

# The RADIANCE Lighting Simulation and Rendering System

Written by Gregory J. Ward  
Lighting Group  
Building Technologies Program  
Lawrence Berkeley Laboratory

COMPUTER GRAPHICS Proceedings, Annual Conference Series, 1994

Presentation by XXXXXXXXXX  
Date: 05 May 2008

## Background

- Few practical applications available for benefiting Architecture and lighting design
- Radiosity has only recently been introduced into the Rendering market
- Methods are too compute-intensive for most applications of the time
- Acceptance should improve once more real-life applicability is proven

# Challenges

- 3 Challenges to physically-based rendering for architecture
  - Calculations must be accurate
  - Must be general
  - Must be practical

# Introducing Radiance

- A rendering system developed by author at Lawrence Berkeley Laboratory (LBL) and Ecole Polytechnique Federale de Lausanne (EPFL) in Switzerland.
- Originally a study in ray-tracing algorithms for saving energy through better lighting design
- Funded by US Department of Energy, later the Swiss Government
- Enhancements made were outcome of real or perceived user requirements
- First licensed in 1989 and has increased in users in the research and design community
- 9 years of user-stimulated software evolution

# System Design Goals

- Ensure accurate calculation of luminance
- Model both electric light and day light
- Support a variety of reflectance models
- Support complicated geometry
- Take unmodified input from CAD systems

# The Approach: Hybrid Deterministic/Stochastic Ray Tracing

- *Radiance* uses Ray-tracing, which is done through evaluating Kajiya's rendering equation:

$$L_r(\varphi_r, \Phi_r) = L_e(\varphi_r, \Phi_r) + \int \int L_i(\varphi_i, \Phi_i) \rho_{bd}(\varphi_i, \Phi_i; \varphi_r, \Phi_r) |\cos\varphi_i| \sin\varphi_i d\varphi_i d\Phi_i$$

Where:

$\varphi \Rightarrow$  Polar angle measured from surface normal

$\Phi \Rightarrow$  Azimuthal angle measured about the surface normal

$L_e(\varphi_r, \Phi_r) \Rightarrow$  Emitted radiance (watts/steradian/meter<sup>2</sup>)

$L_r(\varphi_r, \Phi_r) \Rightarrow$  Reflected radiance

$L_i(\varphi_i, \Phi_i) \Rightarrow$  Incident radiance

$\rho_{bd}(\varphi_i, \Phi_i; \varphi_r, \Phi_r) \Rightarrow$  Bidirectional reflectance-transmittance distribution function (steradian<sup>-1</sup>)

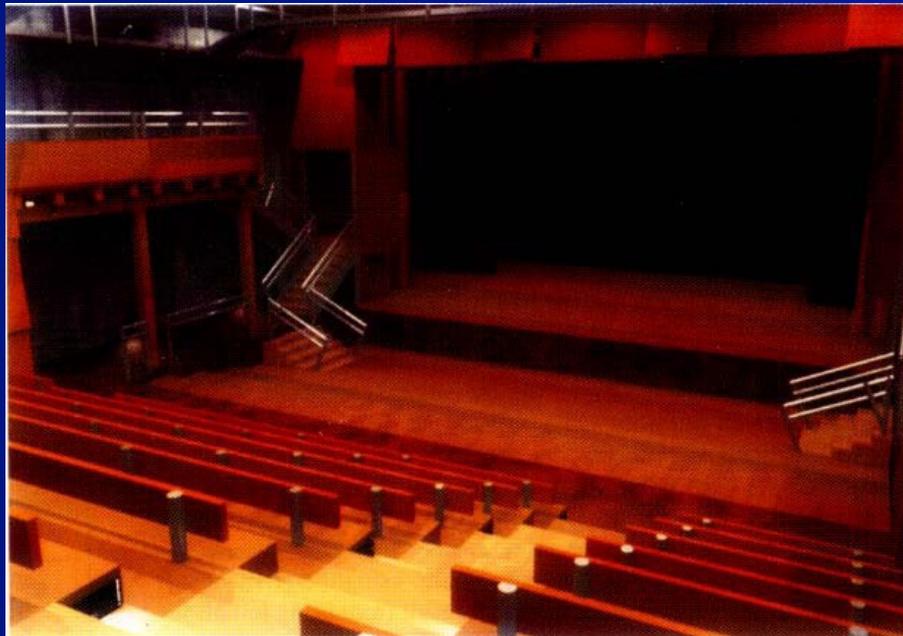
- Energy transfer between two points replaced by energy passing through a point in a specific direction (i.e. radiance)

# The Approach: Cached Indirect Irradiances for Diffuse Interreflection

- Most ray-tracing calculations ignore diffuse interreflection, using a constant “ambient” instead
- Cost of diffuse indirect contributions is too great to recalculate at every pixel
- Indirect lighting changes gradually, making it possible to spread out this influence over many pixels to obtain a smoother result.
- *Radiance* uses cached irradiance values which is enhanced by using gradient information
- Evaluate Kajiya’s equation for indirect diffuse contributions only as needed
- Direct and specular components are still computed on per-pixel basis, but hemispherical sampling occurs less frequently

# The Approach: Adaptive Sampling of Light Sources

- *Radiance* uses a prioritized list of potential source contributions achieved through evaluating Kajjiya's equation.
- Contributions with largest potential are tested for shadows first, stopping when the remainder is below some fraction.
- Remaining sources are added based on statistical likelihood of visibility



A theatre lighting simulation generated by Radiance, circa 1989

# The Approach: User-directed Preprocessing of Secondary Sources

- Handling of secondary sources, such as daylight entering through a skylight, requires significant effort for proper effect, not just Monte Carlo sampling.
- *Radiance* allows the user to identify light source from a window to allow collection of samples of transmitted radiation over all points of the window, a 4-dimensional function.



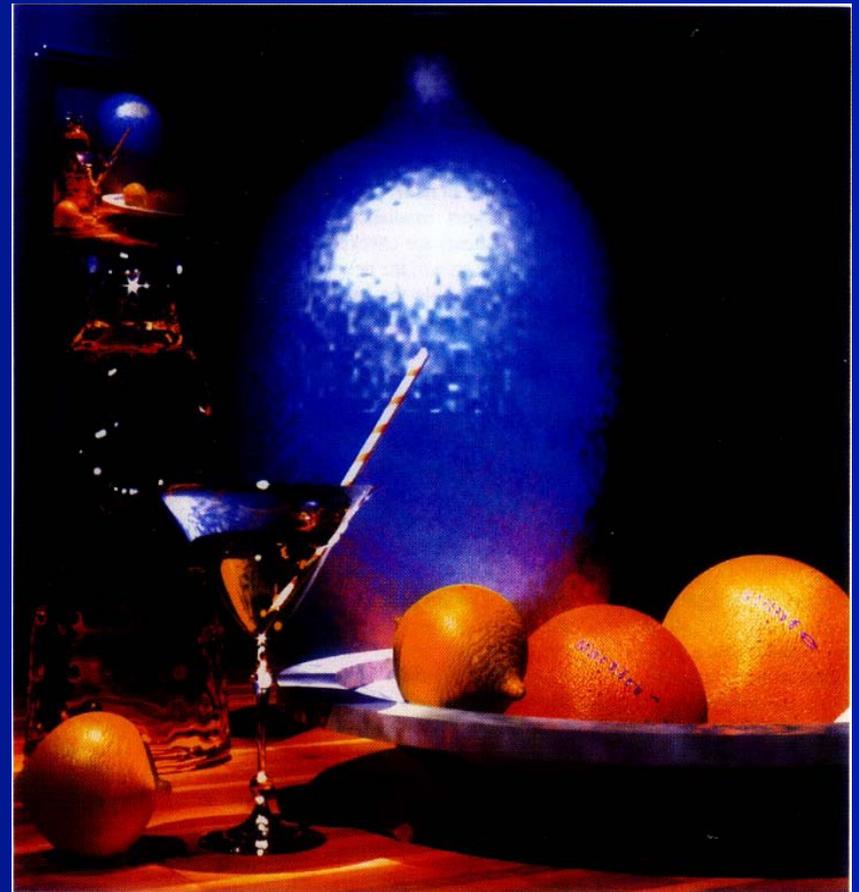
Window and ceiling specified as light sources



Monte Carlo path tracing calculation

# The Approach: Patterns and Textures

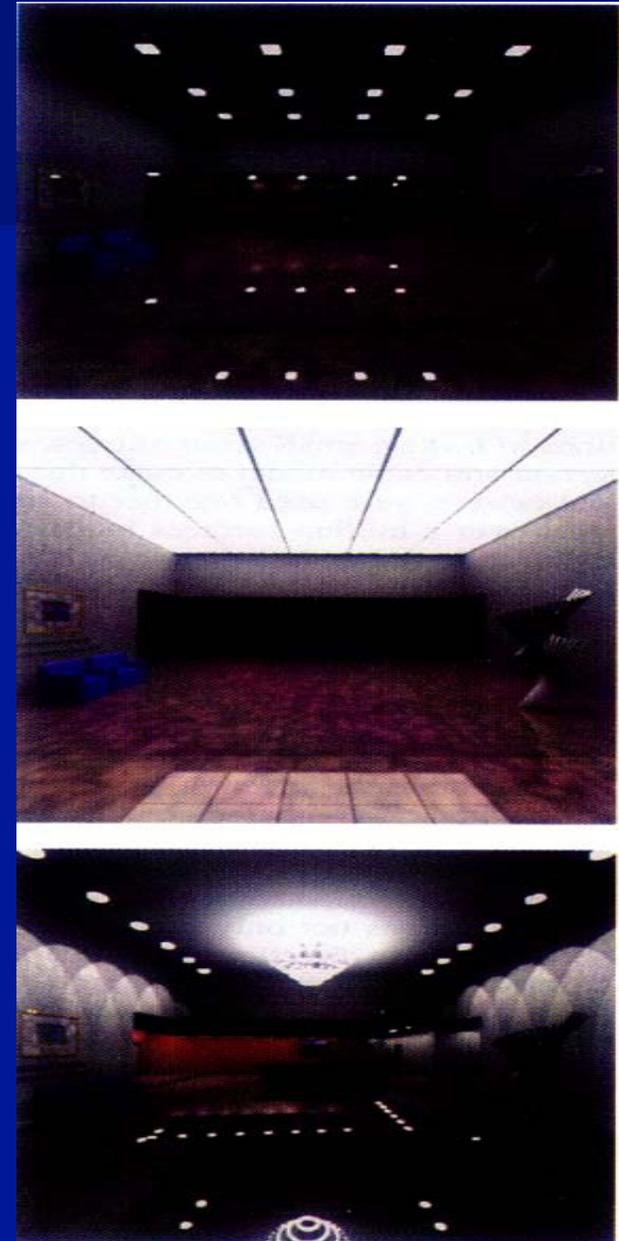
- In *Radiance*, variation in surface color and/or brightness is a pattern.
- Perturbation of the surface normal is texture.
- System contains mapping of lookup functions for multi-dimensional data.
- Inspired by Perlin's flexible shading language, interpreting tabulated or image data is done through the same functional language used for procedural patterns and textures.



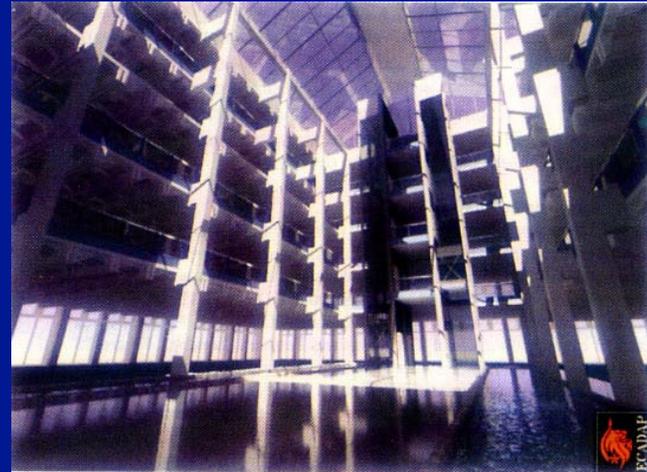
Textures seen on oranges, lemons, and vase are procedural. Pattern is seen on the bowl.

# Electric Lighting

- First domain in *Radiance*
- Several application areas where electric lighting is emphasized, but most obvious is lighting design.
- This figure shows a comparative study of 3 possible lighting alternatives in a hotel lobby space.
- The determination of model is left to the client



# Day Lighting



- Here are two distinct models involving day lighting.
- On the left is the exterior of the Mellencamp Pavillion
- On the right is a more complex model of the U.K. Atrium, using only *Radiance* scene generation utilities.

# Implementation

- *Radiance* consists of a collection of C programs designed to work in concert, compilable on most UNIX platforms
- Sequence of events in modeling using Radiance:
  - Model -> convert -> render -> filter -> display
- A typical session may start with the user creating or modifying a geometric model using CAD.
- The CAD model is translated into a Radiance scene description file
- User may modify model by creating or changing materials, patterns and textures
- The model is compiled by *oconv* into an octree file, passed to interactive renderer, *rview*, to verify desired view and calculation parameters.
- Batch rendering is started with *rpict* and the raw picture is filtered

## Current Status

- Still in Use today
- Continued work has expanded its viability in specific applications in lighting design and architectural applications
- Website for *RADIANCE*:
  - <http://radsite.lbl.gov/radiance/HOME.html>