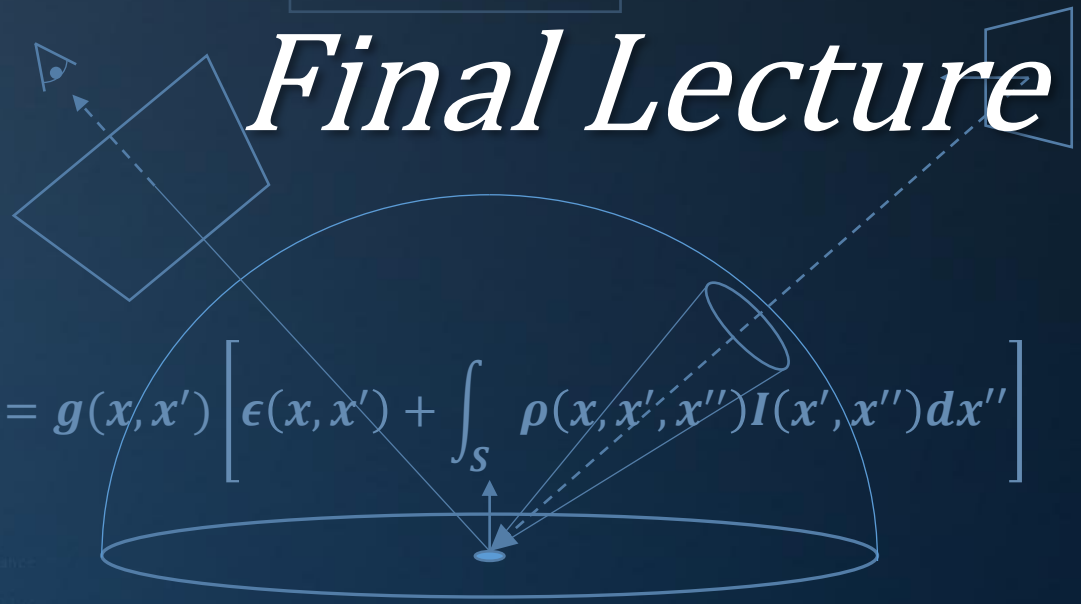
INFOMAGR – Advanced Graphics

Jacco Bikker - November 2022 - February 2023

Final Lecture - "Filtering"



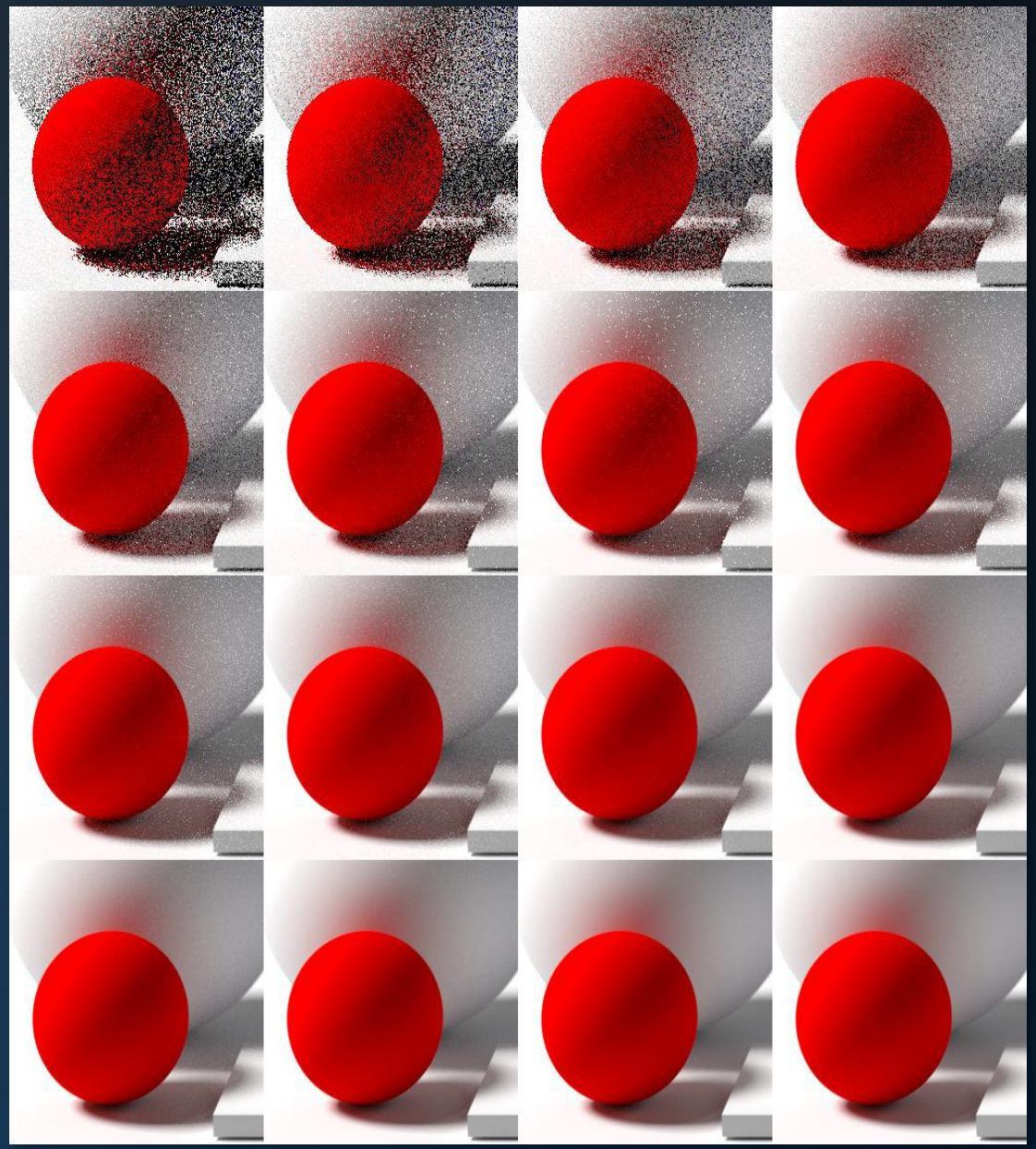
$$I(x,x') = g(x,x')\left[\epsilon(x,x') + \int_S \rho(x,x',x'')I(x',x'')dx''\right]$$

Welcome!

Today’s Agenda:

- Introduction
- Ingredients
- Future Work

Previously in Advanced Graphics...

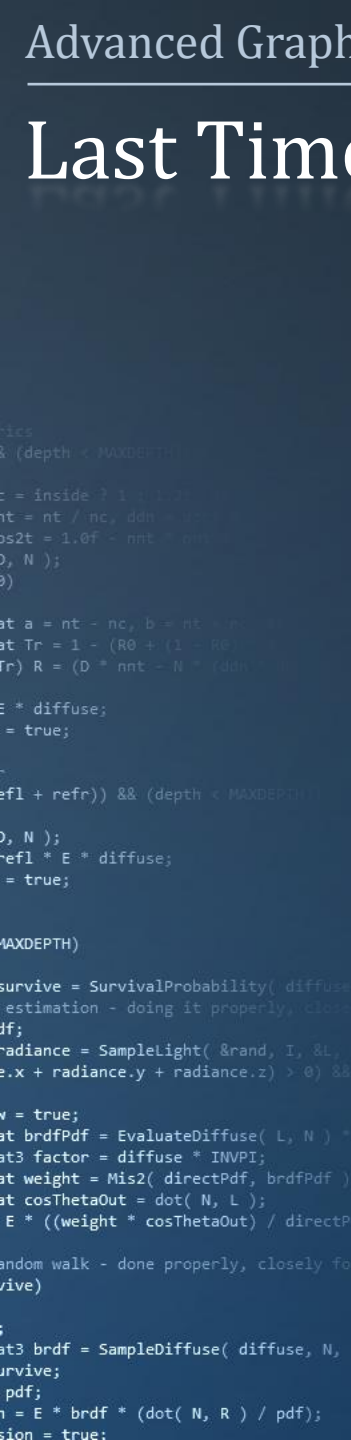


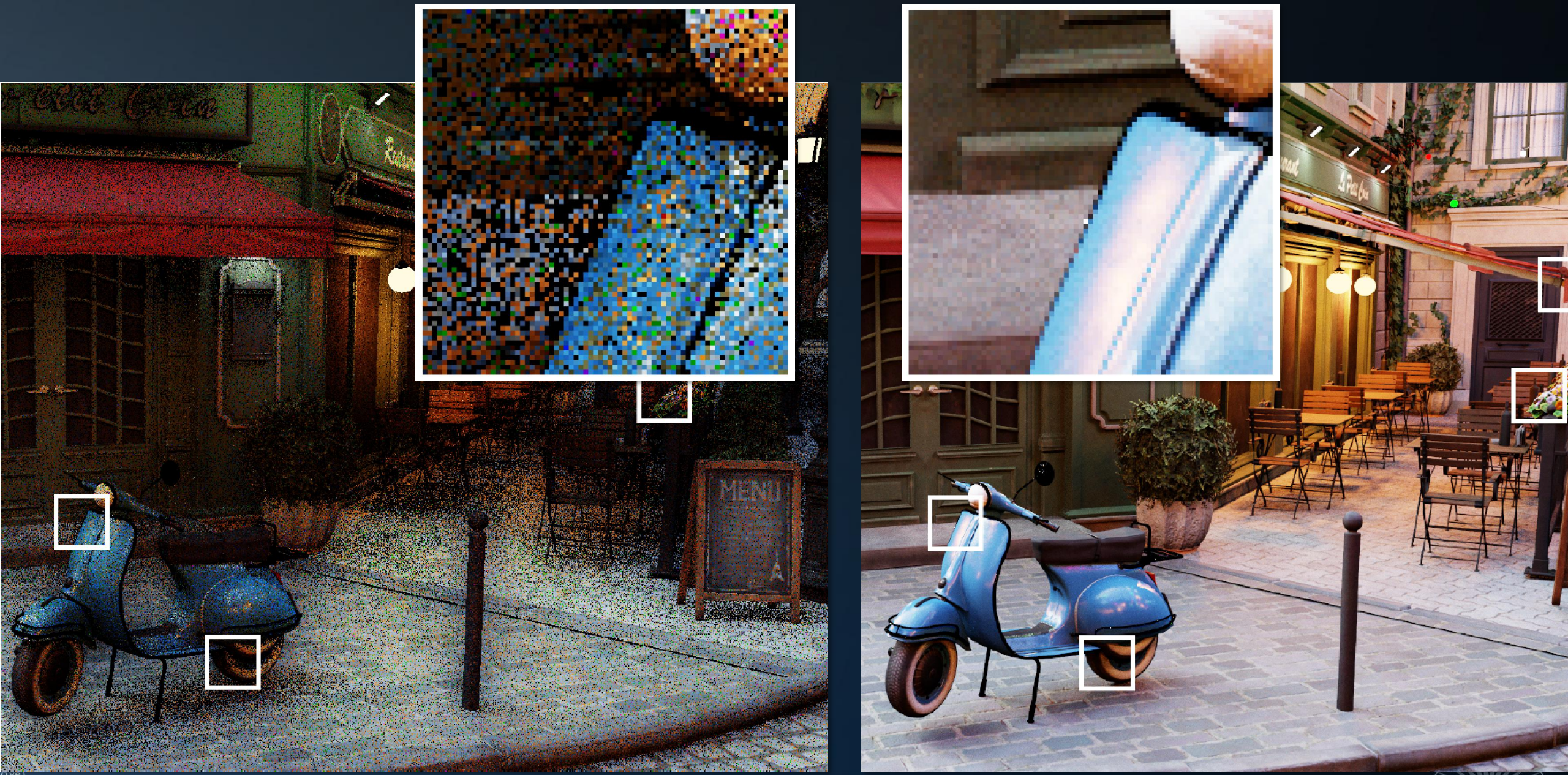
Last Time

Towards Noise-free Path Tracing

“Work smarter, not harder”: generate better samples / send rays where they matter.

Extreme case: ReSTIR, which spends a lot of effort on deciding where to send a shadow ray.





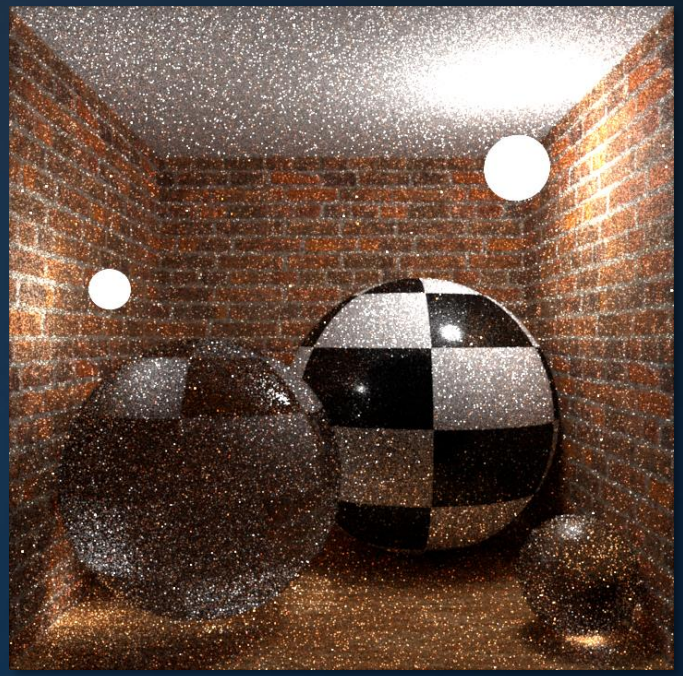
“Rearchitecting Spatiotemporal Resampling for Production”, Wyman & Panteleev, 2021.

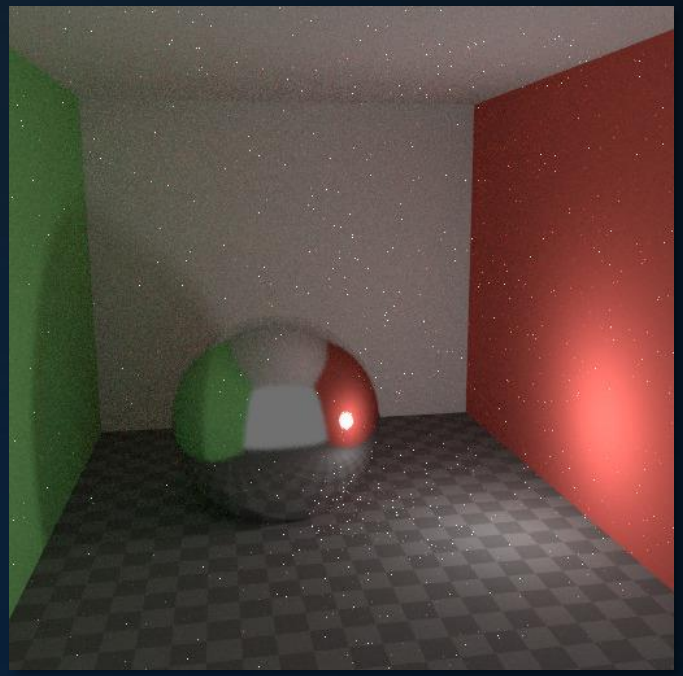
Last Time

We tried everything

…But with an 8spp budget, it's still noisy.

- There is somewhat uniform noise left
- 'Fireflies' indicate presence of 'improbable paths'.



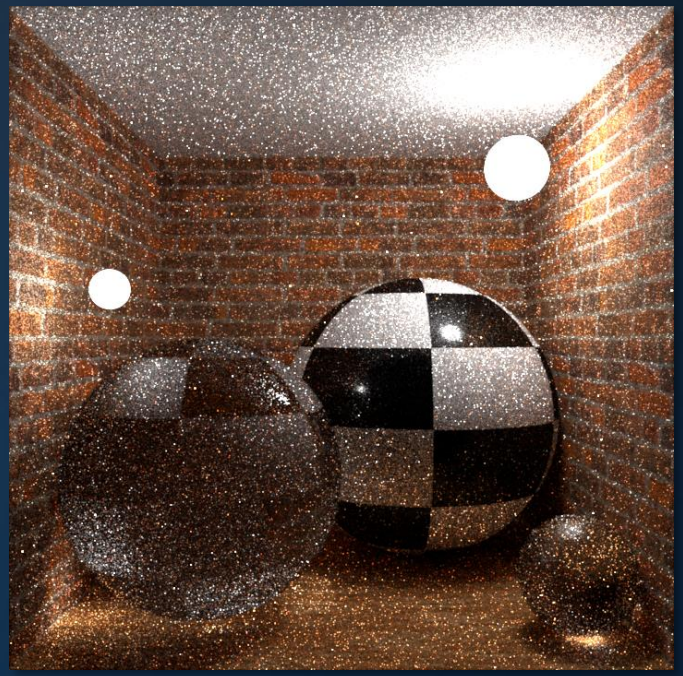


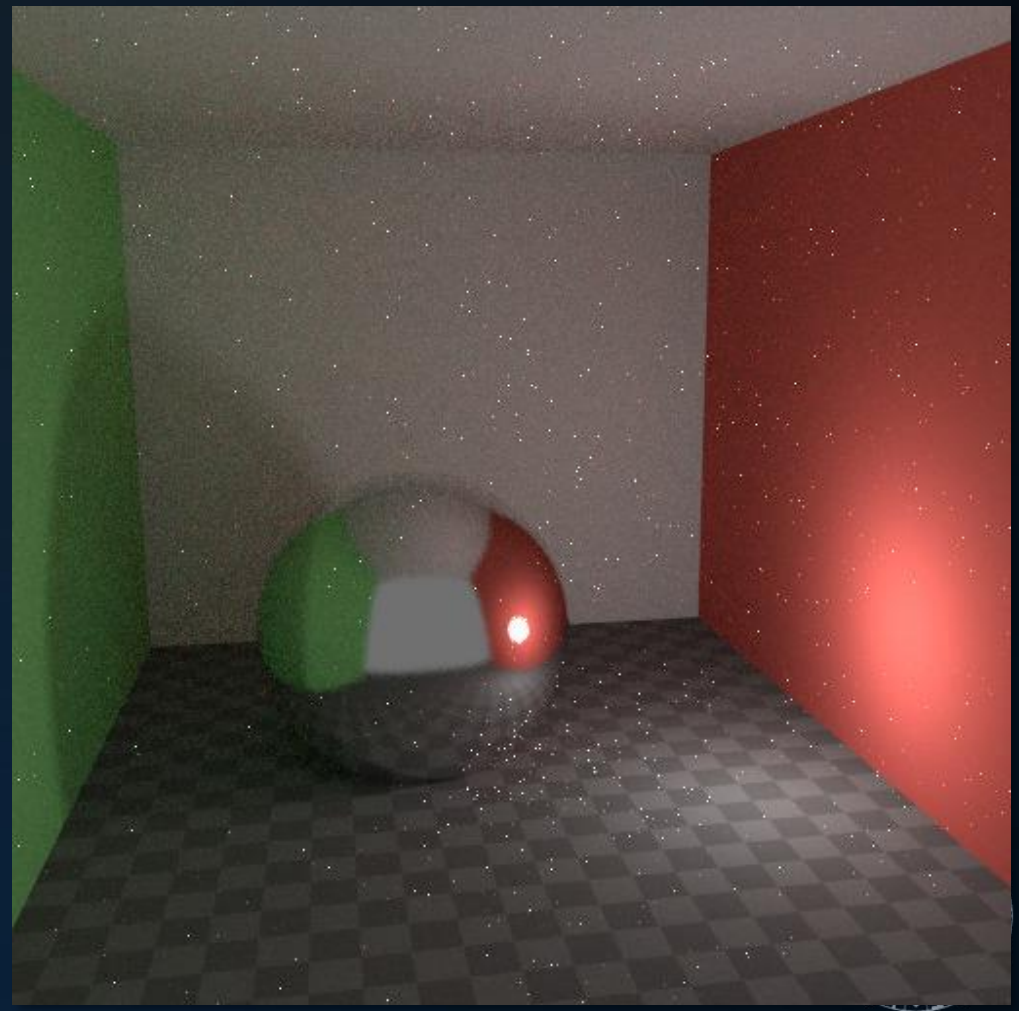
Last Time

We tried everything

…But with an 8spp budget, it's still noisy.

- There is somewhat uniform noise left
- 'Fireflies' indicate presence of 'improbable paths'.





Last Time

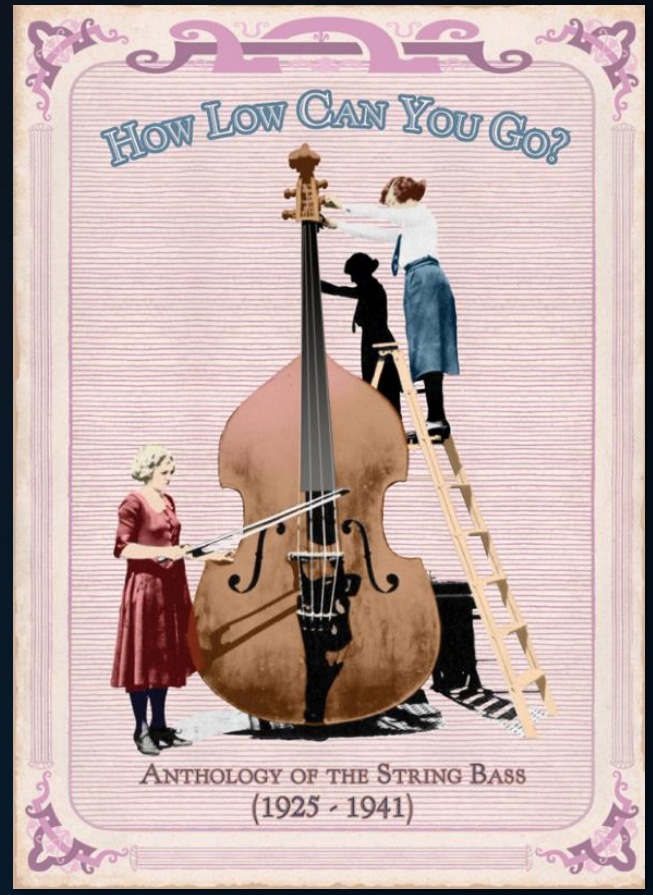
Suppressing Fireflies

"A firefly is easily recognized in the final image: it is a pixel with a value that differs significantly from its neighbors."

- Is this always true?
- How to fix it?
- Is that still correct?

Firefly suppression introduces bias in our estimator.

- Spread out the removed energy over the image / neighborhood
- Just wait it out (additional samples will improve the average)
- Do some adaptive sampling (detect high variance)
- **Just accept it.**



Last Time

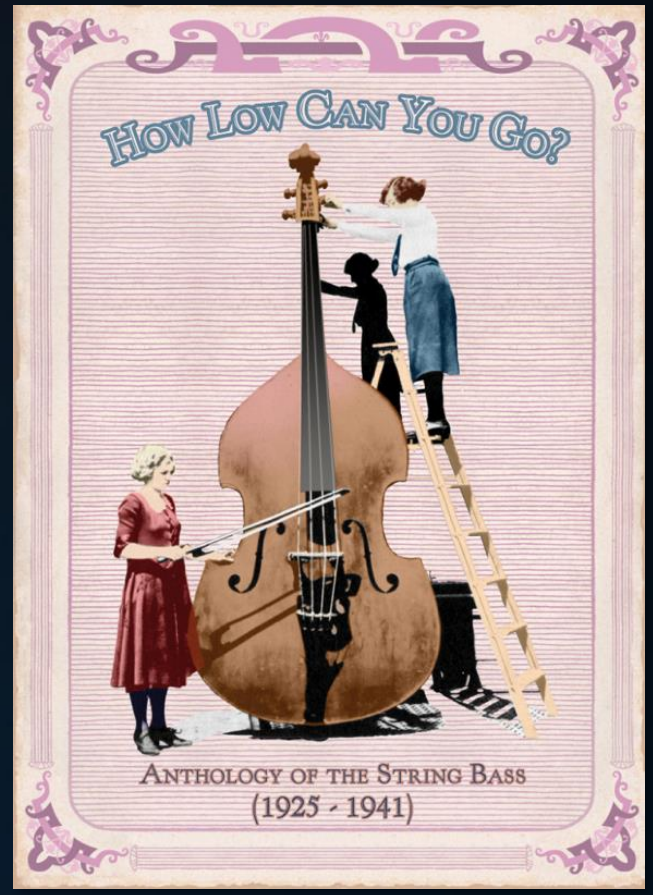
Suppressing Fireflies

“A firefly is easily recognized in the final image: it is a pixel with a value that differs significantly from its neighbors.”

Better approach: *clamp*.*

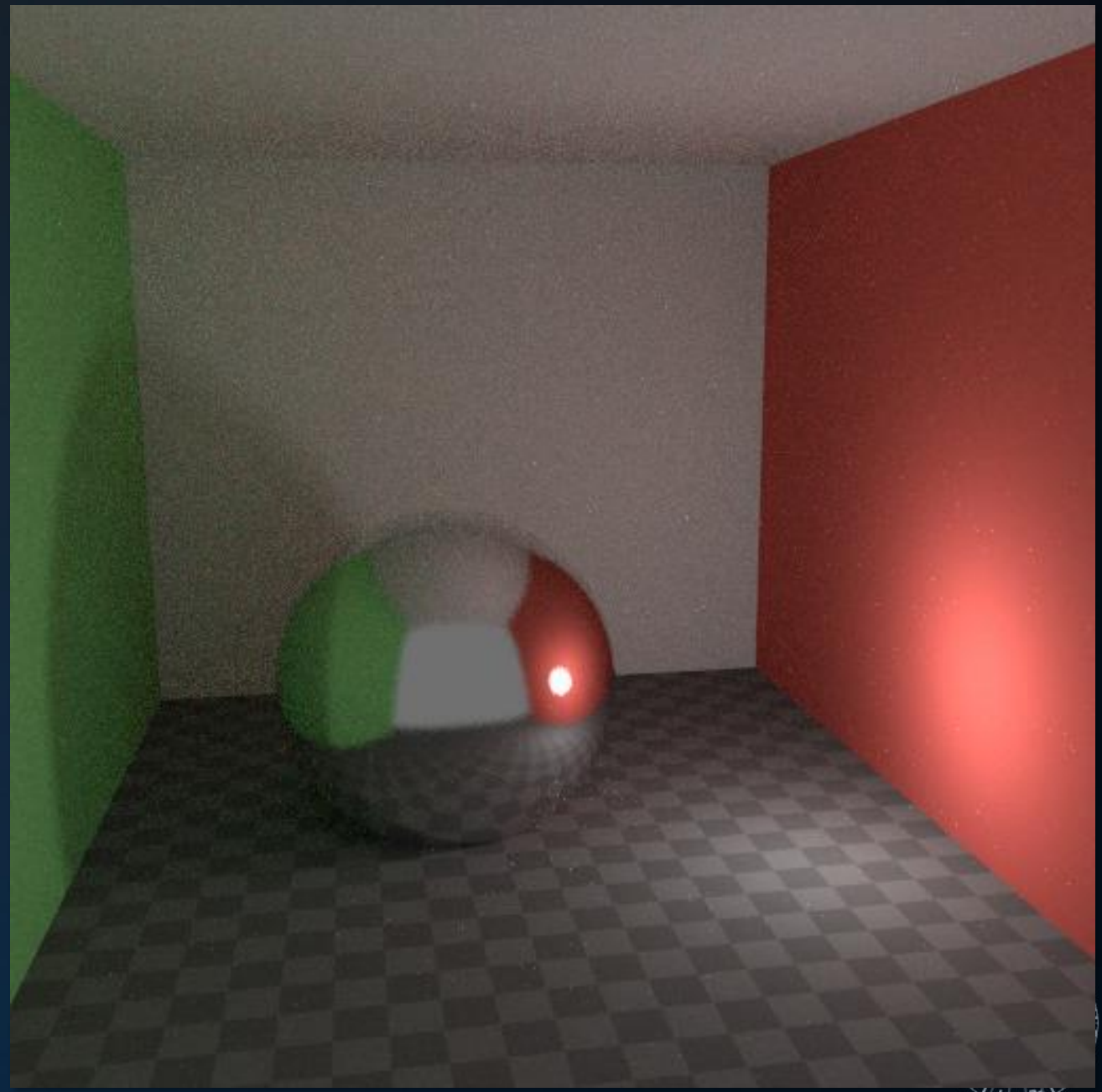
e.g., in Lighthouse 2:

```
#define CLAMPINTENSITY( E ) \
   if (dot( E, E ) > 25) E = 5 * normalize( E );
```



*: The Iray Light Transport Simulation and Rendering System, Section 5.5. Keller et al., 2017.

Last Time



Today’s Agenda:

- Noise
- Ingredients
- Future Work

Ingredients

Reducing the Problem - Filtering

Core idea:

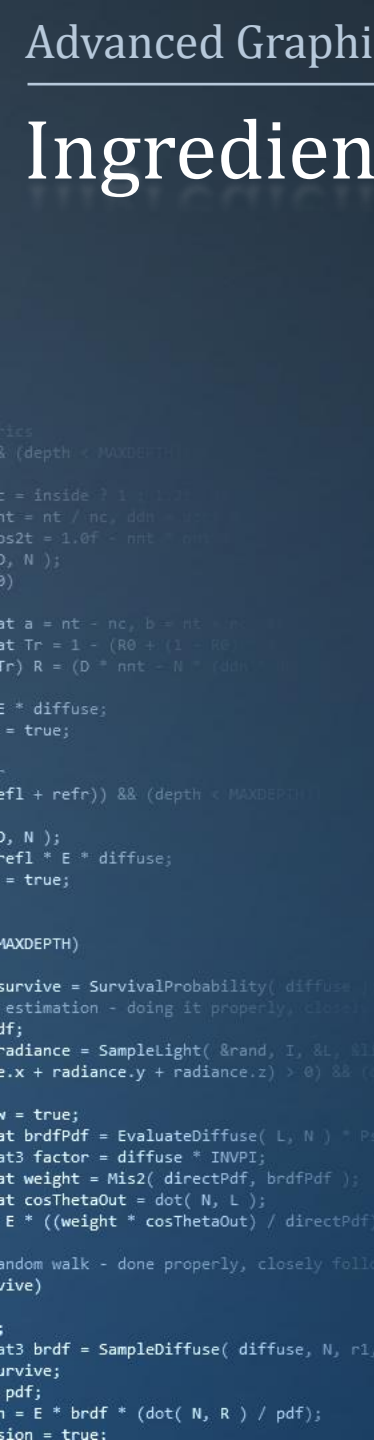
Exploit the fact that illumination is typically low-frequent: Nearby pixels tend to converge to similar values, so we should be able to use information gathered for one pixel to improve the estimate of the next.

Essentially, we are increasing the number of samples per pixel, by including the neighbors.

Note:

Unless neighboring pixels actually converge to the same value, filtering introduces bias.

Filtering thus trades variance for bias.



Ingredients

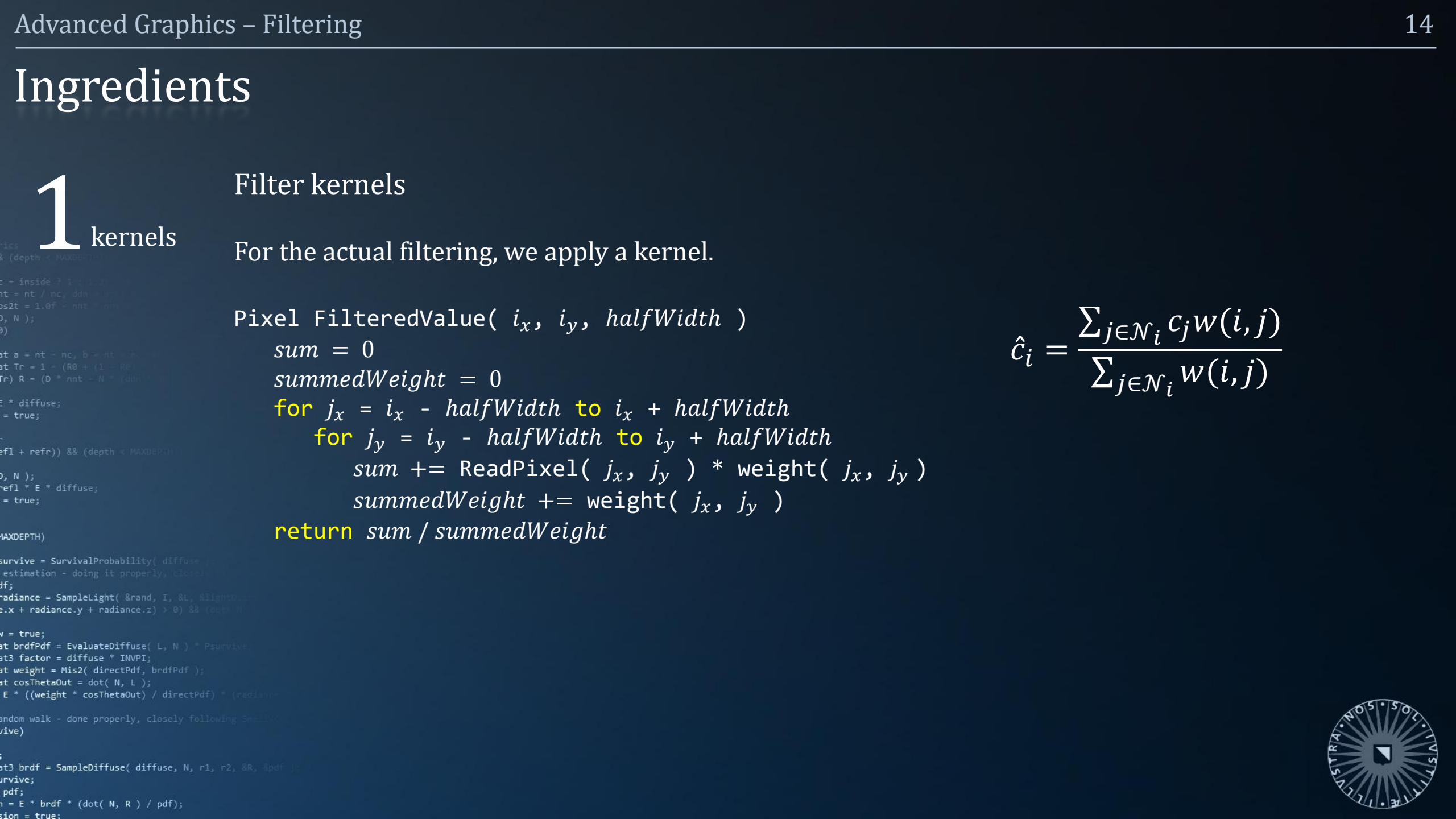
1 kernels

Filter kernels

For the actual filtering, we apply a kernel.

```
Pixel FilteredValue( i_x, i_y, halfWidth )
    sum = 0
    summedWeight = 0
    for j_x = i_x - halfWidth to i_x + halfWidth
        for j_y = i_y - halfWidth to i_y + halfWidth
            sum += ReadPixel( j_x, j_y ) * weight( j_x, j_y )
            summedWeight += weight( j_x, j_y )
    return sum / summedWeight
```

$$\hat{c}_i = \frac{\sum_{j \in \mathcal{N}_i} c_j w(i,j)}{\sum_{j \in \mathcal{N}_i} w(i,j)}$$



Ingredients

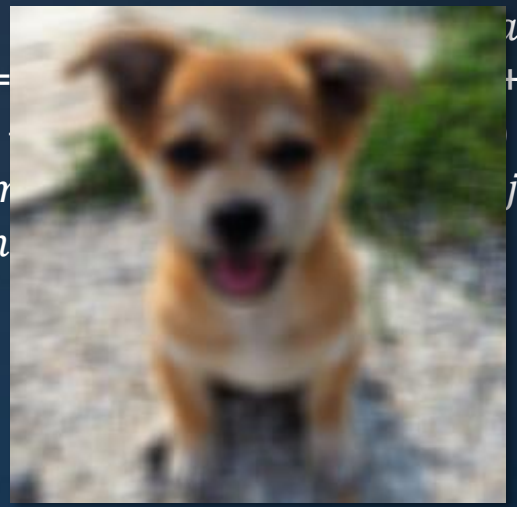
1 kernels

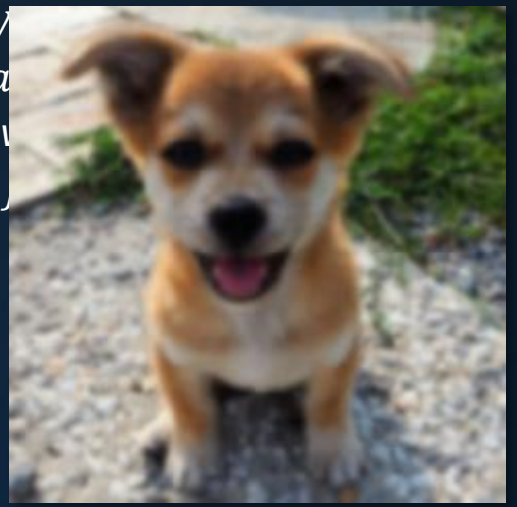
Filter kernels

For the actual filtering, we apply a kernel.

```
Pixel FilteredValue( i_x, i_y, halfWidth )
    sum = 0
    summedWeight = 0
```

$$\hat{c}_i = \frac{\sum_{j \in \mathcal{N}_i} c_j w(i,j)}{\sum_{j \in \mathcal{N}_i} w(i,j)}$$





$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{22}\begin{bmatrix} 1 & 3 & 1 \\ 3 & 6 & 3 \\ 1 & 3 & 1 \end{bmatrix}$$

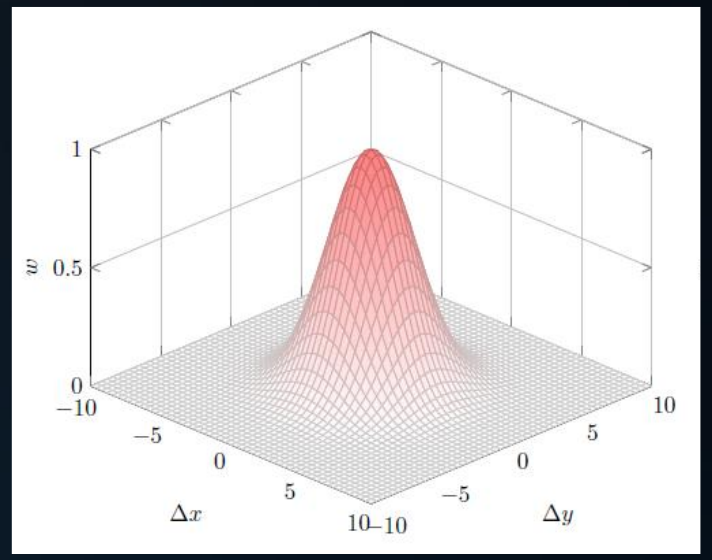
Ingredients

1 kernels

Filter kernels

For the actual filtering, we apply a kernel.

```
Pixel FilteredValue( i_x, i_y, halfWidth )
    sum = 0
    summedWeight = 0
    for j_x = i_x - halfWidth to i_x + halfWidth
        for j_y = i_y - halfWidth to i_y + halfWidth
            sum += ReadPixel( j_x, j_y ) * weight( j_x, j_y )
            summedWeight += weight( j_x, j_y )
    return sum / summedWeight
```



Here, **`weight`** or w is the weight function. We could simply use the Gaussian kernel:

$$w(i,j) = \exp\left(\frac{-\|p_i - p_j\|^2}{2\sigma_d^2}\right),$$

where p_i and p_j are screen space positions and σ_d is the spatial standard deviation of the Gaussian kernel.

Ingredients

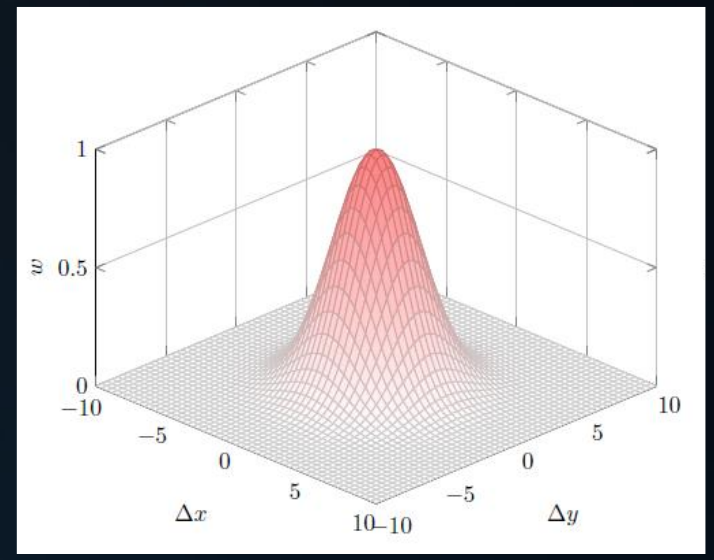
1 kernels

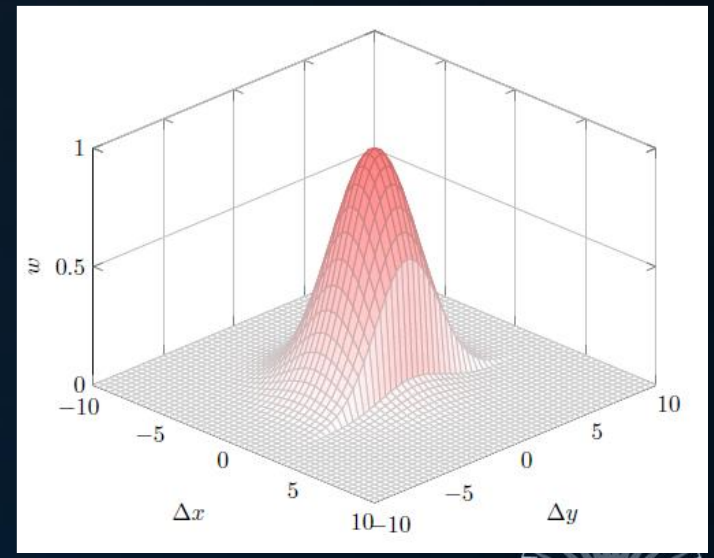
Filter kernels

A Gaussian filter (as well as other low-pass filters) blurs out high frequency details.

We can improve on this using a non-linear bilateral filter*.

$$w(i,j) = \exp\left(\frac{-\|p_i - p_j\|^2}{2\sigma_d^2}\right) \times \exp\left(\frac{-\|c_i - c_j\|^2}{2\sigma_r^2}\right)$$





*: Tomasi & Manduchi, Bilateral filtering for gray and color images. ICCV '98.

Ingredients

1 kernels

Filter kernels

The bilateral filter takes the color of nearby pixels into account. We can take this further, by taking an arbitrary set of features into account.

The cross bilateral filter*:

$$w(i,j) = \exp\left(\frac{-\|p_i - p_j\|^2}{2\sigma_d^2}\right) \times \prod_{k=1}^{K} \exp\left(\frac{-\|f_{k,i} - f_{k,j}\|^2}{2\sigma_k^2}\right)$$

Here, $f_{k,i}$ is the k'th feature vector at pixel i and σ_k is the bandwidth parameter for feature k.

Note that we can use noise-free features to smooth noisy features.
Example of a low-noise feature: normals at the primary intersection point.
Example of a noisy feature: indirect illumination at the primary intersection point.

*: Eisemann & Durand. Flash photography enhancement via intrinsic relighting. ACM Trans. Graph. 23, 3 (Aug. 2004).

Ingredients

1 kernels

Filter kernels, digest

Filtering adds samples to a pixel by ‘borrowing’ them from neighbors.
Filtering trades variance for bias.

We can improve the quality of the borrowed samples using a weight:

- Further away = less relevant
- Different normal, different material, … = less relevant

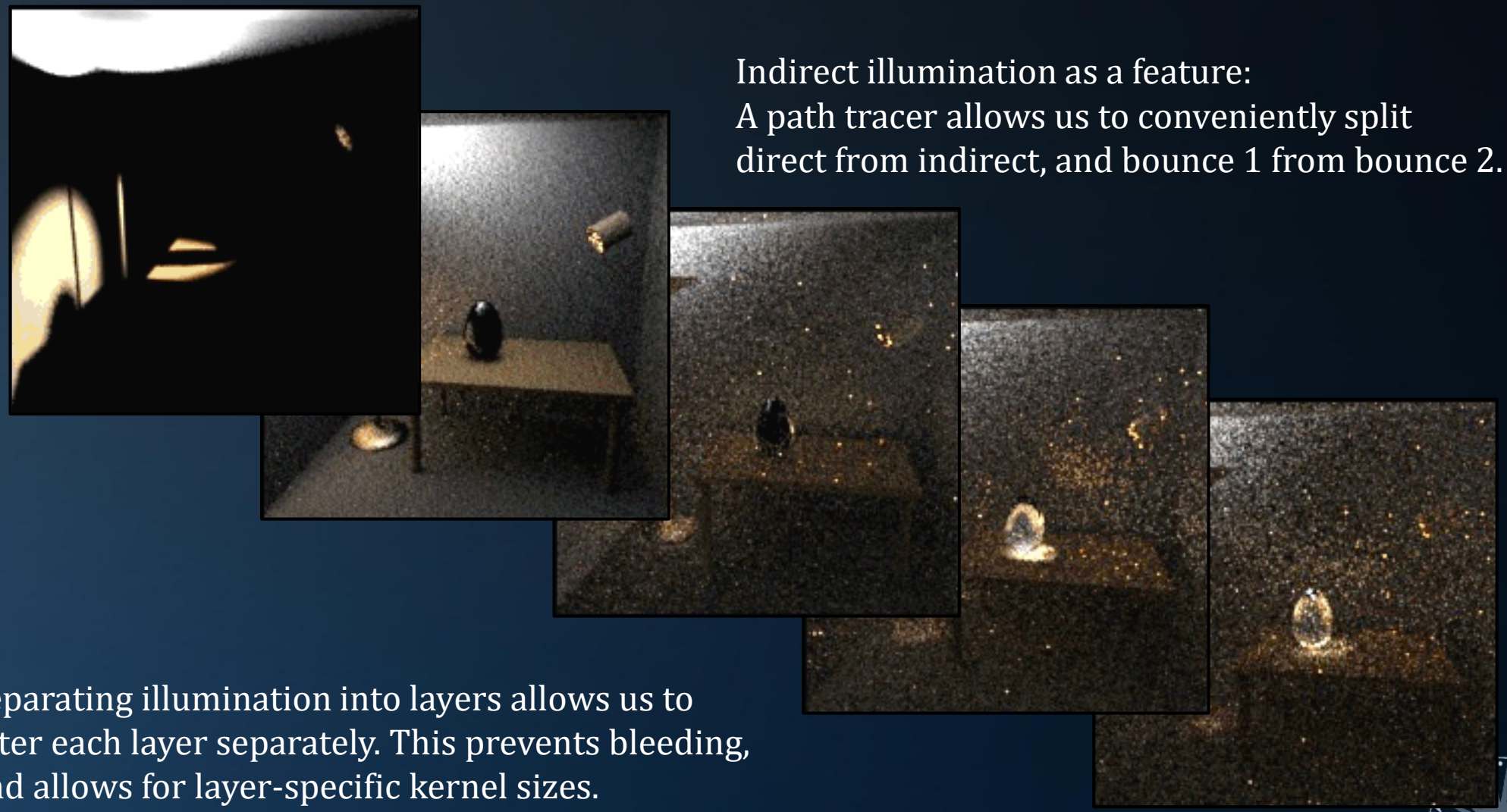
Some considerations:

- Should we take accumulated or individual samples from neighbors?
- Depth of field and AA seriously affect our options.

Ingredients

1 kernels

2 split



Indirect illumination as a feature:
A path tracer allows us to conveniently split direct from indirect, and bounce 1 from bounce 2.

Separating illumination into layers allows us to filter each layer separately. This prevents bleeding, and allows for layer-specific kernel sizes.

We can also separate albedo from illumination.

Ingredients

1 kernels

2 split

Separating albedo from illumination

Adding this separation to an existing renderer:

- store albedo at the primary intersection (simple material property);
- at the end of the pipeline: illumination = sample / max(epsilon, albedo).

Ingredients

1 kernels

2 split

3 temporal

Reprojection

Core idea:

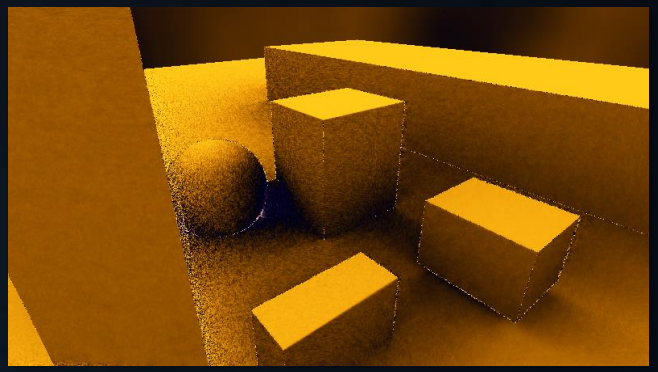
In an animation, samples taken for the previous frame are meaningful for the current frame. *We can supply the filter with more data by looking back in time.*

Ingredients

1 kernels

2 split

3 temporal



https://www.shadertoy.com/view/ldtGWl

Reprojection

Core idea:

In an animation, samples taken for the previous frame are meaningful for the current frame. *We can supply the filter with more data by looking back in time.*

Problem: in an animation, the camera and/or the geometry moves. We need to find the location of a pixel in the previous frame(s).

Solution: use the camera matrices.

$$M_{4x4}\begin{pmatrix} x_{world} \\ y_{world} \\ z_{world} \\ 1 \end{pmatrix} = \begin{pmatrix} x_{screen} \\ y_{screen} \\ z_{screen} \\ 1 \end{pmatrix} \quad \rightarrow \quad M_{4x4}^{-1}\begin{pmatrix} x_{screen} \\ y_{screen} \\ z_{screen} \\ 1 \end{pmatrix} = \begin{pmatrix} x_{world} \\ y_{world} \\ z_{world} \\ 1 \end{pmatrix}$$

(finally, apply the matrix of the previous frame to obtain the screen location in the previous frame.)

Ingredients

1 kernels

2 split

3 temporal

Reprojection

Reprojection using camera matrices:

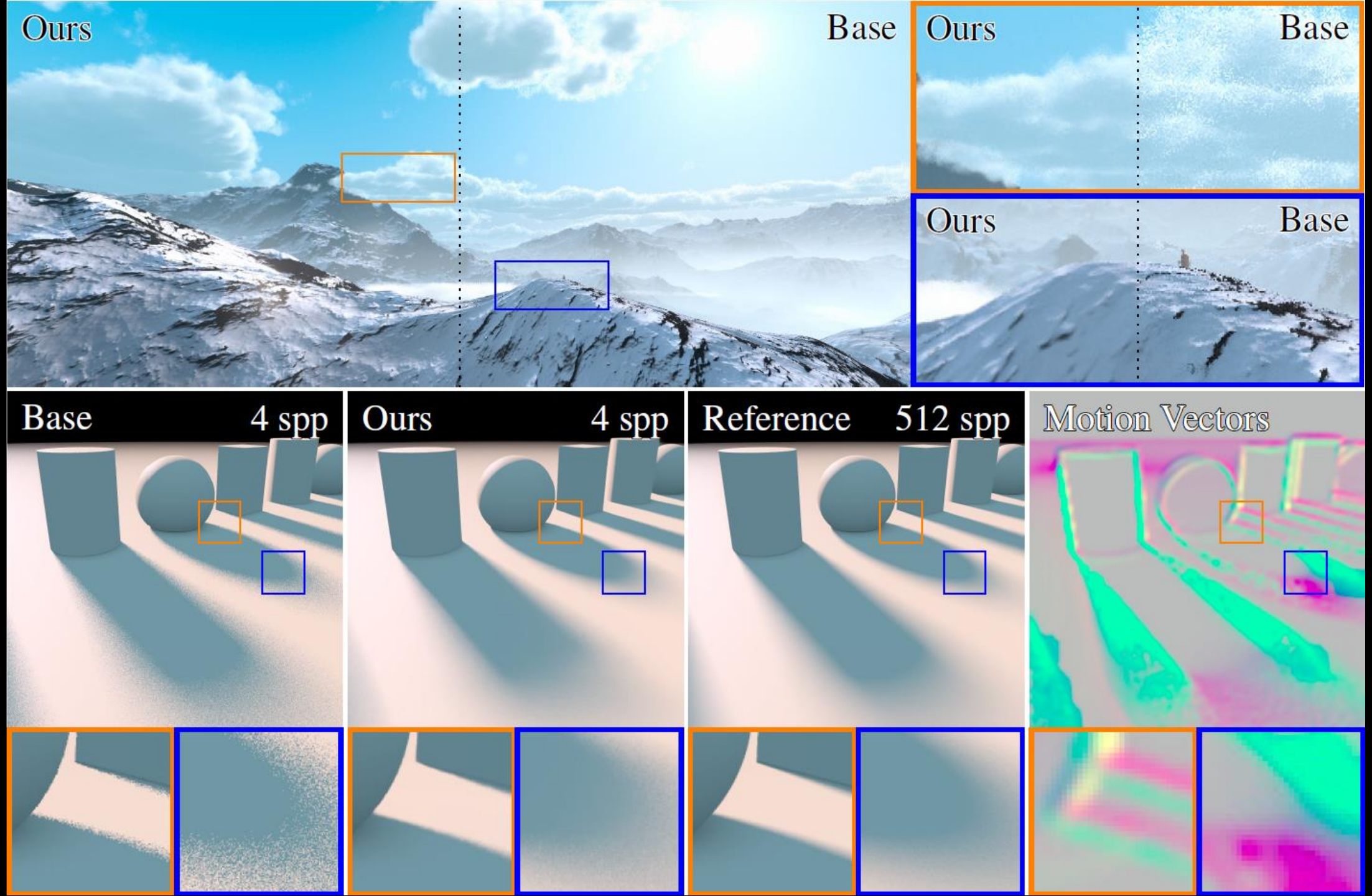
- fails if we have animation
- will not work with depth of field
- will not work with speculars.

A recent paper proposes an alternative*:

For each pixel (i,j), find the shift to similar pixels in the neighborhood by comparing a small patch of pixels around (i,j) to pixels at some distance.

Note: this idea is not new, but the paper makes it efficient using a hierarchical process, where down-sampled versions of the image are used to increase the size of the search window.

*: Fast Temporal Reprojection without Motion Vectors. Hanika & Tessari, 2021.


Ours
Base
Ours
Base
Ours
Base
Base
4 spp
Ours
4 spp
Reference
512 spp
Motion Vectors

Ingredients

1 kernels

2 split

3 temporal

Caching in world space

Instead of searching the current pixel in the previous frame in screen space, we can also maintain a cache in world space*.

Path space filtering:

- Store information in a 3D grid
- Map the grid cells to a hash map
- Update grid cells for each vertex that ‘visits’ it

Note that a single cell may still receive shading information for surfaces with different normals.

*: Binder et al., Massively Parallel Path Space Filtering, 2019.

Massively Parallel Path Space Filtering

Nikolaus Binder, NVIDIA
Sascha Fricke, University of Braunschweig
Alexander Keller, NVIDIA

Figure 1: Path tracing at one path per pixel (left) in combination with hashed path space filtering (middle) very closely approximates the reference solution using 1024 paths per pixel (right). The filtering takes about 1.5 ms in HD resolution. Scene credit: Amazon Lumberyard Bistro, Open Research Content Archive [8].

ABSTRACT

Restricting path tracing to a small number of paths per pixel for performance reasons rarely achieves a satisfactory image quality for scenes of interest. However, path space filtering may dramatically improve the visual quality by sharing information across neighboring pixels that are considered as proximate. Unlike screen space-based

networks [1, 2], hierarchical filtering with weights based on estimated variance in screen space [11] or on improving performance by simplifying the overall procedure [9].

Path space filtering [7], on the other hand, averages contributions of light transport paths in path space, which allows for filtering at non-primary intersections and for a more efficient handling of dis-occlusions during temporal filtering. However, querying contributions in path space so far had been significantly more expensive than filtering the contributions of neighboring pixels in screen space. Neglecting the fact that locations that are in screen space are not necessarily adjacent in screen space [4]. As a consequence,

[cs.GR] 15 Feb 2019

Ingredients

1 kernels

2 split

3 temporal

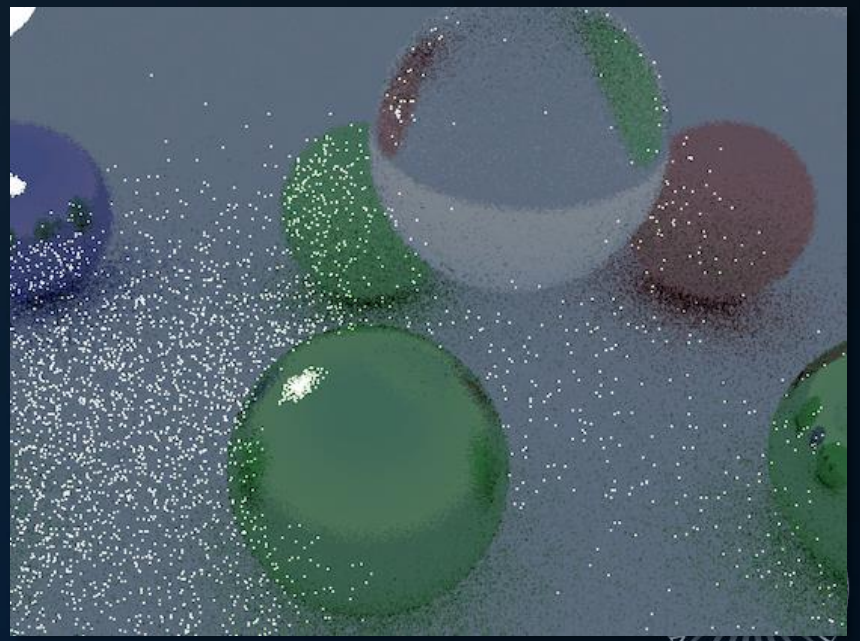
4 adaptive

Adaptive Sampling

Some pixels need more samples than others.
(to reach a certain variance level)

Adaptive Sampling* aims to estimate which pixels still need work.

Note that reliable variance estimation requires more than a few samples; adaptive sampling is generally not applicable to realtime rendering.



*: A Survey of Adaptive Sampling in Realistic Image Synthesis, M. Sik, 2013.

Ingredients

1 kernels

2 split

3 temporal

4 adaptive

Variance-guided Filtering*

A variance estimate is also useful for steering the filter kernel size:

- A pixel with low variance can use a small kernel
  (which prevents overblurring)
- A pixel with high variance needs a larger kernel
  (to include more samples from neighbors)

SVGF combines bilateral filtering with variance guided kernel sizes and temporal reprojection.



*: Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination. Schied et al., 2017.

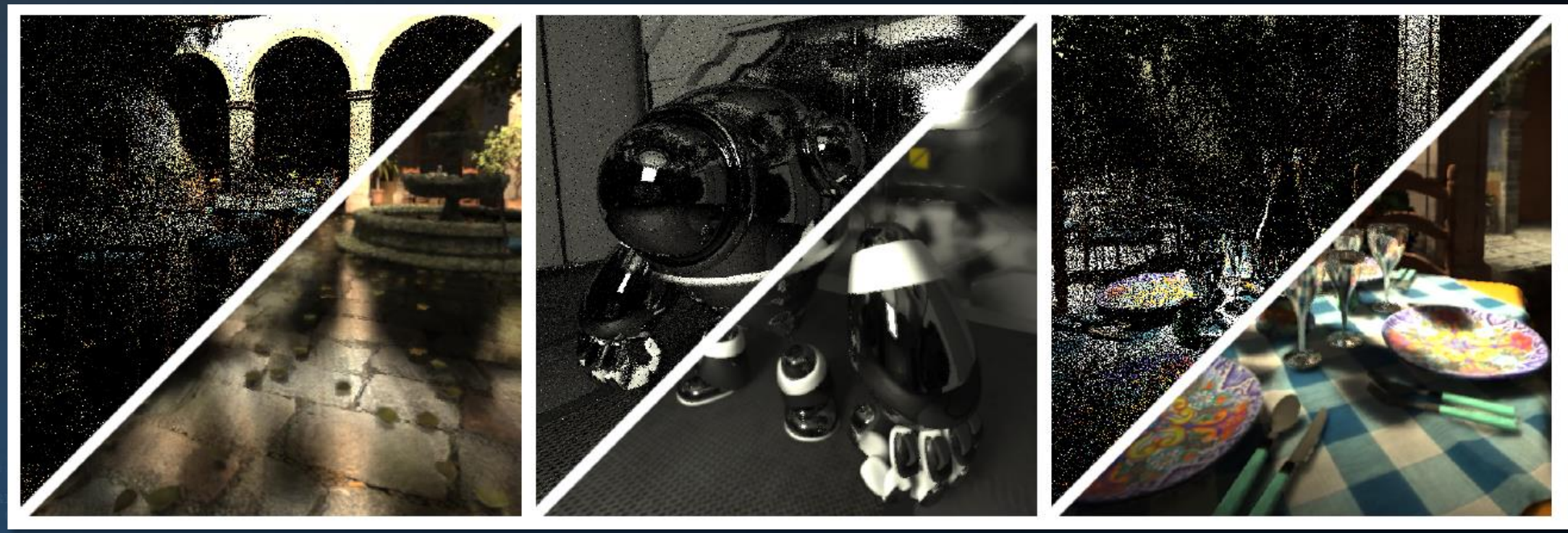
Ingredients

1 kernels

2 split

3 temporal

4 adaptive



https://studenttheses.uu.nl/handle/20.500.12932/29727

Ingredients

1 kernels

2 split

3 temporal

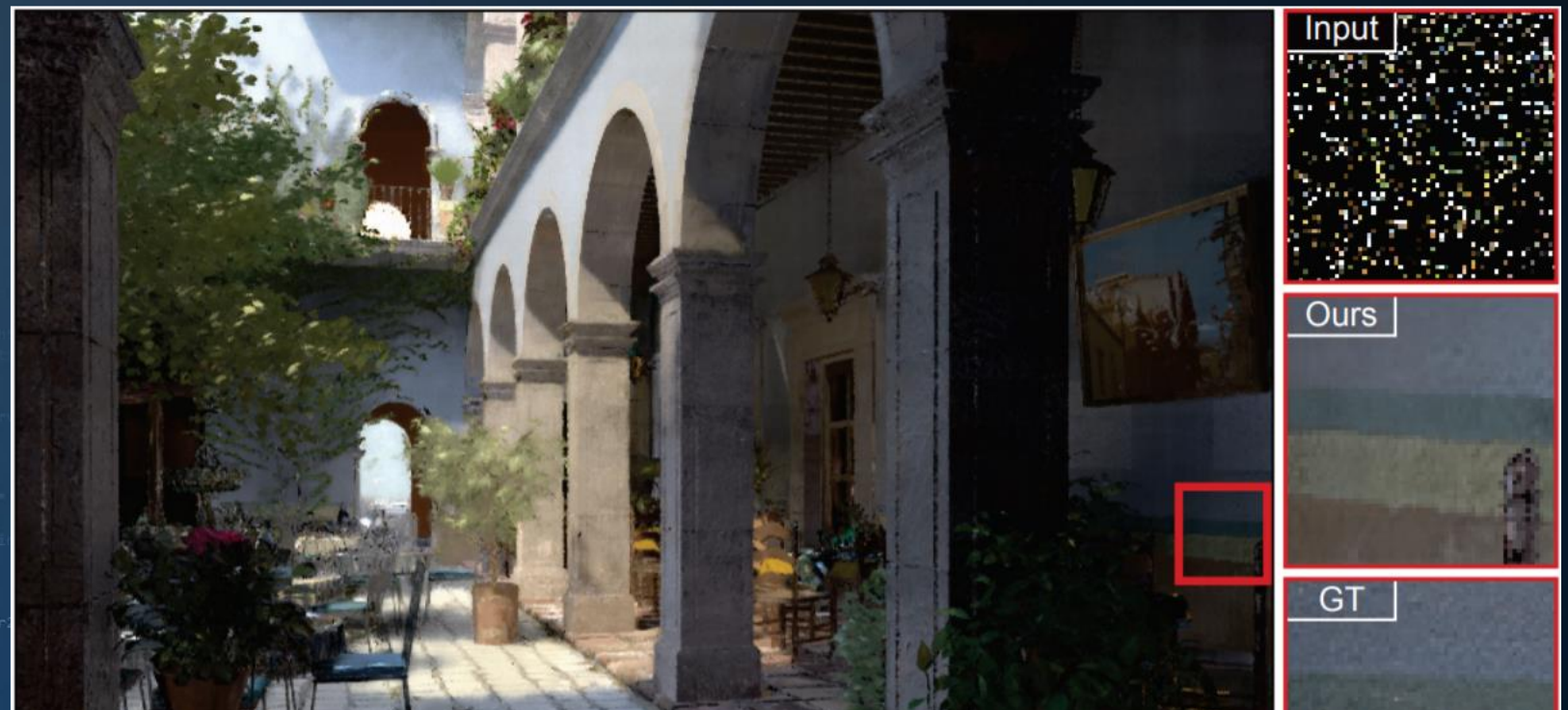
4 adaptive

5 learning

Machine Learning

Neural networks can be used to filter path tracing noise.

E.g., by learning optimal filter parameters:
A Machine Learning Approach for Filtering Monte Carlo Noise. Kalantari et al., 2015.



Ingredients

1 kernels

2 split

3 temporal

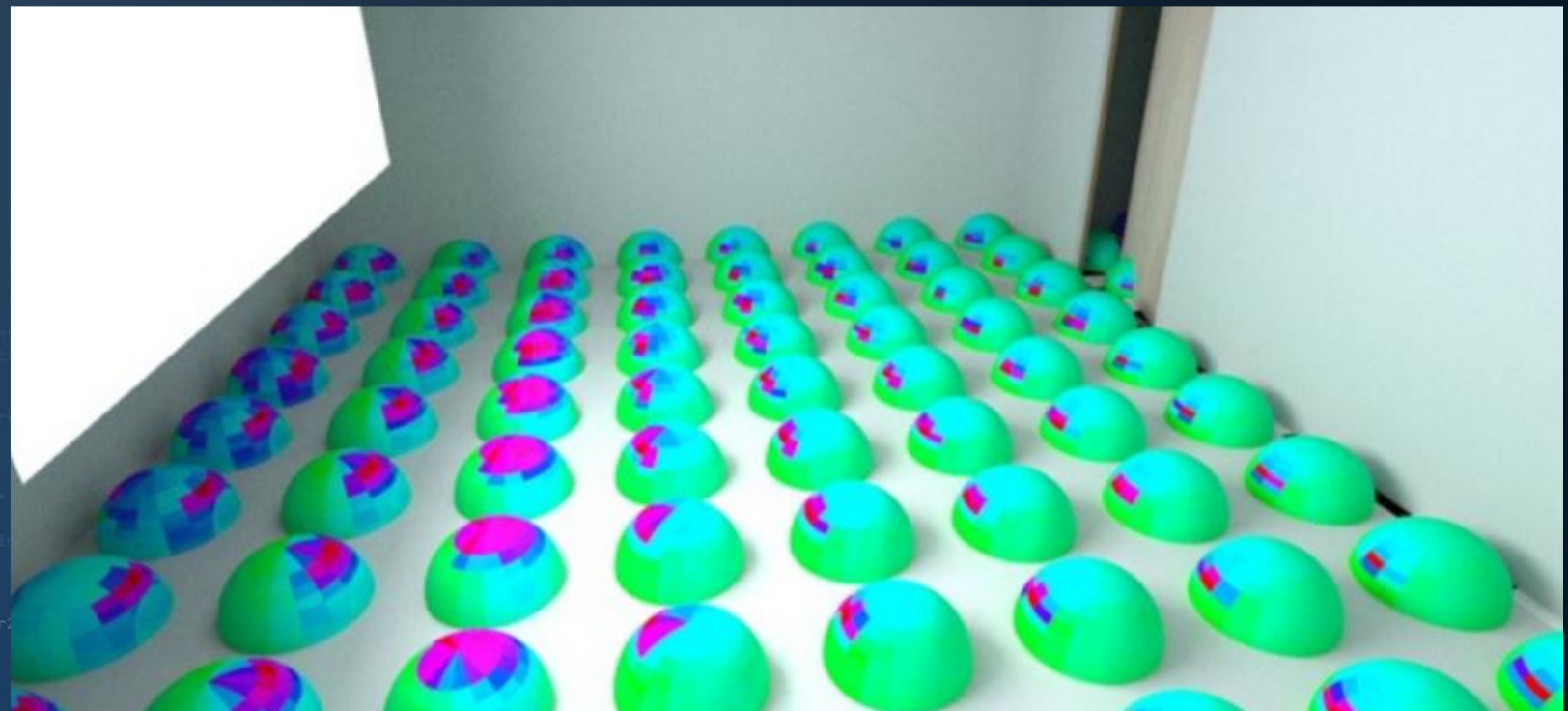
4 adaptive

5 learning

Machine Learning

Reinforcement Learning can be used to importance sample based on experience.

E.g., by learning light transport while rendering:
Learning Light Transport the Reinforced Way. Dahm & Keller, 2017.



Ingredients

1 kernels

2 split

3 temporal

4 adaptive

5 learning

Machine Learning

Reinforcement Learning can be used to importance sample based on experience.

Reinforcement Learning for rendering is often referred to as path guiding:
Path Guiding in Production. Vorba et al., 2019 (SIGGRAPH 2019 course).

Ingredients

1 kernels

2 split

3 temporal

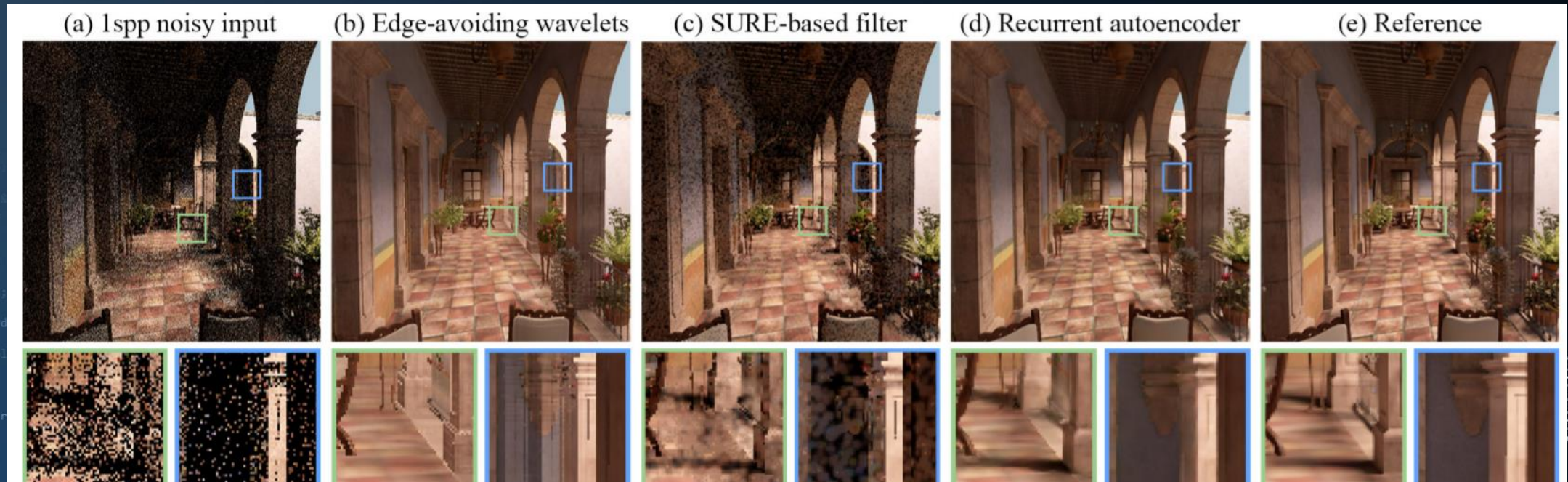
4 adaptive

5 learning

Machine Learning

And finally: convolutional neural networks.

Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings. Disney / Pixar, University of California: Bako et al., 2019.
Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder. NVIDIA, several universities: Chaitanya et al., 2017.



Today’s Agenda:

- Noise
- Ingredients
- Future Work

Digest

Filtering, practical

First of all, provide a high-quality render:

- Few samples can still be HQ samples
- Many filters get more expensive with high spp counts ➔ spend more time per sample

Prepare your input:

- Separate albedo and illumination
- Separate direct and indirect light
- Suppress outliers
- Supply ‘feature buffers’ for the bilateral kernels
- Use a pinhole camera - postpone AA / DOF
- Reproject; go temporal.

Filter:

- Some form of bilateral
- Steer kernel size with variance estimation
- Ideally: sample-based; pixel-based if this is too slow

Digest

Filtering, open problems

Not easy to do:

- DOF, AA
- Transparency

Considerations for real-time:

- Mind temporal stability
- Don’t make it too crisp
- Make some (uniform) noise a feature
- Consider using DLSS


Camera parameters
Material parameters
name: unnamed
R:255 G:255 B:255 color
R: 0 G: 0 B: 0 absorption
0.000 metallic
0.000 subsurface
0.000 specular
0.000 roughness
0.000 specularTin
0.000 anisotropic
0.000 sheen
0.000 sheenTint
0.000 clearcoat
0.000 clearcoatGl
0.000 transmissio
0.000 eta (1/ior)

Today’s Agenda:

- Noise
- Ingredients
- Future Work

INFOMAGR – Advanced Graphics

Jacco Bikker - November 2022 – February 2023

END of “Filtering”

next lecture: “Bits & Pieces, Exam Training”

Exam Questions

A few questions on Importance Sampling

a) Next Event Estimation can be considered a form of Importance Sampling. How?

b) Is Russian Roulette also a form of Importance Sampling?

c) Explain why the following path termination probability for Russian Roulette is a poor choice:

$$p = \frac{r + g + b}{3}$$

d) Write down a better one. Motivate your choice.

e) 'Without Russian Roulette, a path tracer cannot be unbiased'. Is this statement true or false? Why?

f) Describe a scene for which the use of Next Event Estimation increases variance.

Exam Questions

Sampling:

a) How does stratification reduce variance?

b) How does blue noise reduce variance?

c) What is, in the context of stratification, the ‘curse of dimensionality’?

Exam Questions

On GPU ray tracing:

a) In wavefront path tracing, we split up rendering over multiple kernels. Why?

b) List the kernels for wavefront path tracing, and describe their function.

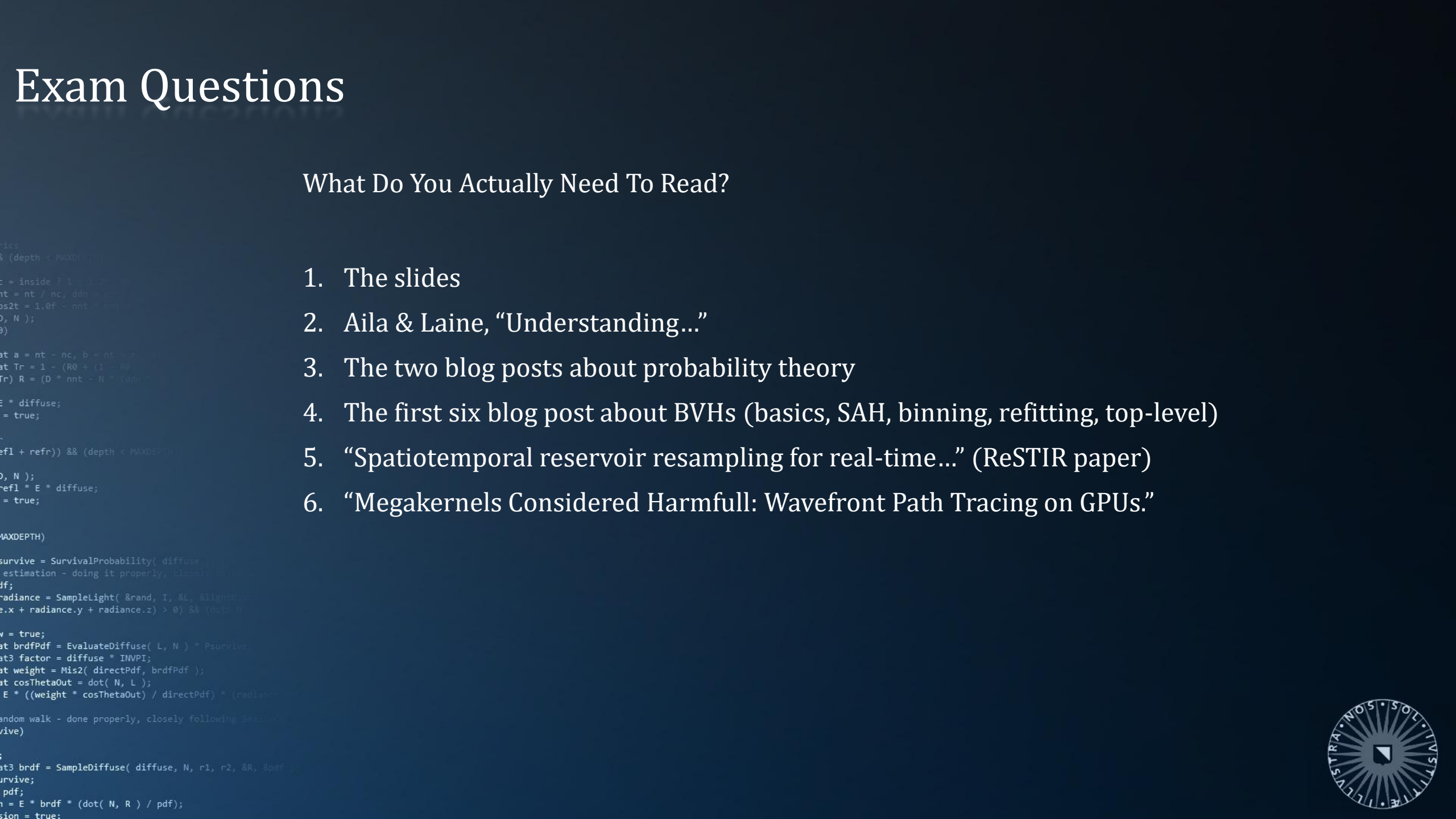
c) Why did early GPU ray tracers try to work without a stack?

d) In the paper “Understanding the Efficiency of Ray Traversal on GPUs”, Aila and Laine conclude that memory bandwidth is not the main bottleneck in GPU ray tracing. What is the bottleneck?

Exam Questions

What Do You Actually Need To Read?

1. The slides
2. Aila & Laine, “Understanding…”
3. The two blog posts about probability theory
4. The first six blog post about BVHs (basics, SAH, binning, refitting, top-level)
5. “Spatiotemporal reservoir resampling for real-time…” (ReSTIR paper)
6. “Megakernels Considered Harmfull: Wavefront Path Tracing on GPUs.”



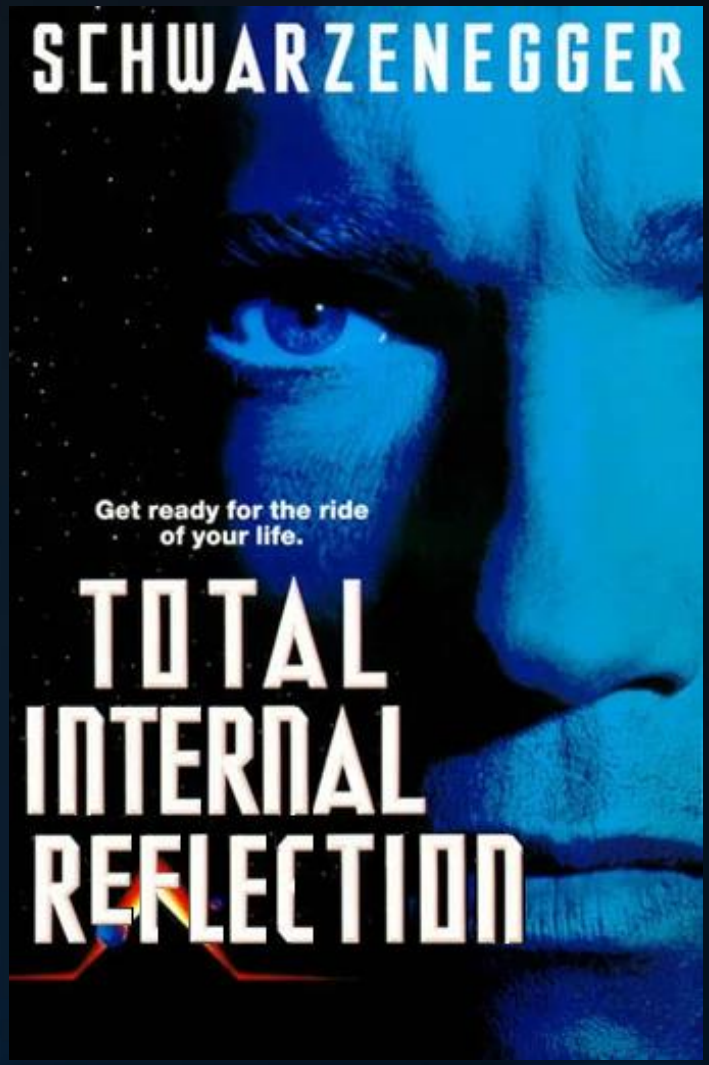
Exam Questions

A few questions on BRDFs:

a) In microfacet BRDFs, we make extensive use of a ‘halfway vector’. What is this vector, and what is it used for?

b) The Phong illumination model, as used in OpenGL, is not physically plausible, for a number of reasons. Name at least two.

c) What is Total Internal Reflection?



Exam Questions

Probability Densities

We set up an experiment where we roll two dice. The first die is a regular six-sided one, with numbers 1…6. The second die is also six-sided, but has numbers 10…60. For the experiment, we pick up a random die and roll it.

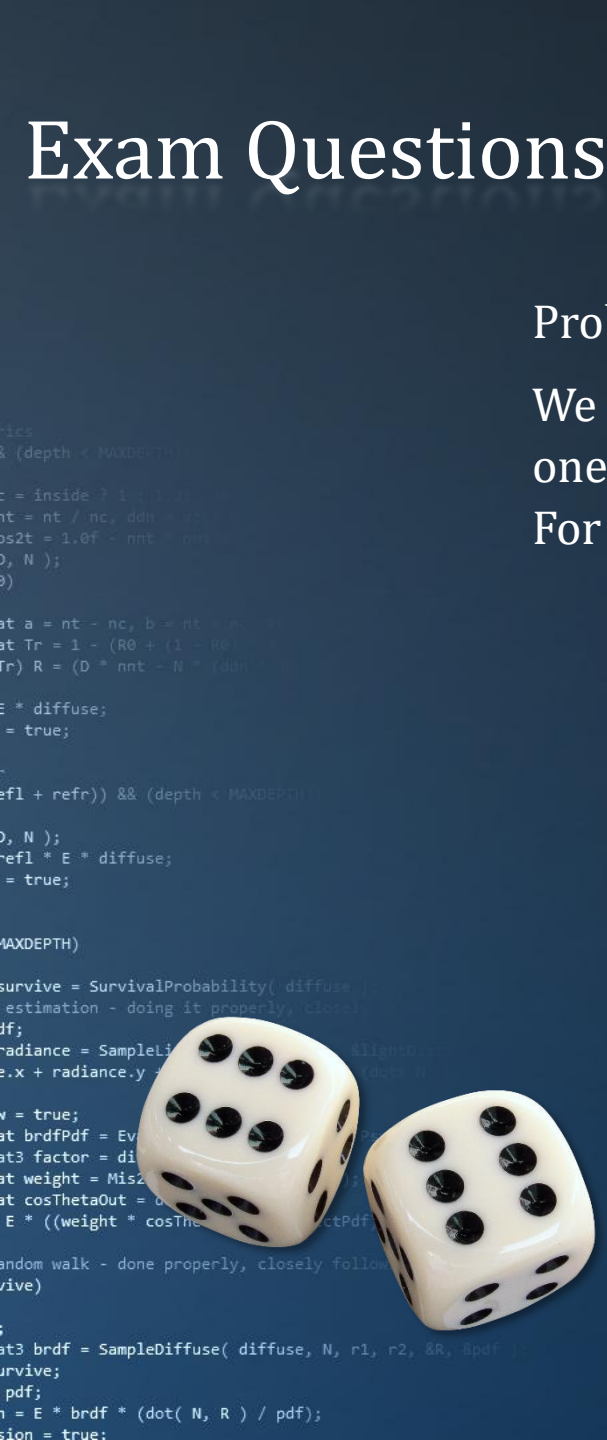
a) What is the expected value of this experiment?

b) Explain how we can optimally use importance sampling for this experiment to reduce variance.

c) What are the requirements for a valid pdf?

d) Write down the pdf used in 4b and show that it is indeed correct.

e) When using Next Event Estimation, we sample the direct illumination with an explicit light ray. The energy that this sample yields is scaled by $1/pdf(X)$, as we normally do with Monte Carlo sampling. Here, $pdf(X)$ is the light pdf. Write down this pdf and show that its integral over the hemisphere equals 1.



Exam Questions

When using *Next Event Estimation* in a path tracer, *implicit light connections* do not contribute energy to the path.

a) What is an ‘implicit light connection’?

b) Why do these connections not contribute energy to the path?

c) When sampling VPLs, are the connections implicit or explicit?

Exam Questions

On Acceleration Structures:

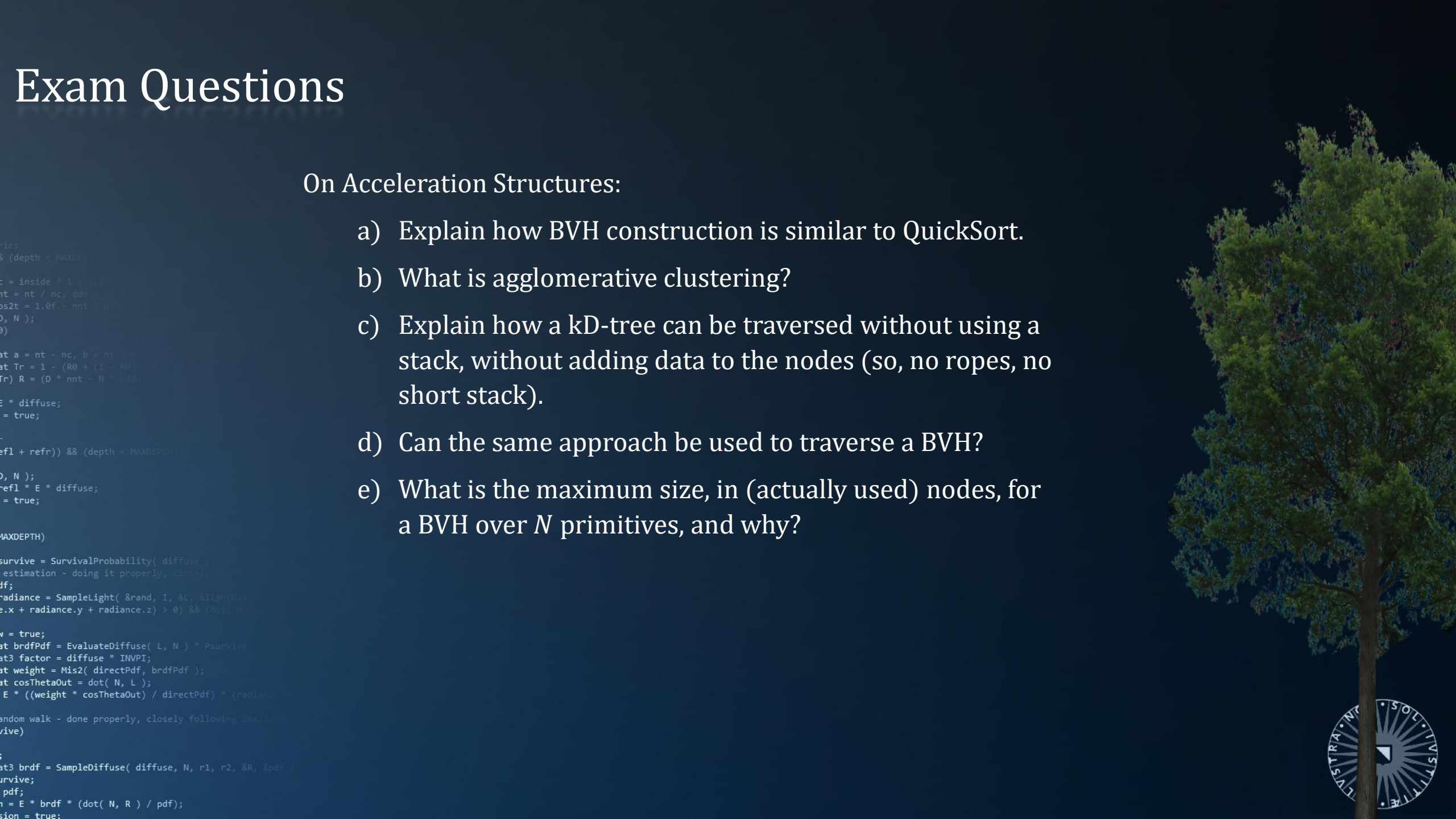
a) Explain how BVH construction is similar to QuickSort.

b) What is agglomerative clustering?

c) Explain how a kD-tree can be traversed without using a stack, without adding data to the nodes (so, no ropes, no short stack).

d) Can the same approach be used to traverse a BVH?

e) What is the maximum size, in (actually used) nodes, for a BVH over N primitives, and why?



Exam Questions

After reading the probability tutorial, answer these:

a) What is a definite integral?

b) What do we mean by an analytical solution?

c) How is the Riemann sum defined (mathematically)?

d) What is ‘univariate’?

e) What is ‘aliasing’?

f) Define, in your own words, ‘expected value’.

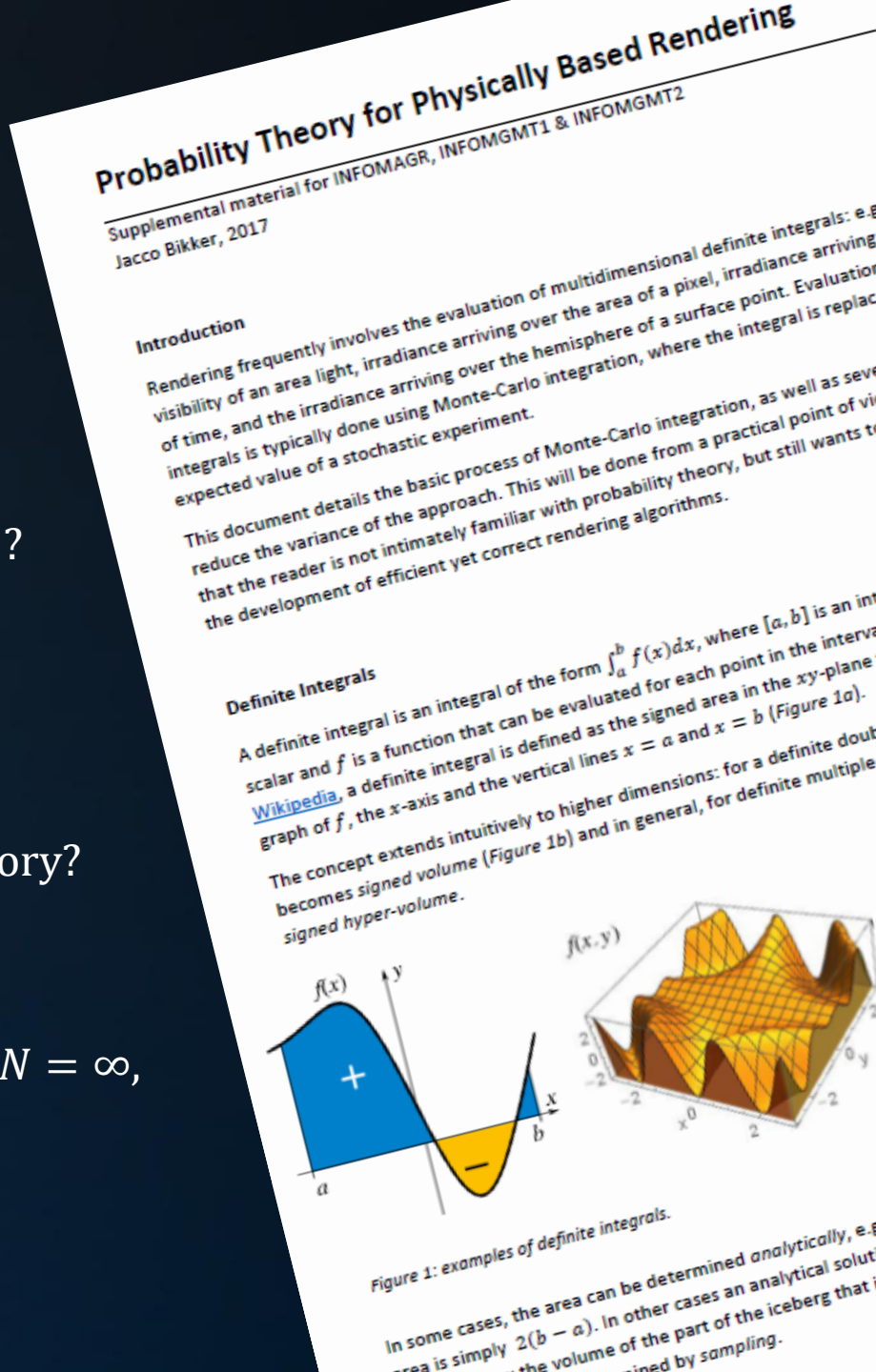
g) What is ‘deviation’ in the context of probability theory?

And, finally:

When using importance sampling, we assume that for $N = \infty$,

$$\frac{b-a}{N}\sum_{i=1}^{N}\frac{f(X)}{p(X)} = \frac{b-a}{N}\sum_{i=1}^{N} f(X) \text{ , if } \int_a^b p(x)dx = 1$$

Provide one example for which this is not true.



Probability Theory for Physically Based Rendering

Supplemental material for INFOMAGR, INFOMGMT1 & INFOMGMT2

Jacco Bikker, 2017

Introduction

Rendering frequently involves the evaluation of multidimensional definite integrals: e.g. visibility of an area light, irradiance arriving over the area of a pixel, irradiance arriving of time, and the irradiance arriving over the hemisphere of a surface point. Evaluation integrals is typically done using Monte-Carlo integration, where the integral is replac expected value of a stochastic experiment.

This document details the basic process of Monte-Carlo integration, as well as seve reduce the variance of the approach. This will be done from a practical point of vi that the reader is not intimately familiar with probability theory, but still wants t the development of efficient yet correct rendering algorithms.

Definite Integrals

A definite integral is an integral of the form $\int_a^b f(x)dx$, where $[a,b]$ is an int scalar and f is a function that can be evaluated for each point in the interva Wikipedia, a definite integral is defined as the signed area in the xy-plane graph of f, the x-axis and the vertical lines $x = a$ and $x = b$ (*Figure 1a*).

The concept extends intuitively to higher dimensions: for a definite doub becomes *signed volume* (*Figure 1b*) and in general, for definite multiple *signed hyper-volume*.

Figure 1: examples of definite integrals.

In some cases, the area can be determined *analytically*, e.g rea is simply $2(b-a)$. In other cases an analytical solut the volume of the part of the iceberg that i ined by *sampling*.

Exam Questions

An exam can be seen as a Monte-Carlo process. Explain why.

Exam Questions

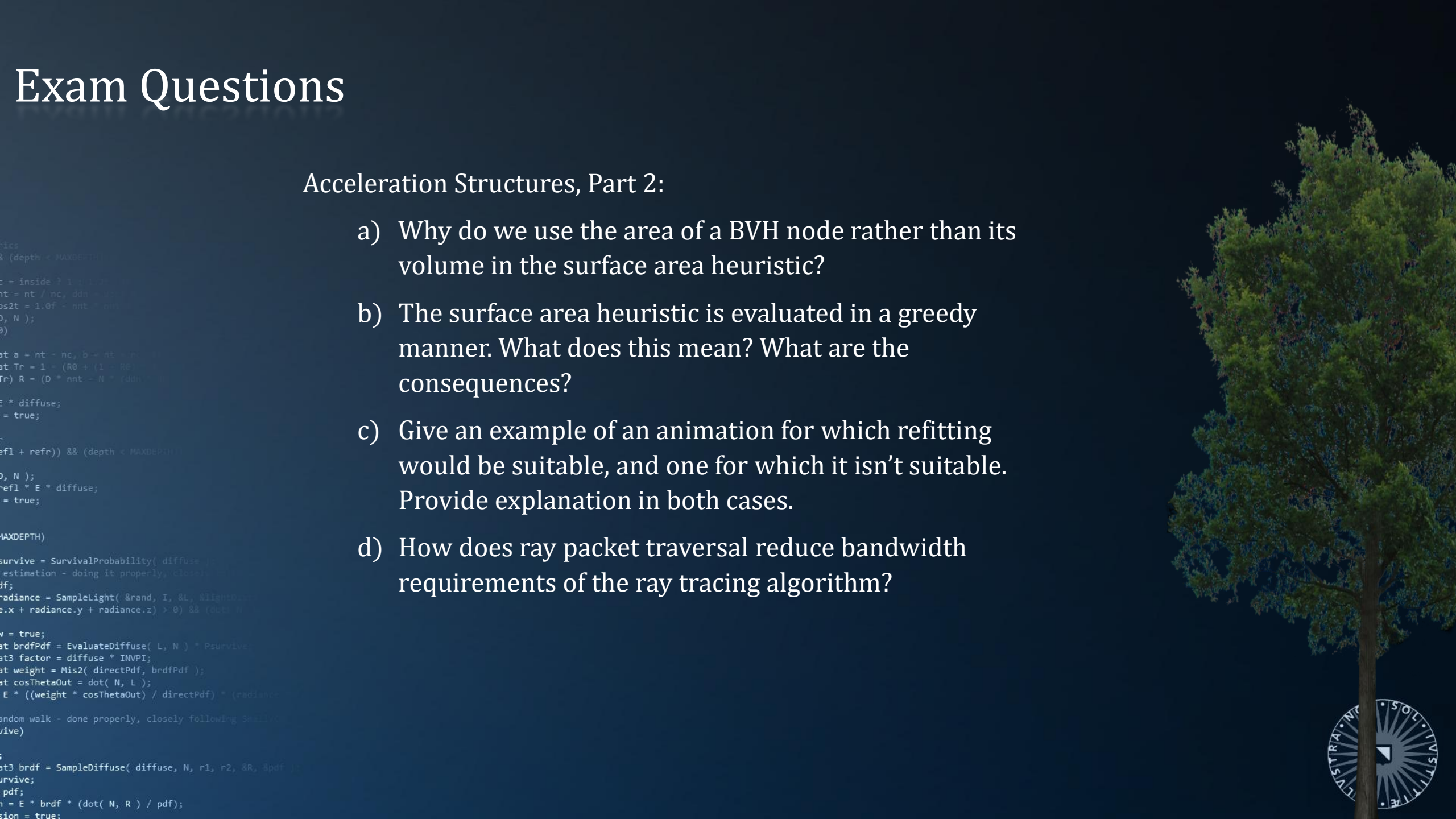
Acceleration Structures, Part 2:

a) Why do we use the area of a BVH node rather than its volume in the surface area heuristic?

b) The surface area heuristic is evaluated in a greedy manner. What does this mean? What are the consequences?

c) Give an example of an animation for which refitting would be suitable, and one for which it isn't suitable. Provide explanation in both cases.

d) How does ray packet traversal reduce bandwidth requirements of the ray tracing algorithm?



Exam Questions

…

INFOMAGR – Advanced Graphics

Jacco Bikker - November 2020 - February 2021

THE END *(for now)*