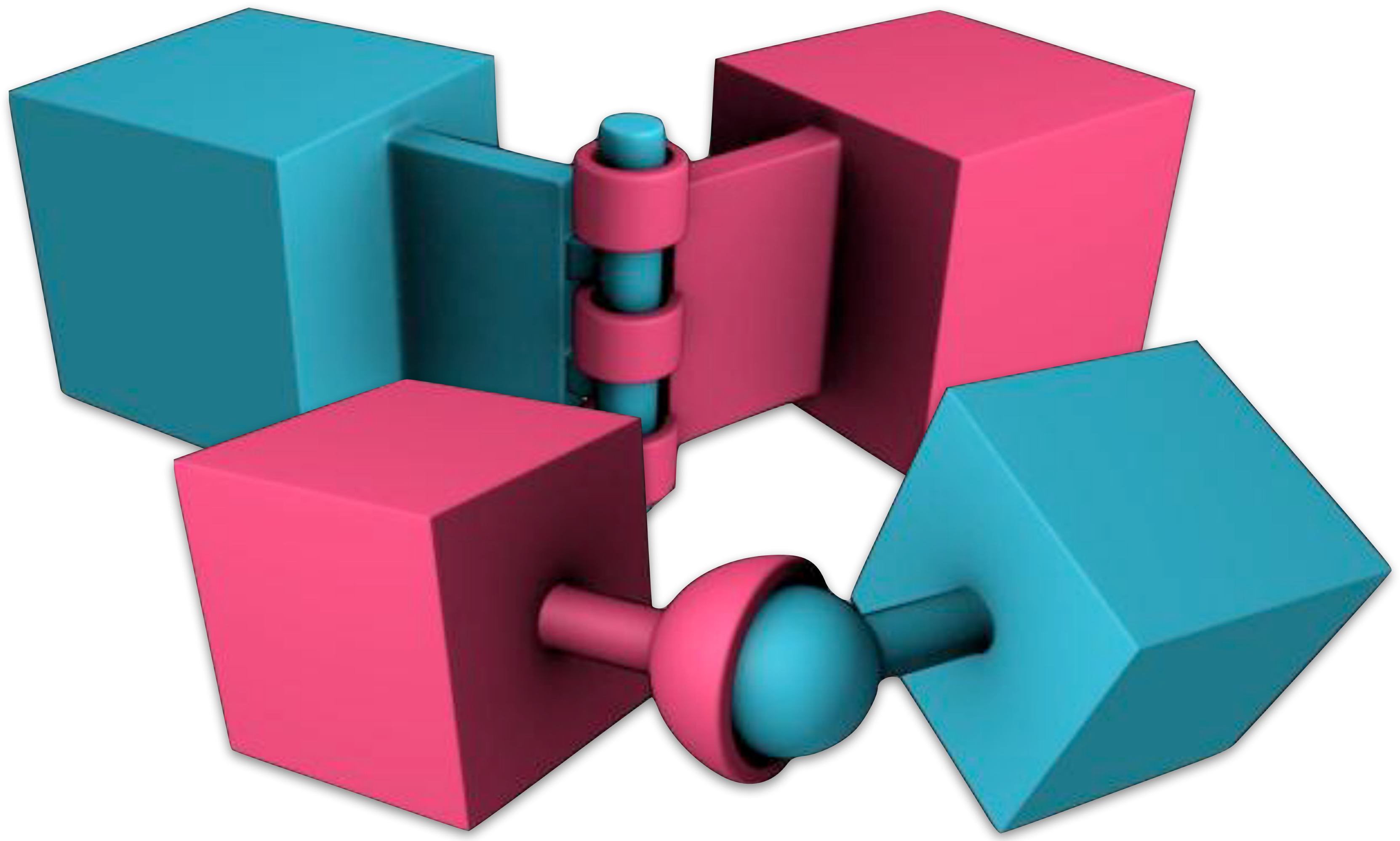
Global illumination, Monte Carlo integration, path tracing, inversion vs. rejection sampling, Russian roulette, importance sampling, …

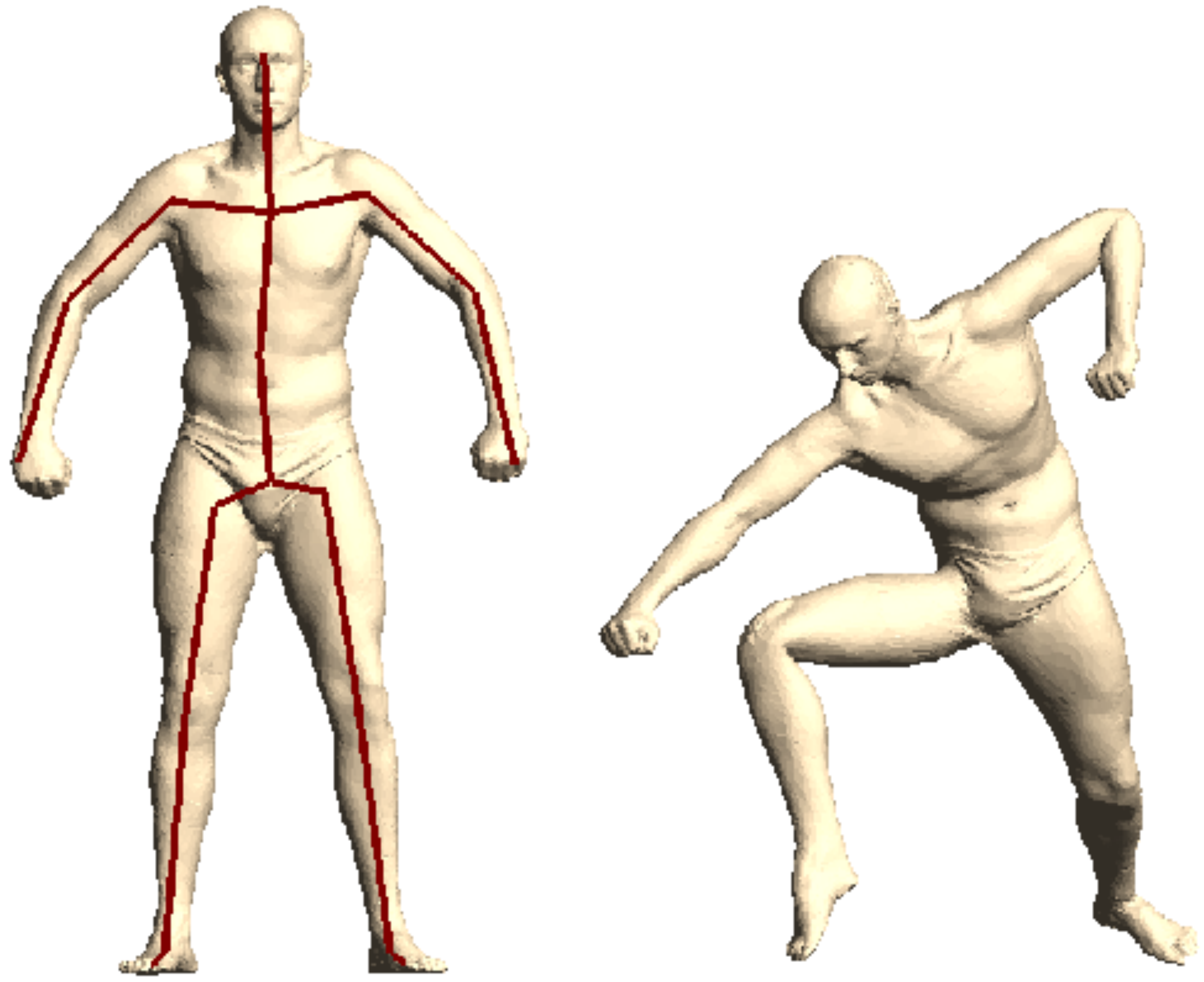
Bidirectional methods, real-time rendering, …

Variations of path tracing (independent samples) vs. photon mapping (reuse of light paths), gathering data from the right viewpoint, precomputed shading, …

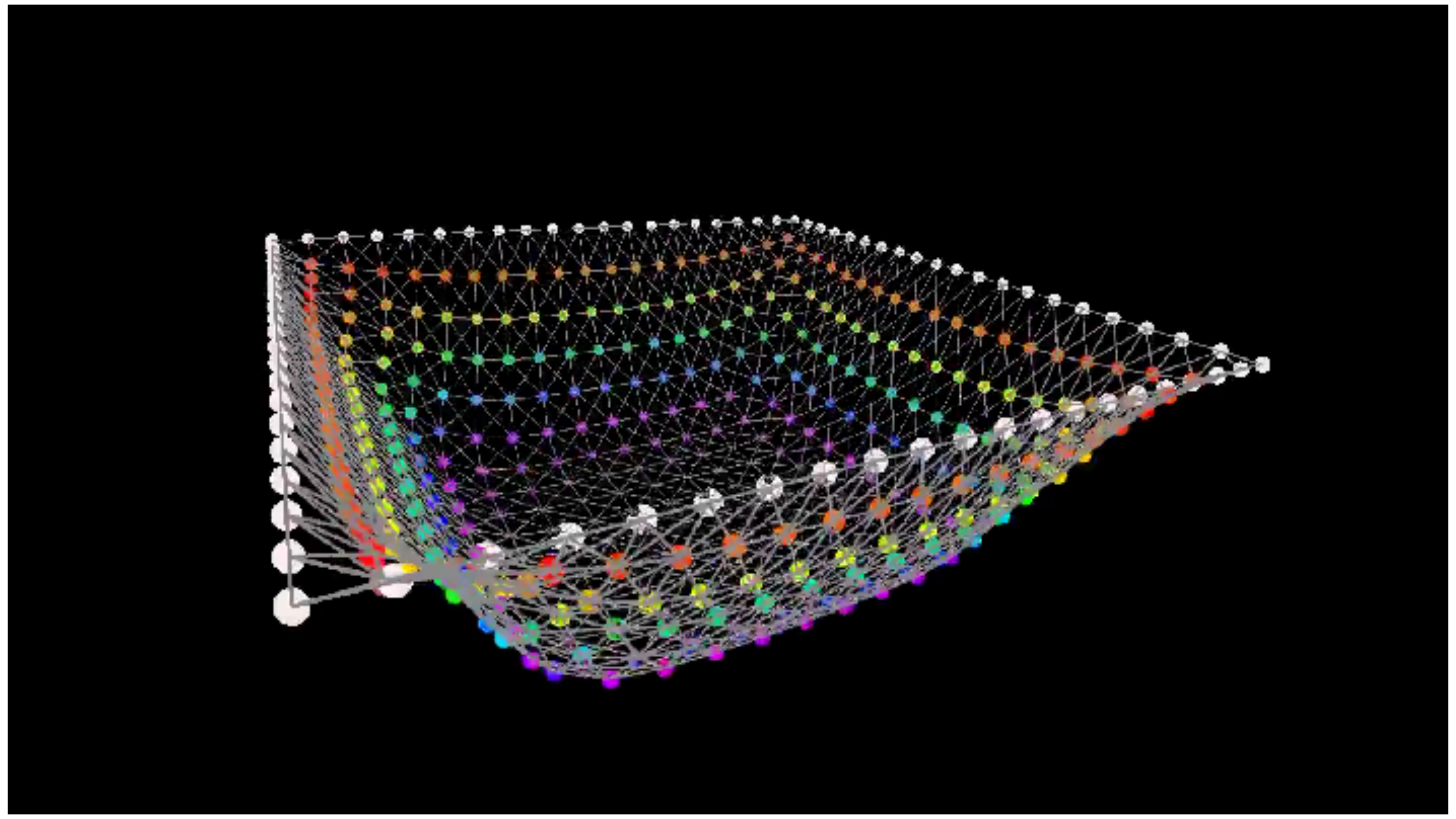
Skeletal animation, skinning, ...

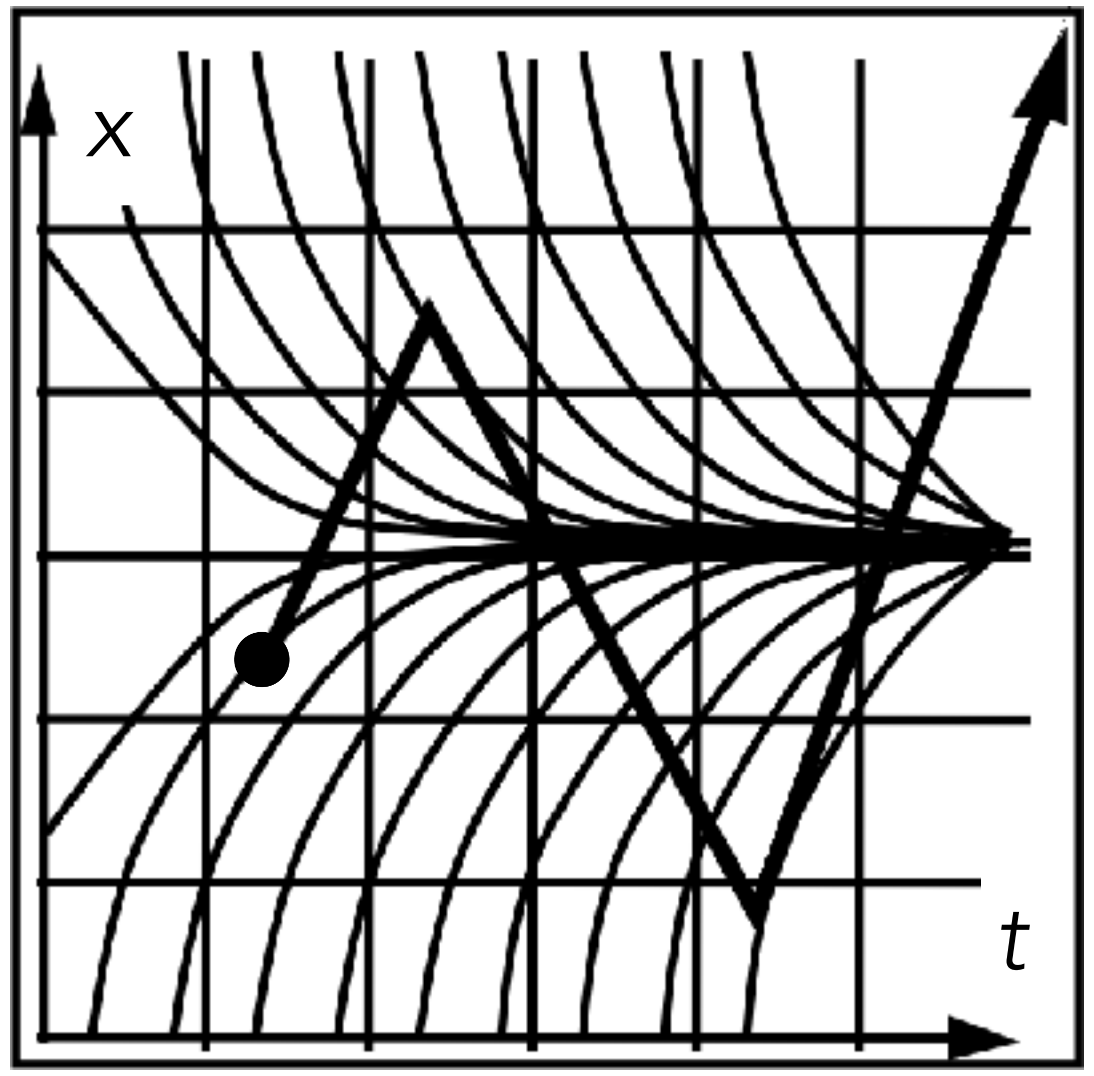




Animation controls, keyframing vs. motion capture, quaternions, forward vs. inverse kinematics, linear blend skinning vs. dual quaternions, ...

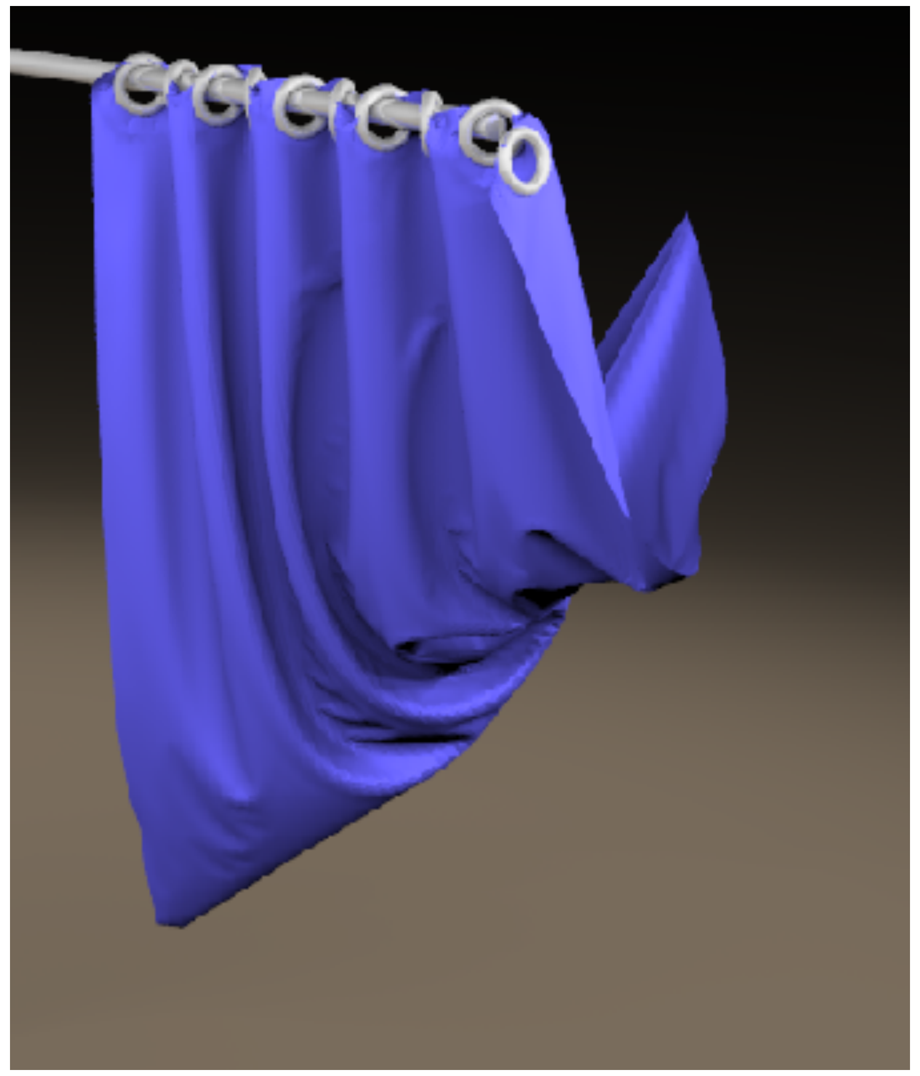
Particles, mass-spring systems, time stepping, …

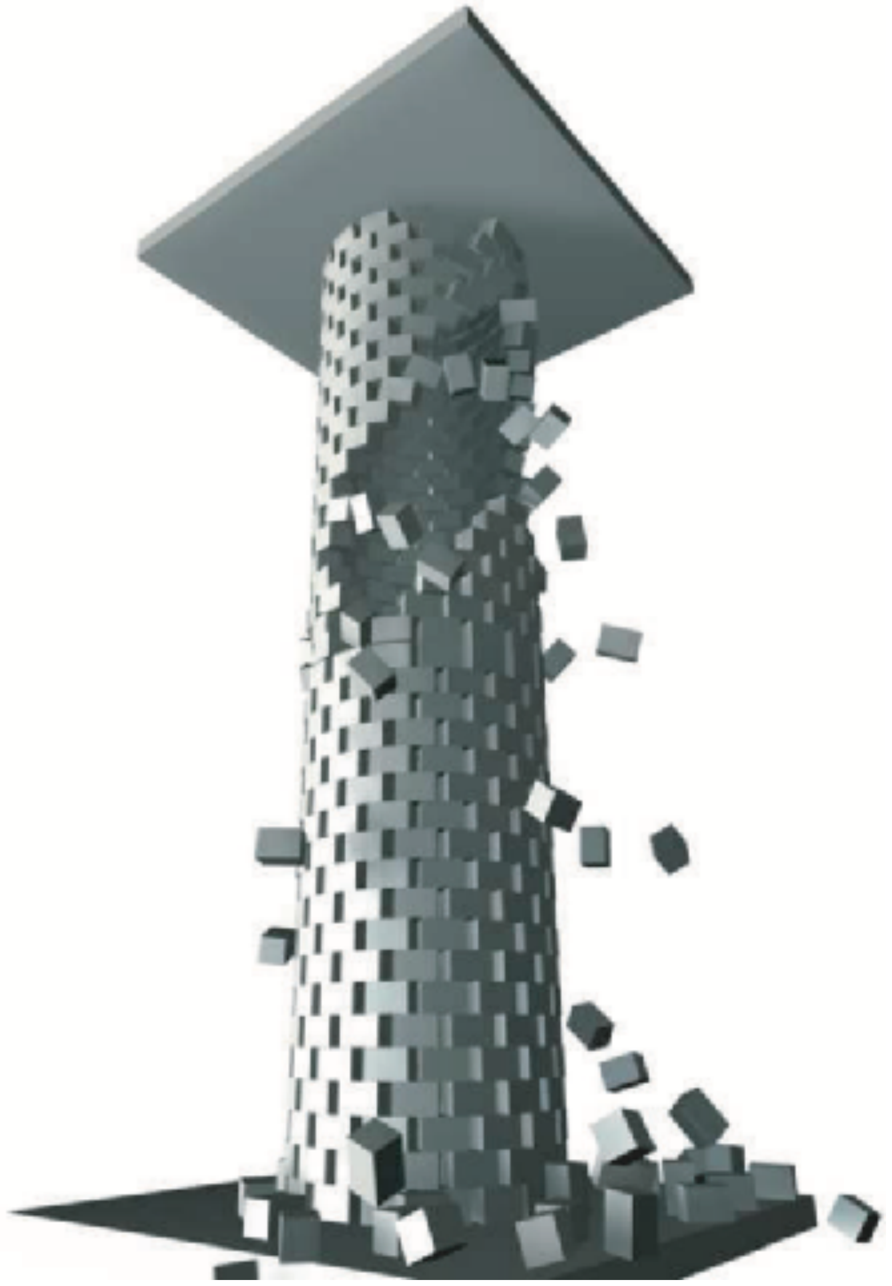


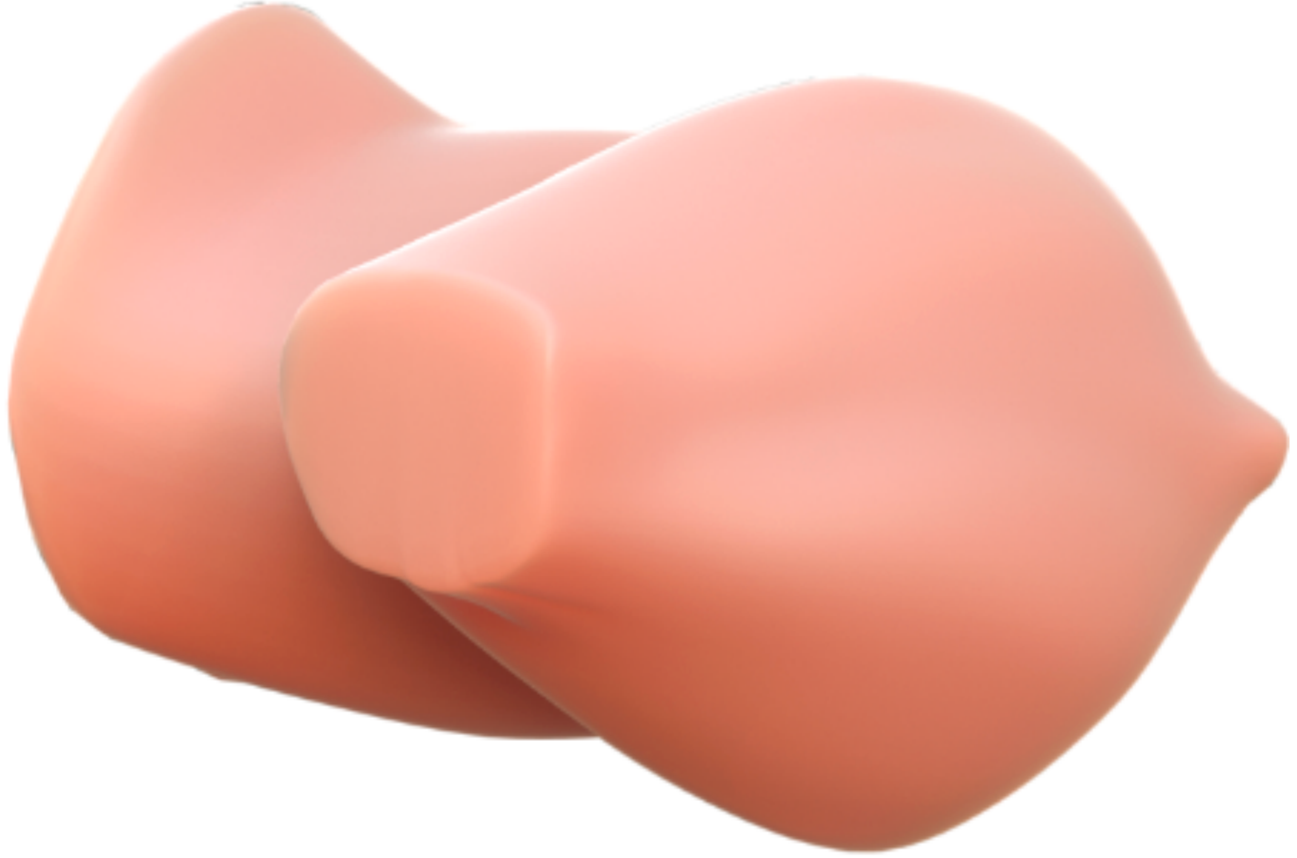


Time stepping, inter-particle interactions, generalized coordinates, forces from potentials, strains, implicit integration, Newton's method, …

Constraints, collisions, continuum models, …







Constraint projection, rigid body dynamics, collision detection vs. response, Laplacian operator, discretization, finite elements, splitting methods, …

Questions?