

# Physically based rendering

## Progressive photon mapping

Kévin Bannier   Billel Hélali   Amaury Louarn

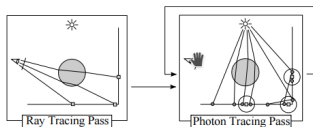
École Supérieure d'ingénieurs de Rennes  
Université de Rennes 1

March 17, 2016

# Definition

- Photon mapping belongs to the family of the particle-tracing algorithms
- Constructing paths from the light onto the objects
- 2 stages:
  - ① Generating and sending the photons, with arbitrary accuracy and without storing the full photon map
  - ② Final gathering
- Repeat this principle several times and calculate the average picture

## More precisely



Multi-pass algorithm:

- ① Ray tracing pass
  - ray tracing to find all the visible surfaces
- ② Photon tracing passes
  - Every photon tracing increases the solution
  - Convergence to an optimal solution in infinite time

## Criteria of comparison

Several parameters for the progressive photon mapper:

**Photons per iteration:** The number of photons to be sent at each iteration

**Maximum depth:** The maximum depth of the final gathering rays

**Russian roulette starting depth:** The depth before starting to eliminate photons with the russian roulette algorithm

**Initial radius:** Initial photon query radius

# Photons per iteration

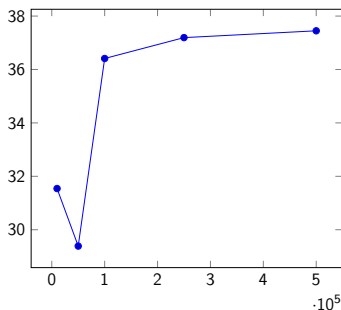


Figure: PSNR according to the number of photons per iteration

# Maximum depth

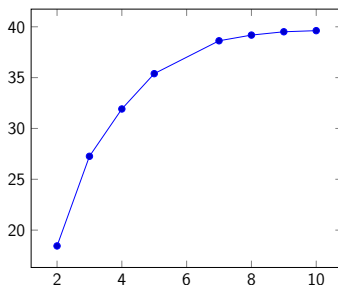


Figure: PSNR according to the maximum depth

# Initial radius

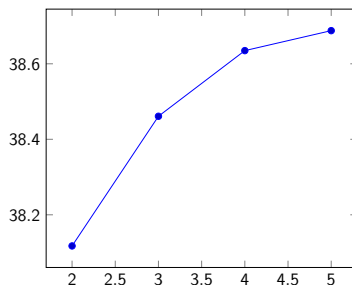


Figure: PSNR according to the initial radius

## Russian roulette starting depth

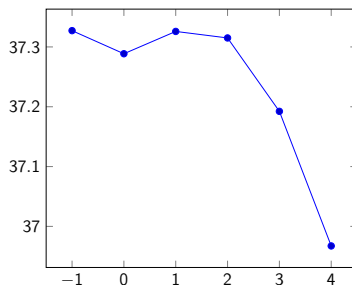


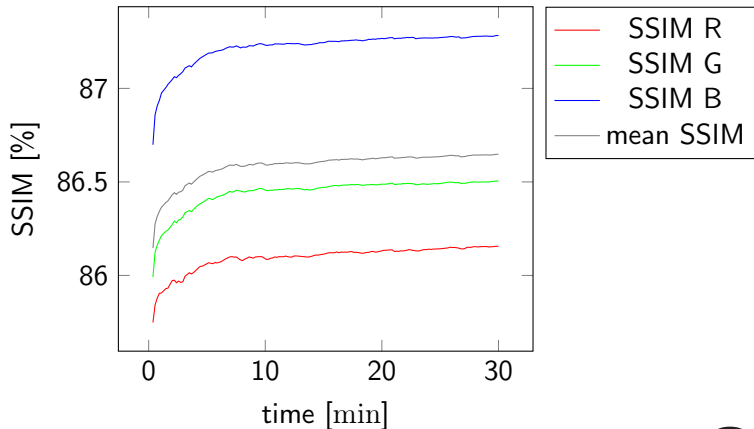
Figure: PSNR according to the russian roulette starting depth



# Context

- Image rendering for the cinema (e.g special effects)
- Time-constrained rendering (each render must not take more than a specified amount of time)
- Possibility of a render farm: the method must be scalable
- Signal-perfect images are not necessary:
  - Each image will not be on the screen for more than 1/24th of a second
  - (Often) presence of motion blur

## Perception by time



## Render process (1/4)



## Render process (2/4)



## Render process (3/4)



## Render process (4/4)

