Real-time Approximation of Photometric Polygonal Lights

Christian Luksch | Lukas Prost | Michael Wimmer

vrvis | rm DATA | TU WIEN

Introduction

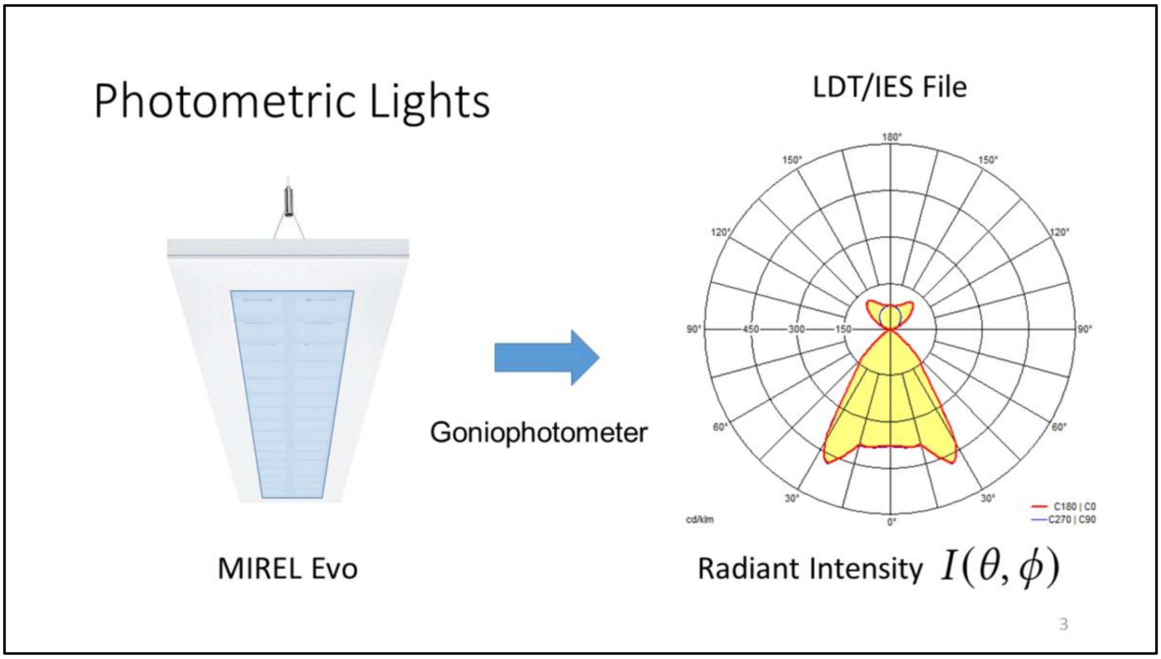
ZUMTOBEL https://mobilevr.zumtobel.com/

Photometric measurement data is used to simulate real-world luminaires for Lighting Design & Architecture

Interactivity is important in these workflows

Real-time rendering system provide a huge benefit, but shading of photometric area lights is currently limited

Our technique provides a suitable real-time approximation for these application

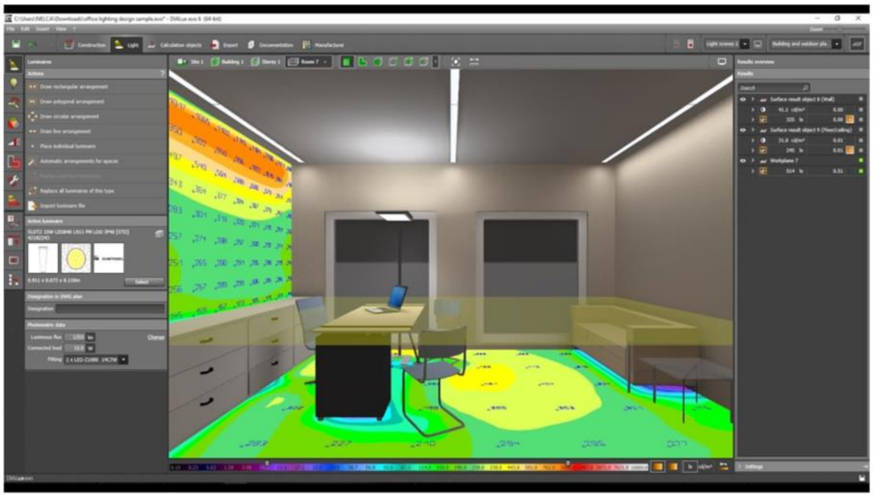


Far-field photometry: Absolute emission of luminaire represented by spherical radiant intensity function
 -> simplification of entire luminaire to point with spherical radiant intensity profile (Anisotropic Point Light)
 -> actually does not allow to simulate the true illumination in near-field

Nevertheless, its is common to model the light emission as area:
 -> valuable depiction of luminaire shape & characteristics in near-field illumination
 -> smoothens details of radiant intensity functions in near-field
 -> avoid singularity

Photometric Report (LDT or IES) also contains meta information such as:
 - Body Shape & Dimension
 - Luminous Area Shape & Dimension

Lighting Design



DIALux Evo

Luminaire model: Dummy Geometry or CAD Model + Light Emitting Objects (Disk or Polygon) linked with Radiant Intensity Profile of a photometric measurement (LDT or IES)

Lighting design tools typically use offline rendering kernels – no difficulty in calculating illumination from area with arbitrary emission

Limitation of interactivity:
- Wait for Render after Interaction
- Separation of CAD viewport without lighting and realistic/calculation render view
- Unlike game engine editors with What-You-See-Is-What-You-Get

Dialux:
- Manual toggle to real-time point representation
- Manual re-calculation after scene modifications

Real-time Rendering

Luksch et al., *Fast Light-Map Computation with Virtual Polygon Lights*, I3D 2013

HILITE

HILITE: Lighting design prototype using a real-time rendering system with baked global illumination
-> real-time point representation during interaction / blending to baked solution after interaction

Motivation

- Enable real-time applications with photometric lights
- Depiction of luminaire characteristic and shape in near-field
- Polygonal emission sufficient for most cases

Point Representation | Our Approximation | Reference Solution

6

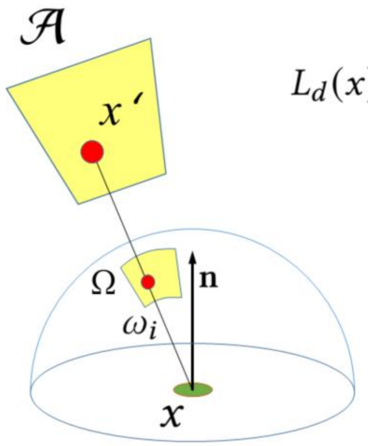
Goal: Minimal difference to offline rendering solution – plausible intensity

Mostly Linear & Rectangular lights -> polygonal emission sufficient for most cases

Accumulation buffer + de-noising is not always practical and not easy to integrate -> closed form solution

Smoothed illumination in near-field should make an approximation possible

Photometric Light Rendering Equation



$$L_d(x) = \frac{\rho}{A\,\pi} \int_{\mathcal{A}} I(-\omega_i) \frac{(\omega_i \cdot n_x)}{||x' - x||^2}\, dx'$$

1. No Visibility
2. Diffuse BRDF
3. Hemisphere Integral

$$L_e(\omega) = \frac{I(\omega)}{A\,(\omega \cdot n_A)}$$

7

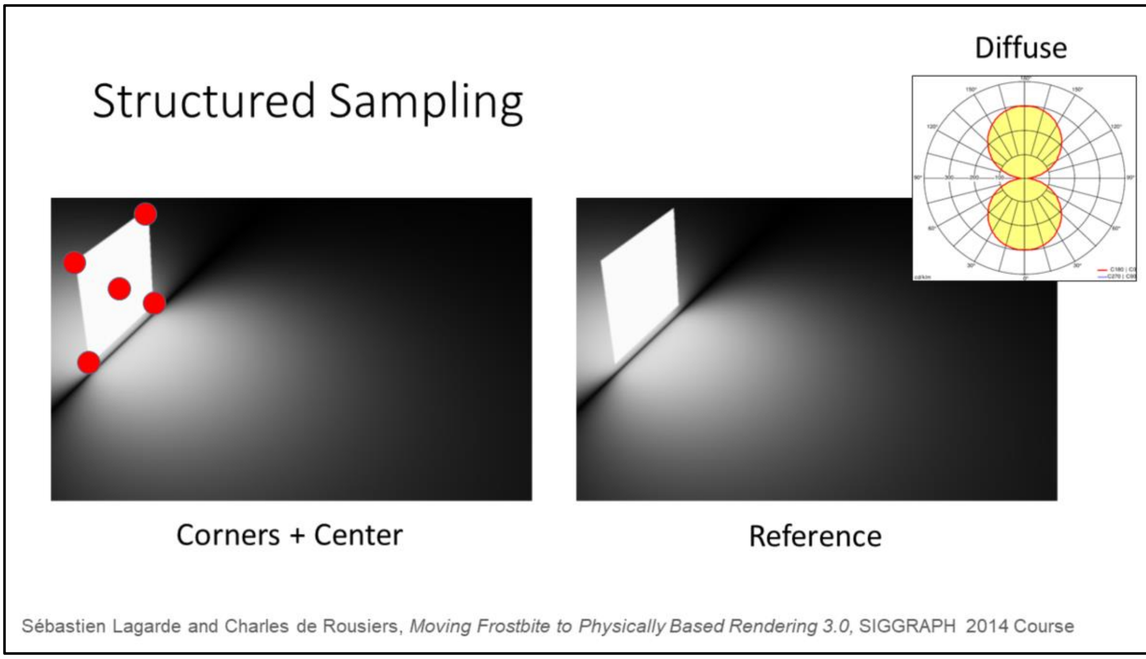
Normalization by area of polygon as intensity profiles gives absolute radiant intensity of entire luminaire

Simplification to allow approximation:
- Diffuse BRDF
- No visibility term

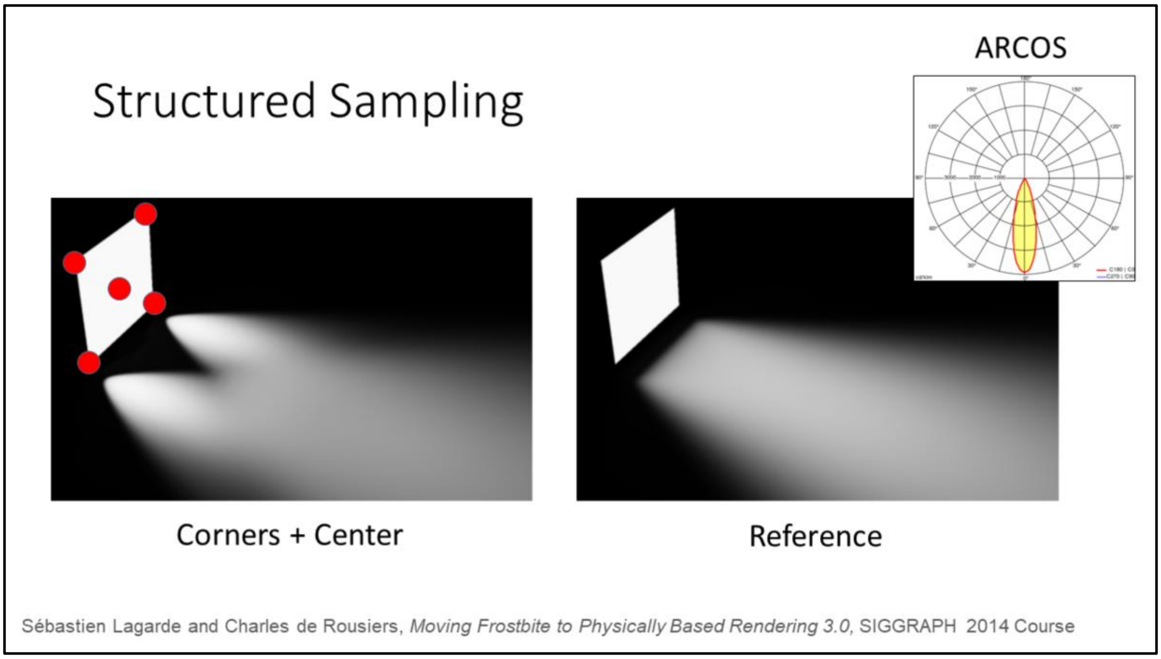
-> Reformulate as Hemisphere Integral to remove d^2 and increase effectiveness of approximation

Radiant Intensity reformulated to Radiance

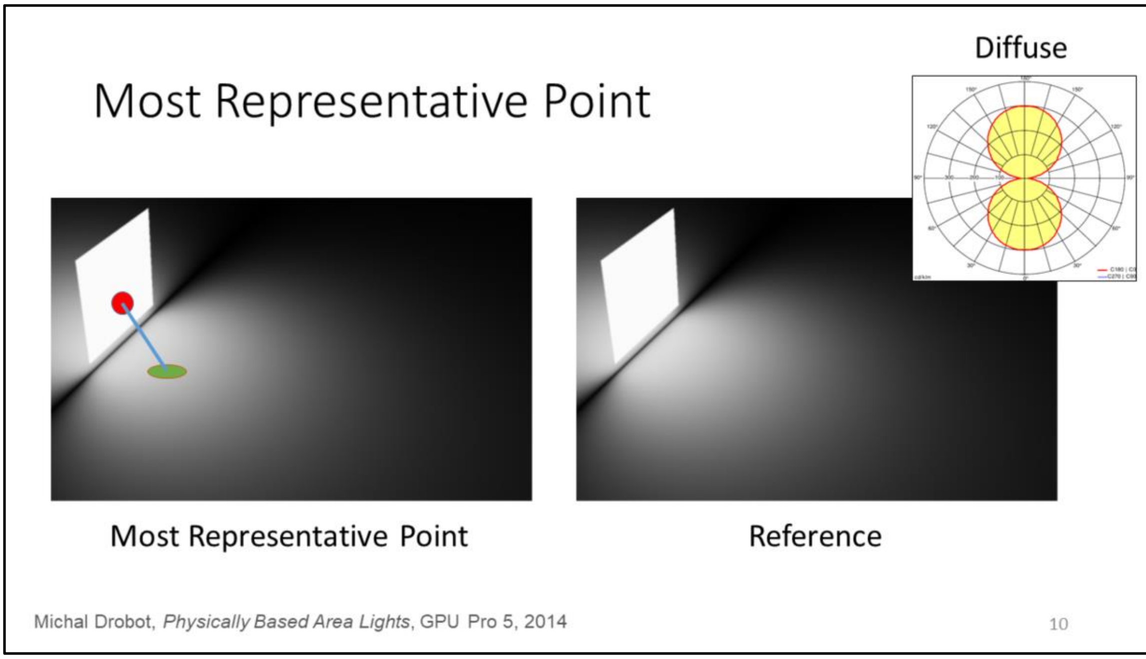
NOTE: Division by „omega dot n“ does not allow to calculate shading within area light plane -> see paper how we avoid artifacts



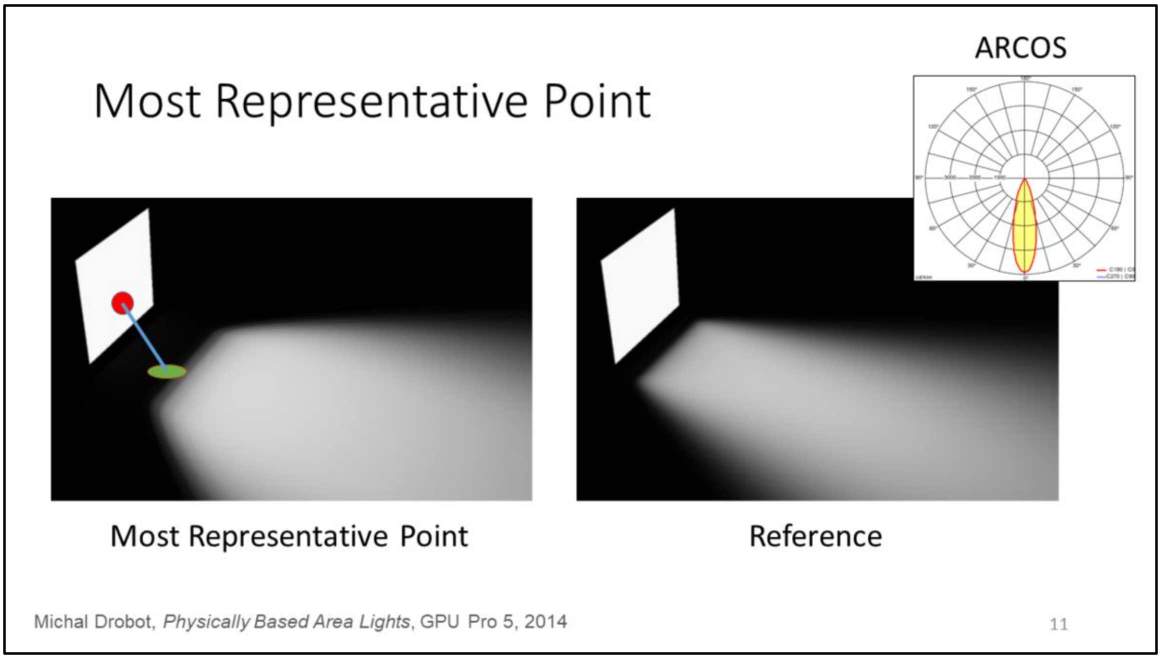
Very good approximation in case of diffuse emission



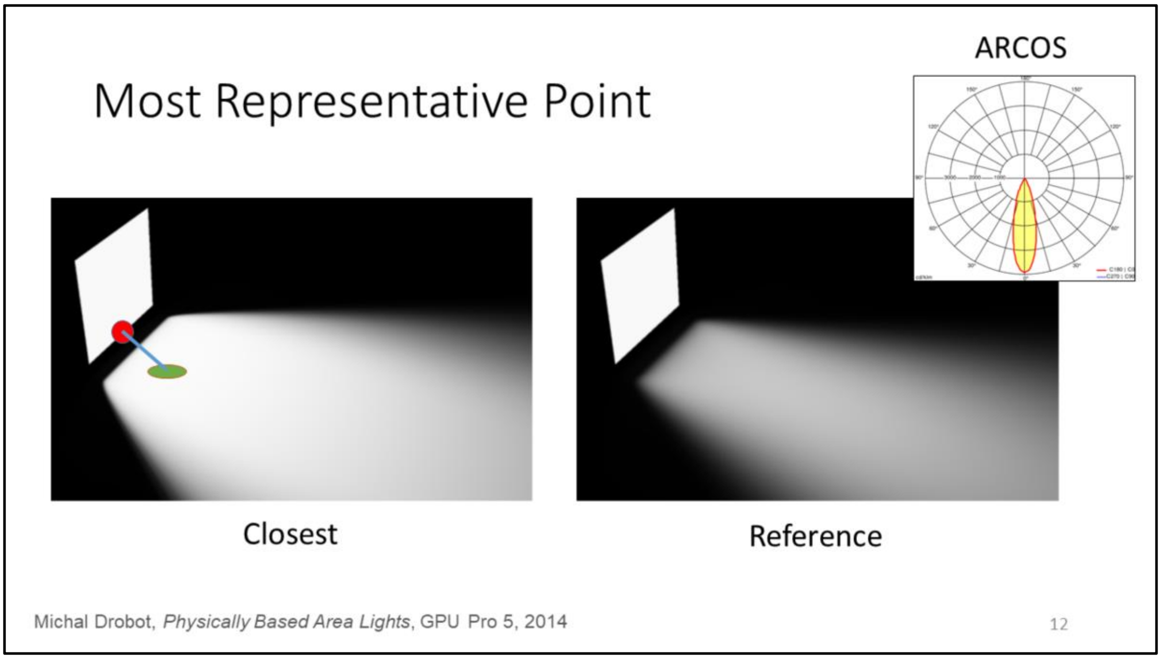
Photometric Emission: Solution not acceptable in near-field, sampling patter visible
-> improved illumiation at medium distance compare to single point



Use spatial heuristics based on light orientation and point to shader to find single point for best approximation -> very good for diffuse lights

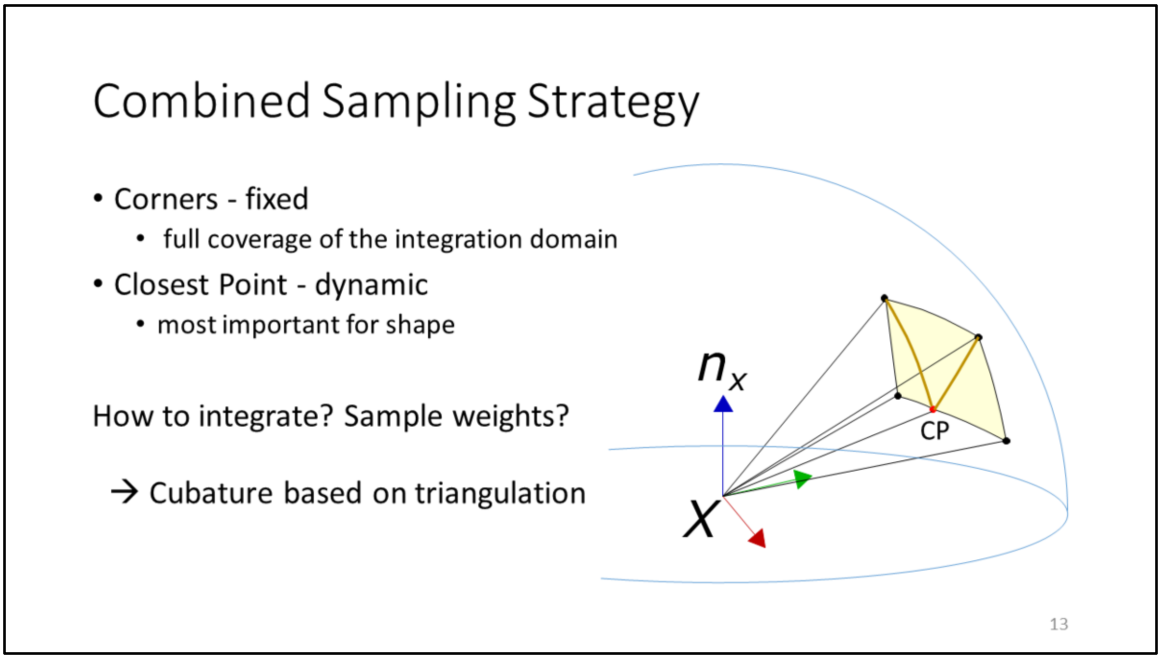


MRP provide sampling pattern-free approximation, but does not cover the integration domain sufficiently
Illumination will always have an offset in the near-field



Closest point: well defined outline, but intensity too high

-> Combine fixed well-selected samples with a dynamic sample point, we chose the closest point

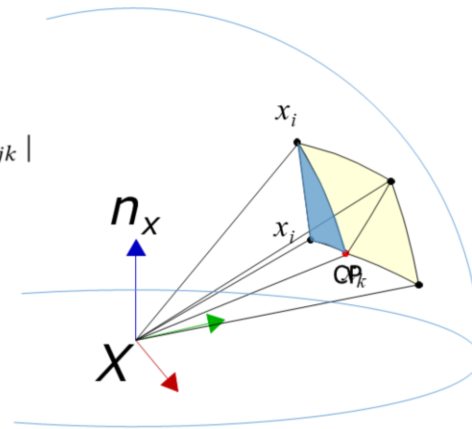


Starting with Polygon projected to hemisphere
Clip Polygon area below hoizon
Combined fixed and dynamic samples: PDF no longer uniform -> cubature based on triangulation
Any triangulation can be used, ideally Delaunay triangulation -> we use triangle fan around closest point

Cubature based on triangulation

$$Q_N = \sum_{\triangle_{ijk} \in D} \frac{f(x_i) + f(x_j) + f(x_k)}{3} | \triangle_{ijk} |$$

$| \triangle_{ijk} |$= solid angle



14

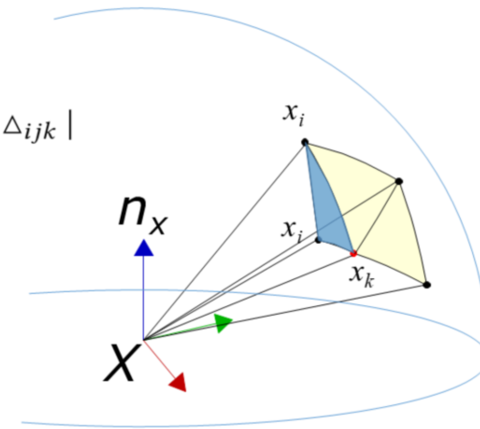
Weighed sum of all triangles average function value
weight = solid angle

Cubature based on triangulation

$$L_d(x) = \frac{\rho}{\pi} \sum_{\triangle_{ijk} \in D} \frac{L_i G_i + L_j G_j + L_k G_k}{3} \, | \triangle_{ijk} |$$

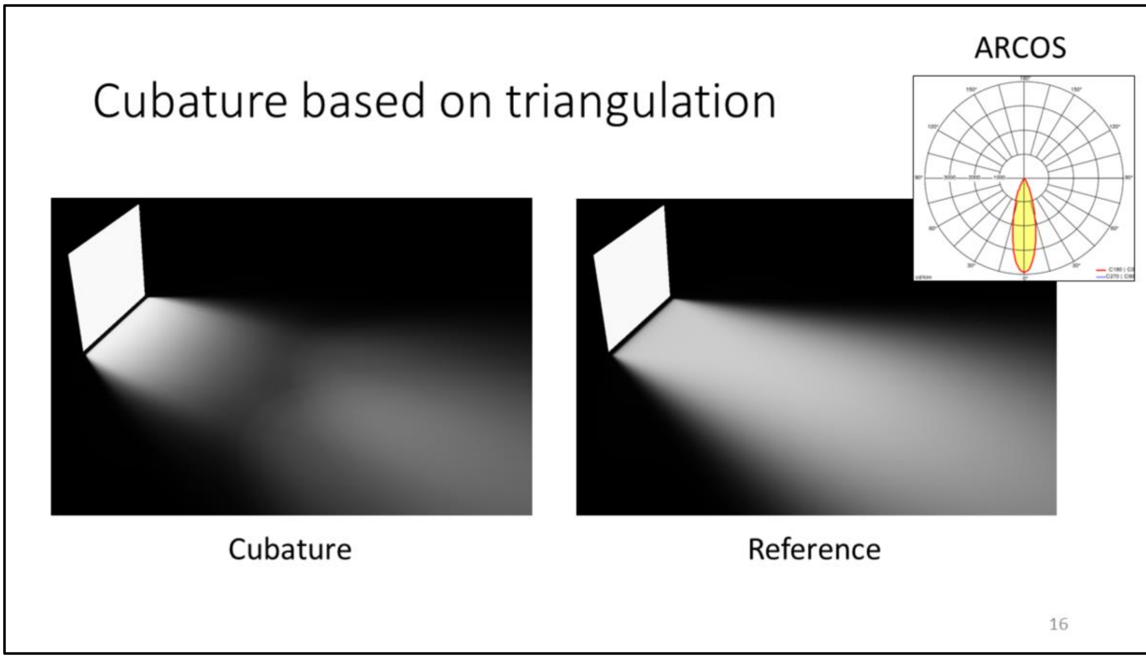
$$L_i = L_e(\omega_i)$$

$$G_i = \mathbf{n}_x \cdot \omega_i$$



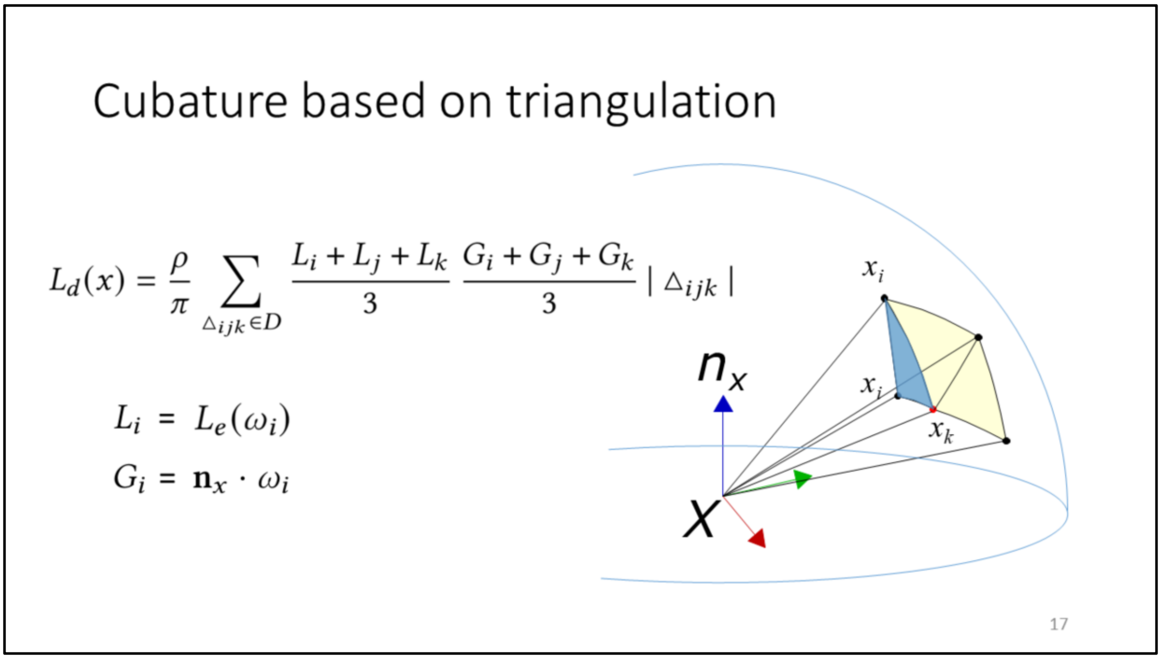
15

Issue: sample at horizon plane x where „n dot omega = 0“ would not contribute to approximation, but also results in uneven radiance weights close above



Issue: sample at horizon plane x where „n dot omega = 0“ would not contribute to approximation, but also results in uneven radiance weights close above

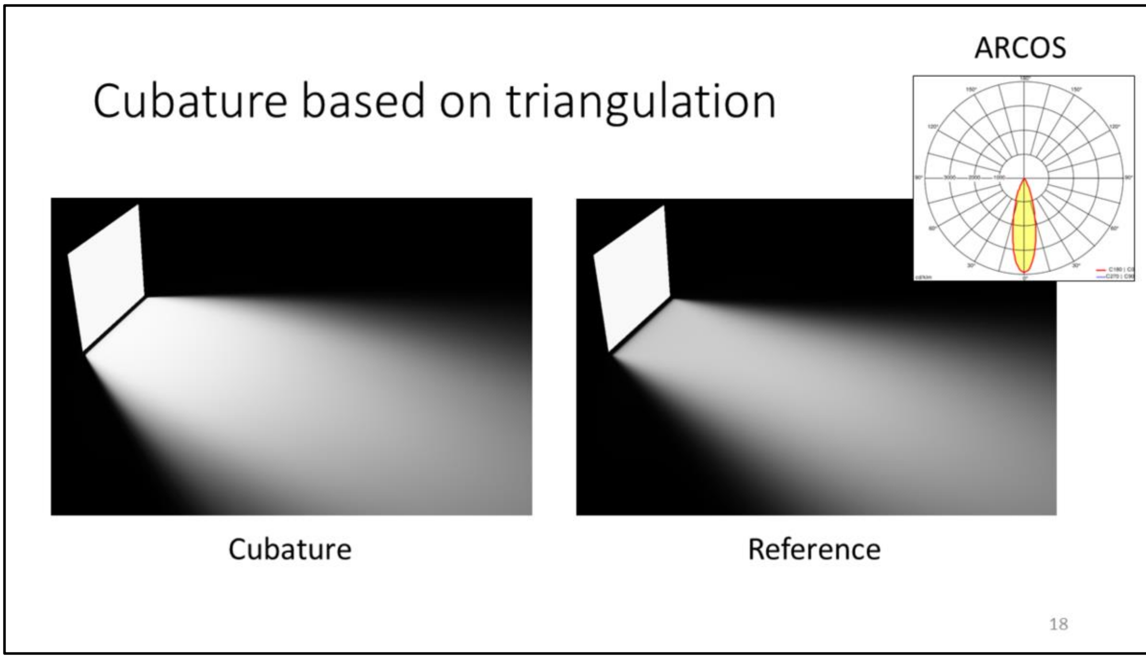
NOTE: Polygon moved even closer to ground plane to emphazie this effect



Modified cubature to include samples on horizon plane of shading point

This is the final shading calculation

First step in setup: In case the polygon intersects with the horizon plane -> clip to visible part



Much better
Intensity a bit too high and outline too wide, but no visual artifacts and totally plausible
Shows that we can handle one of the most challenging data sets (strongly directed) acceptable

Plausible approximation
Robust when light intersects with the shading plane
Sometimes patter visibile in near-field, but are not disturbing

Works in a various real lighting design scenarios where emitting polygon is modeled together with luminaire geometry
Seamless tweak to resolve singularity in mathematical formulation

Performance

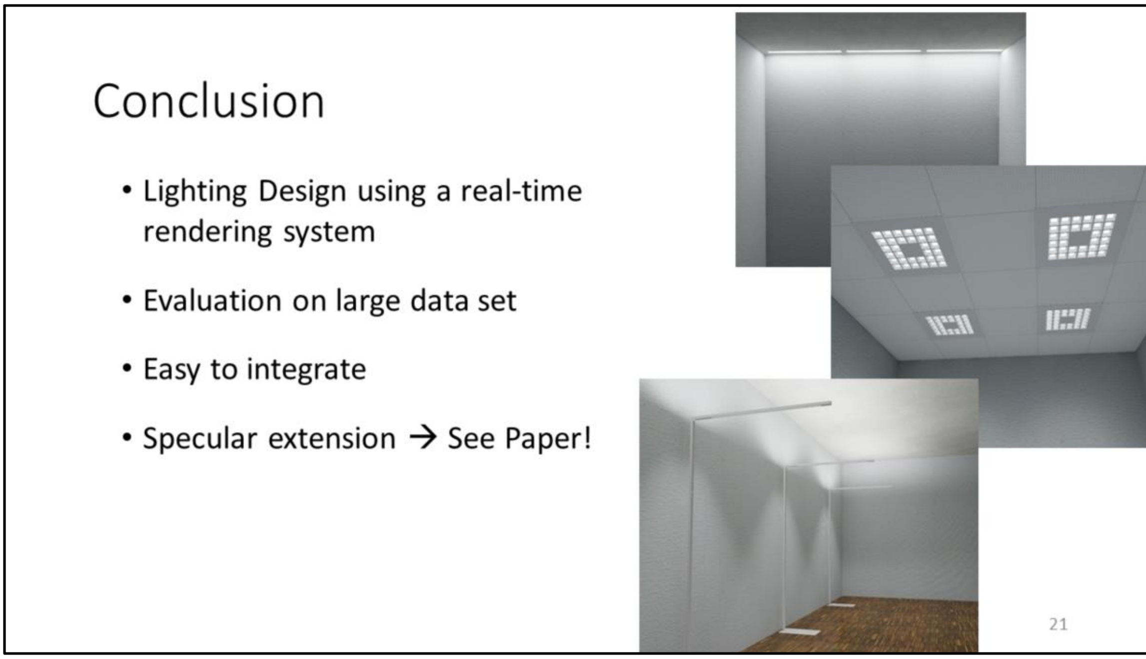
	NV RTX 2080 Ti	
Method	1080p	2160p
Cubature	**0.41 ms**	**1.70 ms**
Point	0.11 ms	0.40 ms
MC 8	0.33 ms	1.41 ms
MC 8 + DN 2x16	0.94 ms	3.02 ms

- No Visibility Test
- No Tone Mapping

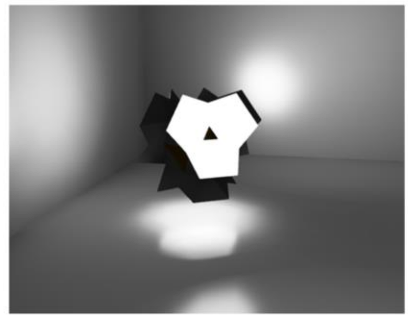
MC: Urena et al., *An Area-Preserving Parametrization for Spherical Rectangles*, CGF 2013

20

Baseline given by fixed single point representation
Our approximation possible at reasonable overhead
Visibility testing and other effects typically make most of the frame time in real-time systems



Our approximation enables real-time rendering system to be used for lighting design
No discontinuities
Robust in any constellation
Easy to integrate in a rendering pipeline
See paper for a specular shading extension based on the Linearly Transformed Cosines technique. (Heitz et al. 2016)



Thank you for your attention!

Demo & Supplemental Material: rtappl.vrvis.at