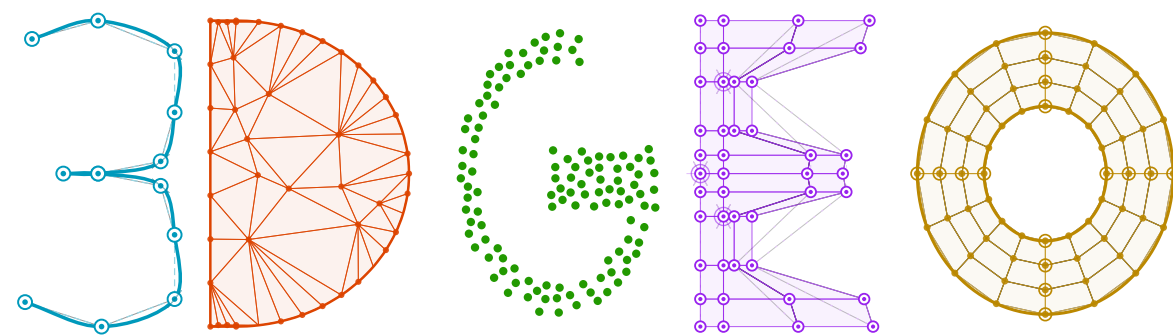


15. Előadás: Inverz Grafika Alapjai

Generatív AI és Inverz Módszerek a Képszintézisben
BME-VIK IIT, 2026

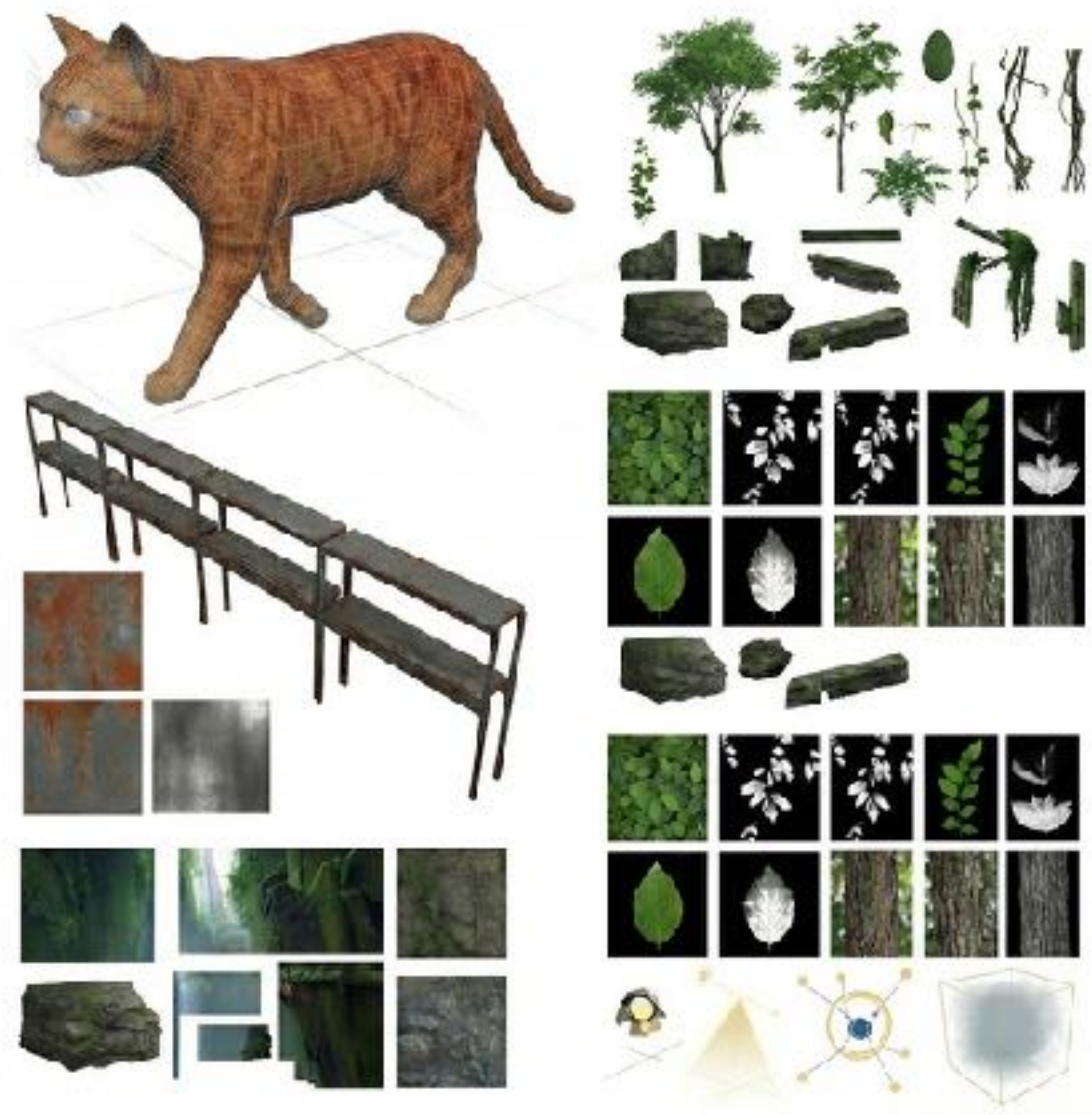


Dr. Vaitkus Márton



Inverz Grafika

Alapfeladat

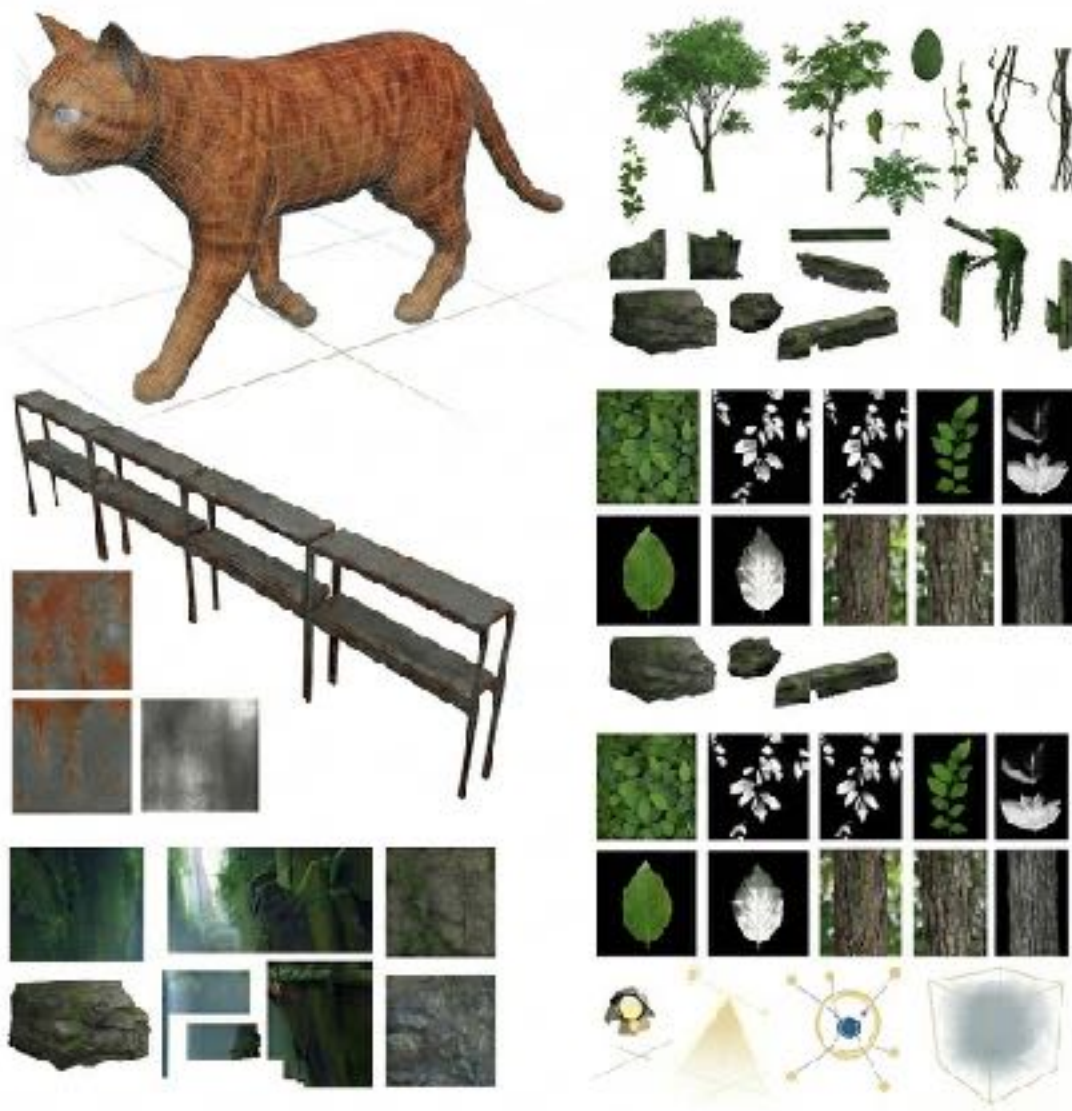


3D Modellek

Inverz Grafika

Alapfeladat

Grafika / Rendering



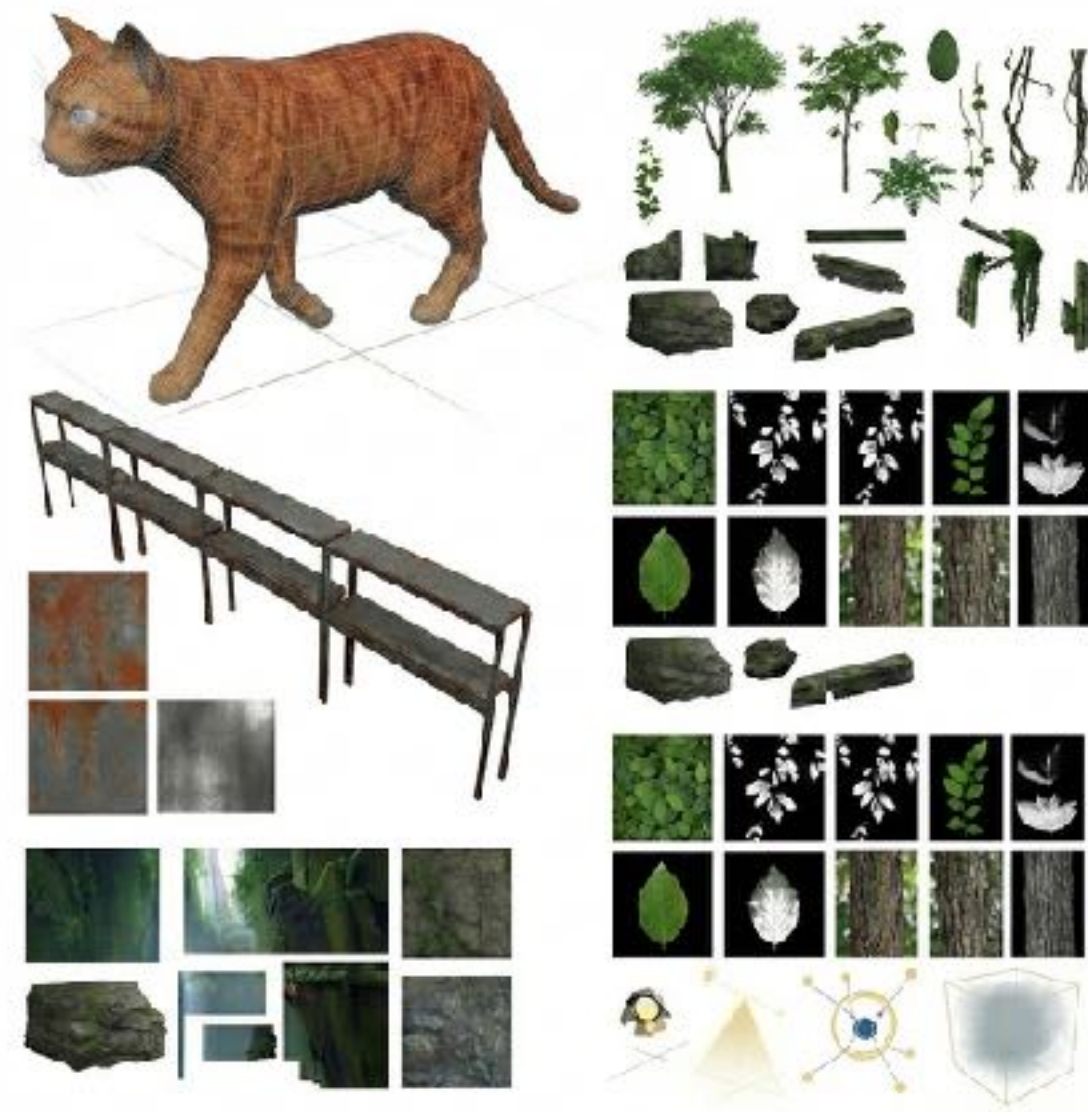
3D Modellek



Képek

Inverz Grafika

Alapfeladat



3D Modellek

Grafika / Rendering



Képek

Inverz Grafika / Rendering

Inverz Grafika

Alapfeladat



3D Modellek

Grafika / Rendering



Képek

Inverz Grafika / Rendering



Inverz Grafika

Alapfeladat



3D Modellek

Grafika / Rendering



Képek

Inverz Grafika / Rendering



Hogyan
lehetséges?

Inverz Grafika

Alapfeladat



Bemeneti Képek



Rekonstruált 3D Modell



Szintetizált Képek

Inverz Grafika

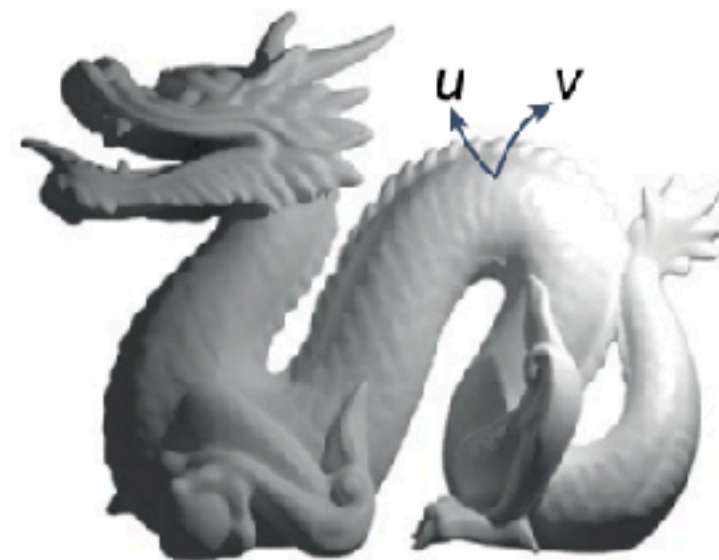
3D Reprezentációk

Surface representations

Mesh



Parametric surface



Point clouds



Signed distance function



Volume representations

Voxel grids

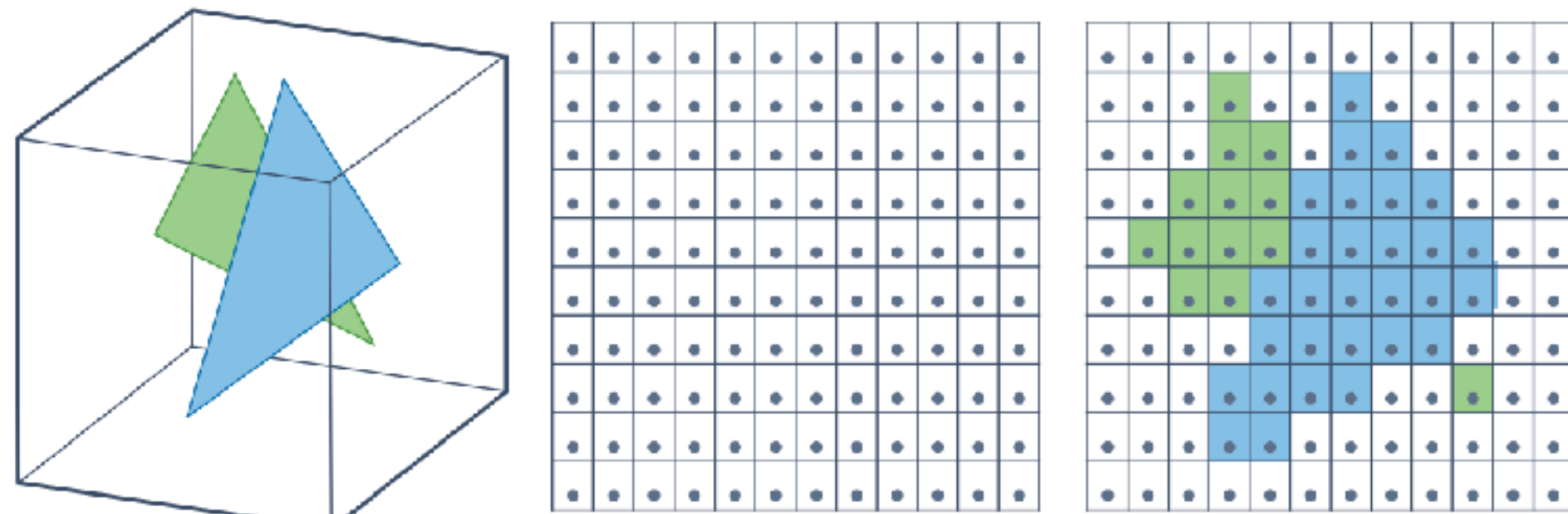


Density and radiance fields

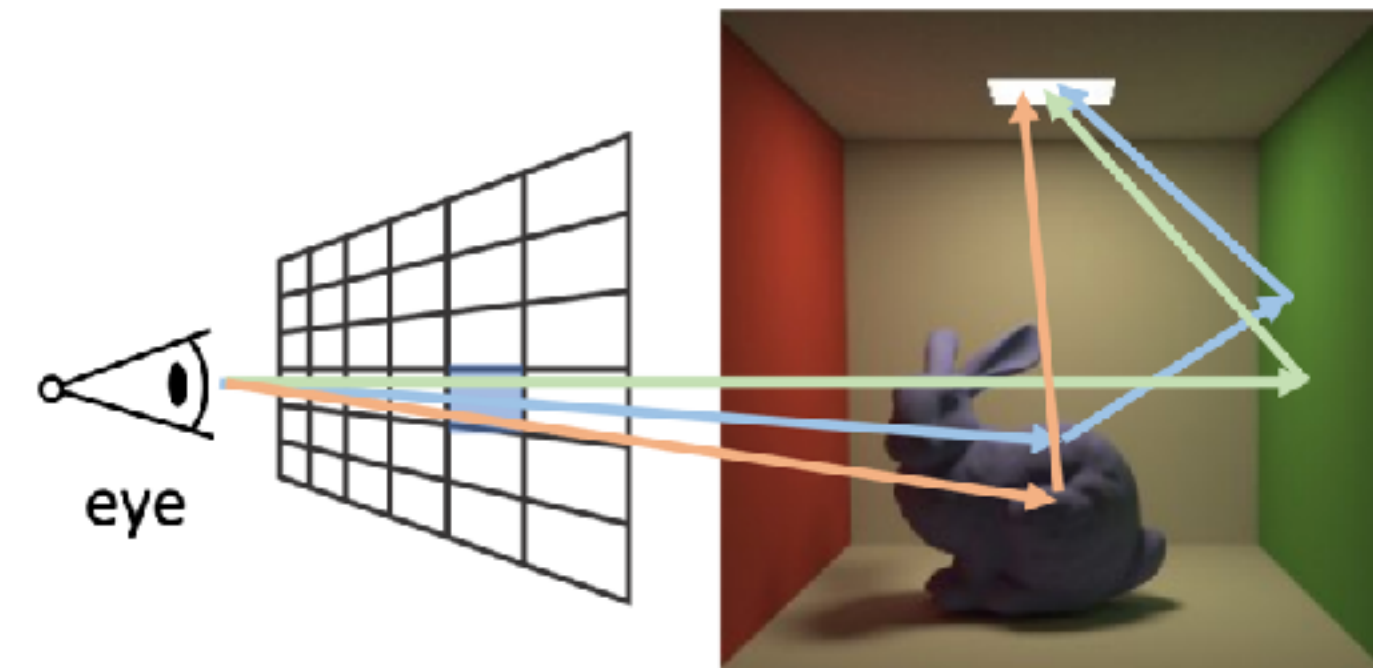


Inverz Grafika

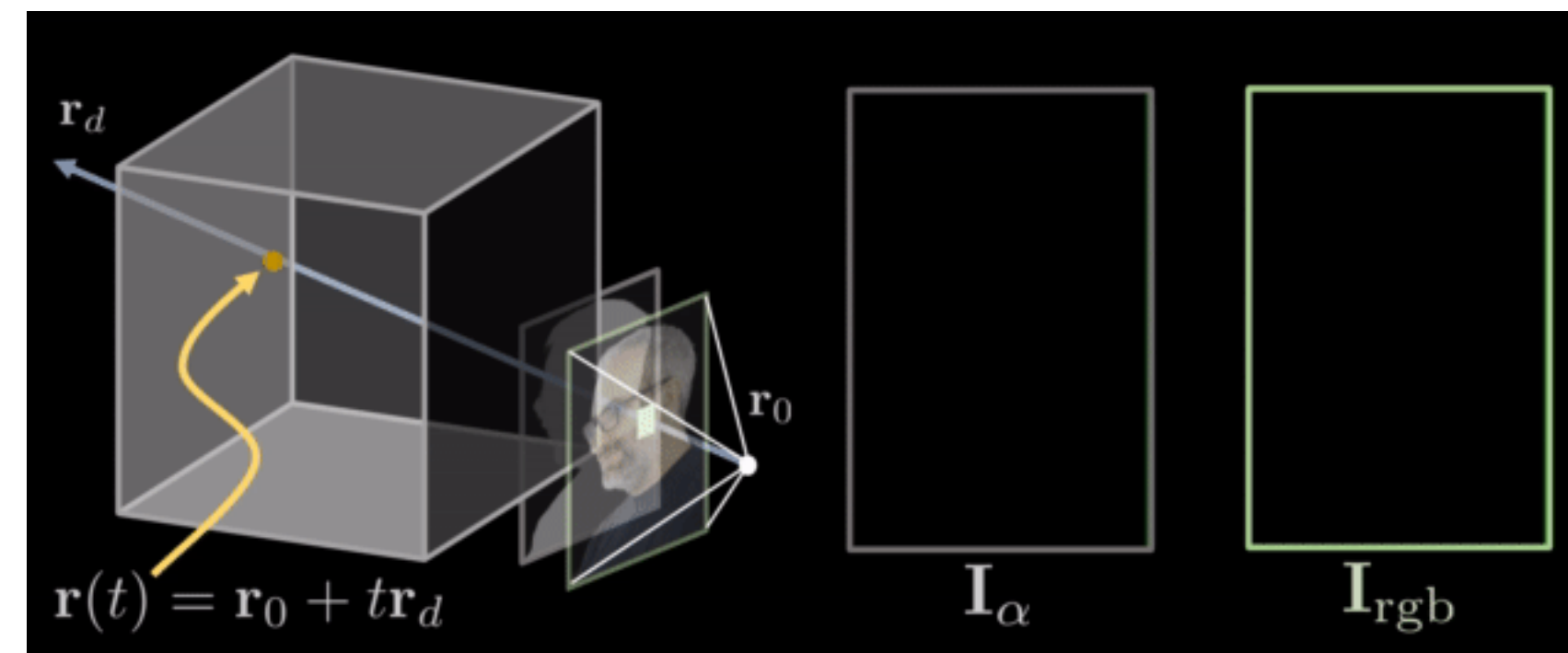
Forward rendering módszerek



Raszterizáció



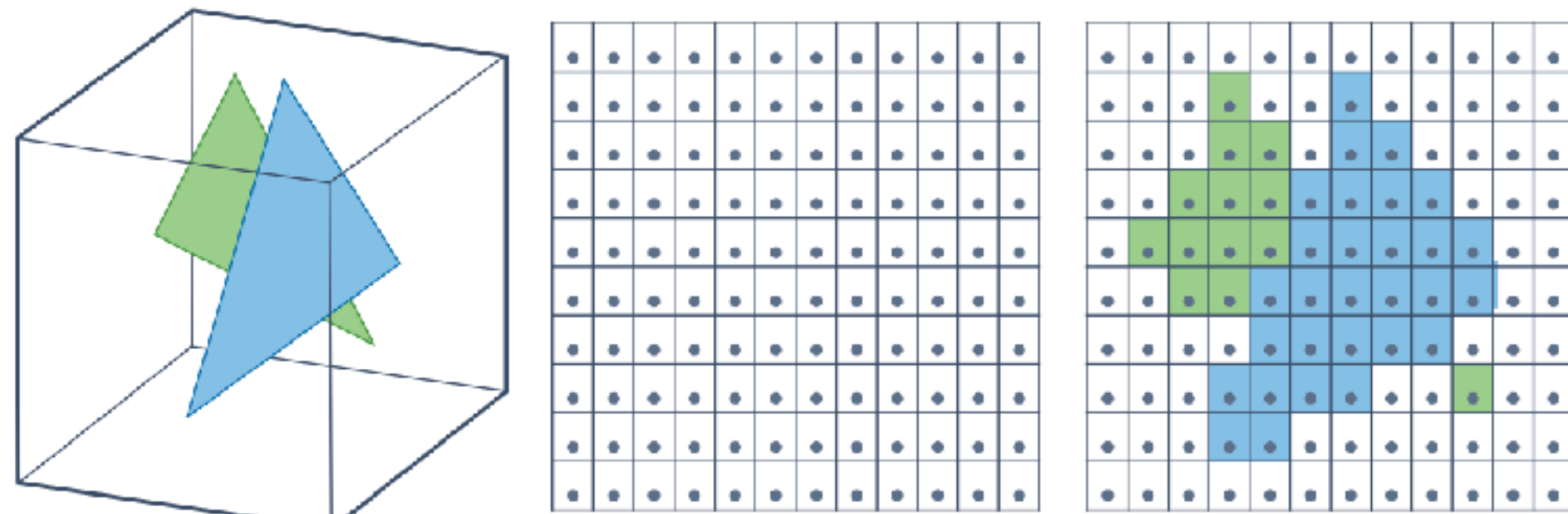
Sugárkövetés (Ray Tracing)



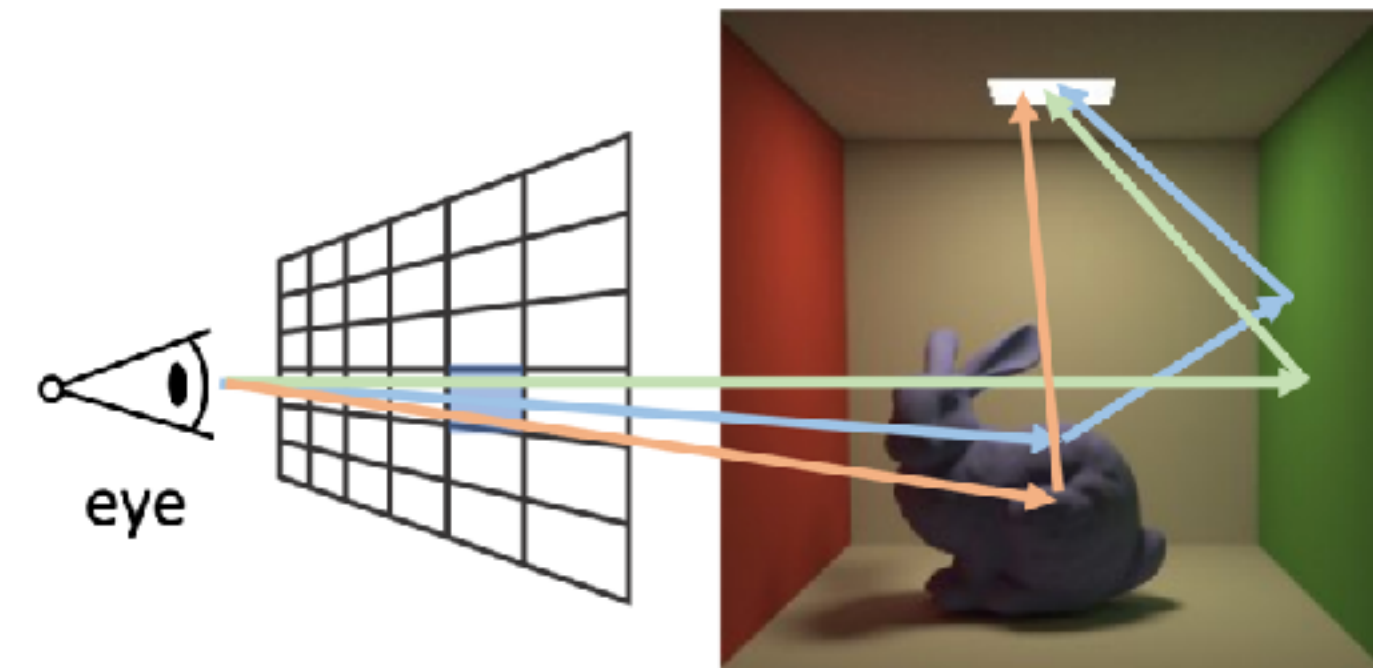
Volumetrikus rendering (Ray Marching)

Inverz Grafika

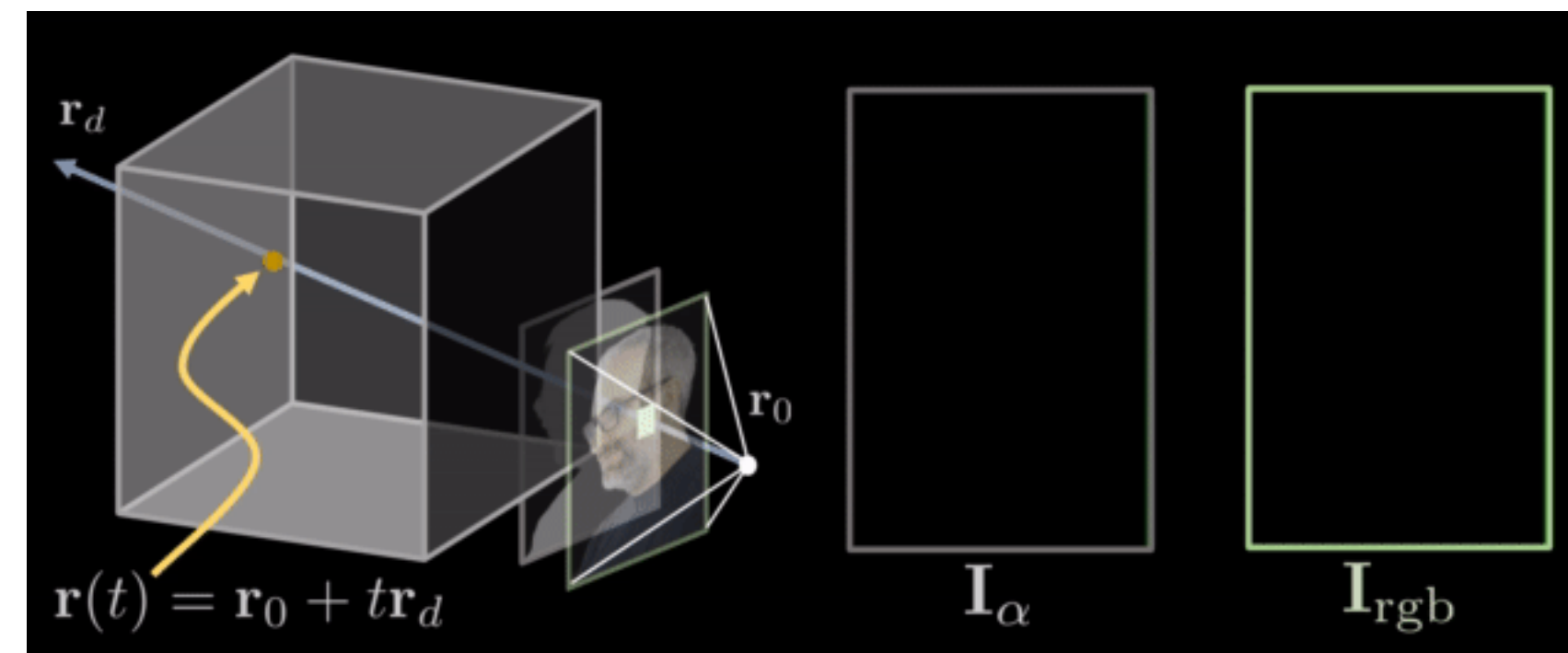
Forward rendering módszerek



Raszterizáció



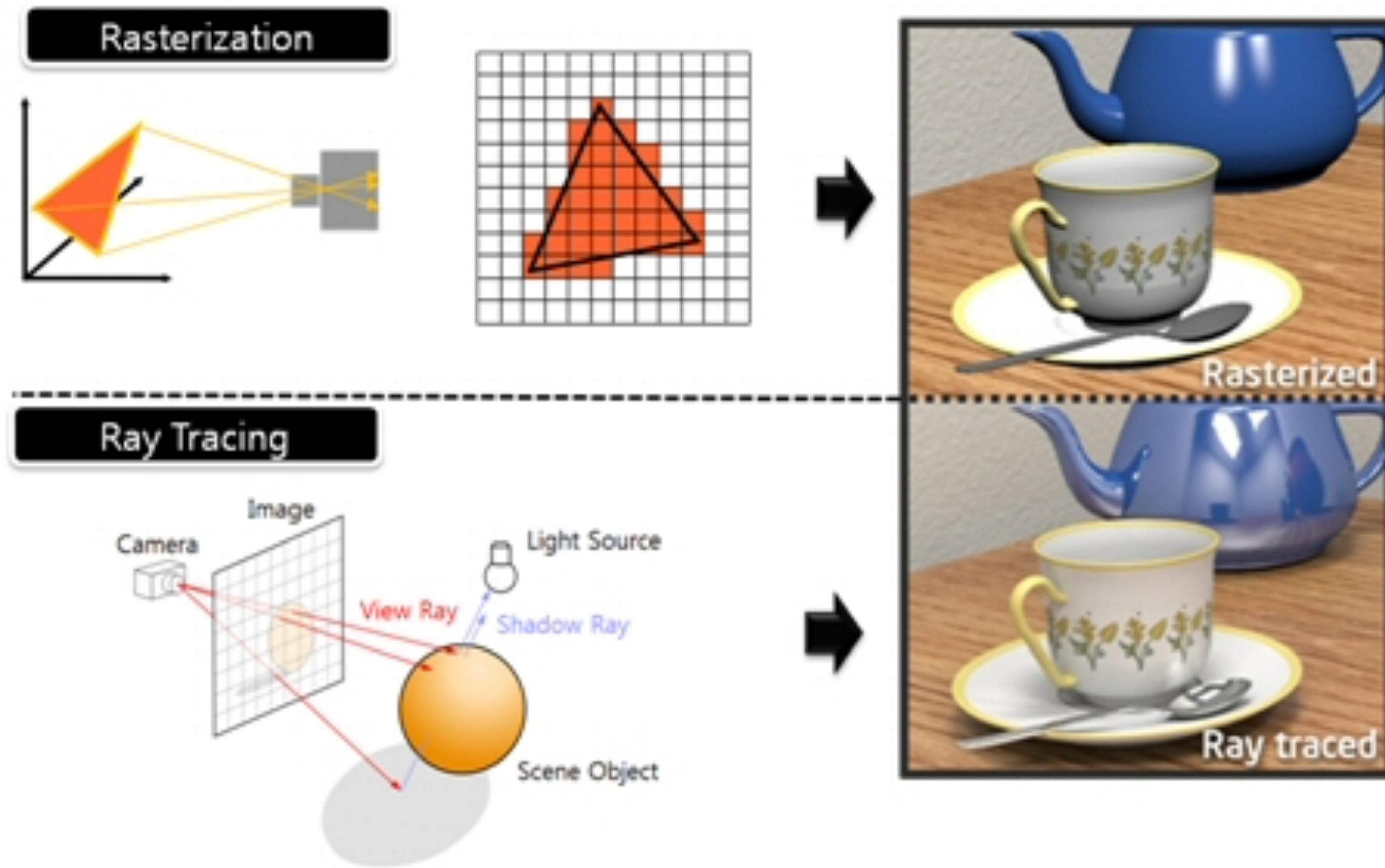
Sugárkövetés (Ray Tracing)



Volumetrikus rendering (Ray Marching)

Rendering Módszerek

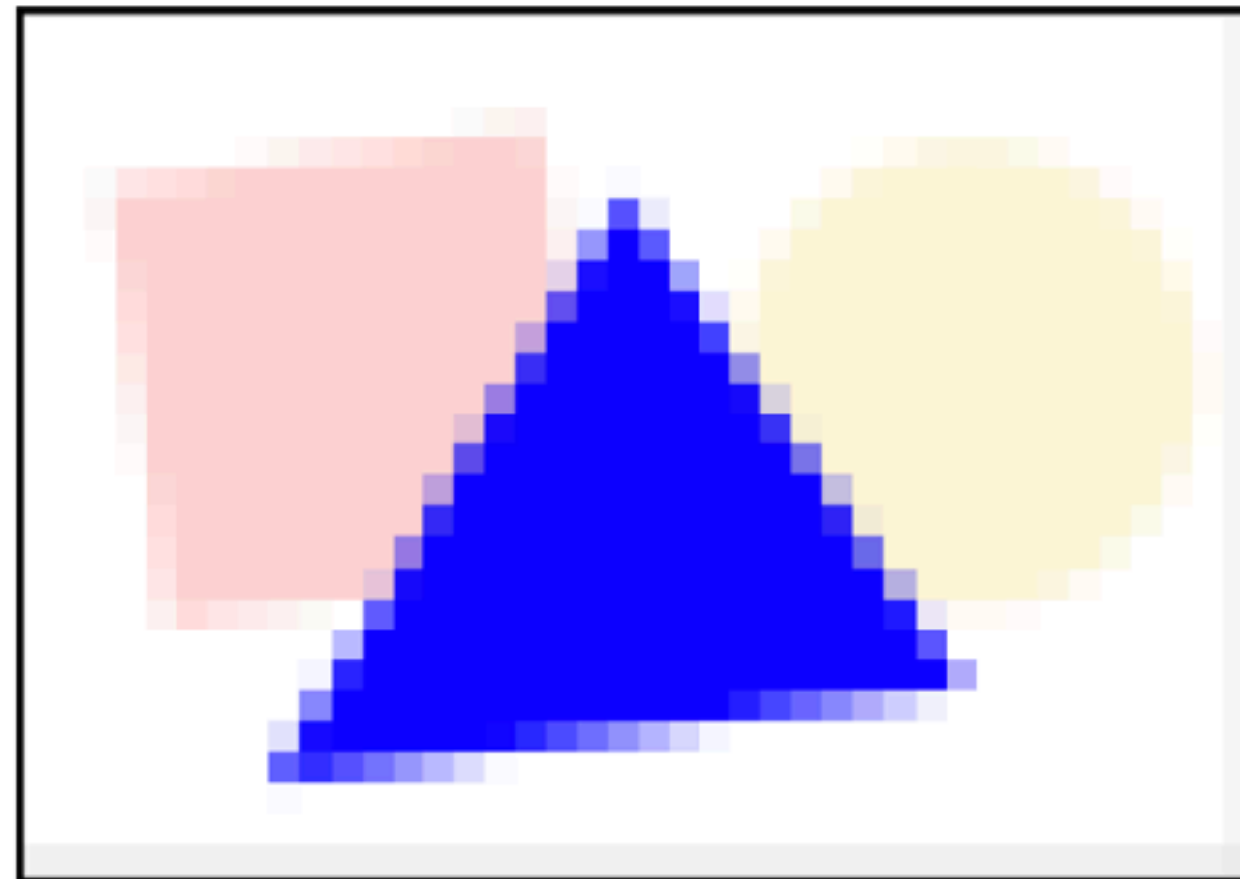
Raszterizáció vs. sugárkövetés



Rendering Módszerek

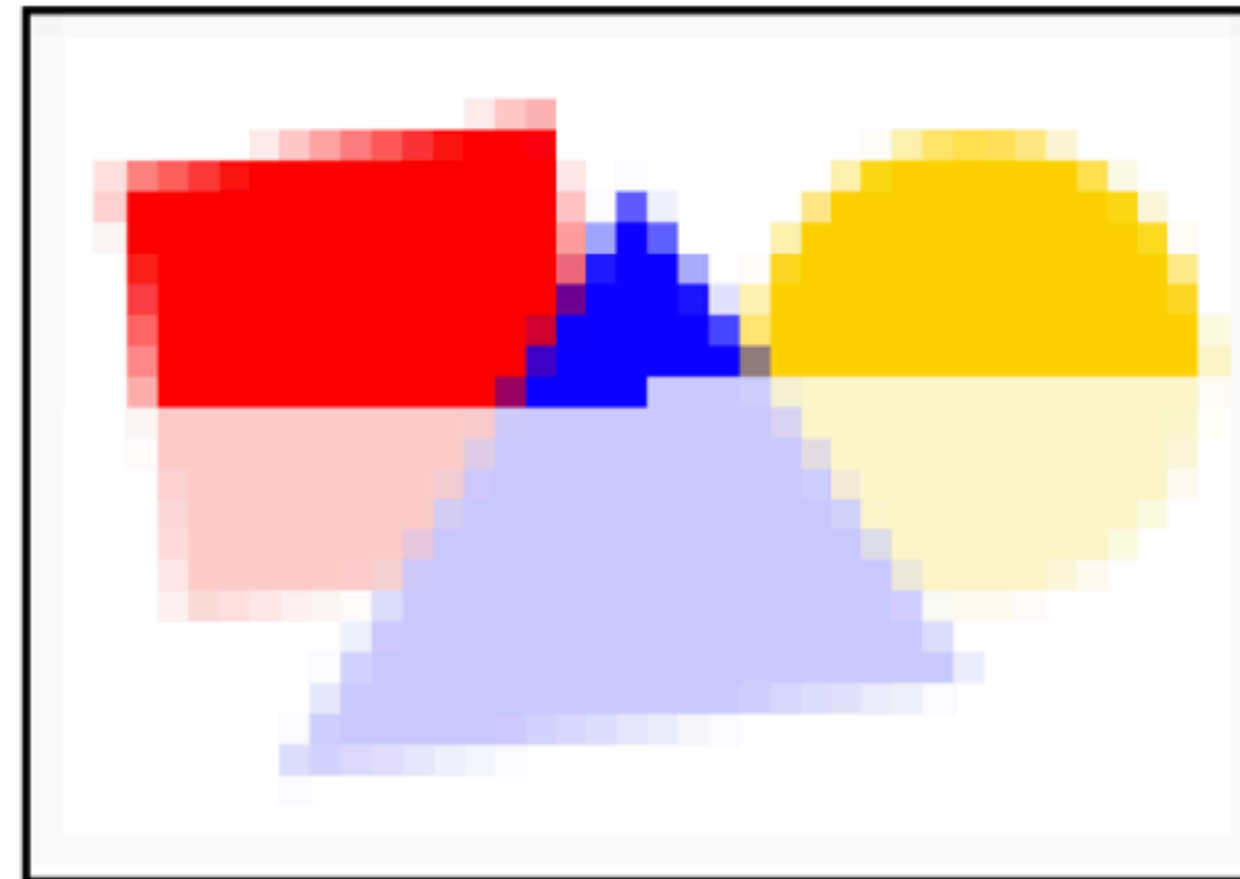
Raszterizáció vs. sugárkövetés

Raszterizáció



```
for each primitive:  
  for each sample:  
    determine coverage  
    evaluate color
```

Sugárkövetés

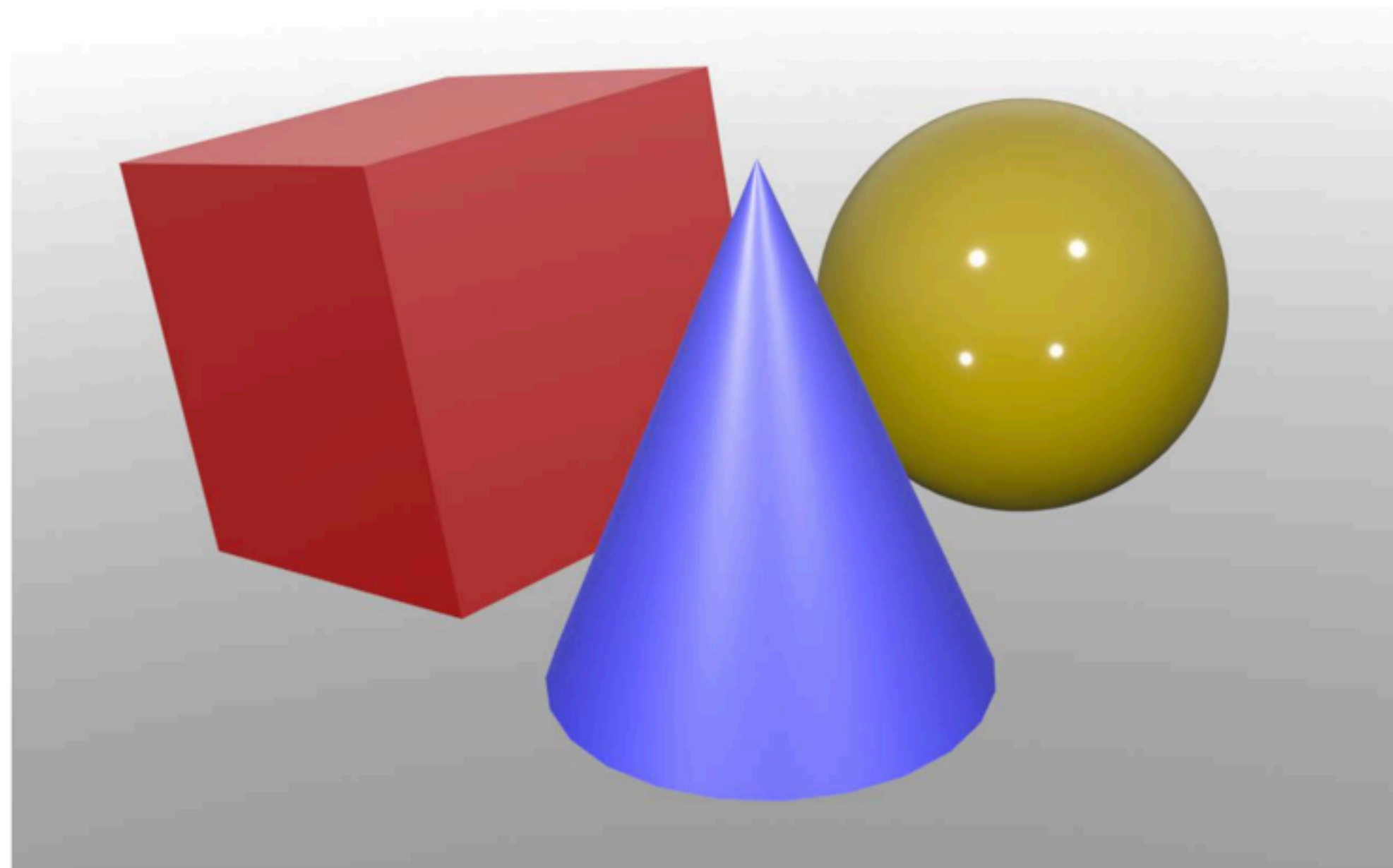


```
for each sample:  
  for each primitive:  
    determine coverage  
    evaluate color
```

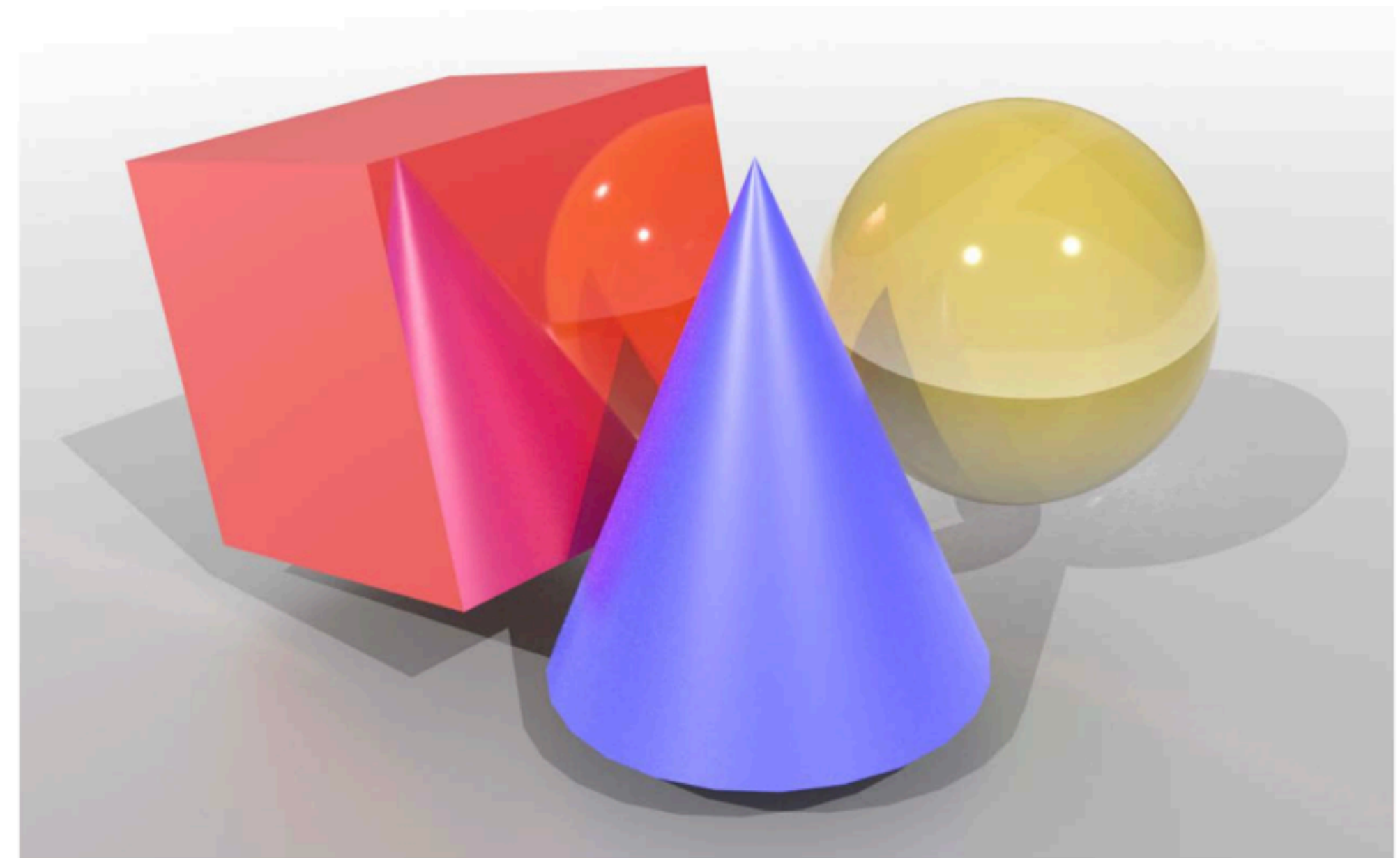
Rendering Módszerek

Raszterizáció vs. sugárkövetés

Raszterizáció

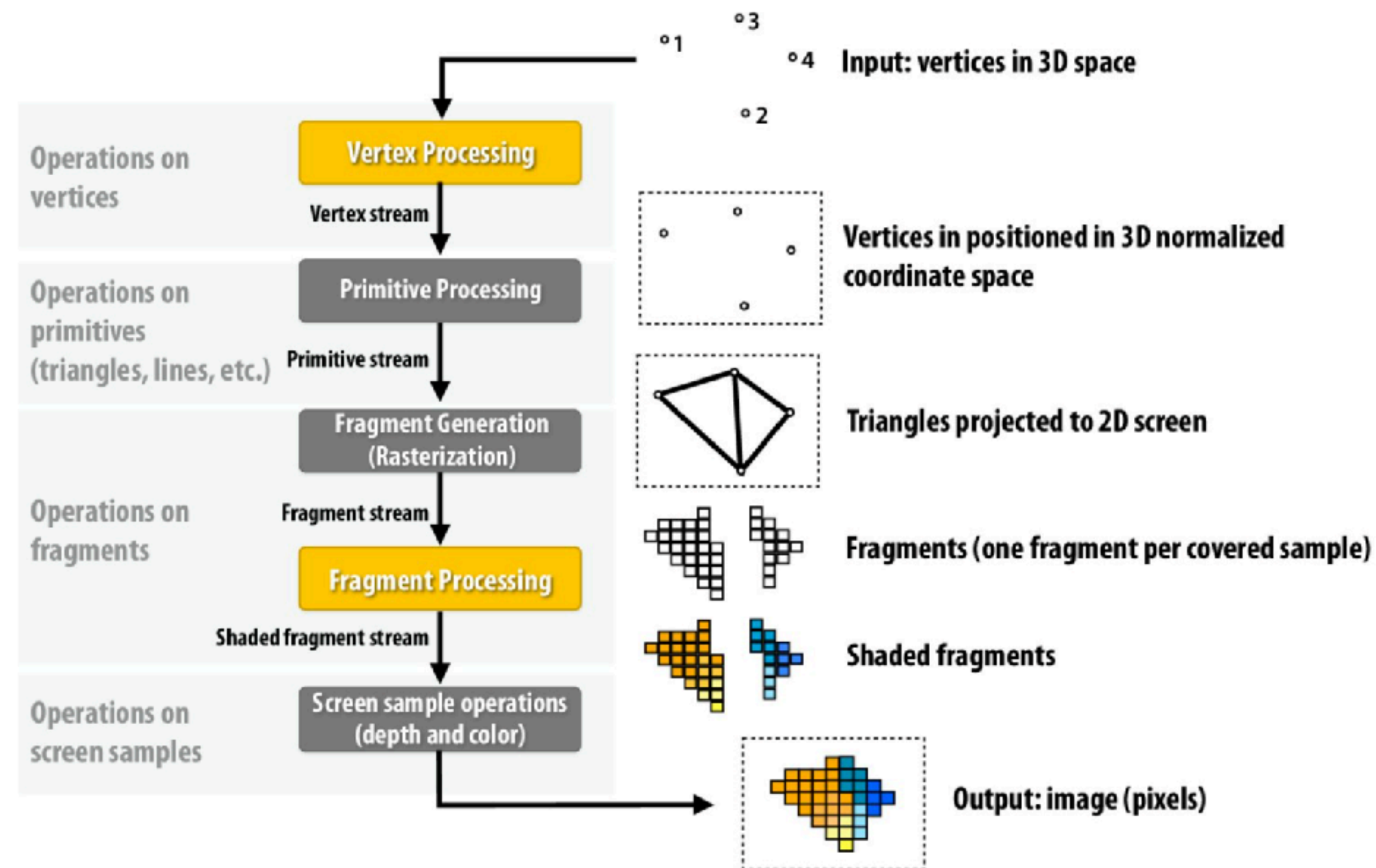


Sugárkövetés



Rendering Módszerek

Raszterizáció – Csővezeték

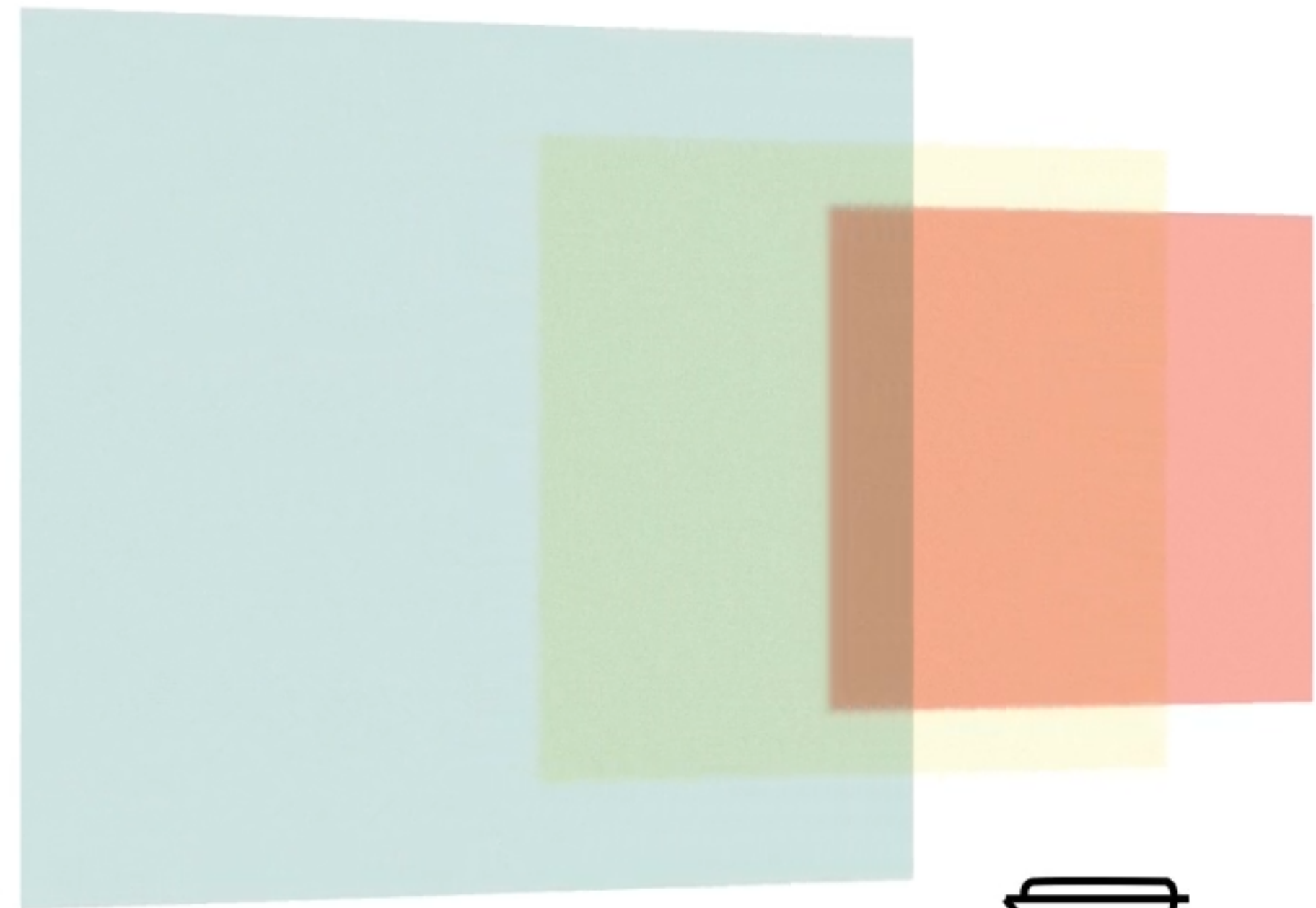


OpenGL/DirectX csővezeték

Rendering Módszerek

Raszterizáció – Alpha blending

$$C = \sum_{i=1}^N \left[c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \right]$$



"Alpha blending"



Rendering Módszerek

Rendering egyenlet

$$L_o(\mathbf{x}, \omega_o, \lambda) = L_e(\mathbf{x}, \omega_o, \lambda) + \int_{\Omega} f(\mathbf{x}, \omega_i, \omega_o, \lambda) L_i(\mathbf{x}, \omega_i, \lambda) (\omega_i \cdot \mathbf{n}) d\omega_i$$

emitted radiance

outgoing radiance in ω_o

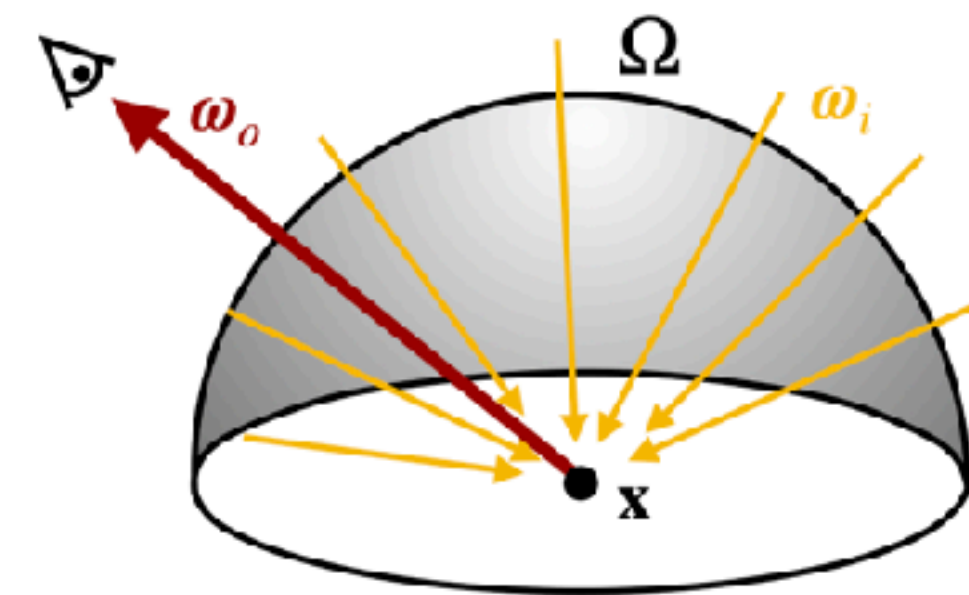
fraction of incoming light that is reflected into ω_o

incoming radiance

angle weight of photon density

solid angle differential

Conservation of energy: $\int_{\omega_i \in \Omega} f(\omega_i, \omega_o) (\omega_i \cdot \mathbf{n}) d\omega_i \leq 1$



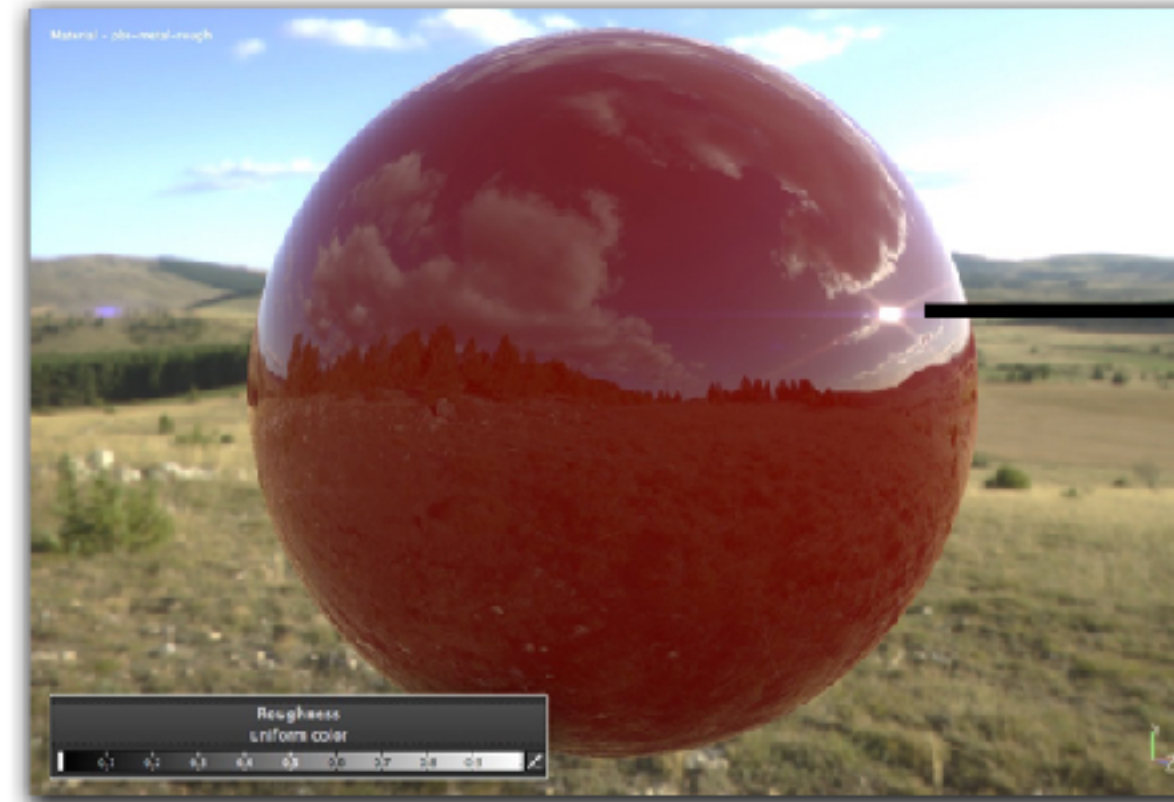
Rekurzív integrálegyenlet — kiértékelés Monte Carlo integrálással (mintavételezéssel)

Rendering Módszerek

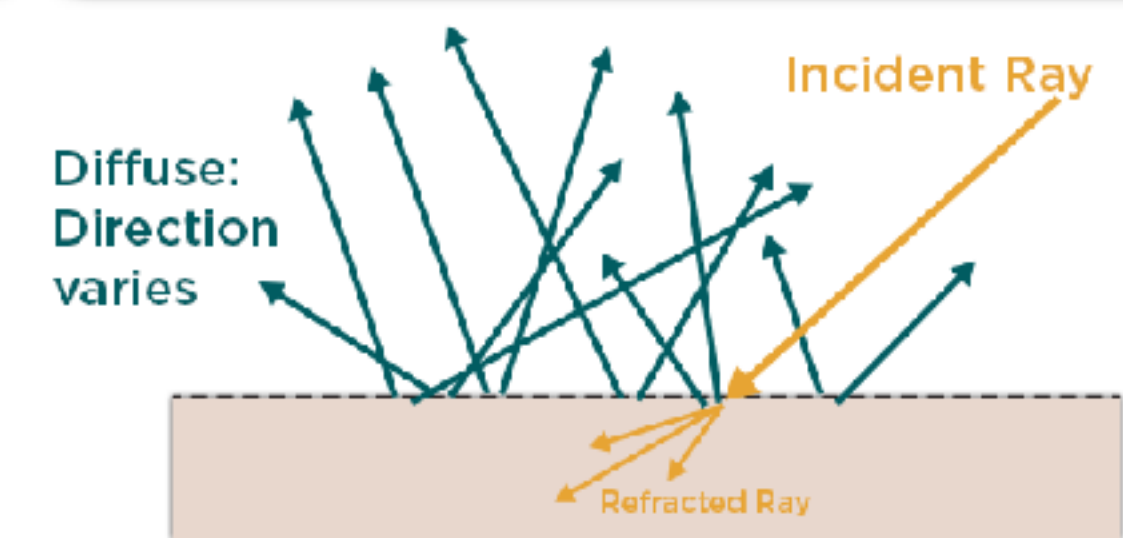
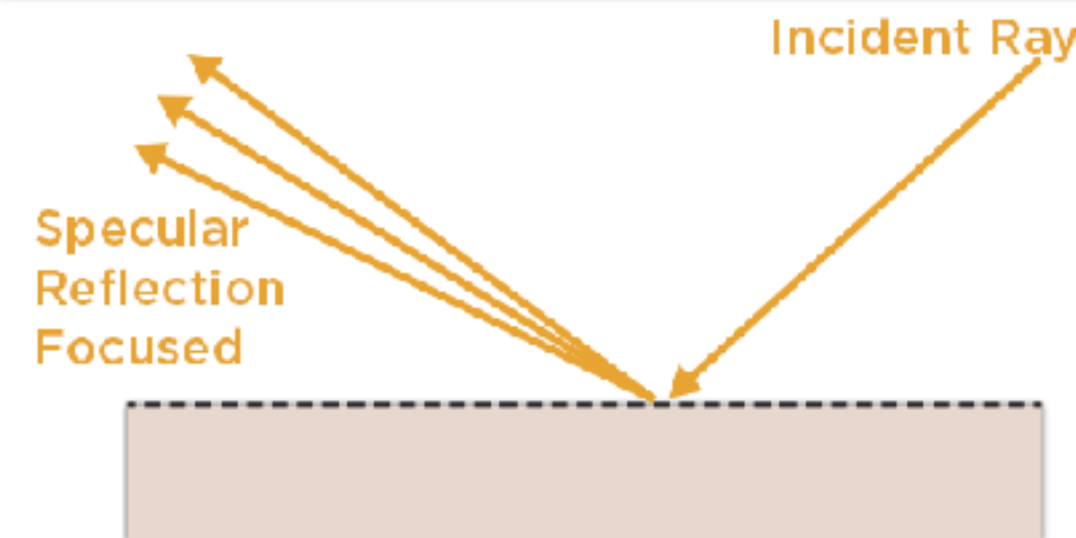
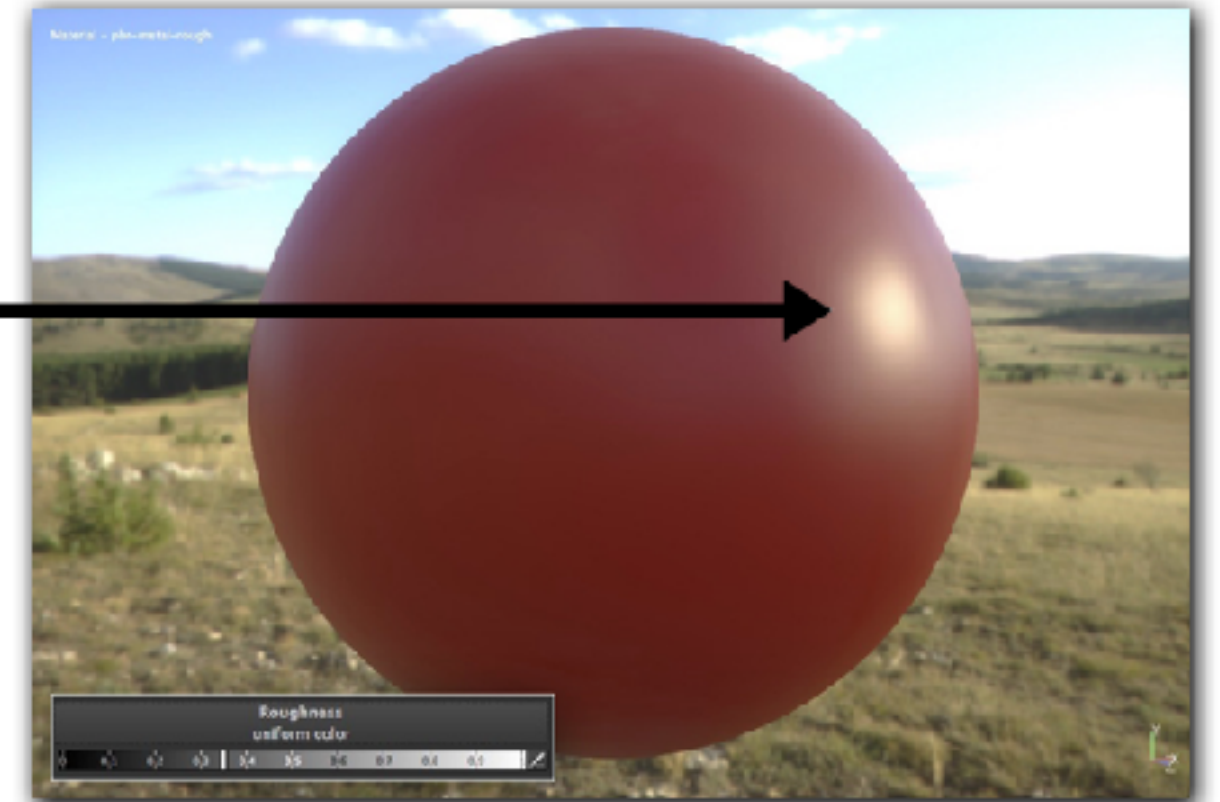
Diffúz vs. spekuláris visszaverődés

- **Spekuláris** visszaverődés: adott irányokba erősebb (sima, fényes felület)
- **Diffúz** visszaverődés: kb. minden irányba hasonló mértékben (durva, matt felület)

Smoother surface focus specular reflection



Rougher surface larger dimmer highlight

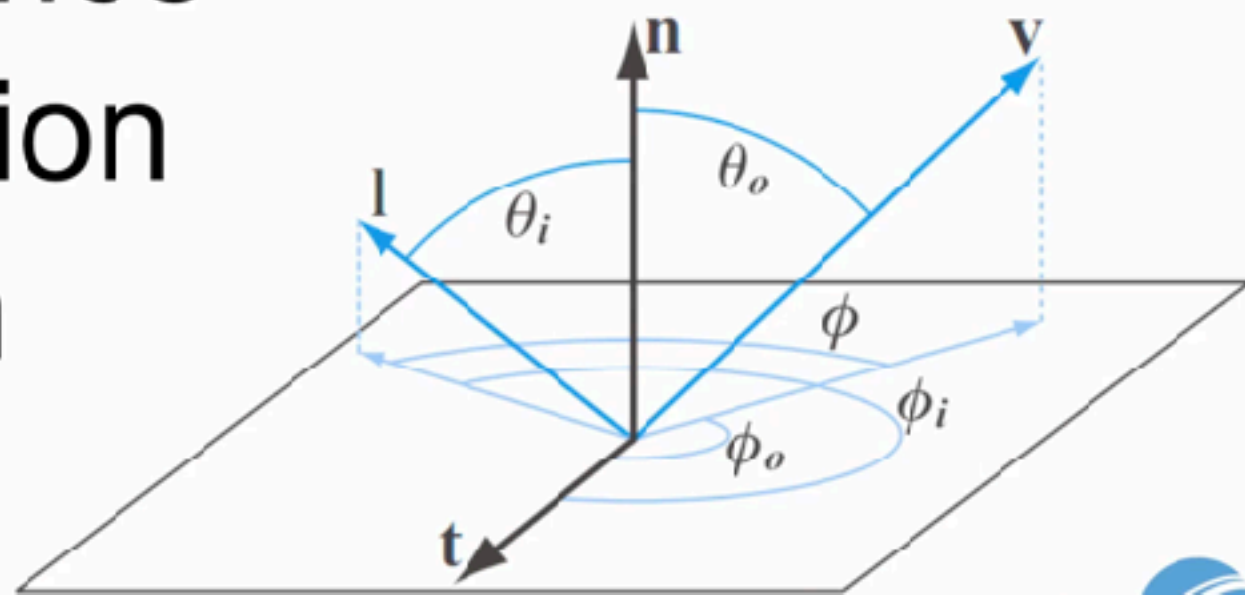


Rendering Módszerek

BRDF

Bidirectional
Reflectance
Distribution
Function

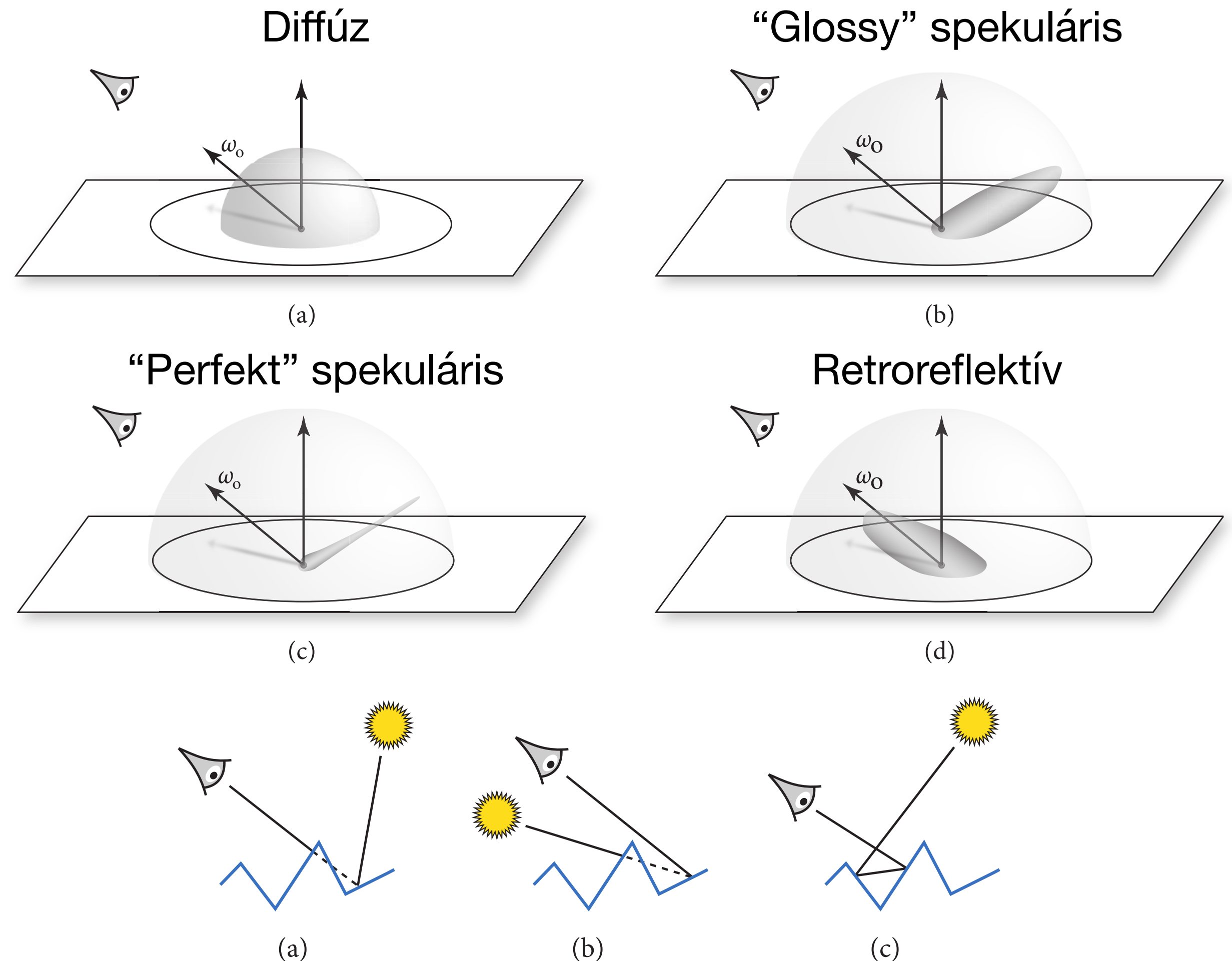
$$f(\mathbf{l}, \mathbf{v})$$



“Microfacet” modell (Cook-Torrance):

Normálvektor disztribúció Fresnel effektus Geometria/Láthatóság

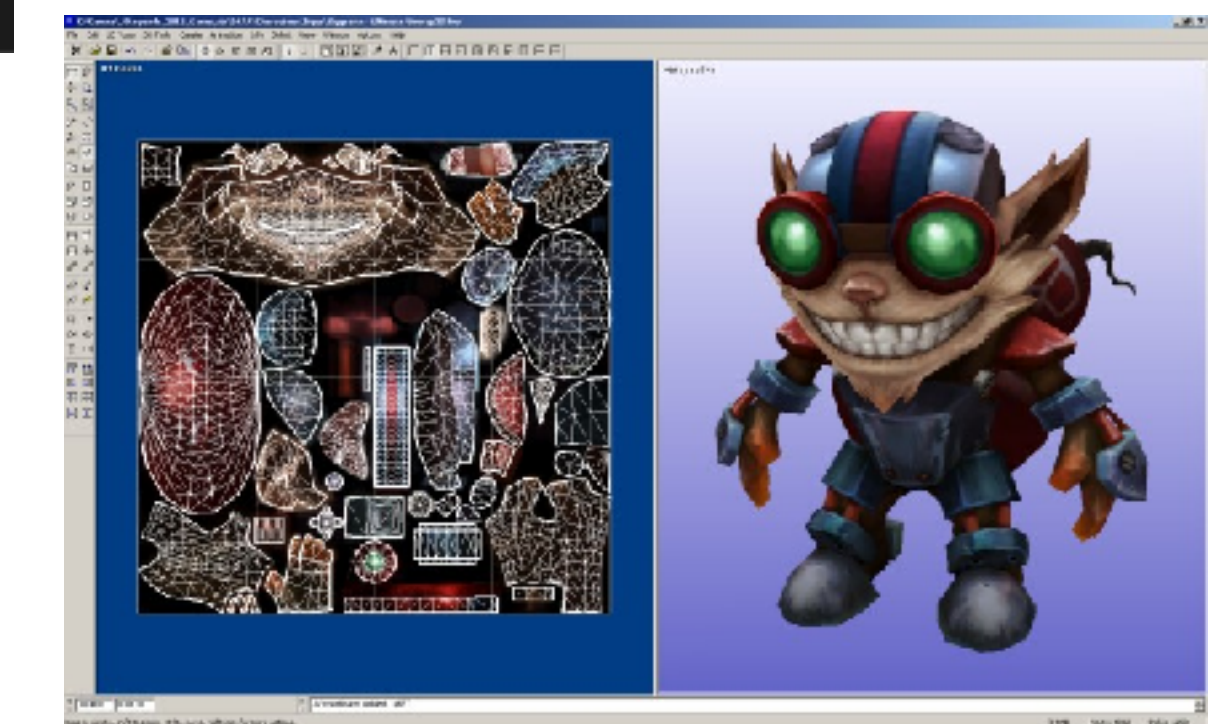
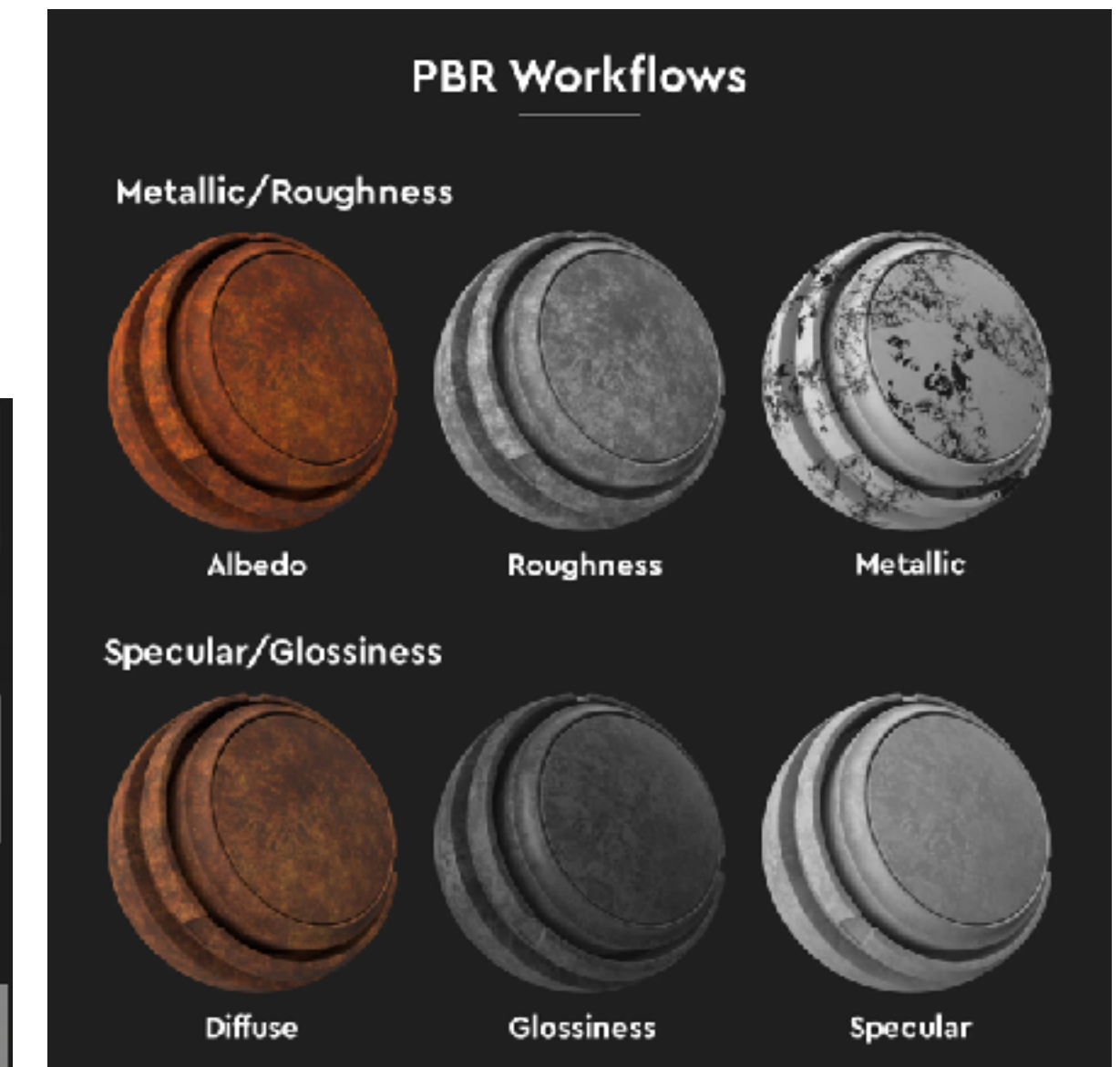
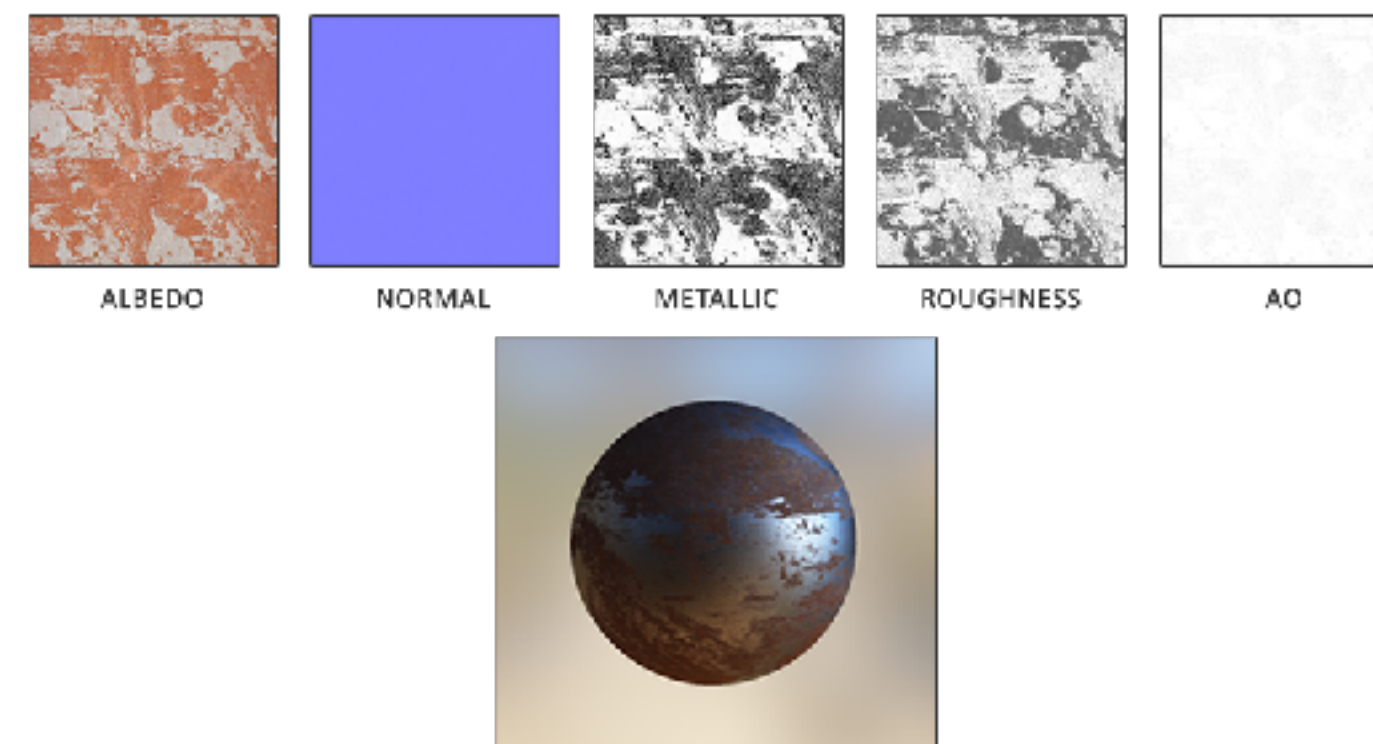
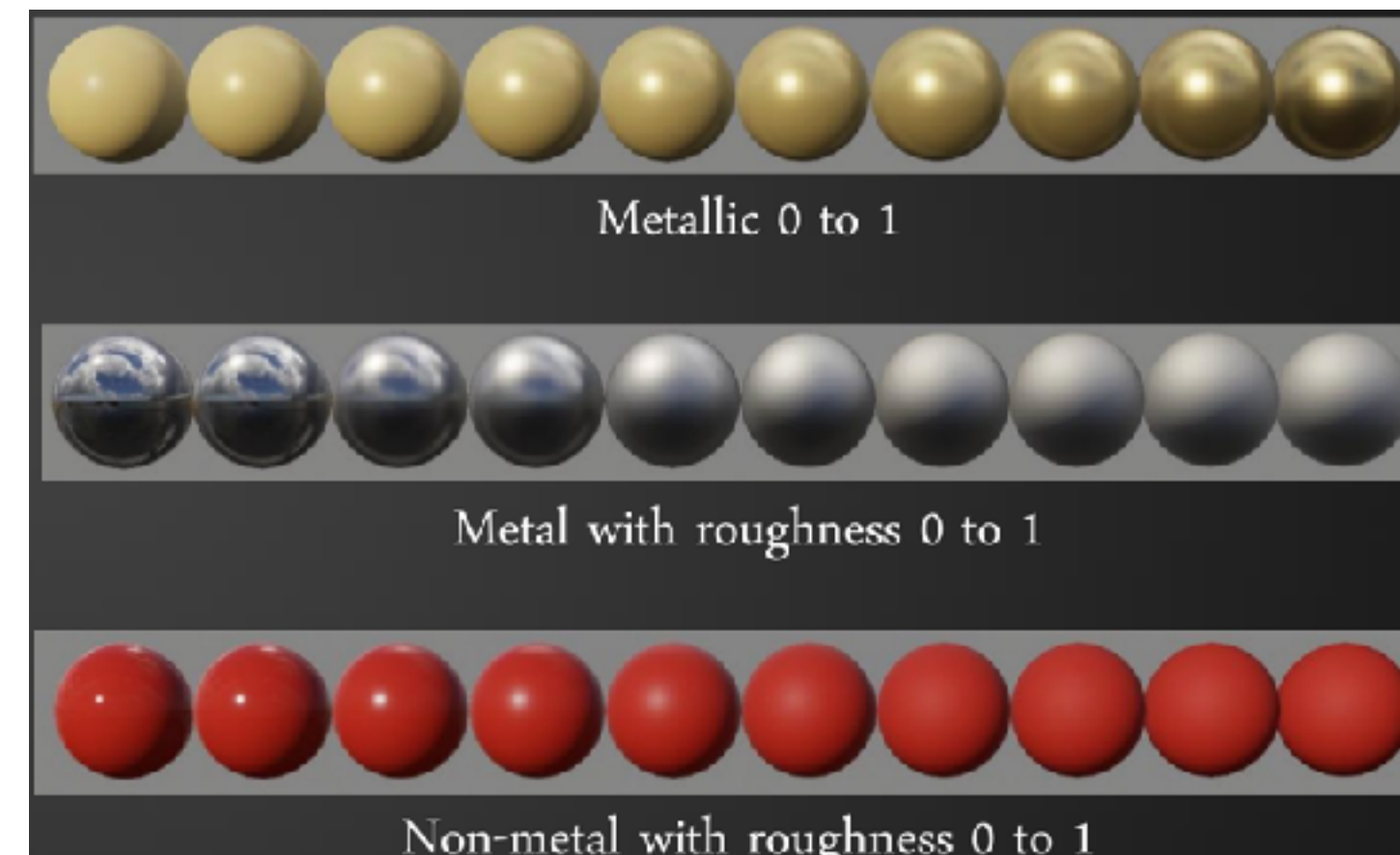
$$f(\mathbf{l}, \mathbf{v}) = \frac{D(\mathbf{h}) F(\mathbf{v}, \mathbf{h}) G(\mathbf{l}, \mathbf{v}, \mathbf{h})}{4 (\mathbf{n} \cdot \mathbf{l}) (\mathbf{n} \cdot \mathbf{v})}$$



Rendering Módszerek

Textúrák, materialok

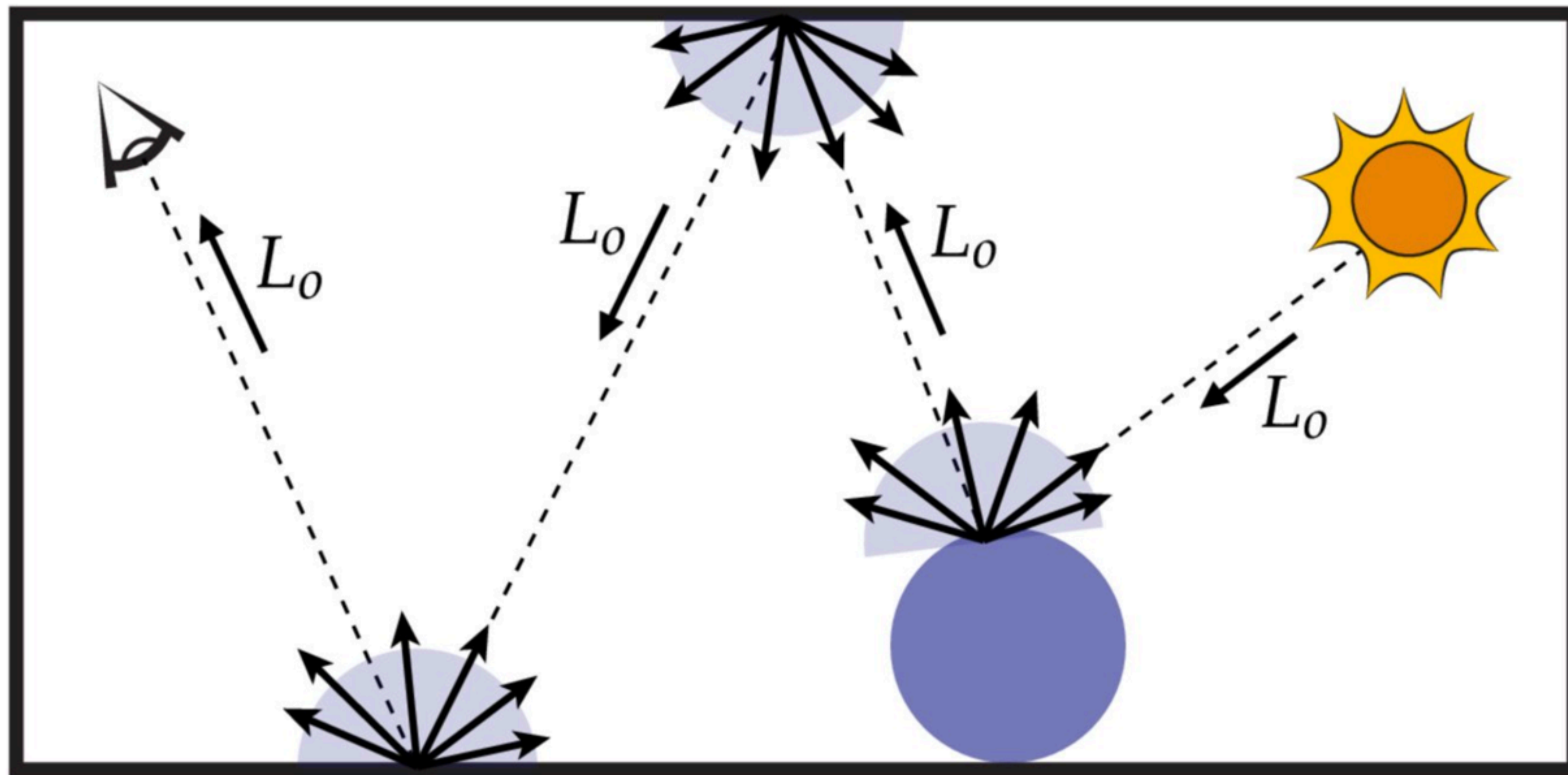
- A gyakorlatban az anyagjellemzőket (**materialokat**) *textúrák* formájában tároljuk
- **Physically-Based Rendering (PBR)** material — szemléletes paraméterek, fizikailag “korrekt” árnyaláshoz:
 - Albedo / Diffuse
 - Roughness / Glossiness
 - Metallic / Specular
- További attribútumok, pl.:
 - Normál map
 - Displacement map
 - Ambient occlusion



Síkba terítés
(paraméterezés /UV mapping)

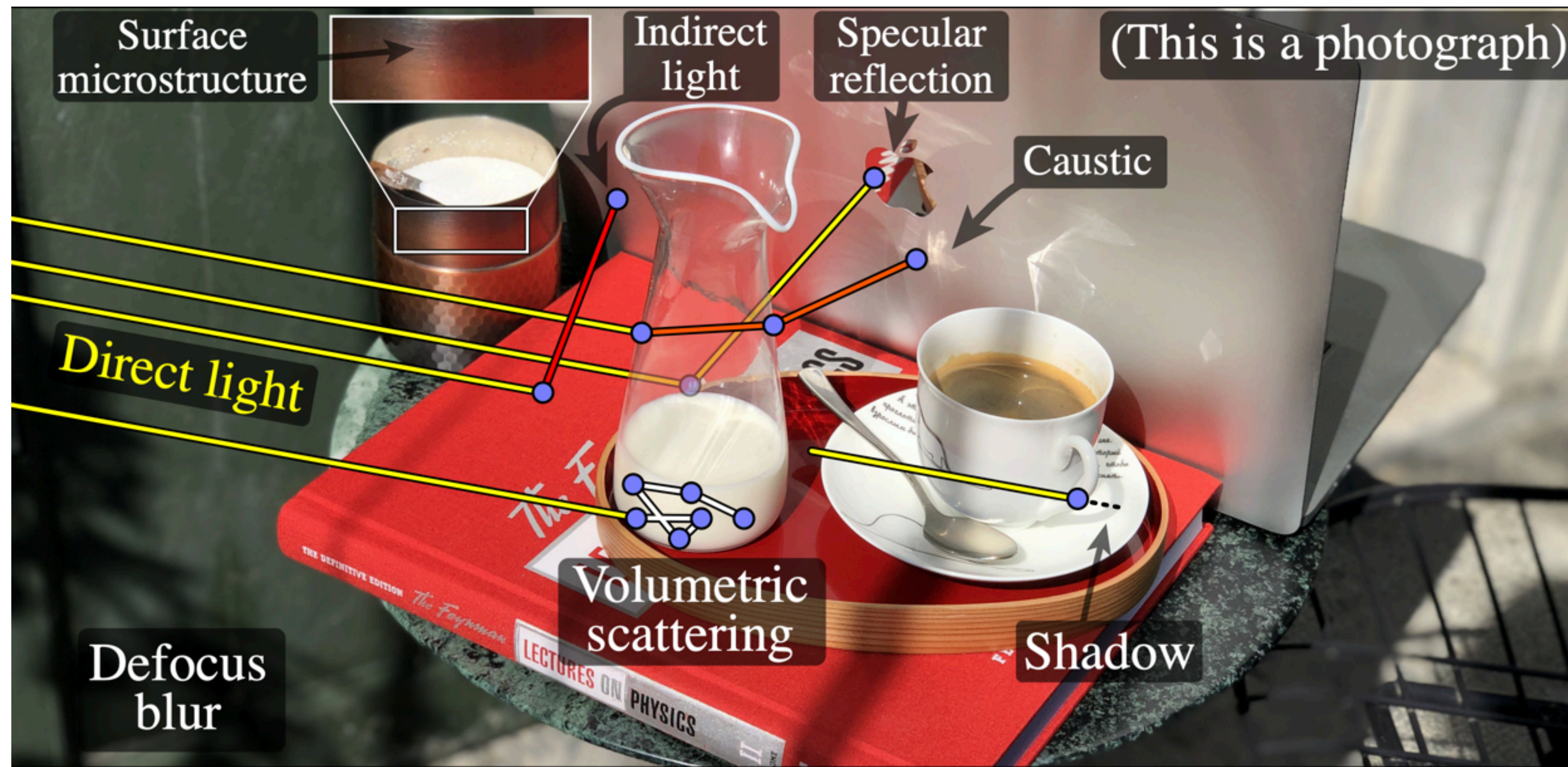
Rendering Módszerek

Sugárkövetés



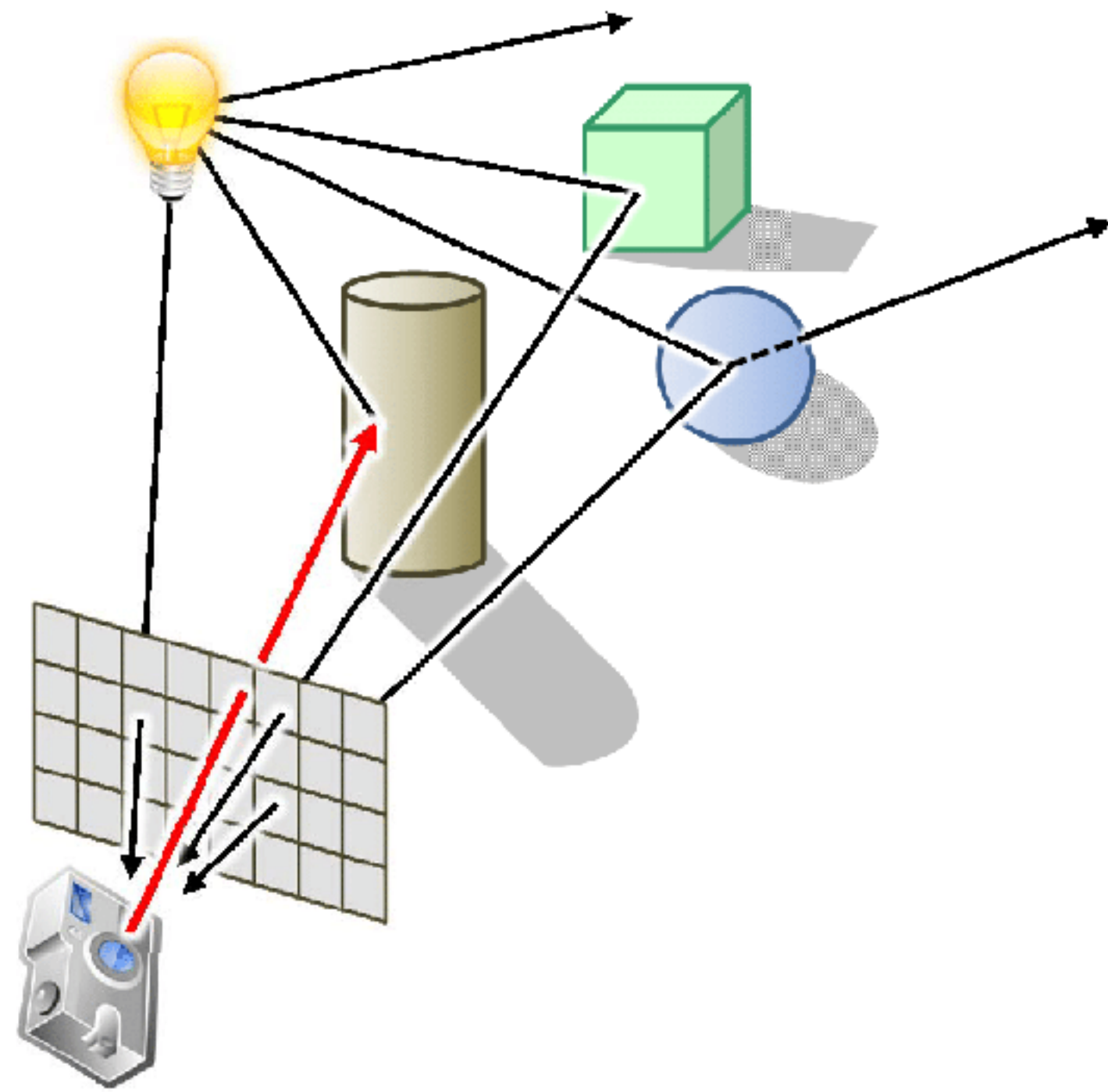
Rendering Módszerek

Sugárkövetés

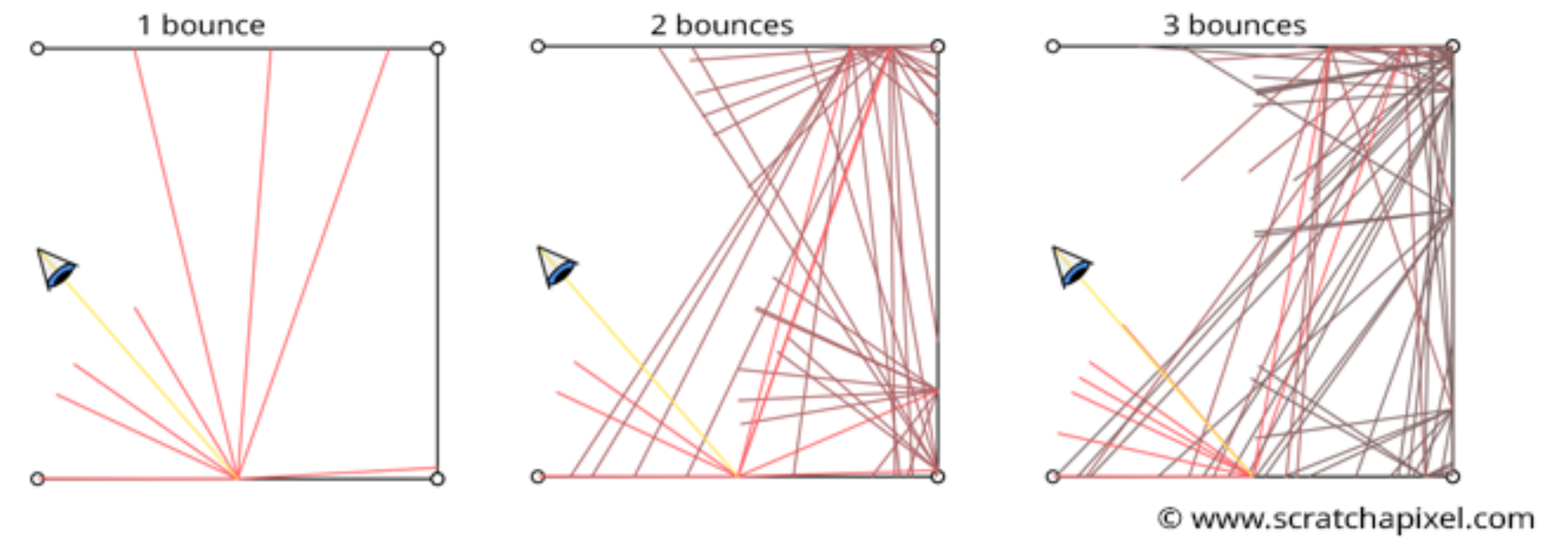


Rendering Módszerek

Sugárkövetés



Random (“Monte Carlo”)
mintavételezés (integrálás)



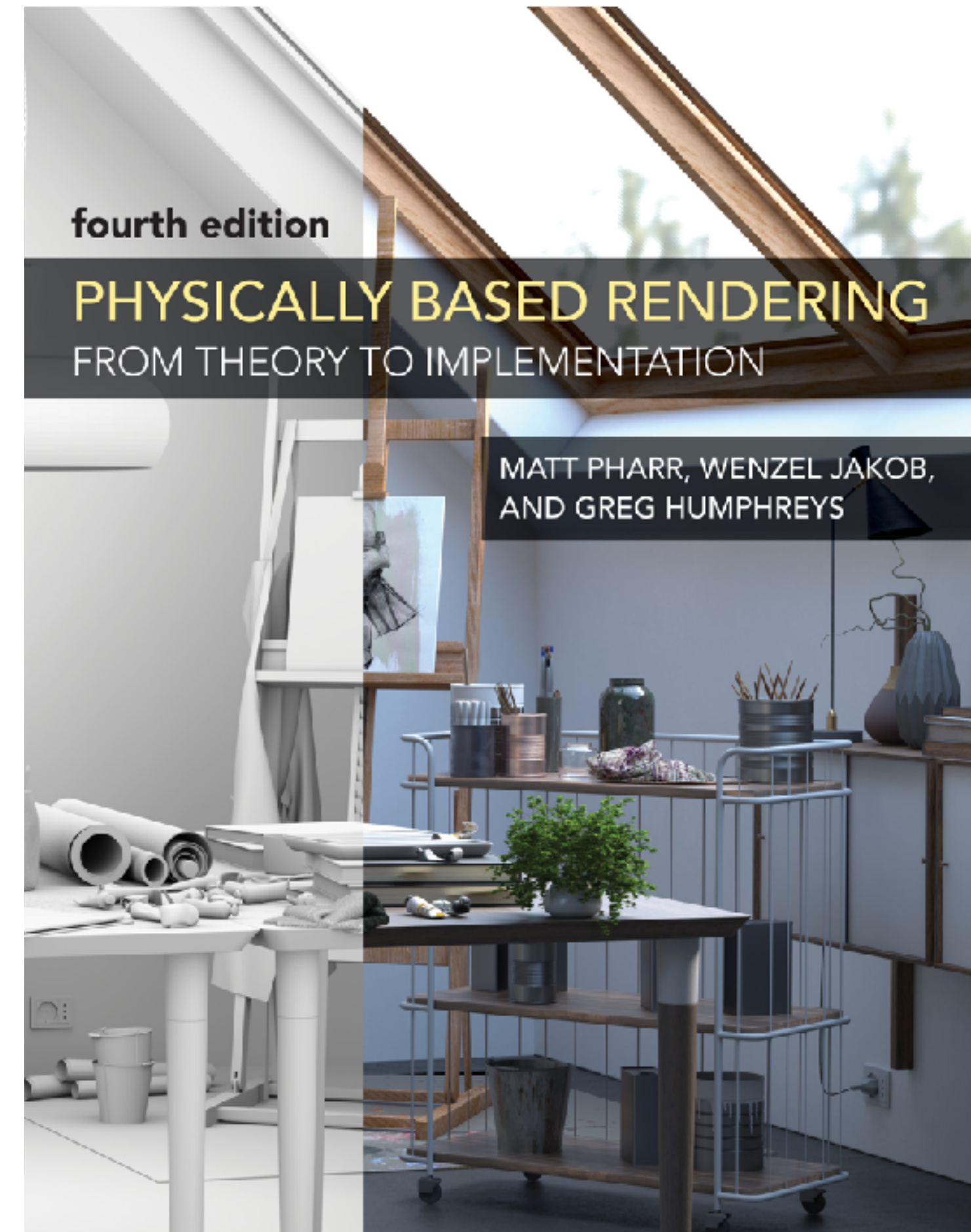
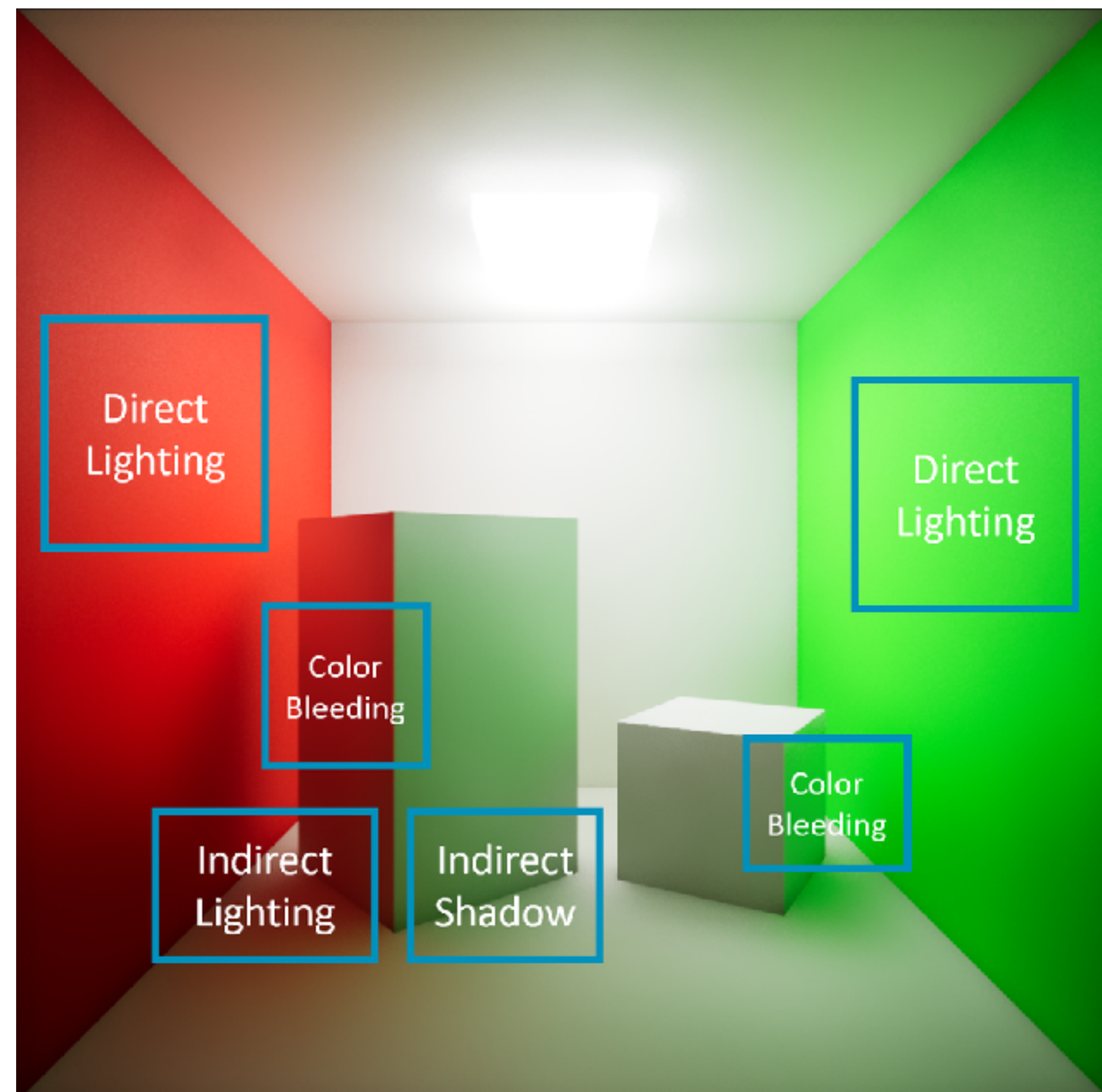
8 minta / pixel



8192 minta / pixel

Rendering Módszerek

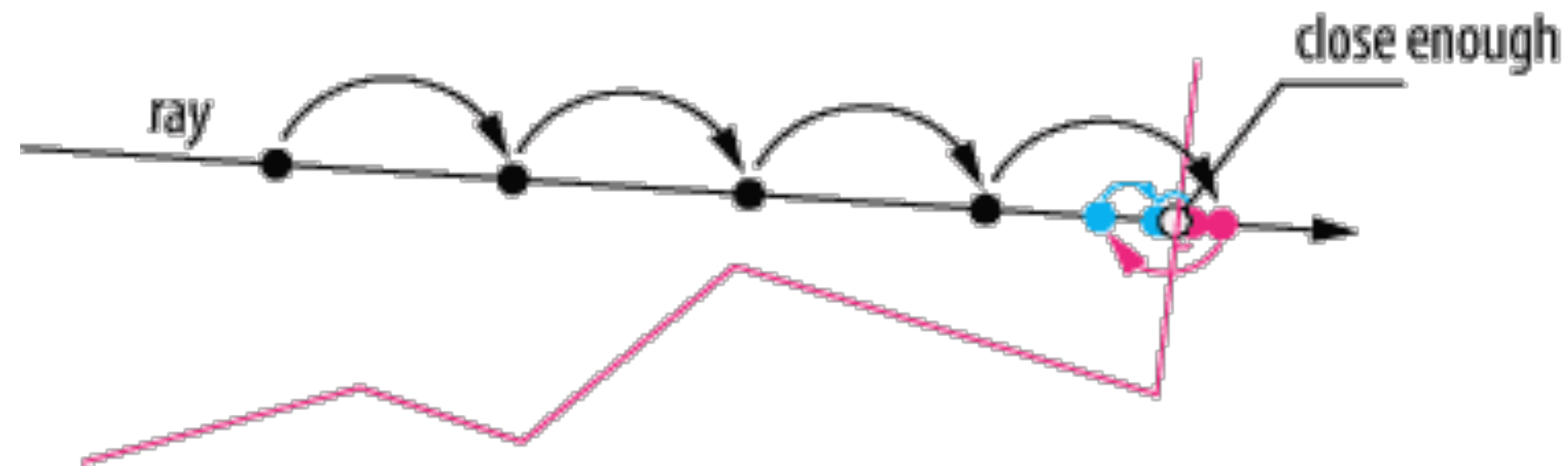
Sugárkövetés



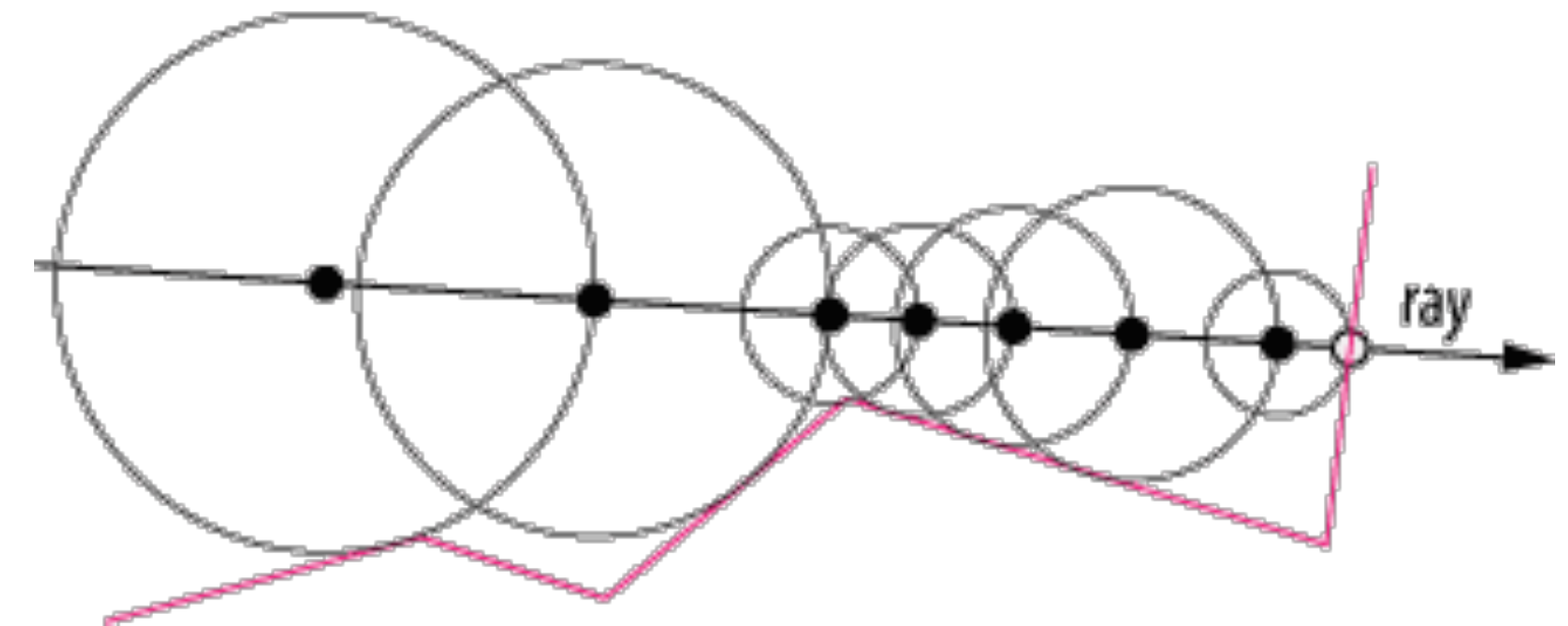
<https://www.pbrt.org/>

Rendering Módszerek

Ray Marching / Sphere Tracing



Ray Marching

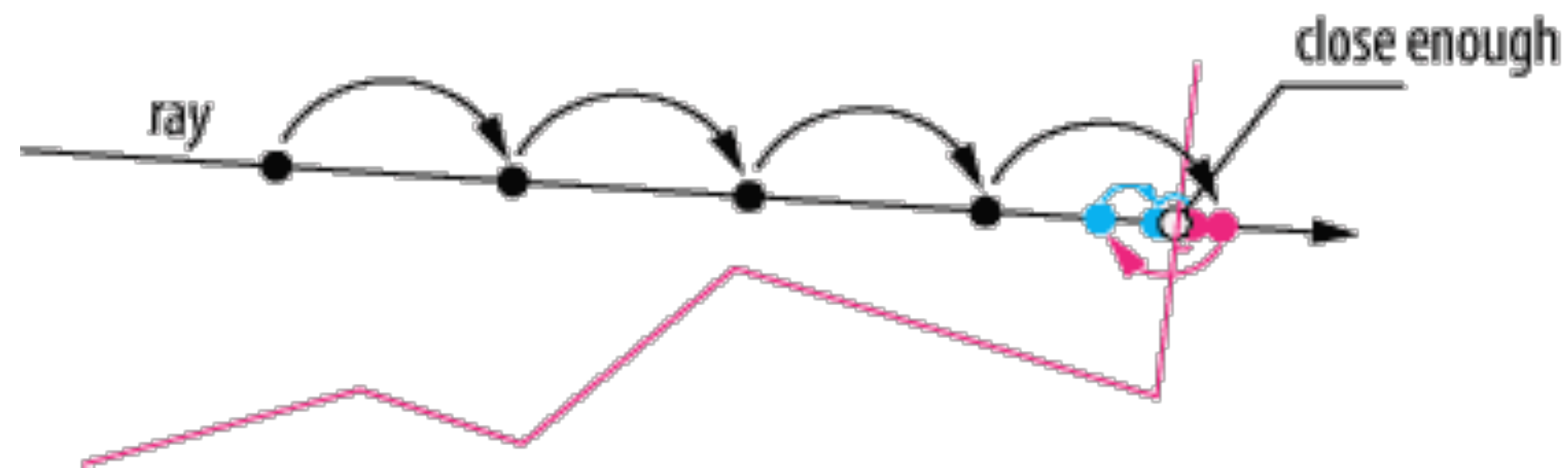


Sphere Tracing

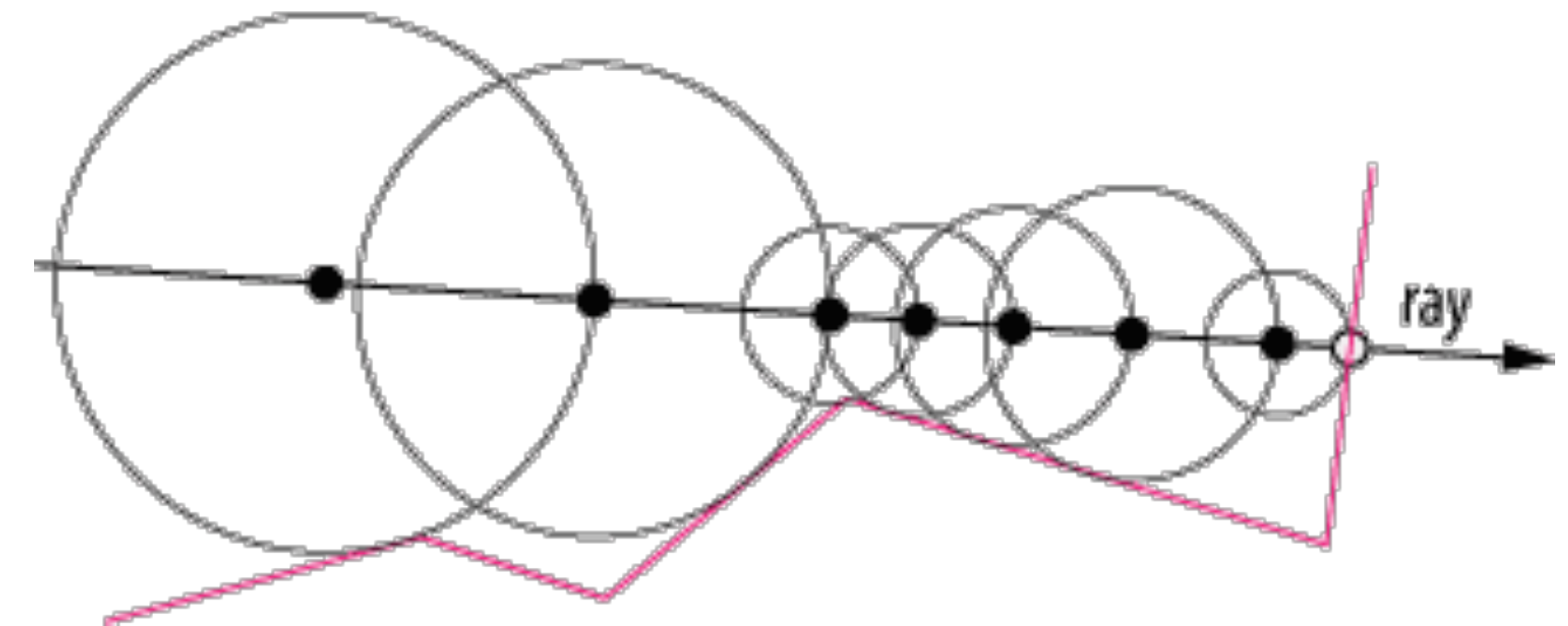
(feltételezi, hogy a derivált korlátos, pl. SDF)

Rendering Módszerek

Ray Marching / Sphere Tracing



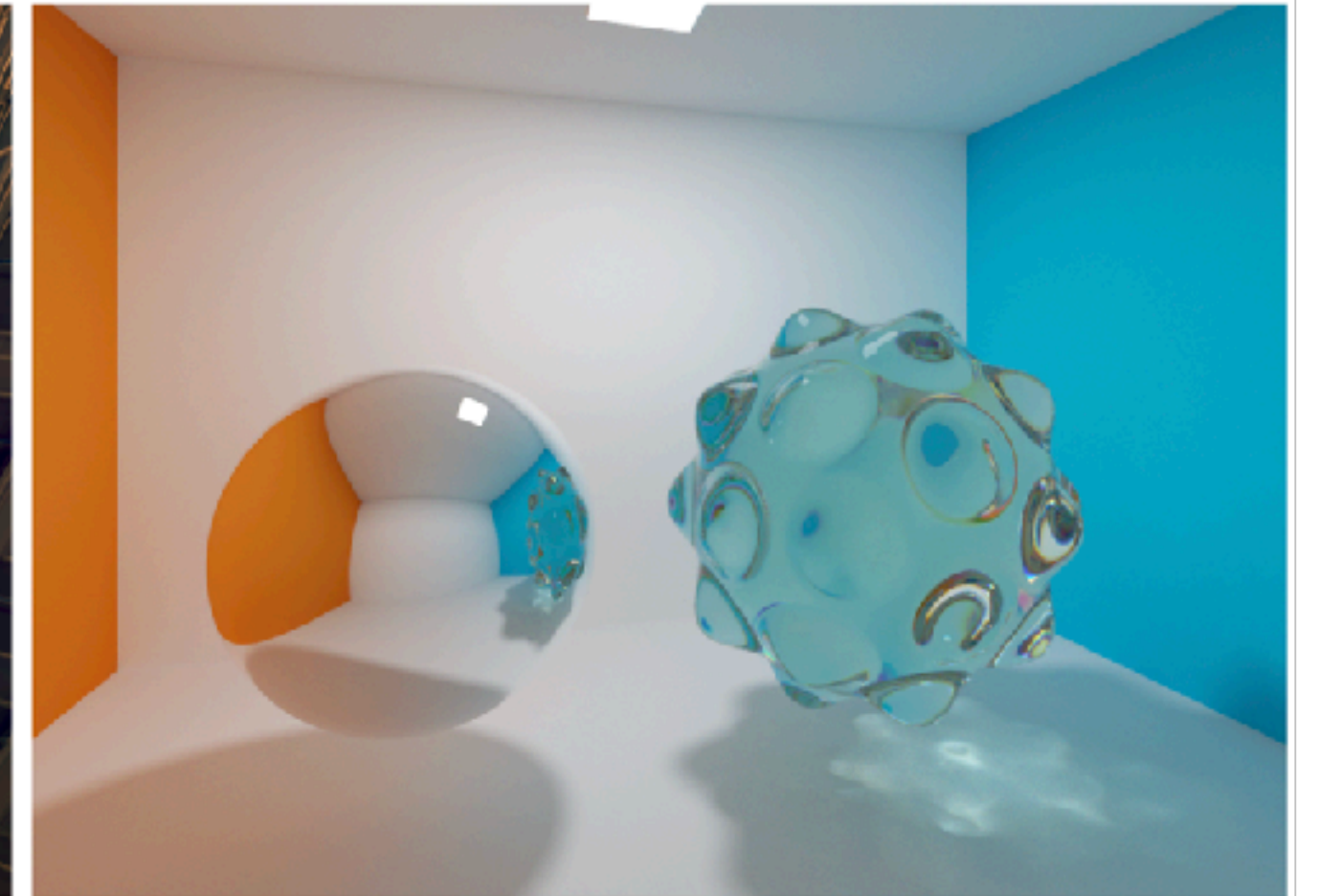
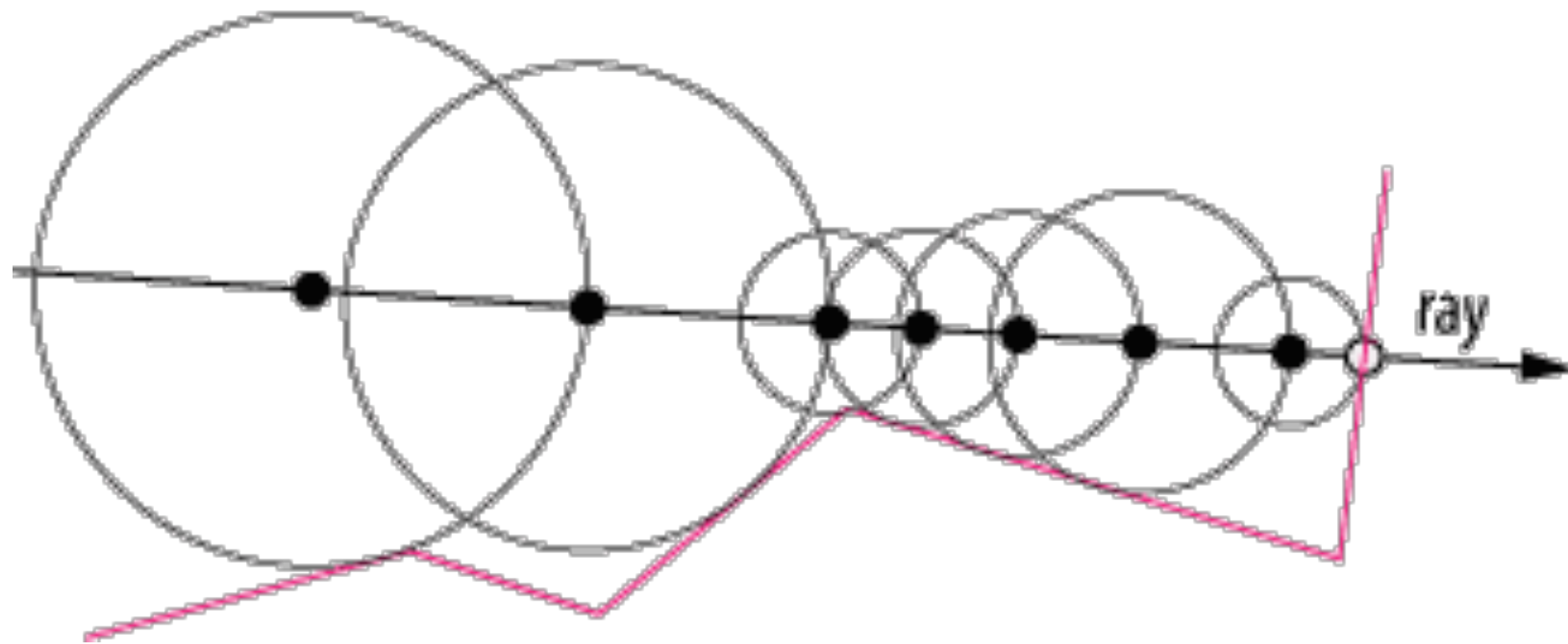
Ray Marching



Sphere Tracing
(feltételezi, hogy a derivált korlátos, pl. SDF)

Rendering Módszerek

Sphere Tracing



Rendering Módszerek

Térfogati renderelés



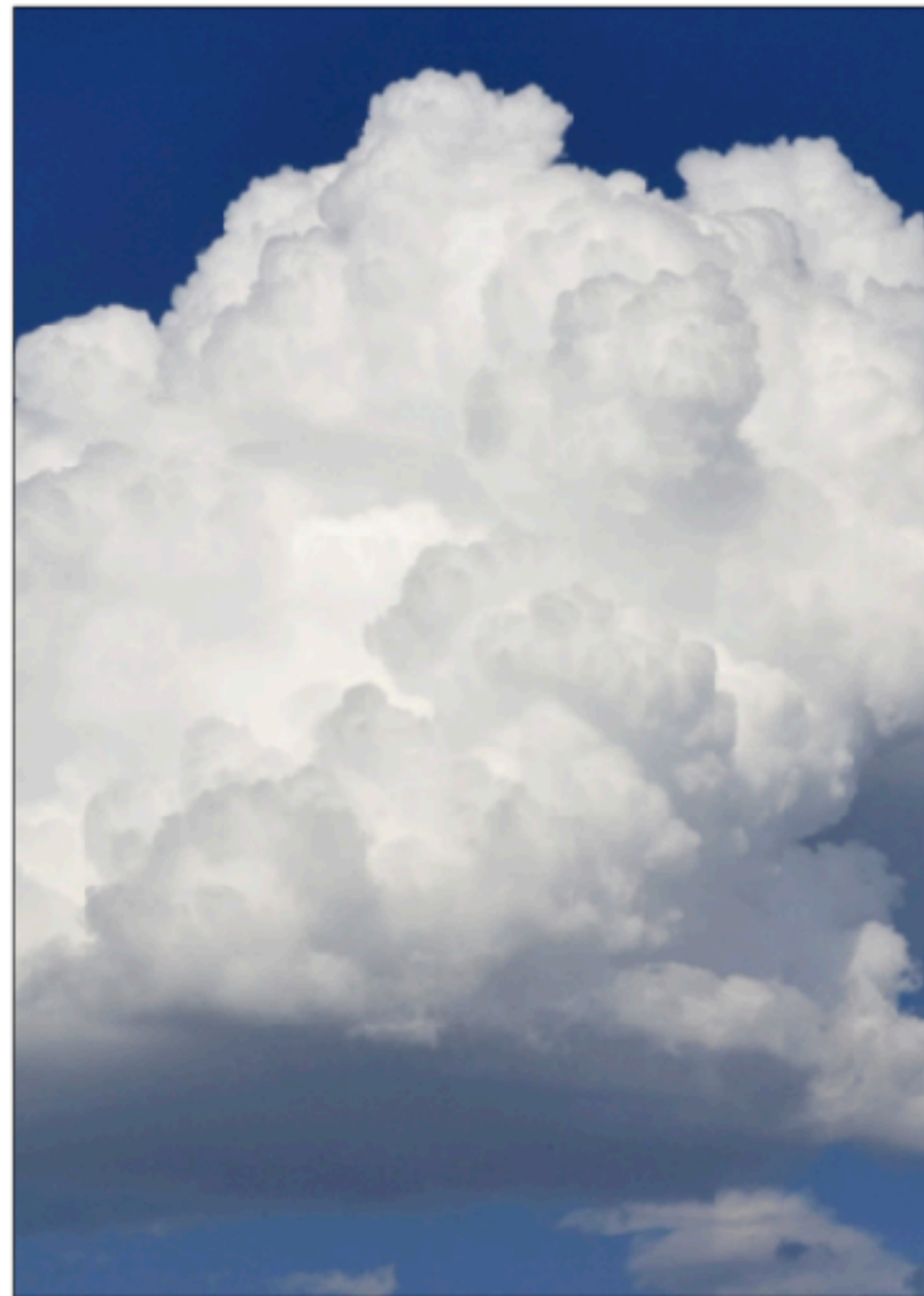
Absorption



<http://commons.wikimedia.org>



Scattering



Emission



<http://iainmaria.com>



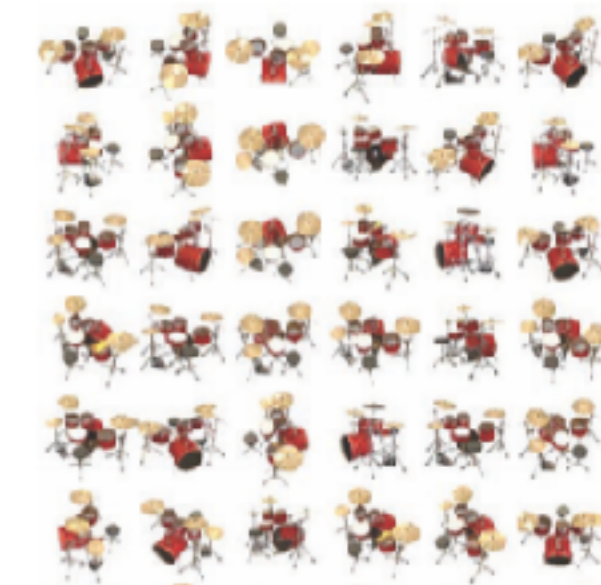
Inverz Grafika

A Kulcs: Differenciálás

Inverz Grafika

A Kulcs: Differenciálás

Eredeti képek



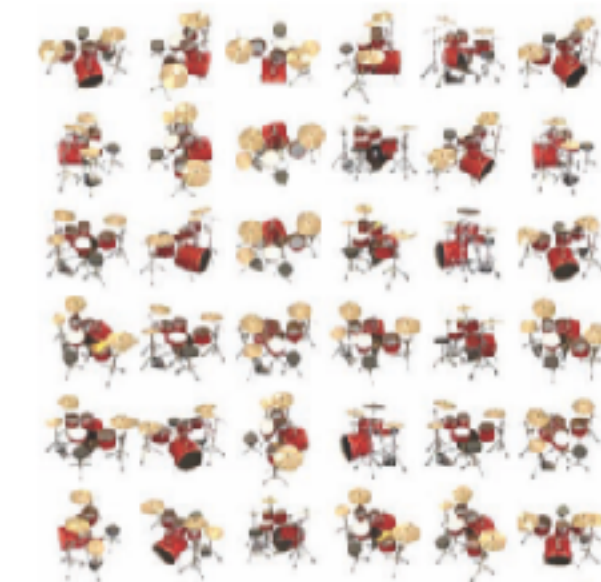
Inverz Grafika

A Kulcs: Differenciálás

3D Modell

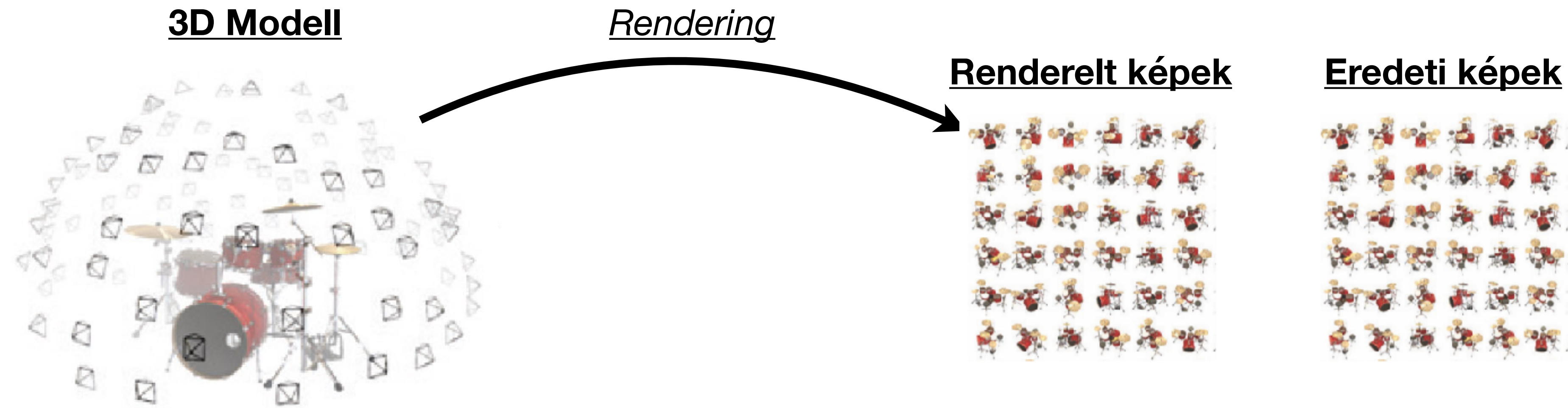


Eredeti képek



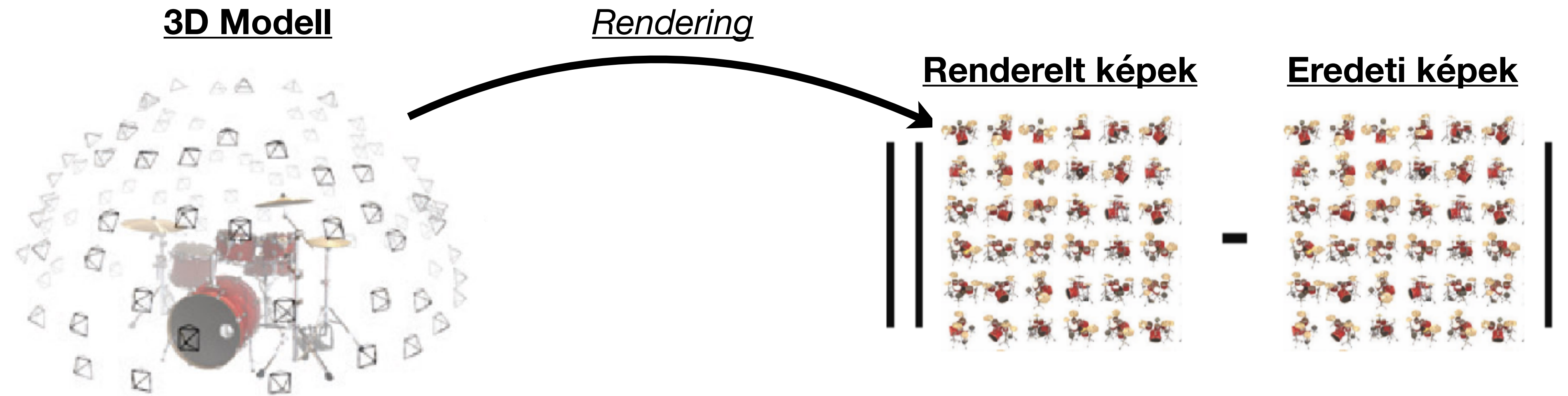
Inverz Grafika

A Kulcs: Differenciálás



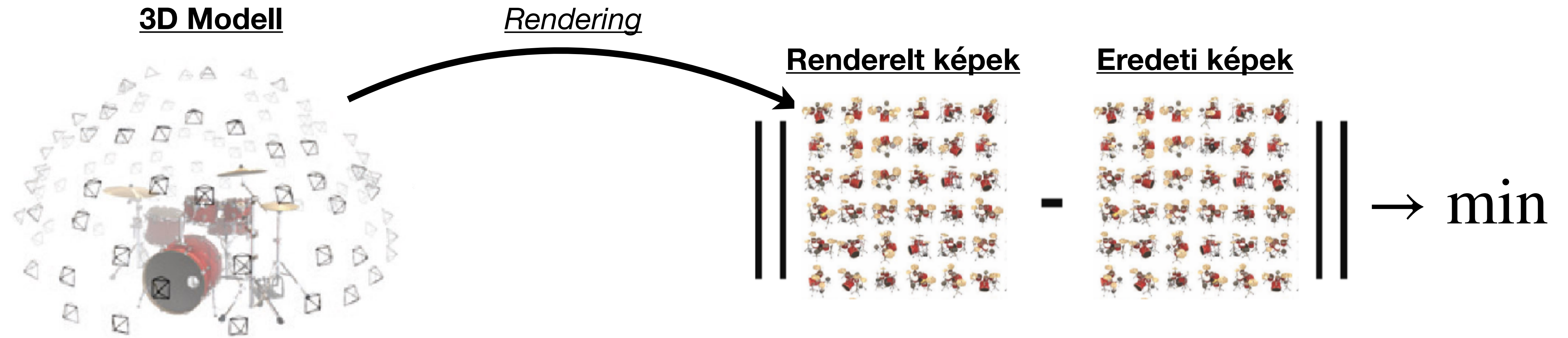
Inverz Grafika

A Kulcs: Differenciálás



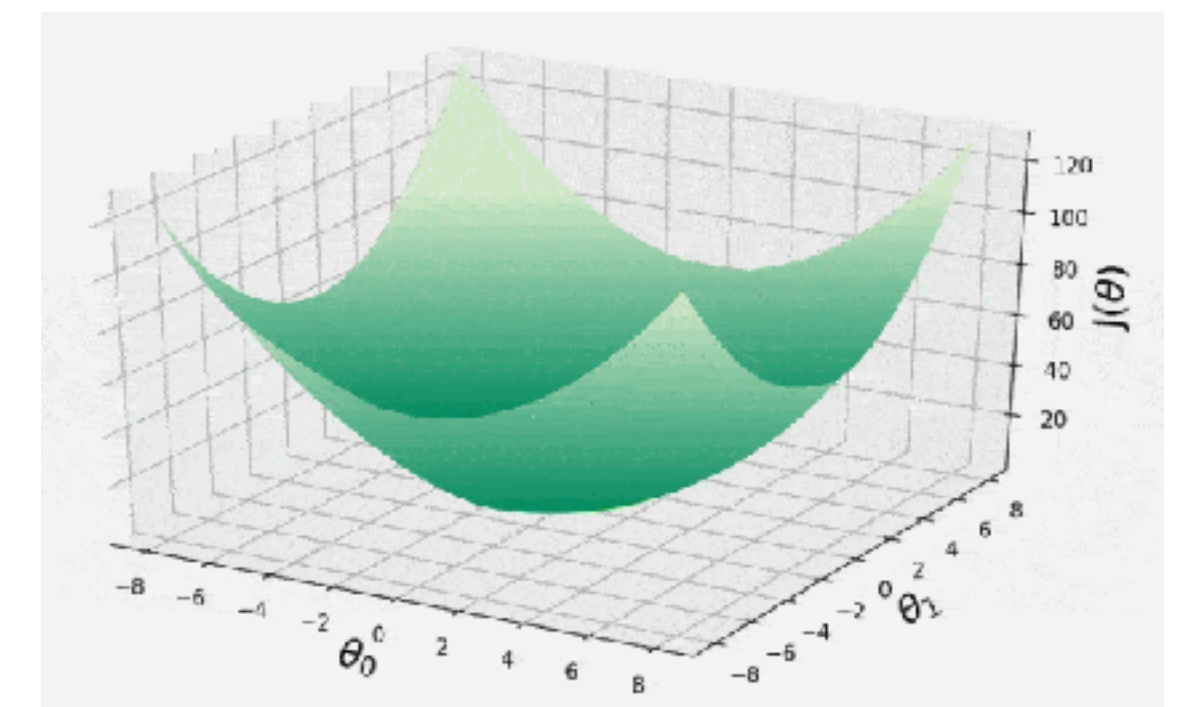
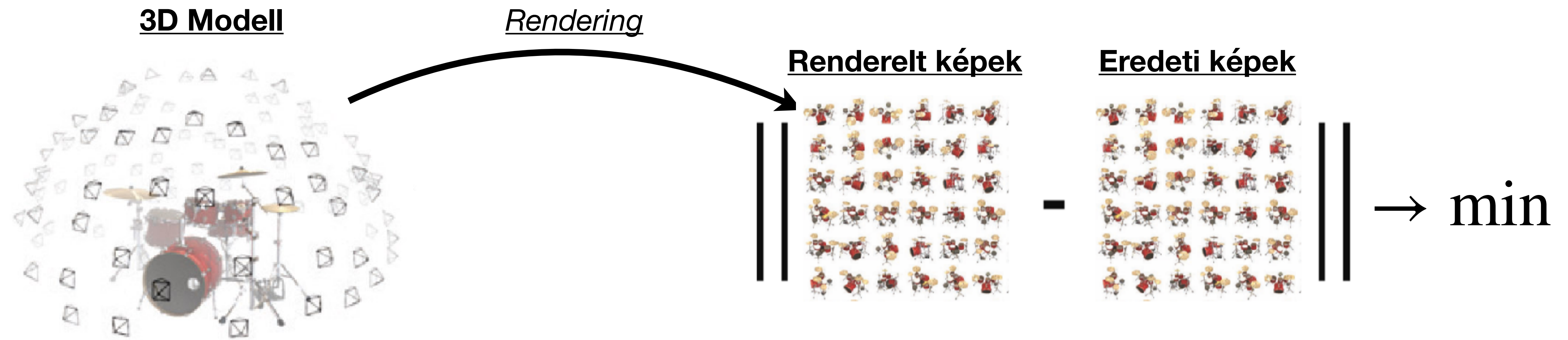
Inverz Grafika

A Kulcs: Differenciálás



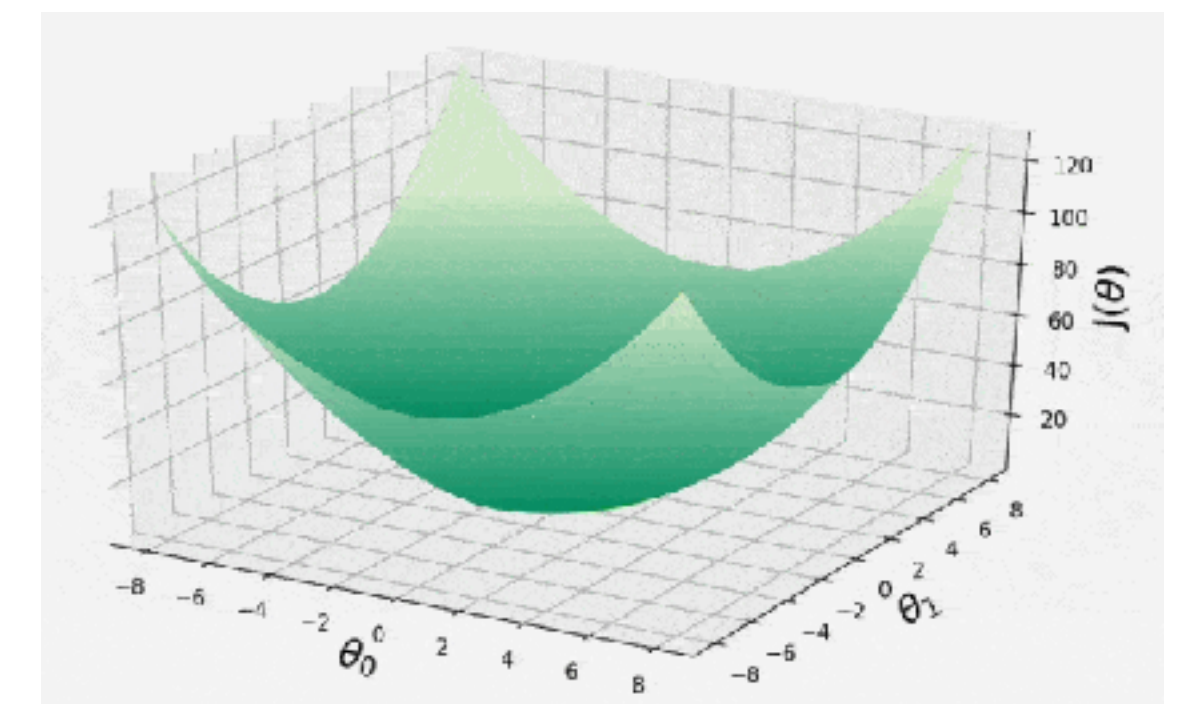
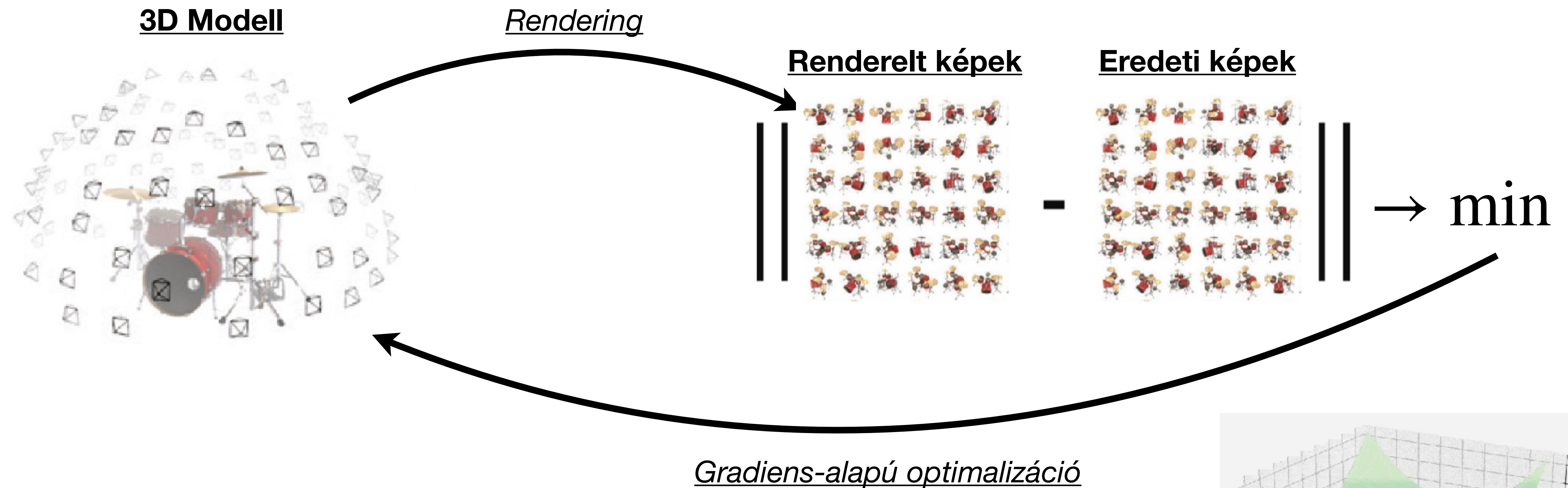
Inverz Grafika

A Kulcs: Differenciálás



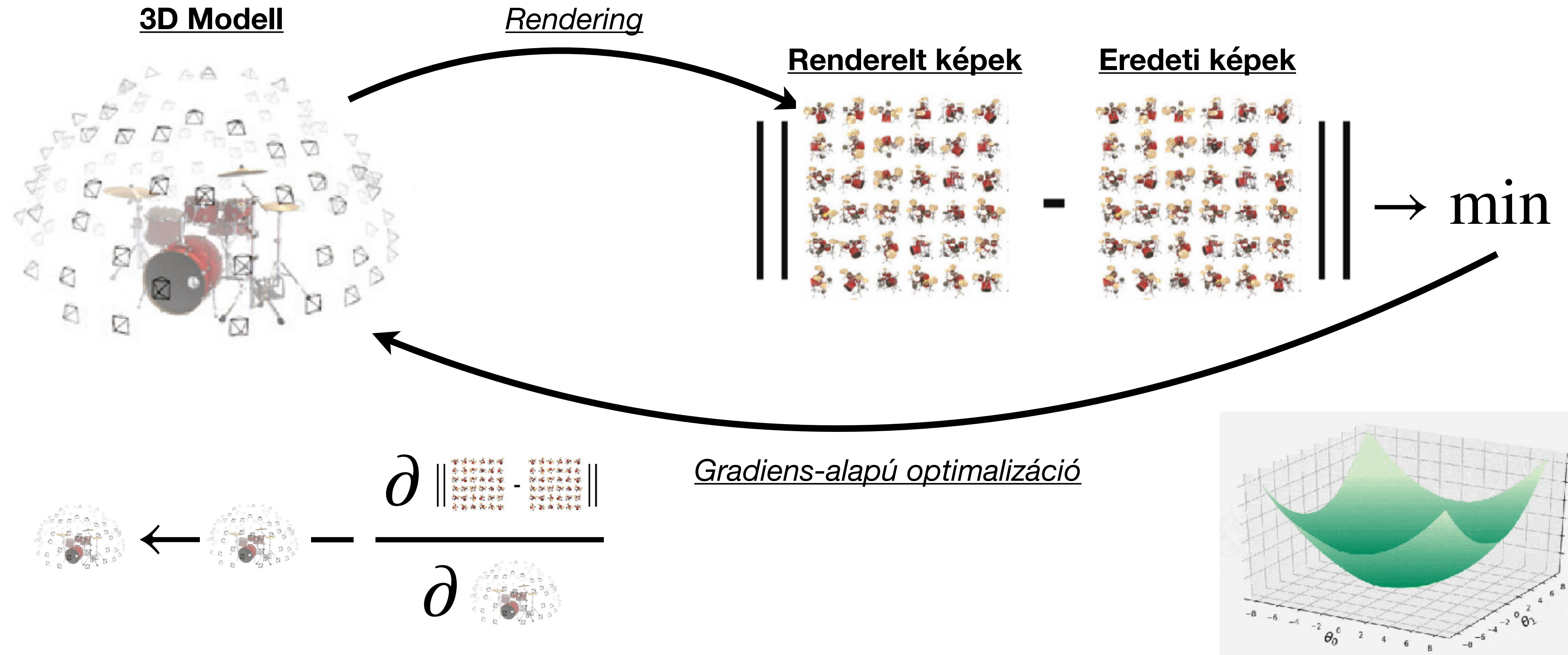
Inverz Grafika

A Kulcs: Differenciálás



Inverz Grafika

A Kulcs: Differenciálás



Inverz Grafika

A Kulcs: Differenciálás

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g} \cdot \frac{\partial g(x)}{\partial x}$$

Differenciálás láncszabálya

$$\frac{\partial \parallel \text{[drumkit grid]} - \text{[drumkit grid]} \parallel}{\partial \text{[drumkit in cloud]}} = \frac{\partial \parallel \text{[drumkit grid]} - \text{[drumkit grid]} \parallel}{\partial \text{[drumkit]}} \cdot \frac{\partial \text{[drumkit]}}{\partial \text{[drumkit in cloud]}}$$

Inverz Grafika

A Kulcs: Differenciálás

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g} \cdot \frac{\partial g(x)}{\partial x}$$

Differenciálás láncszabálya

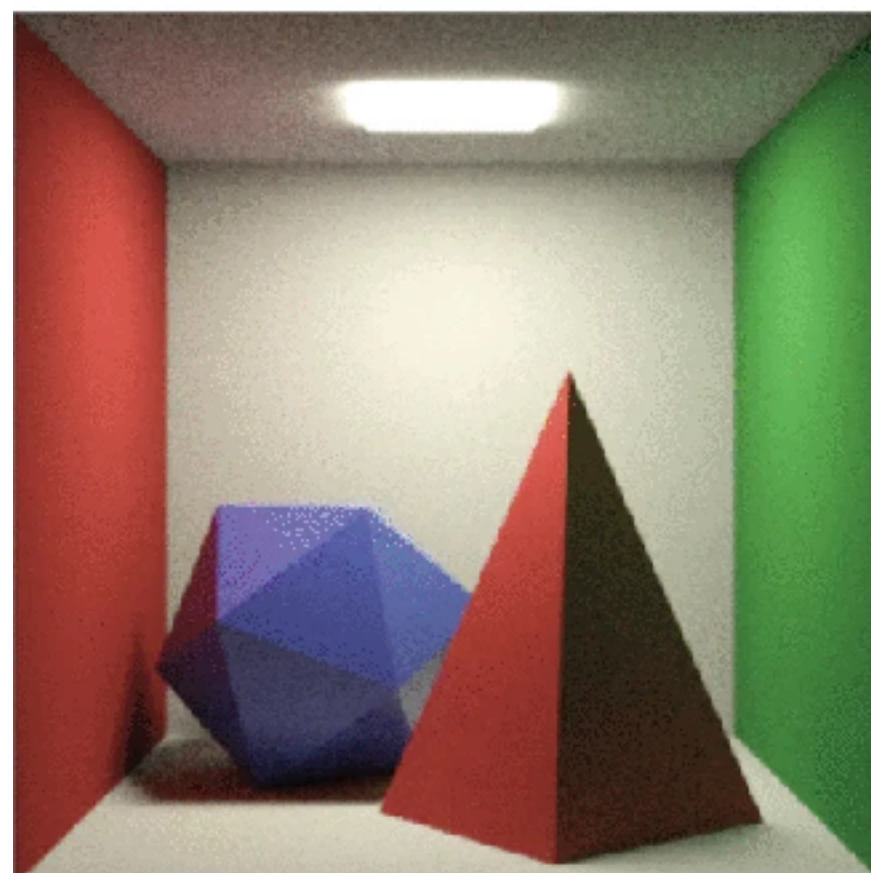
$$\frac{\partial \left\| \begin{array}{c} \text{[Grid of drum icons]} \\ \text{[Grid of drum icons]} \\ \text{[Grid of drum icons]} \end{array} - \begin{array}{c} \text{[Grid of drum icons]} \\ \text{[Grid of drum icons]} \\ \text{[Grid of drum icons]} \end{array} \right\|}{\partial \begin{array}{c} \text{[Drum set in scene with rays]} \end{array}} = \frac{\partial \left\| \begin{array}{c} \text{[Grid of drum icons]} \\ \text{[Grid of drum icons]} \\ \text{[Grid of drum icons]} \end{array} - \begin{array}{c} \text{[Grid of drum icons]} \\ \text{[Grid of drum icons]} \\ \text{[Grid of drum icons]} \end{array} \right\|}{\partial \begin{array}{c} \text{[Drum set image]} \end{array}} \cdot \frac{\partial \begin{array}{c} \text{[Drum set image]} \end{array}}{\partial \begin{array}{c} \text{[Drum set in scene with rays]} \end{array}}$$



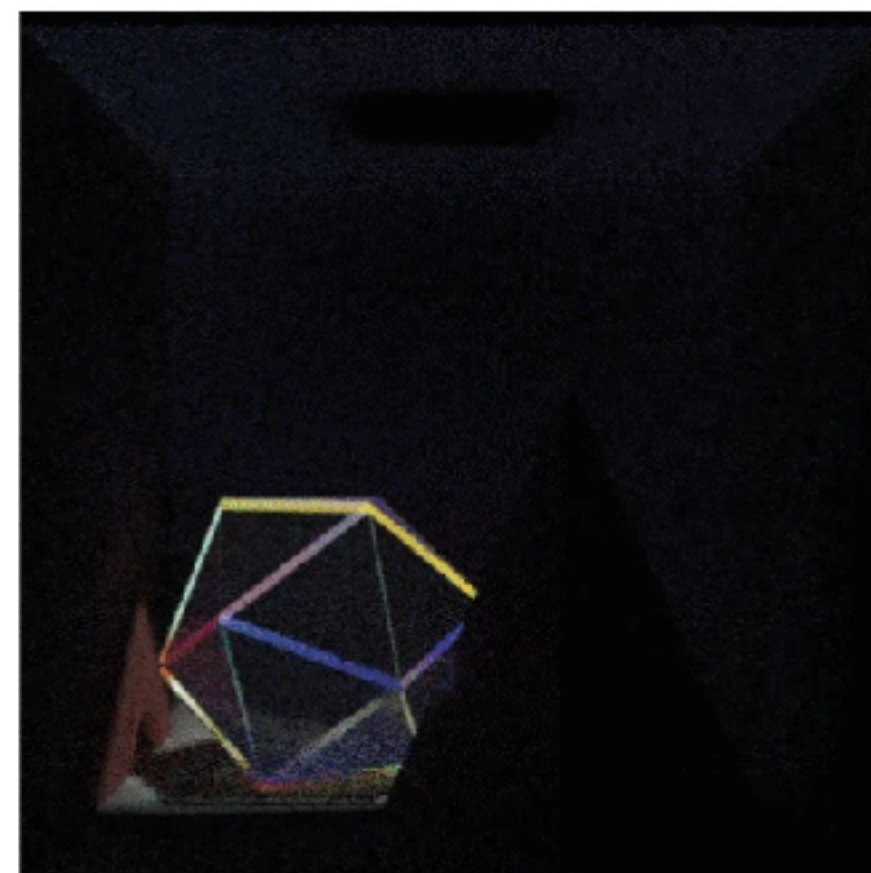
Differenciálható
Rendering?

Inverz Grafika

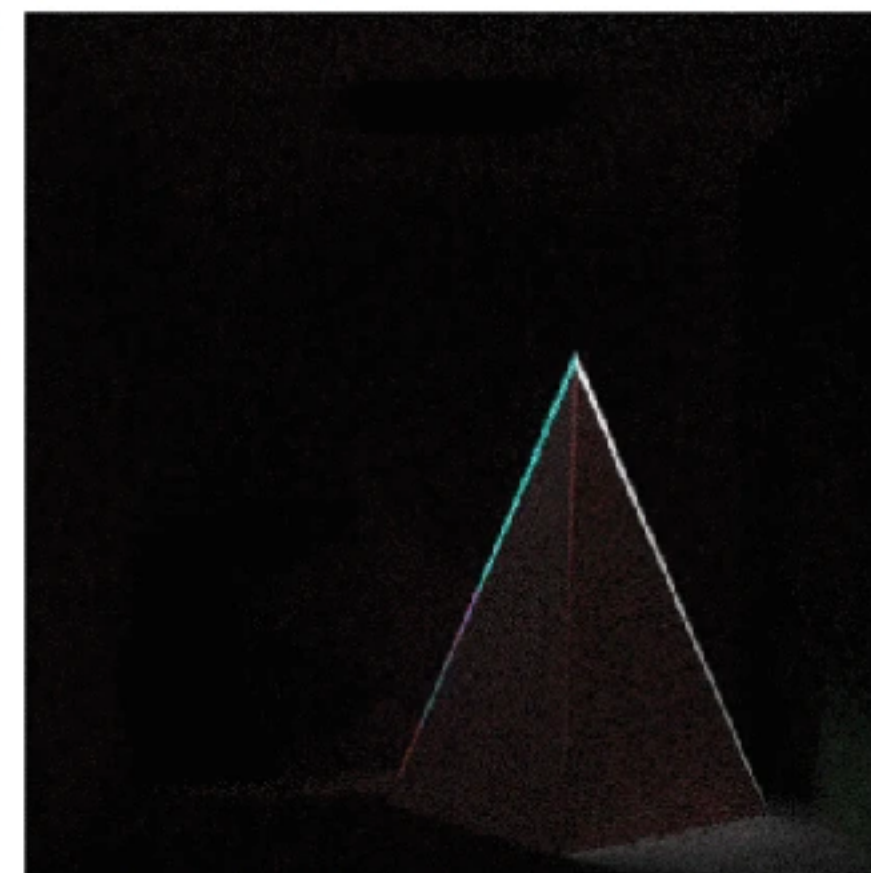
Differenciálható rendering — Gradiensek



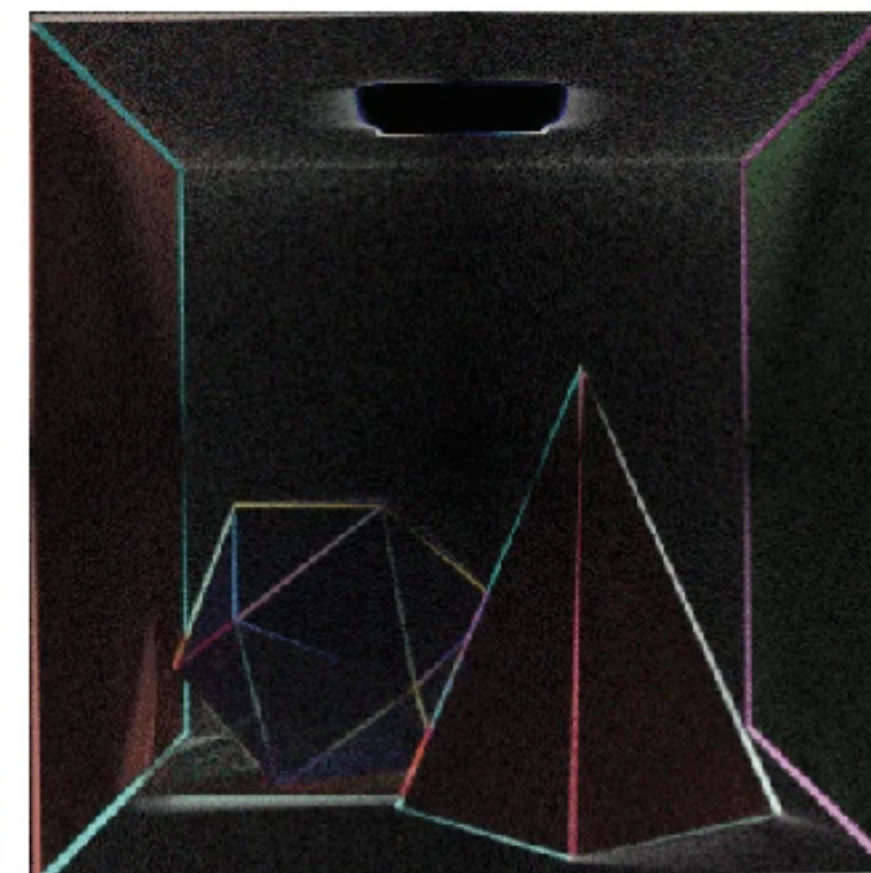
I



$$\frac{\partial I}{\partial \left(\begin{array}{c} \text{object} \\ \text{translation} \end{array} \right)}$$



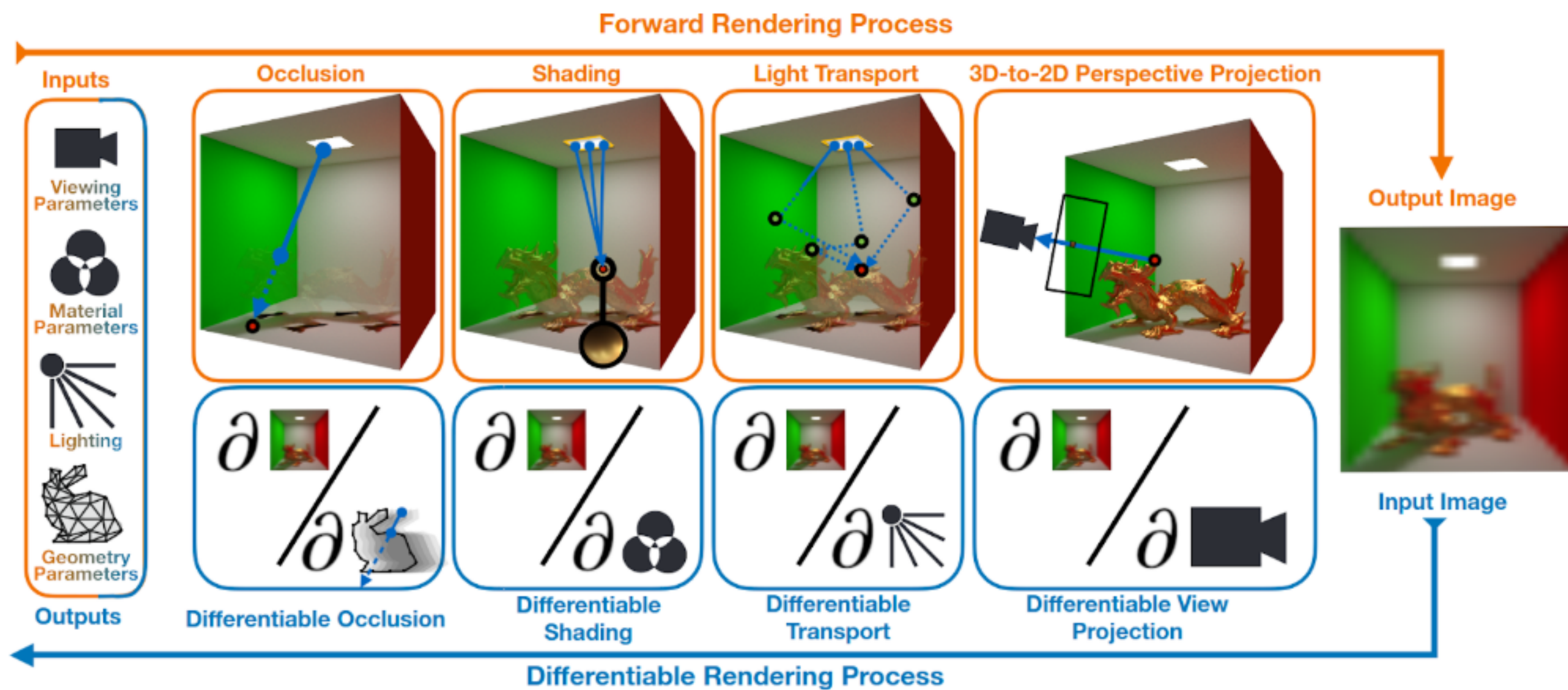
$$\frac{\partial I}{\partial \left(\begin{array}{c} \text{vertex} \\ \text{position} \end{array} \right)}$$



$$\frac{\partial I}{\partial \left(\begin{array}{c} \text{camera} \\ \text{rotation} \end{array} \right)}$$

Inverz Grafika

Differenciálható rendering



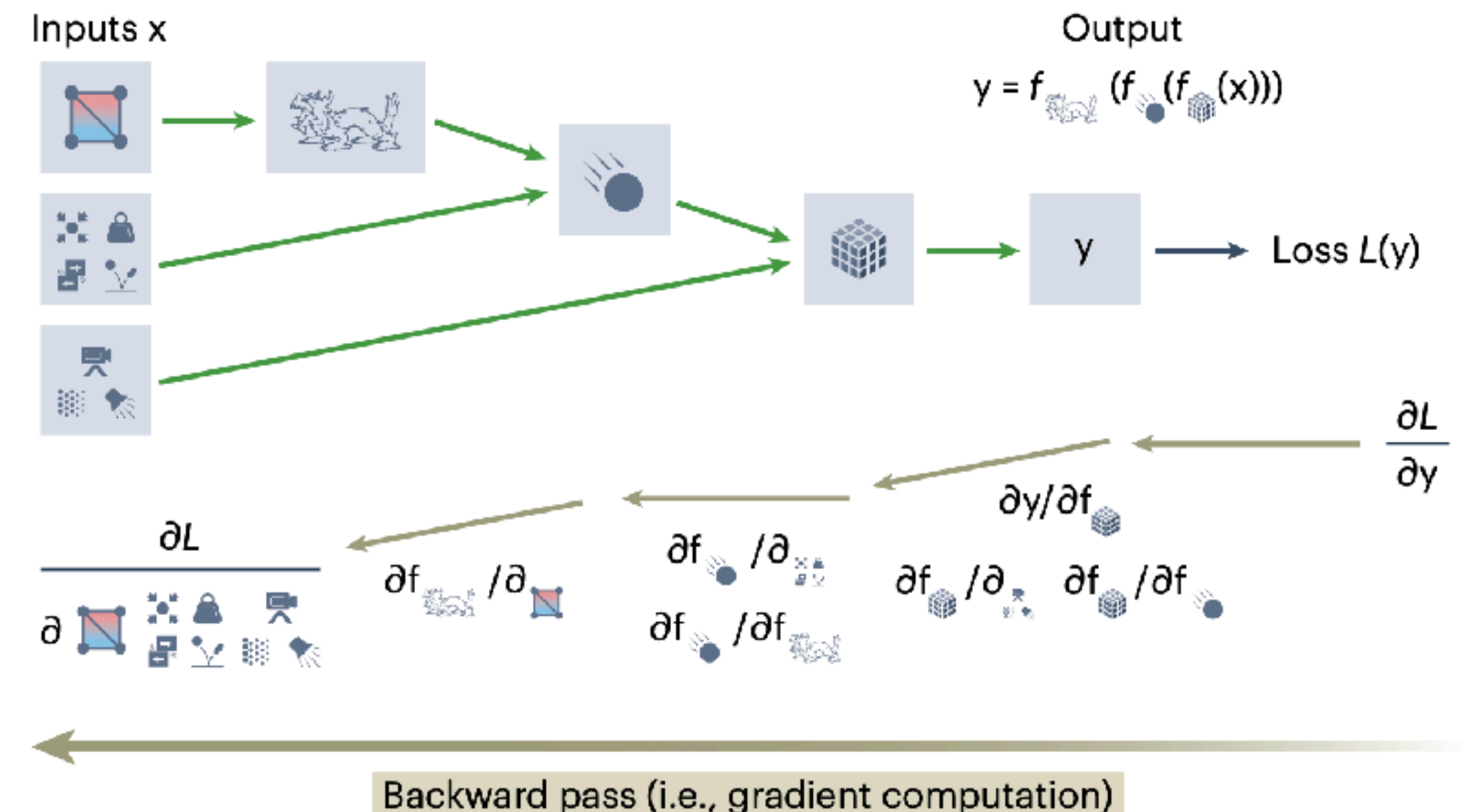
Inverz Grafika

Differenciálható rendering

- *Bármilyen* differenciálható folyamat invertálható a képek rekonstrukciós hibájának gradiens-alapú optimalizációjával!
- Példák: rendering, fizikai szimuláció, képfeldolgozás, neurális hálók
- Fő feladat: optimalizált függvény (pl. pixelhiba loss) deriválása az optimalizált paraméterek alapján

```
def visual_computing (x = { , , , , , ,  }):
    f  = geometry_processing (x)
    f  = animation (x, f )
    f  = rendering (x, f )
    y = loss (f )
    return y
```

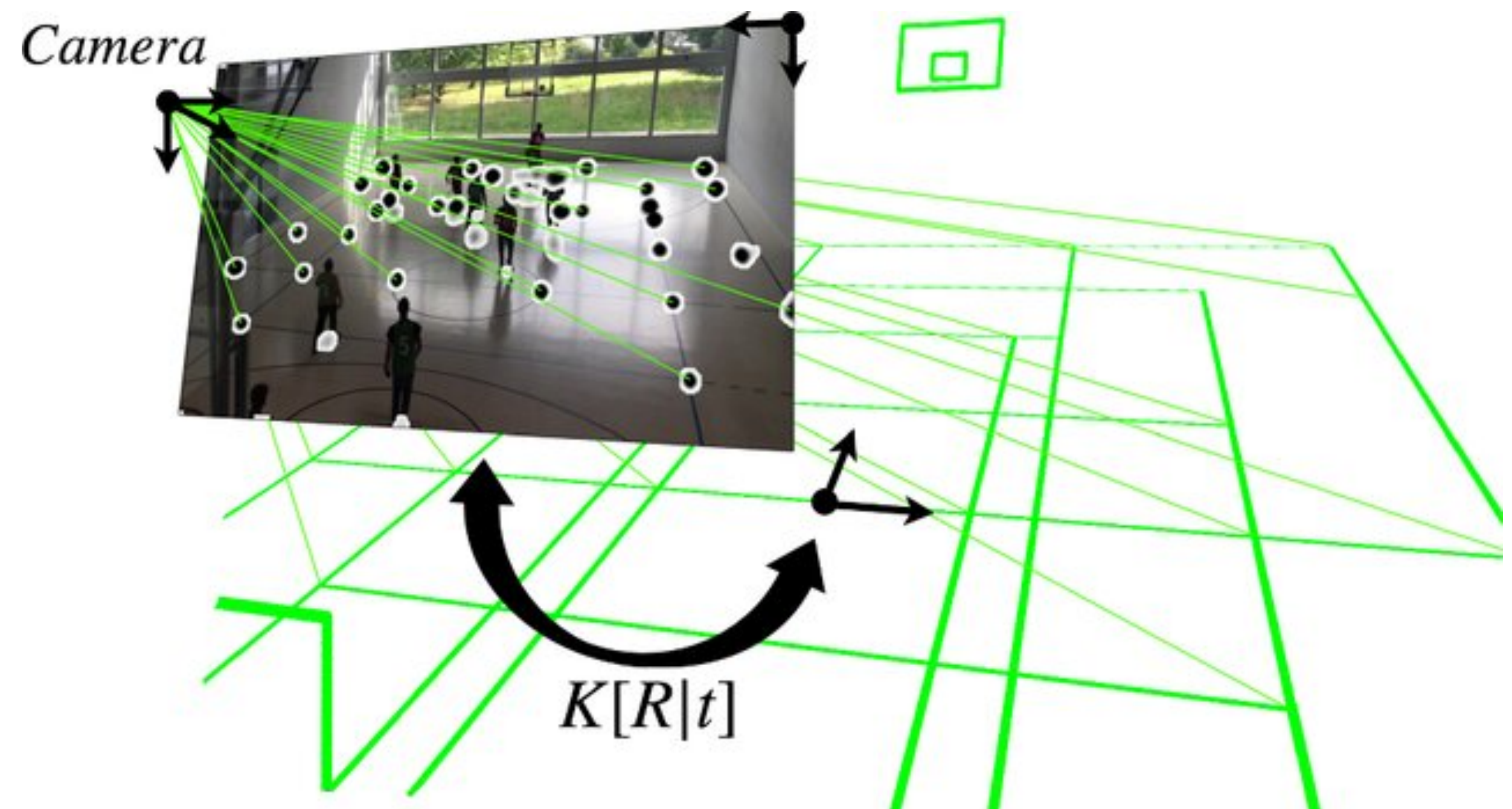
Forward pass (i.e., program execution)



Inverz Grafika

Kamera paraméterek

- A képekhez tartozó kamera paramétereket (fókusz távolság, pozíció, orientáció) az inverz rendering feladatban gyakran ismertnek tekintjük!
- Lehetséges a kamera paramétereket is optimalizálni a 3D geometriával együtt — de a feladat jóval nehezebbé válik
- Praktikus megoldások szorosan kapcsolódnak a fotogrammetria / SLAM módszerekhez
 - Még visszatérünk rá...

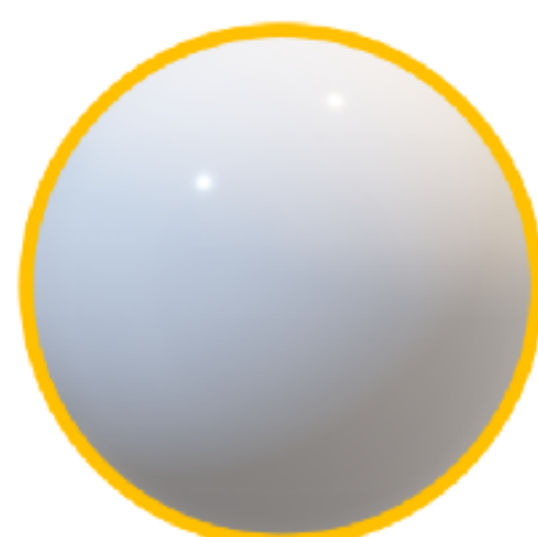


Inverz Grafika

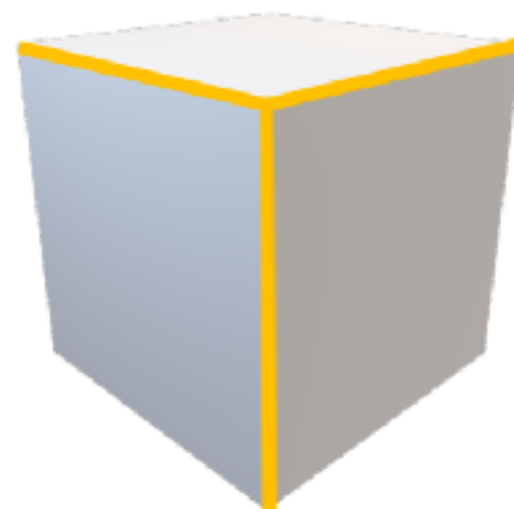
Láthatósági gradiensek / sziluettek problémája



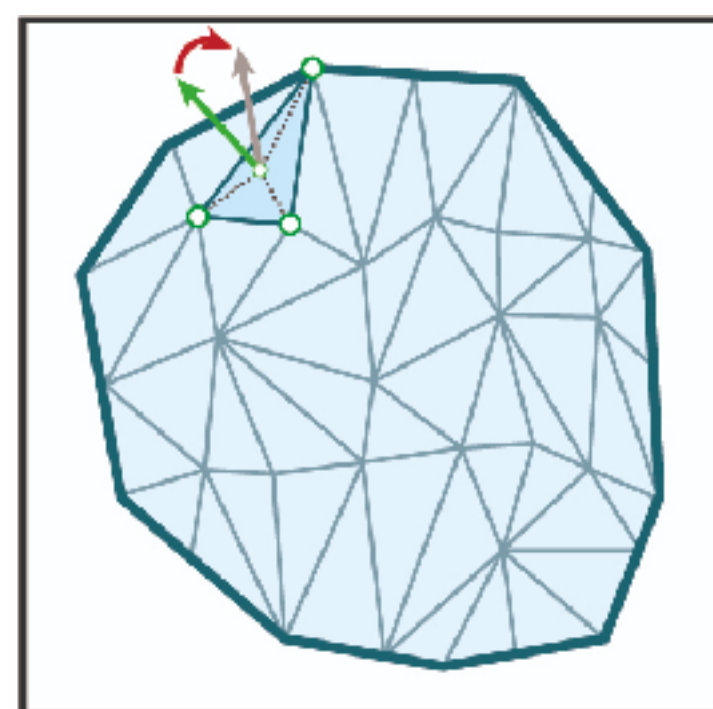
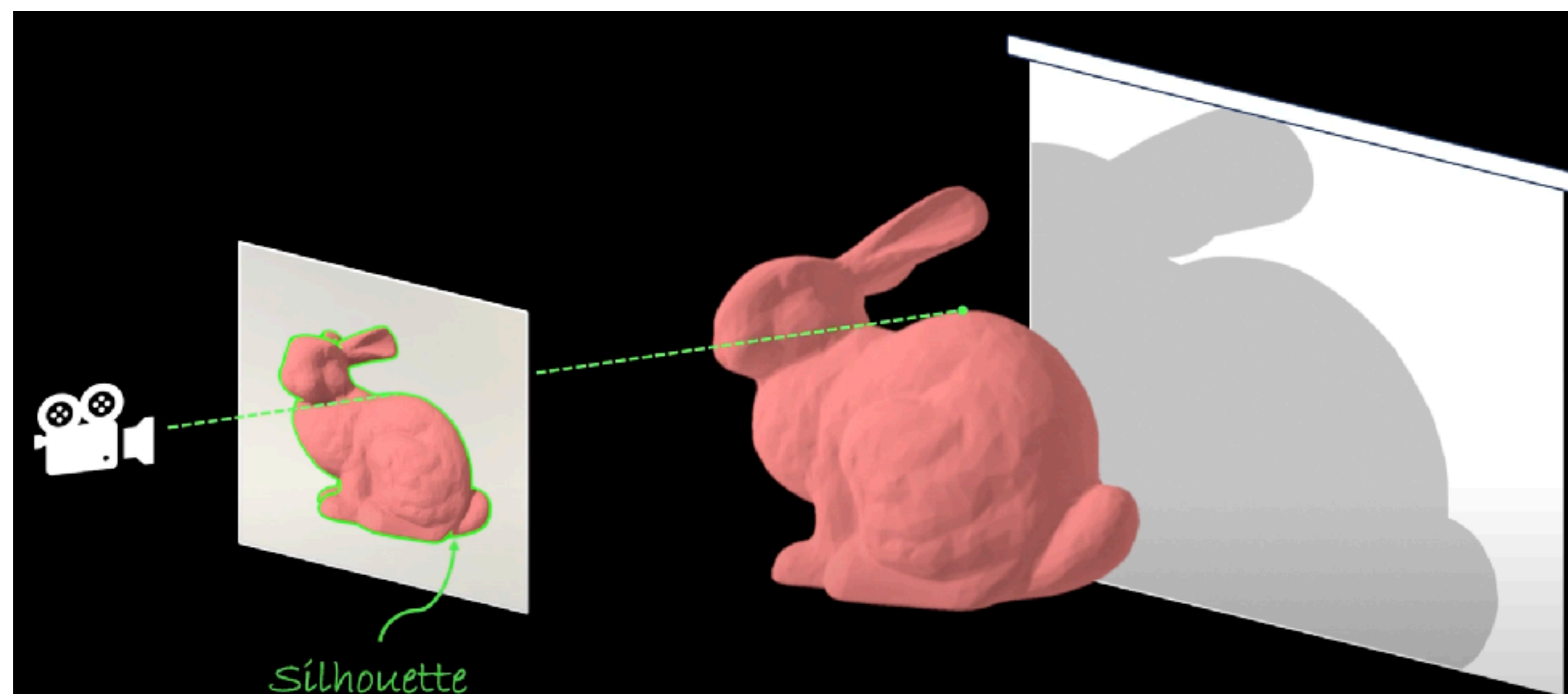
(a) Boundary edges



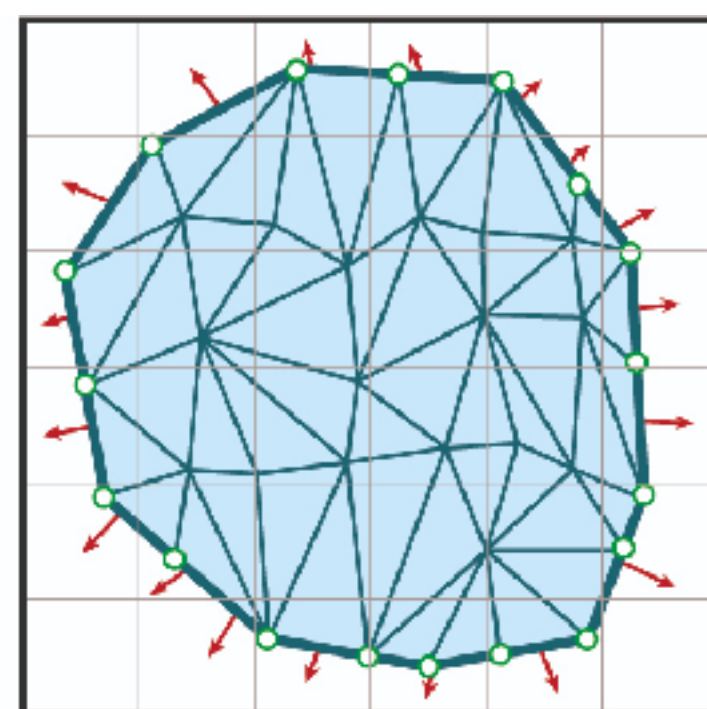
(b) Silhouette edges



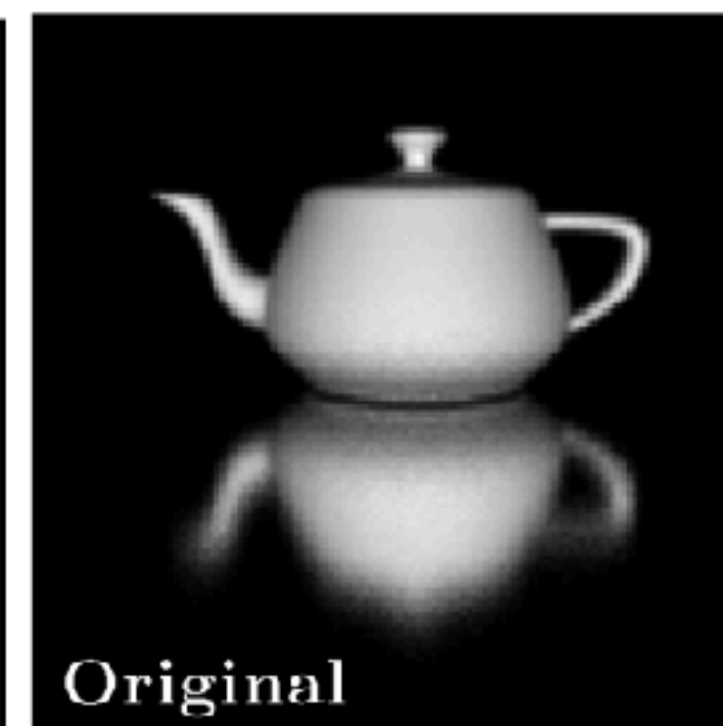
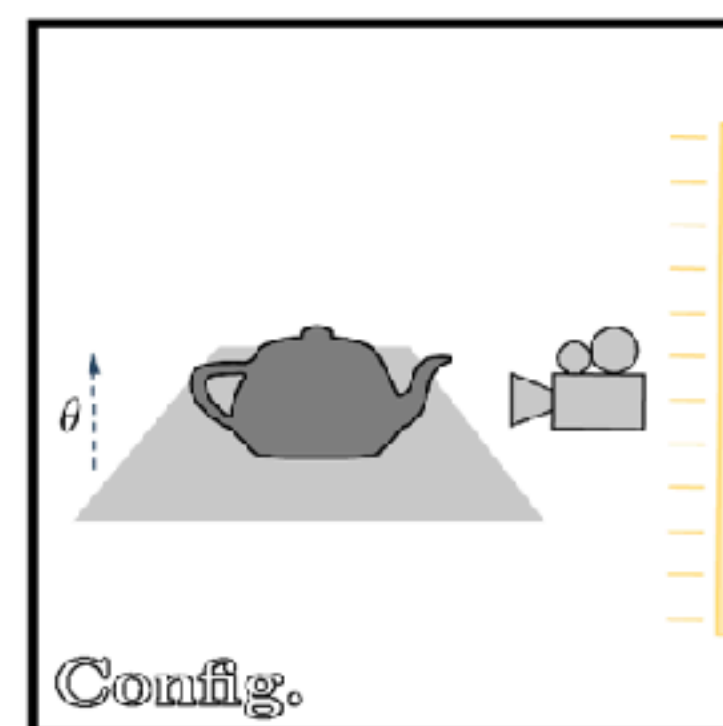
(c) Sharp edges



(a) Shading gradients



(b) Silhouette gradients



Inverz Grafika

Differenciálható raszterizáció

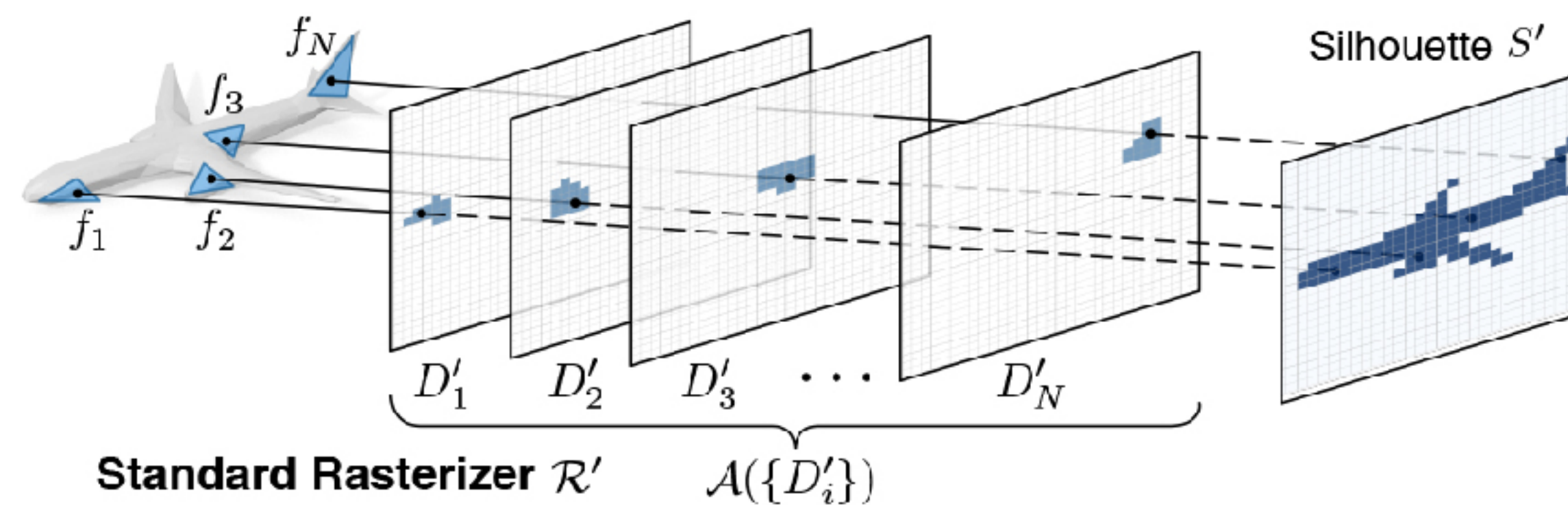
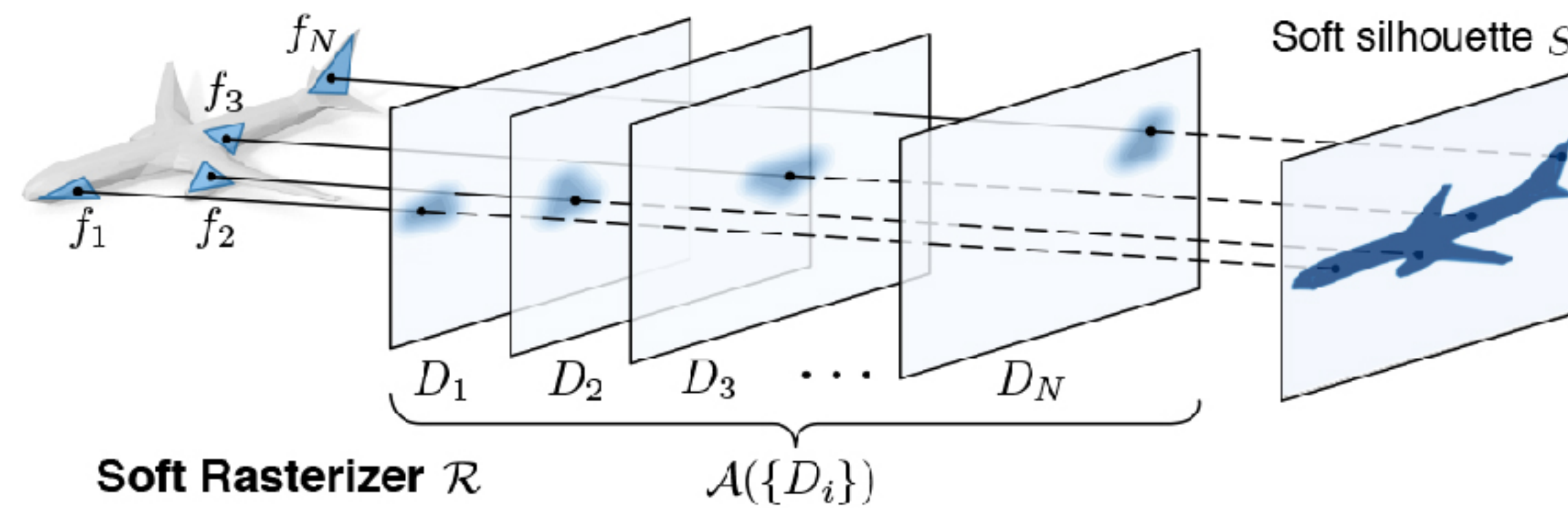
Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning

Shichen Liu^{1,2}, Tianye Li^{1,2}, Weikai Chen¹, and Hao Li^{1,2,3}

¹USC Institute for Creative Technologies

²University of Southern California

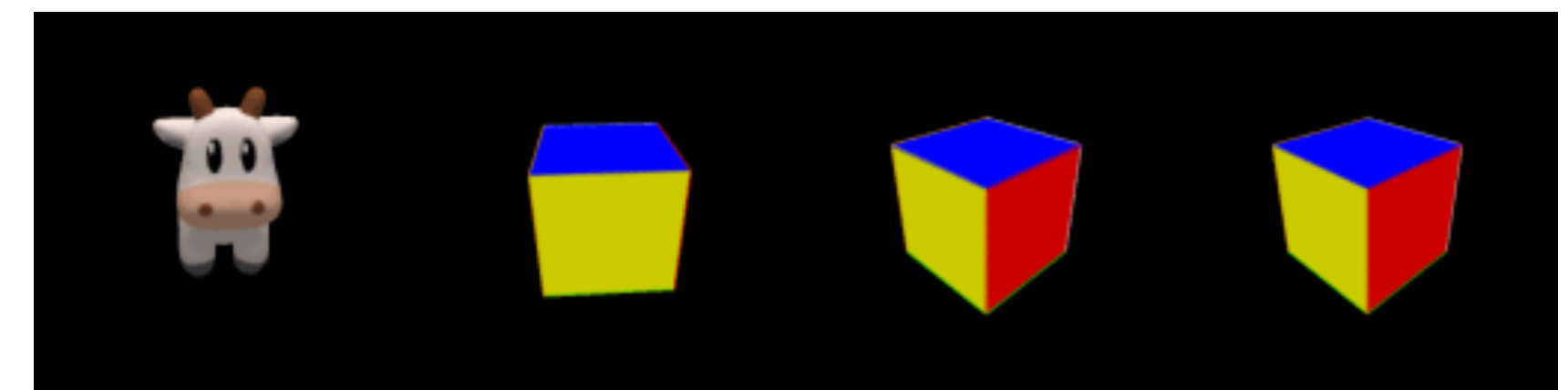
³Pinscreen



(a) ground truth (b) $\sigma = 0.003$ (c) $\sigma = 0.01$ (d) $\sigma = 0.03$

$$\mathcal{D}_j^i = \text{sigmoid}(\delta_j^i \cdot \frac{d^2(i, j)}{\sigma})$$

“Éles” háromszögek helyett elsimított sziluettek!



Inverz Grafika

Differenciálható raszterizáció

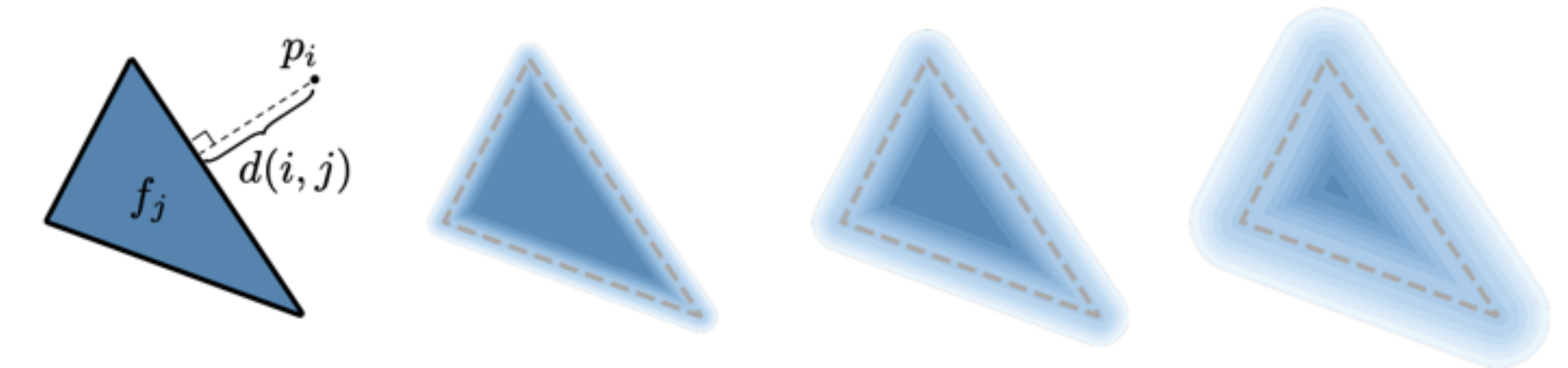
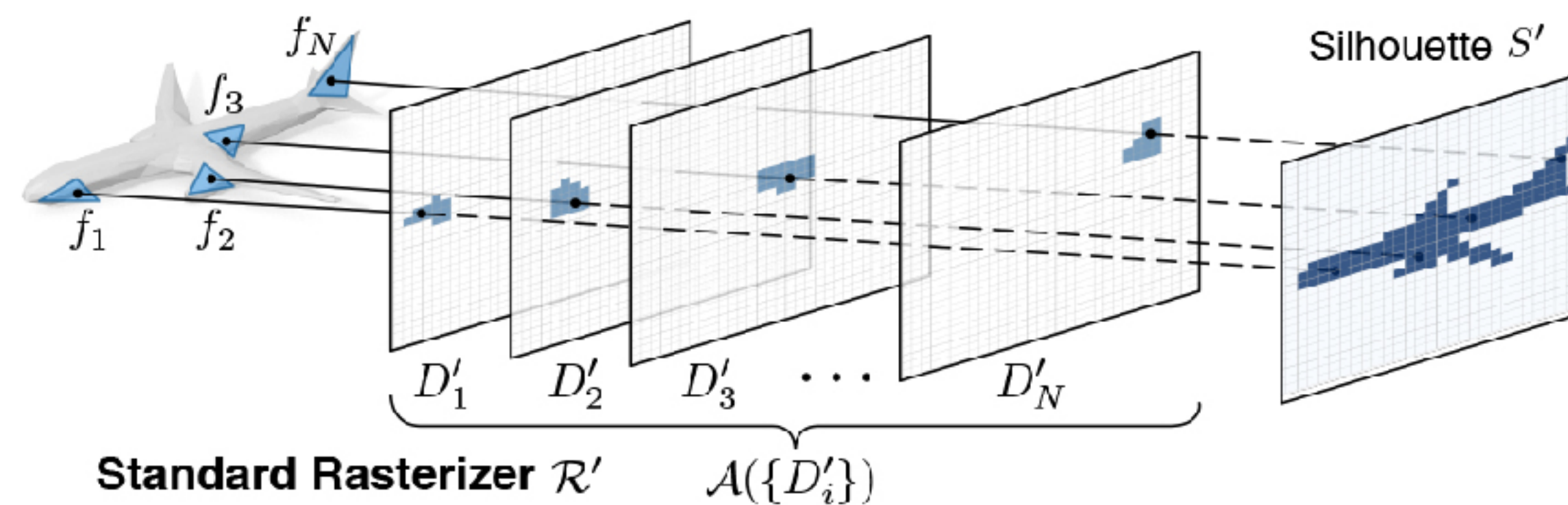
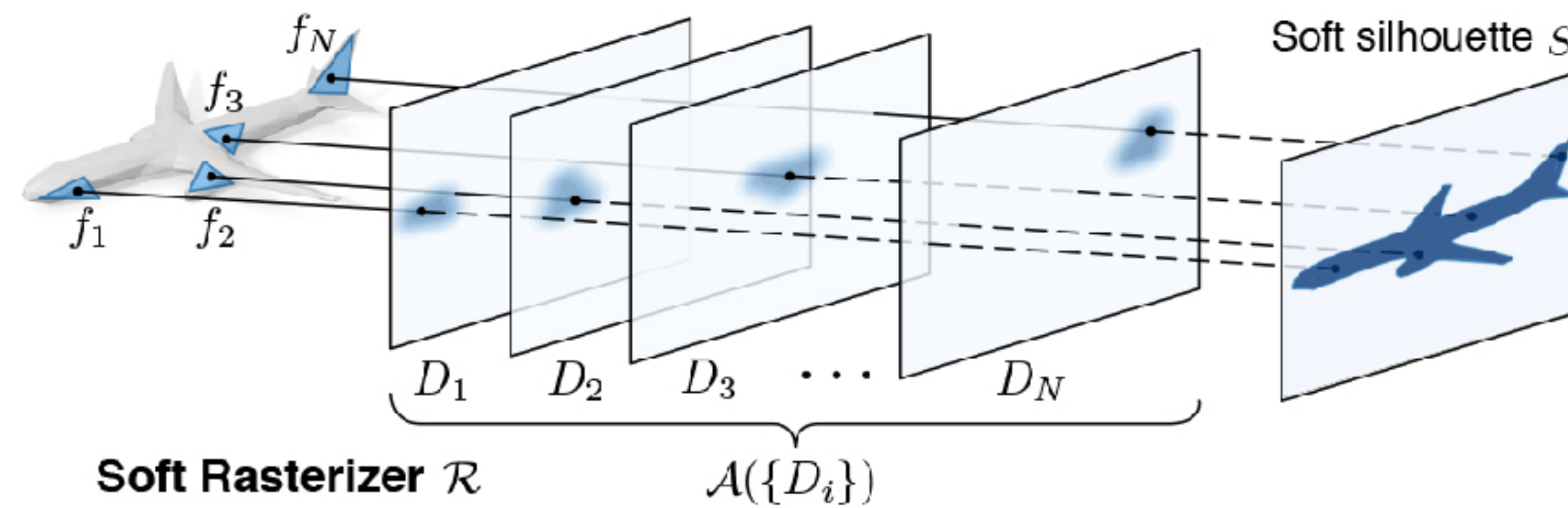
Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning

Shichen Liu^{1,2}, Tianye Li^{1,2}, Weikai Chen¹, and Hao Li^{1,2,3}

¹USC Institute for Creative Technologies

²University of Southern California

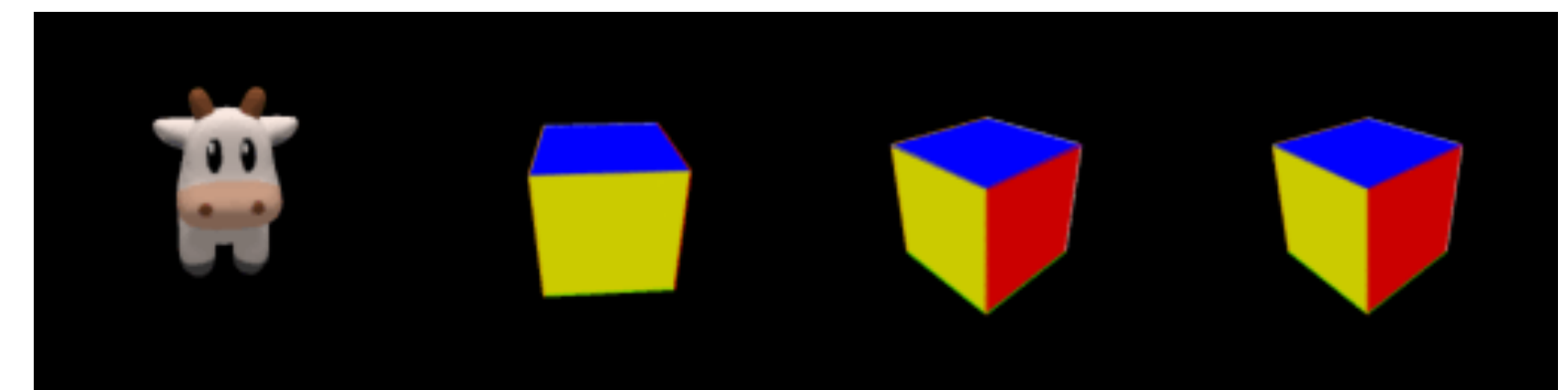
³Pinscreen



(a) ground truth (b) $\sigma = 0.003$ (c) $\sigma = 0.01$ (d) $\sigma = 0.03$

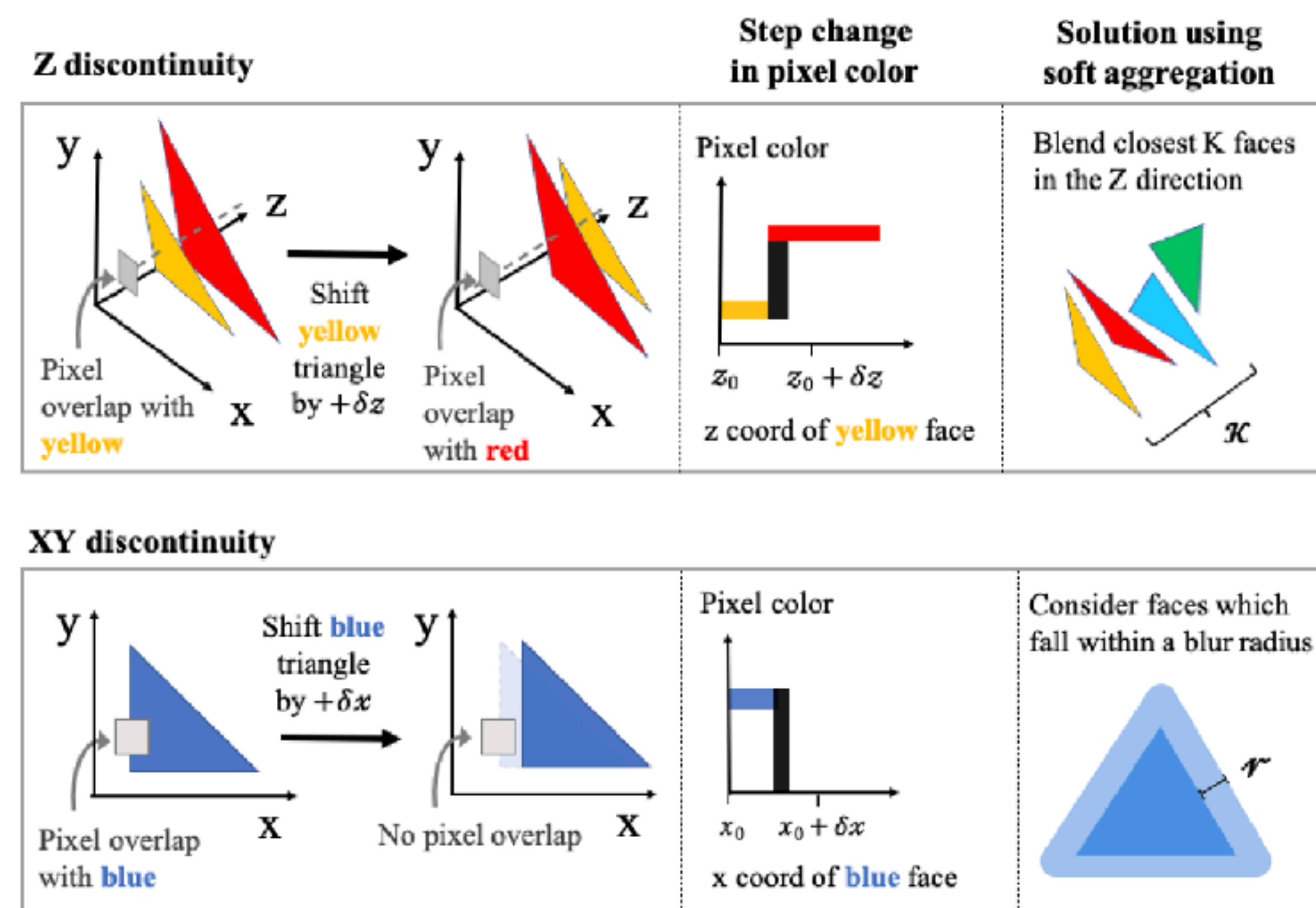
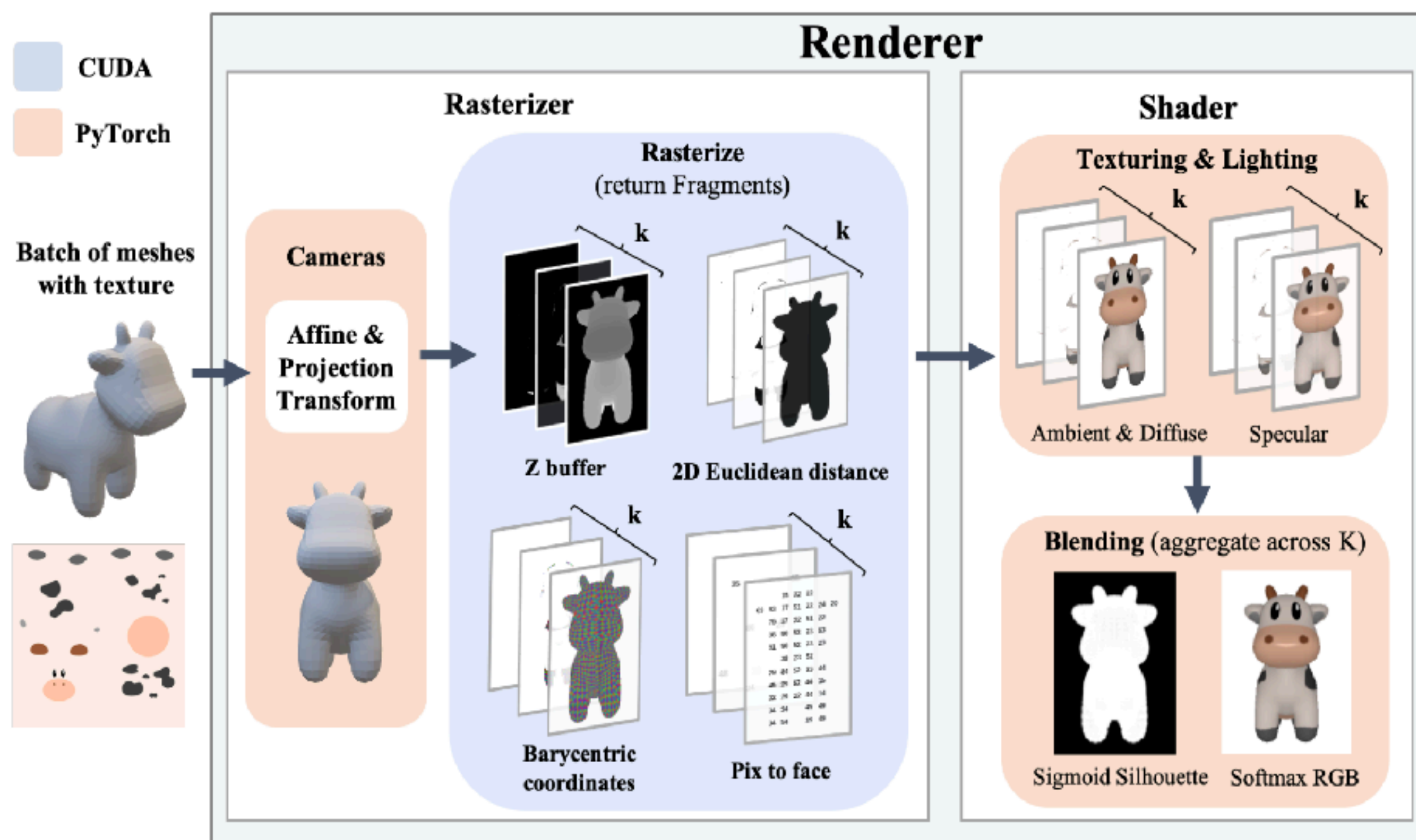
$$\mathcal{D}_j^i = \text{sigmoid}(\delta_j^i \cdot \frac{d^2(i, j)}{\sigma})$$

“Éles” háromszögek helyett elsimított sziluettek!



Inverz Grafika

Differenciálható raszterizáció



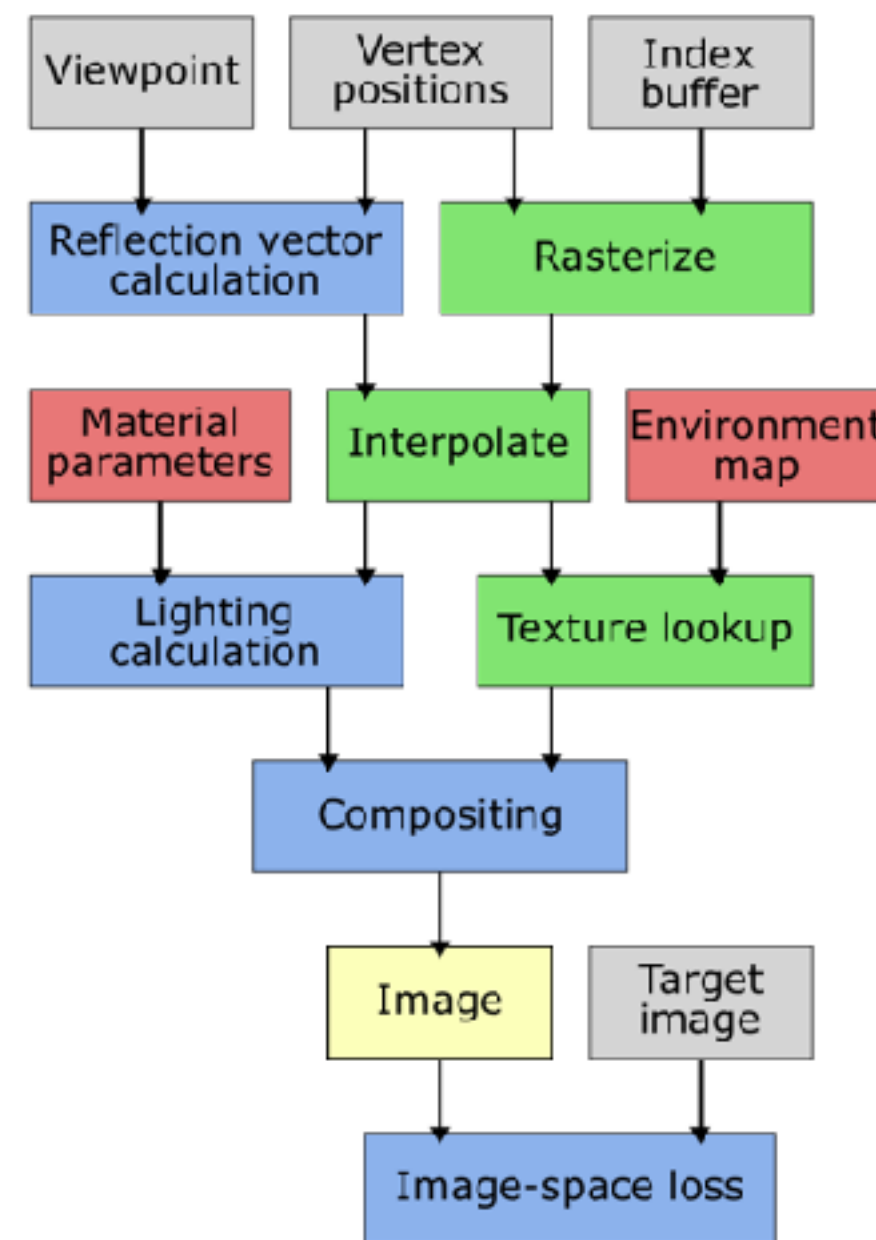
<https://pytorch3d.org/>

Inverz Grafika

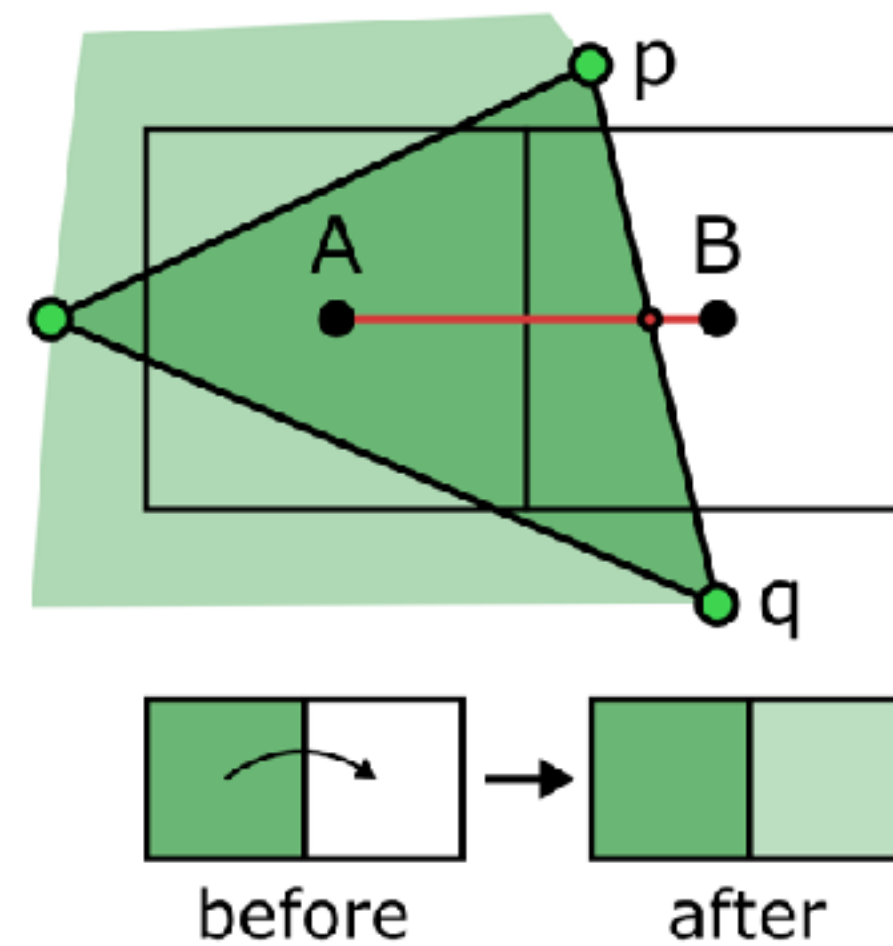
Differenciálható raszterizáció

Nvdiffrast – Modular Primitives for High-Performance Differentiable Rendering

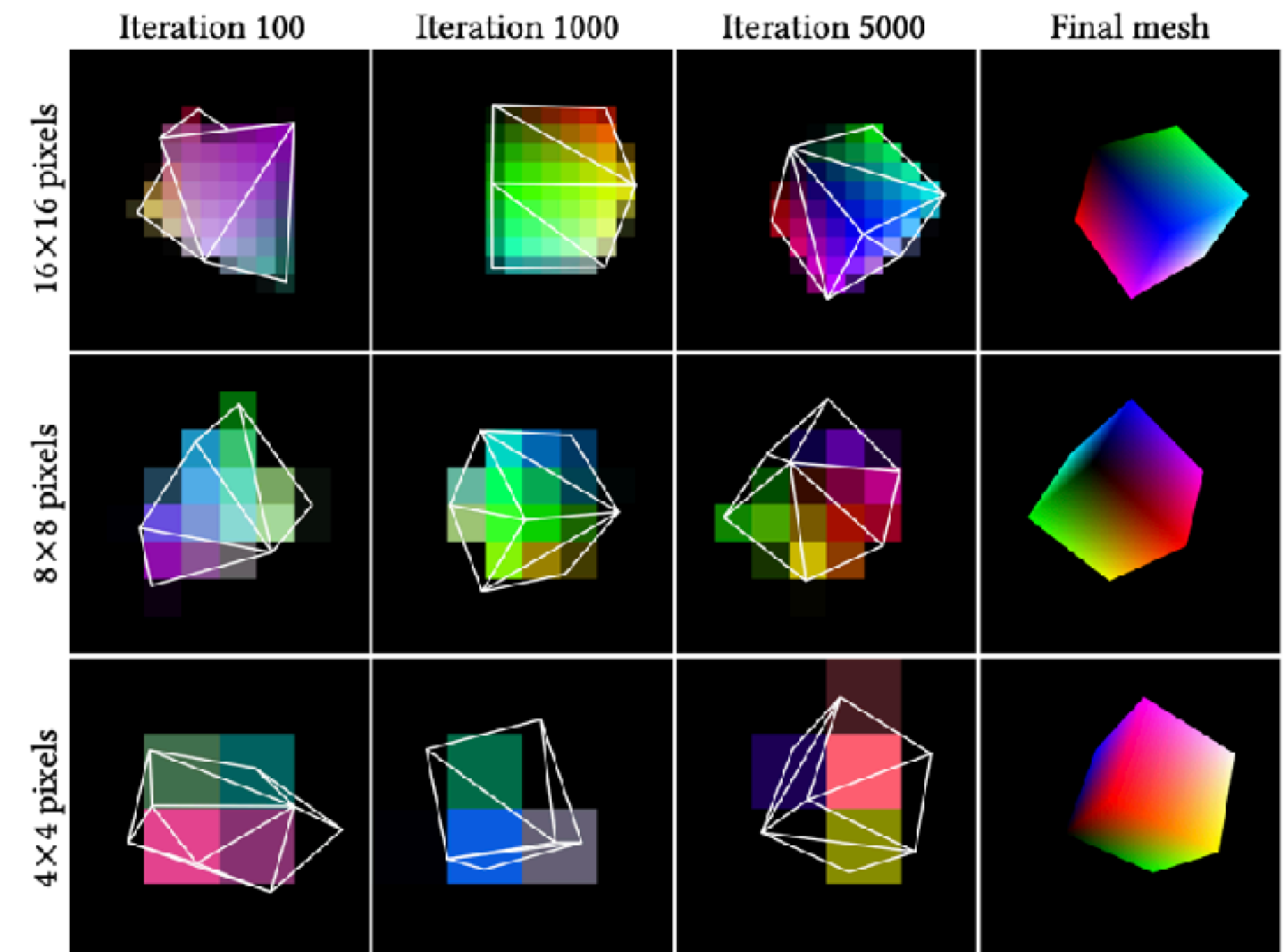
<https://github.com/NVlabs/nvdiffrast>



Moduláris felépítés



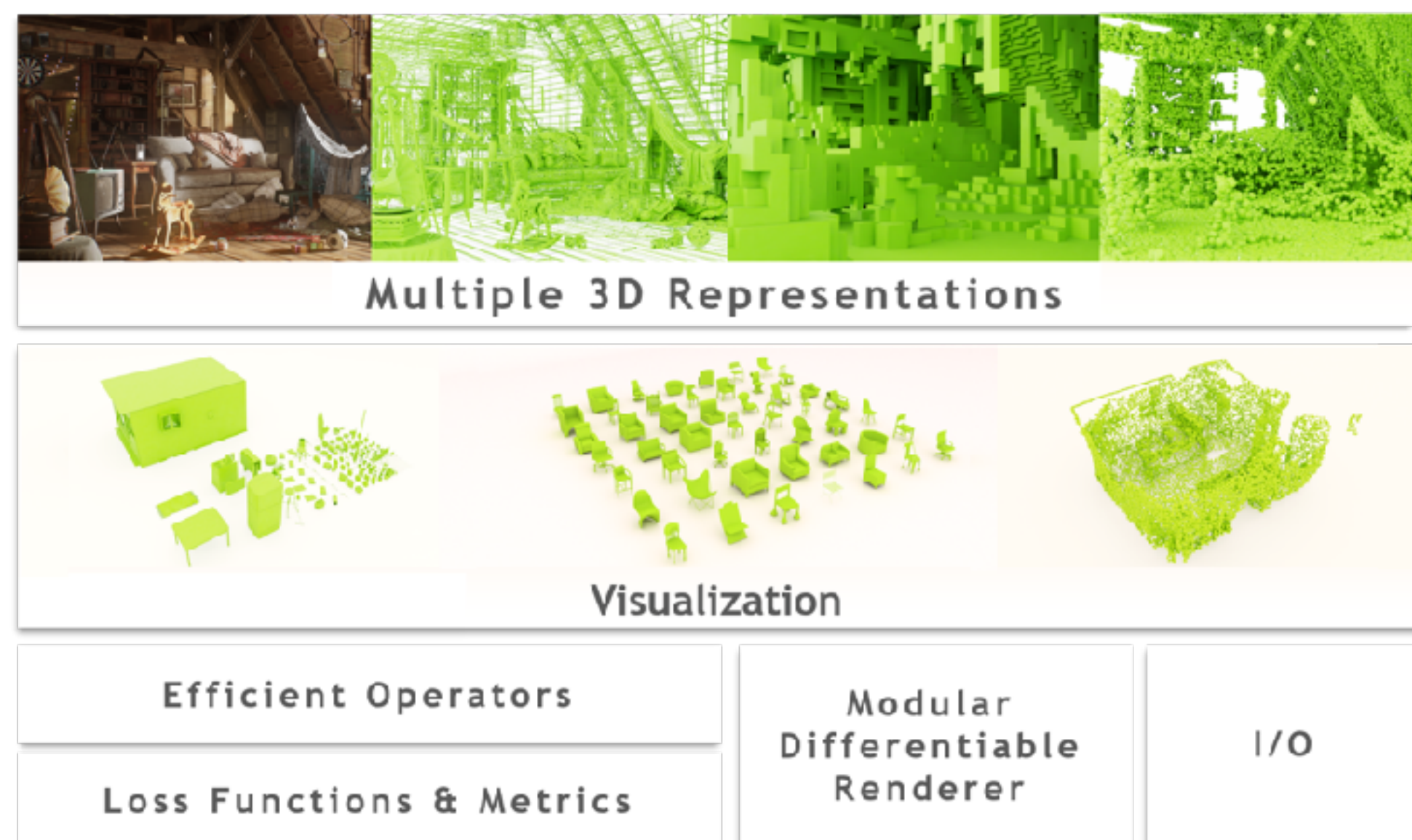
Láthatósági gradiens
képtérbeli anti-aliasing
technikákkal



“Értelmes” gradiens
extrém kis felbontáson is!

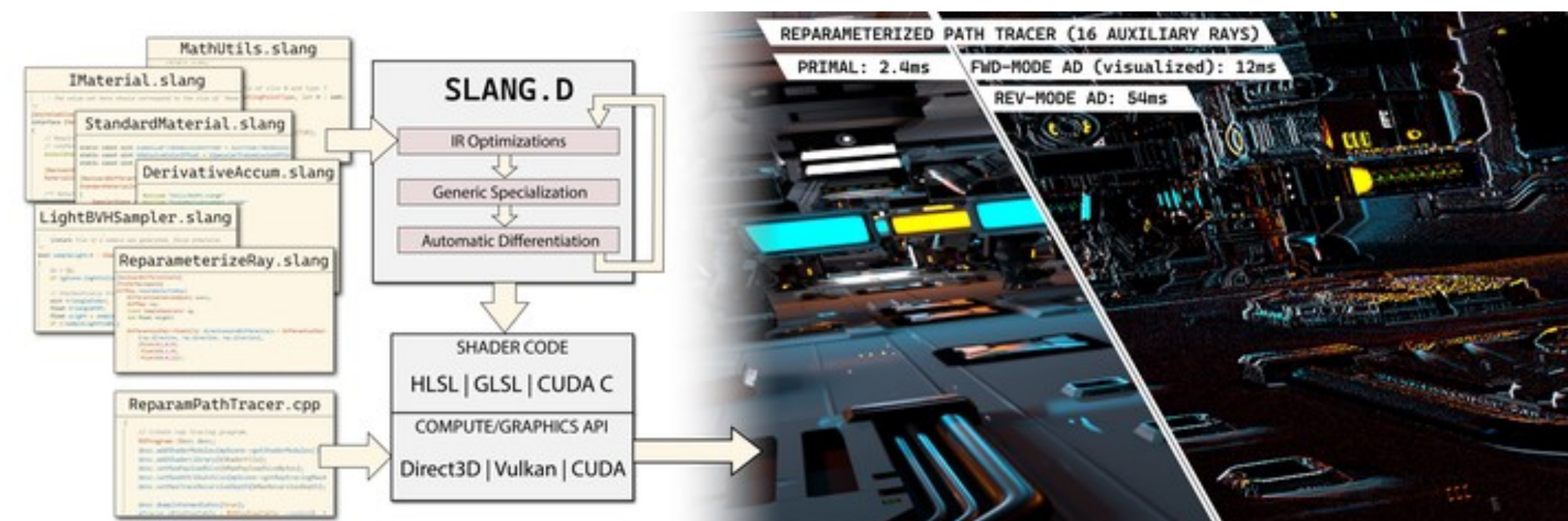
Inverz Grafika

Differenciálható raszterizáció



<https://github.com/NVIDIAGameWorks/kaolin>

Integráció deep learning alkalmazásokkal

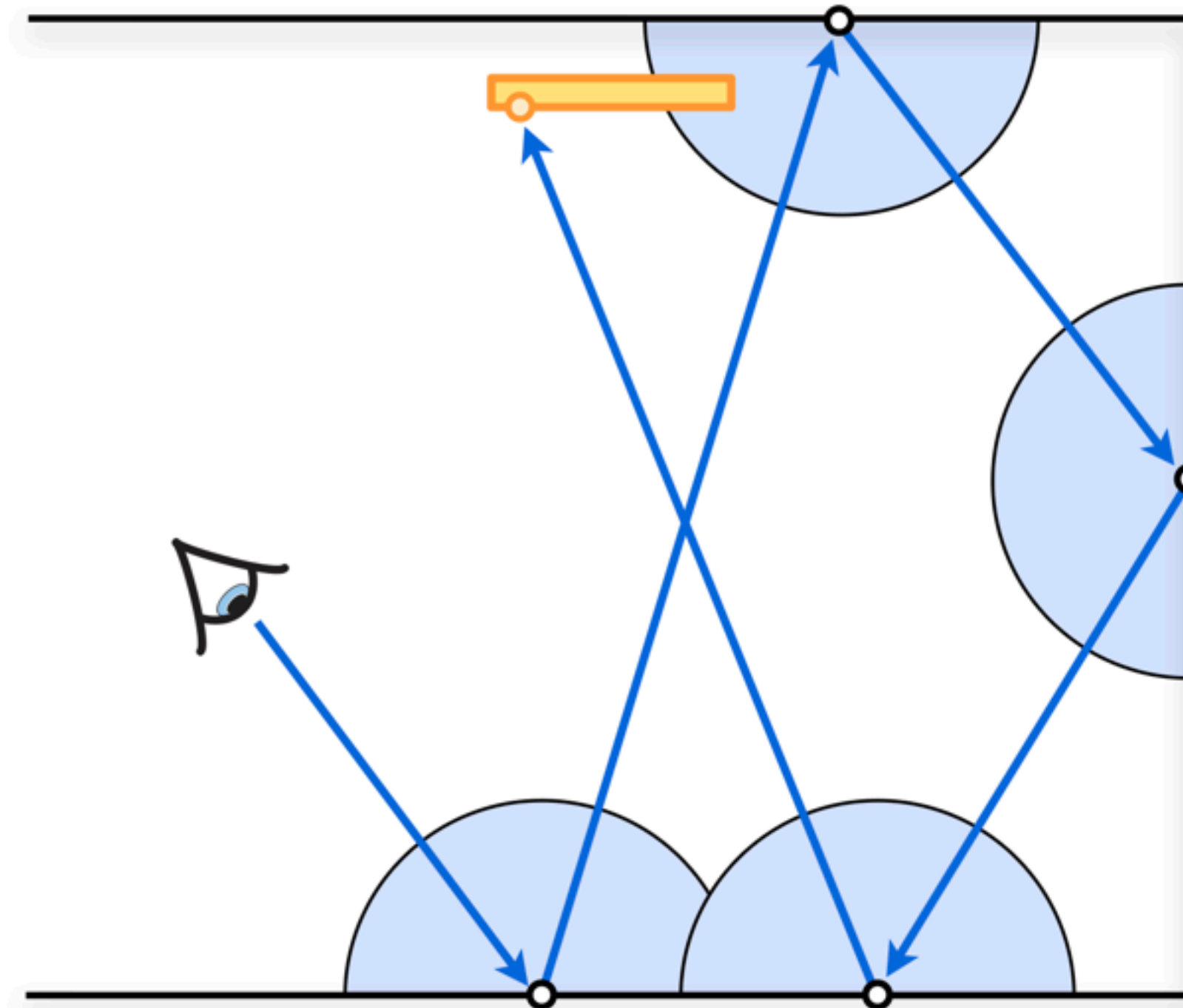


<https://shader-slang.org/>

Differenciálható snadernyelv

Inverz Grafika

Differenciálható sugárkövetés

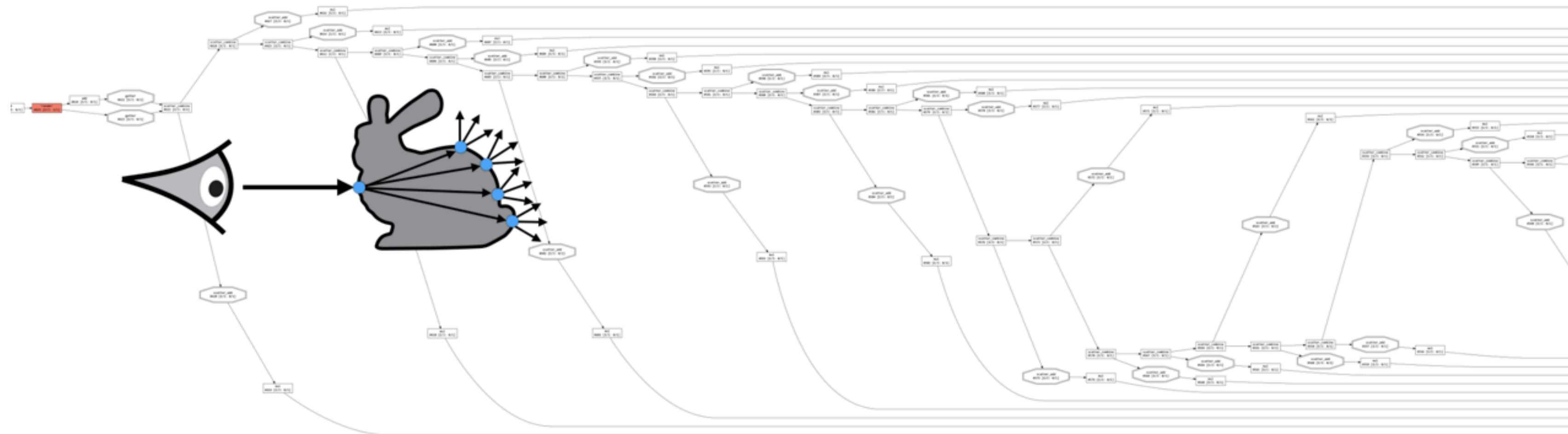


Sugrákővetés: komplex, rekurzív számítások — hogyan differenciáljuk?

Inverz Grafika

Differenciálható sugárkövetés

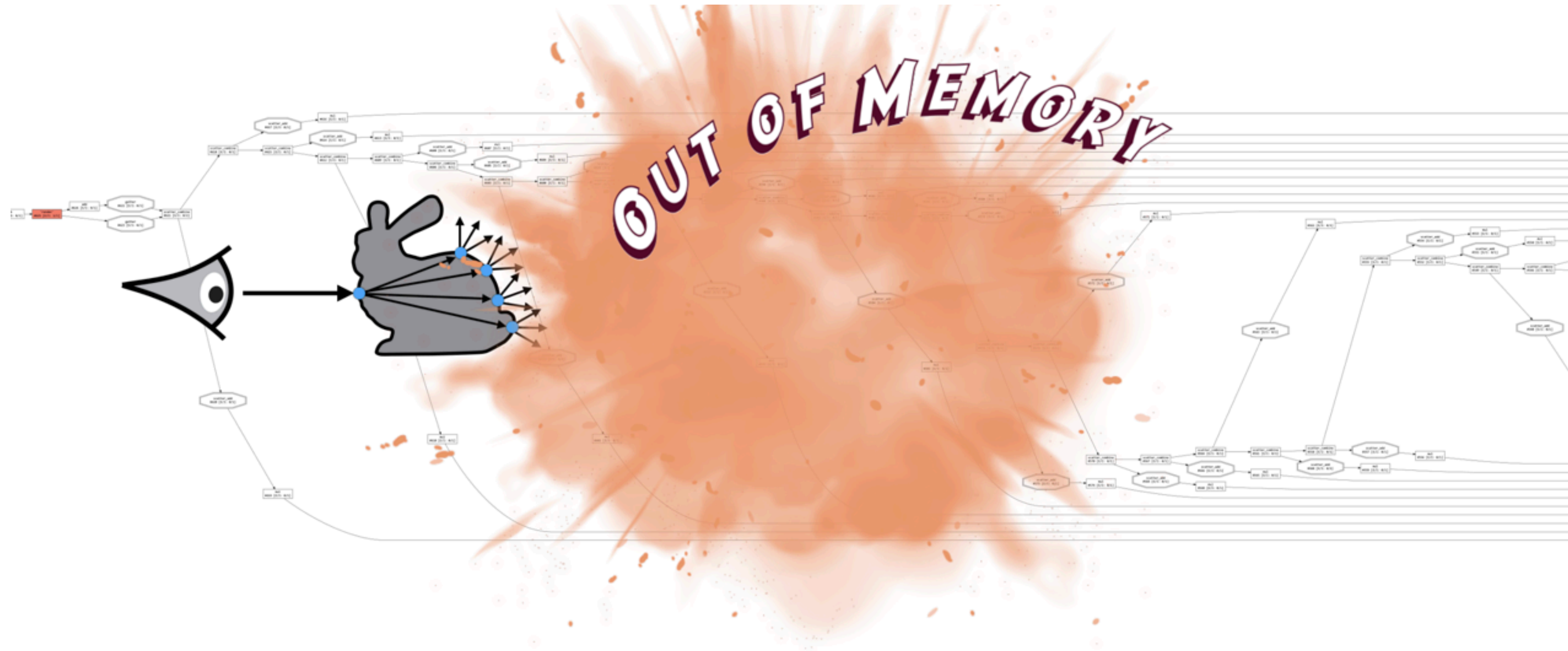
Alkalmazzunk direkt automatikus differenciálást?



Inverz Grafika

Differenciálható sugárkövetés

Alkalmazzunk direkt automatikus differenciálást?

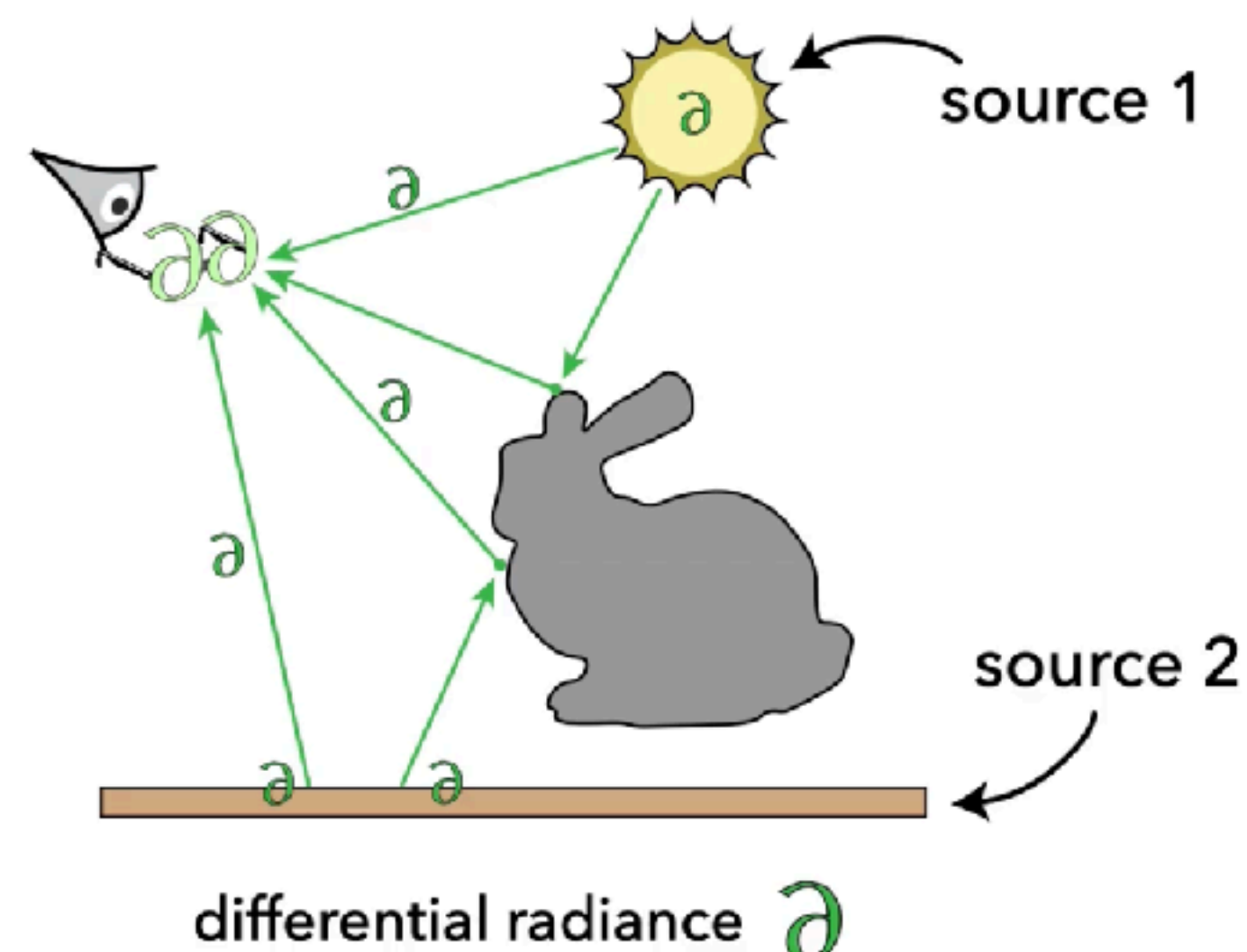
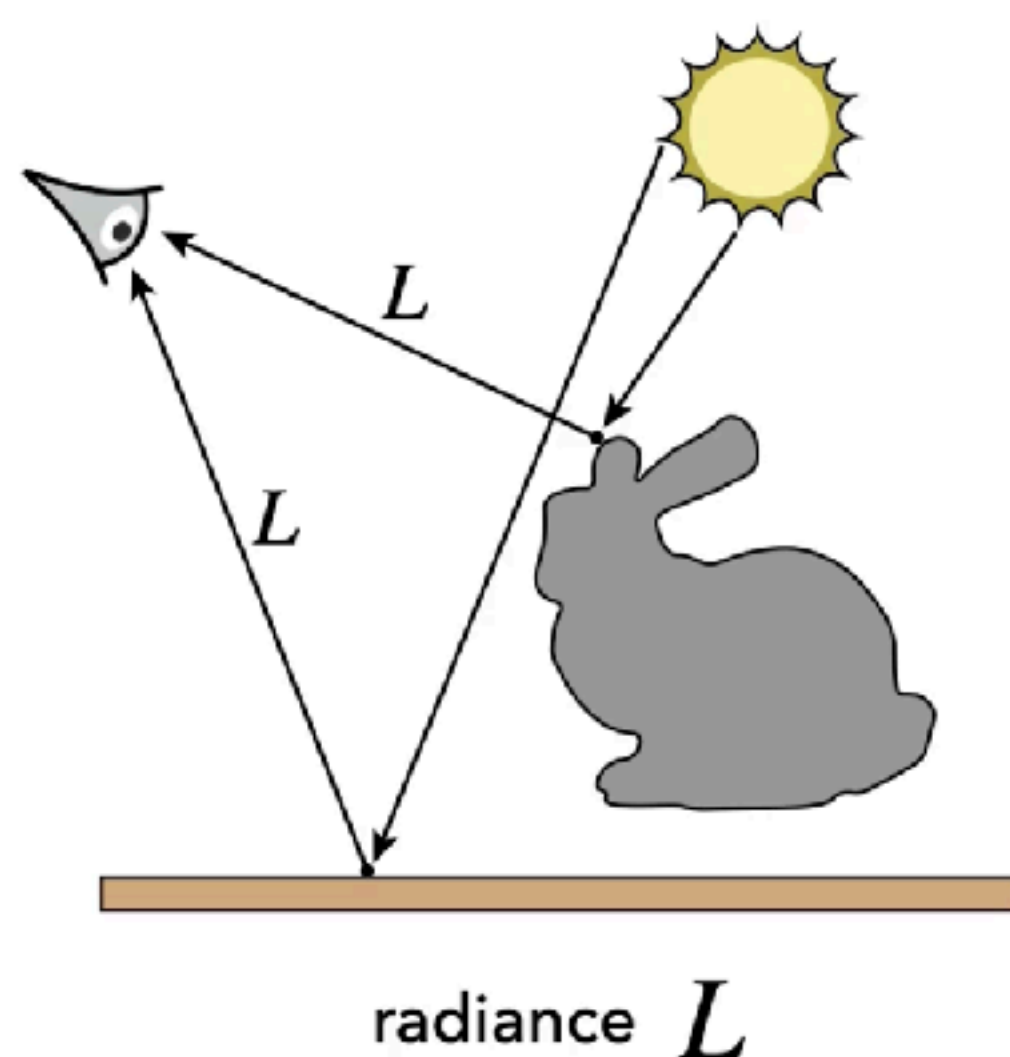


Valami “okosabb” módszerre lesz szükség...

Inverz Grafika

Differenciálható sugárkövetés

Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering



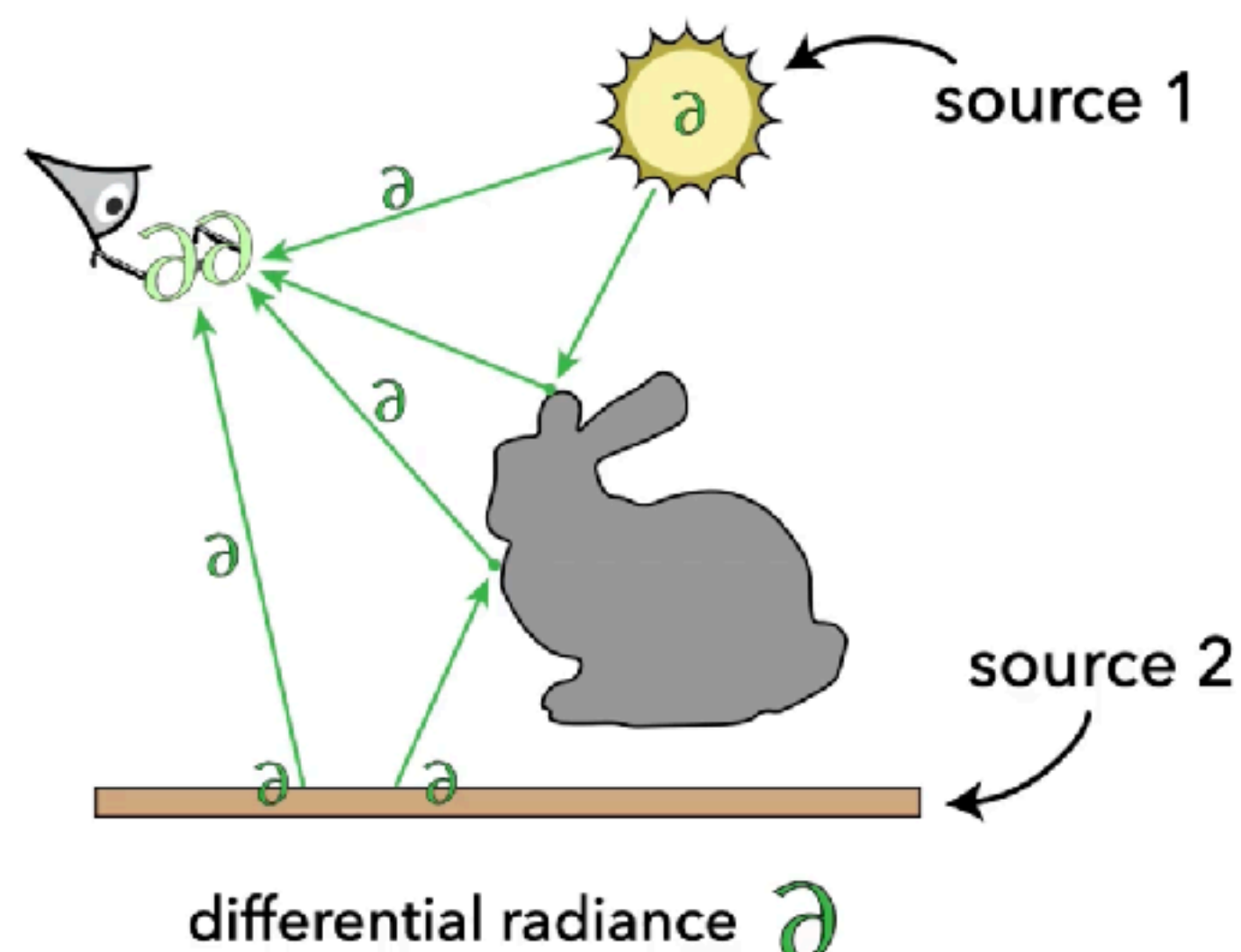
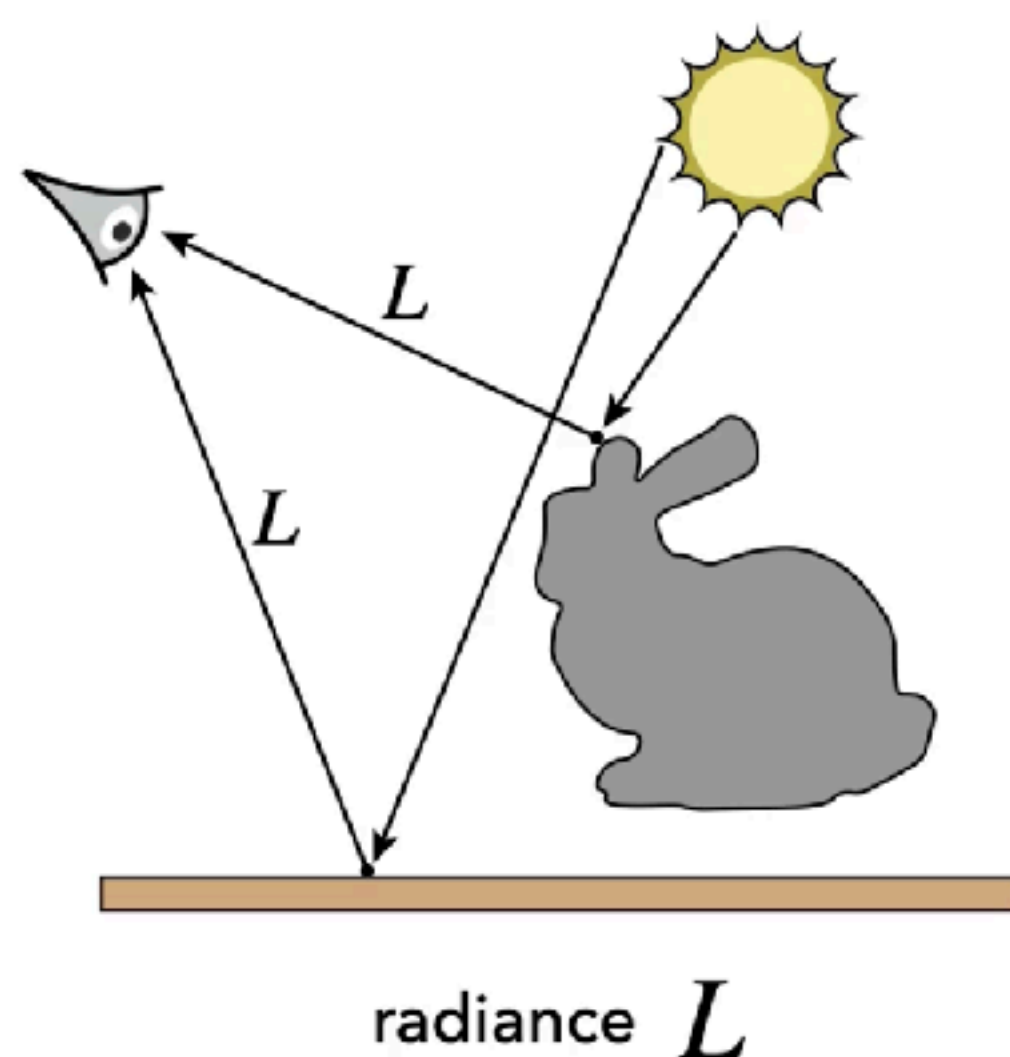
Radiative backprop:

A gradiensek a fényhez hasonlóan terjednek!

Inverz Grafika

Differenciálható sugárkövetés

Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering

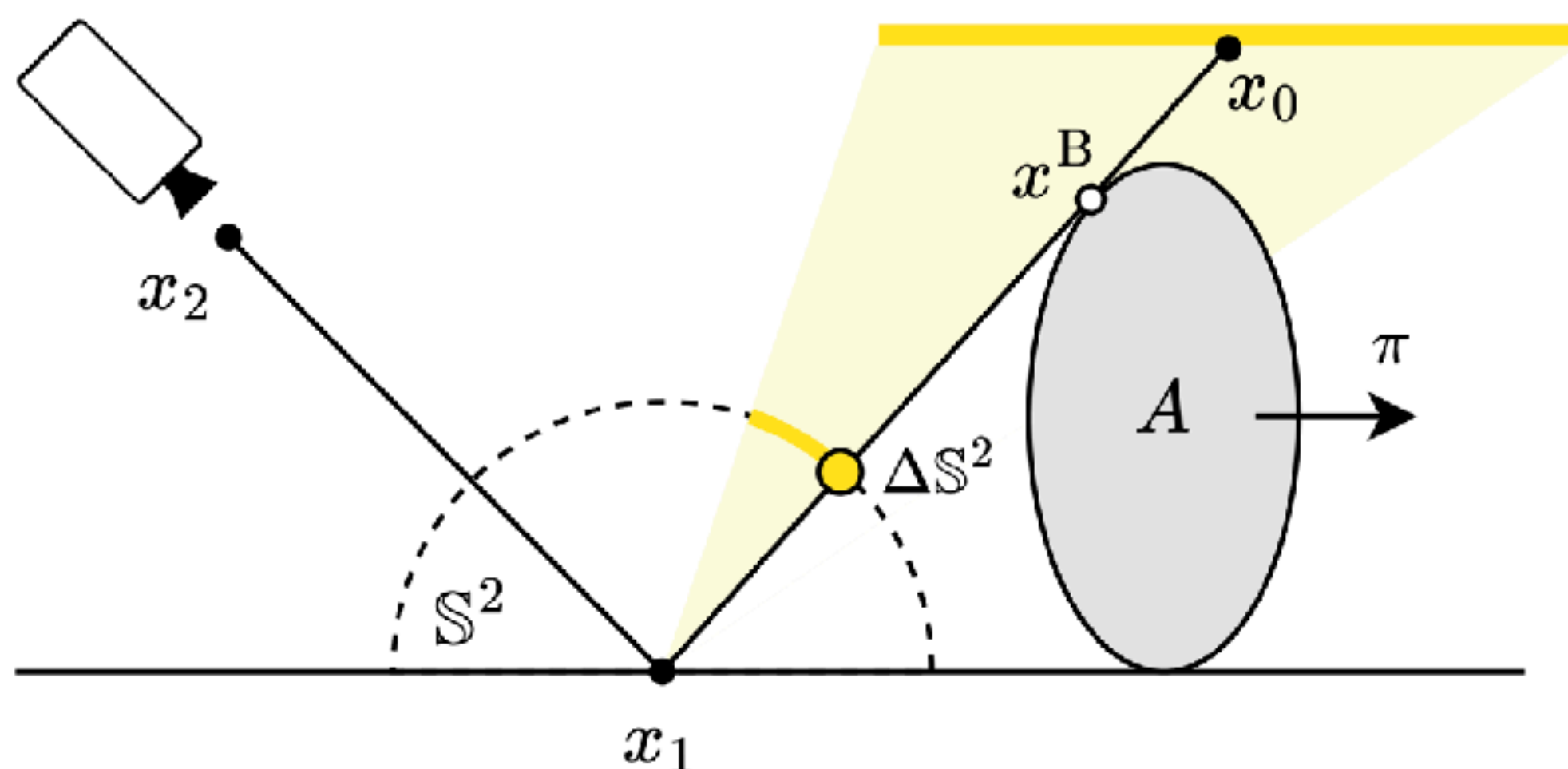
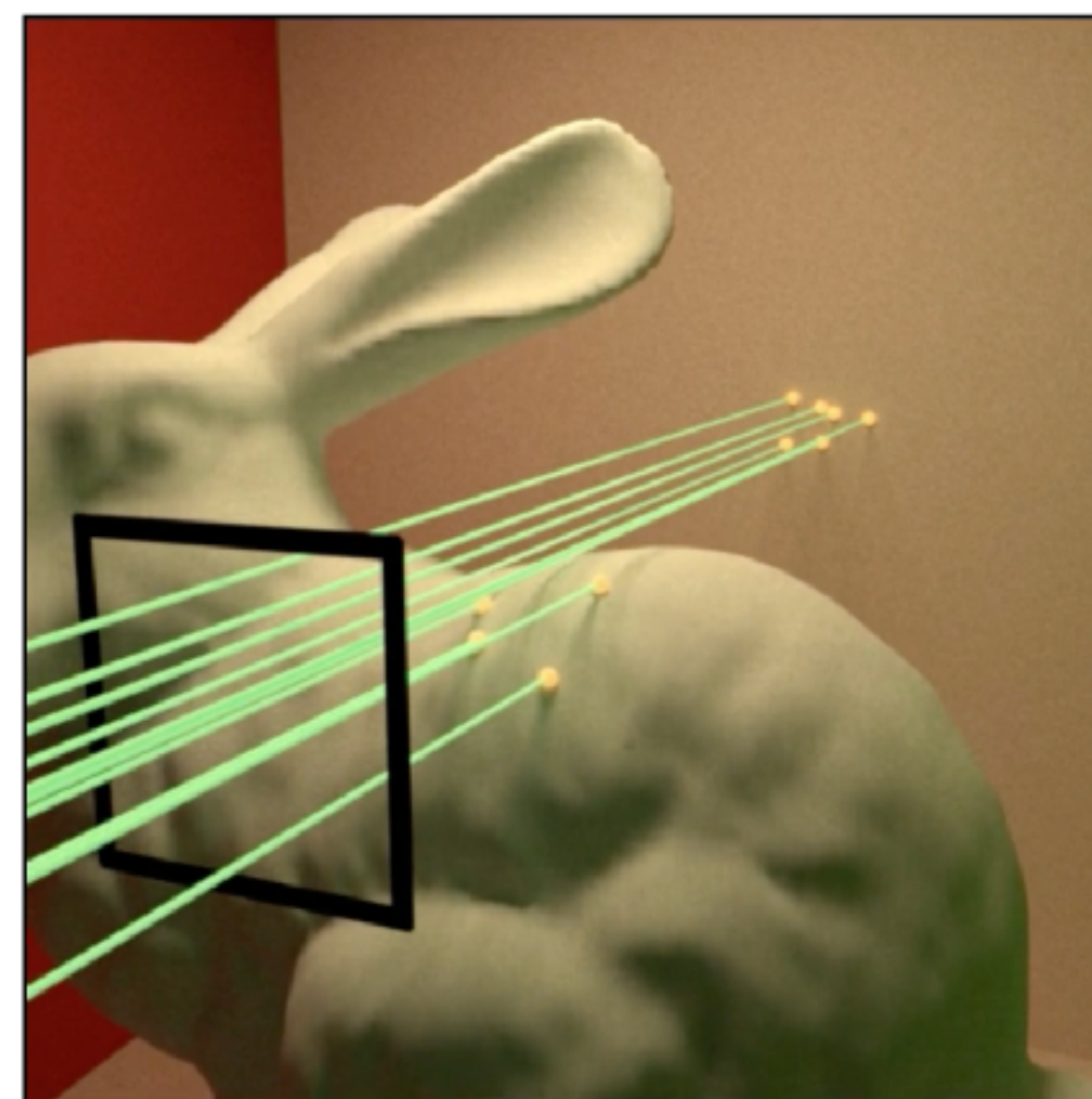


Radiative backprop:

A gradiensek a fényhez hasonlóan terjednek!

Inverz Grafika

Differenciálható sugárkövetés



$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\mathbb{S}^2} L_i(x, \omega_i) f_s(x, \omega_i, \omega_o) d\sigma(\omega_i)$$

$$\frac{\partial L_o}{\partial \pi} = \frac{\partial L_e}{\partial \pi} + \frac{\partial}{\partial \pi} \int_{\mathbb{S}^2} L_i f_s d\sigma(\omega_i)$$

Fundamentális probléma:
nem-folytonos kifejezések (pl. láthatóság/sziluettek)
integrálása & differenciálása!

Inverz Grafika

Differenciálható sugárkövetés*

Differentiation under the integral sign
Also known as the Leibniz integral rule

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) dx + f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi} + \sum_i (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}$$

Interior integral

Move derivative inside integral

Boundary terms

Account for changes in integration limits

Account for discontinuities of integrand that depend on π

Általános minta gradiensekre:
belső “árnyalási” integrál
+
határmenti “sziluett” integrál

Leibniz-szabály

(aka. Reynolds-féle transzporttétel, Cartan-féle “mágikus” formula, stb.)

$$f(x, \pi) = \begin{cases} 0 & \text{if } x < 2\pi \\ 1 & \text{if } x \geq 2\pi \end{cases}$$

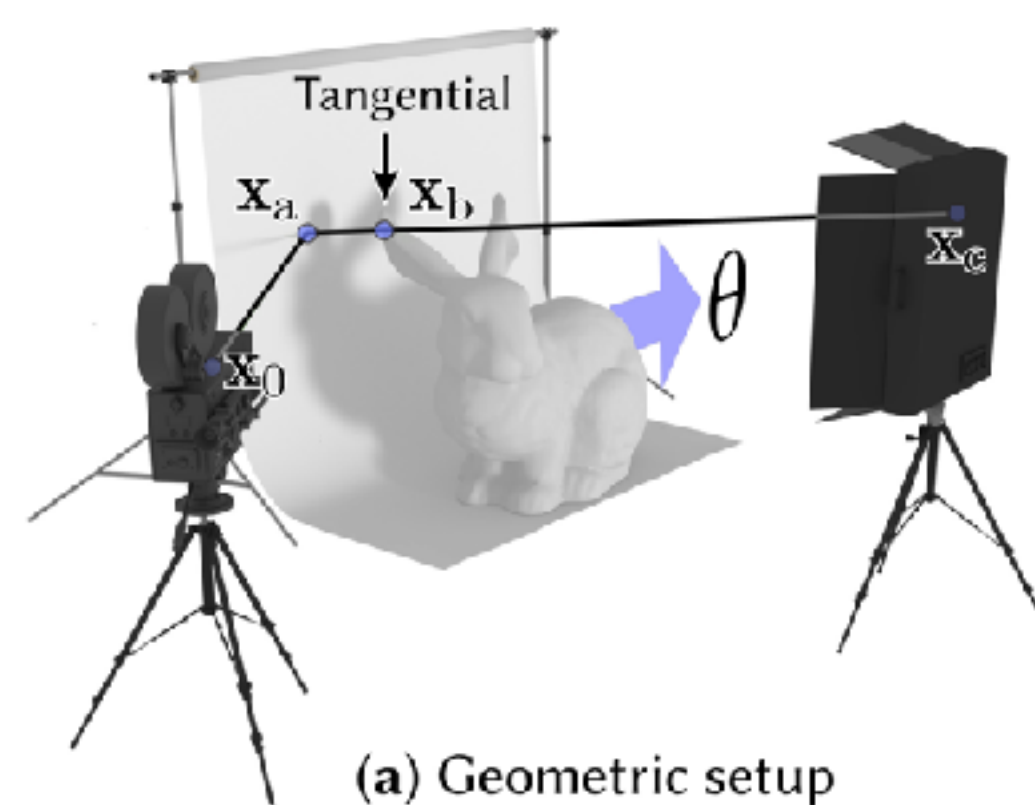
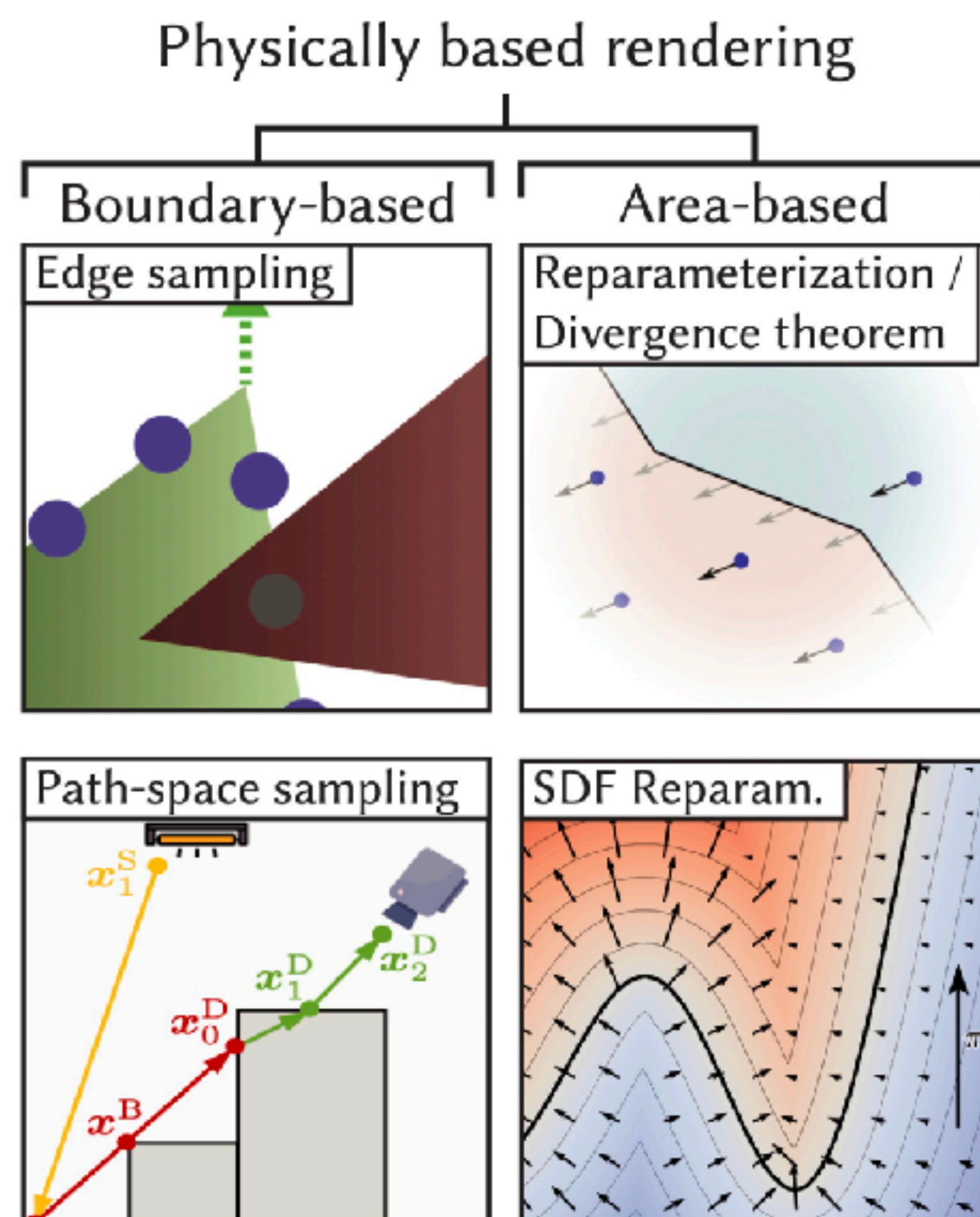
$$\frac{d}{d\pi} \int_0^{4\pi} f(x, \pi) dx = \int_0^{2\pi} \frac{d}{d\pi} 0 dx + \int_{2\pi}^{4\pi} \frac{d}{d\pi} 1 dx + 1 \frac{d(4\pi)}{d\pi} - 0 \frac{d0}{d\pi} + (0 - 1) \frac{d(2\pi)}{d\pi}$$

Account for changes in integration limits

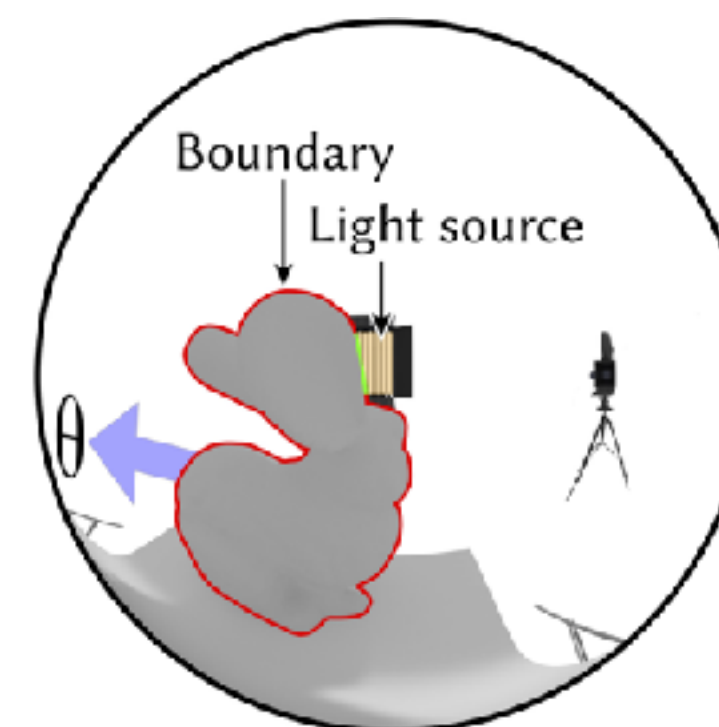
Account for discontinuities of integrand that depend on π

Inverz Grafika

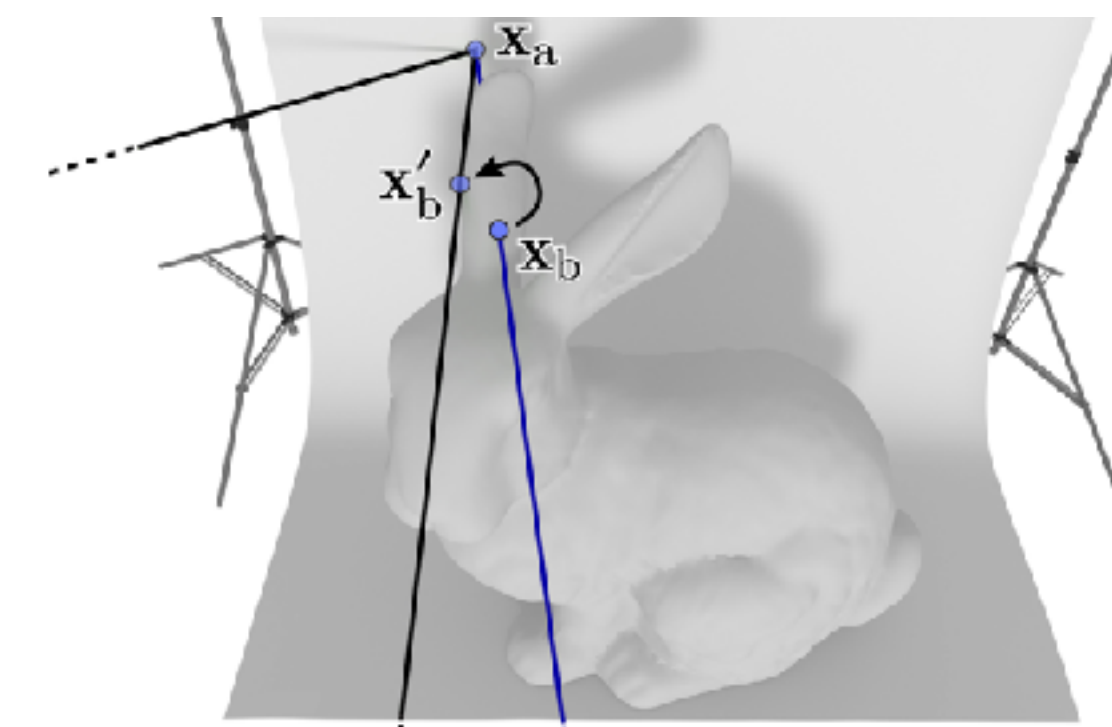
Differenciálható sugárkövetés*



(a) Geometric setup



(b) Discontinuous integral at x_a



(c) Projective sampling

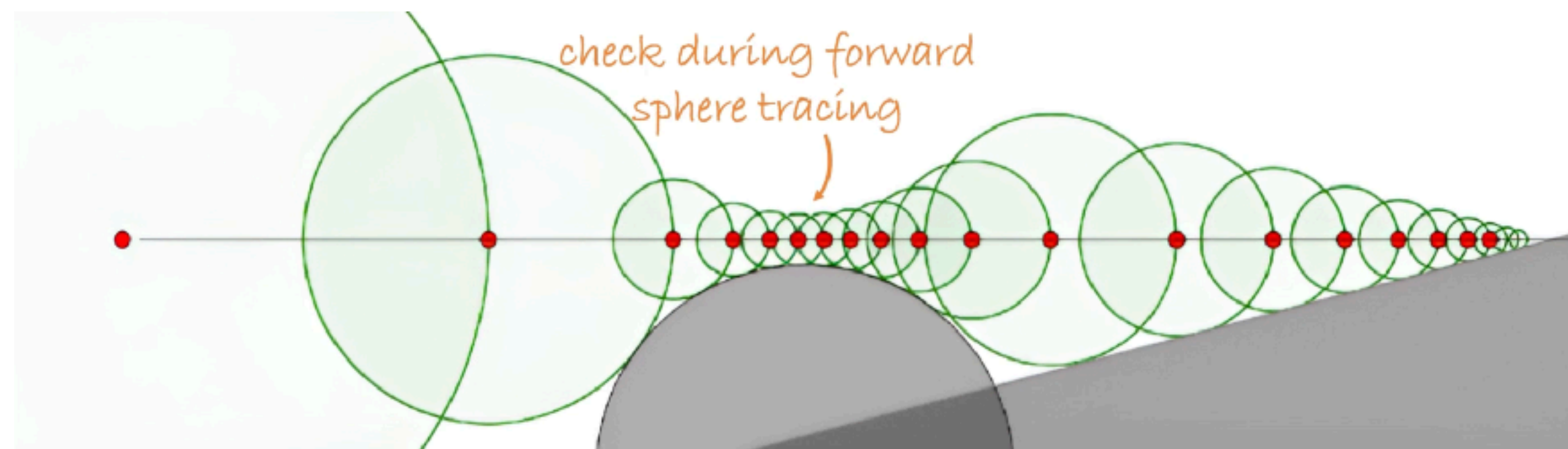
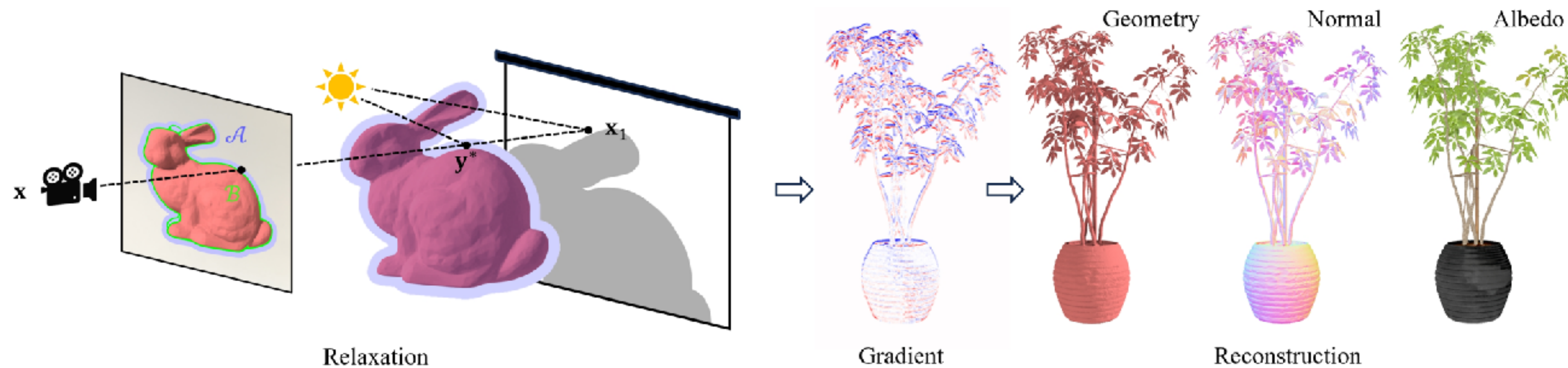
<https://rgl.epfl.ch/publications/Zhang2023Projective>

Számos különböző módszer láthatósági gradiensek becslésére...

Inverz Grafika

Differenciálható sugárkövetés — SDF reprezentáció*

A Simple Approach to Differentiable Rendering of SDFs



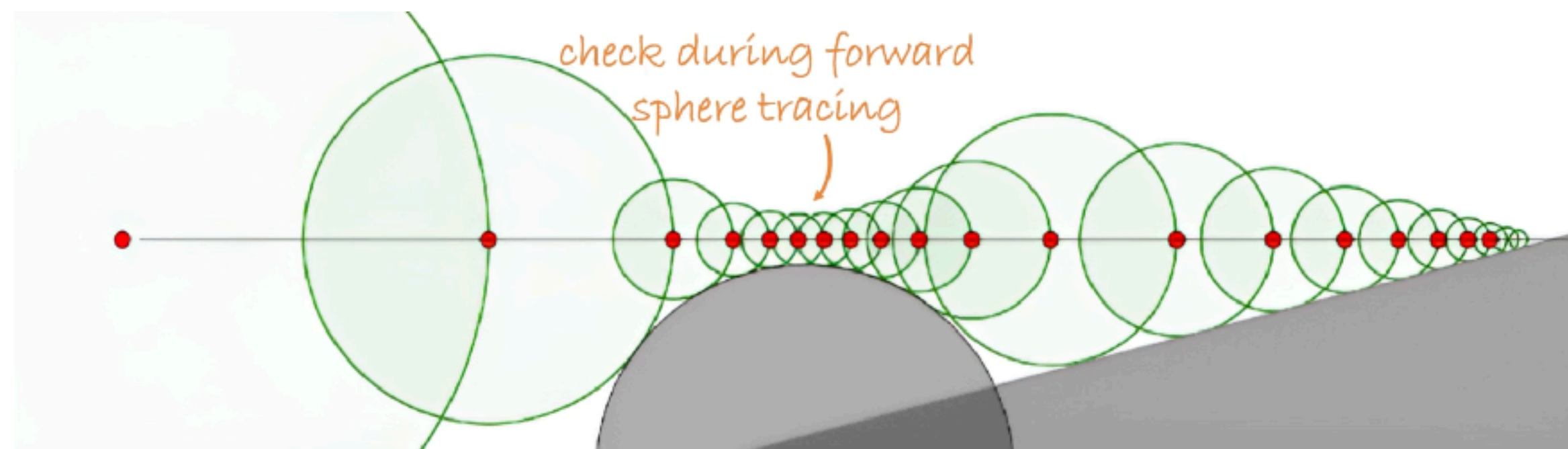
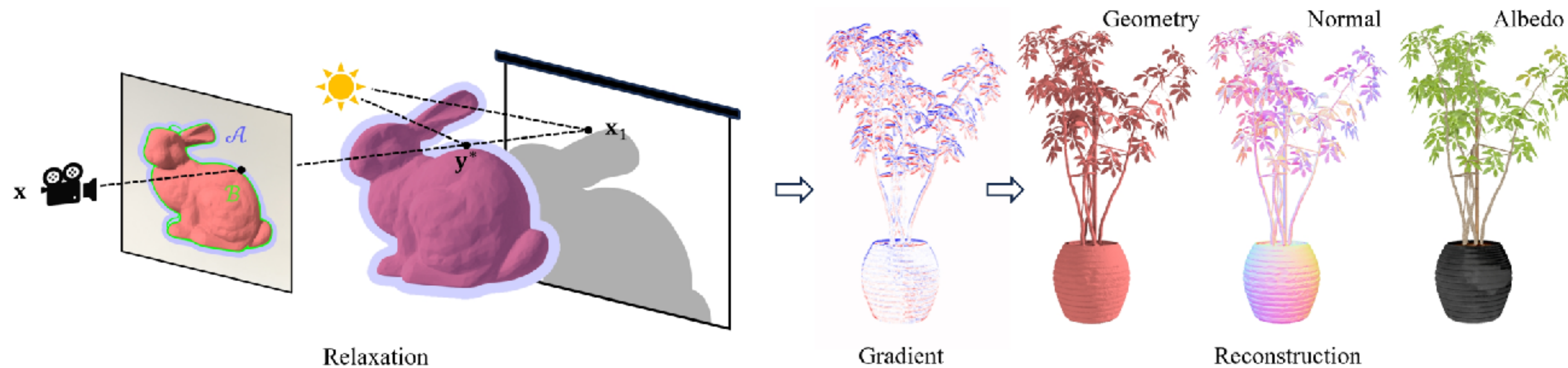
“Relaxált” sziluett =
távolság lokális minimumai sugarak mentén



Inverz Grafika

Differenciálható sugárkövetés — SDF reprezentáció*

A Simple Approach to Differentiable Rendering of SDFs



“Relaxált” sziluett =
távolság lokális minimumai sugarak mentén



Inverz Grafika

Differenciálható sugárkövetés — SDF reprezentáció



<https://dvicini.github.io/differentiable-sdf-rendering/>

Inverz Grafika

Differenciálható sugárkövetés — SDF reprezentáció



<https://dvicini.github.io/differentiable-sdf-rendering/>

Inverz Grafika

Differenciálható sugárkövetés — Könyvtárak



<https://drjit.readthedocs.io>

<https://mitsuba.readthedocs.io/>

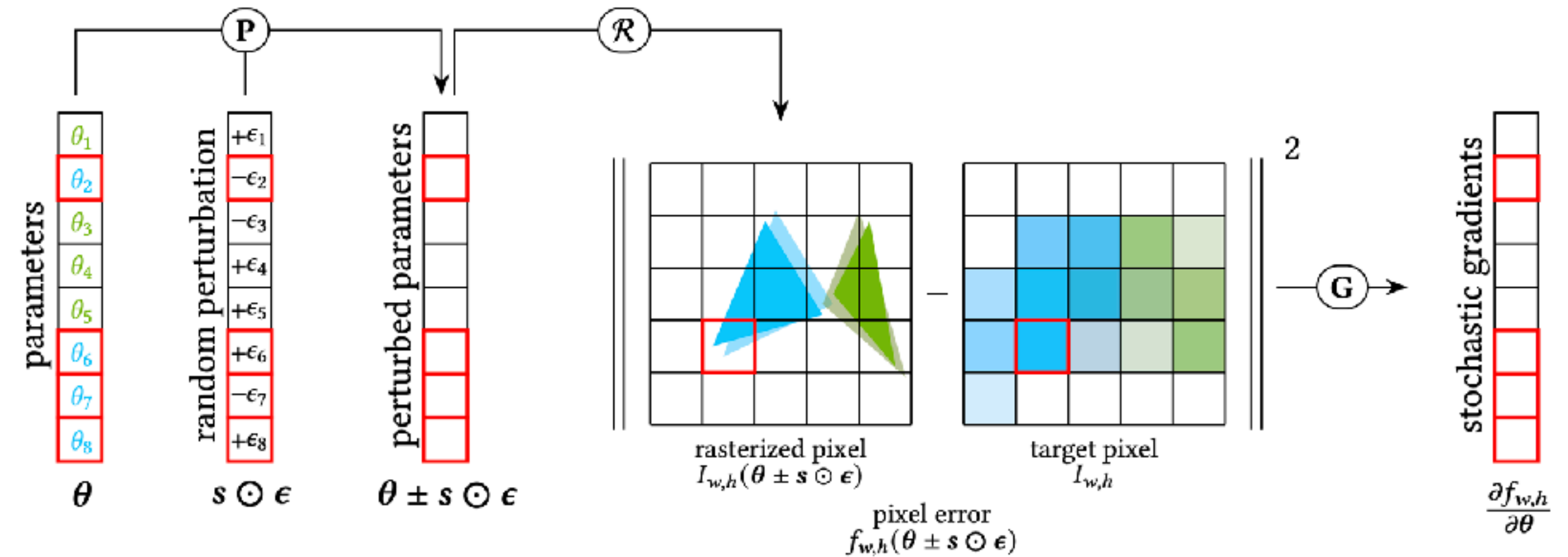
Inverz Grafika

Nem-differenciálható renderer használata

- Mit tegyünk, ha a renderer nem differenciálható és nem is fogjuk azzá tenni?
 - Pl. mert egy jól megírt renderert (pl. Unity/Unreal) szeretnénk használni...
- Ötlet: sztochasztikus gradiens becslés (numerikus véges differencia random perturbációkkal)
- Minden pixelre meghatározható mely paraméterek befolyásolják
- Minimális beavatkozás a motorba (3 db compute shader)

Transforming a Non-Differentiable Rasterizer into a Differentiable One with Stochastic Gradient Estimation

THOMAS DELIOT, ERIC HEITZ, and LAURENT BELCOUR, Intel Corporation, France



Algorithm 2 Compute shader G (gradient)

Require: thread ID (w, h)

```

load  $I_{w,h}(\theta + s \odot \epsilon), I_{w,h}(\theta - s \odot \epsilon), I_{w,h}$  ▷ load 3 float3 (3× rgb)
 $f_{w,h}(\theta + s \odot \epsilon) = \|I_{w,h} - I_{w,h}(\theta + s \odot \epsilon)\|^2$ 
 $f_{w,h}(\theta - s \odot \epsilon) = \|I_{w,h} - I_{w,h}(\theta - s \odot \epsilon)\|^2$ 
for each parameter  $\theta_i$  contributing to pixel  $(w, h)$  do ▷ implementation of Equation (8)
    load  $s_i \epsilon_i$  ▷ load 1 float
    AtomicAdd $\left(\frac{\partial f}{\partial \theta_i} \leftarrow \frac{f_{w,h}(\theta + s \odot \epsilon) - f_{w,h}(\theta - s \odot \epsilon)}{2 s_i \epsilon_i}\right)$  ▷ atomic add 1 float
end for
    
```

Inverz Grafika

Nem-differenciálható renderer használata

Transforming a Non-Differentiable Rasterizer into a Differentiable One with Stochastic Gradient Estimation

THOMAS DELIOT, ERIC HEITZ, and LAURENT BELCOUR, Intel Corporation, France

EWER

Subdivision surface

Optimizing

- + control mesh
- + normals
- + albedo
- + metallic
- + roughness
- + heightmap



Inverz Grafika

Nem-differenciálható renderer használata

Transforming a Non-Differentiable Rasterizer into a Differentiable One with Stochastic Gradient Estimation

THOMAS DELIOT, ERIC HEITZ, and LAURENT BELCOUR, Intel Corporation, France

EWER

Subdivision surface

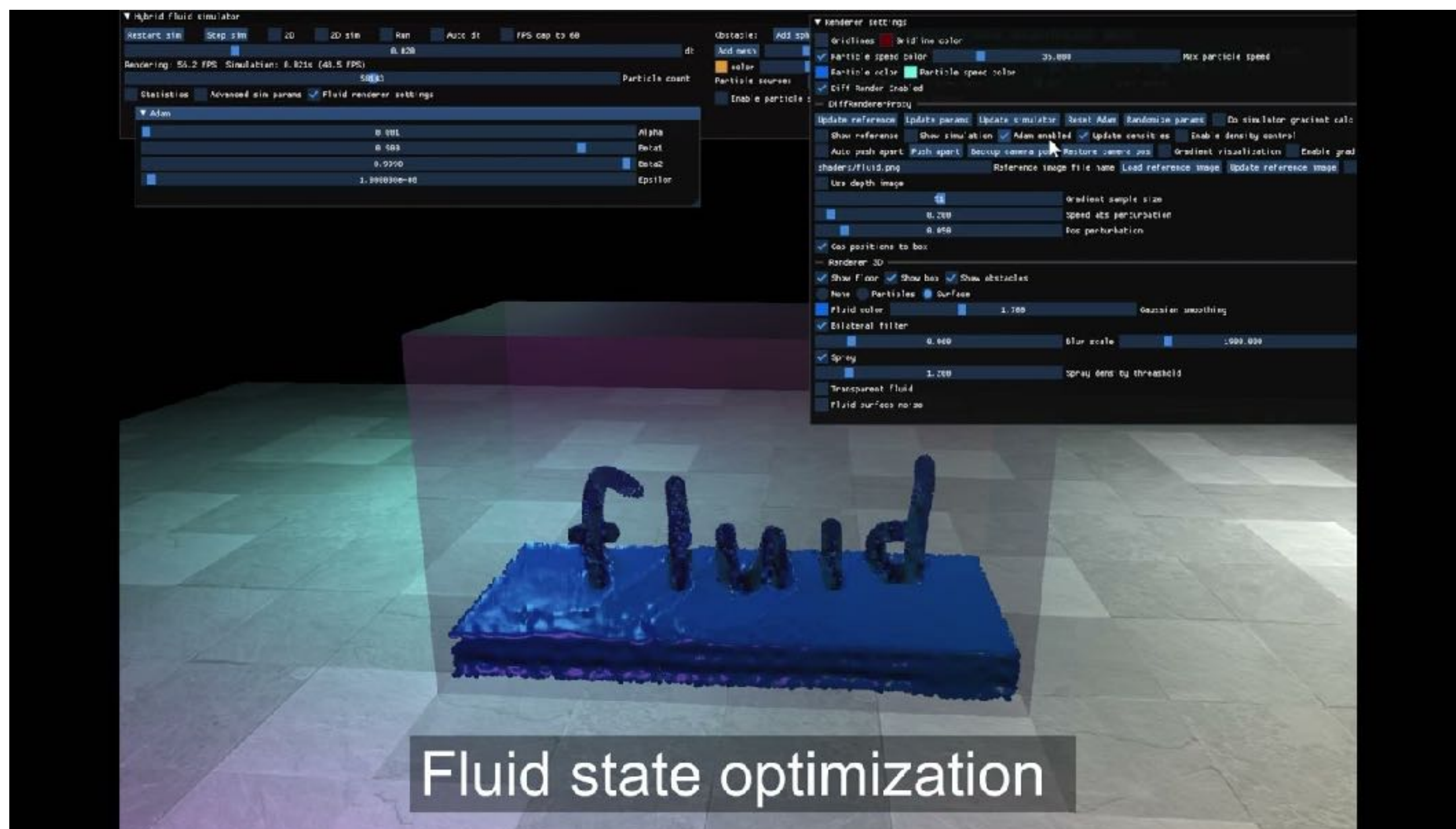
Optimizing

- + control mesh
- + normals
- + albedo
- + metallic
- + roughness
- + heightmap



Inverz Grafika

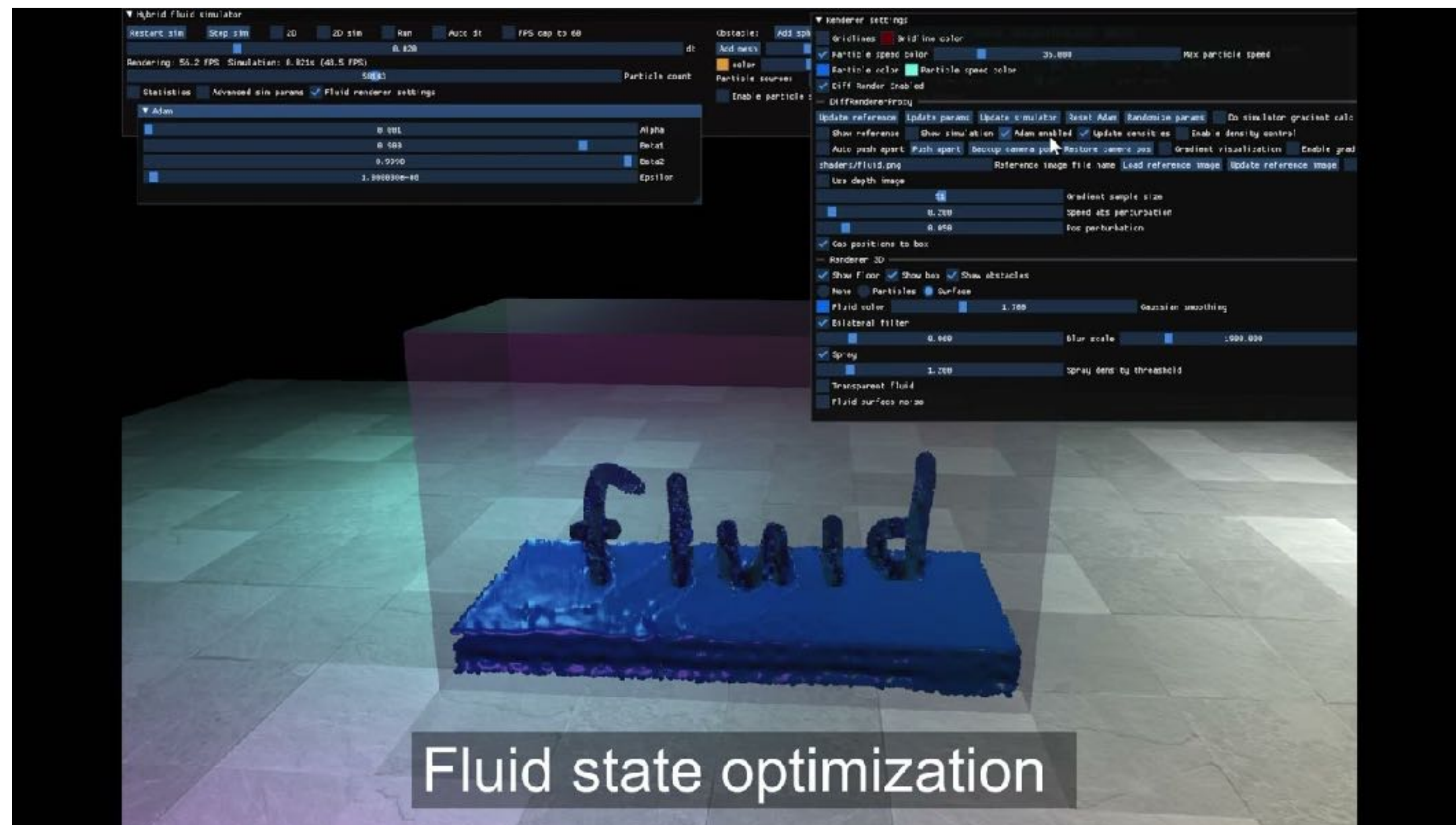
Nem-differenciálható renderer használata



Forrás: Kiovics Dániel [TDK]

Inverz Grafika

Nem-differenciálható renderer használata



Forrás: Kiovics Dániel [TDK]

Inverz Grafika

Geometria optimalizációja

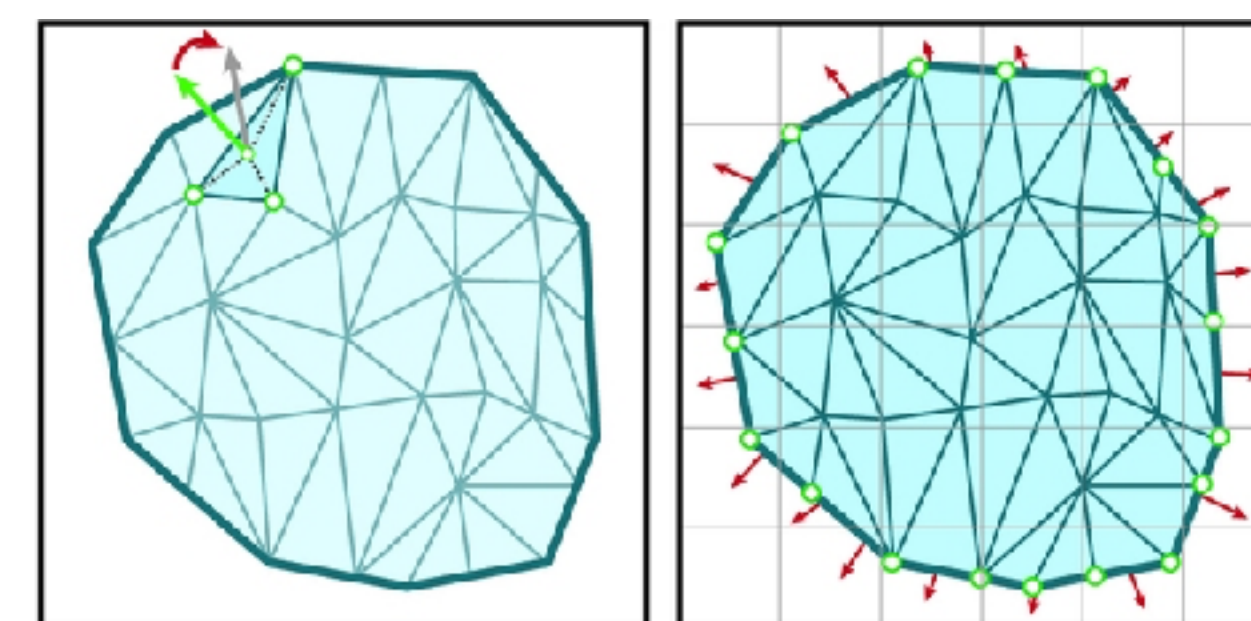
- Felületi (pl. mesh vertex) pozíciók gradiense = felületi “sebesség” vektormező
- Probléma: a gradiens a sziluettek környékén koncentrálódik...
- Megoldás: gradiens mező elsimítása (átlagolás, Laplace szűrés)
 - Aka “Sobolev prekondicionálás”
- Mesh felbontás fokozatos növelése — lokális minimumok elkerülése

Large Steps in Inverse Rendering of Geometry

BAPTISTE NICOLET, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ALEC JACOBSON, University of Toronto, Canada

WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland



(a) Shading gradients

(b) Silhouette gradients

Equal time



Regularized



Naive

Target:



Inverz Grafika

Geometria optimalizációja

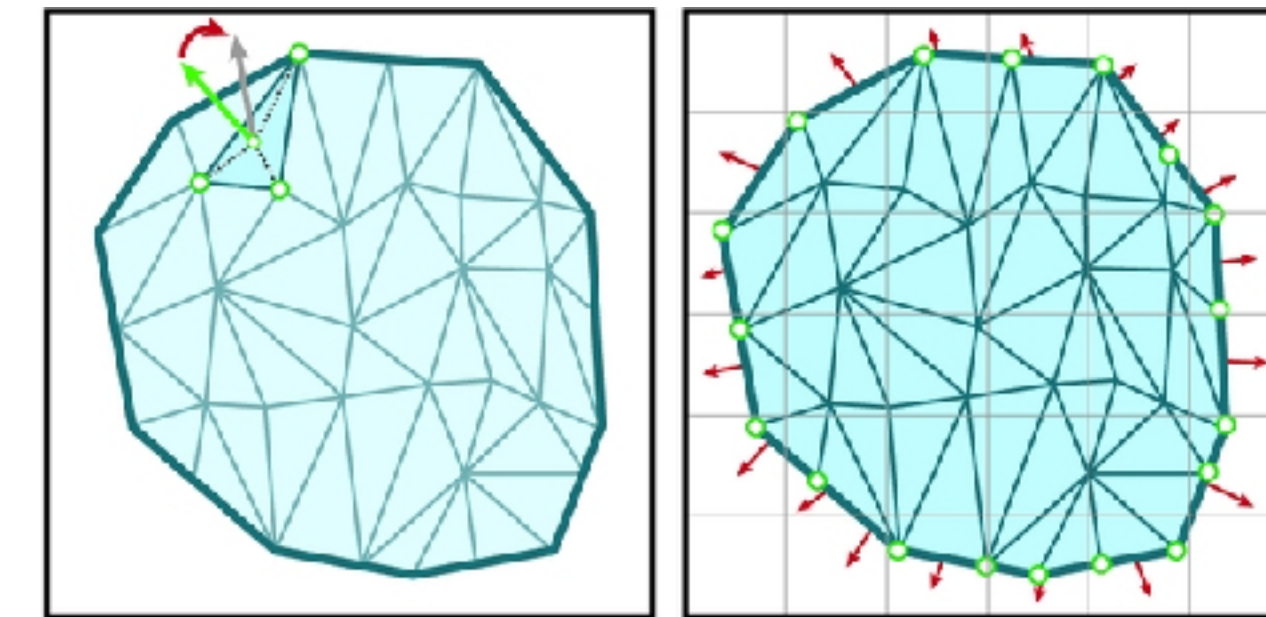
- Felületi (pl. mesh vertex) pozíciók gradiense = felületi “sebesség” vektormező
- Probléma: a gradiens a sziluettek környékén koncentrálódik...
- Megoldás: gradiens mező elsimítása (átlagolás, Laplace szűrés)
 - Aka “Sobolev prekondicionálás”
- Mesh felbontás fokozatos növelése — lokális minimumok elkerülése

Large Steps in Inverse Rendering of Geometry

BAPTISTE NICOLET, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ALEC JACOBSON, University of Toronto, Canada

WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland



(a) Shading gradients

(b) Silhouette gradients

Equal time



Regularized

Target:



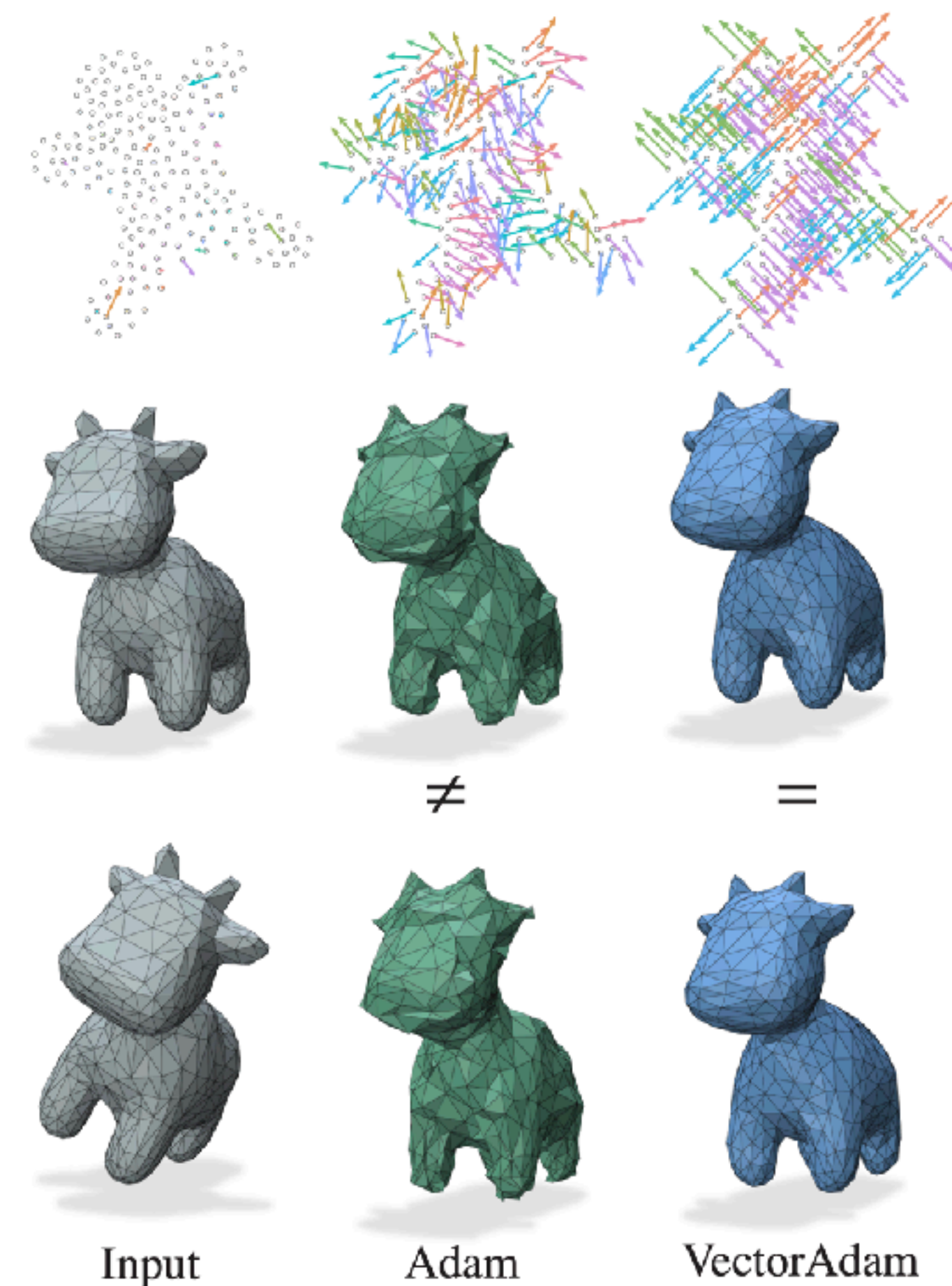
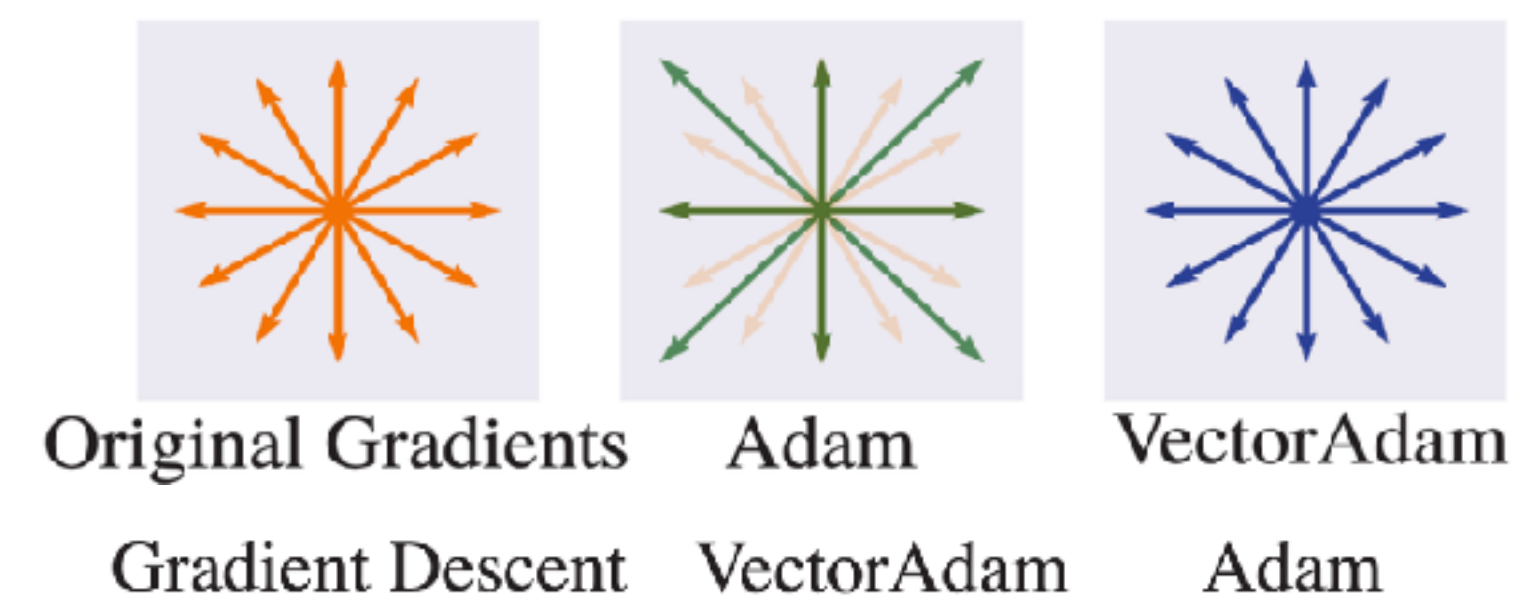
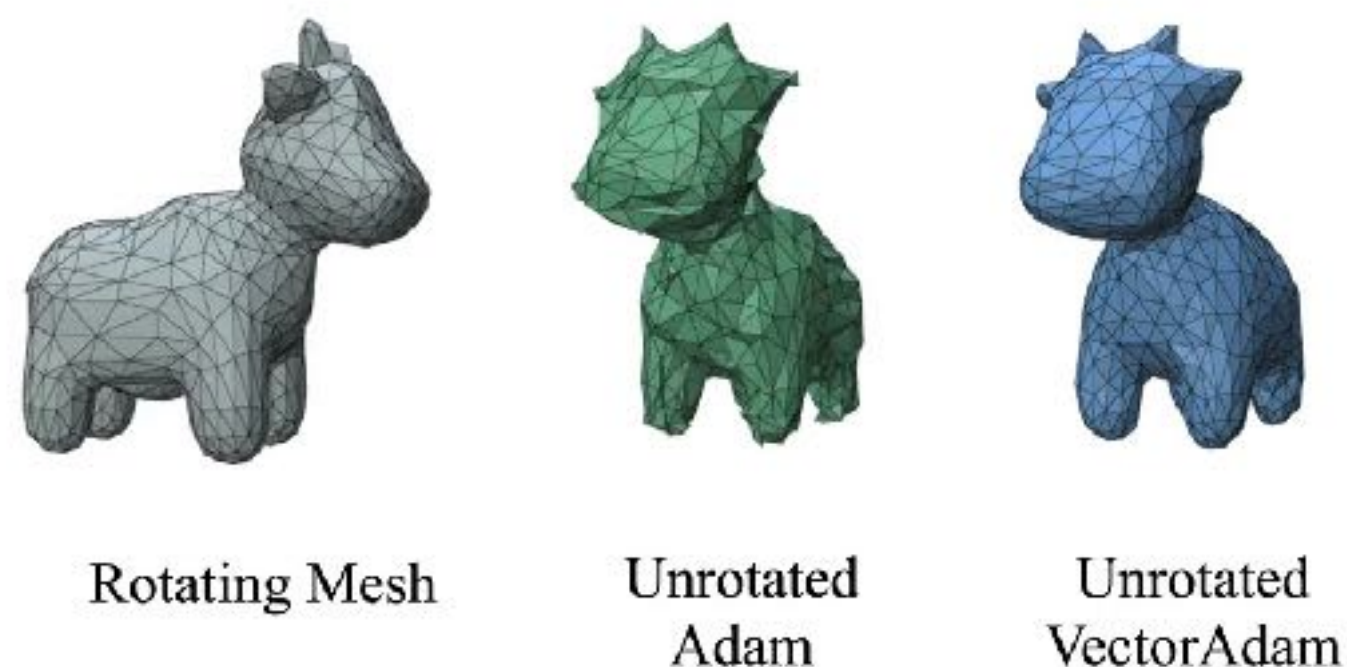
Naive

Inverz Grafika

Geometria optimalizációja — VectorAdam

- Az Adam a gradiens minden koordinátáját ugyanúgy normalizálja — koordináta-függő “diagonális” skálázás!
- VectorAdam: koordinátarendszertől független!

Equivariance and Optimization

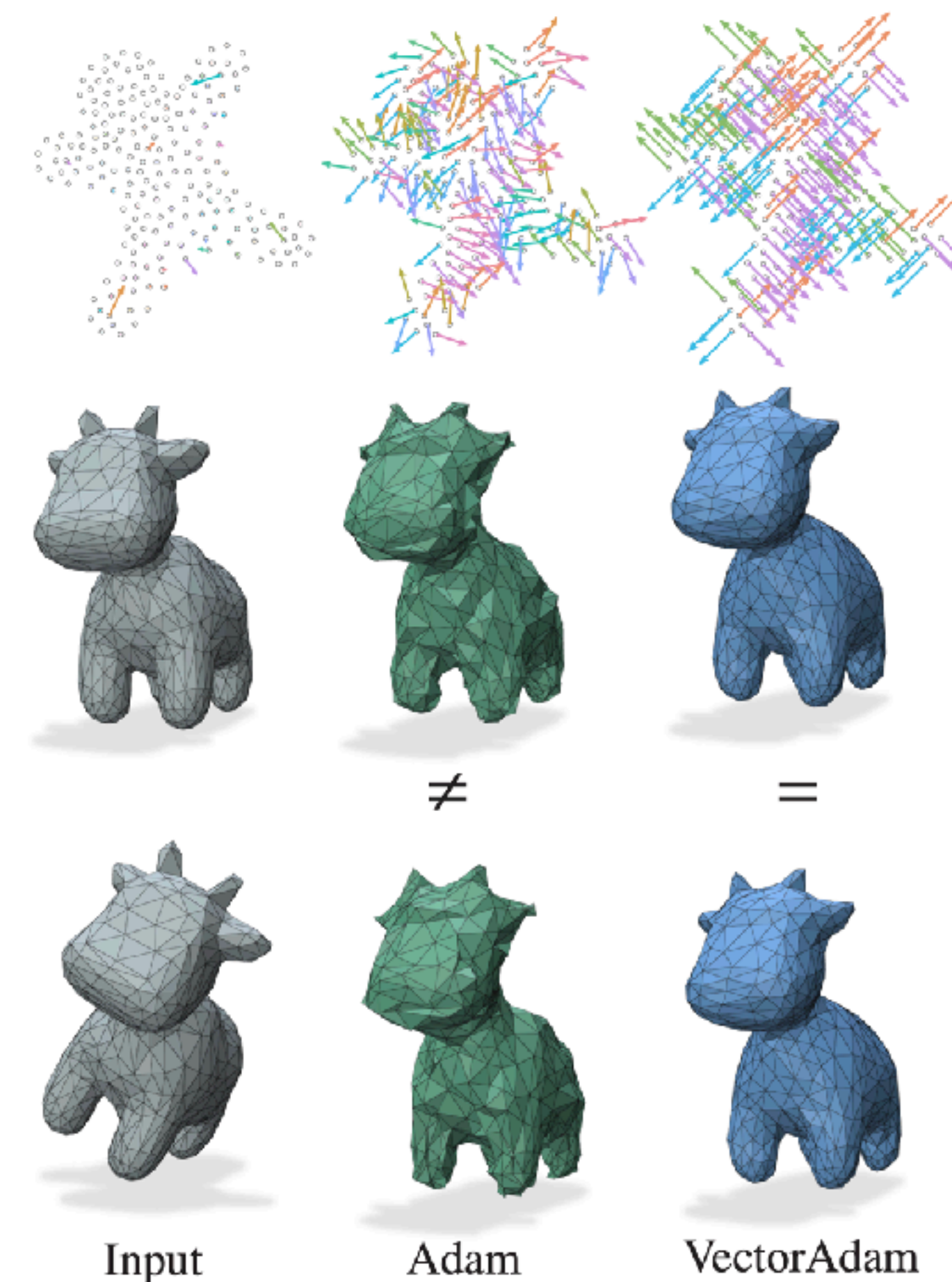
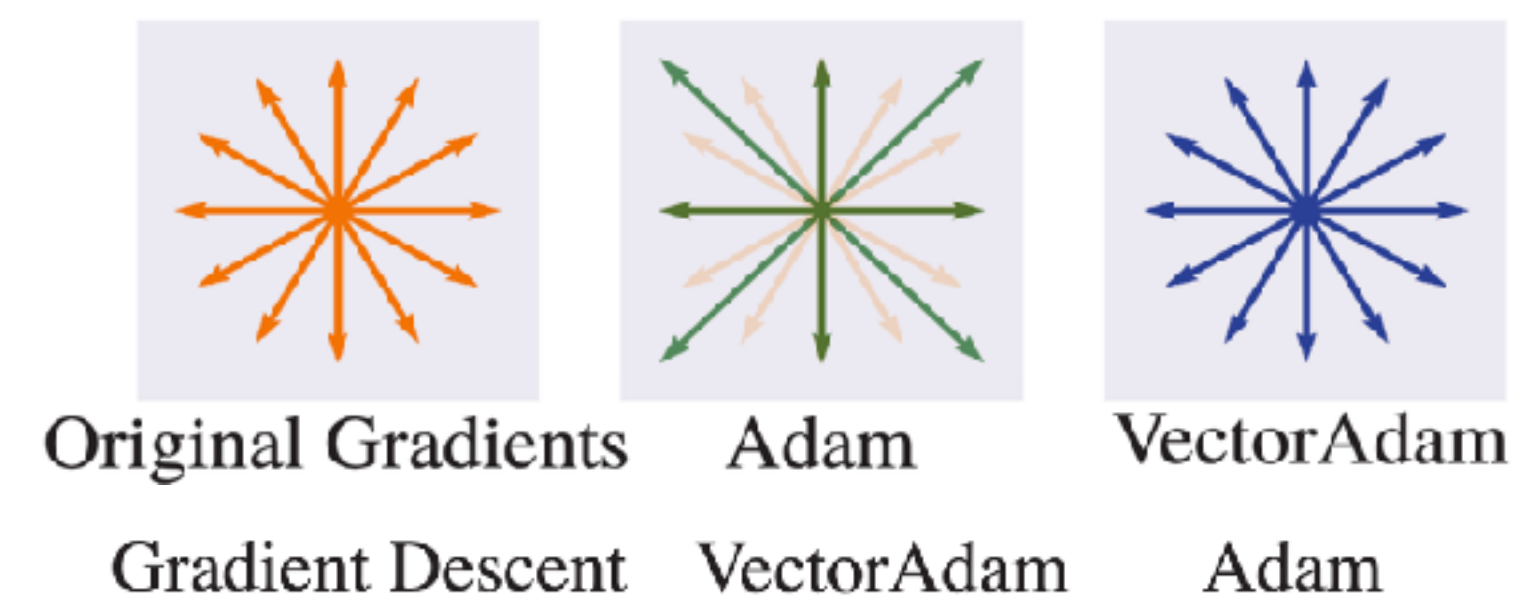
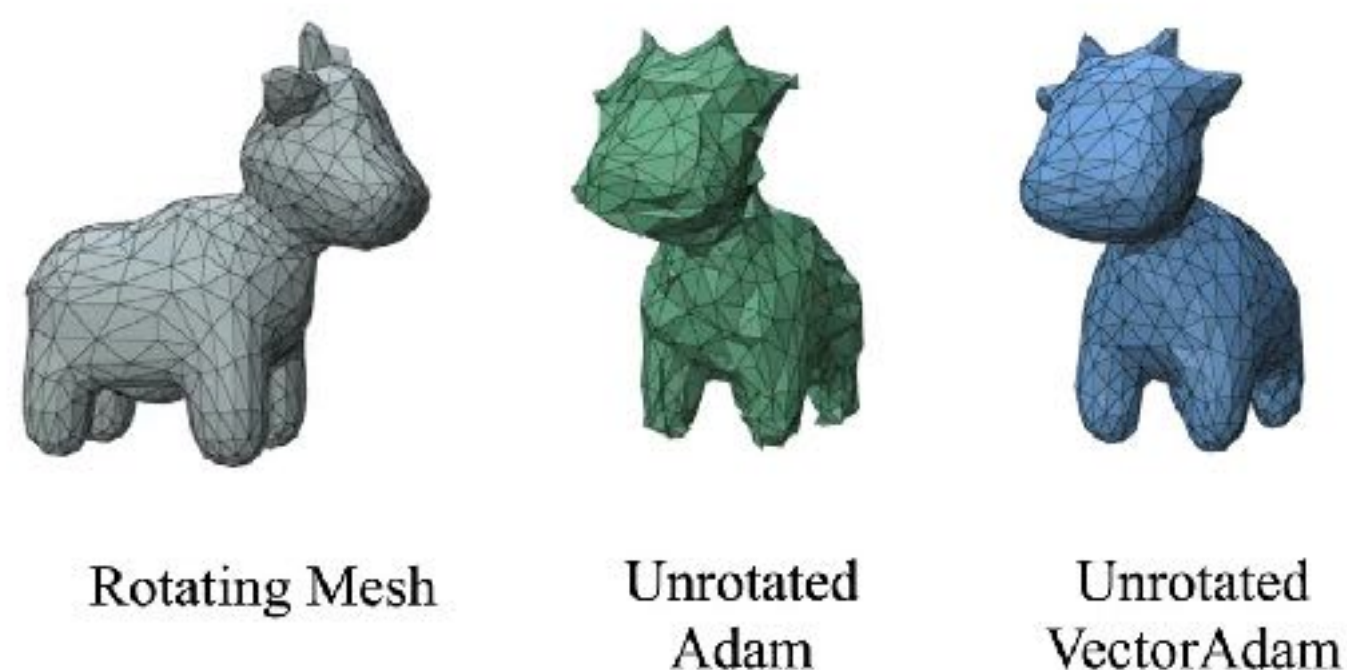


Inverz Grafika

Geometria optimalizációja — VectorAdam

- Az Adam a gradiens minden koordinátáját ugyanúgy normalizálja — koordináta-függő “diagonális” skálázás!
- VectorAdam: koordinátarendszertől független!

Equivariance and Optimization



Inverz Grafika

Implicit felület differenciálása

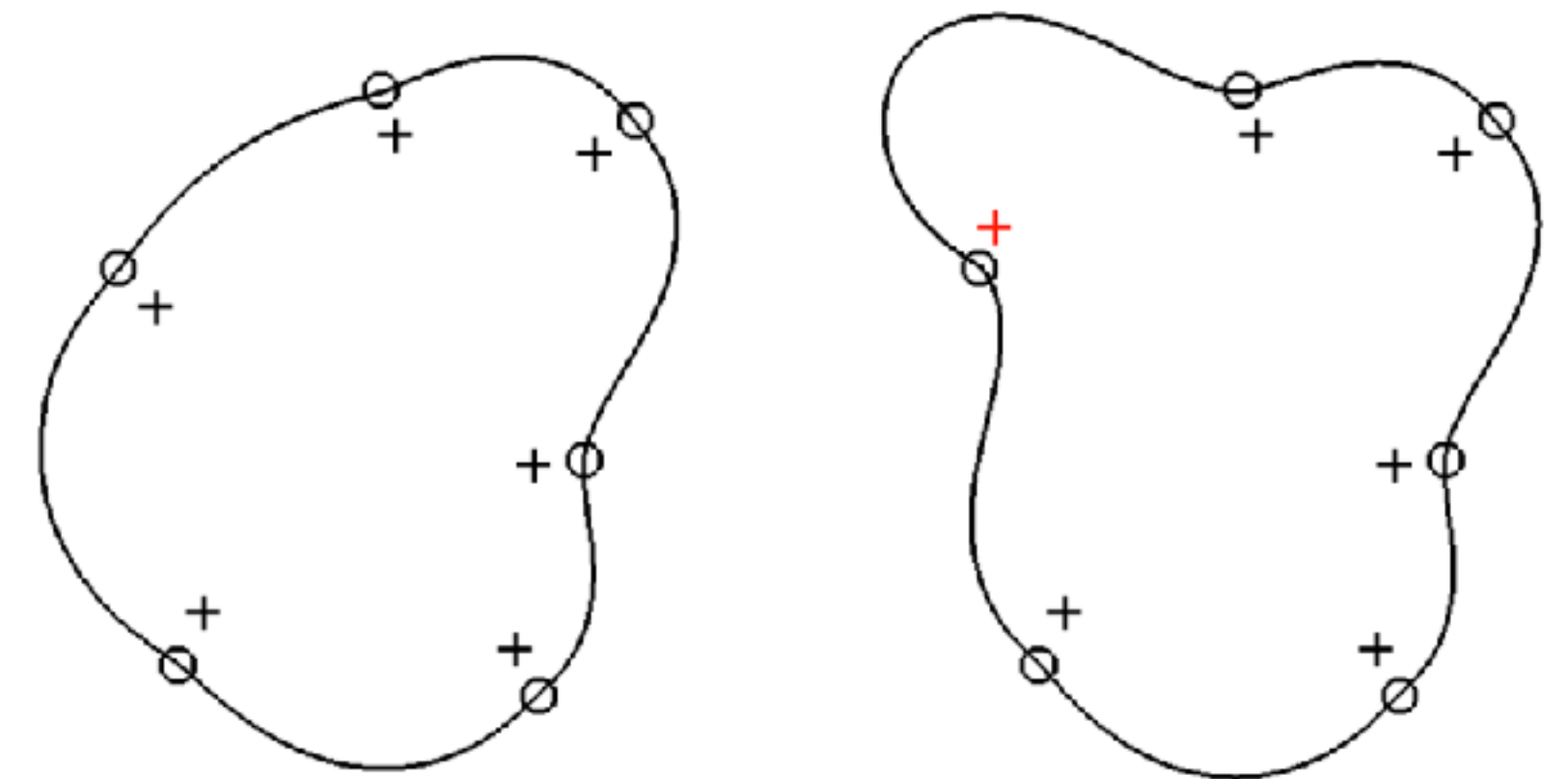
- Időben változó implicit szintfelületre:

$$\frac{dF(x; \theta)}{dt} = \frac{\partial F}{\partial x} \cdot \frac{\partial x}{\partial t} + \frac{\partial F}{\partial \theta} \cdot \frac{\partial \theta}{\partial t} = 0$$

Nemlineáris egyenletrendszer!

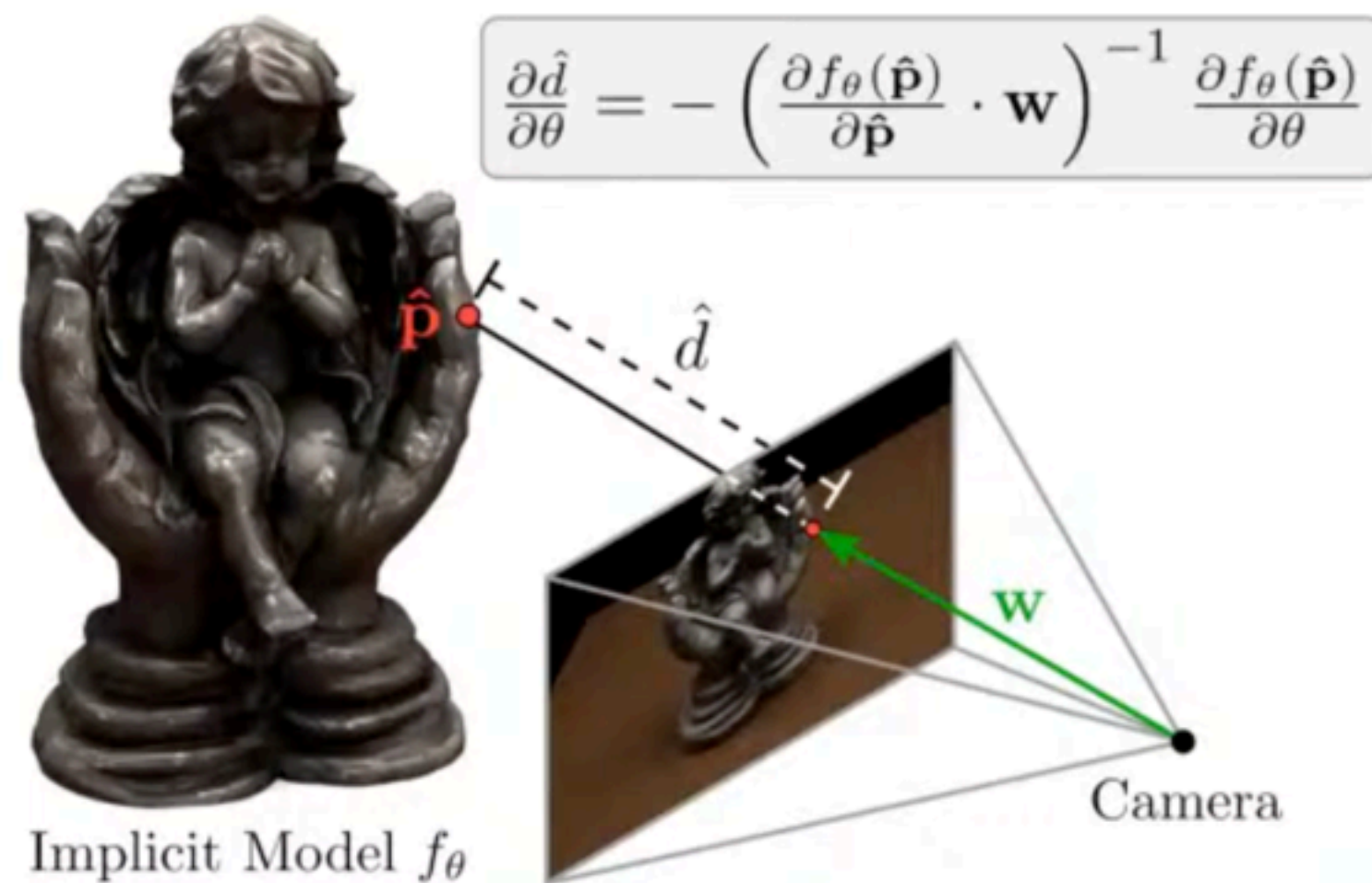
Megoldás $\frac{\partial x}{\partial t}$ -re: x merre mozduljon el, hogy az $F(x; \theta) = C$ implicit felületen maradjon

Megoldás $\frac{\partial \theta}{\partial t}$ -re: $F(x; \theta)$ függvény paraméterei hogyan változzanak, hogy $F(x; \theta) = C$ átmenjen x -en



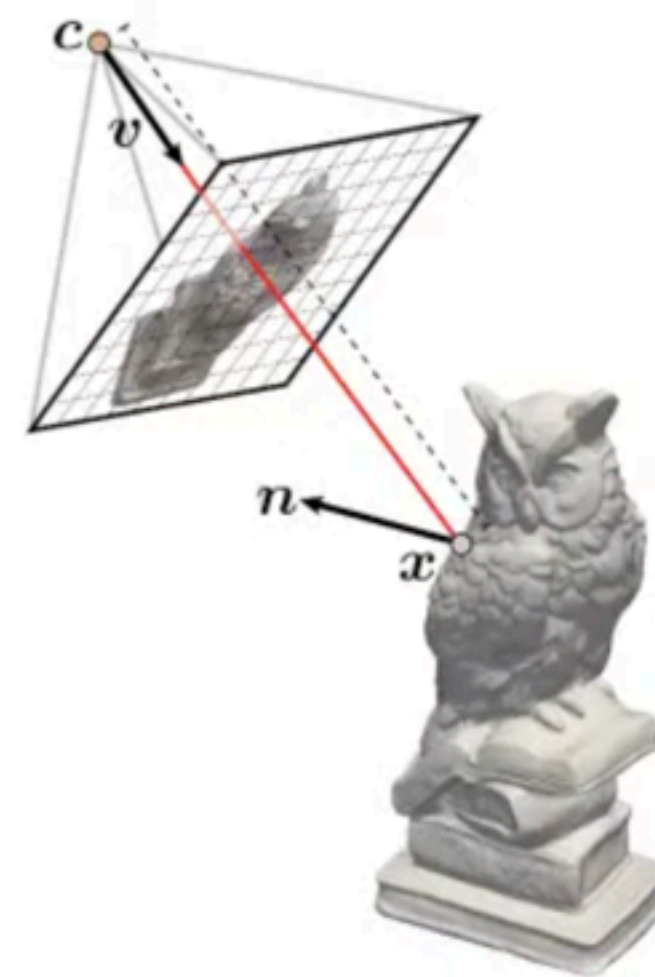
Inverz Grafika

Implicit felület differenciálható renderelése — Sugárkövetés?



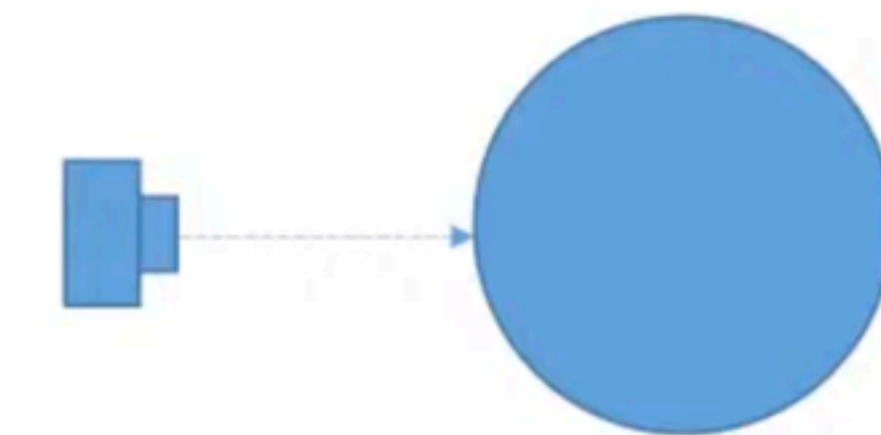
Niemeyer, Michael, et al. "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

DVR



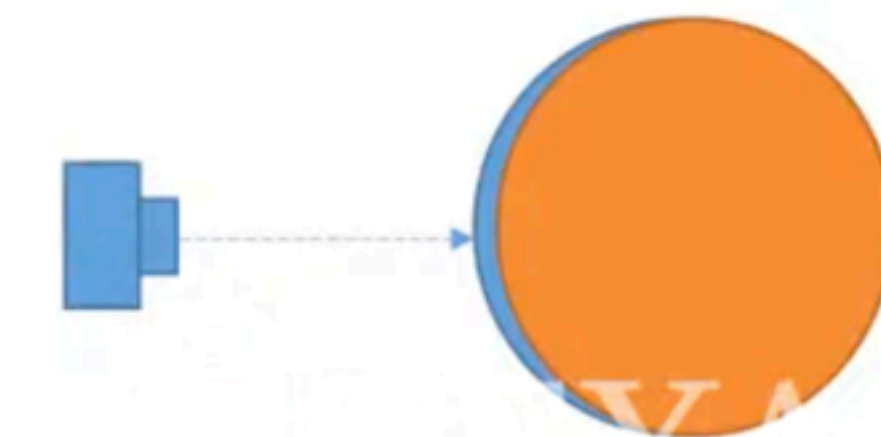
Yariv, Lior, et al. "Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance." Advances in Neural Information Processing Systems 33 (2020).

IDR



1. Sugár-felület metszés

Egy sugáron keresztül csak egyetlen metszéspontba terjed a hiba gradiense!

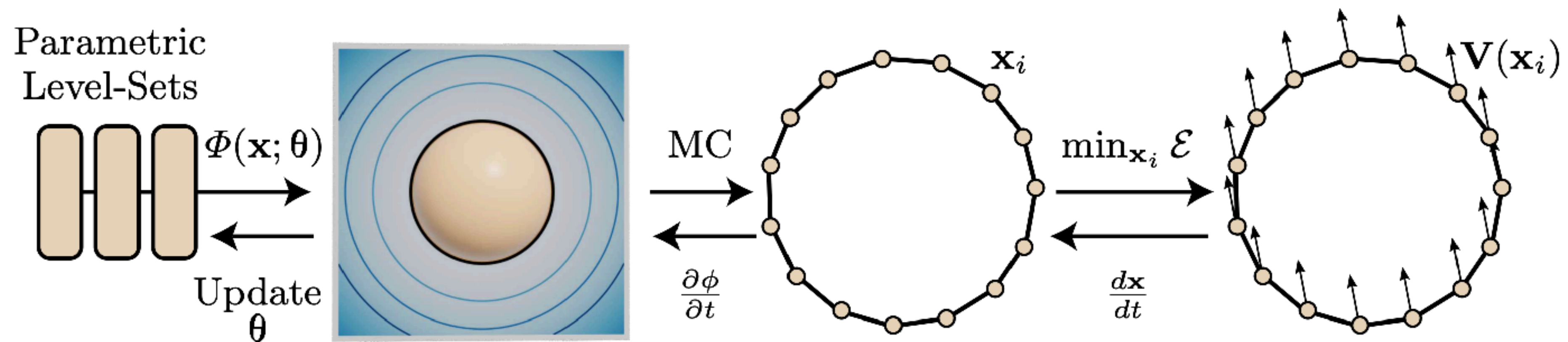


2. Felület optimalizációja

Könnyen lokális optimumban ragadunk...

Inverz Grafika

Implicit felület differenciálható renderelése — Explicit konverzió



Sugárkövetés helyett: implicit felület konverziója explicit reprezentációba (pl. Marching Cubes) majd az explicit gradiensek visszaterjesztése az implicit reprezentációra!

Inverz Grafika

Implicit felület differenciálható renderelése

A Level Set Theory for Neural Implicit Evolution
under Explicit Flows

Ishit Mehta

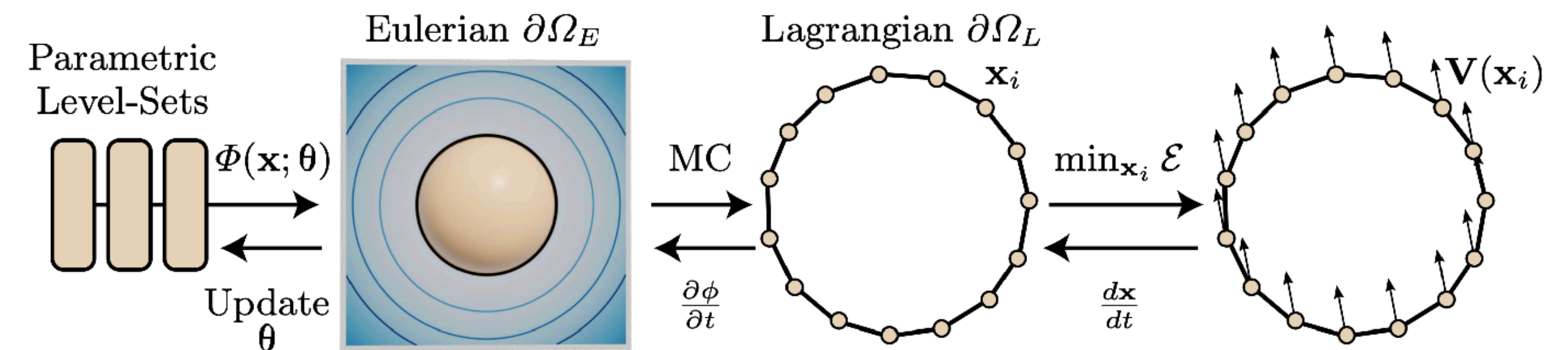
Manmohan Chandraker

Ravi Ramamoorthi

- Korrekt formula a mesh reprezentáción adott $\mathbf{V}$ vektormező visszaterjesztésére a $\phi(\mathbf{x})$ implicit reprezentációra (“**level set equation**”):

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \mathbf{V}$$

- A mozgásnak csak a szintfelületre merőleges komponense érdekes!
- $\mathbf{V}$ lehet mesh-alapú (pl. rendering) loss gradients, vagy akár manuálisan editált “sebességmező”...



<https://ishit.github.io/nie/>

Inverz Grafika

Implicit felület differenciálható renderelése

A Level Set Theory for Neural Implicit Evolution
under Explicit Flows

Ishit Mehta

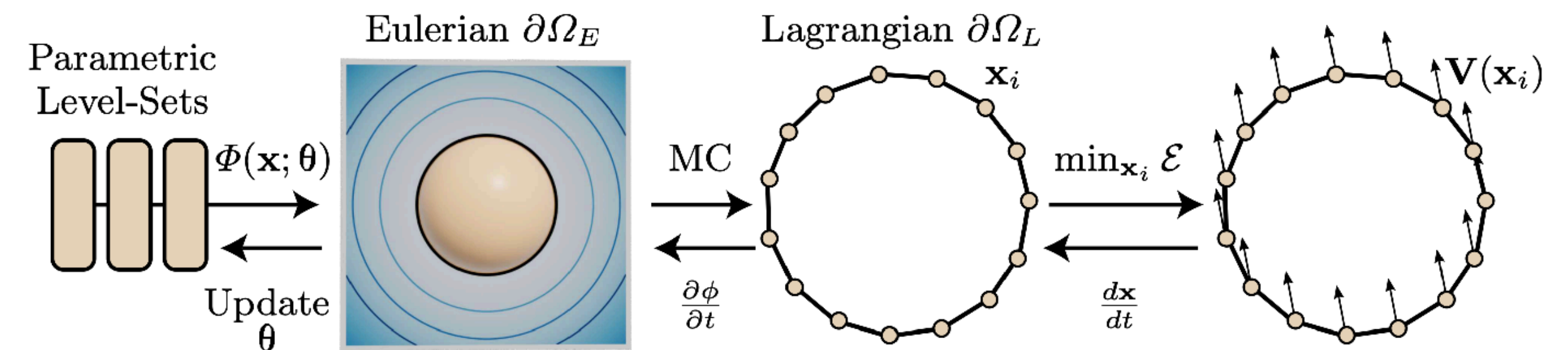
Manmohan Chandraker

Ravi Ramamoorthi

- Korrekt formula a mesh reprezentáción adott $\mathbf{V}$ vektormező visszaterjesztésére a $\phi(\mathbf{x})$ implicit reprezentációra (“**level set equation**”):

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \mathbf{V}$$

- A mozgásnak csak a szintfelületre merőleges komponense érdekes!
- $\mathbf{V}$ lehet mesh-alapú (pl. rendering) loss gradients, vagy akár manuálisan editált “sebességmező”...

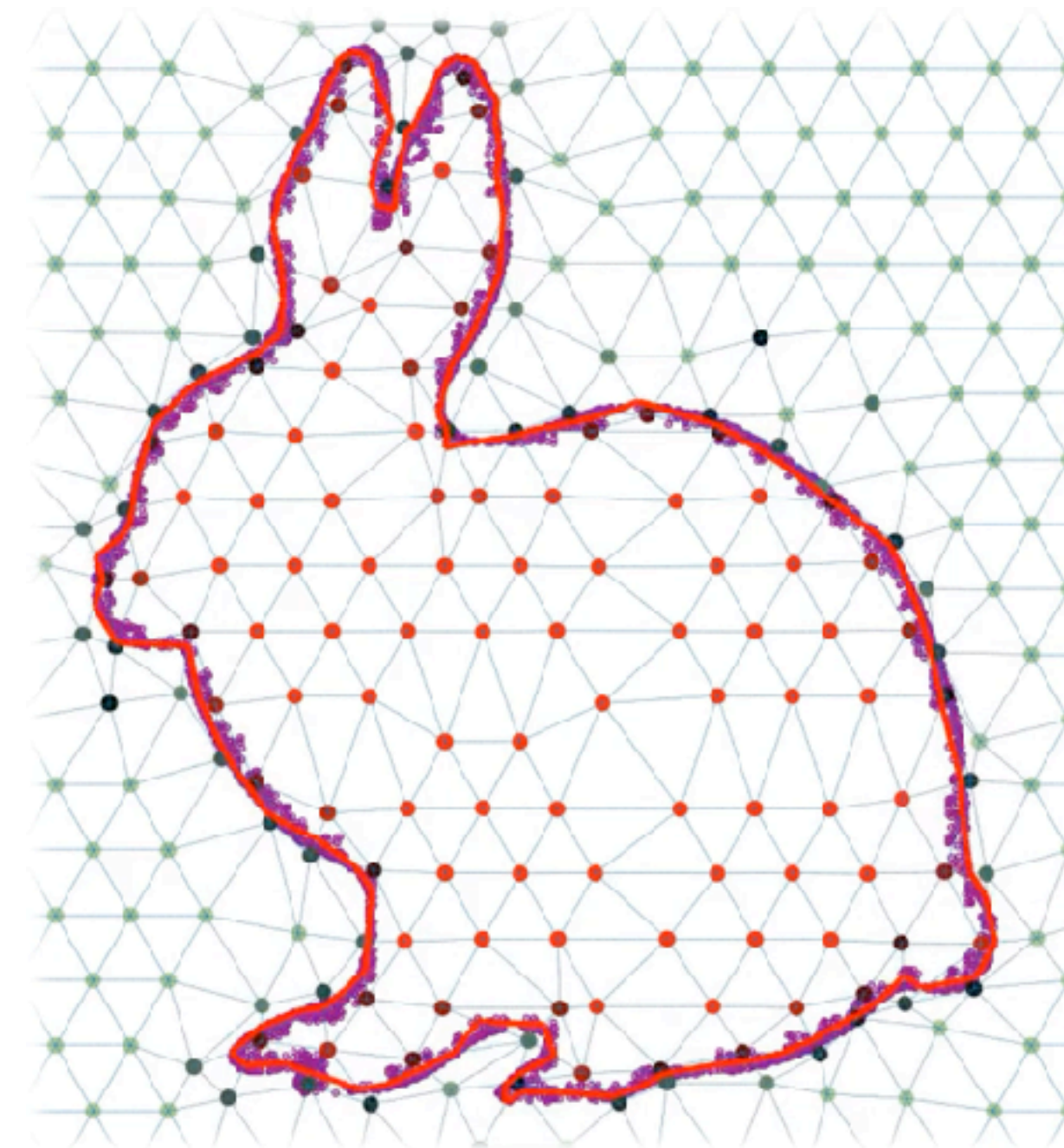
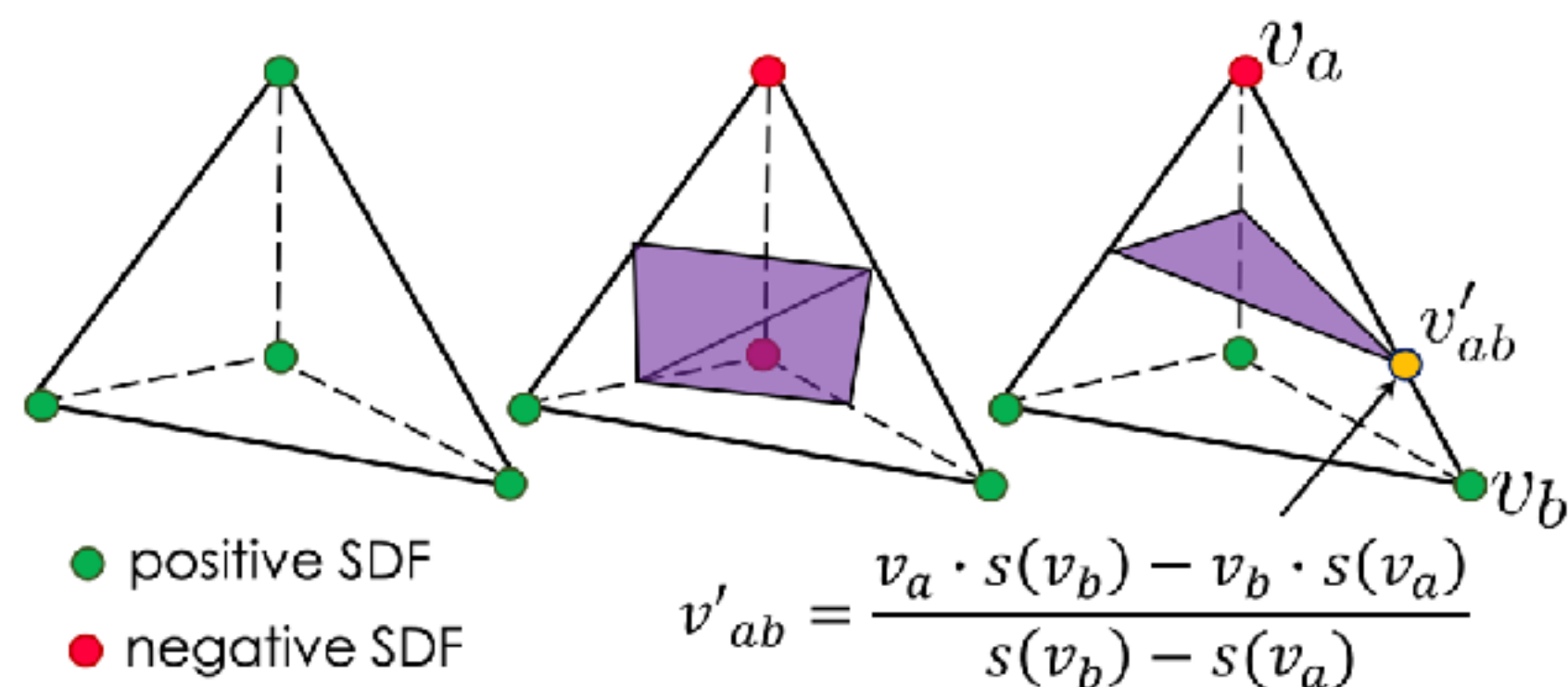


<https://ishit.github.io/nie/>

Inverz Grafika

Implicit felület differenciálható renderelése

https://kaolin.readthedocs.io/en/latest/notes/volumetric_meshes.html#deep-marching-tetrahedra-dmtet



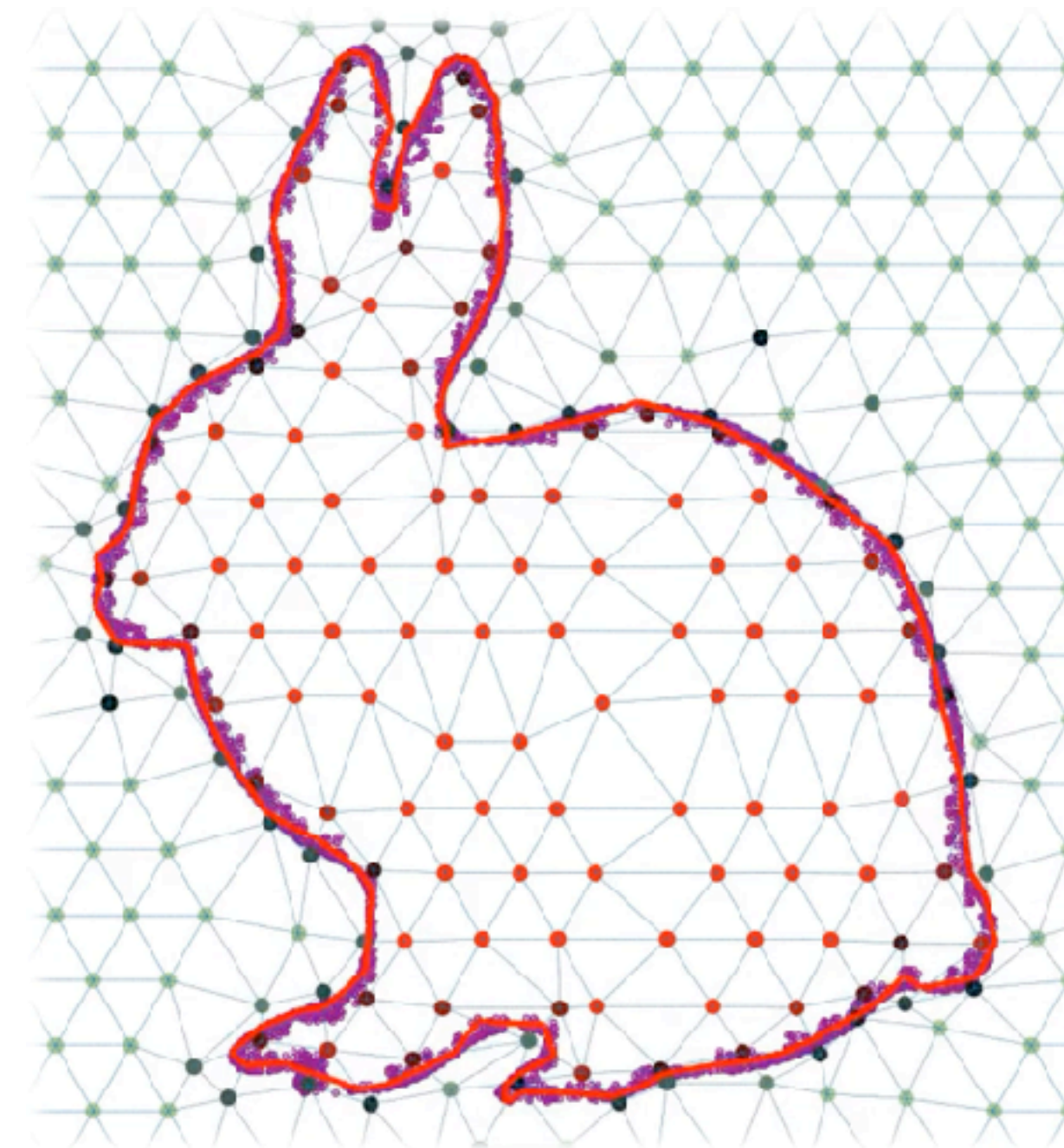
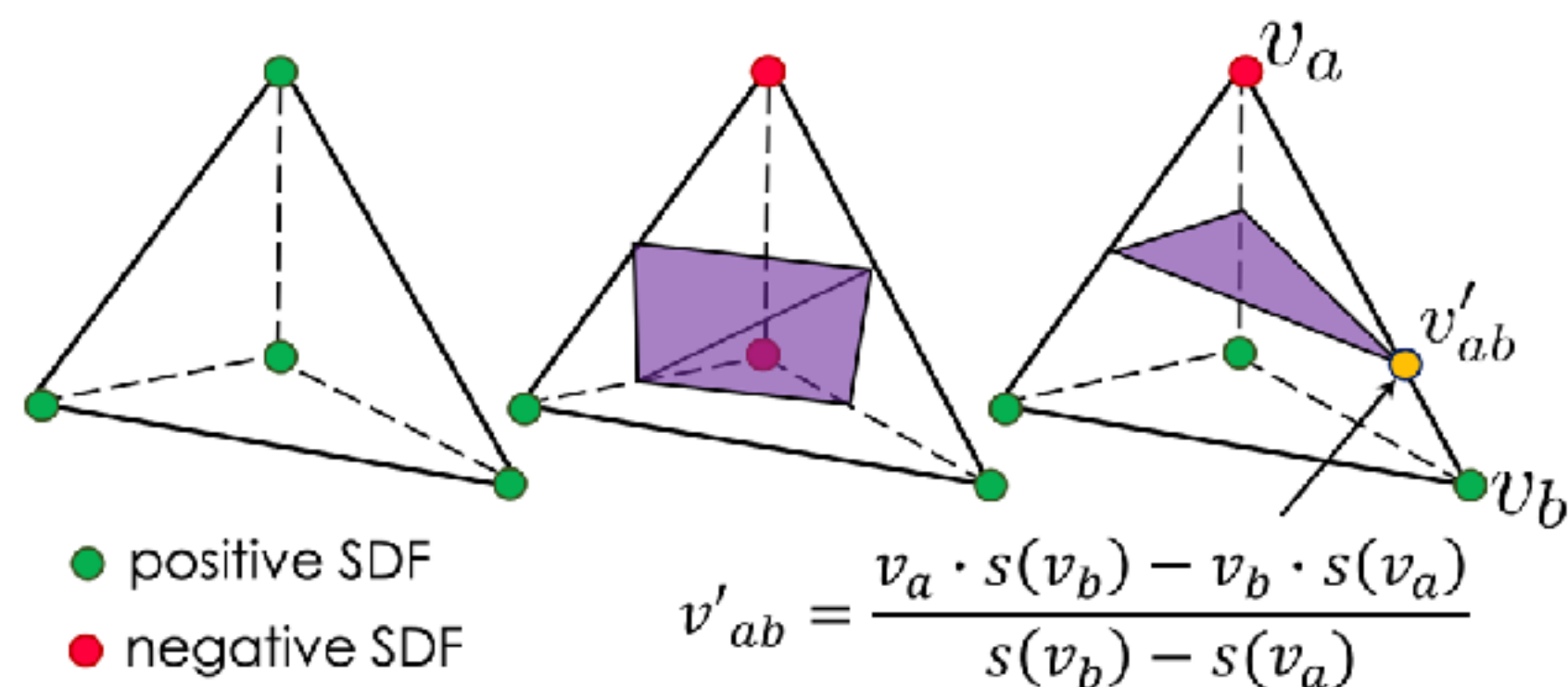
Differenciálható implicit — mesh konverzió
deformált masírozó tetraéderekkel!

*Miért nem inkább masírozó kockák?
Deformált kocka trilineáris interpolációja “invertálódhat!”*

Inverz Grafika

Implicit felület differenciálható renderelése

https://kaolin.readthedocs.io/en/latest/notes/volumetric_meshes.html#deep-marching-tetrahedra-dmtet



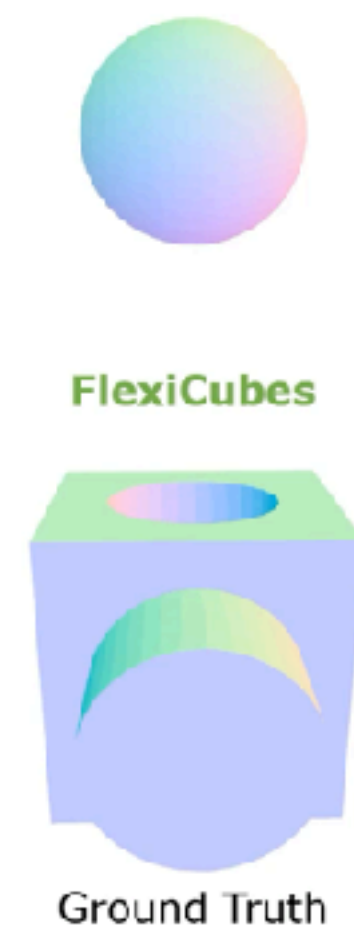
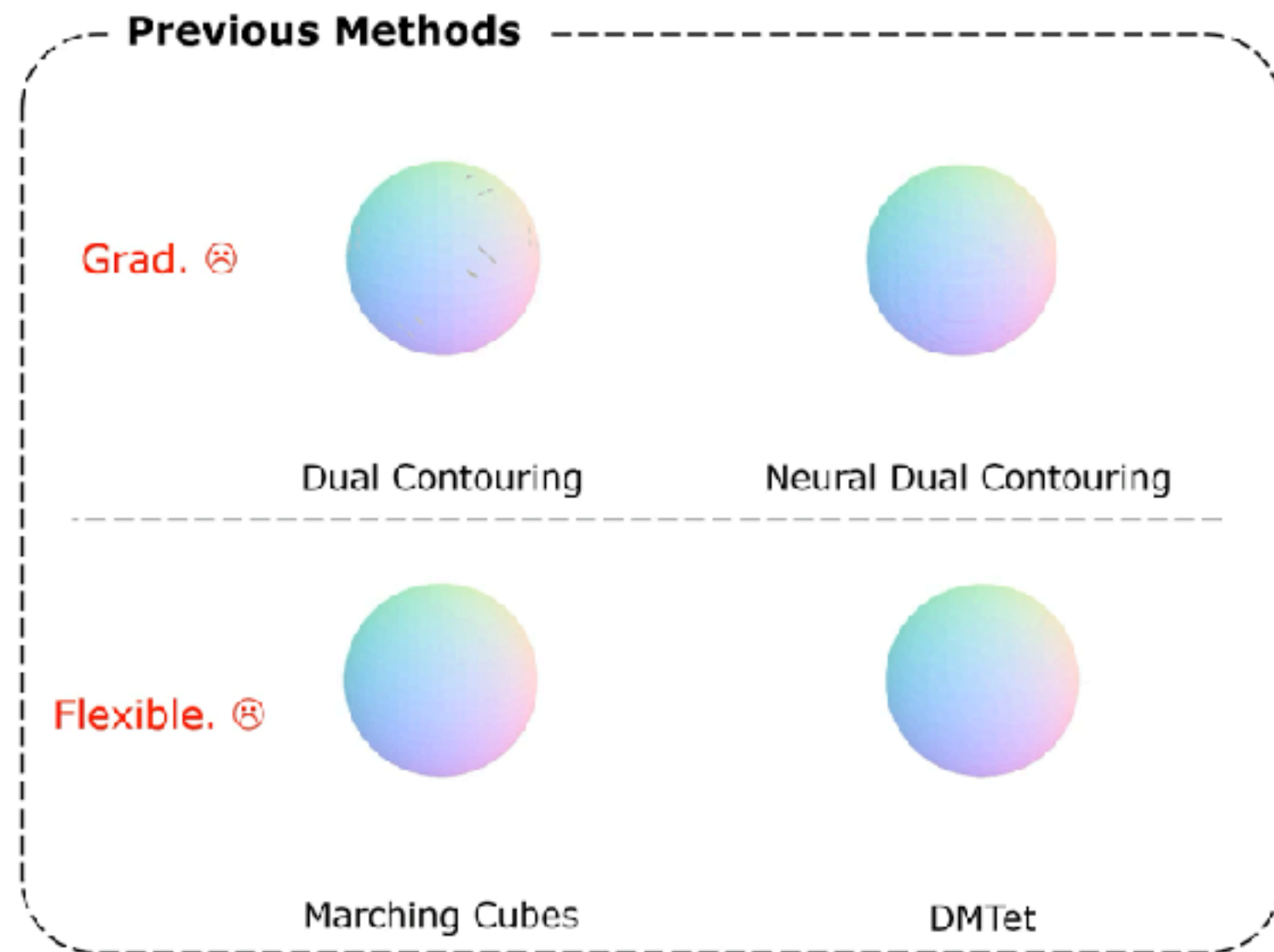
Differenciálható implicit — mesh konverzió
deformált masírozó tetraéderekkel!

*Miért nem inkább masírozó kockák?
Deformált kocka trilineáris interpolációja “invertálódhat!”*

Inverz Grafika

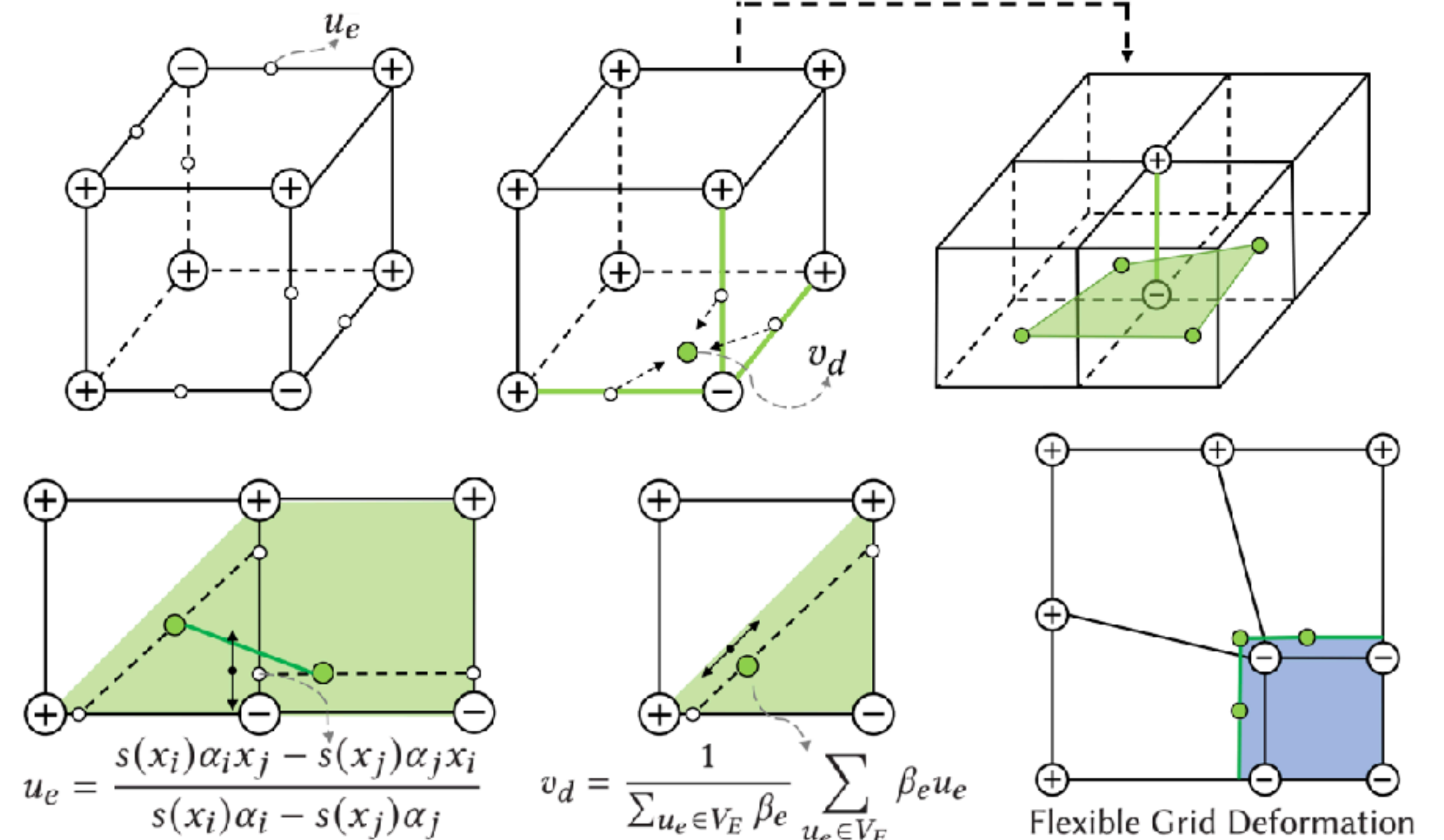
Implicit felület differenciálható renderelése

https://kaolin.readthedocs.io/en/latest/notes/volumetric_meshes.html#flexicubes



FlexiCubes:

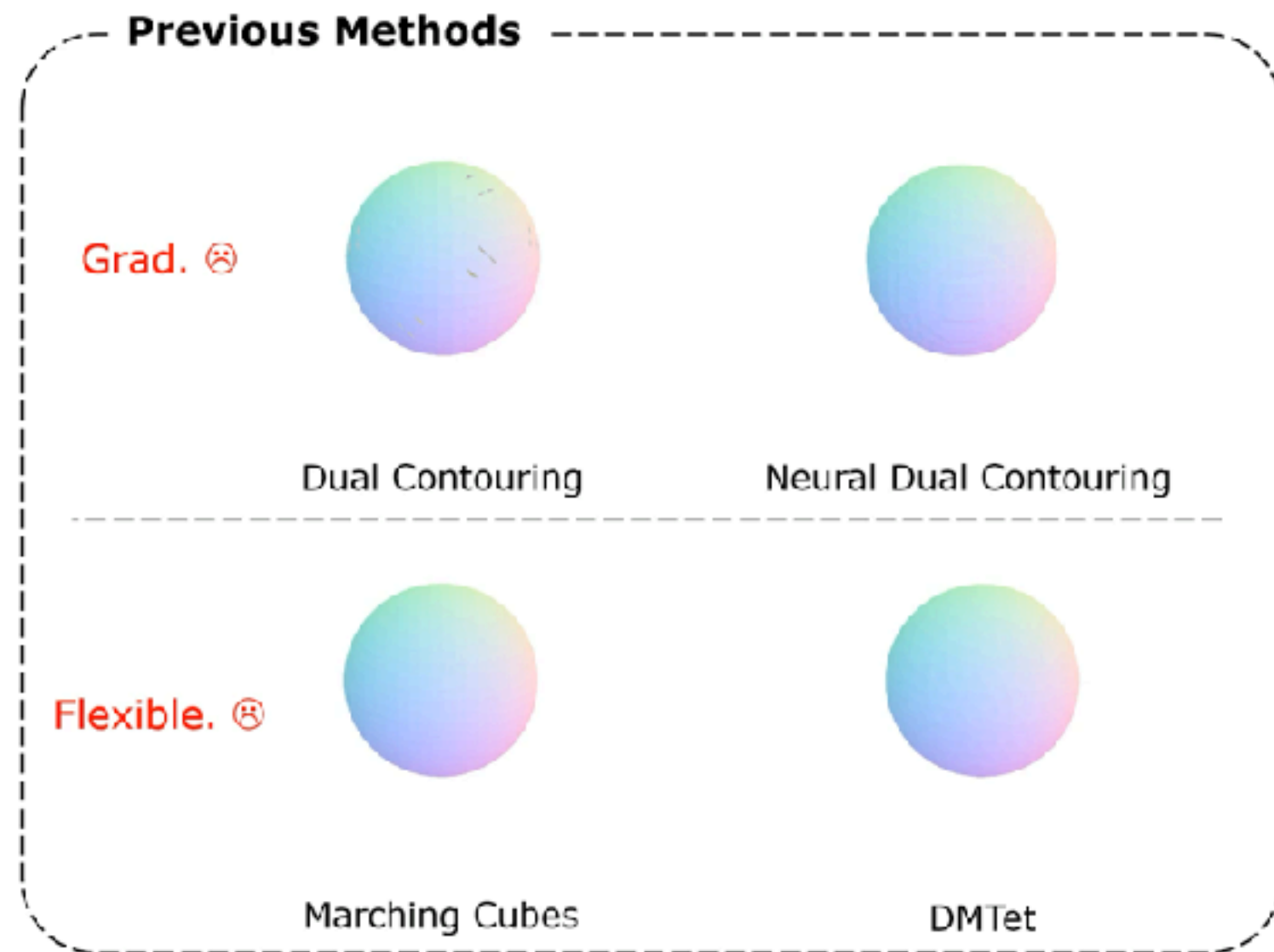
flexibilis, jól differenciálható implicit-mesh konverzió
(Inspiráció: “Dual Marching Cubes”)



Inverz Grafika

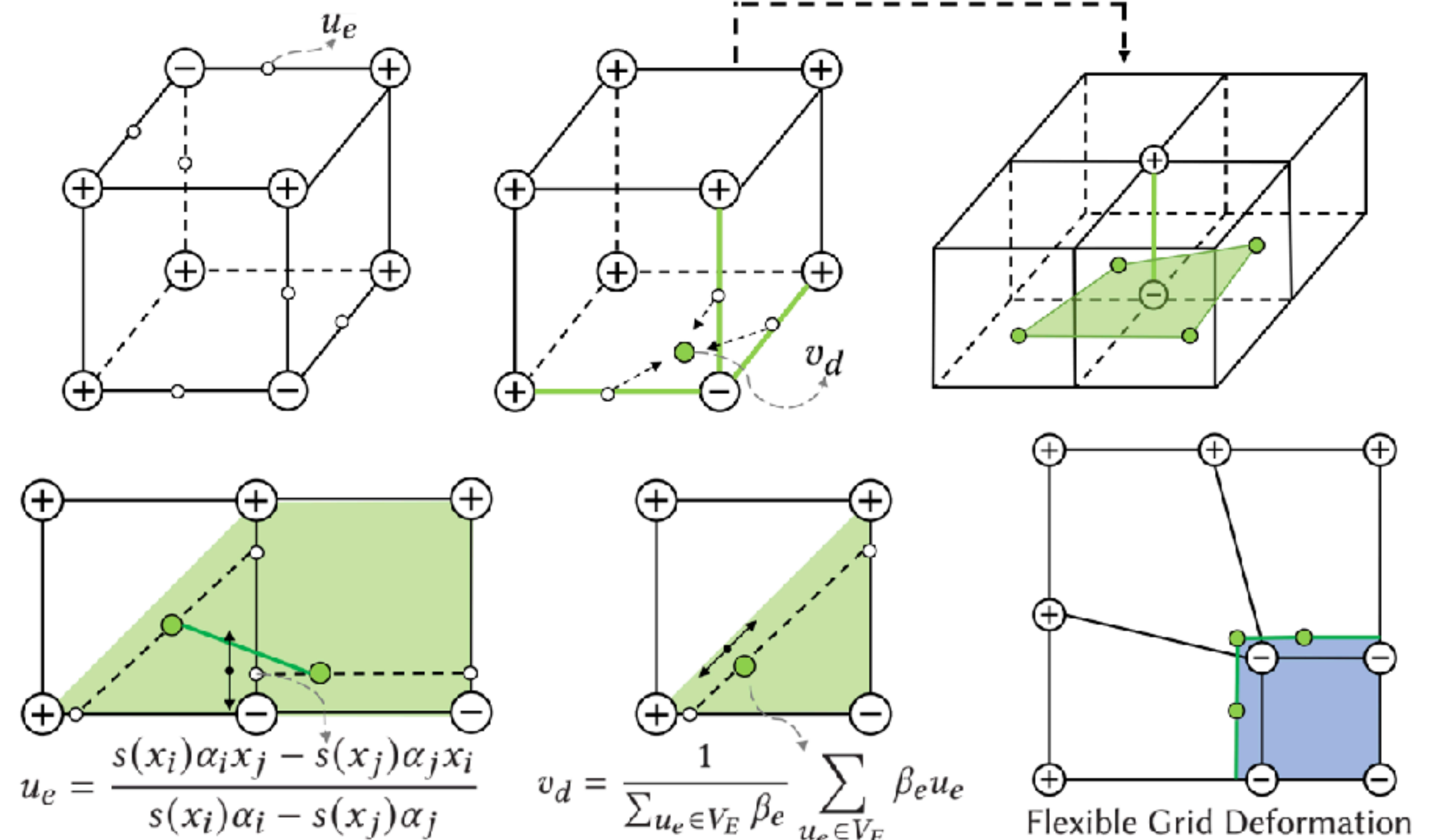
Implicit felület differenciálható renderelése

https://kaolin.readthedocs.io/en/latest/notes/volumetric_meshes.html#flexicubes



FlexiCubes:

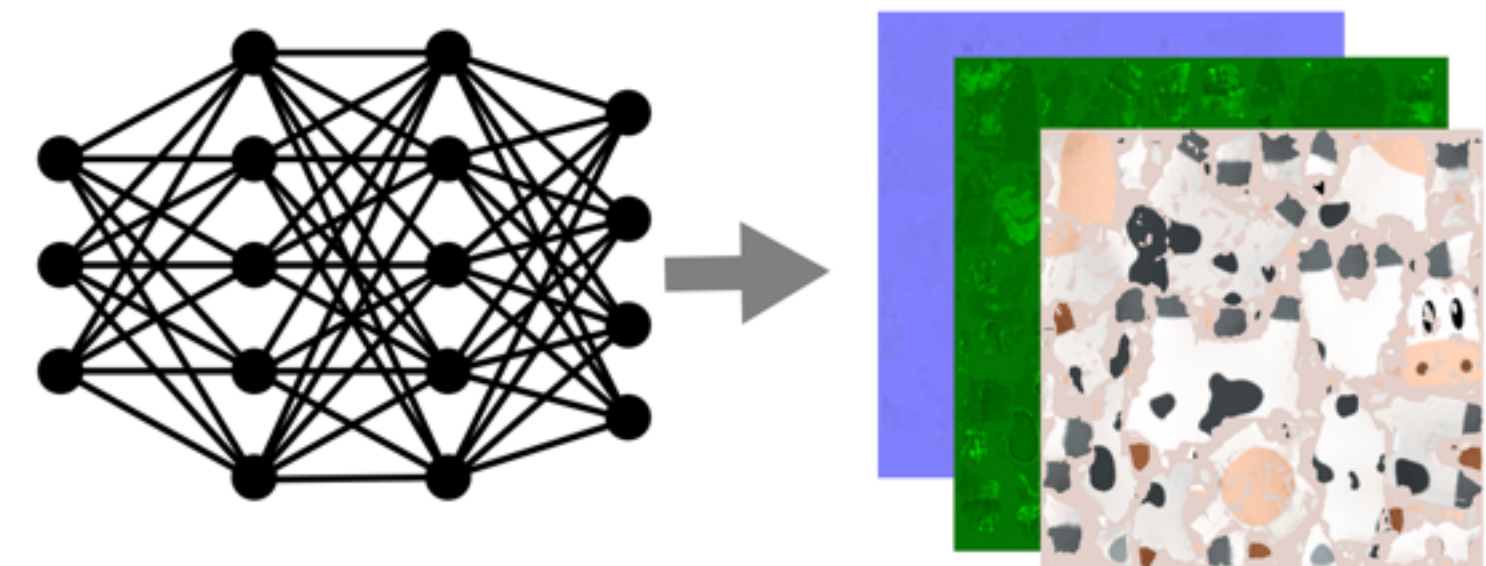
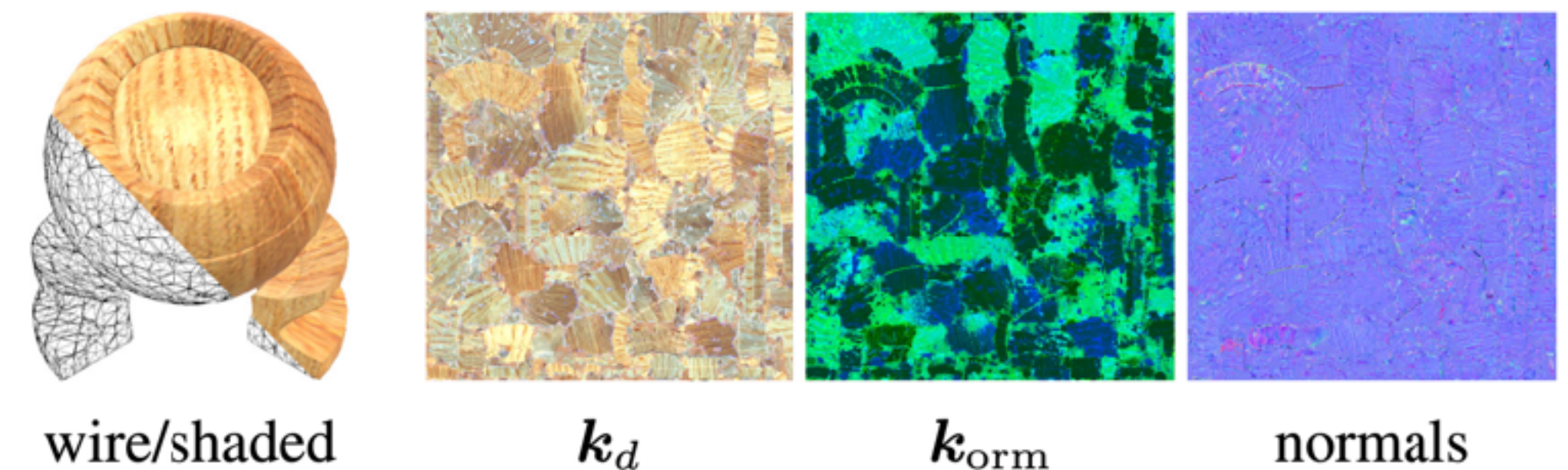
flexibilis, jól differenciálható implicit-mesh konverzió
(Inspiráció: “Dual Marching Cubes”)



Inverz Grafika

Textúrák optimalizációja

- PBR material: albedo, stb., textúrákban tárolva
- Textúra leképezéshez fel kell vágni a mesh-t — nehéz kombinatorikus jellegű probléma...
- Gyakori megoldás: volumetrikus textúra (voxel vagy neurális reprezentáció)
- Ha a geometria már stabilizálódott, a mesh-et UV mappeljük, majd a textúrákat esetleg tovább optimalizáljuk

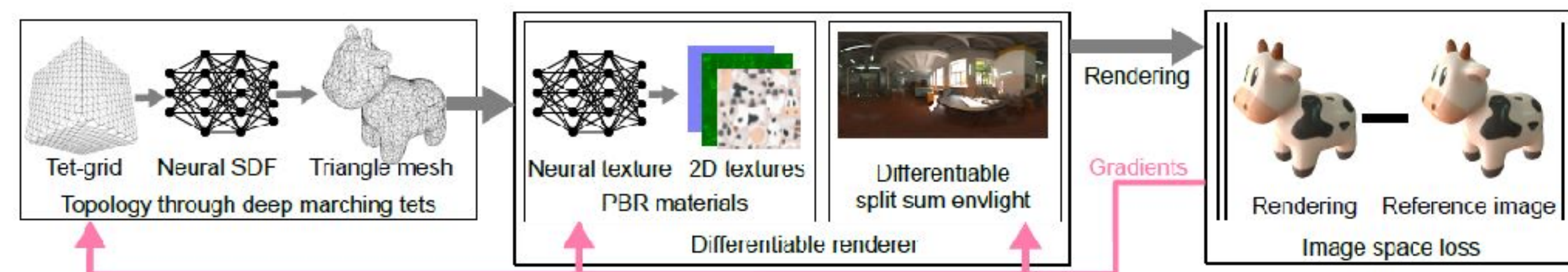
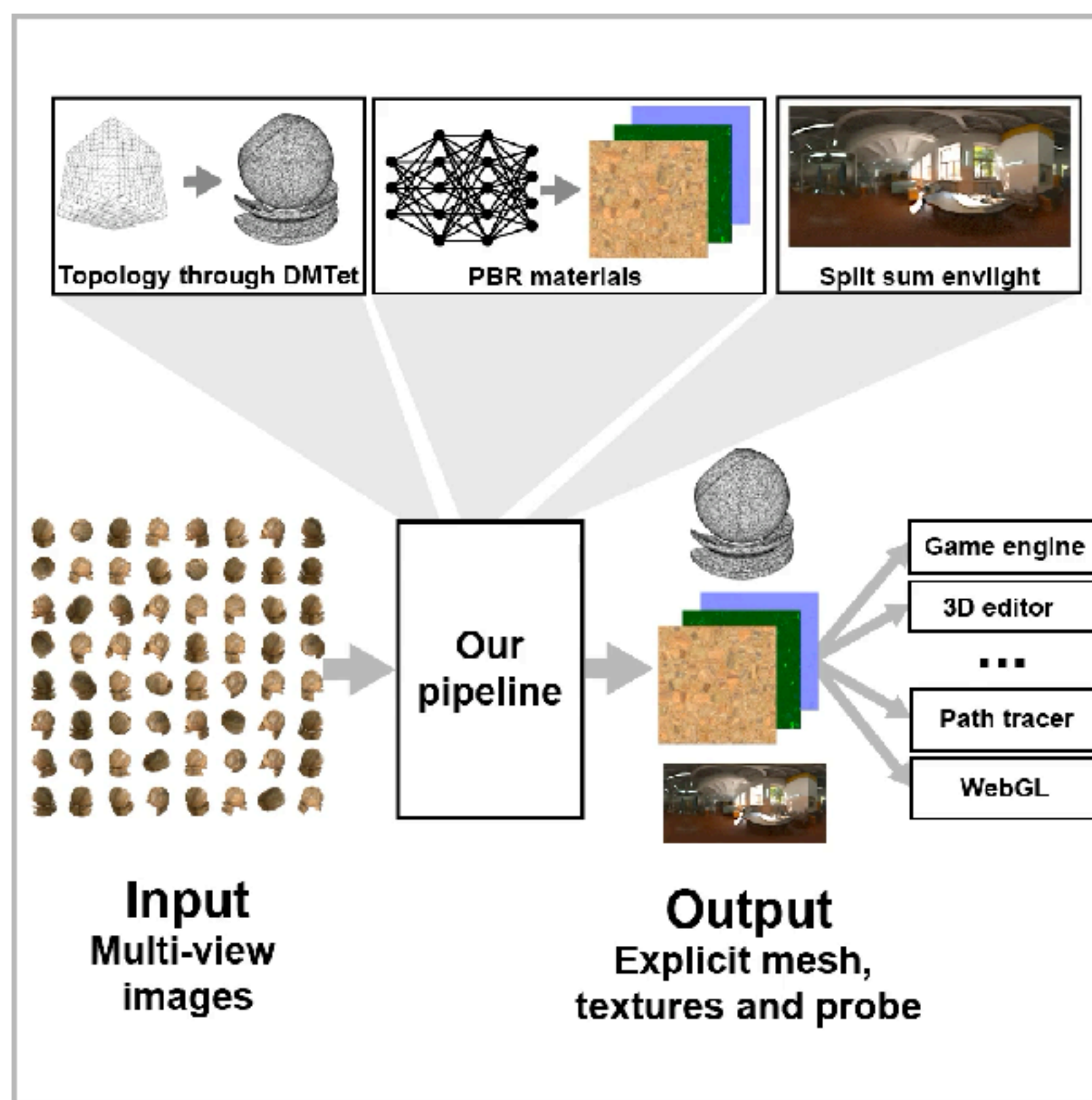


Inverz Grafika

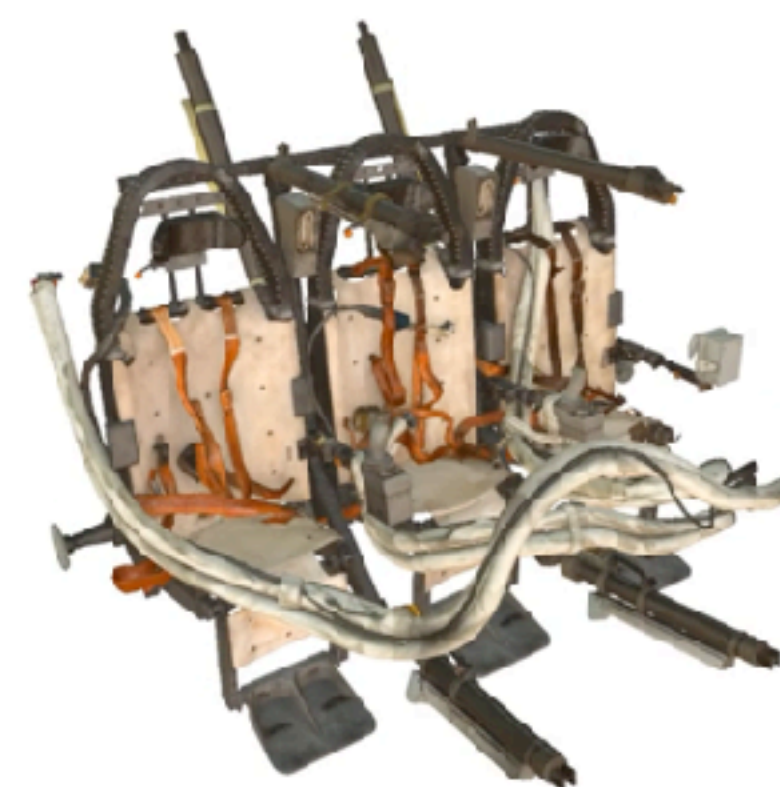
nvdiffrrec

Extracting Triangular 3D Models, Materials, and Lighting From Images

<https://nvlabs.github.io/nvdiffrrec/>



Iteration: 1200



Nvdiffrrec w/ DMTet

Iteration: 1200



Nvdiffrrec w/ FlexiCubes



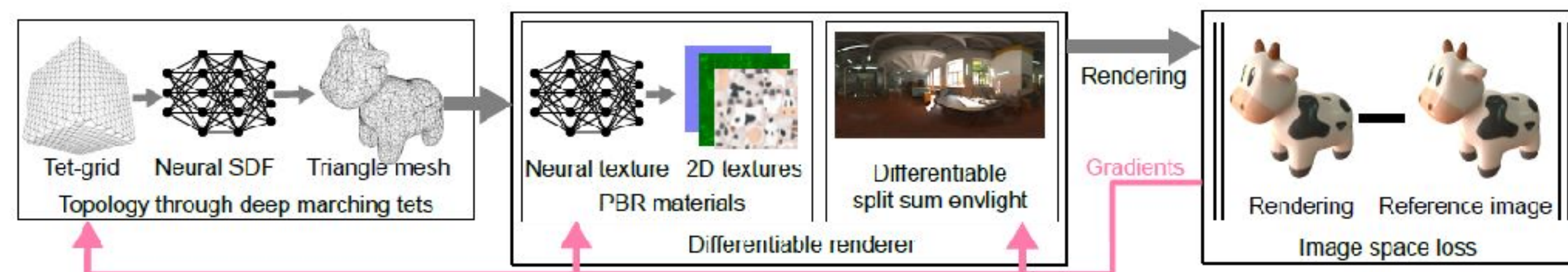
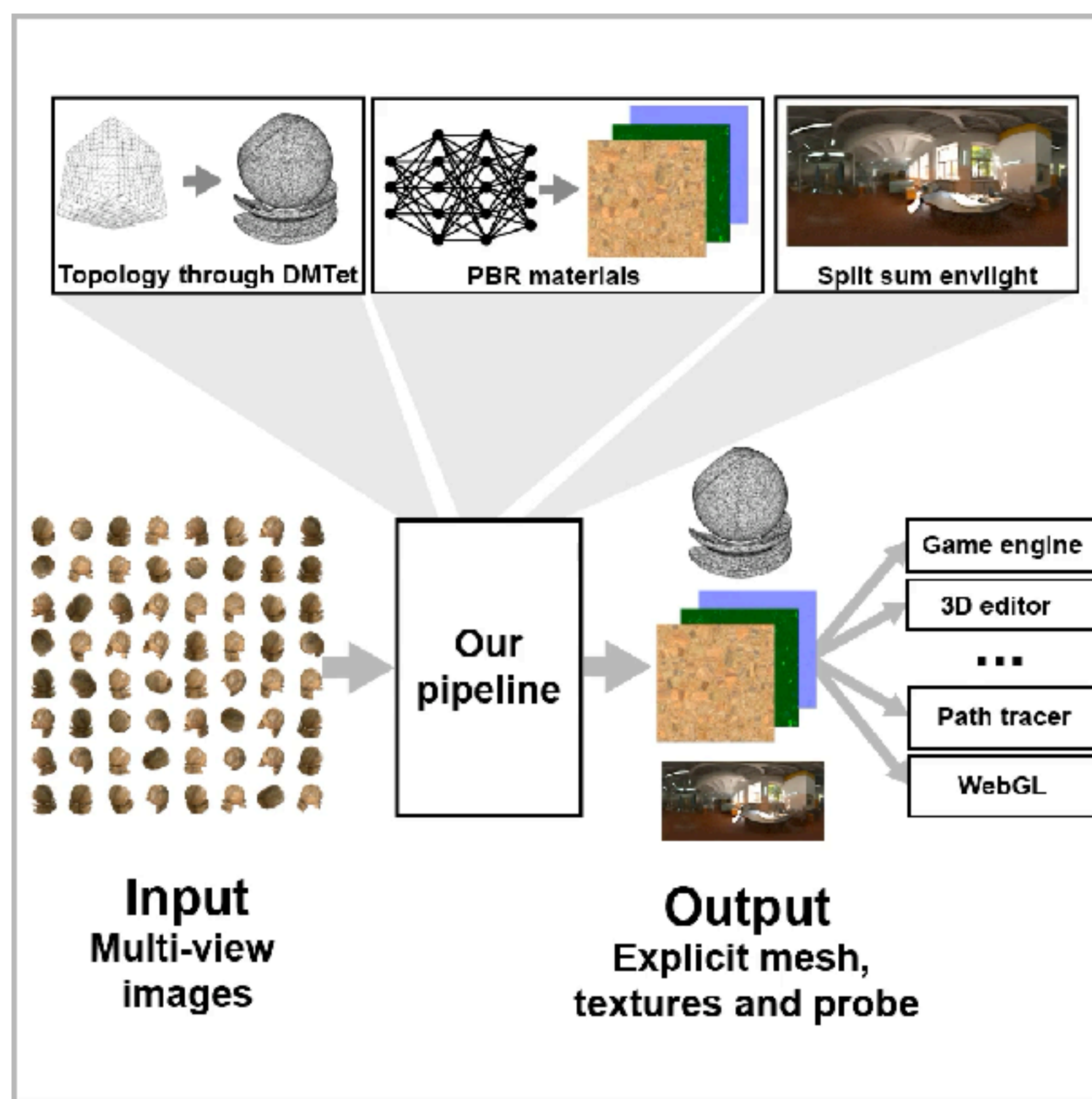
Reference

Inverz Grafika

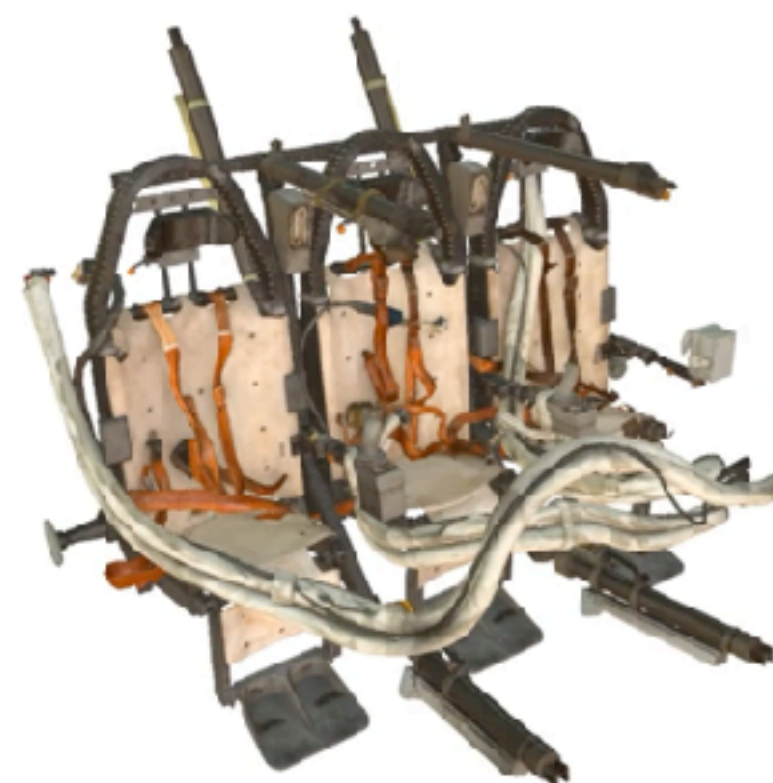
nvdiffr

Extracting Triangular 3D Models, Materials, and Lighting From Images

<https://nvlabs.github.io/nvdiffr/>



Iteration: 1200

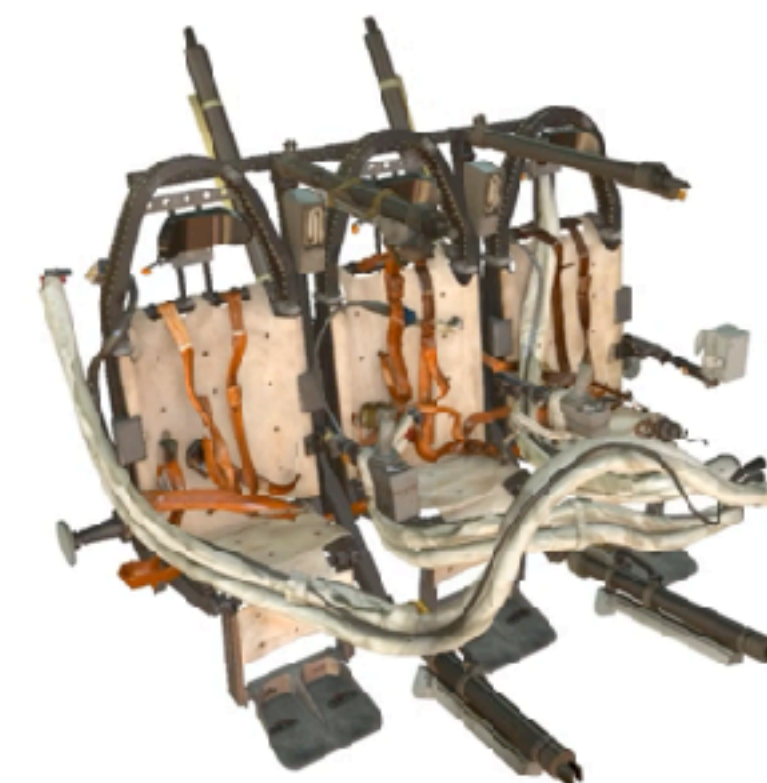


Nvdiffr w/ DMTet

Iteration: 1200



Nvdiffr w/ FlexiCubes



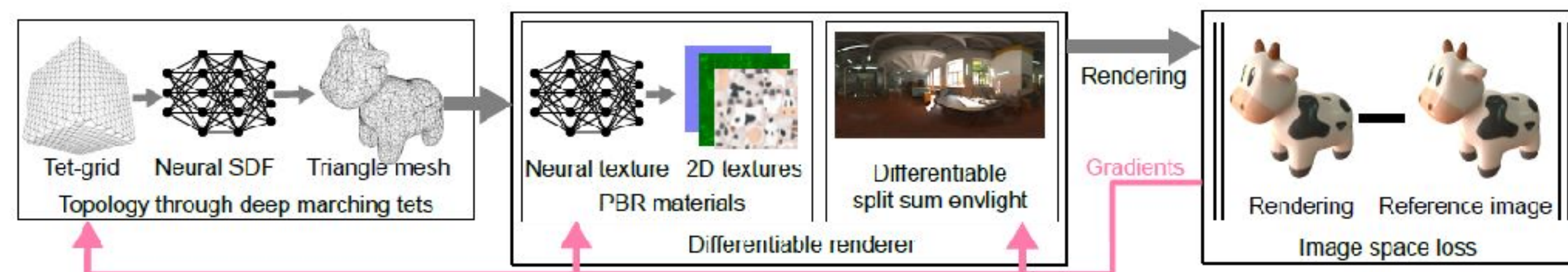
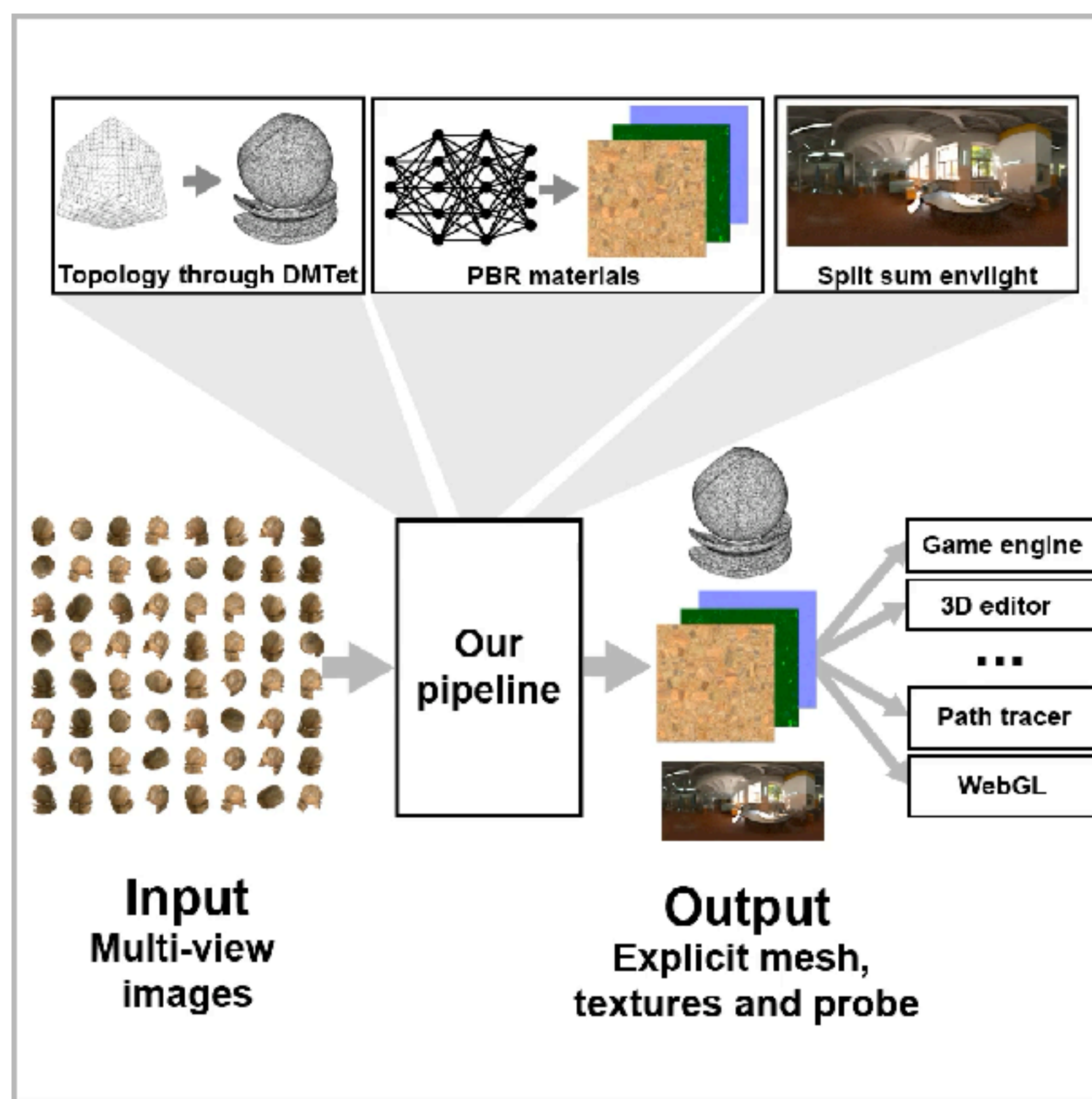
Reference

Inverz Grafika

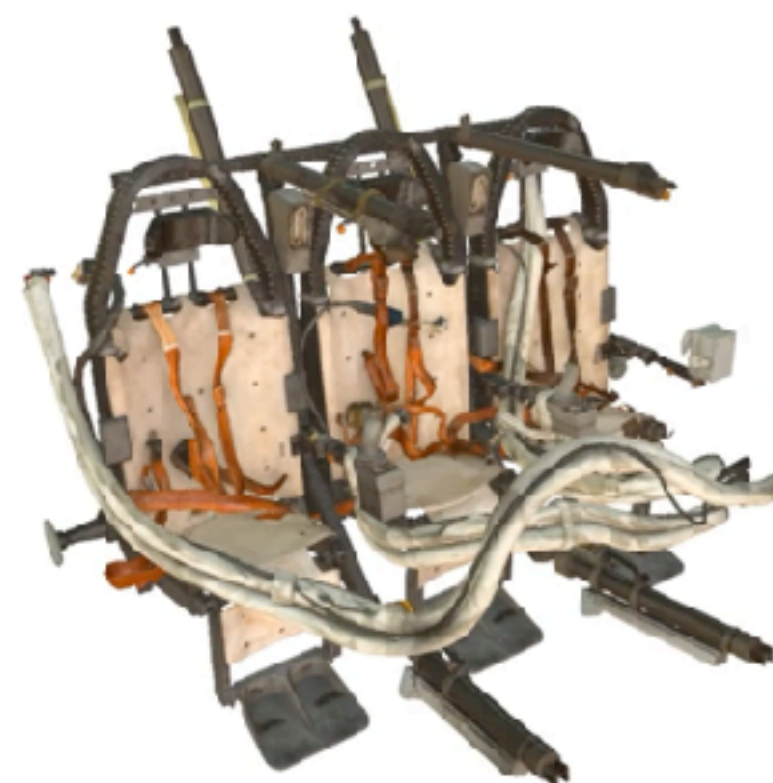
nvdiffrrec

Extracting Triangular 3D Models, Materials, and Lighting From Images

<https://nvlabs.github.io/nvdiffrrec/>



Iteration: 1200

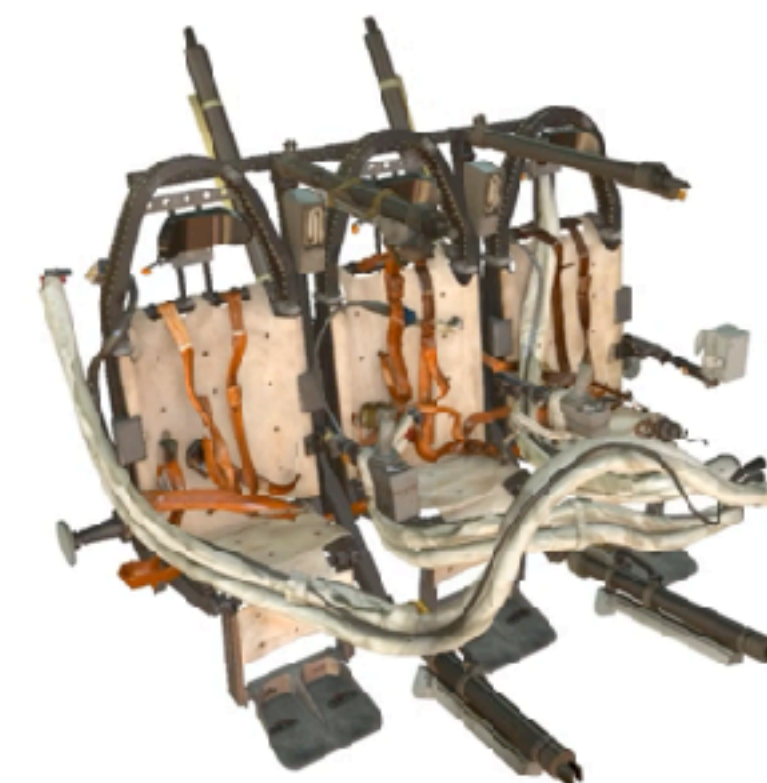


Nvdiffrrec w/ DMTet

Iteration: 1200



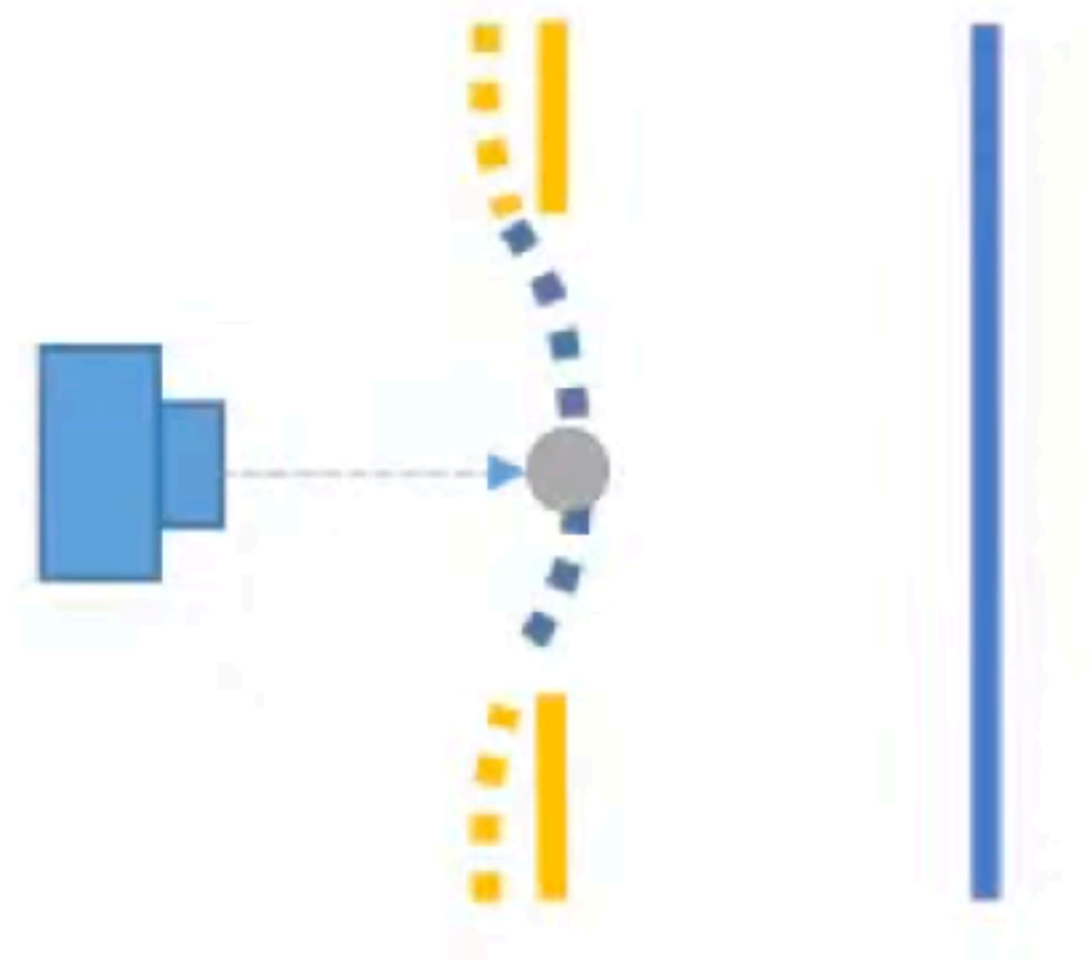
Nvdiffrrec w/ FlexiCubes



Reference

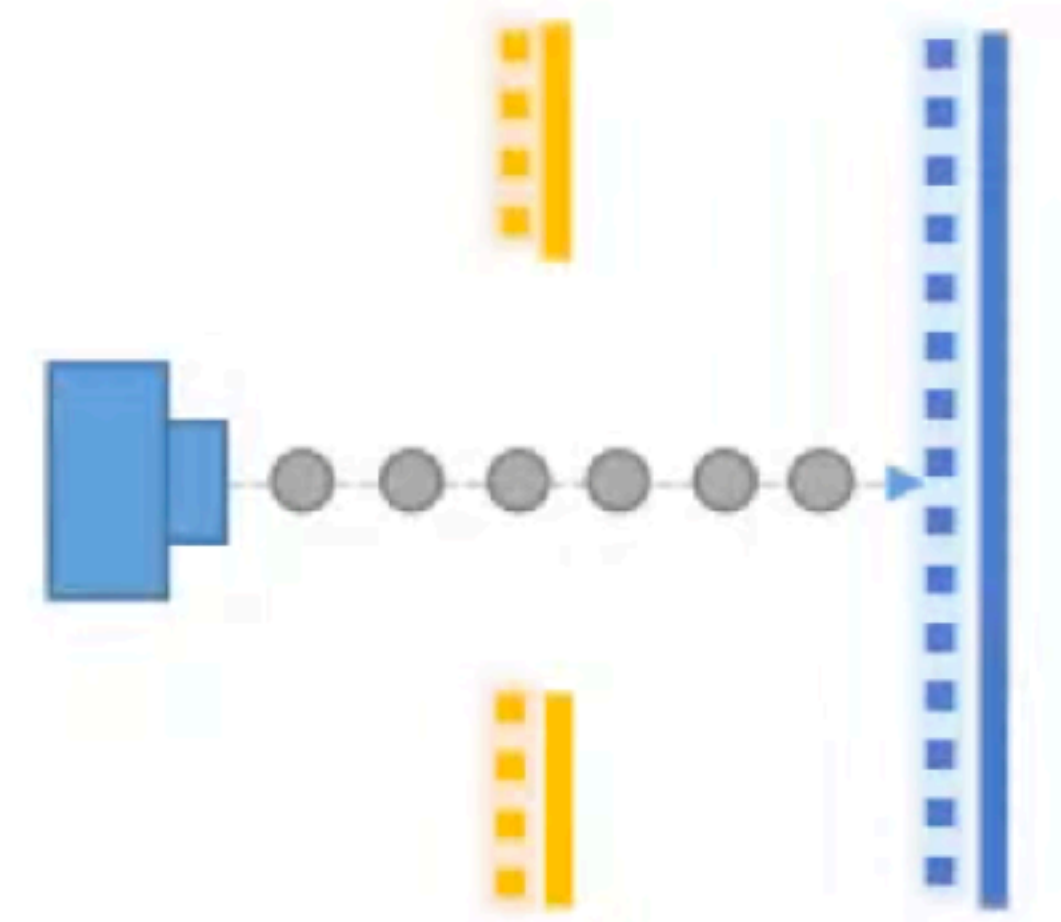
Inverz Grafika

Felületi vs térfogati reprezentációk



Felületi reprezentáció
(pl. mesh/SDF)

“Precíz” felületek
Nehéz optimalizálni (lokális minimumok)
Topológiát nehéz módosítani

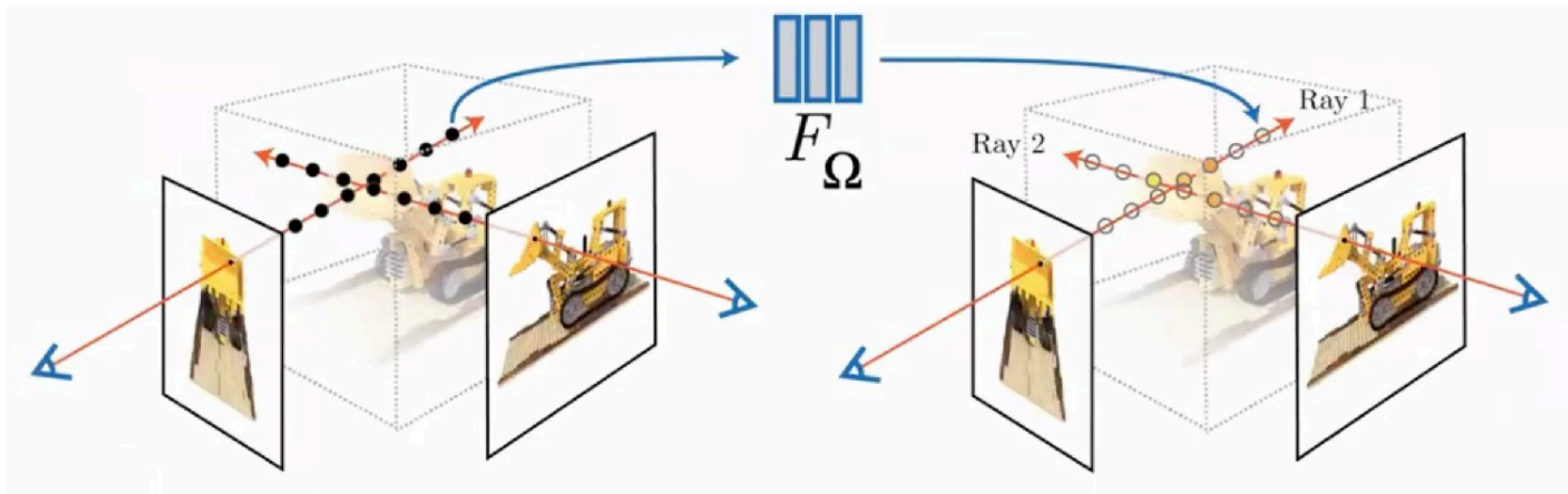


Térfogati reprezentáció
(pl. sűrűség/radiancia mező)

“Imprecíz” felületek
Könnyű optimalizálni
Topológiát könnyű módosítani (lyukat ütünk)

Következő Előadás:

Radiancia mezők



Differenciálható volume rendering, radiancia mezők, NeRF