Interactive Light Transport with Virtual Point Lights

Benjamin Segovia\textsuperscript{1,2}

\textsuperscript{1}ENTPE: Ecole Nationale des Travaux Publics de l’Etat
\textsuperscript{2}LIRIS: Laboratoire d’InfoRmatique en Image et Systèmes d’information
1. Introduction

2. Formalizing the Problem

3. Sampling VPLs: Metropolis Instant Radiosity

4. Accumulating VPL contributions

5. Coherent Metropolis Light Transport

6. Conclusion
Summary

1. Introduction
2. Formalizing the Problem
3. Sampling VPLs: Metropolis Instant Radiosity
4. Accumulating VPL contributions
5. Coherent Metropolis Light Transport
6. Conclusion
Why a Ph.D. in computer graphics?

Movie / FX industry

- Fast and robust rendering algorithms;
- Not necessary real-time but speed is fundamental.

**Figure:** Poseidon, 2006, rendered with Mental Ray
Why a Ph.D. in computer graphics?

**Figure:** Thee Dragon Room, rendered with yaCORT

**Lighting design**
- Physically-based rendering tools;
- Not necessary real-time.
Why a Ph.D. in computer graphics?

Video Games

- The most realistic rendering with strict constraints;
- Real time (more than 30 frames per second).

Figure: A Quake 3 scene, rendered with Qrender
What Does this Ph.D. Contain?

Common approach in science

1. Identify the physical problem
   → Simulating light transport;

2. Propose an appropriate mathematical formalism
   → The related physical quantities and the light transport equations;

3. Design algorithms to solve these equations
   → Computer science \{ Numerical schemes, Algorithms, codes… \}

The contribution of this Ph.D. thesis is mostly contained by the third point
→ Numerical schemes to solve the light transport equations
First, introduction of necessary concepts

- **Physics**: Physics of light transport → quantities and equations;
- **Mathematics**: Roots of Monte-Carlo and introduction of the appropriate formalism;
- **Computer Graphics**: Most common algorithms used to compute virtual pictures.
Overview of the Presentation

Then, presentation of the contributions

Two classes of contributions:

- **Coding Techniques**: Once the set of VPLs has been computed, how can we accumulate their contributions? → we present two techniques using graphics hardware;

- **Sampling Techniques**: How can we generate efficient sets of VPLs?
Summary

1. Introduction

2. Formalizing the Problem

3. Sampling VPLs: Metropolis Instant Radiosity

4. Accumulating VPL contributions

5. Coherent Metropolis Light Transport

6. Conclusion
The Rendering Equation - Three Point Form [Vea97]

Formalizes the behavior of materials and surfaces

\[ L(x' \rightarrow x'') = L_e(x' \rightarrow x'') + \int_M L(x \rightarrow x') f_s(x \rightarrow x' \rightarrow x'') G(x \leftrightarrow x') dA(x) \]

where:
- \( L \) is the equilibrium outgoing radiance function;
- \( L_e(x' \rightarrow x'') \) is the emitted radiance leaving \( x' \) in the direction of \( x'' \);
- \( f_s(x \rightarrow x' \rightarrow x'') \) is the BSDF of the material;
- \( M \) is the union of all the surfaces of the scene;
- \( A \) is the Lebesgue (i.e. uniform) area measure on \( M \);
- \( G(x \leftrightarrow x') \) is the geometric term between \( x \) and \( x' \).
The Measurement Equation

Response of a given captor / sensor

\[ I_j = \int_{\mathcal{M} \times \mathcal{M}} W_e^{(j)}(x \rightarrow x') L(x \rightarrow x') G(x \leftrightarrow x') \, dA(x) \, dA(x') \]

where \( W^{(j)}(x, \omega) \) is the responsivity of sensor \( j \).
Issues with Light Transport

- High dimensional problem: light may bounce many times …
- High frequency problem: many discontinuities (shadows for example).
Issues with Light Transport

- High dimensional problem: light may bounce many times …
- High frequency problem: many discontinuities (shadows for example).

Integrand has very poor properties →
Issues with Light Transport

- High dimensional problem: light may bounce many times …
- High frequency problem: many discontinuities (shadows for example).

Integrand has very poor properties →

Use Monte-Carlo integration!
Monte-Carlo Integration is Basically . . .

We want to integrate

\[ I = \int_{\Omega} f(x) d\mu(x) \]

where

- \( \Omega \) is a given space;
- \( \mu \) is an associated measure;
- \( f \) is a measurable function on \((\Omega, \mu)\).

With sufficient properties . . .

\[ N \text{ random variables } (X_n)_{n \in \{1\ldots N\}} \text{ with density } p, \text{ then:} \]

\[
\lim_{N \to \infty} I_N = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \frac{f(X_n)}{p(X_n)} = I \text{ almost surely}
\]
The Path Integral Formulation

Make the light transport problem an integration one

Inject the rendering eq. into the measurement eq. and expand it:

$$I_j = \sum_{k=1}^{\infty} \int_{M^{k+1}} \left[ L_e(x_k \rightarrow x_{k-1}) G(x_0 \leftrightarrow x_1) W^{(j)}_e(x_1 \rightarrow x_0) \right.$$  

$$\left( \prod_{i=1}^{k-1} f_s(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}) G(x_i \leftrightarrow x_{i+1}) \right) dA(x_0) \ldots dA(x_k) \]$$
The Path Integral Formulation

Make the light transport problem an integration one

Inject the rendering eq. into the measurement eq. and expand it:

\[ I_j = \sum_{k=1}^{\infty} \int_{M^{k+1}} \left[ L_e(x_k \rightarrow x_{k-1}) G(x_0 \leftrightarrow x_1) W_e^{(j)}(x_1 \rightarrow x_0) \right. \\
\left. (\prod_{i=1}^{k-1} f_s(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}) G(x_i \leftrightarrow x_{i+1})) dA(x_0) \ldots dA(x_k) \right] \]

The path integral formulation

\[ I_j = \int_{\Omega} f^{(j)}(\vec{x}) d\mu(\vec{x}) \]

\( \Omega \) is the set of all finite length paths, \( \mu \) its natural measure and \( f^{(j)} \) obtained with the expansion.
OK, a short summary!

Monte-Carlo rendering is:

- Sample a path \( \bar{x} \) with density \( p(\bar{x}) \);
- Evaluate \( \frac{f^{(j)}(\bar{x})}{p(\bar{x})} \);
- Accumulate.
A Short Pause Before the Remainder!

OK, a short summary!

Monte-Carlo rendering is:

- Sample a path $\vec{x}$ with density $p(\vec{x})$;
- Evaluate $\frac{f^{(j)}(\vec{x})}{p(\vec{x})}$;
- Accumulate.

Most Monte-Carlo rendering methods → propose new ways to generate paths $\vec{x}$.

Basically, this Ph.D. presents new Monte-Carlo rendering techniques.
Short Overview of Path Integration

Core algorithm: path tracing [Kaj86]

We generate a light path backward from the camera for each camera sensor (i.e. for each pixel)
Core algorithm: path tracing [Kaj86]

We generate a light path backward from the camera for each camera sensor (i.e. for each pixel)

Many, many similar techniques

- Bidirectional path tracing [VG94, LW93];
- Light tracing [DLW93].
Path Tracing
Path Tracing
Path Tracing
Problems with these "Pure" Path Tracing methods

- No computation coherency
  - Per-pixel computations are independent;
  - No factorization.
Problems with these "Pure" Path Tracing methods

No computation coherency
- Per-pixel computations are independent;
- No factorization.

We must design *efficient* techniques
- Most of them propose to use *biased* estimators:
  - Photon Maps [Jen01, Jen96, Jen97];
  - Radiance / Irradiance Caches [WRC88, War94, K05];
  - And ...
Problems with these "Pure" Path Tracing methods

- No computation coherency
  - Per-pixel computations are independent;
  - No factorization.

Instant Radiosity [Kel97] → Replaces complete paths by "Virtual Point Lights"
Instant Radiosity

Principles

Splits each path \( \overline{x} = \{x_0, x_1, \ldots, x_n\} \) into three parts:

- \( \overline{x}_c = \{x_0, x_1\} \) is the camera sub-path;
- \( x_v \) is a geometric Virtual Point Light (VPL);
- \( \overline{x}_s \) is the remainder of the path connected to a light source.
Instant Radiosity

Principles

- For all sensors (i.e., pixels), use the same \((x_v, x_s)\) light paths;
- Two-pass algorithm:
  - Propagation of light paths from the light sources (sampling);
  - Accumulation of VPL contributions (gathering).

Do not forget: a VPL is a light path, not only a point!
The Two Steps of Instant Radiosity

Particle Propagation
The Two Steps of Instant Radiosity

Particle Propagation
The Two Steps of Instant Radiosity

Particle Propagation

- Particle Propagation
- Accumulating VPL contributions
- Coherent Metropolis Light Transport
The Two Steps of Instant Radiosity

Particle Propagation
The Two Steps of Instant Radiosity

Particle Propagation
The Two Steps of Instant Radiosity

Particle Propagation
The Two Steps of Instant Radiosity

Incoming Radiance Field Integration

- Interactive Light Transport with Virtual Point Lights
The Two Steps of Instant Radiosity

Incoming Radiance Field Integration
The Two Steps of Instant Radiosity

Incoming Radiance Field Integration
The Two Steps of Instant Radiosity

Incoming Radiance Field Integration
The Two Steps of Instant Radiosity

Incoming Radiance Field Integration
Advantages and Drawbacks of Instant Radiosity

Advantages

- **Simple** → the incoming radiance field is replaced by a set of points;
- **Fast** → can be easily implemented with coherent ray tracing or rasterization.
Advantages and Drawbacks of Instant Radiosity

Advantages

- Simple → the incoming radiance field is replaced by a set of points;
- Fast → can be easily implemented with coherent ray tracing or rasterization.

Drawbacks

- Variance problems → how must the VPLs be located?
- Does not handle all lighting phenomena → caustics …
Summary

1. Introduction
2. Formalizing the Problem
3. Sampling VPLs: Metropolis Instant Radiosity
4. Accumulating VPL contributions
5. Coherent Metropolis Light Transport
6. Conclusion
Goal of Metropolis Instant Radiosity (MIR)

Properties of VPLs is *fundamental*

We must find VPLs which illuminate parts of the scene seen by the camera!
Goal of Metropolis Instant Radiosity (MIR)

Solution: Combine the robustness of Metropolis Light Transport and the efficiency of Instant Radiosity

Principle of MIR

- Use the path sequence of Metropolis Light Transport to sample VPLs ("MLT part");
- For each path, store the second point as a VPL;
- Accumulate all VPL contributions ("IR part").

With this sampler, all VPLs will bring the same amount of power to the camera.
Principle (a short version)

- Consider the whole camera integrand \( f^{(c)} \);
- Sample \( N \) paths with a density proportional to \( f^{(c)} \);
- Count for each pixel \( j \), the number \( N_j \) of paths which get into it;
- With \( N_j \), \( N \), and \( \int_{\Omega} f^{(c)}(\overline{x}) d\mu(\overline{x}) \), compute the per-pixel histogram of \( f^{(c)} \);
- We have the intensity of each pixel!
Numerical schemes behind it

- Compute \( \int_{\Omega} f^{(c)}(\mathbf{x}) d\mu(\mathbf{x}) \)
  \(\rightarrow\) Use a standard bidirectional path tracer;
- Sample \( N \) paths with a density proportional to \( f^{(c)} \)
  \(\rightarrow\) Use a Metropolis-Hastings algorithm.
Metropolis Light Transport [VG97]

**Metropolis-Hastings**

- Goal: given function $f$, sequentially sample random variables $X_i$ with a density proportional to $f$;
- $X_{i+1}$ and $X_i$ are correlated by a mutation.
- The density of $X_i$ is not exactly $f$, but with good properties ("ergodicity"), we can use all $X_i$:
  - as if their densities were $f$
  - as if they were independent
MLT - Initial Path
Interactive Light Transport with Virtual Point Lights
MLT - Candidate accepted → Count its contribution
MIR - Compute the power received by the camera

With a bidirectional path tracer (or any other technique) compute the power

\[ P_c = \int_{\Omega} f^{(c)}(\vec{x}) d\mu(\vec{x}) \]

received by the camera.
The core idea of the method

- MLT algorithm provides complete paths \( \{x_0, x_1, x_v, x_s\} \);
- Remove points \( x_0 \) and \( x_1 \) and consider \((x_v, x_s)\) as a "path VPL".
The core idea of the method

- MLT algorithm provides complete paths \( \{x_0, x_1, x_v, \bar{x}_s\} \);
- Remove points \( x_0 \) and \( x_1 \) and consider \( (x_v, \bar{x}_s) \) as a "path VPL".
MIR - Sample ”path VPLs”

The core idea of the method

- MLT algorithm provides complete paths \( \{ \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_v, \mathbf{x}_s \} \);
- Remove points \( \mathbf{x}_0 \) and \( \mathbf{x}_1 \) and consider \( (\mathbf{x}_v, \mathbf{x}_s) \) as a ”path VPL”.

- We do not know the outgoing radiance functions of the VPLs;
- But, we can prove that these VPLs bring the same amount of power to the camera
MIR - Cluster "path VPLs" into "geometric VPLs"

Cluster path VPLs with the same VPL location into one geometric VPL

When applying mutations, VPL locations may remain unchanged:

- The candidate is rejected and the path is duplicated;
- Only the sub-path $\vec{x}_c = \{x_0, x_1\}$ is mutated;
- Only the sub-path $\vec{x}_s$ is mutated.
MIR - Cluster "path VPLs" into "geometric VPLs"

Cluster path VPLs with the same VPL location into one geometric VPL

When applying mutations, VPL locations may remain unchanged:

- The candidate is rejected and the path is duplicated;
- Only the sub-path $\overline{x}_c = \{x_0, x_1\}$ is mutated;
- Only the sub-path $\overline{x}_s$ is mutated.
MIR - Cluster "path VPLs" into "geometric VPLs"

Cluster path VPLs with the same VPL location into one geometric VPL

When applying mutations, VPL locations may remain unchanged:

- The candidate is rejected and the path is duplicated;
- Only the sub-path $x_c = \{x_0, x_1\}$ is mutated;
- Only the sub-path $x_s$ is mutated.
Cluster path VPLs with the same VPL location into one geometric VPL

When applying mutations, VPL locations may remain unchanged:

- The candidate is rejected and the path is duplicated;
- Only the sub-path \( \bar{x}_c = \{x_0, x_1\} \) is mutated;
- Only the sub-path \( \bar{x}_s \) is mutated.

After mutations, the path is adjusted accordingly.
**MIR - Cluster "path VPLs" into "geometric VPLs"**

Cluster path VPLs with the same VPL location into one geometric VPL.

When applying mutations, VPL locations may remain unchanged:

- The candidate is rejected and the path is duplicated;
- Only the sub-path $\bar{x}_c = \{x_0, x_1\}$ is mutated;
- Only the sub-path $\bar{x}_s$ is mutated.

2 path VPLs into 1 geometric VPL

---

Benjamin Segovia  
Interactive Light Transport with Virtual Point Lights
Set of $m$ geometric VPLs $x_{vi}$

- They bring a fixed amount of power to the camera equal to
  \[ P_i = k_i \cdot P_c / n; \]

- $n$ is the total number of path VPLs;
- $k_i$ is the number of path VPLs connected to $x_{vi}$. 
Set of $m$ geometric VPLs $x_{v_i}$

- They bring a fixed amount of power to the camera equal to
  \[ P_i = k_i \cdot P_c / n; \]
- $n$ is the total number of path VPLs;
- $k_i$ is the number of path VPLs connected to $x_{v_i}$.

We do not know the "VPL power"

- Suppose that the power of the VPL is equal to 1;
- Compute the intensity of every pixel;
- Evaluate the total power $P'_i$;
- Scale all pixel intensities by $P_i / P'_i$. 
MIR - Decrease the VPL correlation

Classical Issue with Metropolis-Hastings

Example: If a VPL is on a wall, there is a high probability that the next one will be on the wall too.
MIR - Decrease the VPL correlation

Classical Issue with Metropolis-Hastings

Example: If a VPL is on a wall, there is a high probability that the next one will be on the wall too.

Increase Variance!
MIR - Decrease the VPL correlation

Classical Issue with Metropolis-Hastings

Example: If a VPL is on a wall, there is a high probability that the next one will be on the wall too.

Increase Variance!

Replace Metropolis-Hastings by Multiple-Try Metropolis-Hastings

- Idea (simplified explanation): generate many candidates at once and try to keep the best one;
- Does not really change the conception of the algorithm;
- Details in the Ph.D. thesis.
Results - MH vs MTMH - Same Computation Times

Figure: Exploration of left/right contributions (256 VPLs)
Results and Comparisons

Different tests were made

**Test scenes**

- With directly-lit scenes;
- With many light sources;
- With difficult visibility layouts.
Results and Comparisons

Different tests were made

Other algorithms

- Standard Instant Radiosity [Kel97];
- Power Sampling Technique [WBS03];
- Bidirectional Instant Radiosity.
Results - Simple Scenes - 256 VPLs - Same Computation Times

Reference (standard)

Power Sampling - 0.008%

STD - 0.02%

BIR - 0.007%

MIR - 0.009%

Figure: Tests with Shirley’s Scene 10.
Results - Difficult Visibility - 1024 VPLs - Same Computation Times

Standard / Power Sampling  Bidirectional  MIR

**Figure:** *Indirect illumination stress tests.*
Results - Difficult Visibility - 1024 VPLs - Same Computation Times

Figure: Indirect illumination stress tests.
Advantages

- Thanks to MLT → Robust and fast sampling strategies;
- Thanks to Instant Radiosity → Fast and efficient gathering techniques:
  → We can use IGI;
  → We can use rasterization techniques . . .

Non-intrusive algorithm → Can be used in any pre-existing renderer already using VPLs and Path Tracing.
Drawbacks and Future Work

**Does not handle flickering problems during animations**

- Nothing is made to ensure temporal coherency
  → if one sample changes, the whole sequence is modified;

- **Solution:** Reuse the previous samples with a sequential sampler (see [GDH06]).
And glossy and specular reflections ?!

- What happens if a part of the scene is seen through a highly glossy or a specular reflection ?

- **Solution** - Already implemented in yaCORT:
  → Instead, find the **second** diffuse surface to deposit the VPL with probability $P$;
  → While gathering, compute a camera sub-path which has a length with the same probability.
And Caustics ?!

Does not easily handle caustics.
Summary

1. Introduction
2. Formalizing the Problem
3. Sampling VPLs: Metropolis Instant Radiosity
4. Accumulating VPL contributions
5. Coherent Metropolis Light Transport
6. Conclusion
Non-interleaved Deferred Shading of Interleaved Sample Patterns

Goal

We want to accumulate the contributions of a VPL set.
Non-interleaved Deferred Shading of Interleaved Sample Patterns

Goal
We want to accumulate the contributions of a VPL set.

Issues
- Many light sources $\Rightarrow$ Large fillrate;
- Many light sources $\Rightarrow$ Many rasterization steps.
Non-interleaved Deferred Shading of Interleaved Sample Patterns

Goal
We want to accumulate the contributions of a VPL set.

Issues
- Many light sources $\Rightarrow$ Large fillrate;
- Many light sources $\Rightarrow$ Many rasterization steps.

Strategy: combine two techniques
- Deferred Shading $\rightarrow$ geometry rasterized once;
- Interleaved Sampling $\rightarrow$ decreases fill rate and maintains good image quality.
Deferred Shading [DWS+88, ST90]

**Principles**

- The per-pixel geometric information is stored in a Geometric Buffer (G-buffer) (Normals, positions and colors)
- The G-buffer is then read to perform any lighting computation.

It greatly simplifies the rendering pipeline and it also prevents the geometry from being reprojected each time a shading pass is performed.
Instead of evaluating all VPL contributions for all pixels, we use separate subsets of VPLs for every neighbor pixel.
Interleaved Sampling [KH01]

Instead of evaluating all VPL contributions for all pixels, we use separate subsets of VPLs for every neighbor pixel.
Overview of the Algorithm

- Creation
- Splitting
- Shading
- Gathering
- Discontinuity
- Filtering
- Blending
Overview of the Algorithm

Conservative extension of deferred shading: all algorithms using deferred shading may also be used with our system.
Core of the Algorithm: G-buffer Splitting

Principle

- G-buffer $G$ split into $n \times m$ smaller tiled sub-buffers $G_{i,j}$
- Texel $(x, y)$ from $G$ goes to texel $(x/n, y/m)$ of sub-buffer $G_{i, j}$ with $i = x \ mod \ n$ and $j = y \ mod \ m$. 
### Core of the Algorithm: G-buffer Splitting

#### Principle
- G-buffer $G$ split into $n \times m$ smaller tiled sub-buffers $G_{i,j}$
- Texel $(x, y)$ from $G$ goes to texel $(x/n, y/m)$ of sub-buffer $G_{i, j}$ with $i = x \mod n$ and $j = y \mod m$.

#### Fast Solution - Two-pass splitting
- Small blocks are split;
- Split blocks are translated.

#### Results: fast
- A $1024 \times 1024$ 192 bit G-buffer is split in 7 ms on a 6800GT;
- 20 ms with a one-pass splitting.
Core of the Algorithm: Filtering

Coherency of the pixels

- Discontinuity buffer;
- Box blur on continuous zones of the screen.
Coherency of the pixels

- Discontinuity buffer;
- Box blur on continuous zones of the screen.

Figure: Box Blur
Core of the Algorithm: Filtering

Coherency of the pixels

- Discontinuity buffer;
- Box blur on continuous zones of the screen.

Interleaved  Sub-sampling

Figure: Quality
Results - 500 sources - $1024 \times 768 - IS 8 \times 6$

Fully interactive applications

- No visibility for secondary light sources;
- Fast!

69 f/s
36 f/s ($\times 31$)
64 f/s
29 f/s ($\times 25$)

58 f/s
29 f/s ($\times 26$)
57 f/s
29 f/s ($\times 30$)
Physically based rendering
Visibility for secondary light sources

0.7 s - 14 f/s ($\times 3.4$)

4.0 s - 2.5 f/s ($\times 1.5$)
To sum up ...

- Generic extension of deferred shading;
- Trade-off between quality and speed.
Summary

1. Introduction
2. Formalizing the Problem
3. Sampling VPLs: Metropolis Instant Radiosity
4. Accumulating VPL contributions
5. Coherent Metropolis Light Transport
6. Conclusion
Issues with Virtual Point Lights

VPL based techniques

- Fast;
- Simple;
- Elegant.

But:

- Do not handle all lighting phenomena;
- Use the same VPL family for all pixels.
Issues with Virtual Point Lights

VPL based techniques

- Fast;
- Simple;
- Elegant.

But:

- Do not handle all lighting phenomena;
- Use the same VPL family for all pixels.

Alternative approach

Instead of making Instant Radiosity more robust, make Metropolis Light Transport more coherent and faster.
Advantages and Drawbacks of Metropolis Light Transport

Advantages

- *Conceptually* super simple;
- Very robust → it samples the density we want;
- Handles all kinds of lighting phenomena.
Advantages and Drawbacks of Metropolis Light Transport

**Advantages**
- *Conceptually* super simple;
- Very robust → it samples the density we want;
- Handles all kinds of lighting phenomena.

**Drawbacks**
- Pretty difficult to implement;
- Slow! → does not use efficient techniques like rasterization or coherent ray tracing.
Coherent Metropolis Light Transport

Core idea

Replace standard MCMC mutations by Multiple-try ones. Goal: Generating many paths at the same time.

- Uses SIMD computations;
- Factorizes cache accesses!

Fundamental for any commercial renderer
Split MLT in Three Steps

Step 1: Exploration of the sample space with MCMC mutations

Standard Metropolis Light Transport
Split MLT in Three Steps

**Step 1: Exploration of the sample space with MCMC mutations**

Provides a set of $n$ path samples $(\overline{x}_i)_{i \in [1...n]}$ with density $f(c) / ||f(c)||$ (Squares)
Split MLT in Three Steps

Step 2: Fast exploration of the lens sub-space

Lens sub-space: $ES^*DS^*(L|D)$

Sub-paths from the camera which intersect two diffuse surfaces
Split MLT in Three Steps

Step 2: Fast exploration of the lens sub-space

Use multiple-try mutations. At each step, two families:

- "Candidates" $\bar{x}_1^* \ldots \bar{x}_p^*$ (Disks);
- "Competitors" $\bar{x}_1^{**} \ldots \bar{x}_p^{**}$ (Triangles).
Split MLT in Three Steps

Step 3: Accumulate all sample contributions

- Each family: use of the "expected value": accumulate $\overline{x}^*$ according to $R_g$ and $\overline{x}^{**}$ according to $1 - R_g$;
- Each element: accumulate each element $\overline{x}$ proportionally to $f^{(c)}(\overline{x})$
Implementation - Mutation strategies

Exploration of the entire space: standard MLT with bidirectional mutations only;
Implementation - Mutation strategies

Exploration of the **lens** sub-space: use of lens and caustics perturbations.

- Jittering;
- Gaussian distributions around the initial samples;
- "Packetize" the rays → use SIMD instructions.
Implementation - Mutation strategies

Exploration of the lens sub-space: use of lens and caustics perturbations.

With pictures . . .
Results

Quality equivalent to the quality obtained with MLT but ...

As MLT, some parameters have to be carefully tuned:

- Lengths of MTMH sub-sequences;
Results

Quality equivalent to the quality obtained with MLT but . . .

As MLT, some parameters have to be carefully tuned:

- Lengths of MTMH sub-sequences;
- $\sigma$ and the number of MTMH candidates.

(a) $\sigma = 16$ pixels  
(b) $\sigma = 32$ pixels  
(c) $\sigma = 64$ pixels
Results

Quality equivalent to the quality obtained with MLT but . . .

As MLT, some parameters have to be carefully tuned:
- Lengths of MTMH sub-sequences;
- \( \sigma \) and the number of MTMH candidates.

Overall performance with SIMD

Speed-up from 1.5 to 2.3.
Affinity with caches is *fundamental*

- Distribution ray tracing in complex scenes [CLF+03, Chr06];
- Multi-resolution geometry caching;
- On-the-fly tesselation;
- Memory systems with difficult layouts:
  - Cell processors (PS3, Blade Center)
  - Xenon (XBOX 360)
  - Larabee
  - PC cluster...
## Results - Cache Simulation

<table>
<thead>
<tr>
<th></th>
<th>Theater</th>
<th>Three Dragons</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MLT</td>
<td></td>
<td></td>
<td>4 Tri</td>
<td>16 Tri</td>
<td>128 Tri</td>
<td>4 Tri</td>
</tr>
<tr>
<td>4 Tri</td>
<td>53%</td>
<td>61%</td>
<td>60%</td>
<td>54%</td>
<td>63%</td>
<td>70%</td>
</tr>
<tr>
<td>16 Tri</td>
<td>92%</td>
<td>98%</td>
<td></td>
<td>81%</td>
<td>95%</td>
<td>99%</td>
</tr>
<tr>
<td>128 Tri</td>
<td>99%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMLT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Tri</td>
<td>86%</td>
<td>81%</td>
<td>82%</td>
<td>54%</td>
<td>63%</td>
<td>70%</td>
</tr>
<tr>
<td>16 Tri</td>
<td>92%</td>
<td>95%</td>
<td>95%</td>
<td>63%</td>
<td>95%</td>
<td>99%</td>
</tr>
<tr>
<td>128 Tri</td>
<td>98%</td>
<td>99%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Tri</td>
<td>87%</td>
<td>82%</td>
<td>82%</td>
<td>54%</td>
<td>63%</td>
<td>70%</td>
</tr>
<tr>
<td>16 Tri</td>
<td>93%</td>
<td>95%</td>
<td>95%</td>
<td>63%</td>
<td>95%</td>
<td>99%</td>
</tr>
<tr>
<td>128 Tri</td>
<td>99%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Faster than MLT

- Simple extension of MLT → reorganization of the computations;
- Not real time (and not even interactive) but may be a good alternative.
Faster than MLT

- Simple extension of MLT $\rightarrow$ reorganization of the computations;
- Not real time (and not even interactive) but may be a good alternative.

But ...

As MIR, does not handle flickering problems during animation. It is a major problem with "importance driven" methods.
Summary

1. Introduction
2. Formalizing the Problem
3. Sampling VPLs: Metropolis Instant Radiosity
4. Accumulating VPL contributions
5. Coherent Metropolis Light Transport
6. Conclusion
During three years . . .

- Many implementations: GPU, coherent ray tracing;
- Many numerical schemes.
But ... no "ultimate" renderer was found!

However ... some personal points of view

For absolute realism and large interactivity:

**Monte-Carlo + Brute Force + Carefully Designed Implementation is the only solution**

- No compression, no PRT, no expensive pre-computation;
- Rasterization vs ray tracing → Geometric efficiency vs lighting simulation efficiency?
Merci!
Per Christensen.
Ray Tracing for the Movie "Cars".

Per Christensen, David M. Laur, Julian Fong, Wayne L. Wooten, , and Dana Batali.
Ray Differentials and Multiresolution Geometry Caching for Distribution Ray Tracing in Complex Scenes.

Philip Dutré, Eric Lafortune, and Yves Willems.
Monte Carlo Light Tracing with Direct Computation of Pixel Intensities.

Michael Deering, Stephanie Winner, Bic Schediwy, Chris Duffy, and Neil Hunt.

Abhijeet Ghosh, Arnaud Doucet, and Wolfgang Heidrich.
Sequential Sampling for Dynamic Environment Map Illumination.

Henrik Wann Jensen.  
Global Illumination Using Photon Maps.  

Henrik Wann Jensen.  
Rendering Caustics on Non-Lambertian Surfaces.  

Henrik Wann Jensen.  
*Realistic image synthesis using photon mapping.*  
James Kajiya.
The Rendering Equation.

Alexander Keller.
Instant Radiosity.

Alexander Keller and Wolfgang Heidrich.
Interleaved Sampling.


Takafumi Saito and Tokiichiro Takahashi. *Comprehensible Rendering of 3-D Shapes.*

**Eric Veach.**


**Eric Veach and Leonidas Guibas.**

Bidirectional Estimators for Light Transport.


**Eric Veach and Leonidas Guibas.**

Metropolis Light Transport.

Gregory Ward.
The RADIANCE Lighting Simulation and Rendering System.

Ingo Wald, Carsten Benthin, and Philipp Slusallek.
Interactive Global Illumination in Complex and Highly Occluded Environments.

Gregory Ward, Francis Rubinstein, and Robert Clear.
A Ray Tracing Solution for Diffuse Interreflection.