SIGGRAPH2014
NATURALLY DIGITAL
vancouver

SIGGRAPH2014
vancouver
NATURALLY DIGITAL

The 41st International Conference and Exhibition on Computer Graphics and Interactive Techniques

Karlsruhe Institute of Technology

Comparison of Advanced Light Transport Methods

Anton S. Kaplanyan

Karlsruhe Institute of Technology, Lightrig

METROPOLIS-HASTINGS ALGORITHM

Metropolis-Hastings (MH) Algorithm

- Goal: Random walk according to a desired function f
- Posterior distribution is then proportional to f
  - Accurate to a scaling factor = normalization constant
- Define conditional rejection sampling probability $\min(1, a_{i \to j})$ as

$$a_{i \to j} = \frac{f(x_j)}{f(x_i)} = \frac{f_j}{f_i}$$

  - $a_{i \to j}$ is *acceptance probability* at state i for proposal state j
- Detailed balance $a_{i \to j} P_{i \to j} = a_{j \to i} P_{j \to i}$

4

Assume we have some function f that can be evaluated point-wise. We cannot directly sample from f. However, we still want to sample proportionally to f.
We can construct a random walk, whose posterior distribution converges to the function of interest f accurate to the normalization factor (since f is generally not normalized).
The idea is to run a Markov chain and define a transition distribution of the random walk by a conditional rejection sampling based on ratio of the values of f. This is similar to the ordinary rejection sampling. However, the key difference is that it is conditional on the current state, that is, $a_{i \to j}$ is a conditional probability. At each move n we maintain a Markov chain by either accepting the new proposal state j with probability $a_{i \to j}$ or otherwise rejecting it and keeping the previous state i.
The transition probability should be selected such that the detailed balance is obeyed.

Importance Sampling for M-H

- Cannot fetch proposals directly from f
- Generate a proposal j from some *proposal distribution* T
  - T can depend on the current state i: $T_{i \to j}$
  - Similar to importance sampling in Monte Carlo
- Acceptance probability is then (from detailed balance):

$$a_{i \to j} = \left(\frac{f_j}{T_{i \to j}}\right) / \left(\frac{f_i}{T_{j \to i}}\right)$$

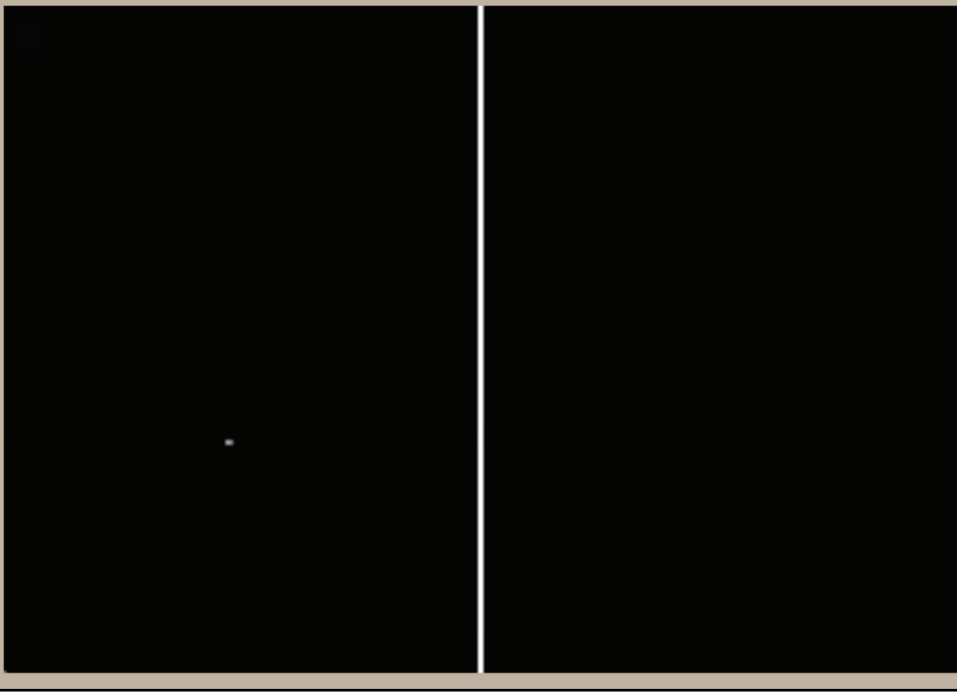
5

In practical situations, doing just a uniform random walk around the state space might lead to a poor exploration (important features might be missing or under-sampled).
In this case, one can generate new proposals with some *proposal distribution T*, which is somewhat similar to *f*. The proposal distribution *T* is the conditional probability density of proposing a state x_j given x_i. This is almost equivalent to the importance sampling technique used in Monte Carlo.

The key difference is that such importance function in MCMC can conditionally depend and rely on the current state.

And the acceptance rate should account for this transition probability similarly to Monte Carlo methods (by dividing the value of *f* by the probability of sampling it).

Example: MH Replicates an Image



6

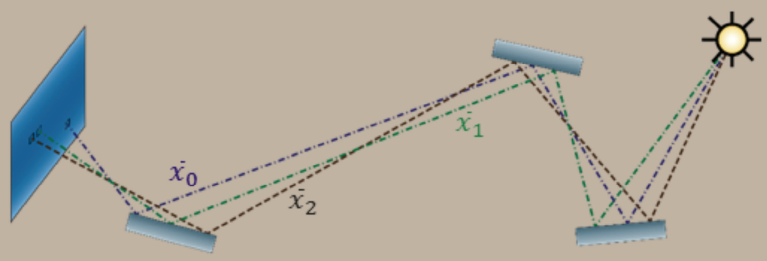
Here we show the MH algorithm on a simple example of copying an unknown image. The image (on the left) is a black box function and Markov chain evaluates the image only point-wise at the current and the proposal position. The image pixels *are* the states of the Markov chain. The proposal is selected from a narrow normal distribution around the current pixel. The posterior histogram (so called “trace”) of the random walk is shown on the right and is always normalized.
If we increase the step size, then the image exploration improves significantly.
After speeding up the process, the histogram on the right finally converges to the actual unknown image.

METROPOLIS LIGHT TRANSPORT

Now, let us see how to apply MH algorithm in the context of light transport.

Metropolis Light Transport

1. Generate initial path $\bar{x}_0$ using PT/BDPT
2. Mutate with some proposal distribution $T_{\bar{x}_i \to \bar{x}_j}$
3. Accept new path $\bar{x}_j$ with probability $a_{\bar{x}_i \to \bar{x}_j}$
4. Accumulate contribution to the image plane
5. Go to step 2



8

First, our Markov chain states are the complete light paths connecting light source to the camera. We want to mutate this light path and walk in the space of all possible paths according to the energy carried by these paths to the image plane.
The outline of MLT is

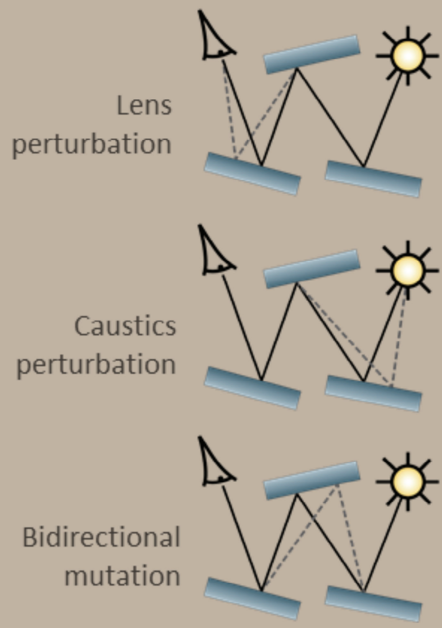
1. We generate an initial state (full path) of a Markov chain using one of the existing sampling methods (e.g. PT or BDPT).
2. Then we start the actual mutation process. Mutate the current path using one of the available mutation strategies, compute the proposal density.
3. Compute the acceptance probability (discussed earlier), accept the new proposed path according to this probability.
4. Accumulate the contribution of the current path to the image plane, apply pixel filter to bin the path into the corresponding pixel in-place.
5. Proceed to step 2 to cycle the random walk.

EXISTING MUTATION STRATEGIES

Here comes an overview of the existing mutation strategies.

Veach Mutations

- Minimal changes to the path
  - Lens, caustics, multi-chain perturbations
- Large changes to the path
  - Bidirectional mutation
    - BDPT-like large step
  - Lens mutation
    - stratified seeding on the image plane



10

Eric Veach has first introduced MLT and proposed the original set of mutation strategies.
The first group perturbs the current path slightly, thus such mutations are called perturbations.
They are crafted to efficiently explore certain effects such as caustics and chains of them.
Another group of mutations tries to do large changes to the path.
 - Namely, bidirectional mutation works similarly to BDPT, but it resamples only a subpath of the current path.
 - Lens mutation reseeds the chain with a path from the pool of paths stratified over the image plane.

Kelemen Mutation

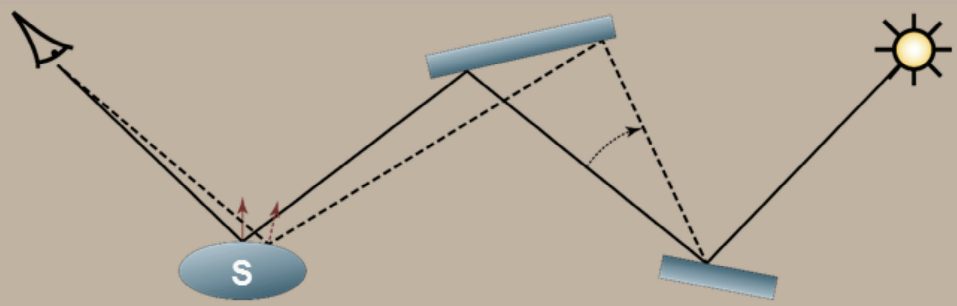
- Mutate a “random” vector that maps to a path
- Use the “random” vector for importance pdfs
  - Primary space: importance function domain
  - Assume the importance sampling is good

11

A popular mutation proposed by Kelemen is to perturb the path in a so-called primary sample space, that is, the original space of the importance functions used for constructing the path in BDPT and PT.
Usually it is represented as a vector of random numbers in the unit hypercube, which is perturbed using some symmetric probability, like a multidimensional Gaussian distribution.
The major assumption is that the importance sampling functions already make the integrand flat enough that we can walk it using some uniform random walk in this primary space. This mutation is quite easy to implement.

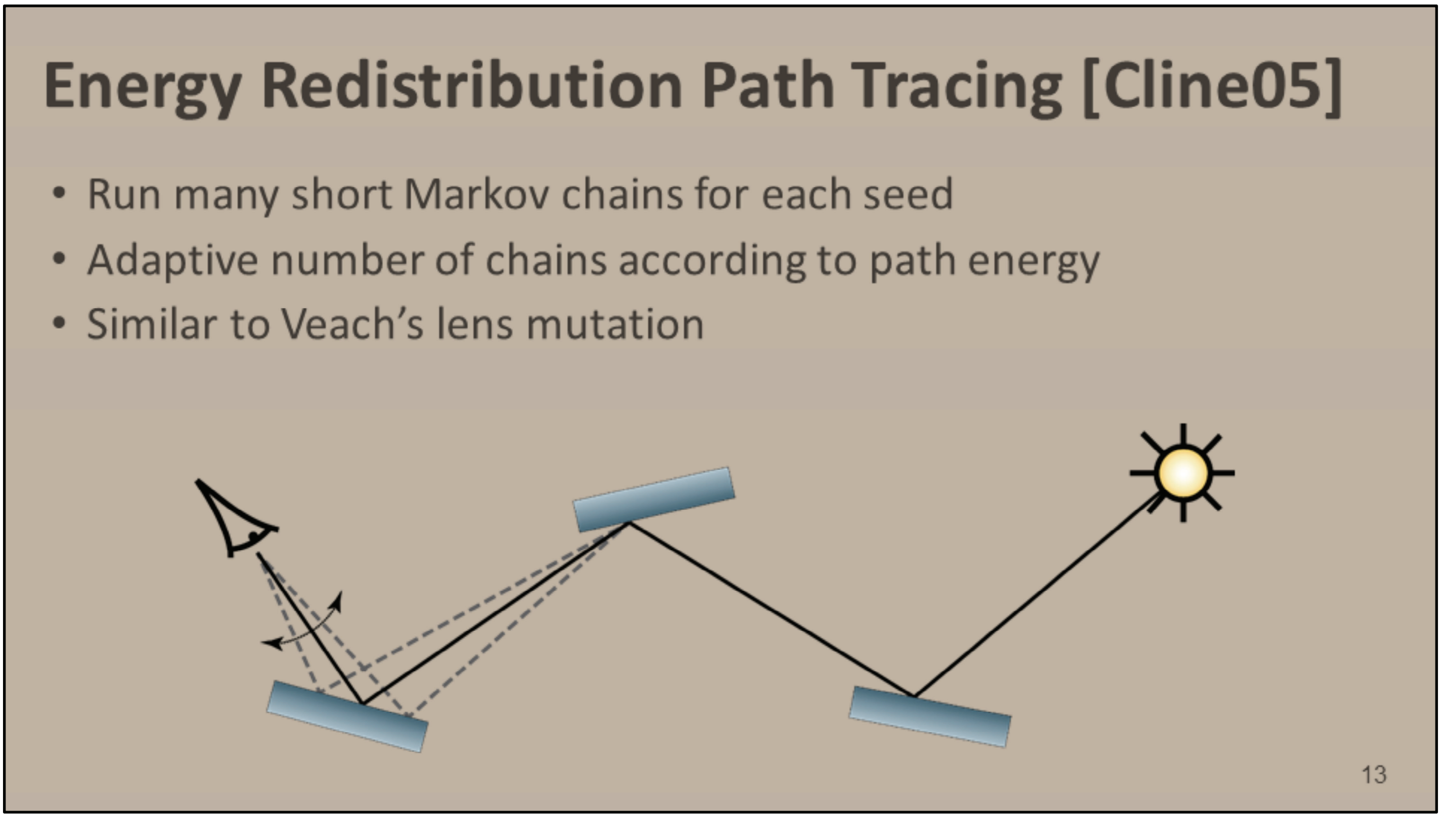
Manifold Exploration Mutation

- Uses local parameterization of current path
- Can connect through a specular chain
- Fixes integration dimensions
  - Tries to obey specular constraints



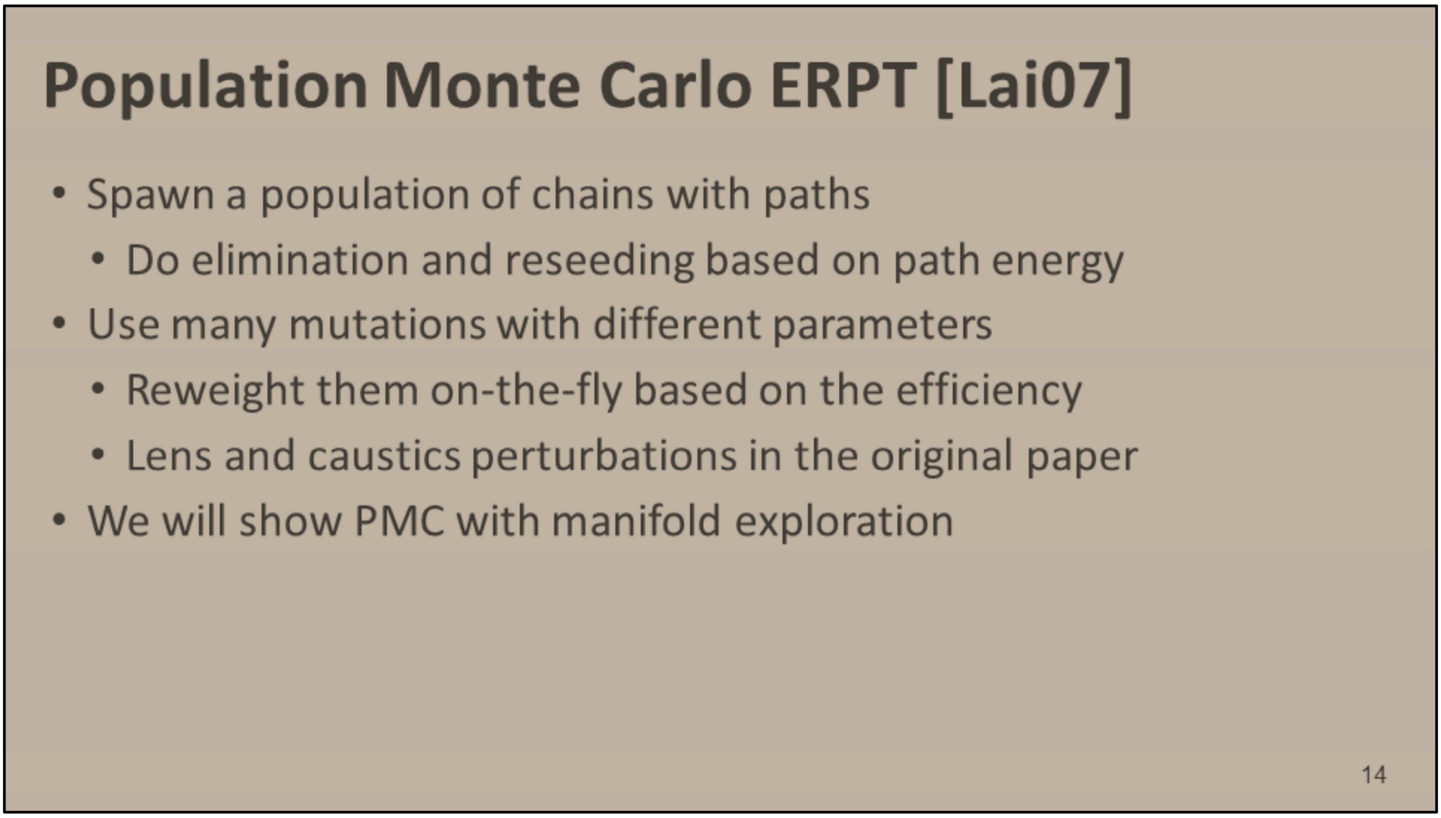
12

Yet another recent mutation strategy is called manifold exploration.
This supplementary mutation strategy is designed to replace the set of Veach's perturbations.
The idea is to perturb the path starting from some vertex, and then, in order to construct the new subpath, it first computes a local on-surface tangent frame around the current path and then tries to iteratively construct the new path in the space of this local tangent frame using its geometric derivatives and Newtonian iterations.
This mutation tries to preserve hard constraints, such as specular reflections, by utilizing the local knowledge about the geometry around the current path.
This way, manifold exploration can construct a connection from one point to another through a chain of specular or highly-glossy interactions.



ERPT is a variation of MLT. It utilizes the fact that independent samplers, such as BDPT, already provide a full distribution of light paths.
The idea is to try to redistribute the amount of energy carried by each initial path using perturbations.
In order to do that, multiple short Markov chains are started with the same seed path, the number of chains is computed adaptively based on the path energy.

Efficiency of ERPT heavily depends on how good is the seeding sampler. For example, BDPT *without* MIS provides very unbalanced sampling, leading ERPT to get stuck for a long time in some regions due to high redistribution workload.
Moreover, the redistribution region is manually set by the length of the Markov chains, making it non-trivial to tweak the parameters to achieve the best redistribution vs. stratification trade-off.

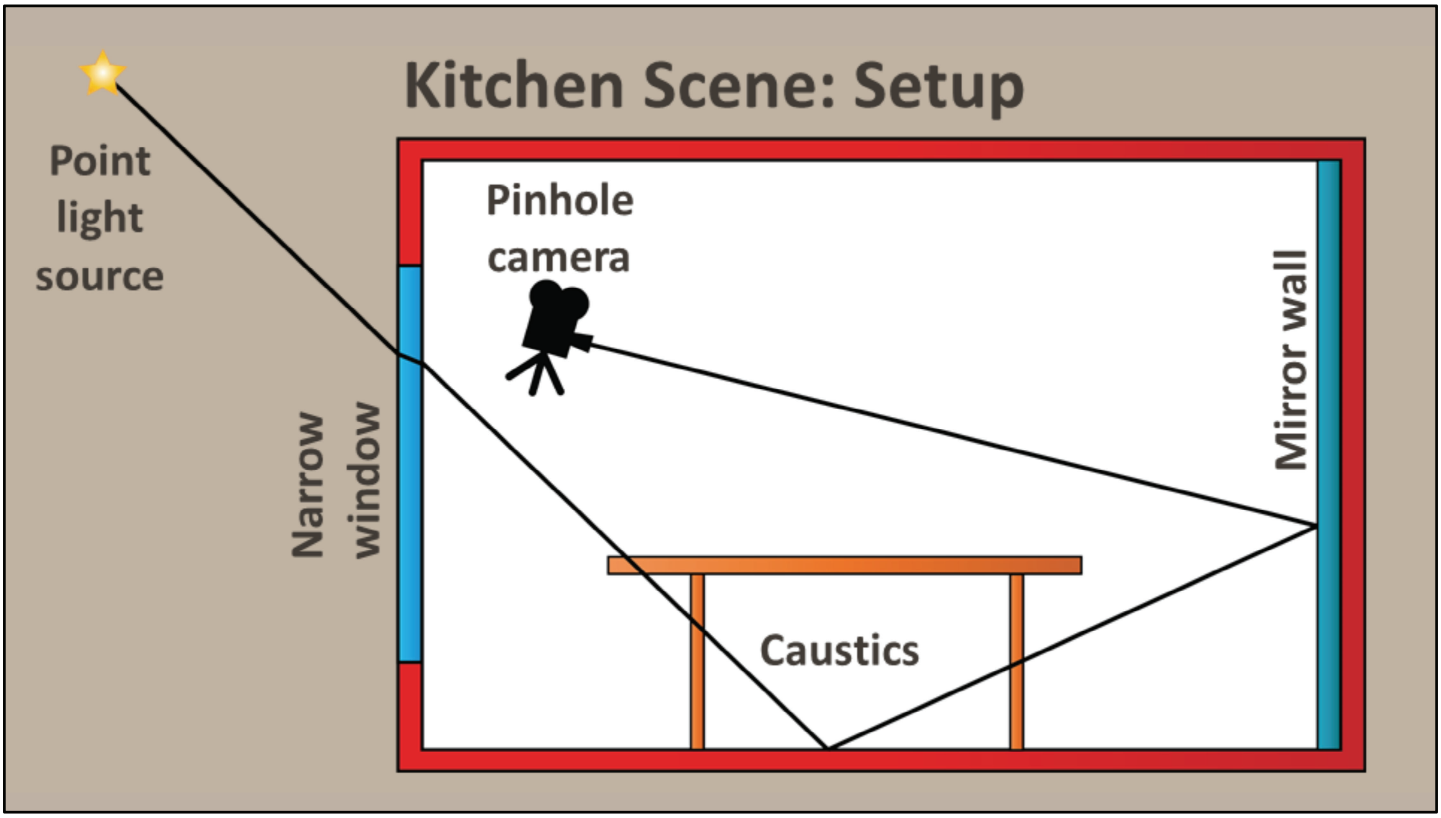


Population Monte Carlo framework was applied to light transport by Lai et al. in the context of ERPT.
The process keeps the constant population of chains by reseeding the chains with low contribution from a pool of stratified paths.
The core idea is to use a set of existing mutation strategies, where each strategy can be present multiple times with different user-defined parameters, like step size. The process emphasizes mutations with good performance, making the transition probability adapting to the data.

Equal Time Comparison

- Rendering time: 1 hour on GeForce 580
  - Some are rendered with Mitsuba [Jakob10]
- *Kitchen* – a hard-to-render scene
  - Light occluded by glass, caustics
  - Many mirrors, reflected caustics
  - Glossy materials
  - Curved glass with many specular interactions

Now I would like to provide a qualitative comparison of various existing methods.
Most of the renderings are done in equal time (1 hour) on the Nvidia GeForce 580 GTX GPU.
Some images are rendered on CPU with Mitsuba renderer with a performance equivalent of 1 hour on GPU.
I have set up a scene with multiple difficult lighting features in order to evaluate state-of-the-art light transport methods.

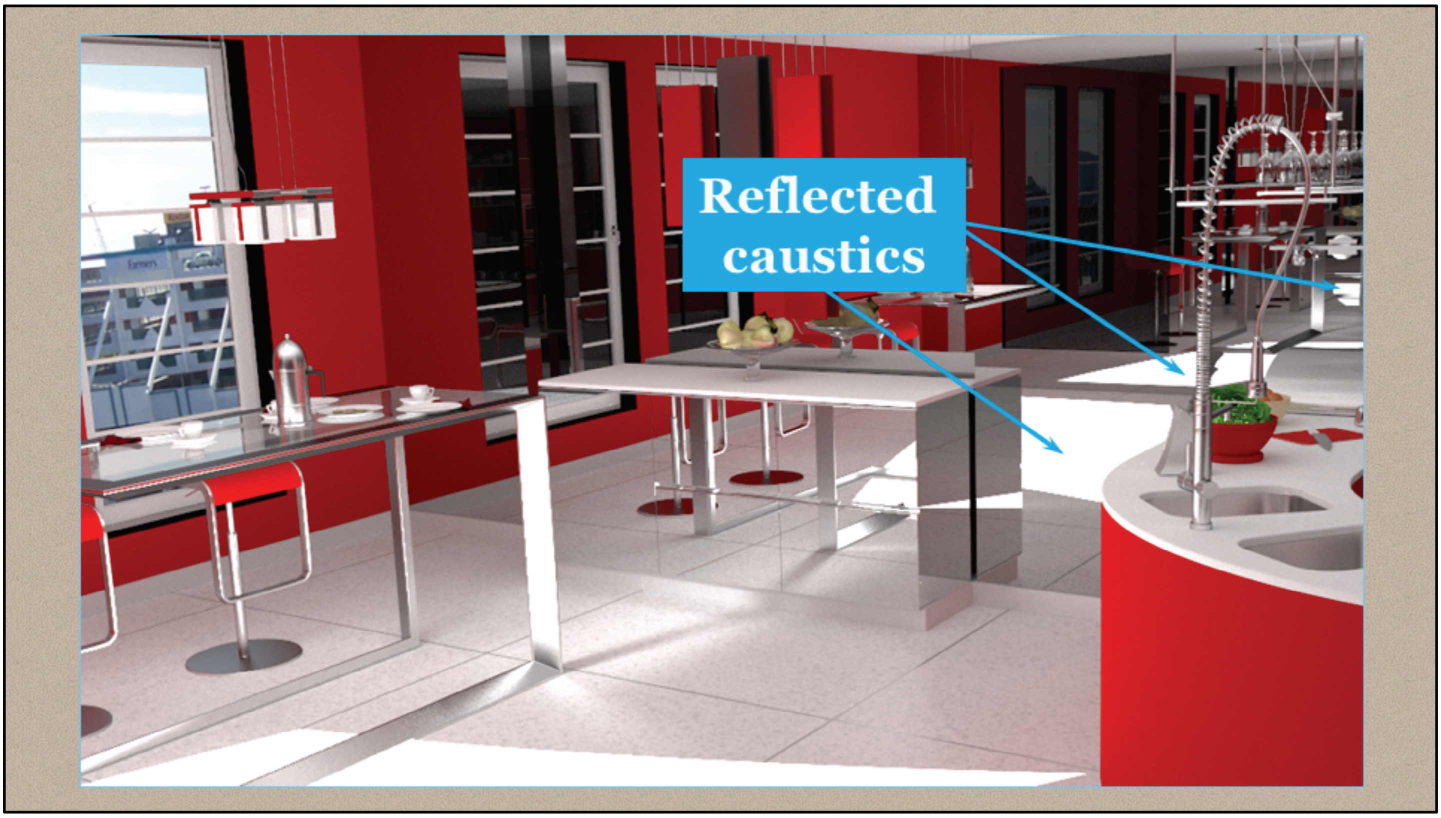


Here is the scene configuration.
Point light source is placed outside, illuminating the room through a glass window, causing a caustic on the floor.
The scene has also a mirror wall causing a lot of reflections.

This is the “Kitchen” scene rendered only with diffuse and glass materials to show the geometric complexity.

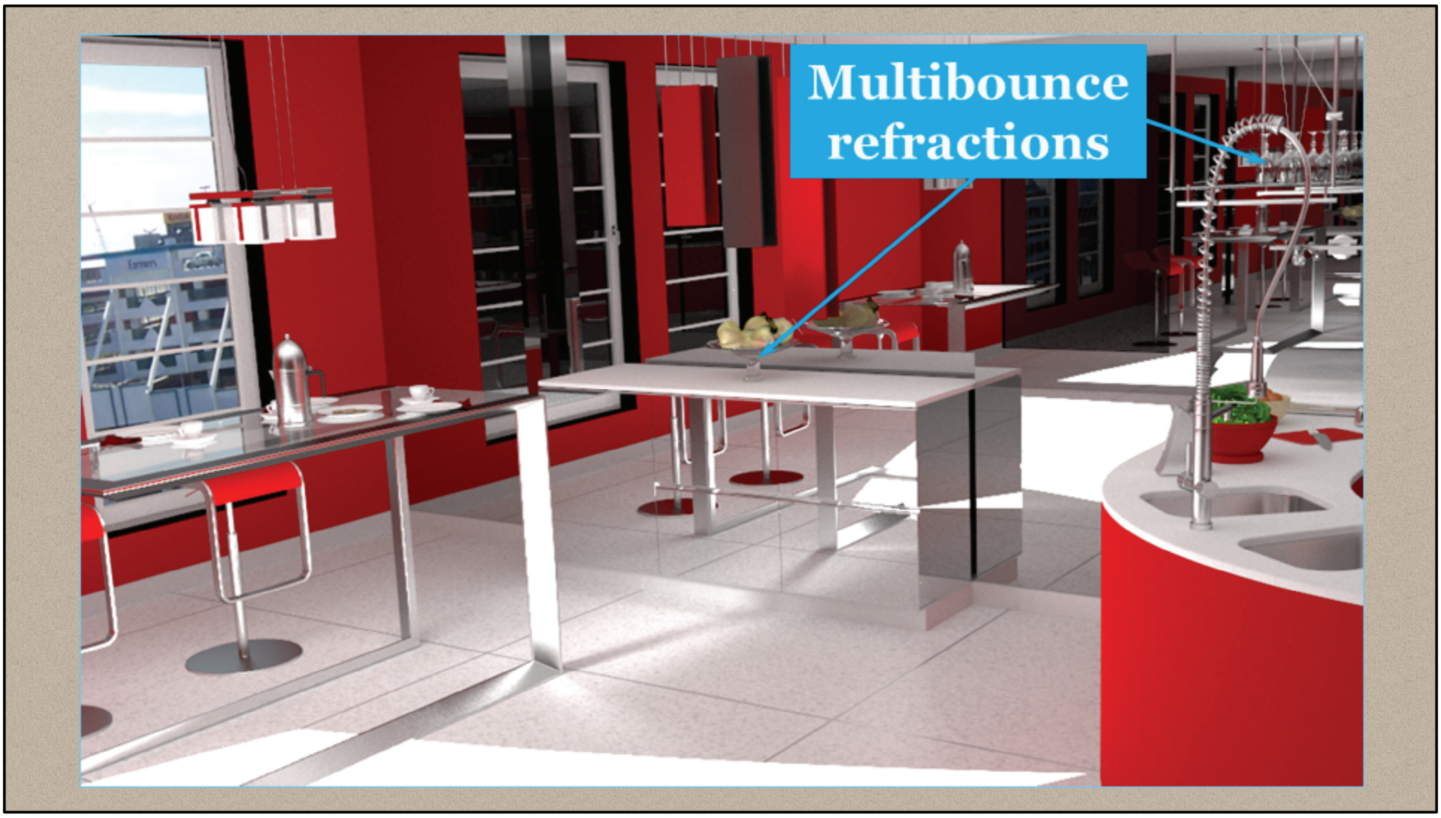
This is the reference image of the “Kitchen” scene with final materials. Rendered with Vertex Connection and Merging technique in 24 hours.

Here are some difficult features.
First of all, the direct lighting is highly occluded, since the window visible from the light source takes a small part of the emissive sphere of directions.
This situation is similar to the classical complex light transport scenarios, like an ajar door [Veach97].



Then, the reflected caustics caused by the point light source are impossible to find with all unbiased methods.

Also multi-bounce glossy caustics (window->table->floor) pose a difficult sampling problem for some Monte Carlo methods.



Refractions with multiple bounces (specular chains) has been also known to be a hard sampling problem [JakobMarschner12].

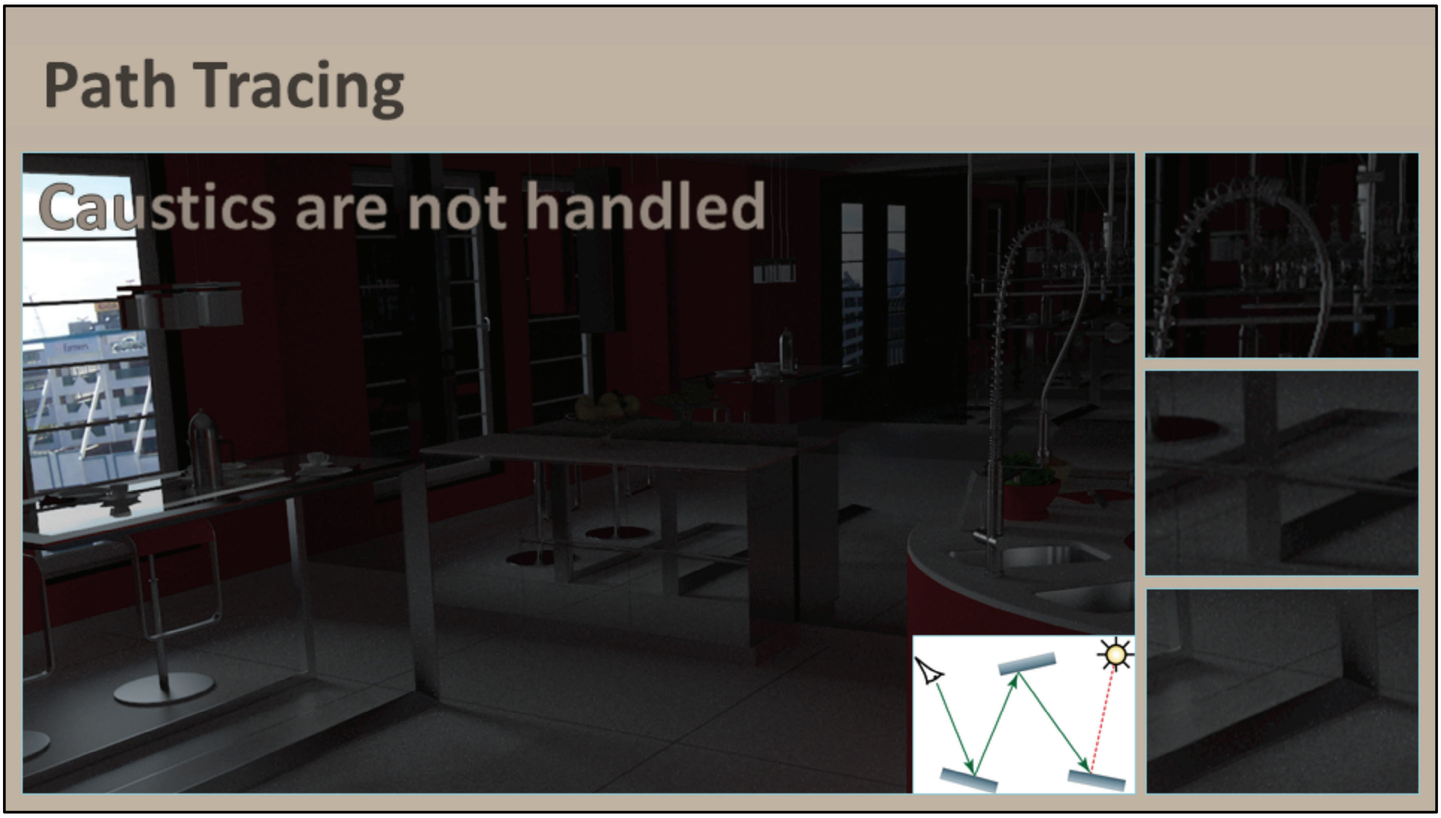
Finally, caustics from highly glossy objects are hard to find in path space (they are the rare events), thus might be difficult to discover even by advanced MCMC methods.

ORDINARY
MONTE CARLO METHODS

Ordinary Monte Carlo Methods

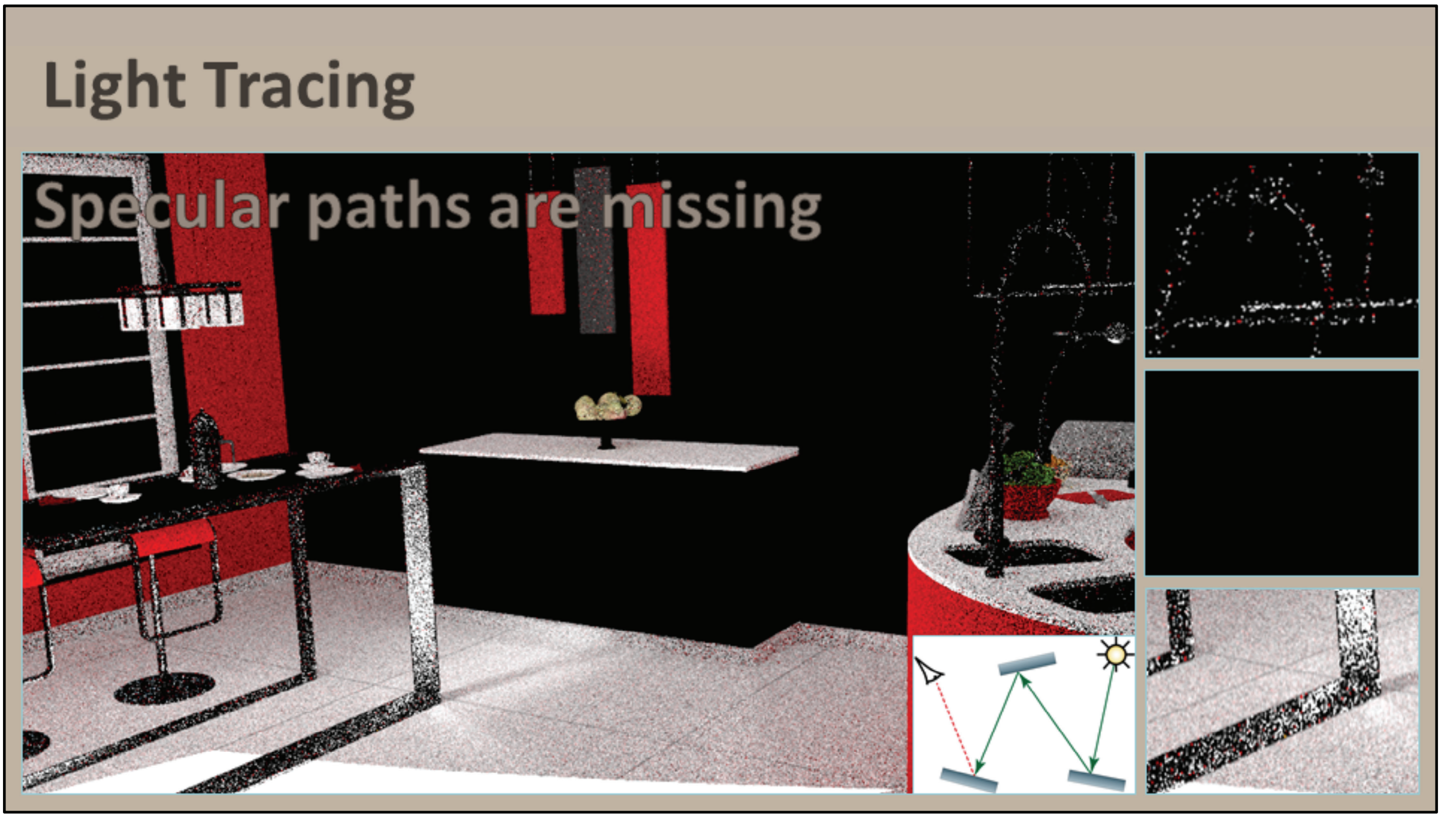
- Path tracing [Kajiya86]
- Light tracing [Arvo86, Dutre93]
- Bidirectional path tracing [LaFortune93, Veach94]
- Vertex connection and merging [Hachisuka12, Georgiev12]

We start with ordinary Monte Carlo methods. They can usually handle majority of scenes.
We will show the following Monte Carlo methods.

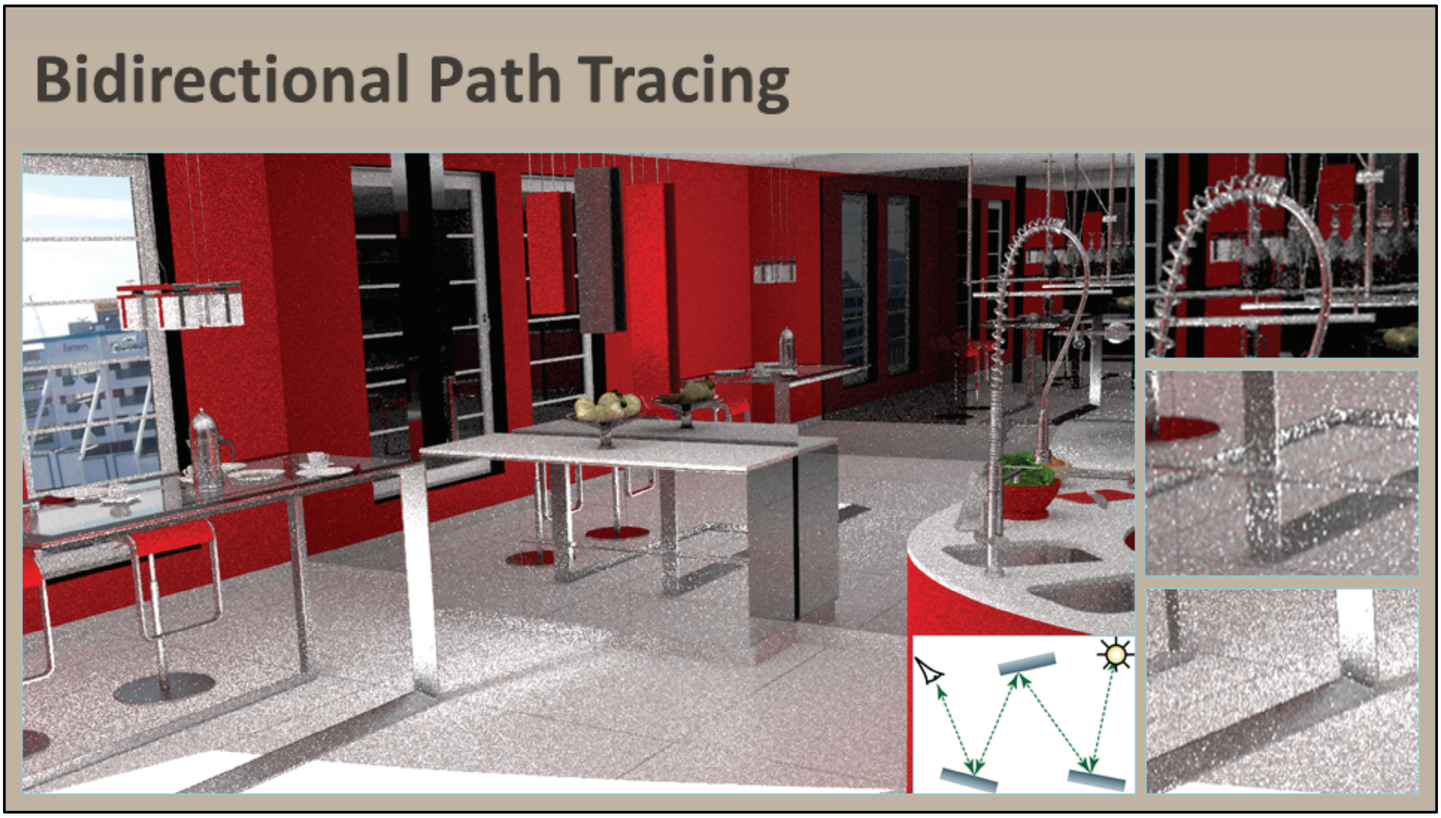


The first method: path tracing. It can handle only the small fraction of light coming from the environment.
This is due to the fact that all illumination coming from the point light source first comes through the glass. This makes it impossible to directly connect to the light source.

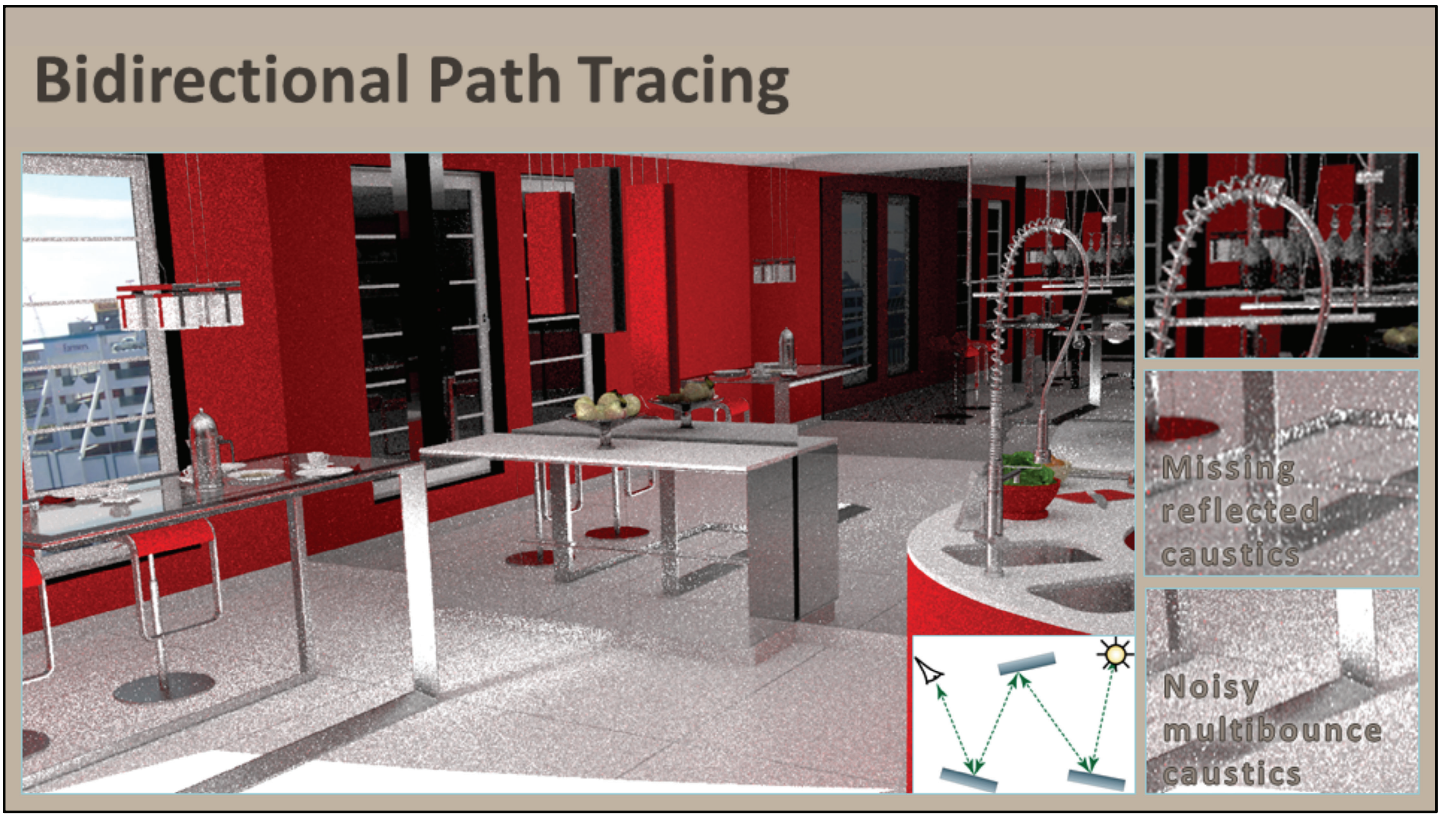
Note the close-ups on the right show difficult regions of the image. Also the small illustration on the bottom right shows how the current method constructs light paths.



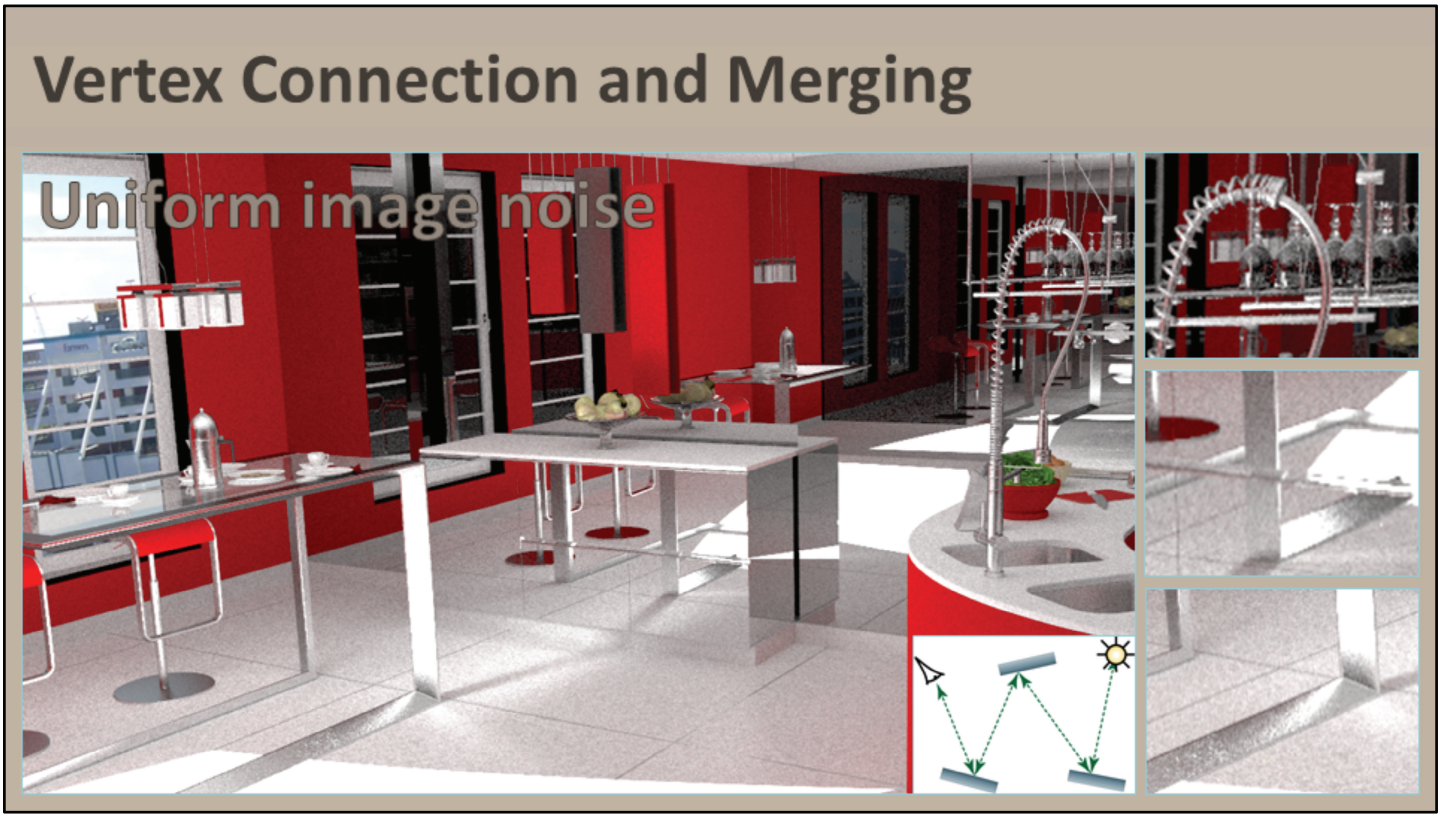
In opposite, the light tracing technique handles caustics coming through the window quite well, yet failing all specular reflections and refractions, because they cannot be constructed using connection to camera through specular surfaces.



Bidirectional path tracing handles almost all possible paths, however some difficult paths are noisy or missing.



BDPT has difficulties sampling caustics that go through multiple bounces. The reason is that the mass of such caustics in the path space is small, making it difficult to find them stochastically.
And the reflected caustics from point light cannot be sampled due to missing connection opportunity (that is, a path edge without singular materials).



Recent vertex connection and merging technique can handle caustics much better. Moreover it can even sample reflected caustics. However, the method has uniform image noise due to the highly occluded light source (which is being sampled stochastically).

Why Monte Carlo methods work well in most of the cases?
You have probably seen the recent femtosecond photography of real world light propagation.
(Video #0) I decided to do the same, but with the light transport simulation.
(Video #1) This video demonstrates that the light transport turns into a diffusion process very quickly.
(Video #2) The same happens when I propagate the virtual wavefront of “importance” from the camera. Note that it is not a physically valid value, yet it shows how the adjoint quantity is propagated, e.g., in bidirectional path tracing.
Both quantities spread around the scene very quickly, making it easy to establish the connection for unidirectional and bidirectional methods.
(Video #3) However let’s take a look at the glossy example: the diffusion is not that prominent anymore. This makes such configuration difficult to sample by ordinary Monte Carlo methods, because occasional sampling and stochastic connections of two subpaths lead mostly to low or zero contribution of the path on the image plane.

MARKOV CHAIN MONTE CARLO METHODS

However as we will see, some difficult illumination features and visibility situations can be handled more efficiently with Markov chain Monte Carlo methods.

Markov Chain Monte Carlo Methods

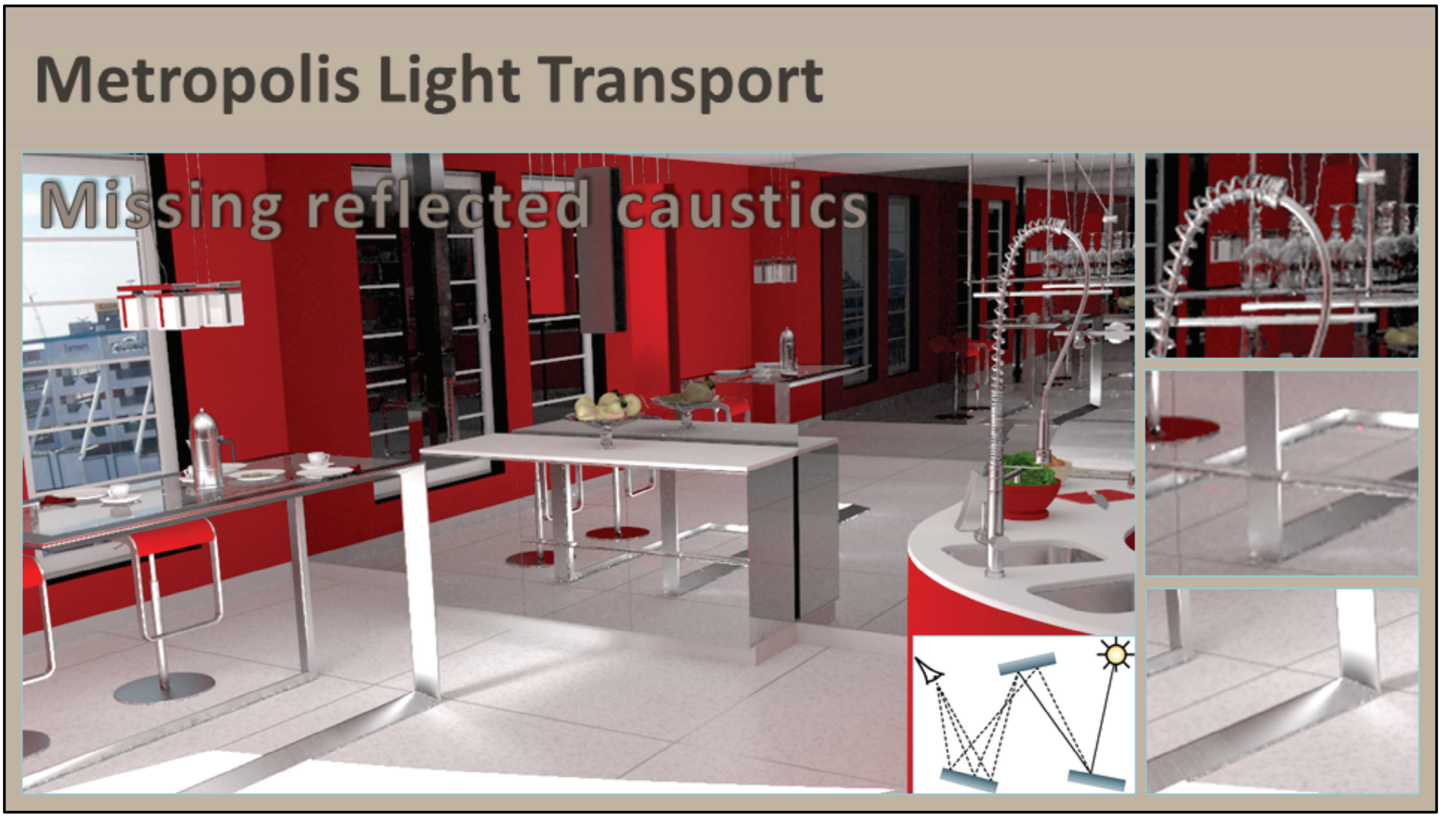
- Metropolis light transport [Veach97]
- Different mutations
  - Primary sample space mutation [Kelemen02]
  - Path space mutations [Veach97]
  - + Manifold exploration mutation [Jakob12]
- Energy redistribution path tracing [Cline05]

I'll present the following techniques. It is mostly various techniques in MLT framework.

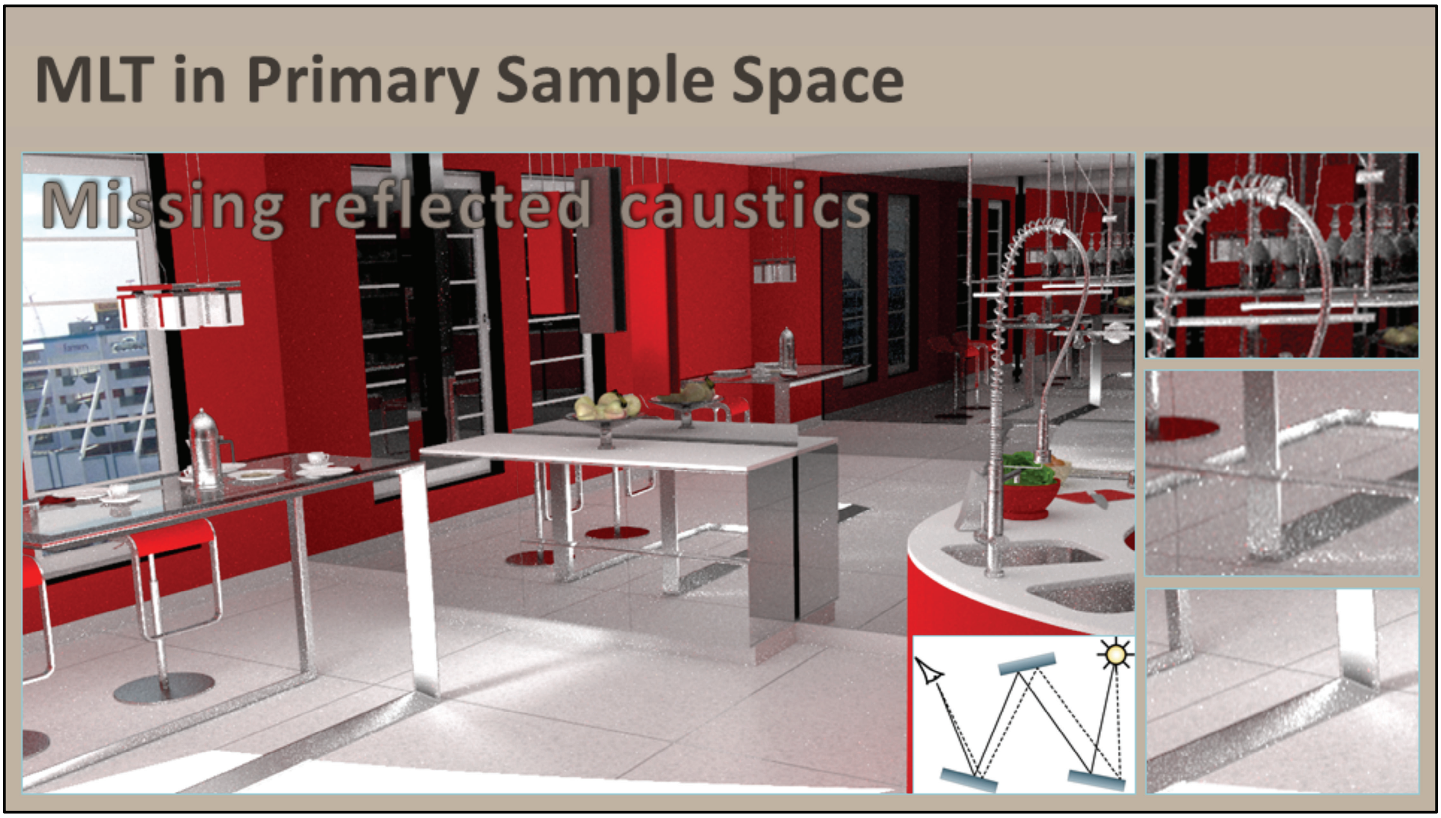
Markov Chain Monte Carlo Methods, part II

- PPM with MCMC photon tracing [Hachisuka11]
- Population Monte Carlo energy redistribution [Lai07]

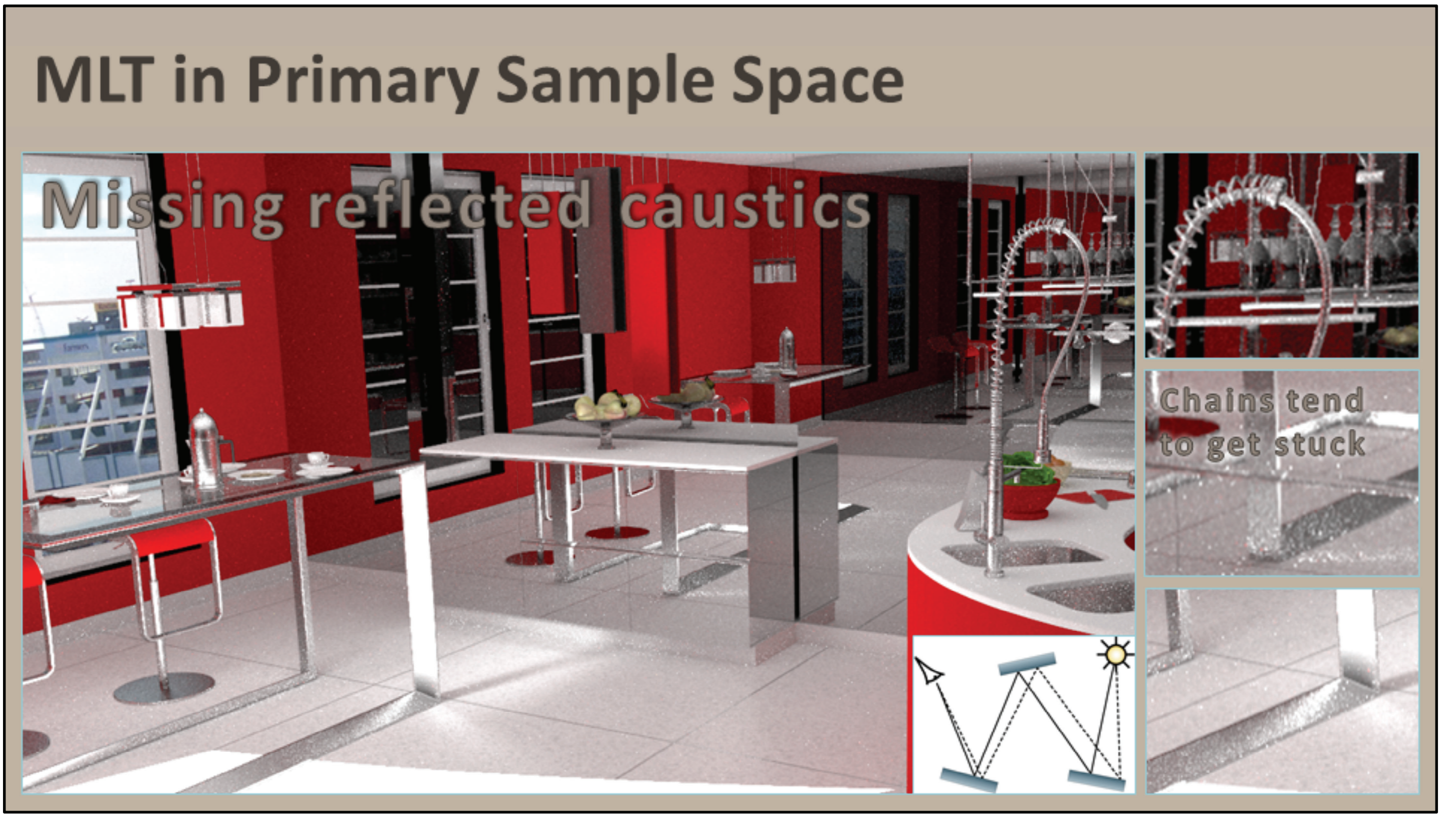
I will also show these recent and advanced MCMC methods.



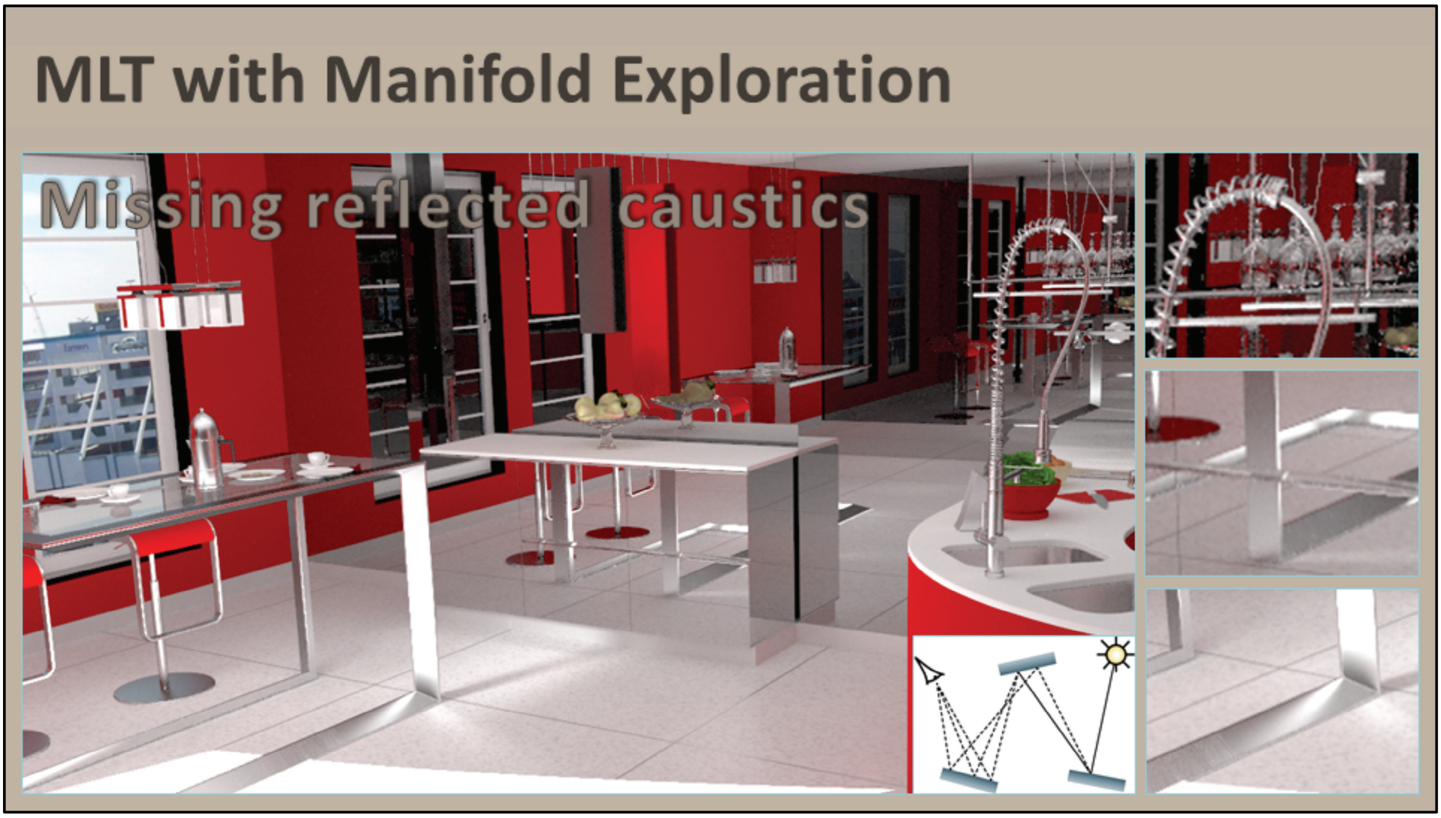
The first MCMC method (also chronologically) is the original Metropolis light transport algorithm [Veach97] with the original set of mutation strategies.
The image is handled well, without noticeable noise. Yet reflected caustics are missing, as expected from an unbiased method.
Note some under-sampling at geometric boundaries, especially around glossy surfaces (for example, the farther horizontal table leg on the floor has some bright splotches).
This is due to the fact that the glossy transport poses narrow constraints on the path, which are hard to satisfy with the original set of mutations.



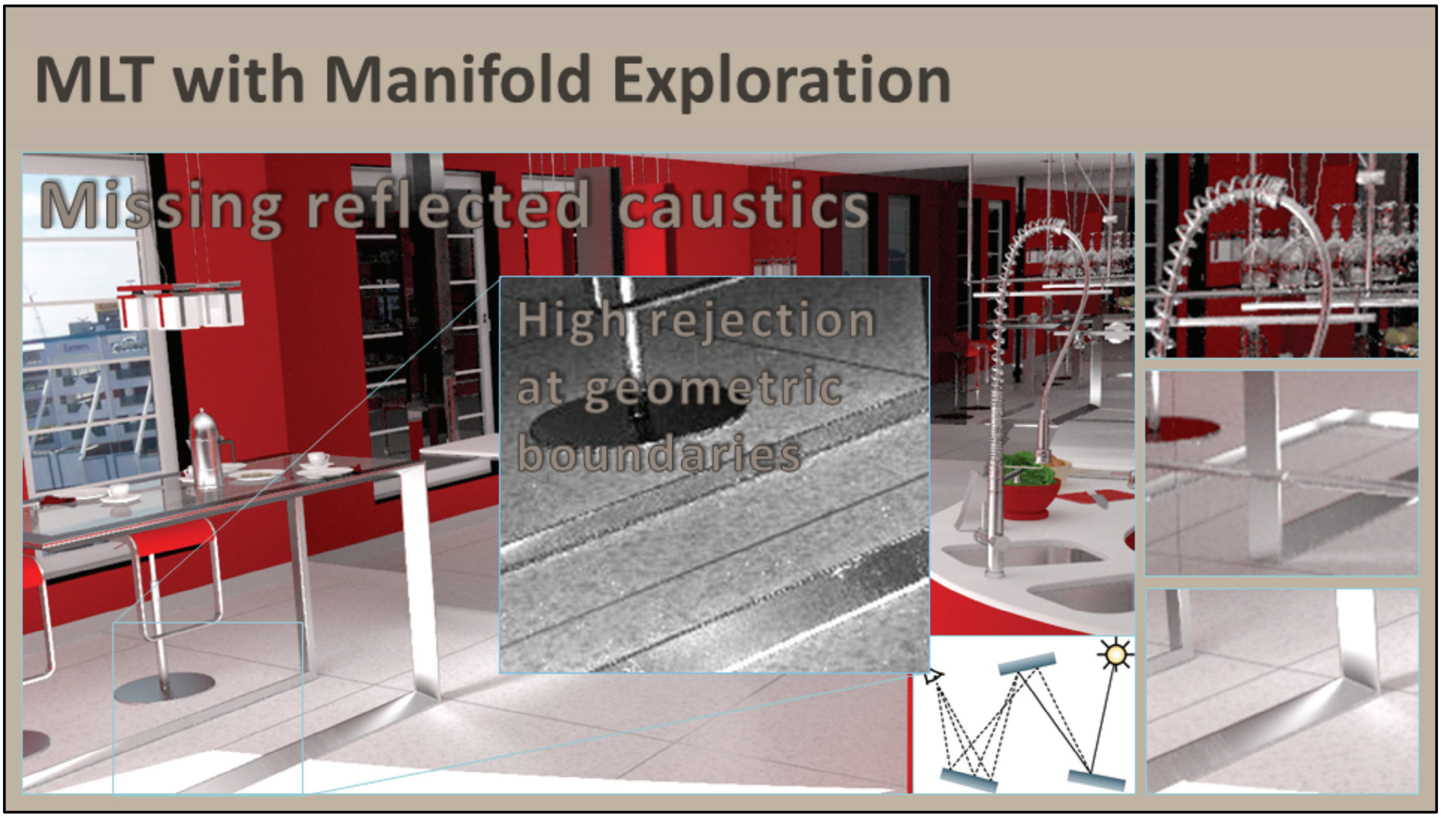
Here is Metropolis light transport with mutations in primary sample space [Kelemen02].



Note that some chains get stuck in complex caustics paths.
This happens because of many rejections since the whole path is usually perturbed, thus constantly jumping from its valid form.
Also the small step size is specified globally for random numbers and is not adapted for long or highly-occluded paths.

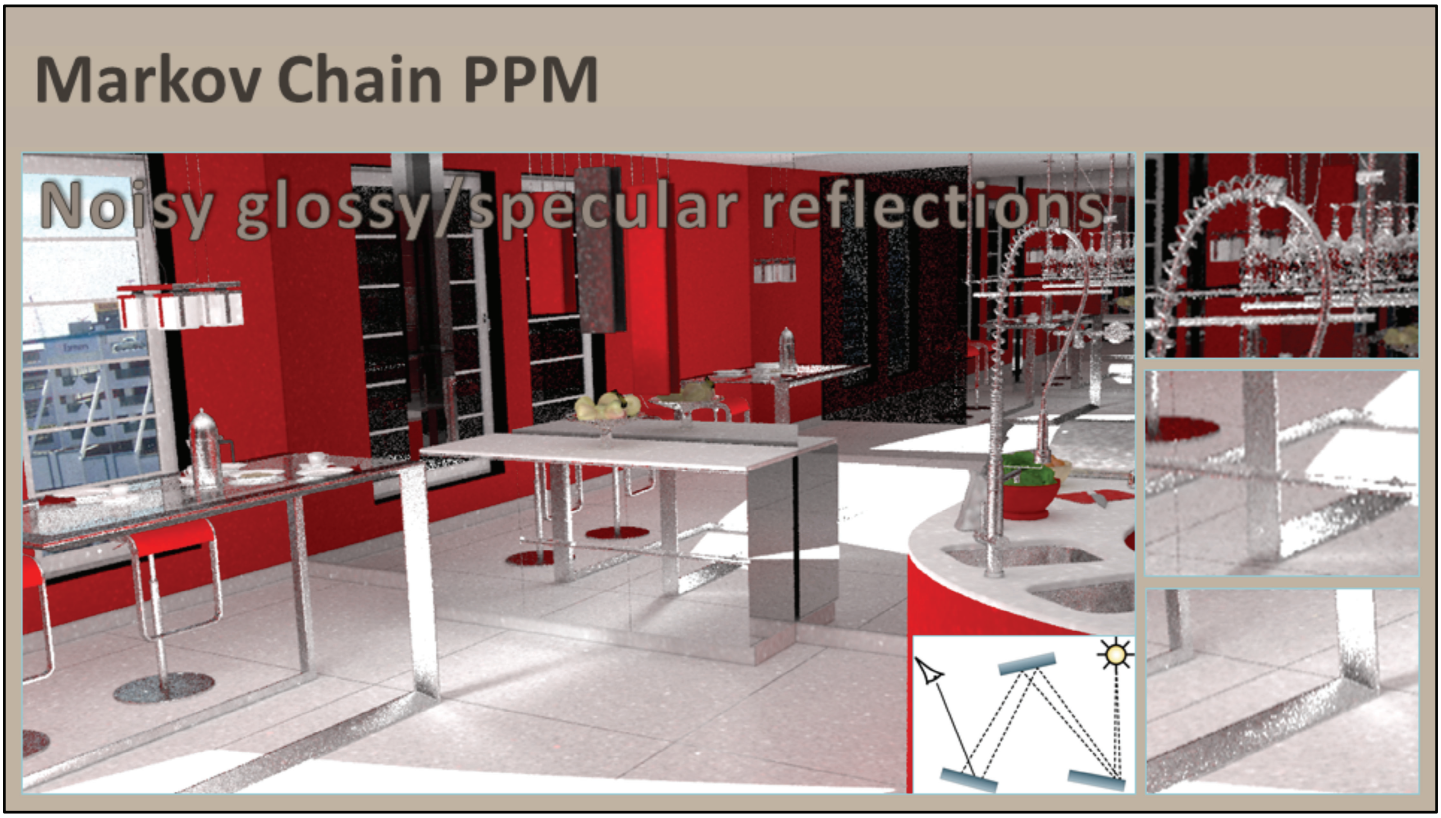


Here is the original MLT algorithm with the recently introduced manifold exploration mutation [Jakob12].
Note how much better the multibounce refractions are handled.
Also note that the reflection of the glossy caustics from the table leg are handled much better than with the original MLT.
This is due to the fact that this mutation can connect through specular interactions, while satisfying specular constraints.



Also note that some subtle under-sampling is present along geometric edges, because of high rejection rate as the mutation still tries to perform a connection in between the fixed chains, jumping off the glossy highlights.

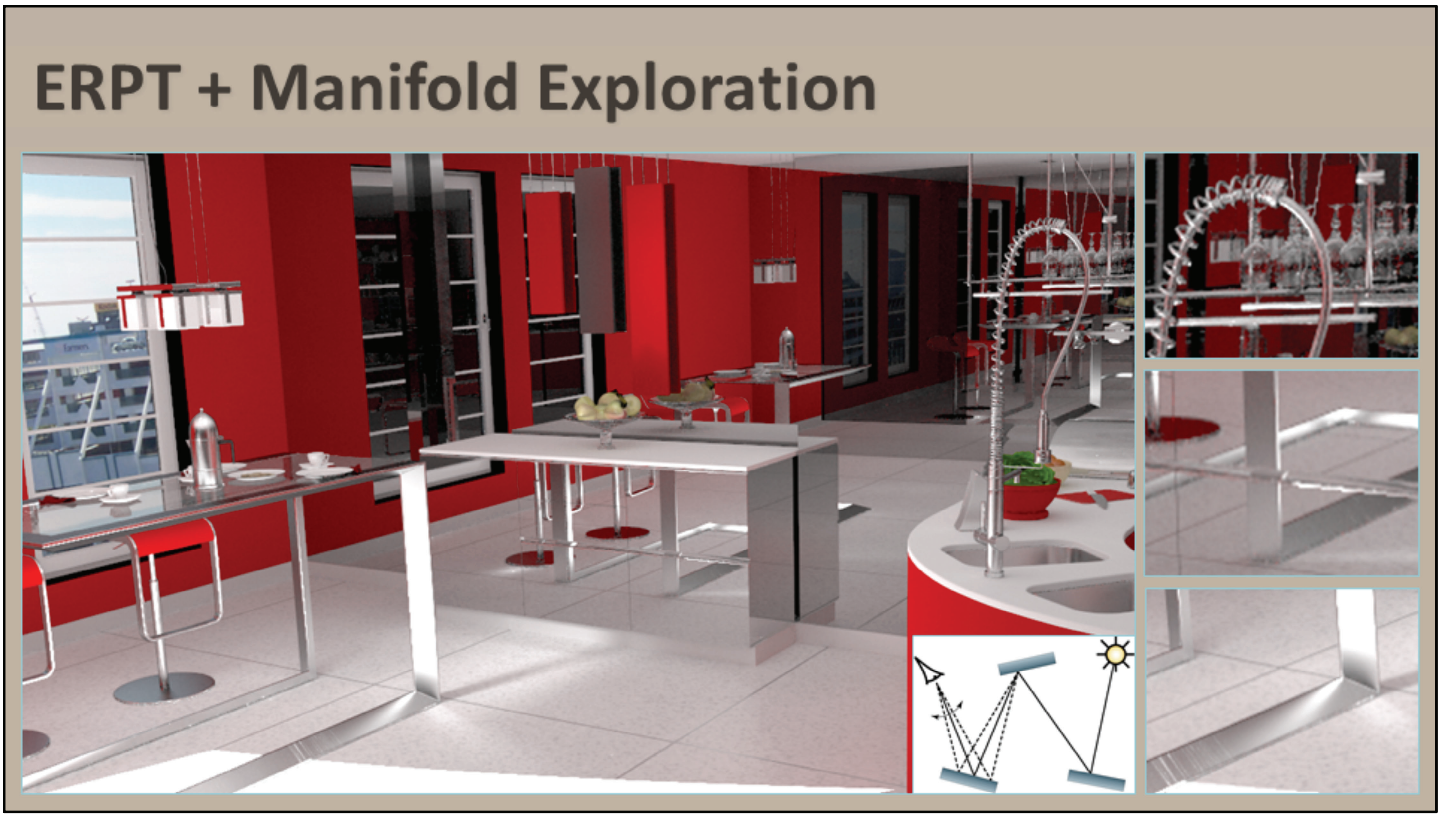
We will present a new mutation strategy, which improves on that issue, tomorrow.



Here is MCPPM, another recent method by Toshiya Hachisuka.
It is an extension to stochastic progressive photon mapping [Hachisuka08] with Markov chain photon shooting [Hachisuka11].
The original Kelemen mutation strategy with recommended parameters was used to mutate photon paths.
In order to keep the target distribution constant, the camera subpaths should be updated rarely.
This leads to under-sampling at directly visible glossy and refractive surfaces.

Energy Redistribution Path Tracing

Here is the original energy redistribution path tracing method [Cline05] with the original set of mutations [Veach97], seeded with BDPT.
We used one chain per pixel (on average) and 100 mutations per chain (being the original values recommended by the author).
As we can see, ERPT has difficulties efficiently redistributing energy from glossy paths, especially if they also experience one or more perfect reflections/refractions. This is the “insufficient techniques” problem stemming from the original set of mutations, as discussed before.
Moreover, some difficult parts of the image are under-sampled in the same rendering time comparing to MLT. This happens because Markov chains are re-seeded with BDPT much more often, yet BDPT doesn’t provide a good quality of seeding, as we have seen from the BDPT rendering (due to the highly occluded light source and complex light transport).



Another combination is ERPT with the original set of mutation, enriched by the new manifold exploration.
As we can see, manifold exploration allows to redistribute the energy of complex specular and glossy paths much better, leaving just a few splotches on the image.

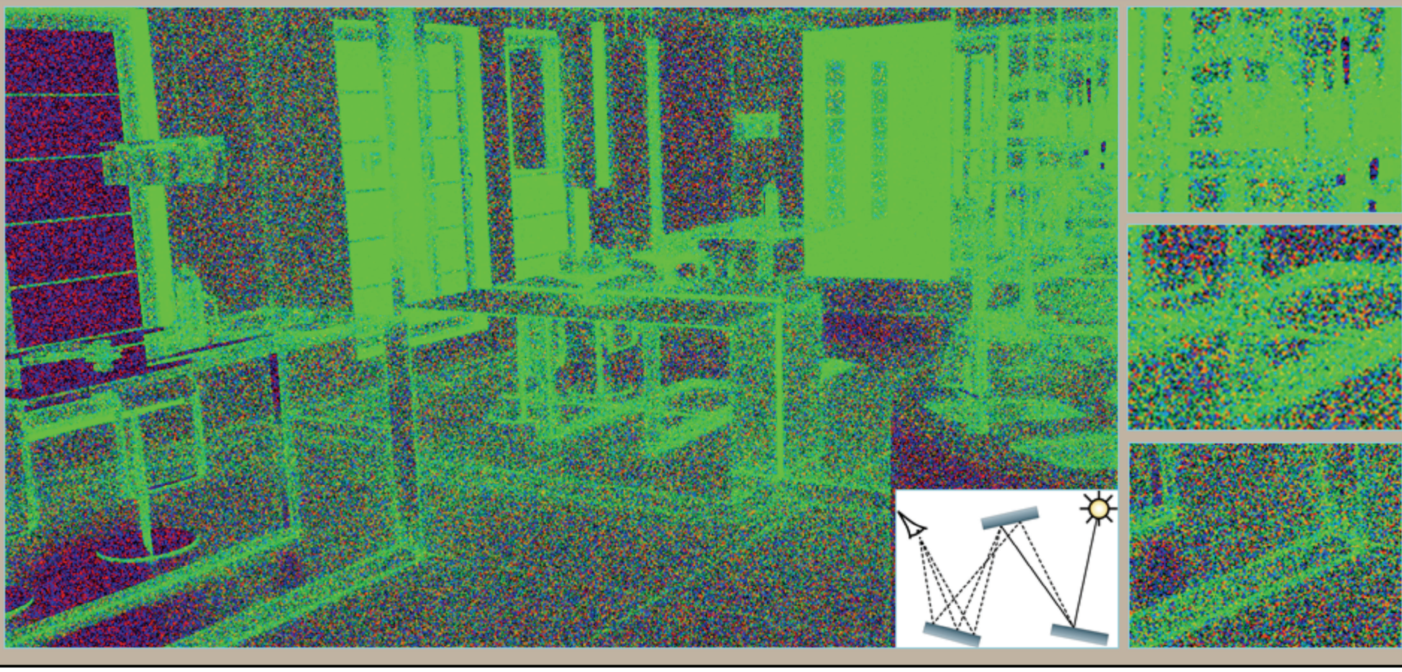
ERPT, Manifold Exploration Only

Another interesting experiment is to provide *only* manifold exploration as an available redistribution mutation, while relying on BDPT for the stratification and search of new features.
As we can see it also works relatively well, leading to the conclusion that this single mutation strategy is enough for achieving good results with ERPT.
However the equilibrium is not achieved with any of ERPT variations in a given time budget of 1 hour on this scene.

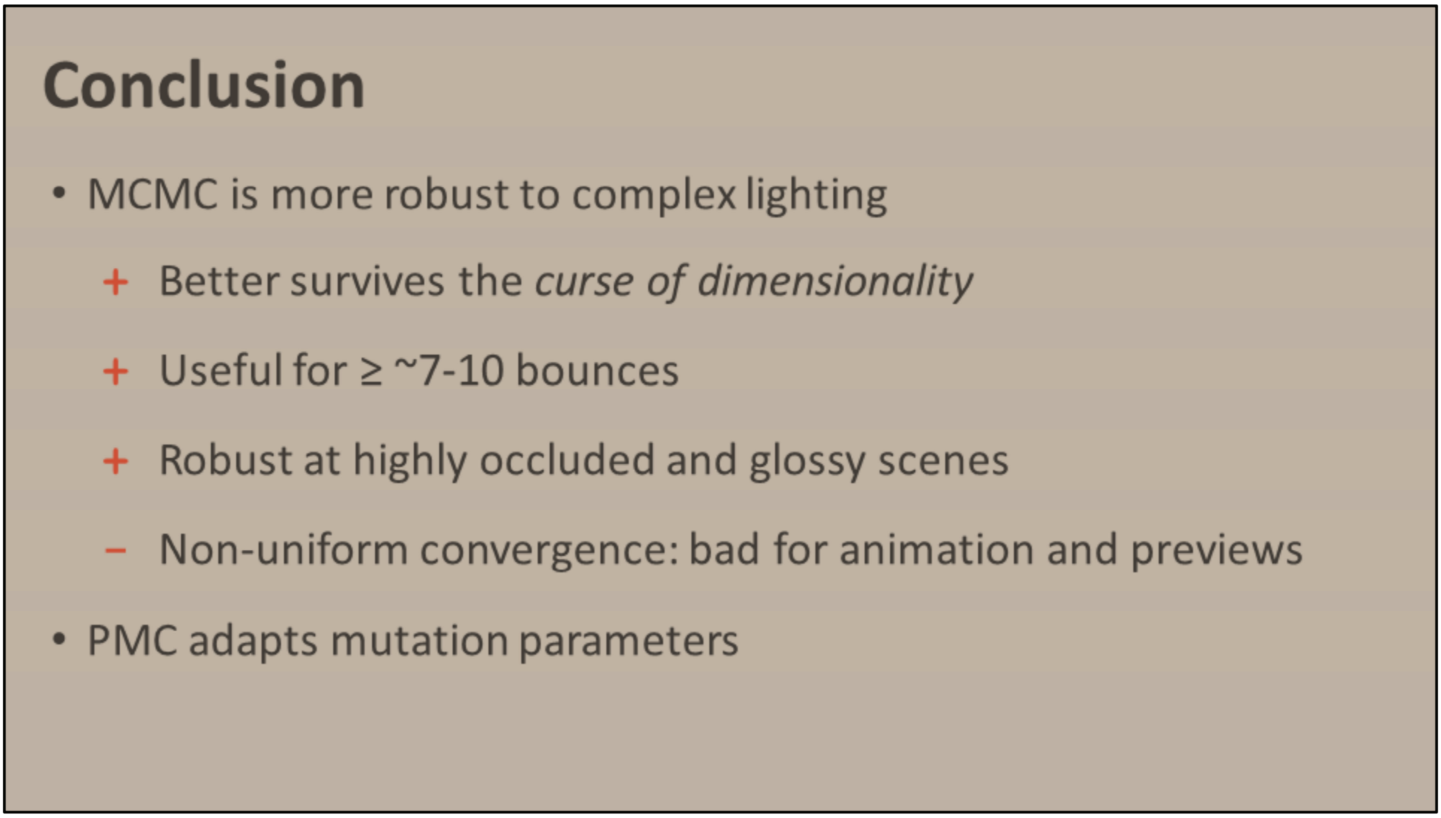
Population Monte Carlo ER + ME

In this rendering we use the Population Monte Carlo framework applied on top of ERPT [Lai et al. 2007] with multiple manifold exploration mutations.
We use a population of 1024 chains and various values of step parameter *lambda* ranging from 20 to 200 for manifold exploration mutation.
Note, that edging of the glossy table is sampled much better, which is even more noticeable in the reflection of the table legs.
This is due to the fact that the population adapts and selects the mutation strategy with smaller perturbation step.

Population Monte Carlo ER + ME



This is a false-colored overlay of weights for different selected strategies.
Greener color corresponds to the preference of mutations with smaller step sizes.
These weights are reset every time the chain is re-seeded from BDPT, thus the weights are noisy.
Note how the method prefers mutations with smaller step along geometric discontinuities.



Modern MCMC method survive the curse of dimensionality better than ordinary MC methods (like PT and BDPT).
So, whenever the majority of light transport in the image is high-dimensional (that is, roughly more than 7-10 bounces), MCMC usually behaves better in such scenarios.
It also helps exploring narrow islands and peaks in path space caused by highly occluded lighting configuration and/or highly glossy materials.
However, MCMC is famous for its non-uniform convergence, that is, it might take a while before Markov chain finds some important feature on the image, like some distant reflected caustic. E.g., a feature can suddenly appear on the image rendered for a long time.
Additionally, population Monte Carlo framework provides an easy and unbiased adoption of parameters for mutation strategies and also provides slightly better redistribution.

Thank You for Your attention.

QUESTIONS?