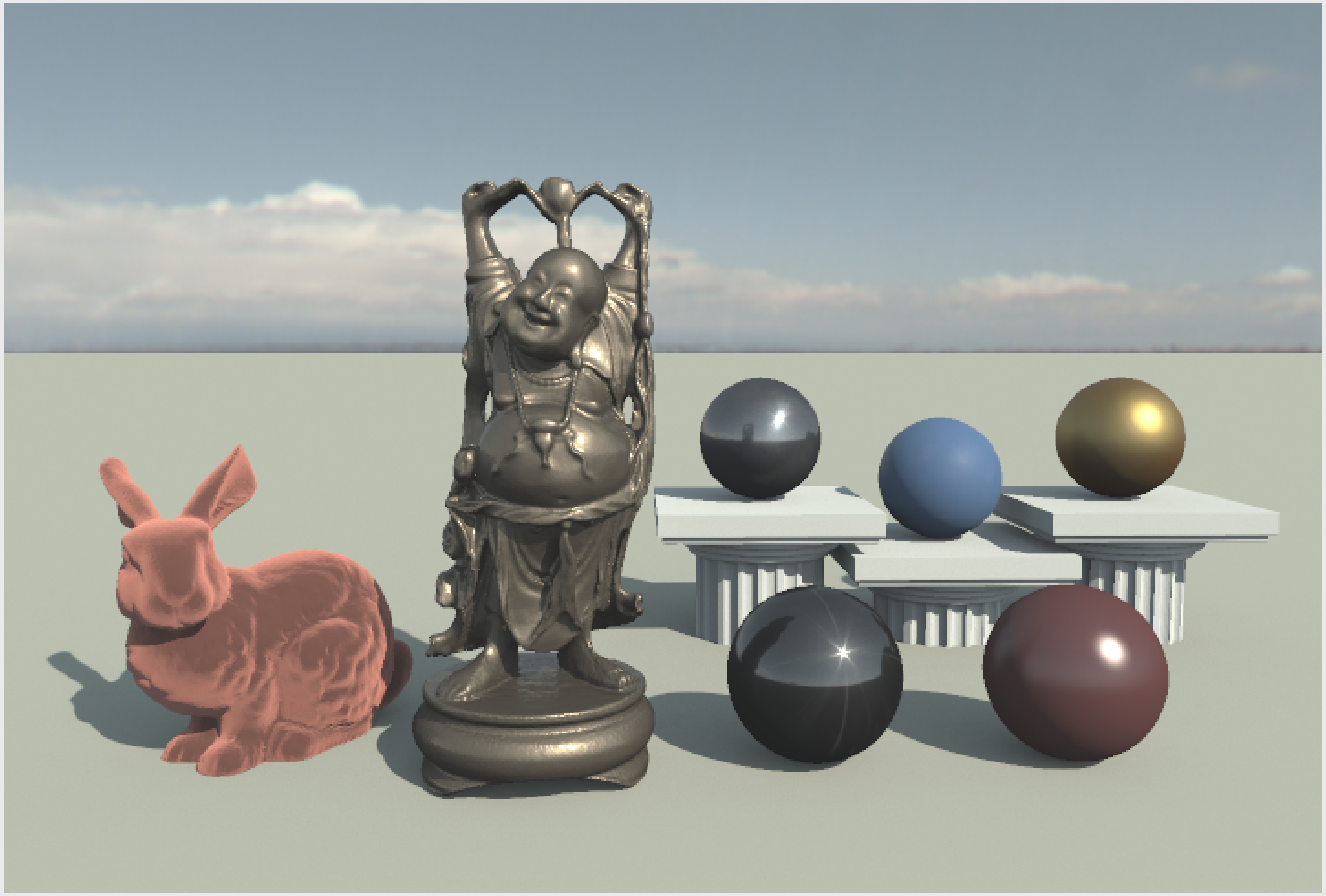
Efficient Product Sampling using Hierarchical Thresholding

Fabrice Rousselle
EPFL / University of Montreal

Petrik Clarberg
Lund University

Luc Leblanc, Victor Ostromoukhov, Pierre Poulin
University of Montreal

Objective



Plan

1. Introduction
2. Description of Hierarchical Thresholding
   - Basic algorithm (Ostromoukhov *et al*, Siggraph 2004)
   - Our extensions
3. Application to direct illumination
4. Results
5. Conclusion and future work

Introduction

- Monte Carlo ray tracing
  - Widely used in photo-realistic rendering
  - Many samples are needed for noise-free results

$$\langle F \rangle = \frac{1}{n} \sum_{i=1}^{n} f(x_i), \text{ with } n \text{ the number of samples}$$

- Importance sampling
  - Offers significant noise reduction by concentrating the sampling to important regions
  - Ideally, the importance function $p(x)$ should be proportional to the function sampled

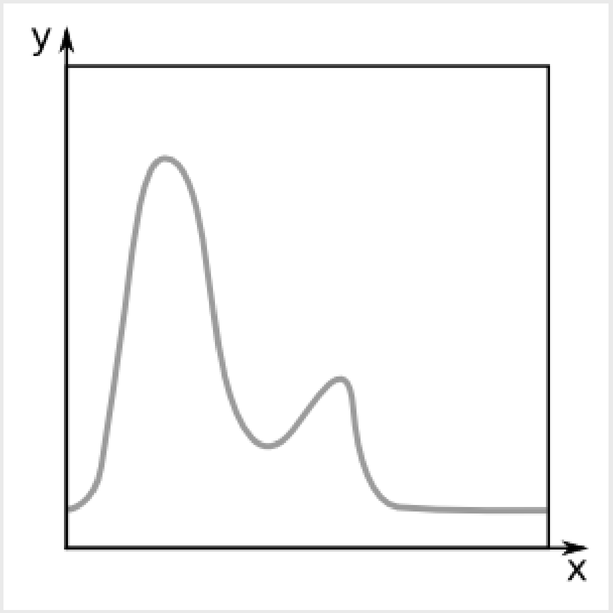
$$\langle F \rangle = \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)}, \text{ with } \int p(x)dx = 1$$

Introduction

- Proposition: Hierarchical Thresholding (HT)
  - A simple sampling scheme
  - Applicable to products of multiple functions
  - Can be easily integrated in a Monte Carlo ray tracer

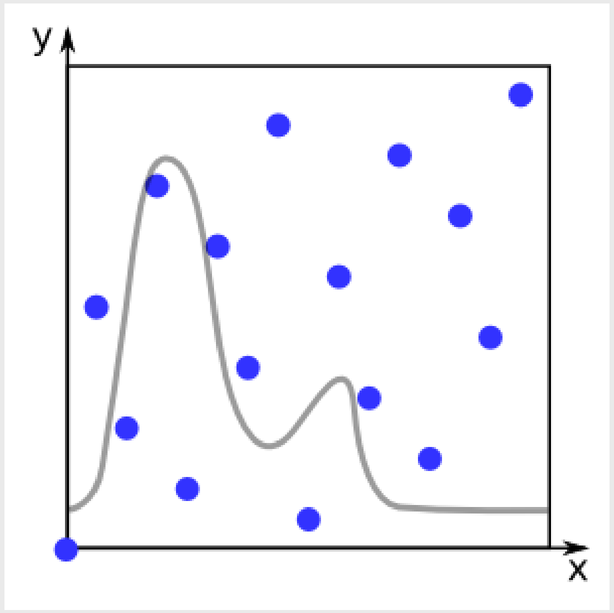
HT: Basic Algorithm

- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



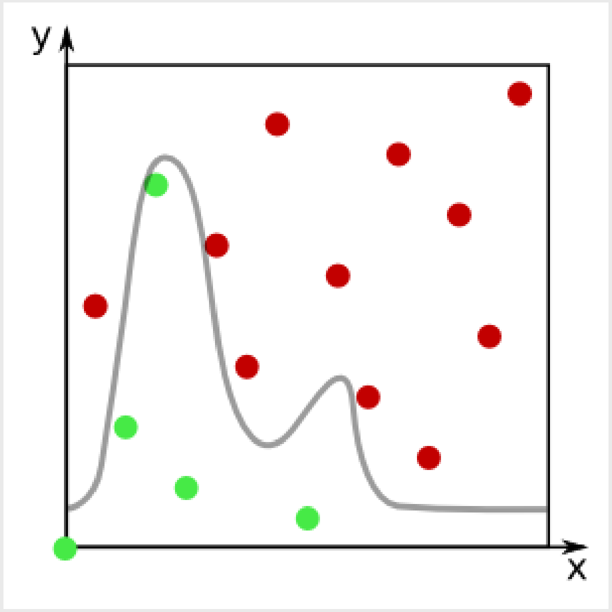
HT: Basic Algorithm

- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



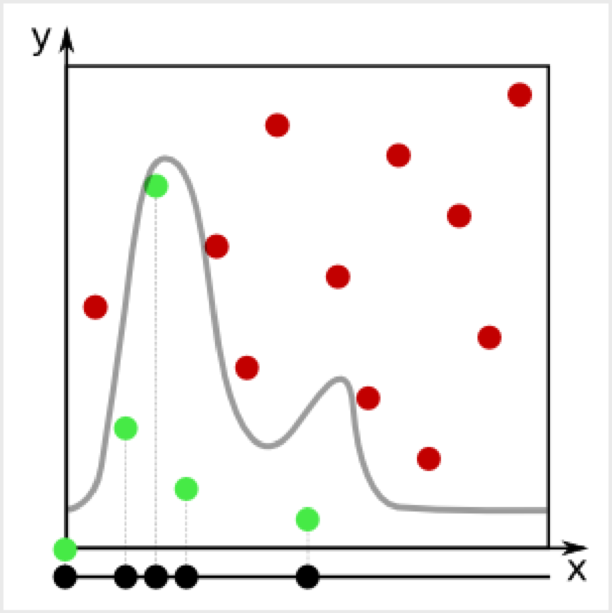
HT: Basic Algorithm

- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



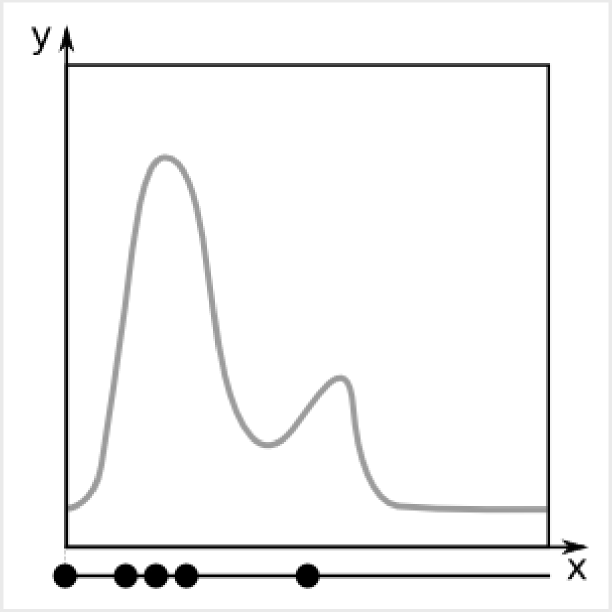
HT: Basic Algorithm

- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



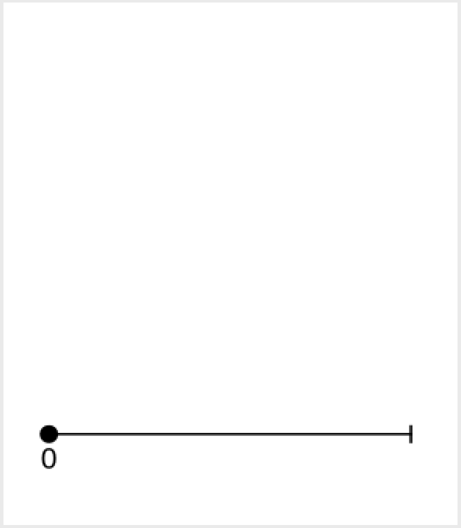
HT: Basic Algorithm

- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



HT: Basic Algorithm

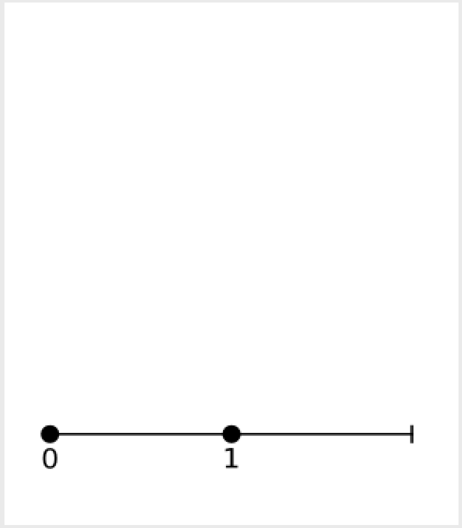
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

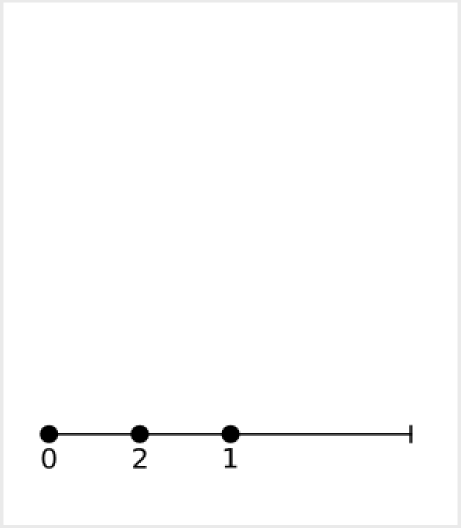
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

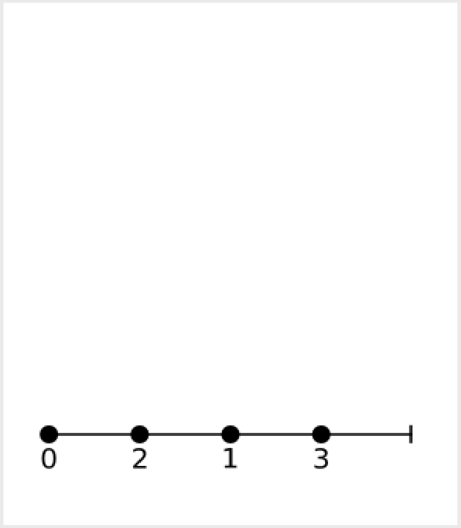
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

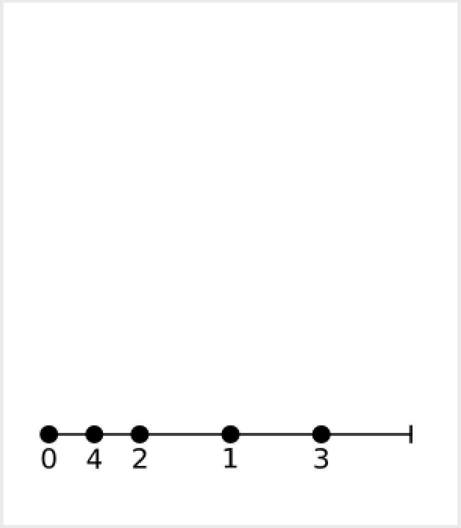
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

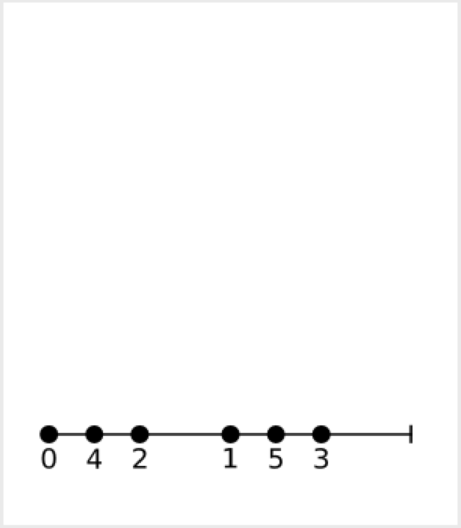
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

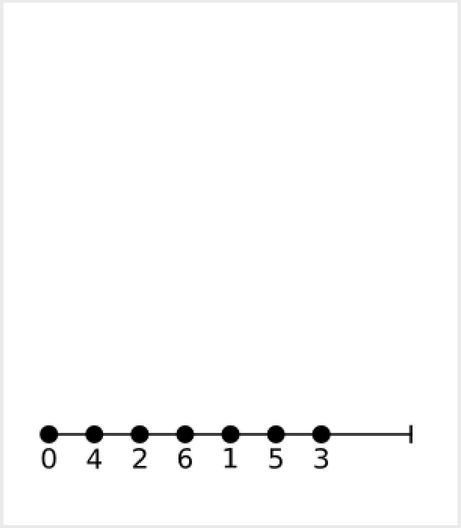
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

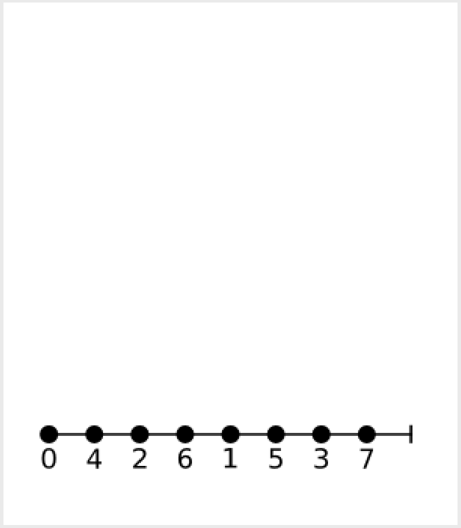
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

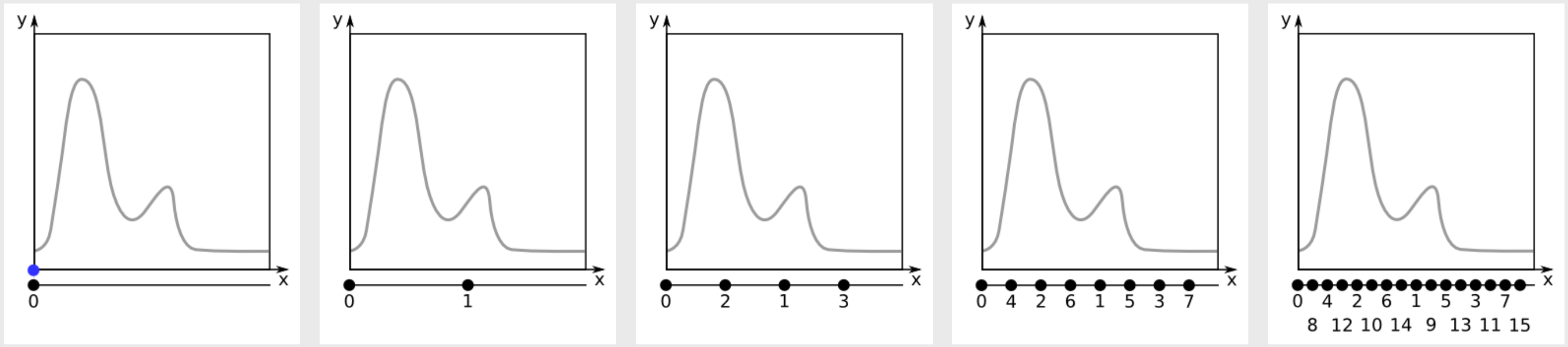
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

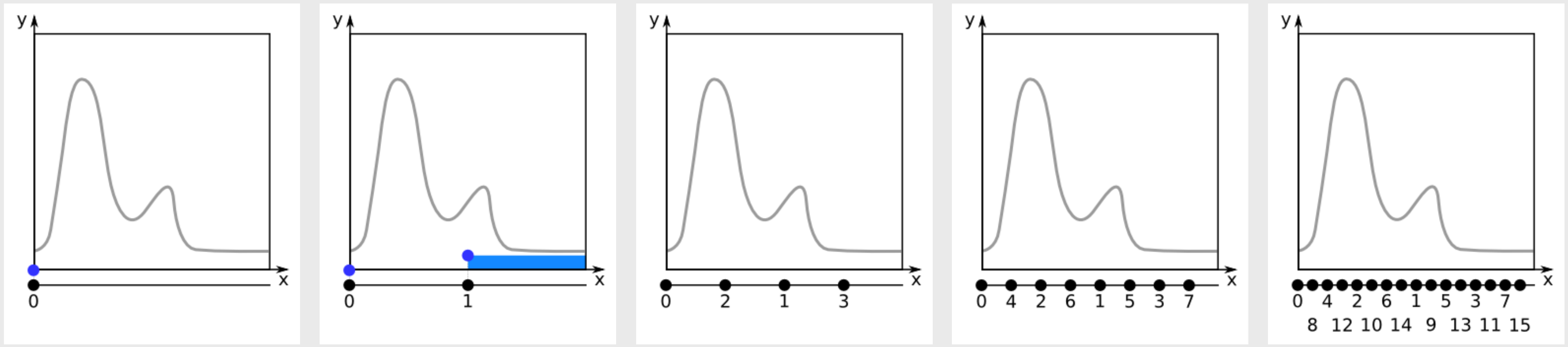
- The number of candidate samples generated is: $N = b^L$
  - with *b* the base of the VDC sequence (2 in this example)
  - with *L* the maximum level of subdivision (4 in this example)
- A sample value is *i/N*, with *i* its index in the sequence



Van der Corput sequence in base 2

HT: Basic Algorithm

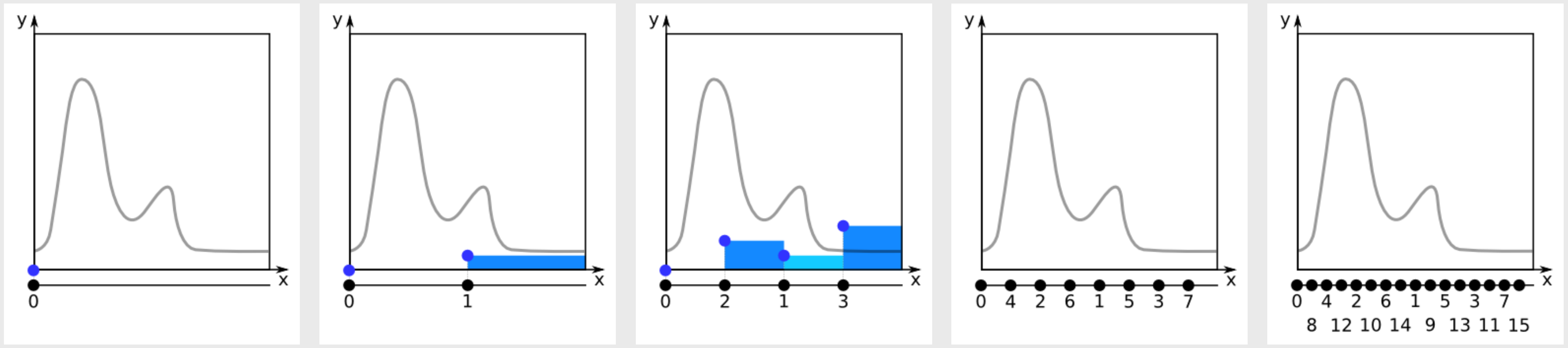
- The number of candidate samples generated is: $N = b^L$
  - with *b* the base of the VDC sequence (2 in this example)
  - with *L* the maximum level of subdivision (4 in this example)
- A sample value is *i/N*, with *i* its index in the sequence



Van der Corput sequence in base 2

HT: Basic Algorithm

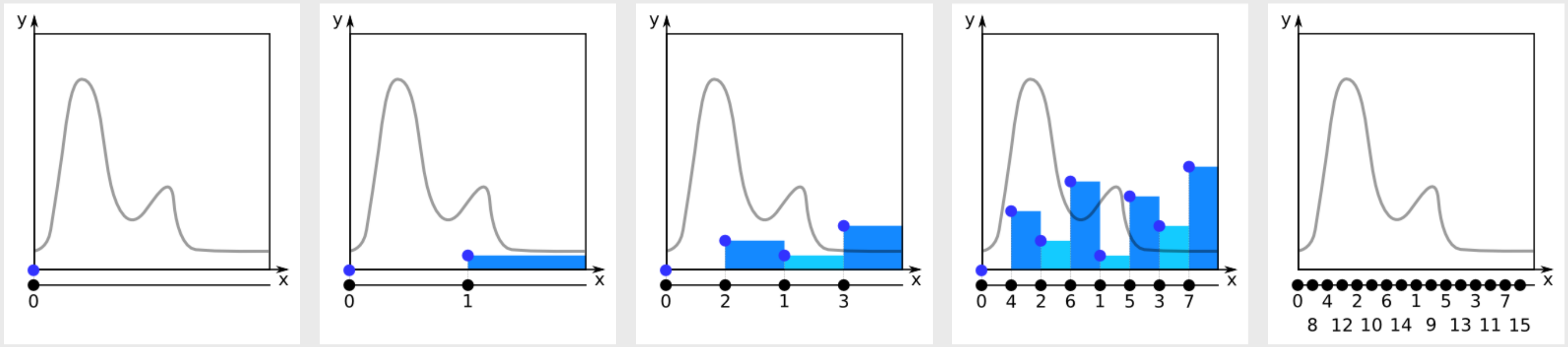
- The number of candidate samples generated is: $N = b^L$
  - with b the base of the VDC sequence (2 in this example)
  - with L the maximum level of subdivision (4 in this example)
- A sample value is i/N, with i its index in the sequence



Van der Corput sequence in base 2

HT: Basic Algorithm

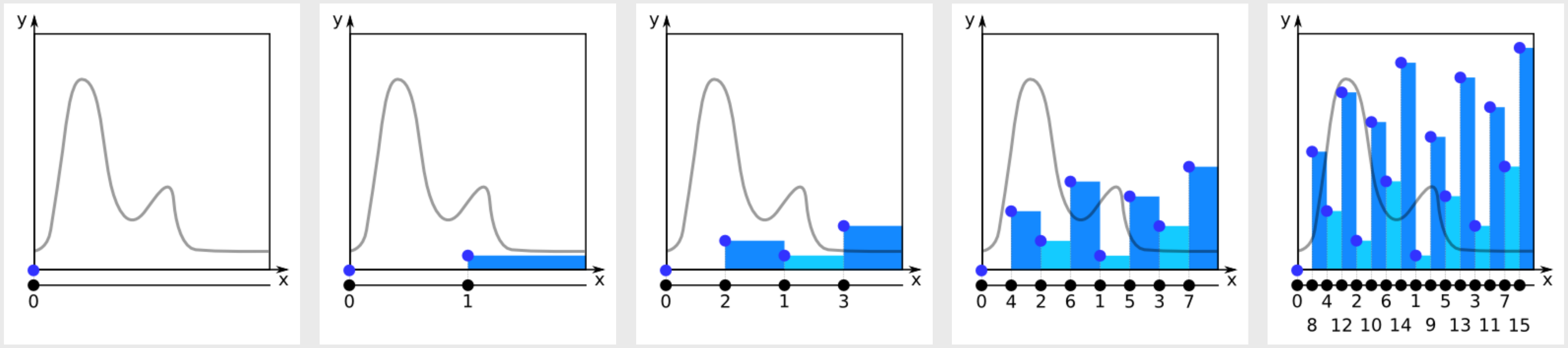
- The number of candidate samples generated is: $N = b^L$
  - with b the base of the VDC sequence (2 in this example)
  - with L the maximum level of subdivision (4 in this example)
- A sample value is i/N, with i its index in the sequence



Van der Corput sequence in base 2

HT: Basic Algorithm

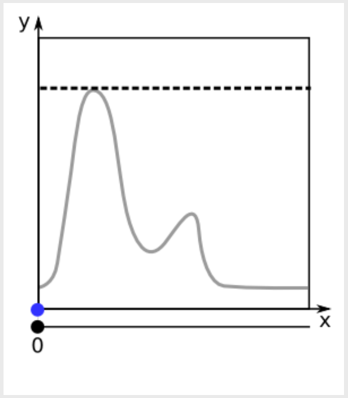
- The number of candidate samples generated is: $N = b^L$
  - with b the base of the VDC sequence (2 in this example)
  - with L the maximum level of subdivision (4 in this example)
- A sample value is i/N, with i its index in the sequence



Van der Corput sequence in base 2

HT: Basic Algorithm

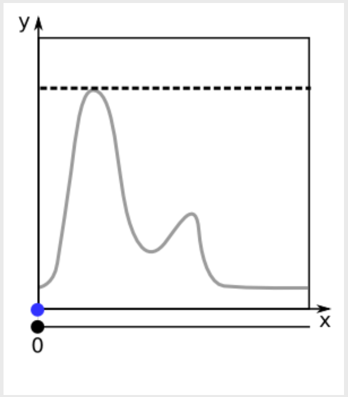
- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate

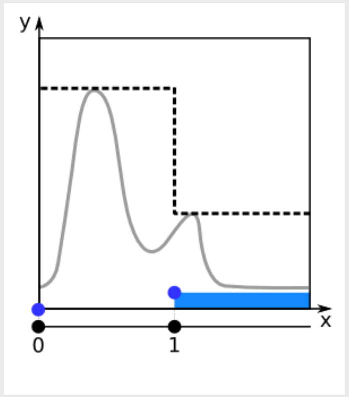


Van der Corput sequence in base 2

HT: Basic Algorithm

- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate

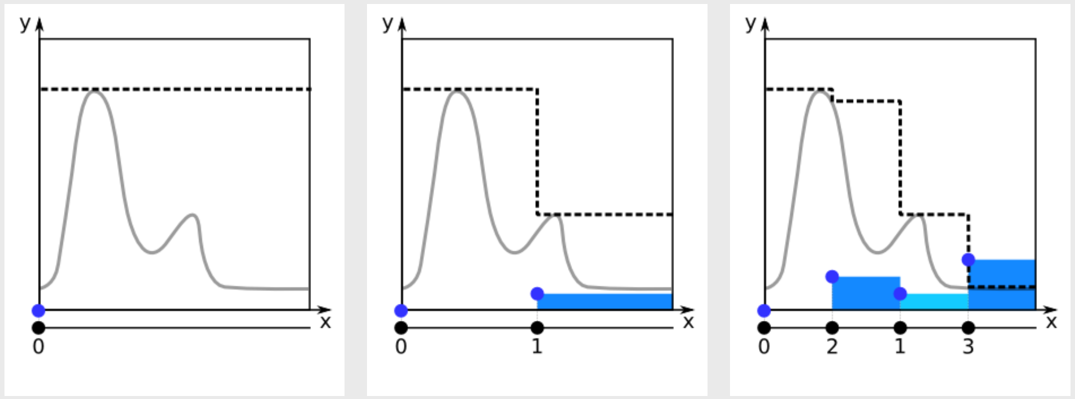




Van der Corput sequence in base 2

HT: Basic Algorithm

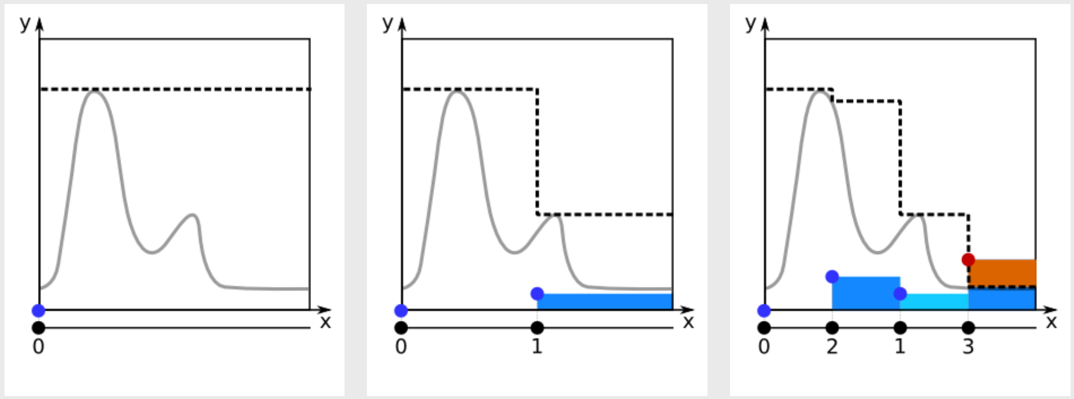
- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate



Van der Corput sequence in base 2

HT: Basic Algorithm

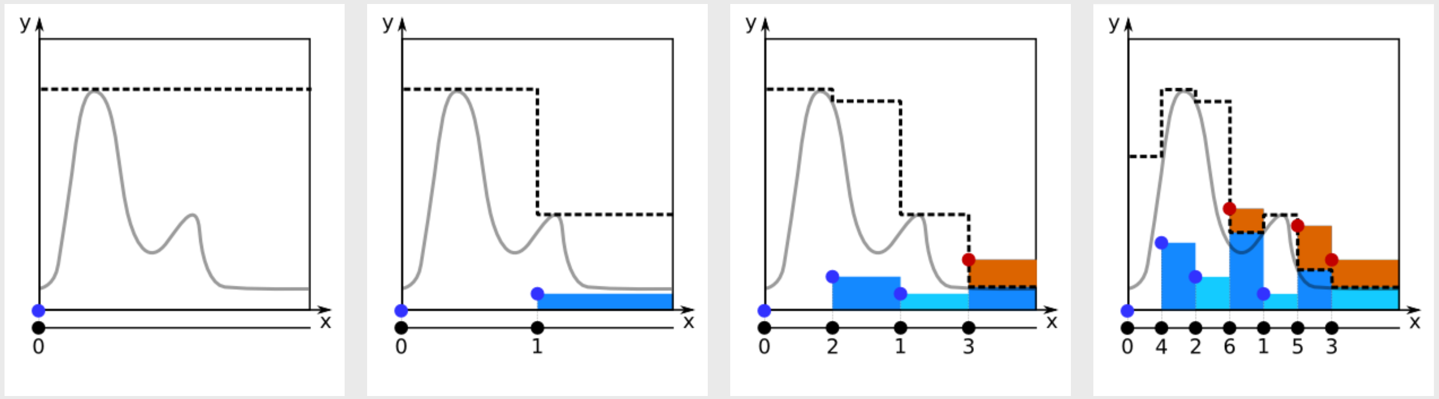
- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate



Van der Corput sequence in base 2

HT: Basic Algorithm

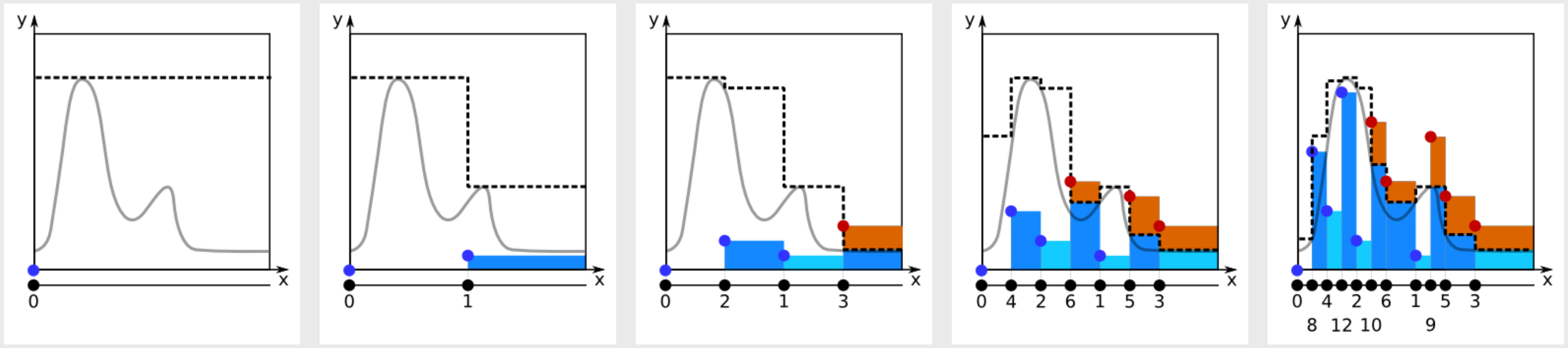
- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate



Van der Corput sequence in base 2

HT: Basic Algorithm

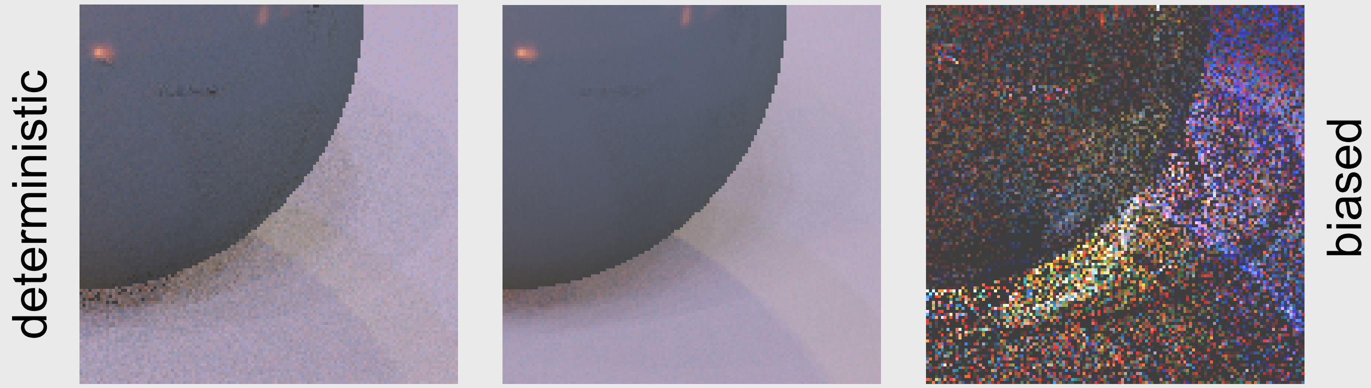
- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate



Van der Corput sequence in base 2

HT: Avoiding Bias

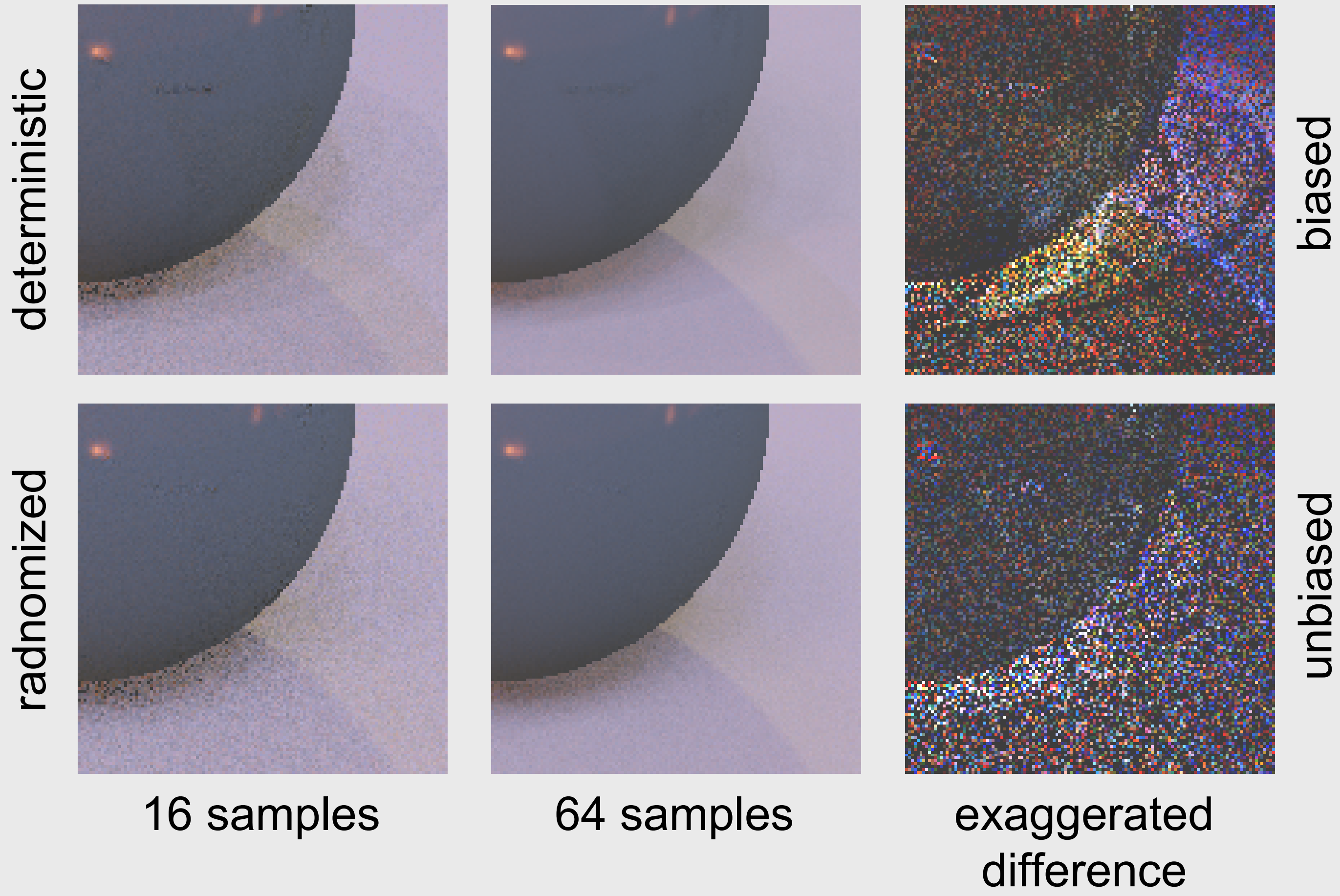
- Low discrepancy sequences are deterministic



16 samples 64 samples exaggerated difference

HT: Avoiding Bias

- Low discrepancy sequences are deterministic
  - Randomize key steps of the processus (see paper for details)



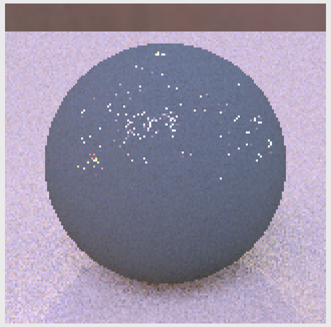
HT: Use of Max/Avg Tree

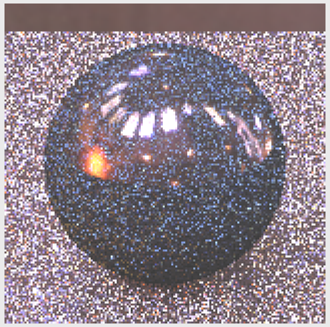
- Local maximum value
  - Needed for pruning branches
- Local average value
  - Predict the number of selected samples
  - Speed up the rejection test (avoid BRDF evaluation)
  - Perform further performance optimizations
  - Yields a piecewise-constant importance function: $h(x)$

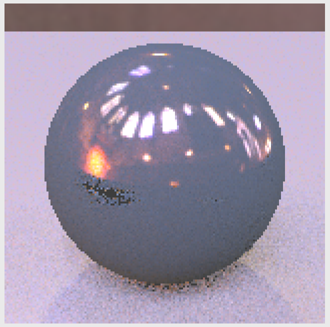
- The local maximum and average values are pre-computed for all nodes of the hierarchy

HT: Product of Functions

- Direct illumination: sampling a product of functions
  - Environment only (works for diffuse surfaces)
  - BRDF only (works for specular surfaces)
  - Environment x BRDF (works well, but problematic occlusions)


Environment

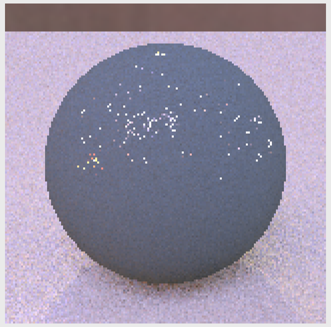

BRDF

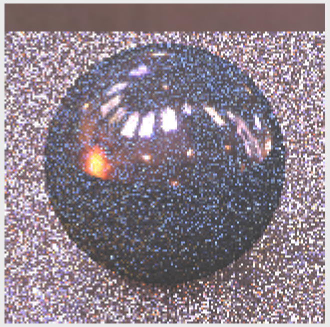

Env x BRDF

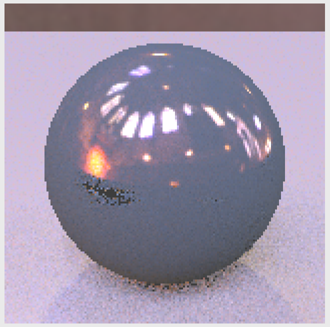
Rendered using 16 samples

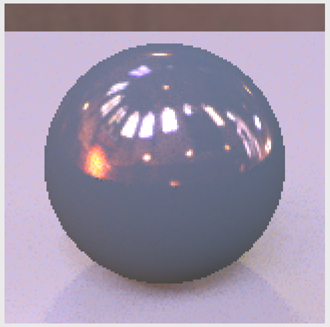
HT: Product of Functions

- Direct illumination: sampling a product of functions
  - Environment only (works for diffuse surfaces)
  - BRDF only (works for specular surfaces)
  - Environment x BRDF (works well, but problematic occlusions)
  - Environment x BRDF x visibility


Environment


BRDF


Env x BRDF


Env x BRDF x Vis

Rendered using 16 samples

HT: Product of Functions

- Our algorithm extends to the product of n functions:

$$f(x) = \prod_{i=1}^{n} f_i(x) \tag{1}$$

- The local maximum is conservatively approximated:

$$\max f(x) \leq \prod_{i=1}^{n} \left(\max f_i(x)\right), \qquad x \in [a, b] \tag{2}$$

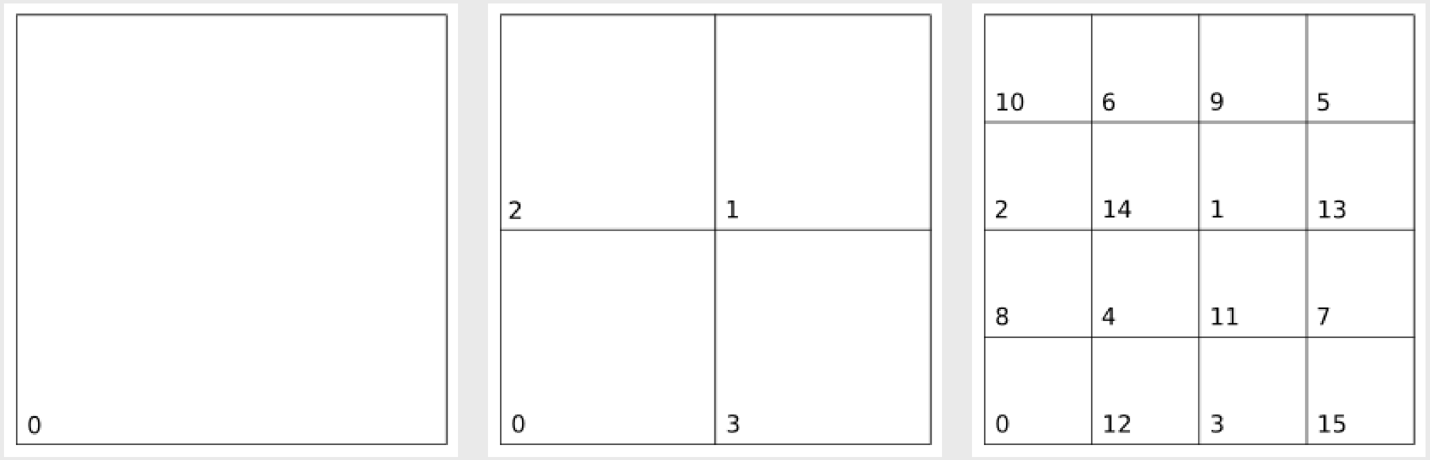
- The local average is similarly approximated:

$$\text{avg } f(x) \approx \prod_{i=1}^{n} \left(\text{avg } f_i(x)\right), \qquad x \in [a, b] \tag{3}$$

  – The error of this approximation only affects the variance of the estimation, the estimator still convergences to the right result

HT: Sampling in 3D

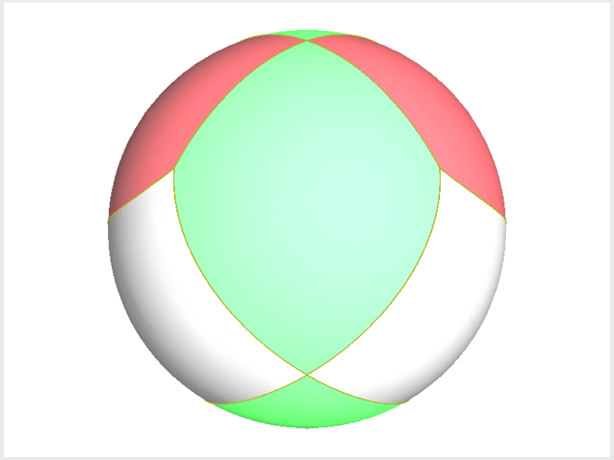
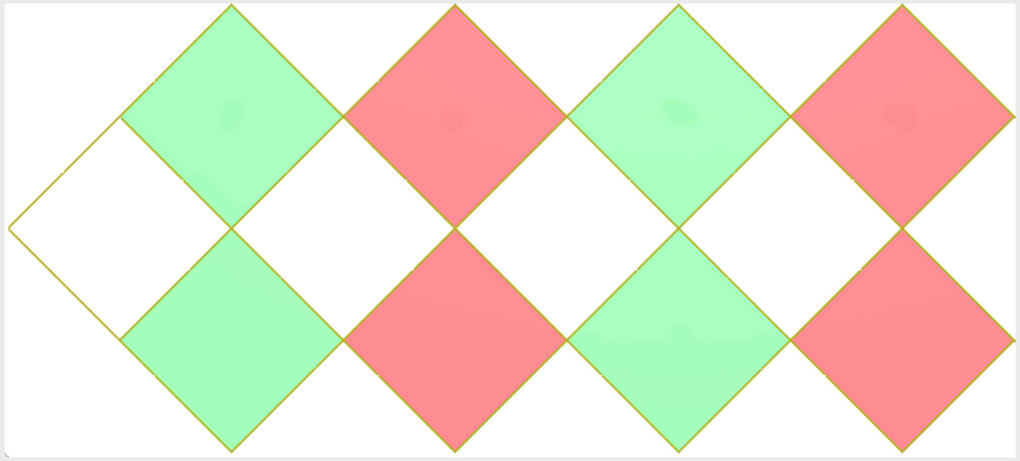
- Samples are positioned on a 2D plane



Sample indexing according to the VDC sequence in base 4

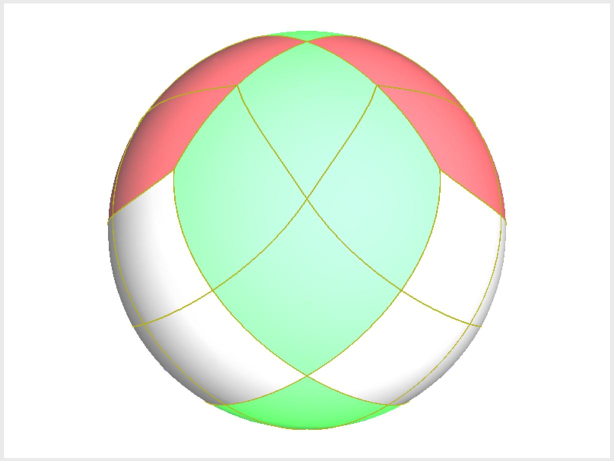
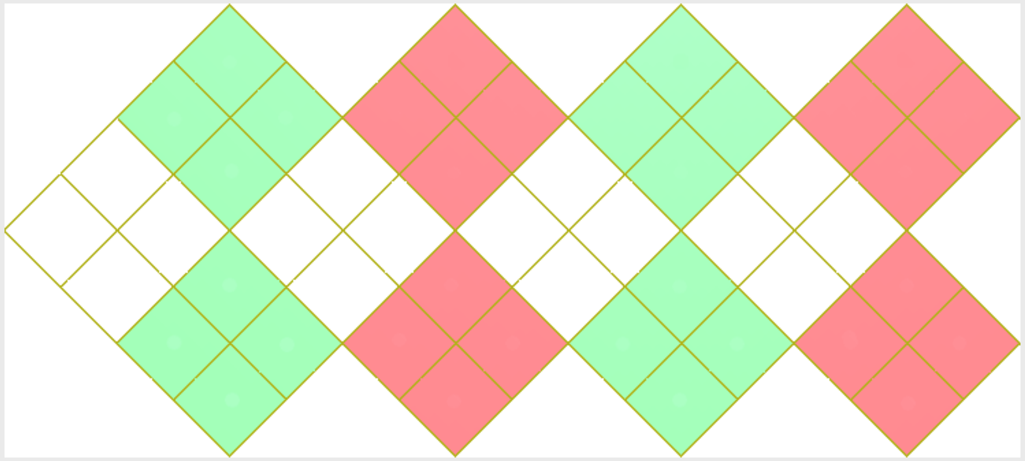
Application: Direct Illumination

- The mapping used: HEALPix
  - hierarchical representation (needed for our algorithm)
  - area preservation
  - low distorsion (allows dynamic rotations)
- Each face is sampled individually

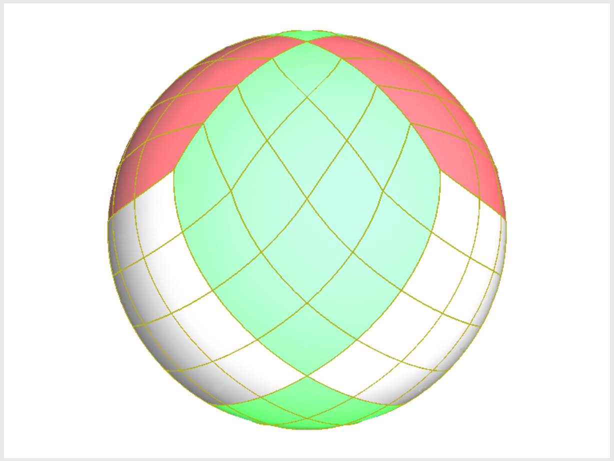
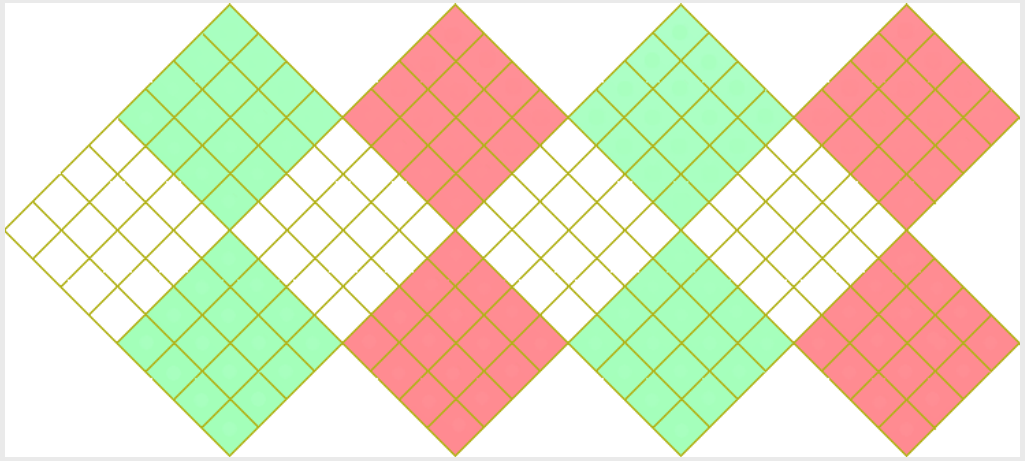
Application: Direct Illumination

- The mapping used: HEALPix
  - hierarchical representation (needed for our algorithm)
  - area preservation
  - low distorsion (allows dynamic rotations)
- Each face is sampled individually

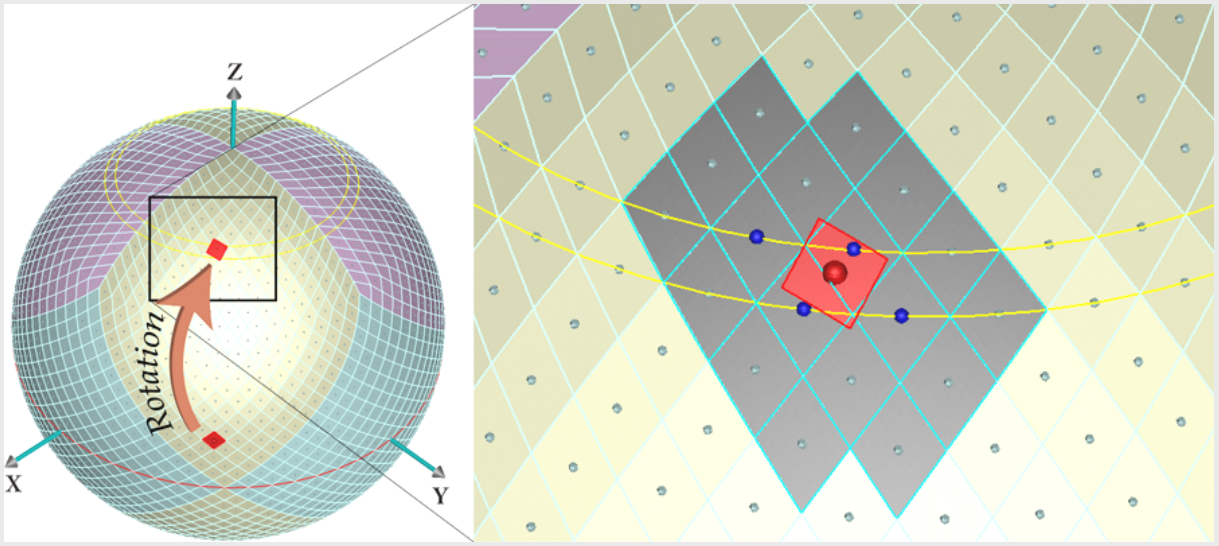
Application: Direct Illumination

- The mapping used: HEALPix
  - hierarchical representation (needed for our algorithm)
  - area preservation
  - low distorsion (allows dynamic rotations)
- Each face is sampled individually

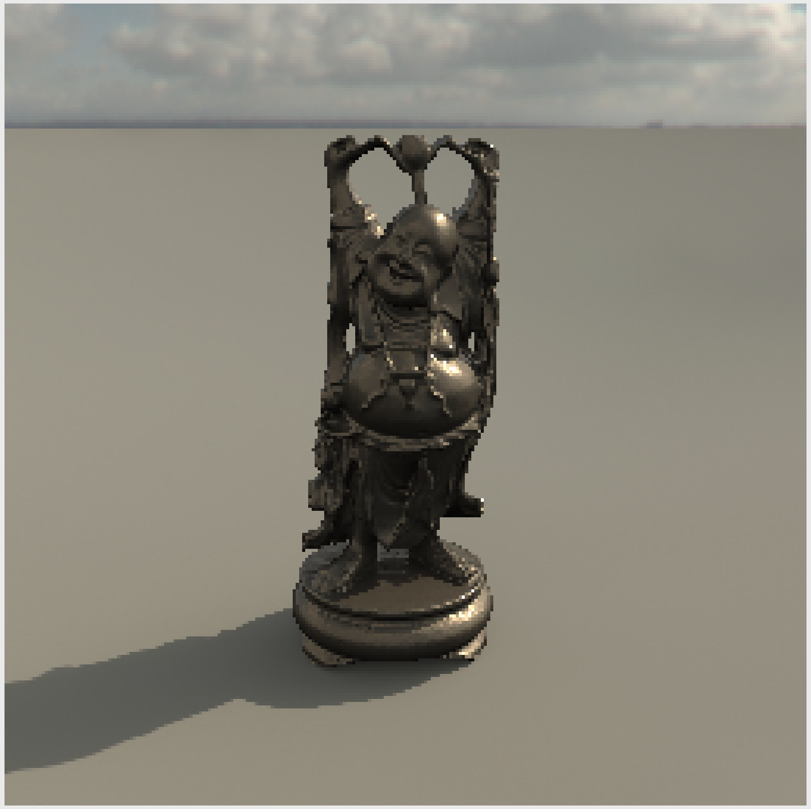
HEALPix: Rotations

- Environment Map in world space
- BRDF in local space
  - Fetching the BRDF value implies a rotation
  - The max area is enlarged to remain conservative
  - The avg is interpolated from the four neighbors



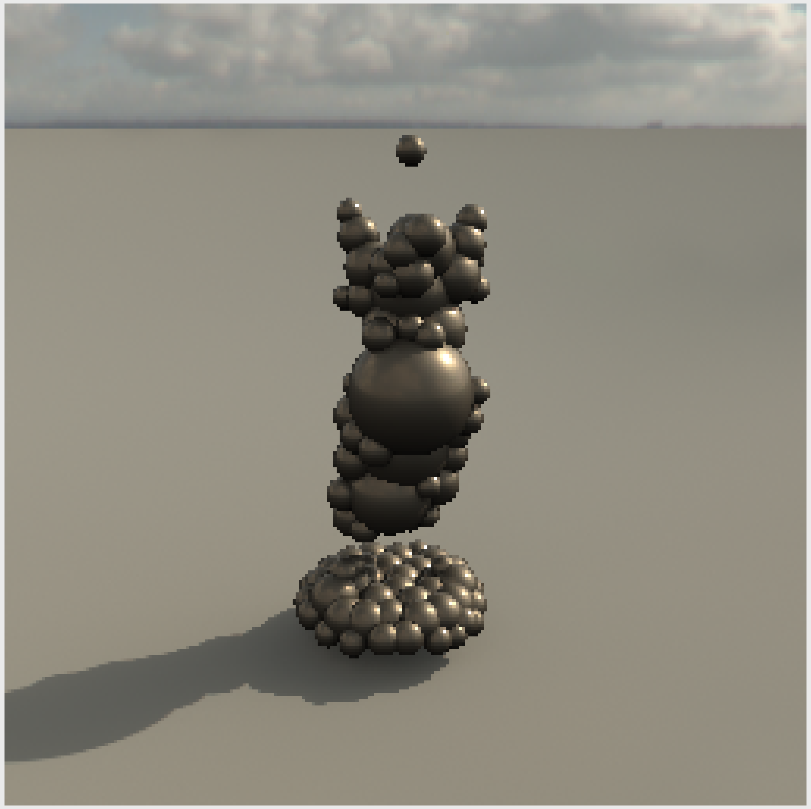
Application: Visibility Computation

- Computed from a simplified model using inner spheres



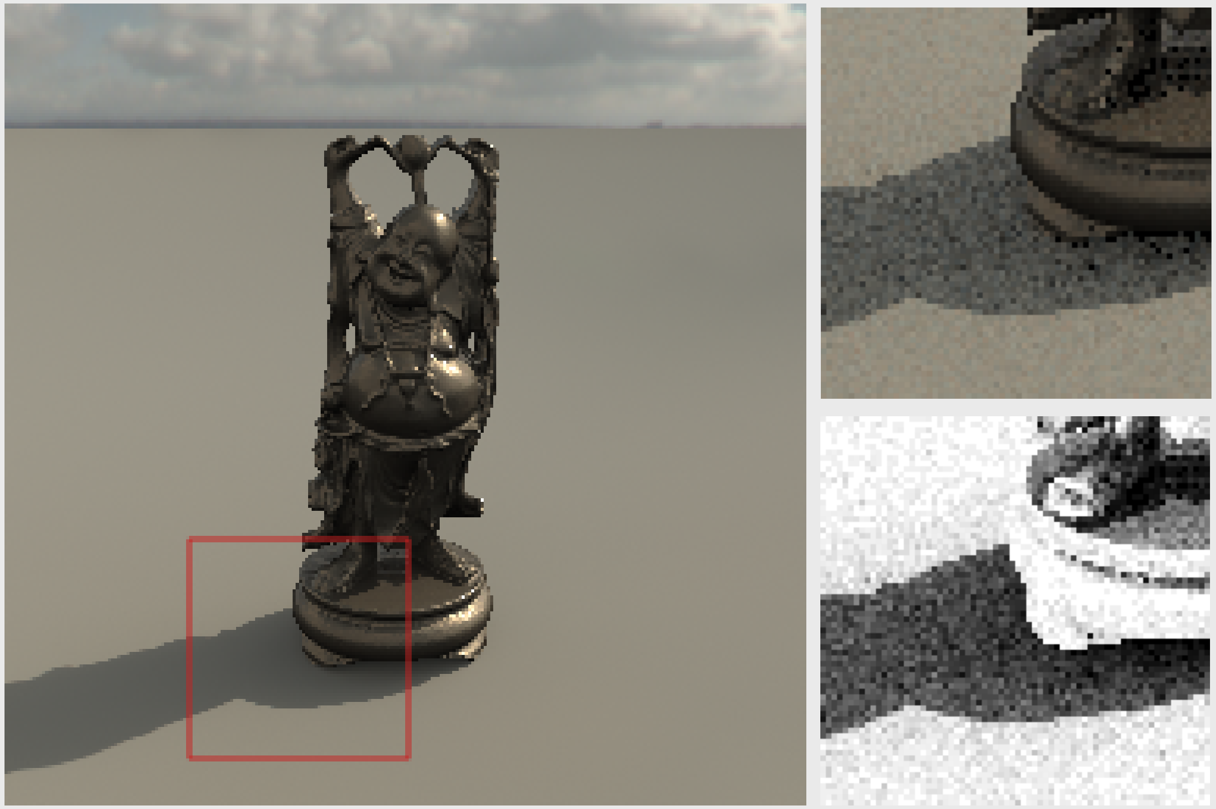
Application: Visibility Computation

- Computed from a simplified model using inner spheres



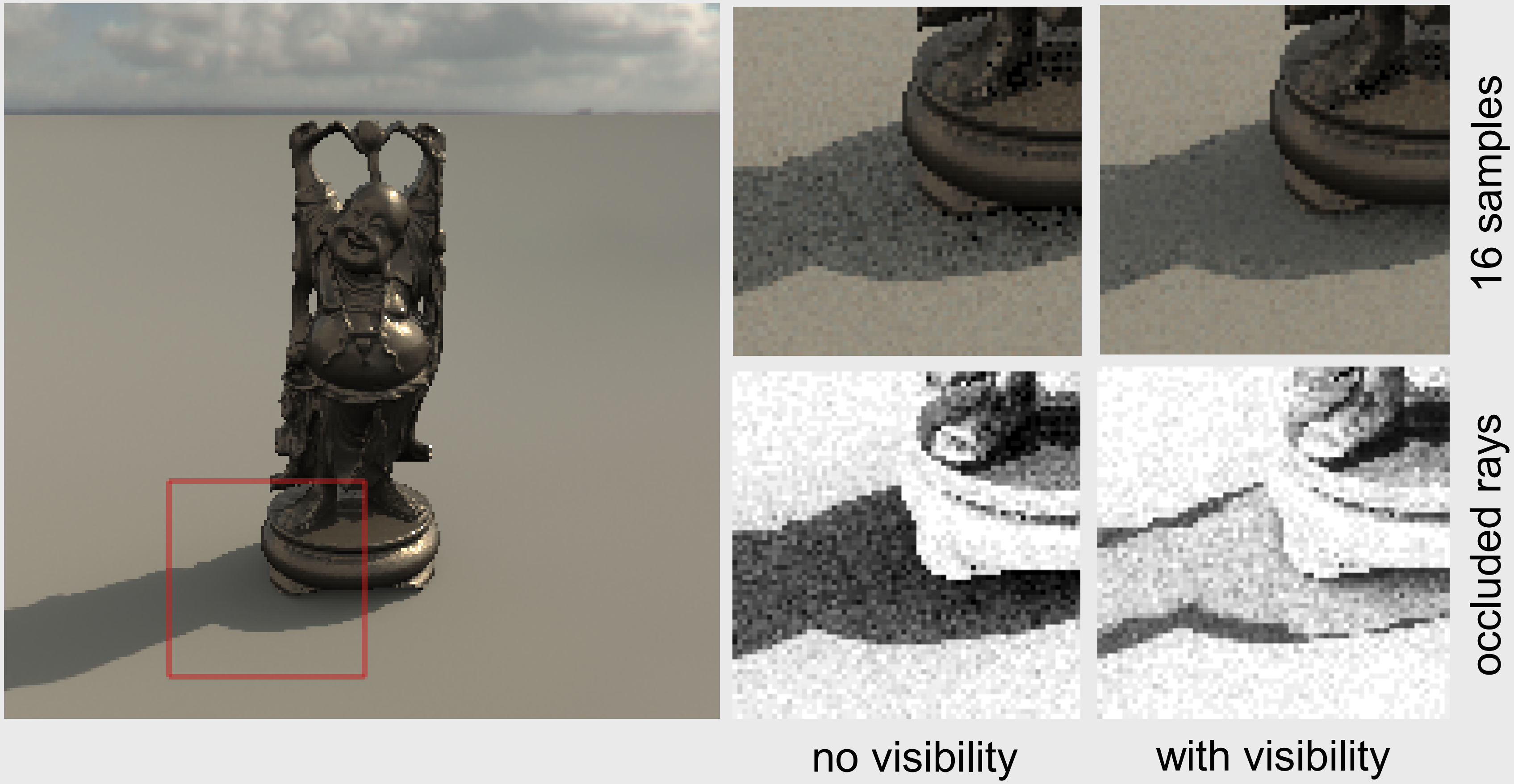
Application: Visibility Computation

- Computed from a simplified model using inner spheres



Application: Visibility Computation

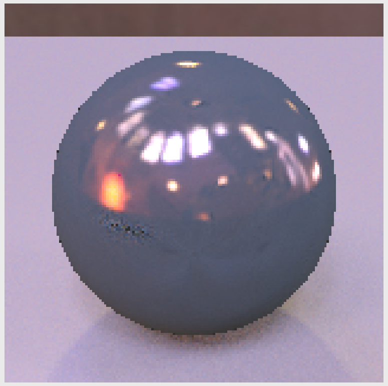
- Computed from a simplified model using inner spheres

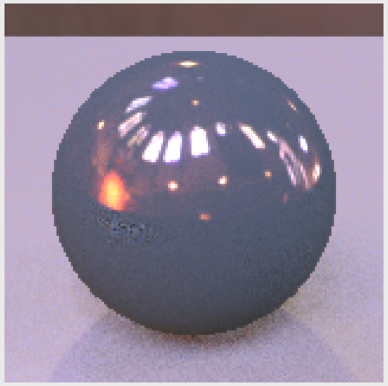


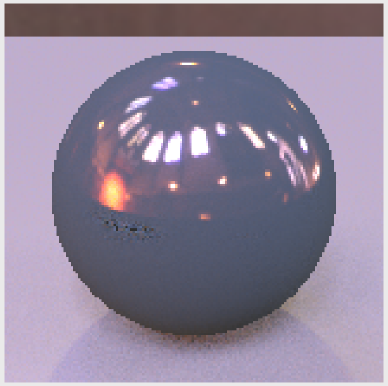
Results

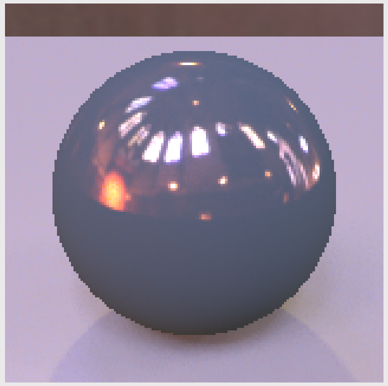
- Comparison with other product sampling methods

64 samples

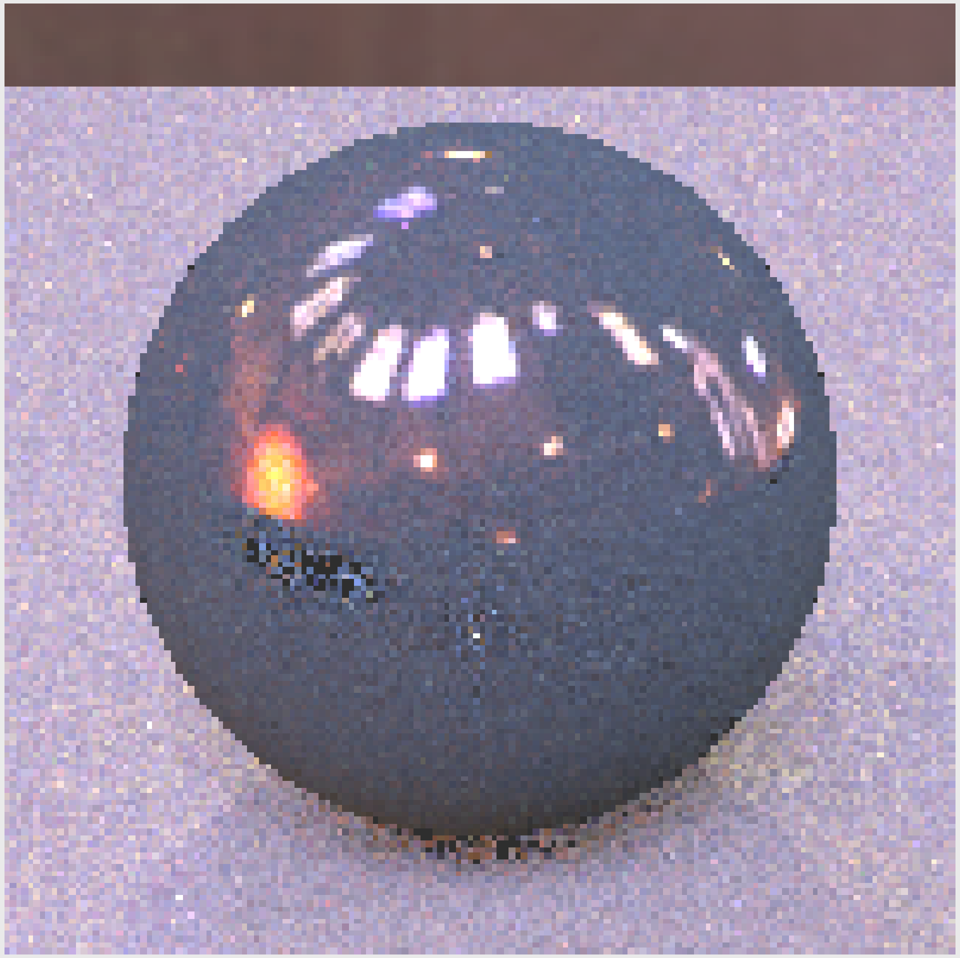

WIS biased


TSIS unbiased


HT no visibility

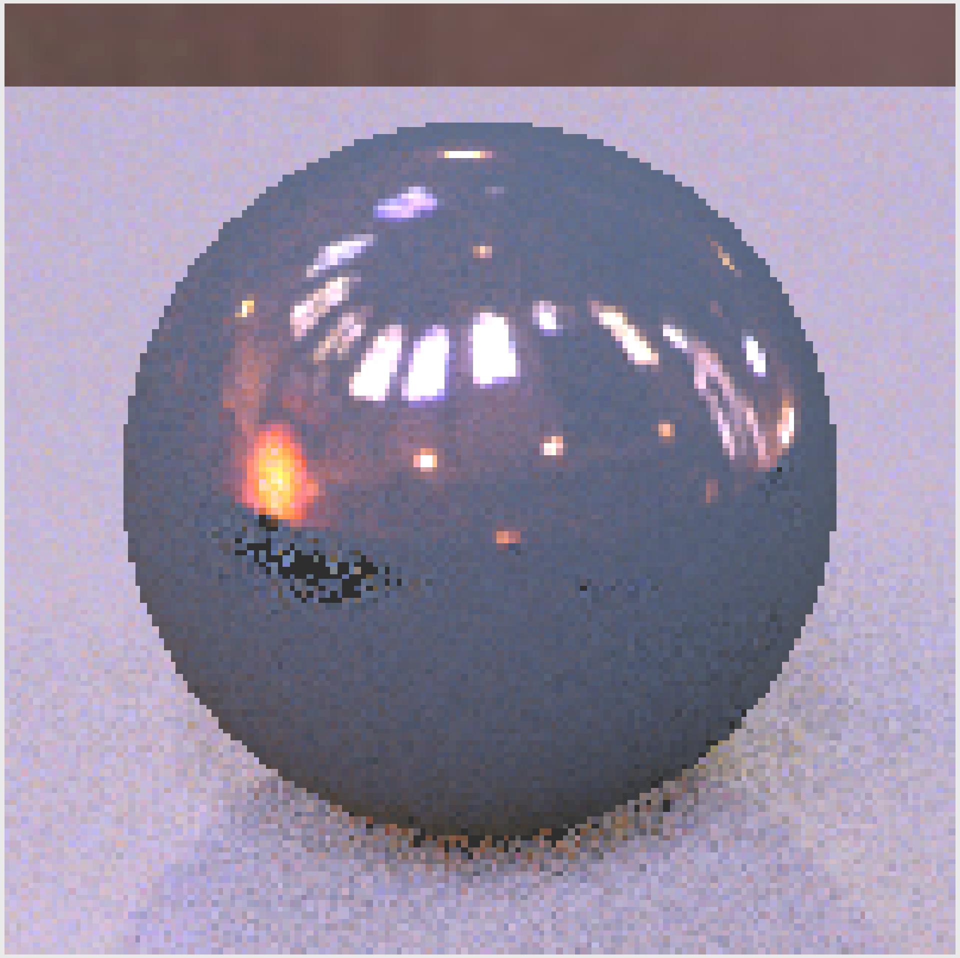

HT with visibility

Wavelet Importance Sampling

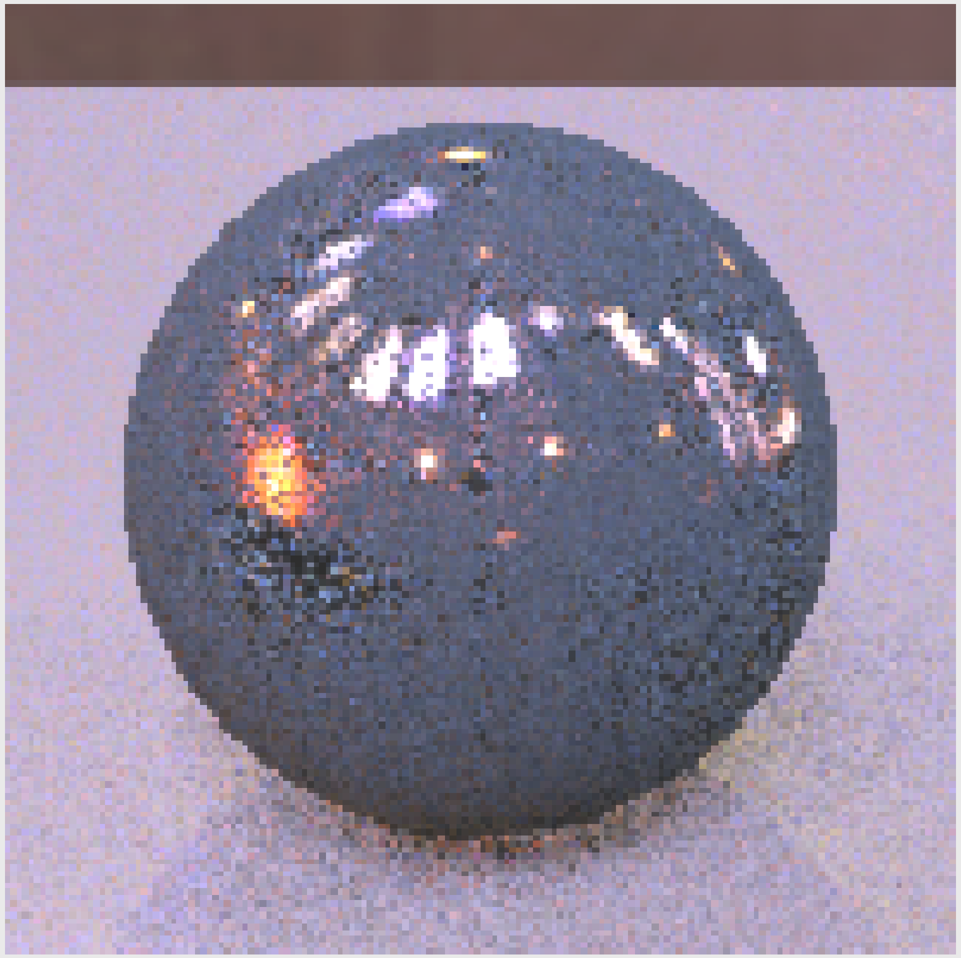


Clarberg *et al*. Siggraph 2005 - 16 samples

HT: no visibility

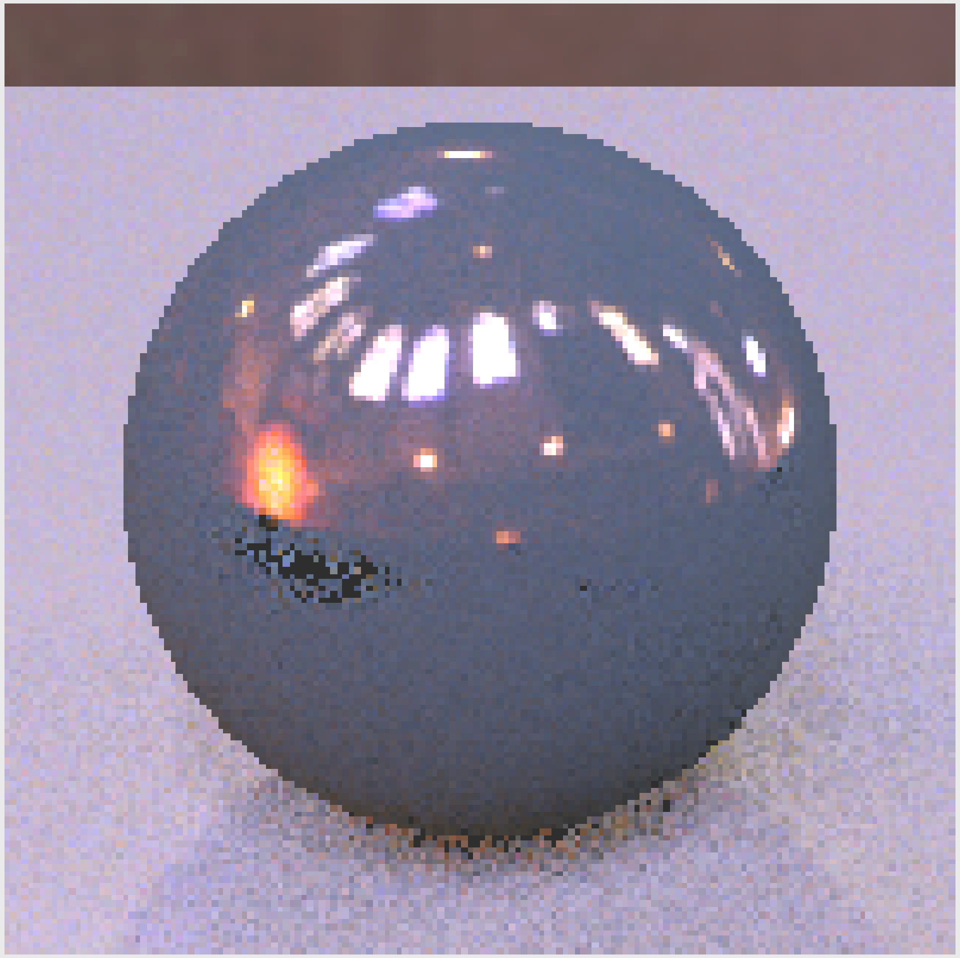

16 samples

Two Stage Importance Sampling

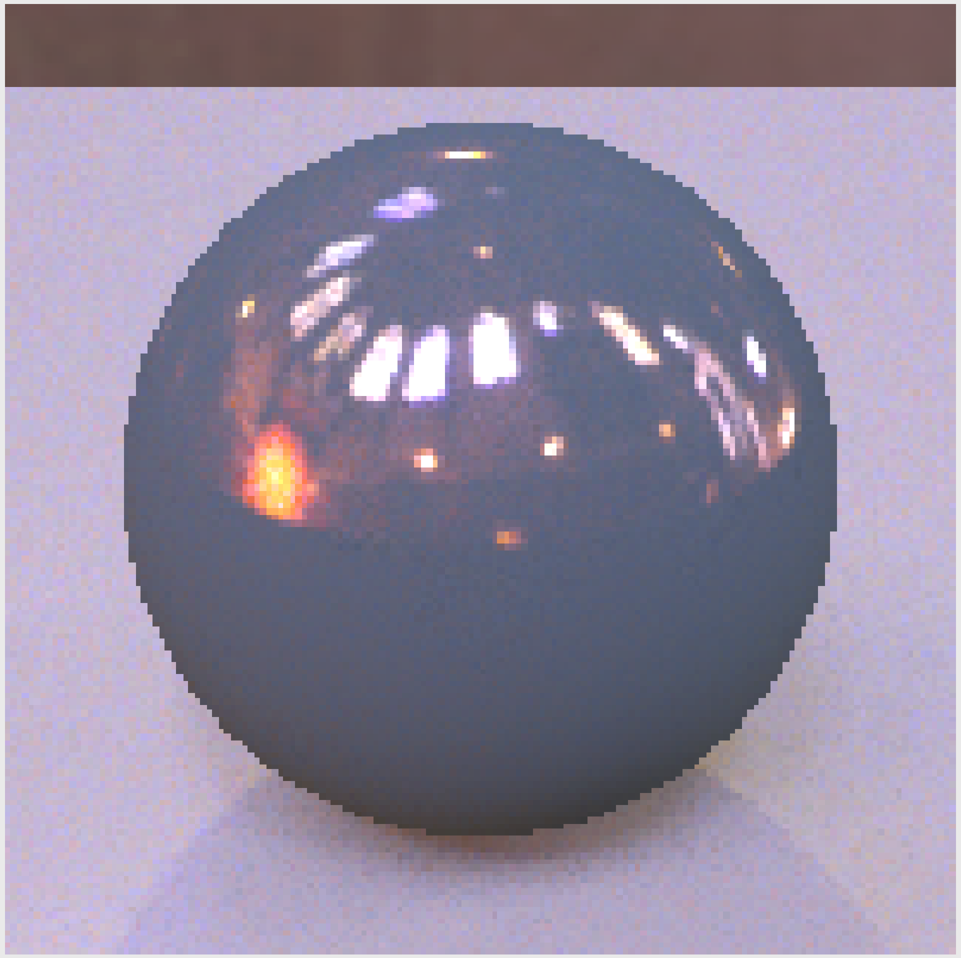


Cline *et al*. EGSR 2006 - 16 samples

HT: no visibility


16 samples

HT: with visibility

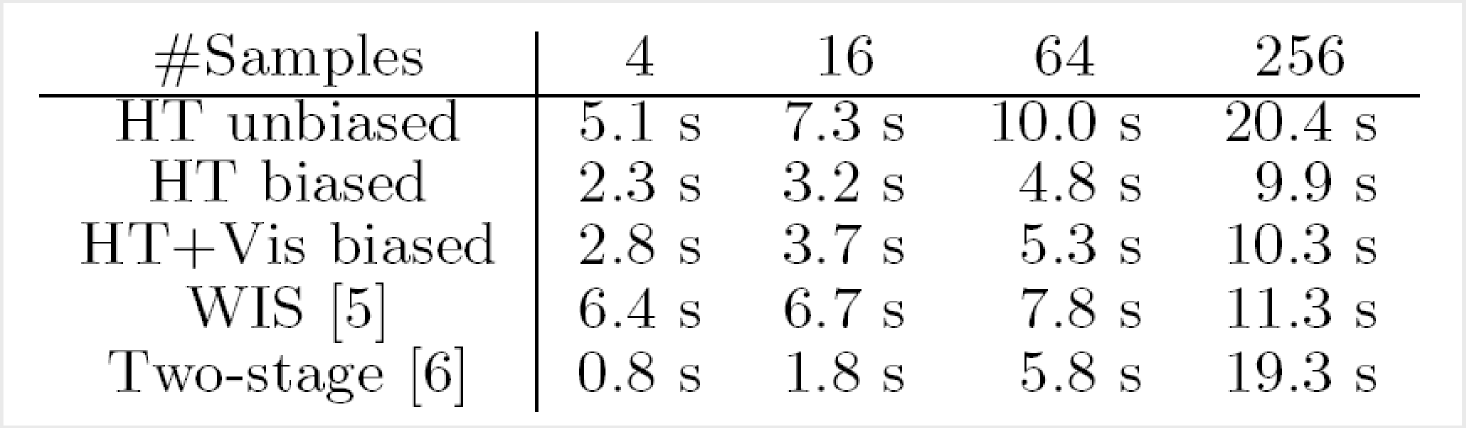

16 samples

Conclusion

- Positive
  - Fast sample generation
  - Simple (based on rejection sampling)
  - Flexible (functions must only be bounded)
  - Extendable to *n* functions
- Negative
  - Relies on precomputed BRDFs
- Future work
  - Compute the BRDF max on the fly
    (removes precomputation and dynamic rotations)
  - Smarter visibility computation

Rendering Times

- Core 2 Duo 2.4GHz (1 core)

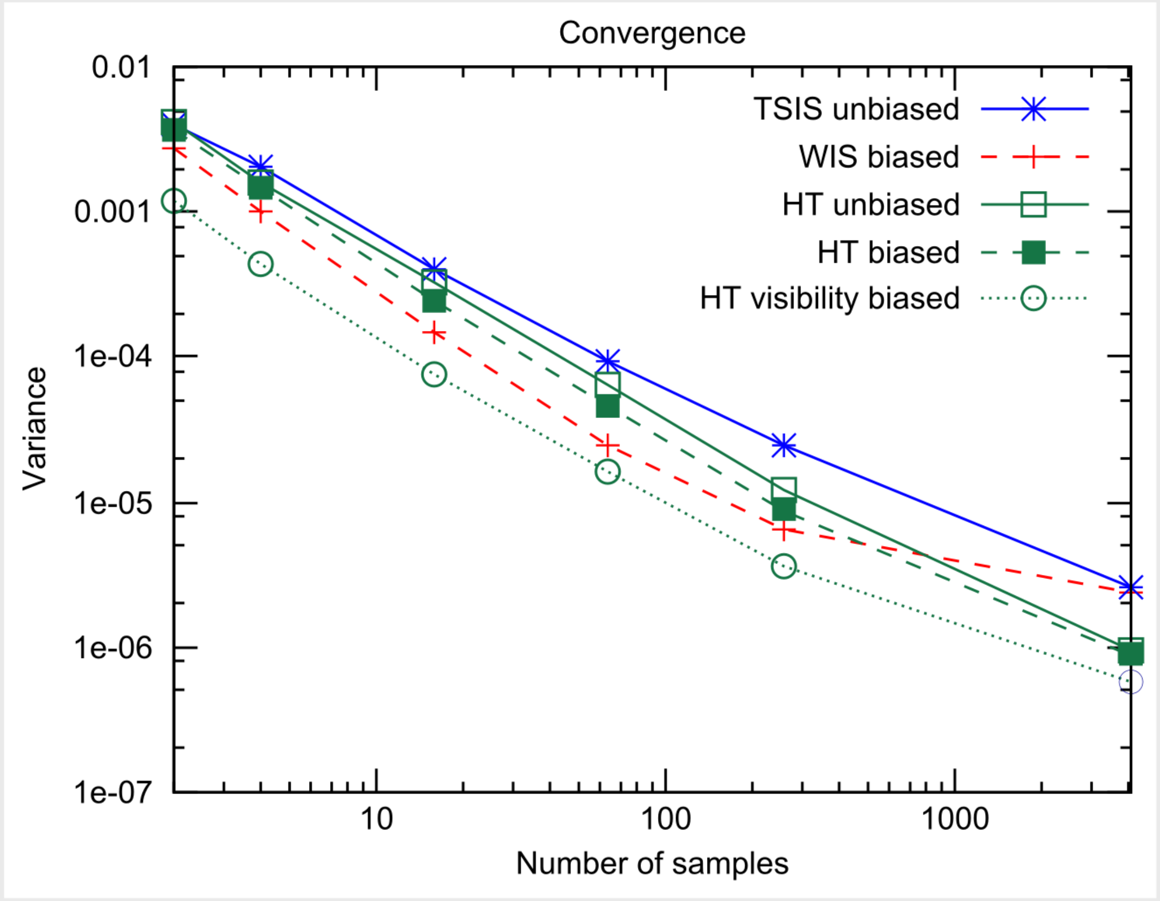


#Samples	4	16	64	256
HT unbiased	5.1 s	7.3 s	10.0 s	20.4 s
HT biased	2.3 s	3.2 s	4.8 s	9.9 s
HT+Vis biased	2.8 s	3.7 s	5.3 s	10.3 s
WIS [5]	6.4 s	6.7 s	7.8 s	11.3 s
Two-stage [6]	0.8 s	1.8 s	5.8 s	19.3 s

Questions

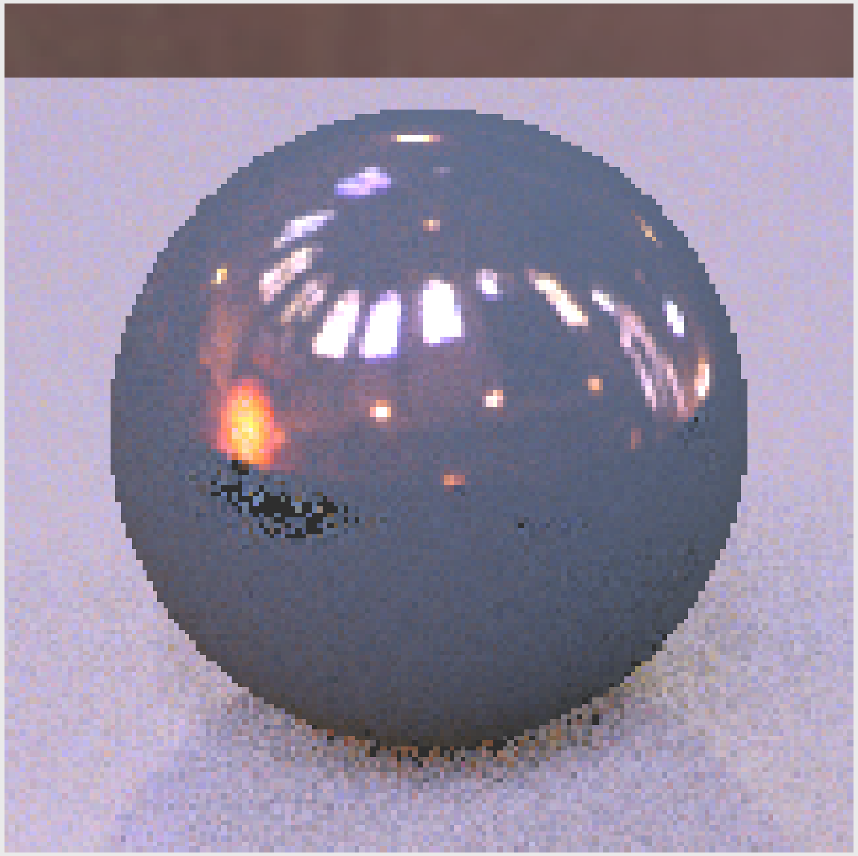
Results

- Comparison with other state of the art methods



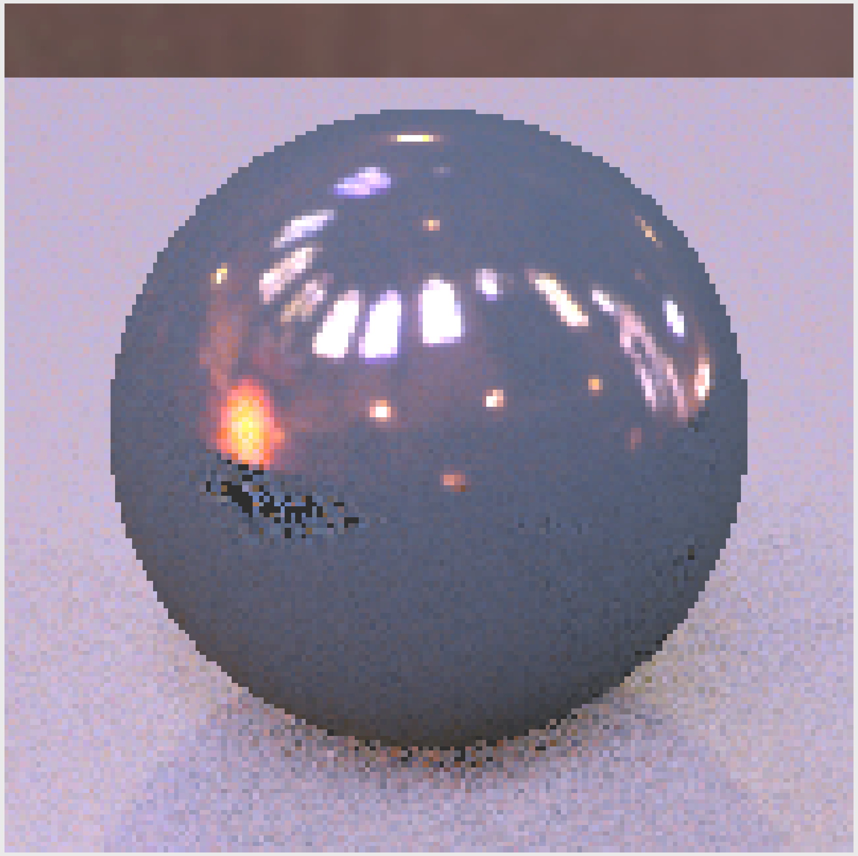
HT: Improving Performances

- Relies on a more agressive branch pruning: faster rendering
- Use the local average instead of computing the exact sample contribution: introduces a bias
- Seamlessy blend to exact sample contribution if the local average resolution is too coarse


unbiased

HT: Improving Performances

- Relies on a more agressive branch pruning: faster rendering
- Use the local average instead of computing the exact sample contribution: introduces a bias
- Seamlessy blend to exact sample contribution if the local average resolution is too coarse


biased

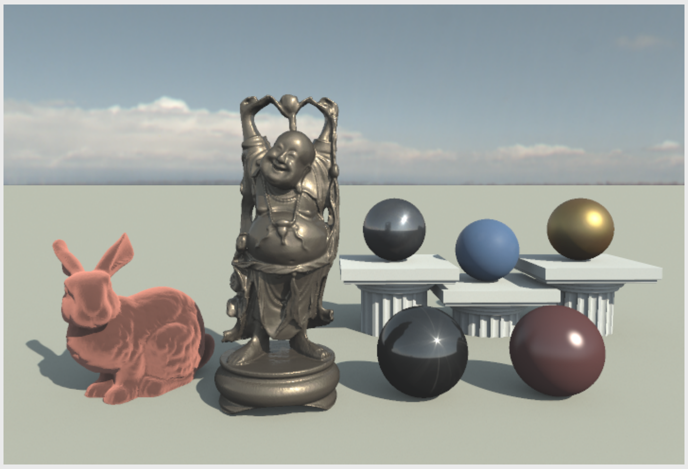
Efficient Product Sampling using Hierarchical Thresholding

Fabrice Rousselle
EPFL / University of Montreal

Petrik Clarberg
Lund University

Luc Leblanc, Victor Ostromoukhov, Pierre Poulin
University of Montreal

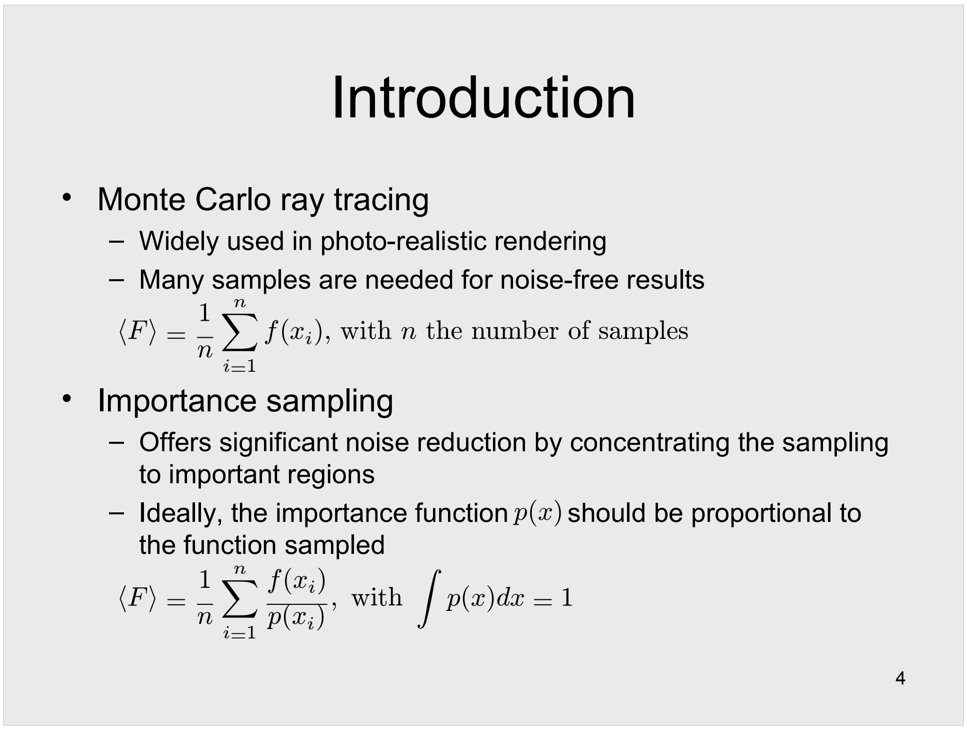
Objective



The objective of our work is to produce photo-realistic renderings such as this one using a wide range of surface reflectances going from the very diffuse (such as the ground) to the highly specular (such as the mirror ball in the front row), while lighting the scene with HDR environment maps. This map has a dynamic range of ~1:10^6.

Plan

1. Introduction
2. Description of Hierarchical Thresholding
   - Basic algorithm (Ostromoukhov *et al*, Siggraph 2004)
   - Our extensions
3. Application to direct illumination
4. Results
5. Conclusion and future work



Introduction

- Monte Carlo ray tracing
  - Widely used in photo-realistic rendering
  - Many samples are needed for noise-free results

$$\langle F \rangle = \frac{1}{n}\sum_{i=1}^{n} f(x_i), \text{ with } n \text{ the number of samples}$$

- Importance sampling
  - Offers significant noise reduction by concentrating the sampling to important regions
  - Ideally, the importance function $p(x)$ should be proportional to the function sampled

$$\langle F \rangle = \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)}, \text{ with } \int p(x)dx = 1$$

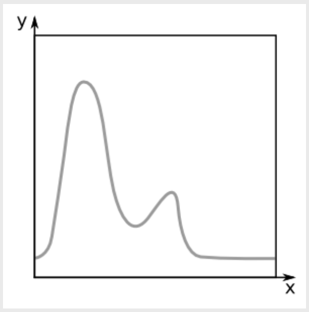
Standard Monte Carlo methods use random sampling, while importance sampling uses samples drawn from an importance function. Our method addresses the question of defining the importance function and drawing samples from it.

Introduction

- Proposition: Hierarchical Thresholding (HT)
  - A simple sampling scheme
  - Applicable to products of multiple functions
  - Can be easily integrated in a Monte Carlo ray tracer

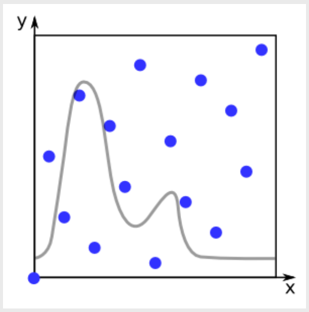
HT: Basic Algorithm

- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



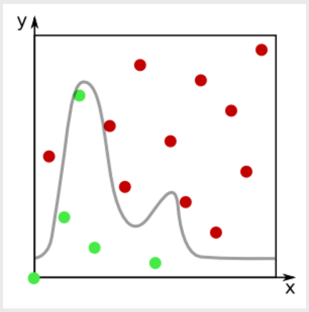
HT: Basic Algorithm

- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



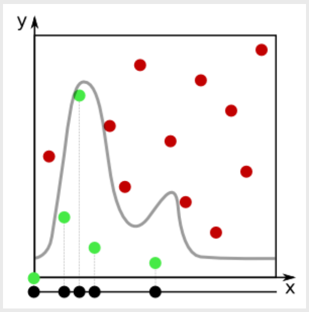
HT: Basic Algorithm

- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



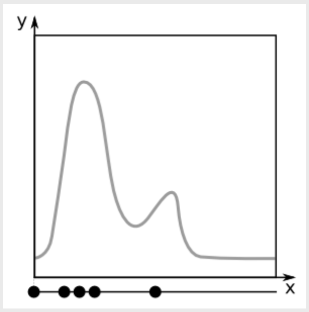
HT: Basic Algorithm

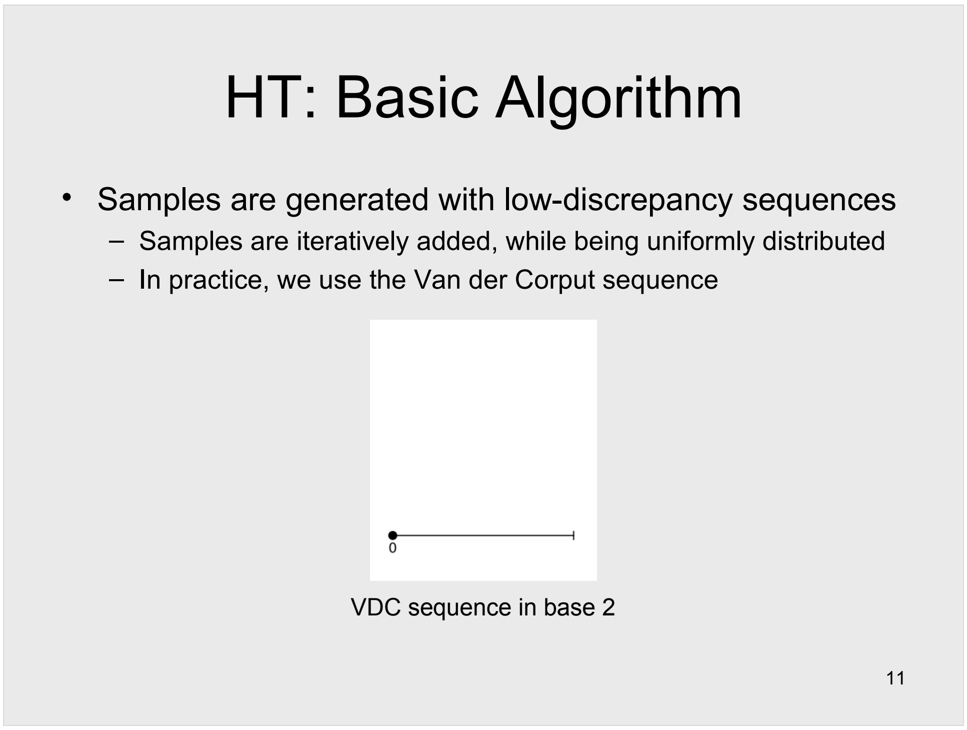
- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value



HT: Basic Algorithm

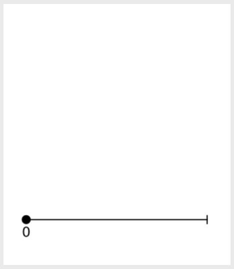
- HT is a rejection sampling scheme
  - Generate a uniform distribution
  - Reject all points outside the integral volume
  - The sample density is proportional to the function value





HT: Basic Algorithm

- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence

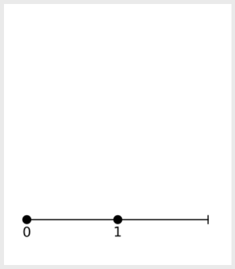


VDC sequence in base 2

Here, only the first 8 points of the sequence are shown. We seen that, as samples are added, the distribution remains relatively uniform.

HT: Basic Algorithm

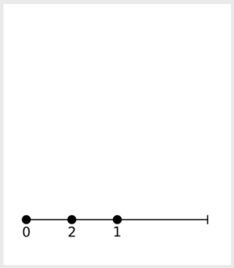
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

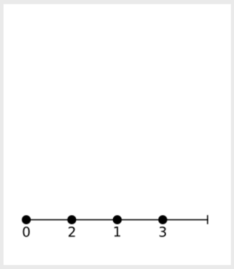
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

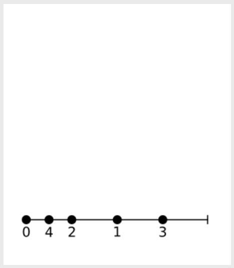
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

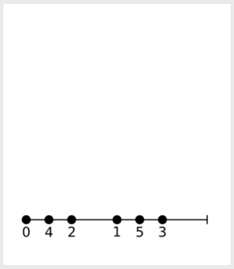
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

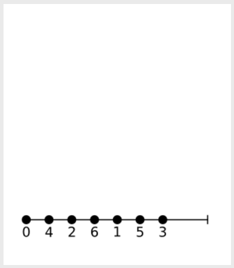
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



VDC sequence in base 2

HT: Basic Algorithm

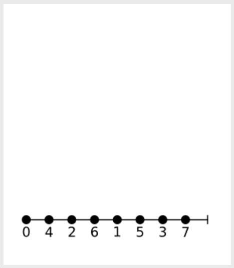
- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence



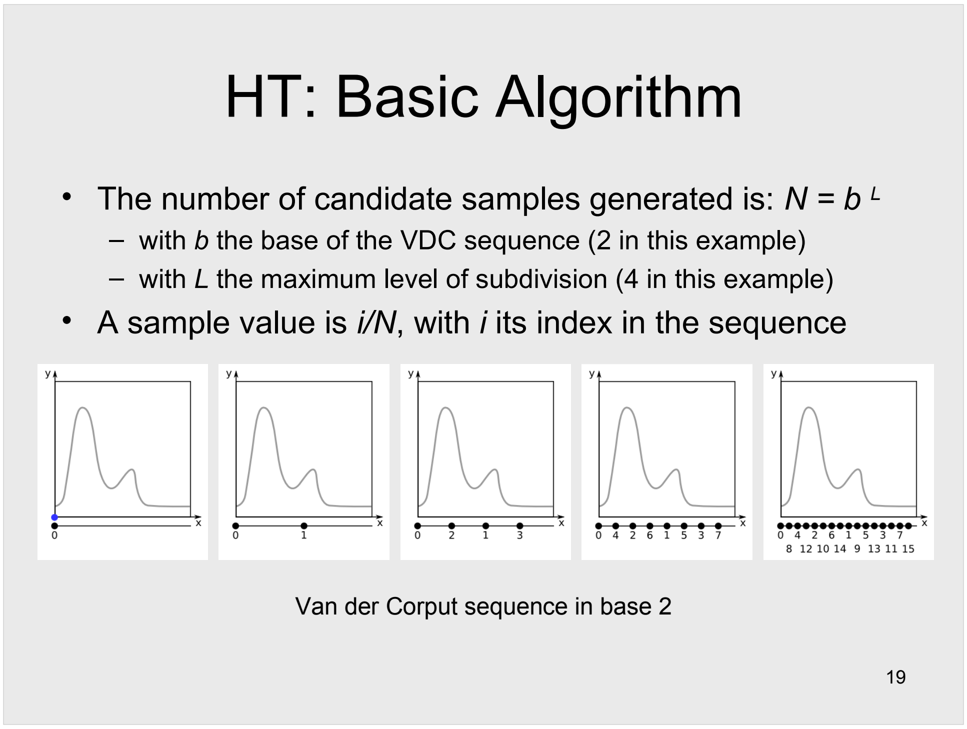
VDC sequence in base 2

HT: Basic Algorithm

- Samples are generated with low-discrepancy sequences
  - Samples are iteratively added, while being uniformly distributed
  - In practice, we use the Van der Corput sequence

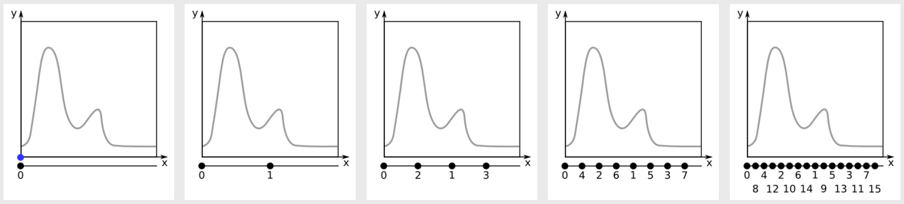


VDC sequence in base 2



HT: Basic Algorithm

- The number of candidate samples generated is: $N = b^L$
  - with b the base of the VDC sequence (2 in this example)
  - with L the maximum level of subdivision (4 in this example)
- A sample value is i/N, with i its index in the sequence

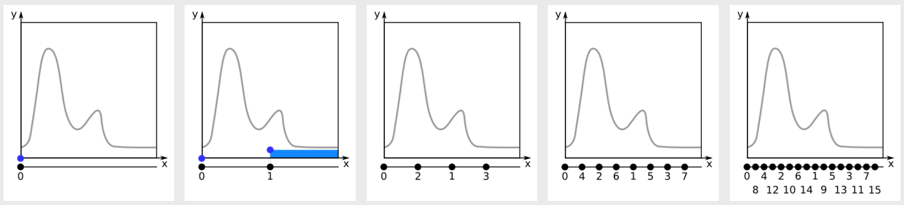


Van der Corput sequence in base 2

The VDC sequence generates points in 1D, we need to project them in 2D. As the projection uses the sample index as its value, we end up progressively filling the sampling domain, from the bottom to the top. This property is key and will be exploited to improve the efficiency of the algorithm.

HT: Basic Algorithm

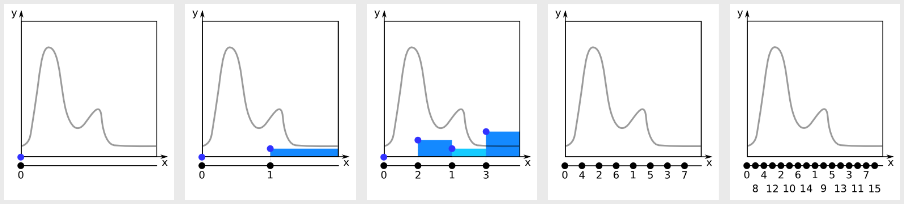
- The number of candidate samples generated is: $N = b^L$
  - with b the base of the VDC sequence (2 in this example)
  - with L the maximum level of subdivision (4 in this example)
- A sample value is i/N, with i its index in the sequence



Van der Corput sequence in base 2

HT: Basic Algorithm

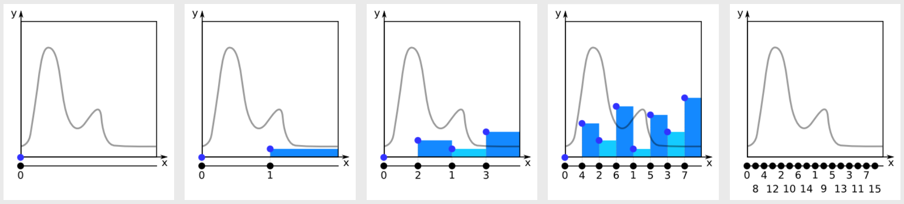
- The number of candidate samples generated is: $N = b^L$
  - with b the base of the VDC sequence (2 in this example)
  - with L the maximum level of subdivision (4 in this example)
- A sample value is i/N, with i its index in the sequence



Van der Corput sequence in base 2

HT: Basic Algorithm

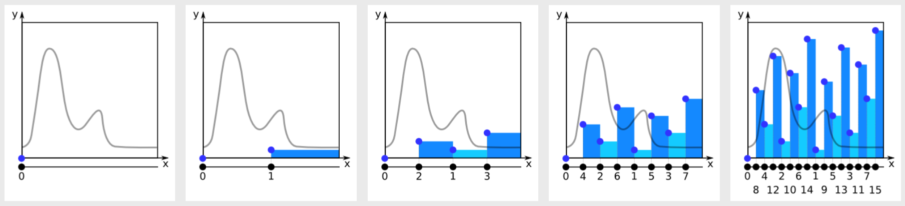
- The number of candidate samples generated is: $N = b^L$
  - with b the base of the VDC sequence (2 in this example)
  - with L the maximum level of subdivision (4 in this example)
- A sample value is i/N, with i its index in the sequence



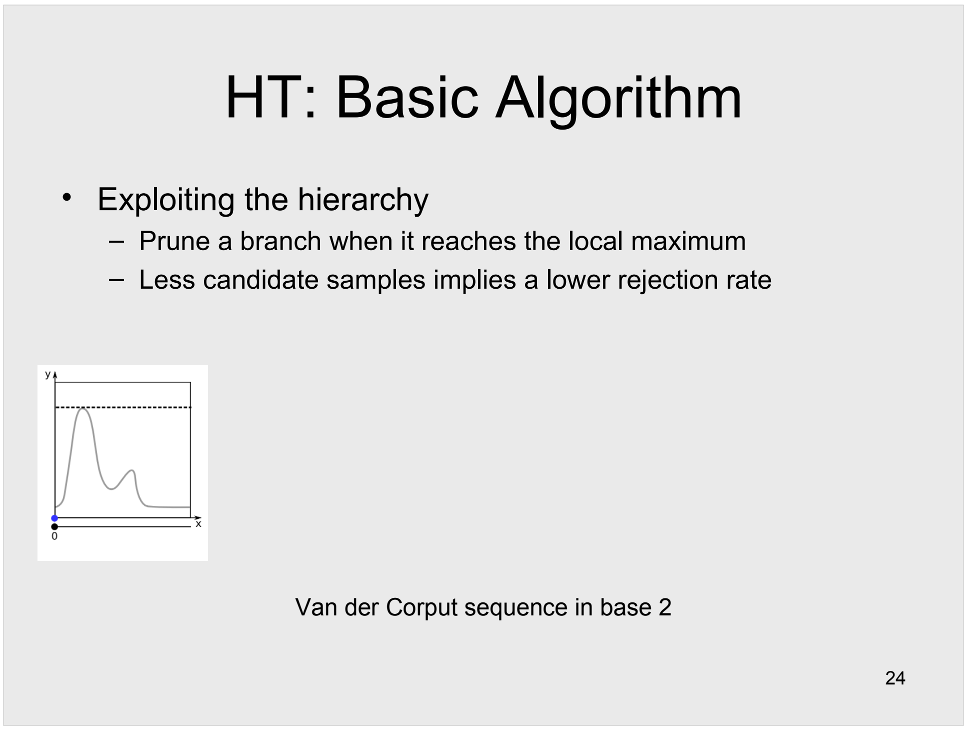
Van der Corput sequence in base 2

HT: Basic Algorithm

- The number of candidate samples generated is: $N = b^L$
  - with b the base of the VDC sequence (2 in this example)
  - with L the maximum level of subdivision (4 in this example)
- A sample value is i/N, with i its index in the sequence

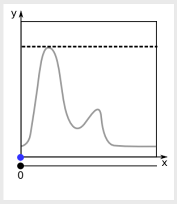


Van der Corput sequence in base 2



HT: Basic Algorithm

- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate

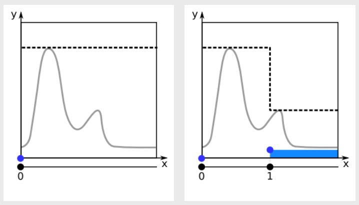


Van der Corput sequence in base 2

We add a constraint to the algorithm: we subdivide nodes until we reach the maximum level of subdivision (4 in this example) or the local maximum value.

HT: Basic Algorithm

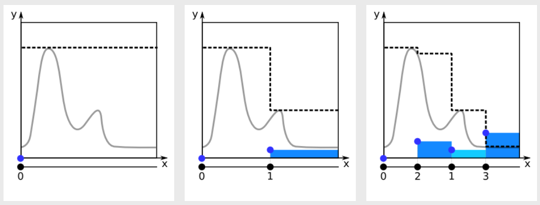
- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate



Van der Corput sequence in base 2

HT: Basic Algorithm

- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate

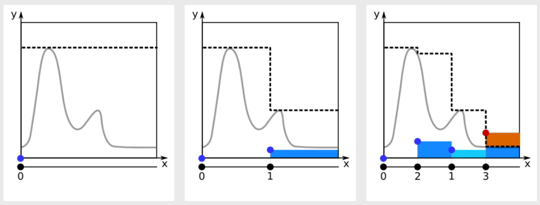


Van der Corput sequence in base 2

Here we see that the last node (on the right) has reached the local maximum. This node sample will necessarily be rejected and, as we progressively fill the domain, all subsequent samples would be placed « higher » and therefore be necessarily rejected. We therefore prune that branch (ie. we won't subdivide it anymore).

HT: Basic Algorithm

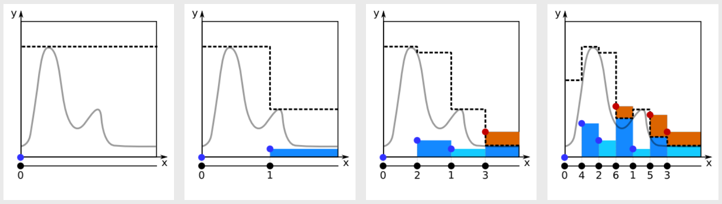
- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate



Van der Corput sequence in base 2

HT: Basic Algorithm

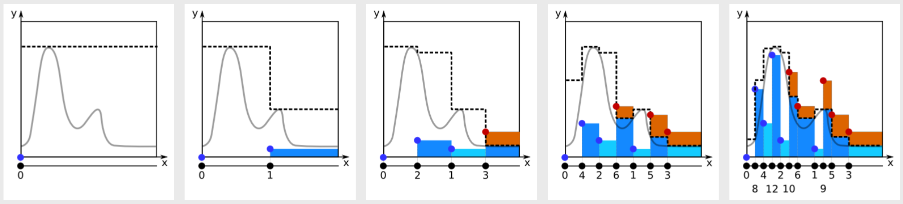
- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate



Van der Corput sequence in base 2

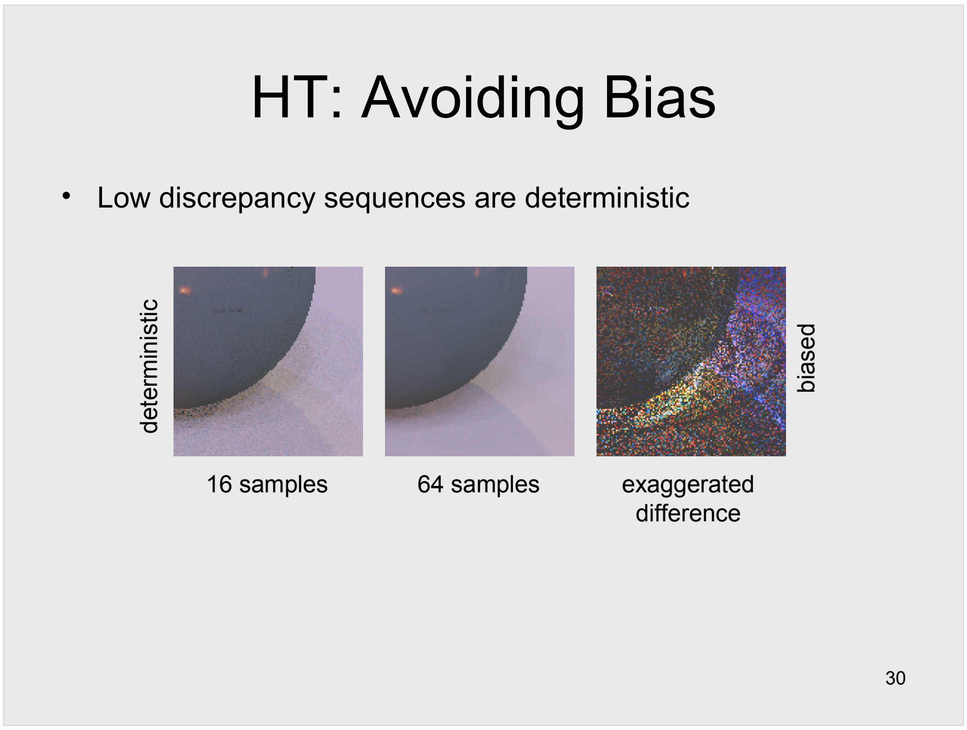
HT: Basic Algorithm

- Exploiting the hierarchy
  - Prune a branch when it reaches the local maximum
  - Less candidate samples implies a lower rejection rate



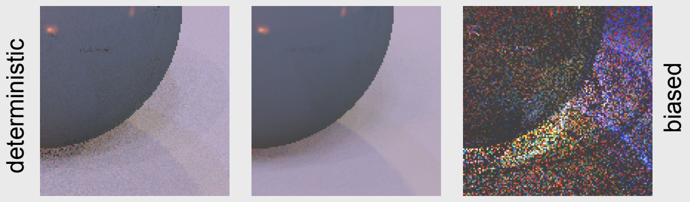
Van der Corput sequence in base 2

In this example, we ended up with 11 instead of 16 candidate samples



HT: Avoiding Bias

- Low discrepancy sequences are deterministic

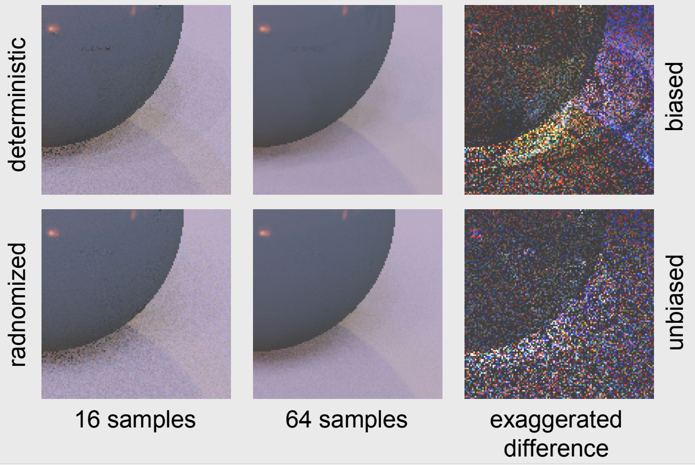


16 samples 64 samples exaggerated difference

Sample distribution is deterministic. This induces a local coherence in the image, which creates a bias. Shown here: the visual difference between an image rendered using 16 and 64 samples. The bias in sample placement yields a color shift (blue in the top right and yellow in the middle).

HT: Avoiding Bias

- Low discrepancy sequences are deterministic
  - Randomize key steps of the processus (see paper for details)



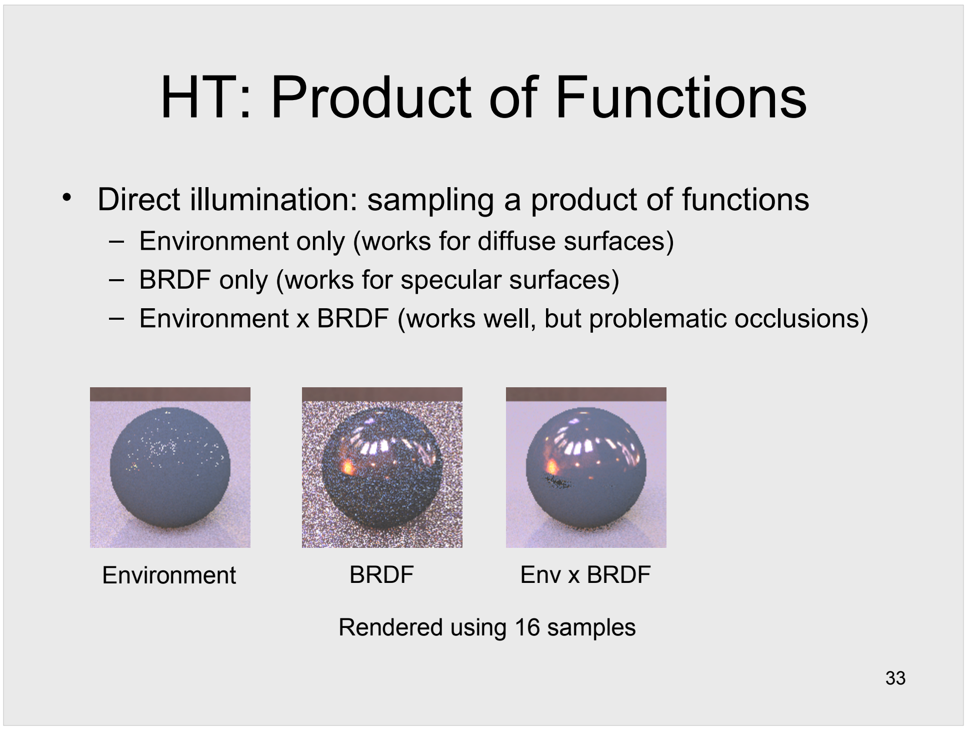
31

By randomizing some key steps of the algorithm, the distribution is effectively randomized, yielding an unbiased distribution. The visual difference is now « white noise » illustrating the unbiased estimation. The randomizing does not affect the uniformity of sample distribution and the fact that it progressively fills the domain.

HT: Use of Max/Avg Tree

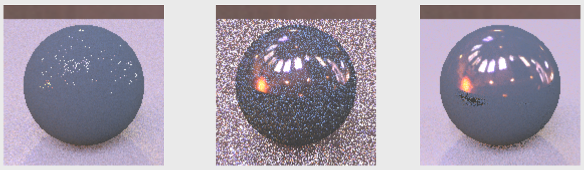
- Local maximum value
  - Needed for pruning branches
- Local average value
  - Predict the number of selected samples
  - Speed up the rejection test (avoid BRDF evaluation)
  - Perform further performance optimizations
  - Yields a piecewise-constant importance function: $h(x)$

- The local maximum and average values are pre-computed for all nodes of the hierarchy



HT: Product of Functions

- Direct illumination: sampling a product of functions
  - Environment only (works for diffuse surfaces)
  - BRDF only (works for specular surfaces)
  - Environment x BRDF (works well, but problematic occlusions)



Environment BRDF Env x BRDF

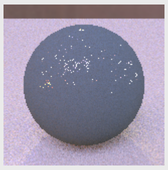
Rendered using 16 samples

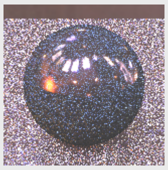
33

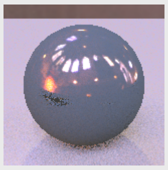
The point of this slide is to illustrate what can be gained by sampling the product of functions.

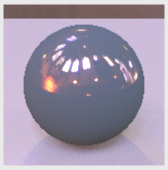
HT: Product of Functions

- Direct illumination: sampling a product of functions
  - Environment only (works for diffuse surfaces)
  - BRDF only (works for specular surfaces)
  - Environment x BRDF (works well, but problematic occlusions)
  - Environment x BRDF x visibility


Environment


BRDF


Env x BRDF


Env x BRDF x Vis

Rendered using 16 samples

HT: Product of Functions

- Our algorithm extends to the product of n functions:

$$f(x) = \prod_{i=1}^{n} f_i(x) \tag{1}$$

- The local maximum is conservatively approximated:

$$\max f(x) \leq \prod_{i=1}^{n} \left(\max f_i(x)\right), \qquad x \in [a, b] \tag{2}$$

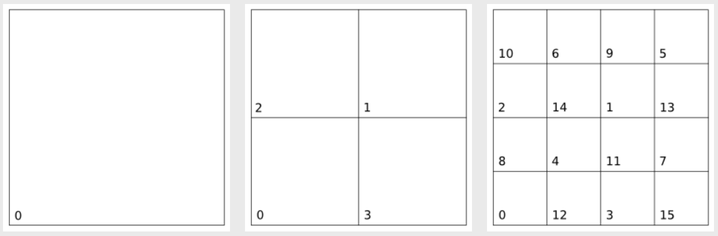
- The local average is similarly approximated:

$$\text{avg } f(x) \approx \prod_{i=1}^{n} \left(\text{avg } f_i(x)\right), \qquad x \in [a, b] \tag{3}$$

  – The error of this approximation only affects the variance of the estimation, the estimator still convergences to the right result

HT: Sampling in 3D

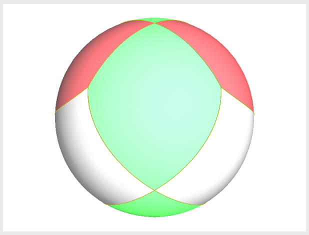
- Samples are positioned on a 2D plane

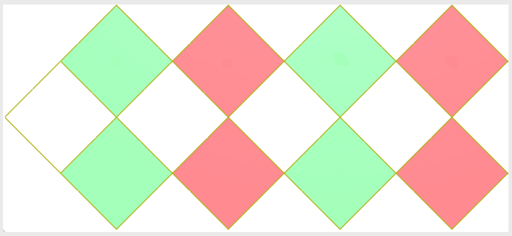


Sample indexing according to the VDC sequence in base 4

Application: Direct Illumination

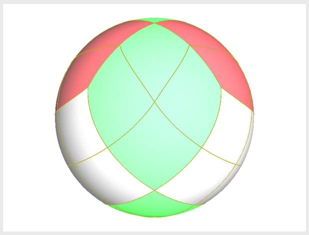
- The mapping used: HEALPix
  - hierarchical representation (needed for our algorithm)
  - area preservation
  - low distorsion (allows dynamic rotations)
- Each face is sampled individually

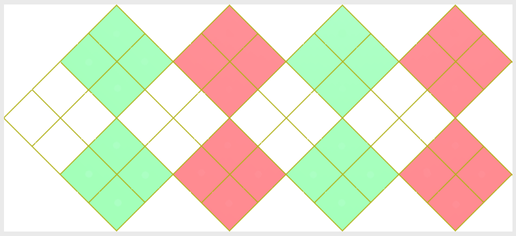




Application: Direct Illumination

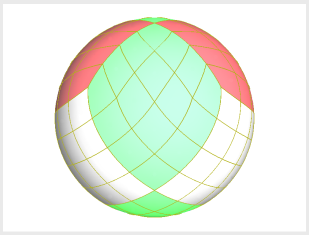
- The mapping used: HEALPix
  - hierarchical representation (needed for our algorithm)
  - area preservation
  - low distorsion (allows dynamic rotations)
- Each face is sampled individually

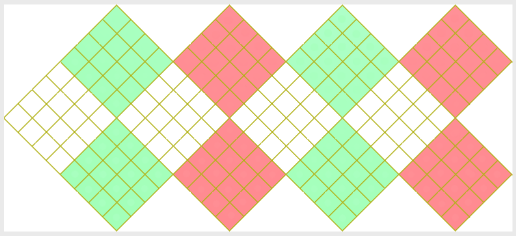




Application: Direct Illumination

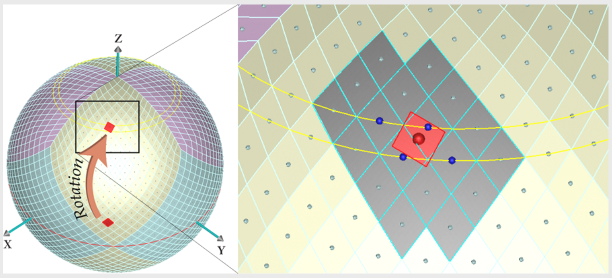
- The mapping used: HEALPix
  - hierarchical representation (needed for our algorithm)
  - area preservation
  - low distorsion (allows dynamic rotations)
- Each face is sampled individually





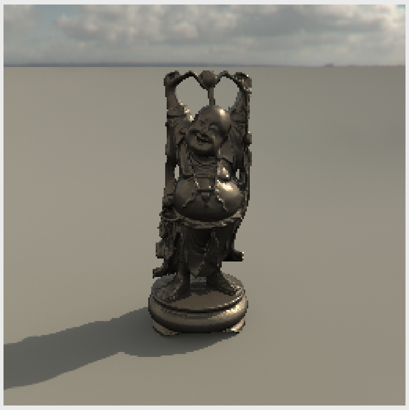
HEALPix: Rotations

- Environment Map in world space
- BRDF in local space
  - Fetching the BRDF value implies a rotation
  - The max area is enlarged to remain conservative
  - The avg is interpolated from the four neighbors



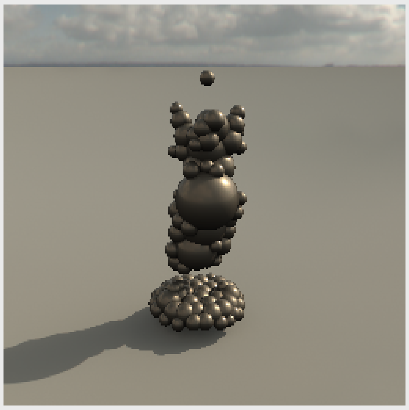
Application: Visibility Computation

- Computed from a simplified model using inner spheres



Application: Visibility Computation

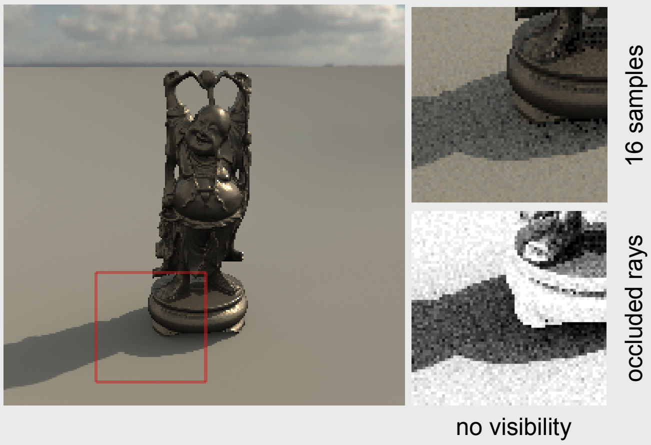
- Computed from a simplified model using inner spheres



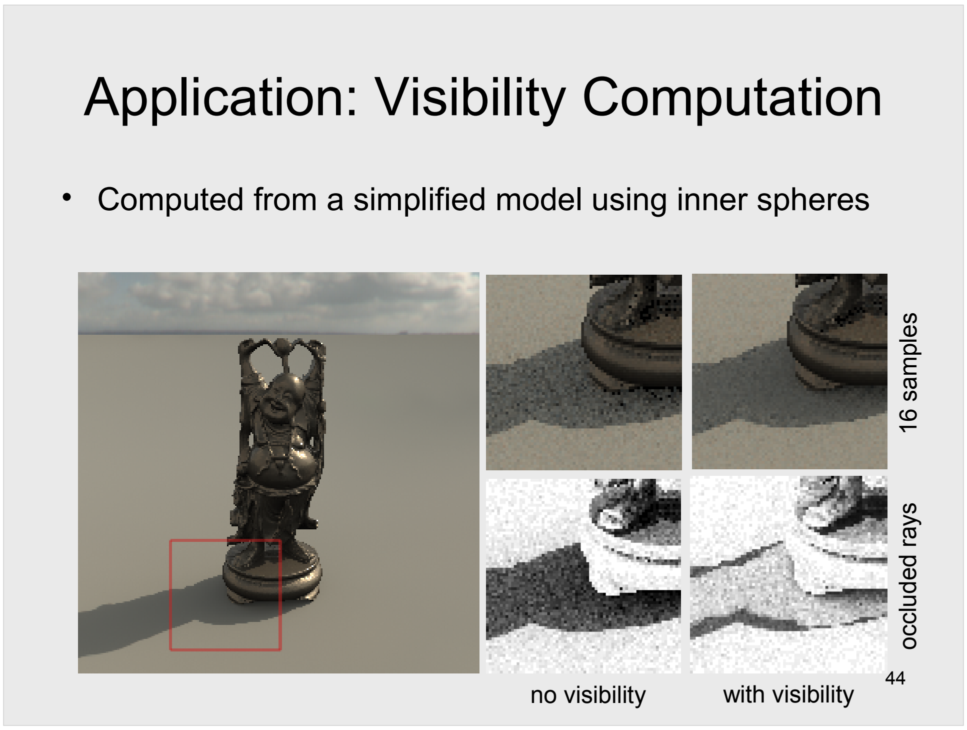
The inner sphere model gives a conservative estimate of the real model shadow.

Application: Visibility Computation

- Computed from a simplified model using inner spheres



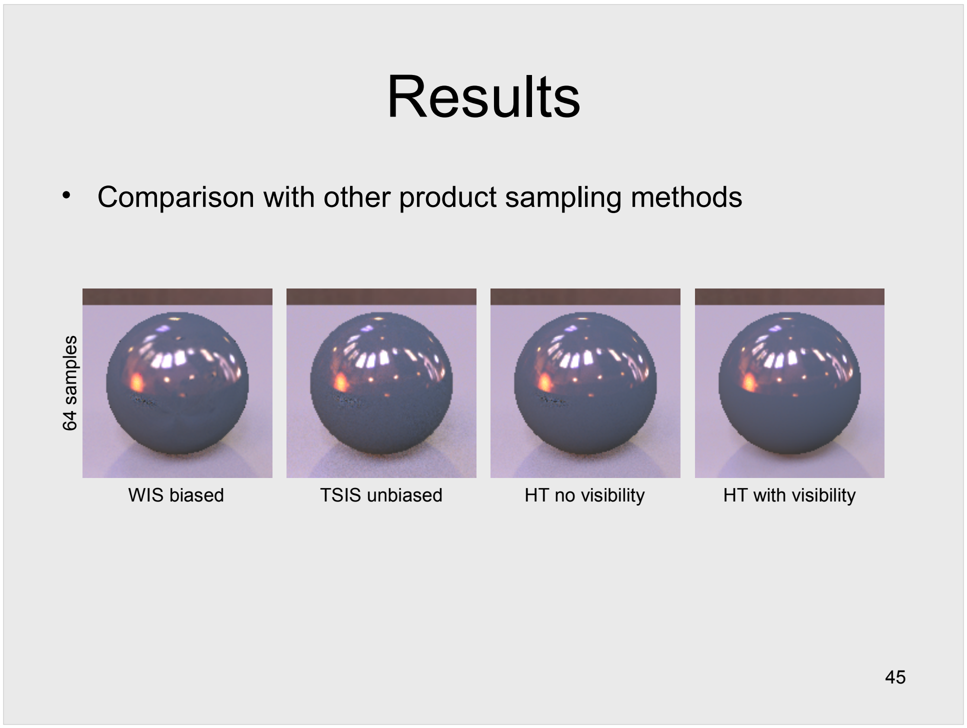
Noise level in the shadow is increased: samples are drawn to the sun which is occluded by the happy buddha. This shows in the color coded image at the bottom (the darker the image, the higher the percentage of occluded rays).



Application: Visibility Computation

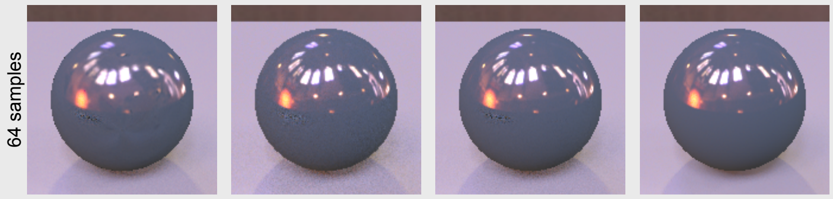
- Computed from a simplified model using inner spheres

Using the occlusion information of the spheres, we can significantly reduce the percentage of occluded rays. As we use a conservative estimate, we cannot improve the behavior at the edges of the shadow or in between the feet of the buddha.



Results

- Comparison with other product sampling methods

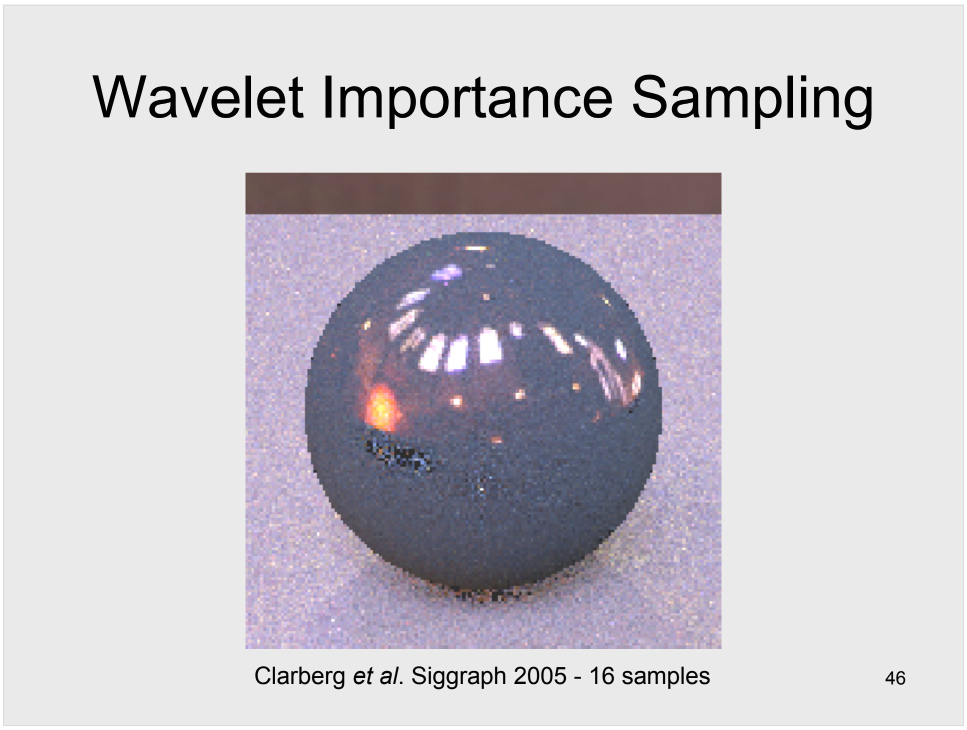


WIS biased | TSIS unbiased | HT no visibility | HT with visibility

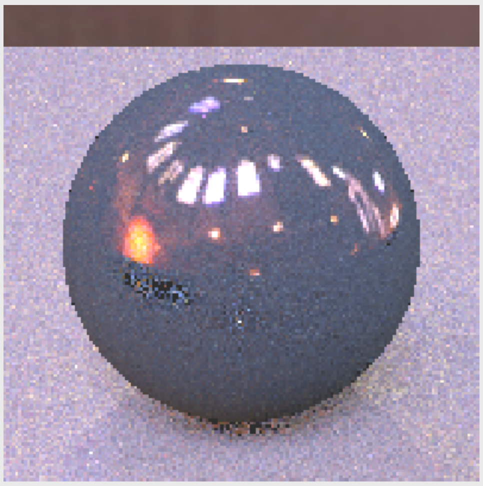
45

Just show the 4 usefull ones

If the previous slide is removed, biased results should also be removed from this comparison



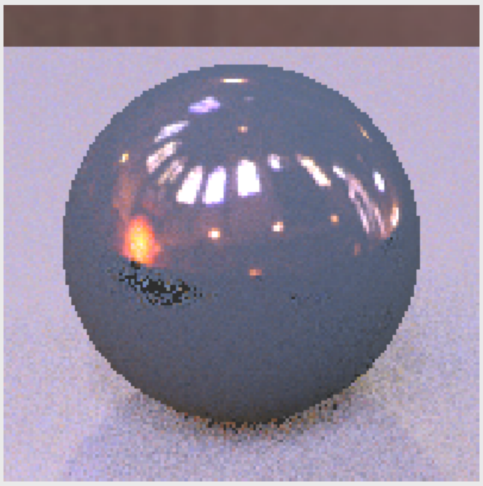
Wavelet Importance Sampling

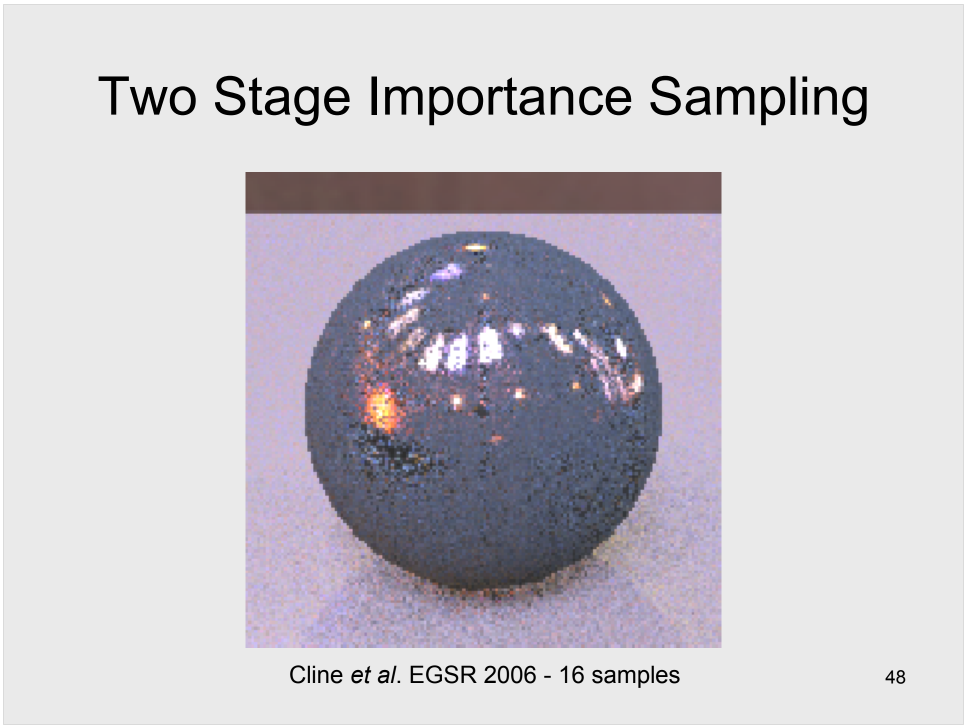


Clarberg *et al*. Siggraph 2005 - 16 samples

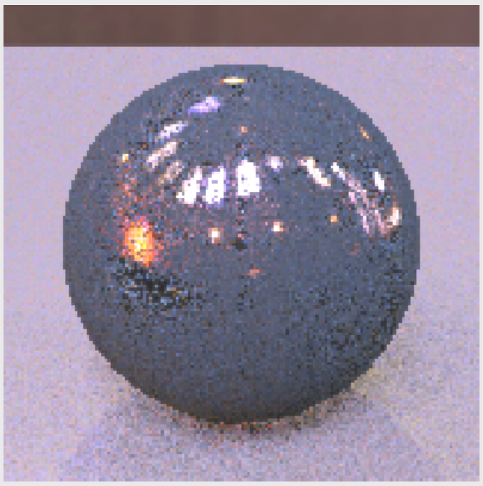
WIS relies on relatively low-resolution environment maps, resulting in a less accurate importance function and therefore higher level of noise.

HT: no visibility


16 samples



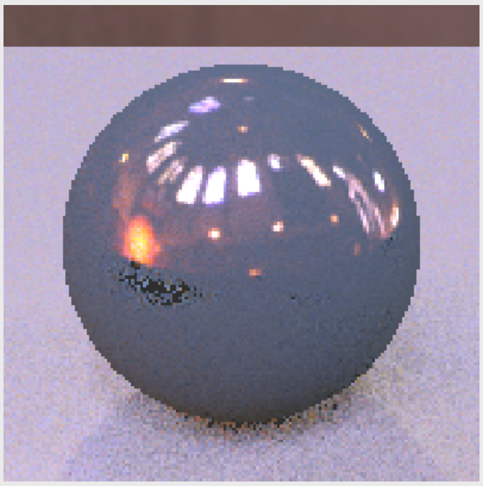
Two Stage Importance Sampling



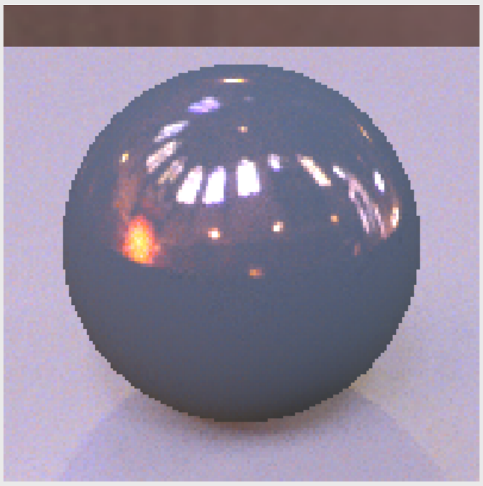
Cline *et al*. EGSR 2006 - 16 samples

While TSIS uses high resolution environment maps (which allow for high quality rendering of the plane) its refining approach does not work well for highly specular surfaces when using a small number of samples.

HT: no visibility


16 samples

HT: with visibility

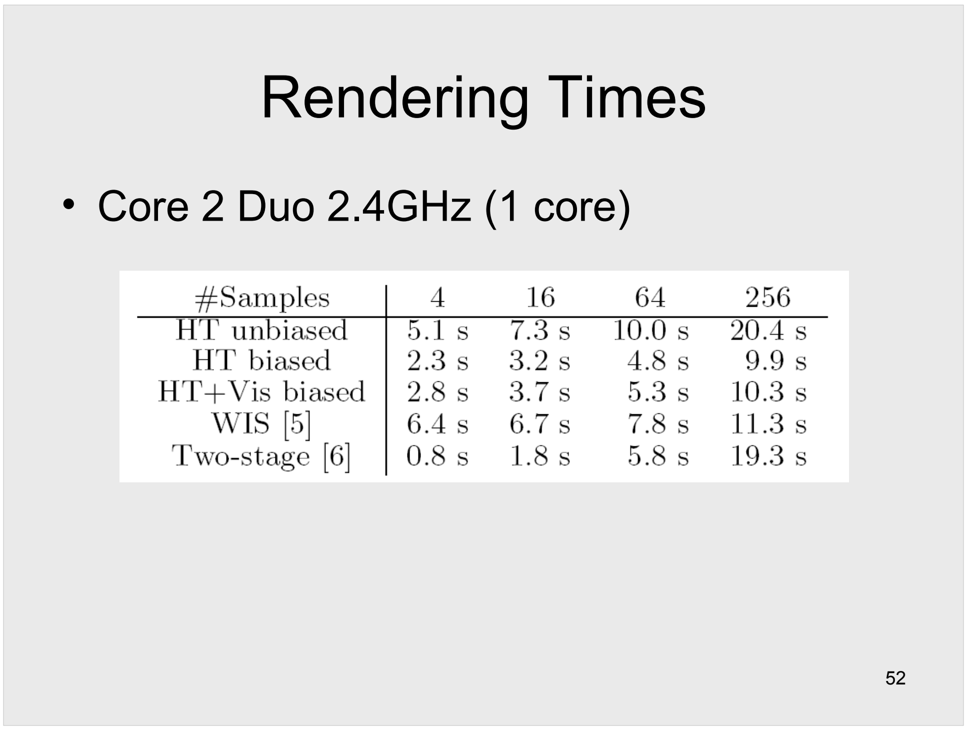


16 samples

This is the ideal case for our method: the visibility can be accurately computed as we have only an analytical sphere on an infinite plane. As such this represents the extend to which the visibility could improve the estimation.

Conclusion

- Positive
  - Fast sample generation
  - Simple (based on rejection sampling)
  - Flexible (functions must only be bounded)
  - Extendable to *n* functions
- Negative
  - Relies on precomputed BRDFs
- Future work
  - Compute the BRDF max on the fly
    (removes precomputation and dynamic rotations)
  - Smarter visibility computation



Rendering Times

- Core 2 Duo 2.4GHz (1 core)

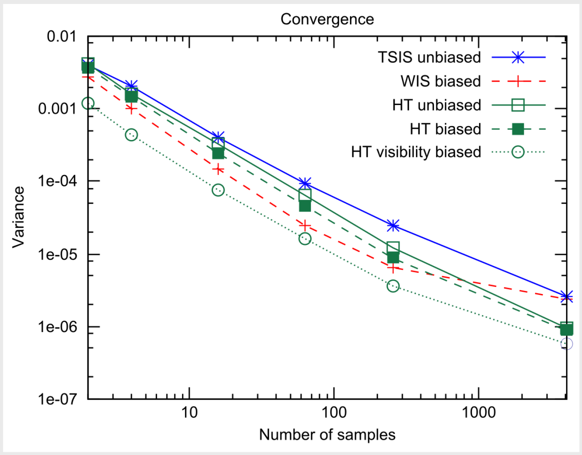
#Samples	4	16	64	256
HT unbiased	5.1 s	7.3 s	10.0 s	20.4 s
HT biased	2.3 s	3.2 s	4.8 s	9.9 s
HT+Vis biased	2.8 s	3.7 s	5.3 s	10.3 s
WIS [5]	6.4 s	6.7 s	7.8 s	11.3 s
Two-stage [6]	0.8 s	1.8 s	5.8 s	19.3 s

All methods have comparable speeds. The « HT biased » method is detailed in the paper and illustrated at the end of the presentation.

Questions

Results

- Comparison with other state of the art methods

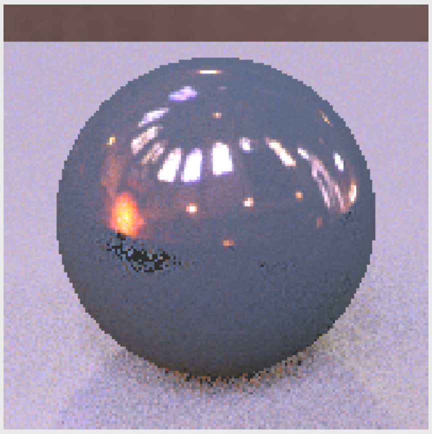


Shown here for WIS is the biased rendering only which does not converge to the right value. The unbiased version would have higher variance than all methods shown here.

This graph also illustrates the seemless blend from biased to unbiased values for our biased implementation, ensuring that we converge to the same value.

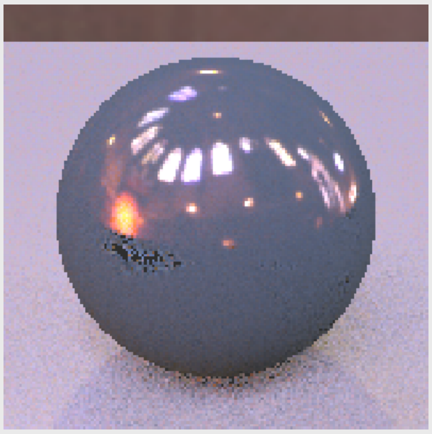
HT: Improving Performances

- Relies on a more agressive branch pruning: faster rendering
- Use the local average instead of computing the exact sample contribution: introduces a bias
- Seamlessy blend to exact sample contribution if the local average resolution is too coarse


unbiased

HT: Improving Performances

- Relies on a more agressive branch pruning: faster rendering
- Use the local average instead of computing the exact sample contribution: introduces a bias
- Seamlessy blend to exact sample contribution if the local average resolution is too coarse


biased