Computer Graphics
Looking Beyond Games

Ian Ashdown, P. Eng. (Ret.), FIES
Senior Scientist
SunTracker Technologies Ltd.

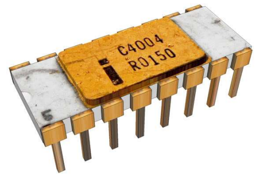
When we think about computer graphics, the first thing that comes to mind is computer games. After all, it is computer games that drives the innovation of massively parallel processing GPUs at nVidia and AMD. Everything else would appear to pale in comparison for these companies. (Well, apart from bitcoin mining, which is another topic entirely.)

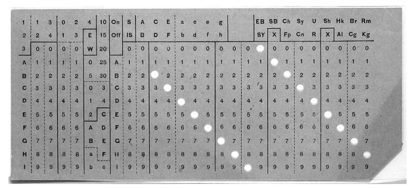
To be honest, it is also computer games that have driven computer graphics research and development for the past several decades. From mobile phones to Hollywood blockbusters, it has all come from the need to develop ever more engaging and exciting computer games.

But I am not here to talk about computer games.

Computers at UBC (1970s)

- UBC Computer Centre
  - IBM 360 mainframe
  - Fortran IV
  - Hollerith punch cards
  - 2 <u>seconds</u> of CPU time per EE course
- Intel 4004
  - Introduced in 1971
  - 4-bit CPU
  - 46 instructions
  - 750 kHz clock
  - 640 addressable bytes



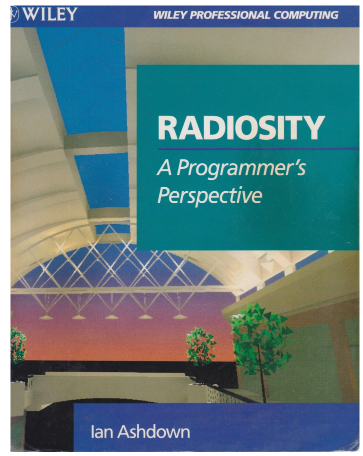


Rather than discuss my background, I will quickly mention that I graduated from UBC in 1973 as an electrical engineer. The opportunities to work with computer were at that time were, shall we say, somewhat limited. The UBC Computer Centre had an IBM 30 mainframe that electrical engineering students could write programs for using Fortran IV and Hollerith punch cards. We received a total of two <u>seconds</u> of CPU time per course.

But the future had arrived, even if nobody noticed at the time. In 1971, Intel released the 4004 CPU, featuring a 4-bit arithmetic logic unit, a 750 kHz clock and 640 addressable bytes.

Radiosity: A Programmer's Perspective

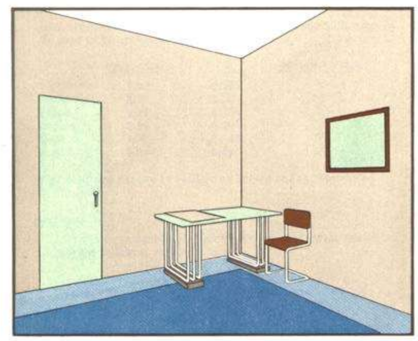
- Published October 1994
- 500 pages
- Complete introduction to computer graphics
- Includes full C++ source code for Windows
- Freely available on www.researchgate.net



Fast-forward twenty years, during which time I taught myself software engineering and computer graphics … which led to my writing a book on what was then one of the hottest topics in computer graphics research, *radiosity*.

Radiosity History

- Illumination Engineering
  - Yamauti (1926)
  - Buckley (1927)
  - Higbie (1934)
- Thermal Engineering
  - Hottel(1954)
  - Eckbert & Drake (1959)
- Computer Graphics
  - Goral (1985)
  - Nishita & Nakamae (1985)



Moon & Spencer (1948) – First Radiosity Image

The radiosity method has a curious history. It was developed in the 1920s as a mathematical approach to modeling the transfer of visible light between ideal diffuse surfaces, and then formalized as a lighting calculation technique in a 1934 book called "Lighting Calculations." In an era of hand-cranked adding machines, however, it was never considered practical and soon all but forgotten.

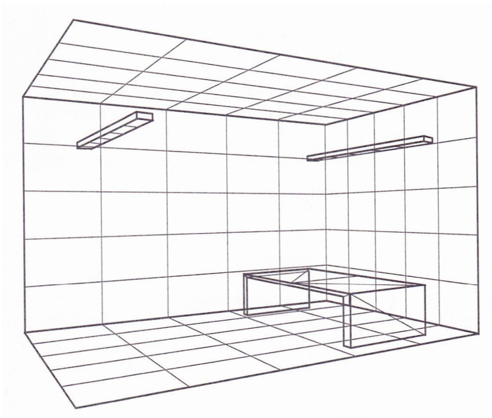
The method was rediscovered in the 1950s by thermal engineering researchers interested in modeling the transfer of radiant heat (that is, infrared radiation) between ideal diffuse surfaces. This led to entire books of complex analytic equations for the transfer of radiant heat between surfaces with different shapes, but nothing that resembled computer graphics.

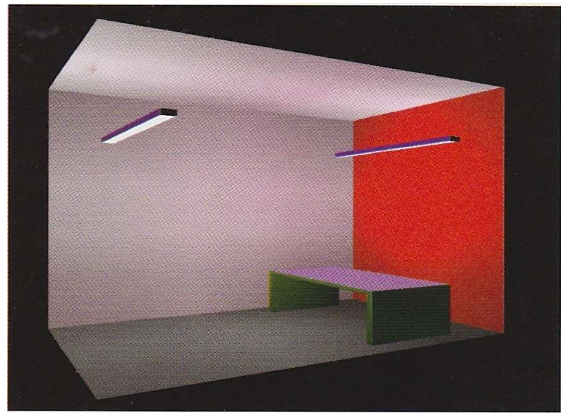
This all changed in 1985 with the independent re-discovery of the thermal engineering method by computer graphics researchers at Cornell and in Japan. By 1992, it was possible to generate photorealistic and physically-based renderings using high-end workstations that cost (in today's currency) $150,000.

(There is an important lesson here, by the way – if you invent something, never assume that it is new. The chances are that somebody already thought of it years and possibly decades before you.)

What is Radiosity?

- Follow the bouncing … light





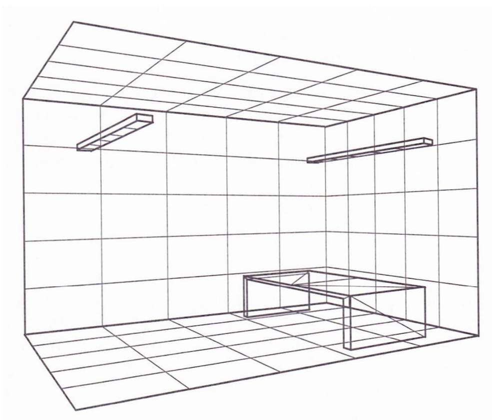
My self-appointed challenge in 1992 was to write a book on the topic, even though I knew <u>nothing</u> about computer graphics at the time. Ignorance however can be a wonderful thing, as you do not know that what you are trying to do is “impossible.” Eighteen months later, I had written the book and, more important, written the C++ software to run on a Windows 3.1 machine with four megabytes of memory. The book has been out of print for twenty years, but the (much enhanced) software is still being used by some 40,000 professional lighting designers today.

So how does radiosity work as a computer graphics technique? Looking at the wireframe model, each surface is divided into an array of “patches.” The light emitted by each lighting fixtures (or more properly, “luminaire”) is distributed to each patch fully or partially visible to it. Some of the light received by each patch will, depending on its reflectance, be absorbed, but the remainder is ready to be diffusely reflected.

The radiosity method then proceeds iteratively by choosing the patch with the most amount of diffusely reflected light, which it distributes (or “shoots”) to all other patches visible to it. The method terminates when the amount of remaining light is below a predetermined threshold (typically five percent). Colloquially, you can think of light “bouncing” from surface to surface.

What is Radiosity?

- Calculating form factors … with analytic geometry



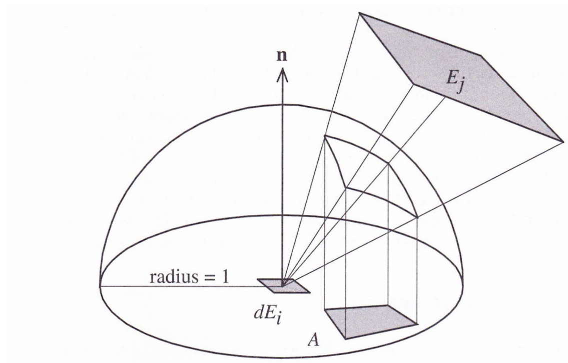


Figure 5.7 Nusselt's analogy

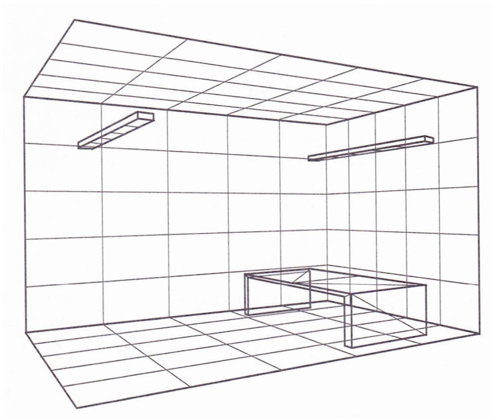
But the devil is in the details. The question that must be answered for each bounce is, "How much light reflected from one patch is received by another?" There is a conceptually simple method of determining this using something called "Nusselt's analogy," but I will quote someone who wrestled with the mathematics behind it:

"Although this task appears very simple, its solution is considerably more knotted than one would expect … the highly laborious computation would fill even the most patient with disgust and drive them away."

This was written by a pioneer researcher in photometry named Johann Lambert … in 1760! Some problems take literally centuries to solve.

What is Radiosity?

- Calculating form factors … the lazy way



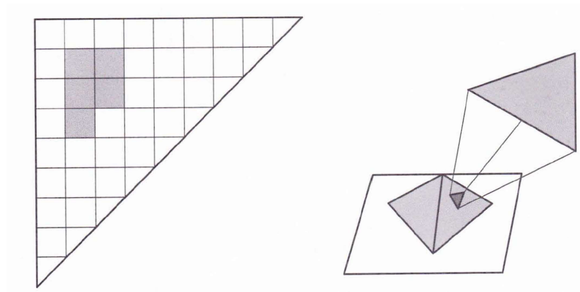


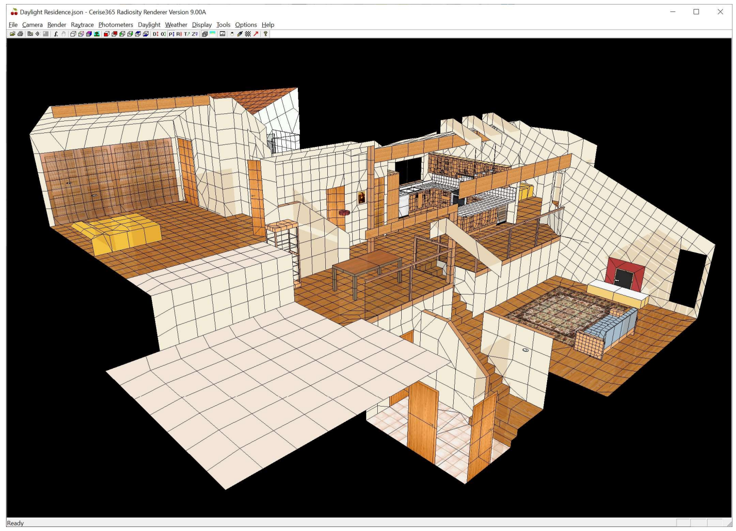
Figure 5.16 Polygon projection onto a cubic tetrahedron face

The solution came in 1988 with a fresh approach from computer graphics. Imagine that you are at the center of a patch and looking out into the environment through three triangular windows. If you think of each window as being a computer display, you need to do no more than count the number of pixels representing whatever other patch you are interested in, even if it is partially obscured. That’s it – there is no need for analytic or spherical geometry, nor complex equations of any kind.

There is another lesson here by way of analogy. In the 1970s, it was not at all clear how three-dimensional objects should be displayed on a computer screen. There were numerous algorithms developed to solve the problem, but in the end the simplest approach won – the “Painter’s algorithm” relied on simple brute force and fast hardware. Analytic solutions are wonderful for understanding a problem, but a successful engineering solution only needs to be “good enough” to solve it.

Radiosity Today

- CAD model import
- Physical light sources
- Mostly diffusely reflective materials
- Architectural glazing
- Texture maps



This brings us to today's use of the radiosity method. While computer games development today is focused on real-time ray tracing supported with machine learning to denoise images on the fly, radiosity is still being used in the background to generate geometry with static lighting that can be precomputed and "baked" into texture maps.

But let us look beyond computer games. Professional lighting designers need to take architect's three-dimensional CAD models, add luminaires, and calculate the distribution of light within the virtual environment. Unlike computer graphics, there is a strict requirement that the result be physically accurate. This means using intensity distribution data measured by the luminaire manufacturer, along with mostly diffusely-reflective materials and physically-based models of architectural glazing. Fine details are best handled with texture maps.

Radiosity Today

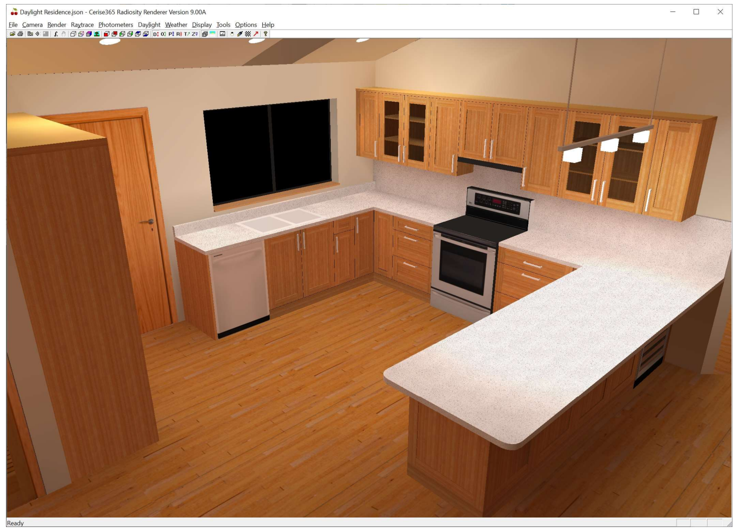
- Photorealistic images
- Physically correct
- 10,000 bounces in 14 seconds
- Equivalent to 70 million rays / second

The results include photorealistic images to show the architects and their clients. More impressive rendering can be obtained from architectural visualization programs, but the important issue is that the lighting calculations are physically-based. (This is a critical requirement for engineering tools that are used for architectural lighting design.)

The radiosity method is also fast – this particular environment was rendered in 14 seconds using 10,000 bounces of light while running as a multithreaded application on a commodity desktop computer. This is equivalent to about 70 million rays per second, and done without any GPU support. (The radiosity method is unfortunately incompatible with GPU architectures.)

Radiosity Today

- Interactive 3D
- Virtual photometers
- Isolux plots



A particular advantage of the radiosity method is that unlike ray-traced architectural renderings, it is view-independent. This means that the user can interactively tour the three-dimensional environment.

More important, the lighting calculations are physically correct – a virtual photometer can be used to take illuminance measurements anywhere within the environment. For lighting designers, this means being able to generate isolux plots, and from them determine whether the light levels are appropriate or meet building standards throughout the design space.

Radiosity as a Tool

- Radiosity is an engineering tool
- Pioneer 10 & 11 gravity anomaly
  - Launched in early 1970s to explore Jupiter and Saturn
  - Unexplained acceleration ($8.74 \pm 1.33 \times 10^{-10}$ m/sec) towards Sun resulted in 400 km course error by 1998
  - Problem finally solved in 2012 … using radiosity method

Francisco, F., et al. 2012. "Modelling the Reflective Thermal Contribution to the Acceleration of the Pioneer Spacecraft," arxiv.org/abs/1103.5222.

The point here is that the radiosity method is much more than just another computer graphics technique for computer games development; it is an engineering tool.

An interesting example involves the Pioneer 10 and 11 spacecraft that were launched in the early 1970s, and which a half-century later are exploring the boundary between the heliosphere and interstellar space. They have been experiencing a small but unexplained acceleration for decades. This problem was finally solved in 2012 by carefully modeling the spacecraft and then using the radiosity method to predict the distribution of internally-generated radiant heat between its surfaces. With this information, it was possible to explain that the spacecraft surfaces were acting like solar sails, pushing them towards the Sun.

In a broader sense, it is not just the radiosity method that can serve as an engineering tool. Ray tracing methods have also been used, notably by researchers using Lawrence Berkeley National Laboratories' *Radiance* software suite.

Radiosity Equation

- Beguilingly simple problem:

$$\begin{bmatrix} M_{o1} \\ \vdots \\ M_{on} \end{bmatrix} = \begin{bmatrix} 1 - \rho_1 F_{11} & \dots & -\rho_1 F_{1n} \\ \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & \dots & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} M_1 \\ \vdots \\ M_n \end{bmatrix} \quad \text{or } \mathbf{M}_o = (\mathbf{I} - \mathbf{RF})\mathbf{M}$$

- Fredholm integral of second kind requires iterative solution:

$$\mathbf{M} = \mathbf{M}_o + \mathbf{RFM}_o + (\mathbf{RF})^2\mathbf{M}_o + (\mathbf{RF})^3\mathbf{M}_o + (\mathbf{RF})^4\mathbf{M}_o + \cdots$$

- In physical terms, follow the bouncing light!

Following the idea of computer graphics techniques as the basis of engineering tools, there is another thread to follow here. Many of the computer graphics techniques, whether they involve modeling lighting light, particle systems, fluid dynamics, shape recognition, crowd behaviour, image analysis, or whatever, can be described using wondrously complex and often fascinating mathematics. Once these are embedded in commercial production tools however, whether it is Maya for computer animation or Ansys for computational fluid dynamics analysis, we tend to accept them for what they are.

It sometimes pays, however, to look under the hood and really understand what is going on. At the risk of appearing single-minded, the radiosity method provides an interesting example. The radiosity equation that underlies the method is a simple matrix equation, but it can only be solved iteratively as a “Fredholm integral of the second kind.” What is not at all evident from these equations is that each term represents one bounce of light between surfaces.

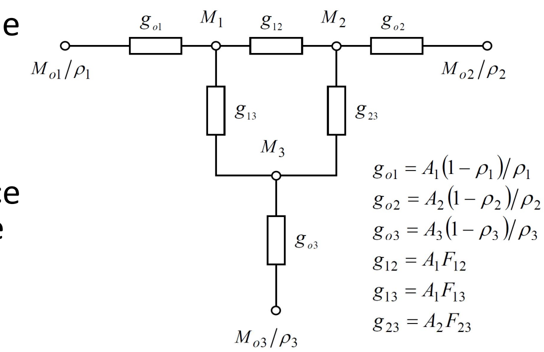
Once you understand the physical significance of an equation, it becomes much easier to reason about it and see whether it might be improved upon.

Eigenvector Radiosity

- Eight years (1993 – 2001) to solve the problem
- Partial solution came from electrical engineering (network theory)
- Another seven years (2008) to reduce a 150-page MSc thesis to one simple equation:

$$\mathbf{M} = \sum_{i=1}^{k} \sigma_i^{-1} v_i u_i^T \mathbf{M}_o$$

using the singular value decomposition of $(\mathbf{I} - \mathbf{RF})$

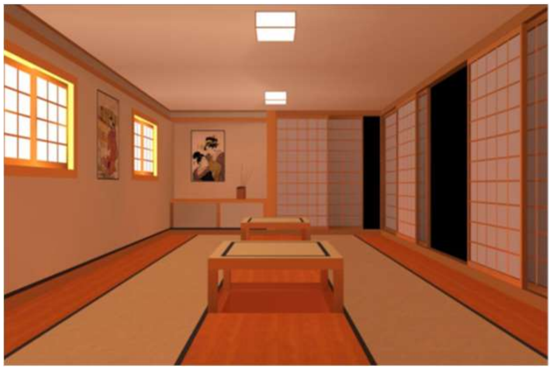


There is another lesson here – whenever you have a difficult problem in your field, look outside the field for parallels with other disciplines. In the case of my Master's thesis at UBC, I understood the physical significance of the radiosity equation, but I was nagged by the suspicion that there might be a direct method for solving the equation rather than an iterative one. After eight years of work (but only the last two as a graduate student), I found a partial solution in a long-forgotten electrical engineering paper from the 1950s.

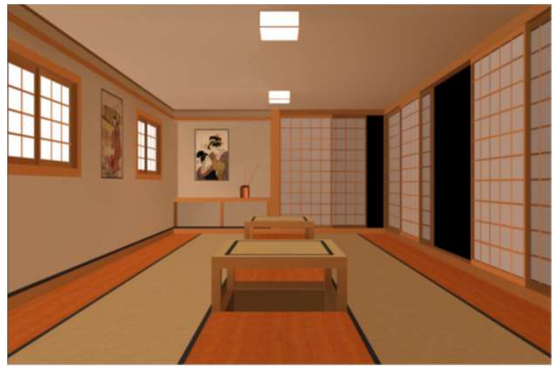
This resulted in a 150-page MSc thesis titled "Eigenvector Radiosity," but it took another seven years to reduce the solution to a simple equation.

Eigenvector Radiosity

- Ignoring the mathematics …



Progressive Radiosity
1,112 bounces
41 seconds



Eigenvector Radiosity (20 eigenvectors)
8 bounces
11 milliseconds

So why does this matter? Well, the equation results in a speed-up of the radiosity calculations by a factor of several thousand times – from 41 seconds to 11 milliseconds on a desktop computer twenty years ago.

Unfortunately, this approach introduces a new problem – determining the singular value decomposition of a matrix with millions of elements. We are still working on this problem, lately with the assistance of machine learning techniques.

This leads to yet another lesson: whatever ground-breaking new idea you may have, it will likely take at least 10,000 person-hours of work to turn it into a commercial success. You will also find along the way that few if any "good ideas" stand alone – they require an entire ecosystem of related technologies to bring them to fruition.

Computer Games

- PONG was created by Allan Alcom in 1972
- In 1992, SIGGRAPH staged a PONG game with 5,000 people in a Los Angeles auditorium
- Each person was given a paddle with reflective red and green sides
- The PONG paddle was controlled by video cameras and majority voting … RED! GREEN! GREEN! RED!
- After participating in this mayhem, any other multi-player game has seemed somehow pointless.

But first an interlude – I need to explain why I have not played a computer game of any description in nearly thirty years.

I attended my first ACM SIGGRAPH conference in 1992 when it was a relatively small gathering of fewer than 10,000 people. Five thousand of us crowded into a Los Angeles Convention Center auditorium, where we were given paddles with reflective red and green sides. The lights went down, and we were looking at a massive screen displaying the very first video game, PONG.

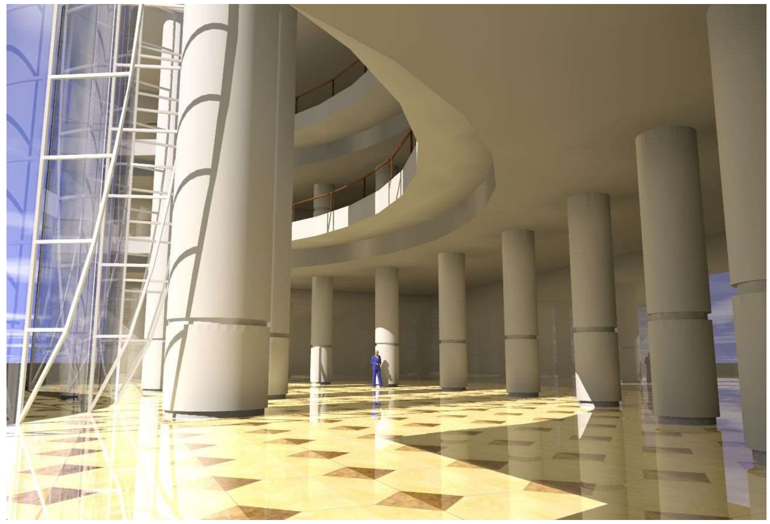
PONG is moronically simple – a small rectangle that bounces back and forth between the vertical edges of the screen, and paddles moving vertically on each side for players to (hopefully) hit the ball back to the other side.

… except in our case, we had 2,500 people on each side, frantically waving their paddles while shouting “RED! GREEN! GREEN! RED!” to direct the on-screen paddle by majority vote.

How can any multiplayer video game ever compare to such mayhem?

Beyond Computer Games

- Lighting design
  - Architectural

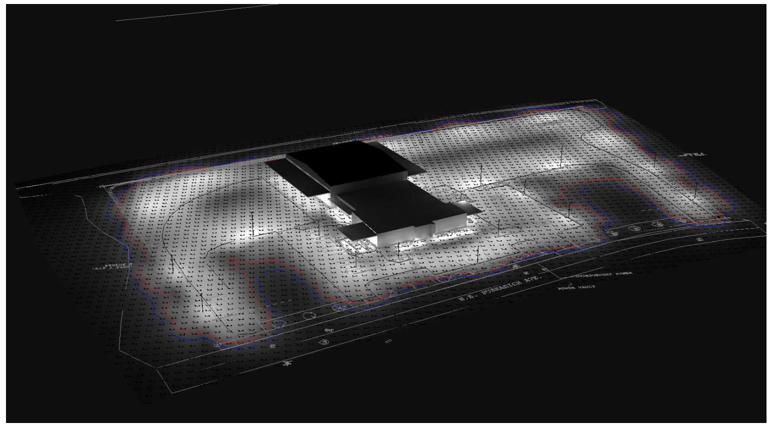


Now, back to the radiosity method to answer the question, what can you do with it besides develop ambient lighting renderings for computer games?

Architectural lighting design is one obvious answer. This particular image relied on the radiosity method to compute most of the lighting, followed by ray tracing for the direct sunlight. However, it is just as easily accomplished using radiosity for the diffuse lighting and OpenGL or WebGL to provide the direct sunlight effects. The advantage is that the all of the calculated lighting values are physically correct, and you can tour the environment interactively.

Beyond Computer Games

- Lighting design
  - Architectural
  - Roadway / sports



Next up is lighting design for roadways and sports fields. These are almost separate disciplines for professional lighting designers, as there are few surfaces for light to bounce between. Instead, the focus is on the even distribution of light while avoid in visual glare.

Beyond Computer Games

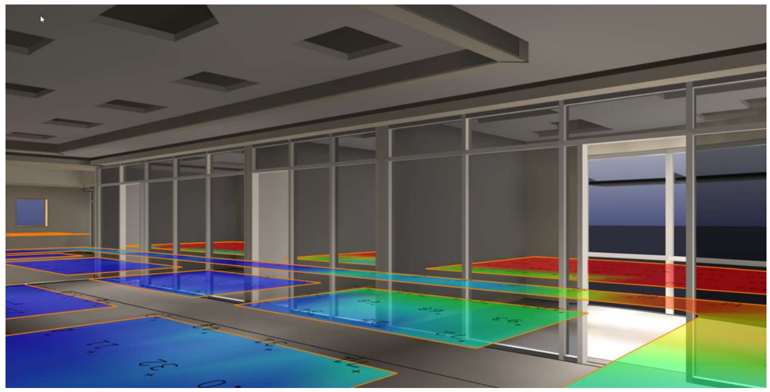
- Lighting design
  - Architectural
  - Roadway / sports
  - Industrial

Industrial lighting has its own challenges, where it is necessary to ensure that the light levels are in accordance with industrial safety standards.

These are admittedly straightforward applications, but there are more interesting ones to come.

Beyond Computer Games

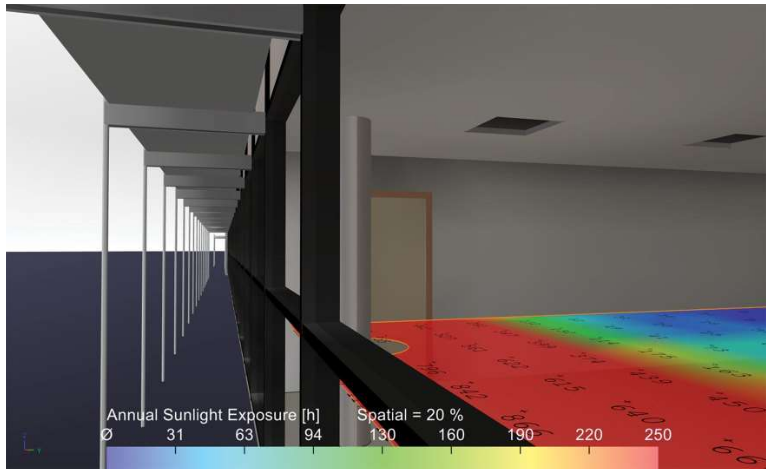
- Lighting design
  - Architectural
  - Roadway
  - Industrial
  - Daylighting



Daylighting design is where it starts to get interesting. Understanding this field requires an in-depth knowledge of astrometrics (i.e., the position and movement of the Sun across the sky), meteorology (i.e., weather), atmospheric physics, and architectural glazing material properties. Computer graphics techniques asre involved, but they are only the beginning.

Beyond Computer Games

- Building Energy Modeling
  - Climate-based annual daylighting
  - 4,380 hourly records per year
  - Solar insolation
  - Visual glare
  - Thermal comfort



Daylighting design becomes even more interesting when building energy modeling is involved. This is a relatively small but increasingly important field of architectural engineering, particularly as we move towards green and net-zero energy building designs with renewable energy sources.

The problem here is that architects and engineers in this field rely on one program – *Radiance* – to model daylight throughout the year using archival weather records. There are a number of commercial engineering programs available, but they all provide user interfaces to the *Radiance* lighting simulation engine.

The challenge here is that *Radiance* is too slow to offer users an interactive design tool. The radiosity method has been extended to do everything *Radiance* does for climate-based annual daylight calculations, but two hundred times faster. For anyone looking for a computer graphics challenge beyond computer games, this is an open problem.

Beyond Computer Games

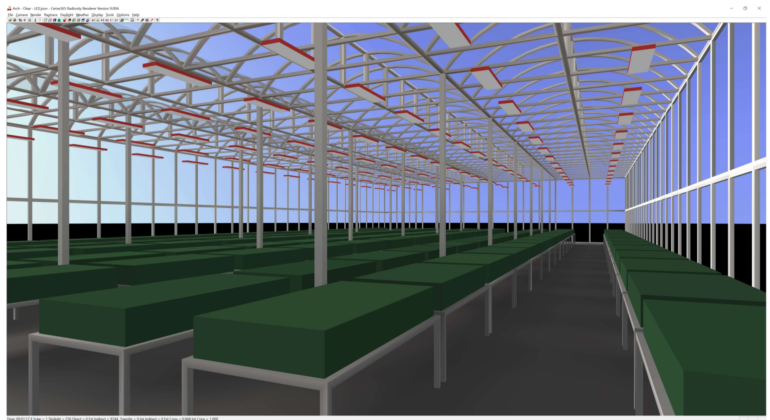
- Predictive Daylight
  - Zero-energy buildings
  - Solar energy modeling
  - Real-time lighting control
  - Weather prediction
  - Thermal modeling

Another open problem in architectural design is predictive daylighting. If you can predict the weather hours to days in advance, you have the opportunity to design real-time building controls that save energy. Electric lighting alone in large buildings can consume 20 to 25 percent of their power, and further energy savings can be realized by controlling the heating, ventilation, and air conditioning (HVAC) systems.

This might seem an unusual career choice for someone with in-depth computer graphics experience, but it remains that this an open problem, and there are likely solutions that involve computer graphics for real-time predictive modeling.

Beyond Computer Games

- Greenhouses
- Climate-based annual daylighting

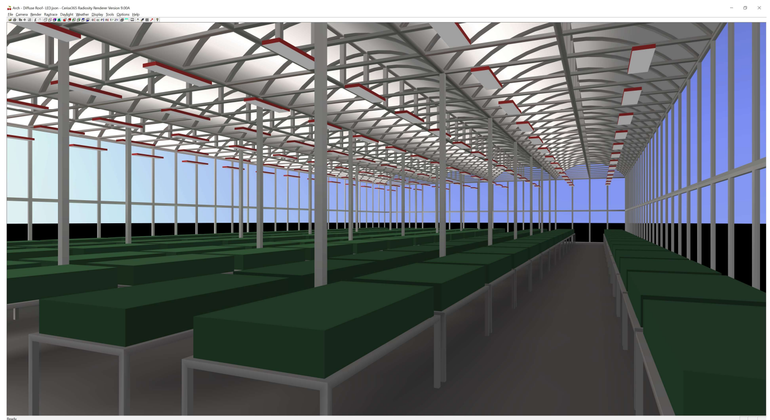


… and from net-zero energy buildings to … greenhouses?

Well, yes. The horticultural industry is a multi-billion industry, and yet it has no design tools whatsoever for electric lighting and daylighting design. One greenhouse design textbook simply says, “The light level in the greenhouse should be adequate and uniform for plant growth.”

Beyond Computer Games

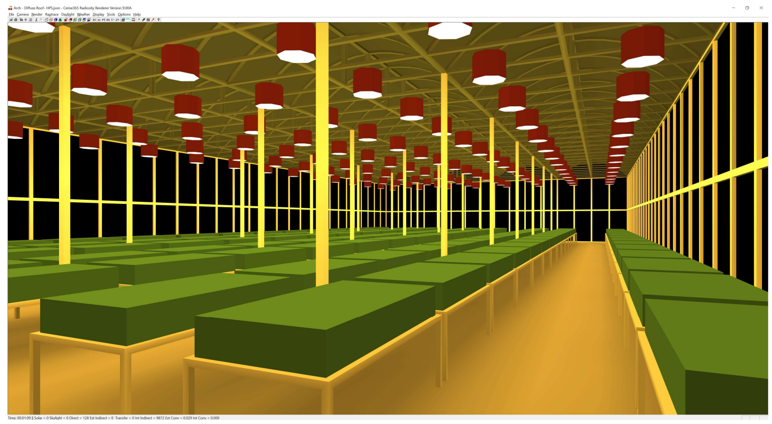
- Greenhouses
- Climate-based annual daylighting



This then is yet another open problem in search of a solution. We have the computer graphics tools for modeling climate-based annual daylighting in greenhouses, but the horticultural industry does not yet have the software tools for greenhouse design in which to implement them.

Beyond Computer Games

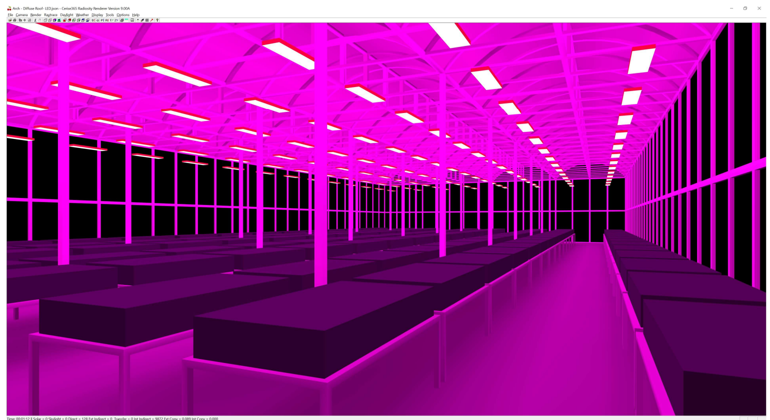
- Horticulture
  - Greenhouses
  - Vertical farms
- Plant biology
  - Photosynthesis
  - Photomorphology
  - Shade avoidance
  - Ultraviolet A/B responses
  - Astronomical light pollution



This problem extends to electric lighting in greenhouses and indoor “vertical farms,” where lettuce and other greens are grown in stacked trays. It may be straightforward to design the lighting for these buildings, but the real problem is that plants respond to light very differently than do humans. To fully understand the problem, the software engineer needs to understand plant biology in considerable detail, including photosynthesis, photomorphology, shade avoidance responses, ultraviolet-A and ultraviolet-B responses, and even astronomical light pollution issues.

Beyond Computer Games

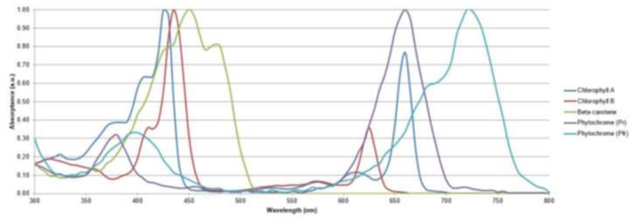
- Horticulture
  - Greenhouses
  - Vertical farms
- Horticultural lighting design
  - Plant spectral requirements
  - Solid-state lighting (LED) technology
  - Horticultural lighting metrics

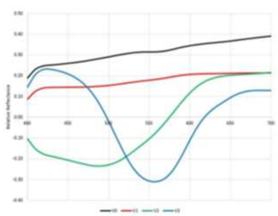


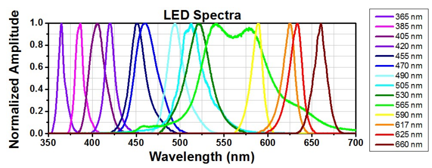
The challenges continue. In addition to understanding plant biology, you also need to understand plant spectral requirements (a topic that is being actively researched, with over one thousand papers to date), solid-state lighting (aka light-emitting diode, or LED) technology, and a variety of horticultural lighting metrics that are still being developed by various standards organizations.

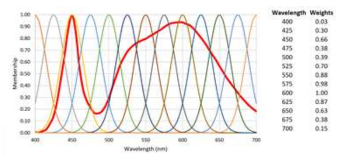
Beyond Computer Games

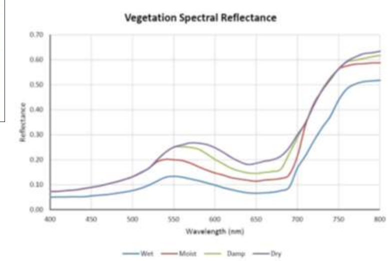
- Horticulture
  - Photopigments
  - Spectral sensors
  - Astrobiology
  - Eigencolors











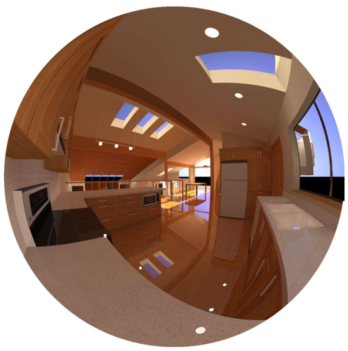
… and yet more challenges. These are images from several papers that I have written over the past few years, all applying to horticultural lighting and all involving computer graphics to varying degrees.

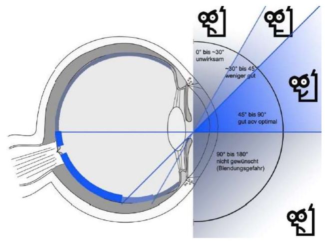
You may not recognize the term "eigencolors," but this is understandable – it has so far appeared only in a patent that is scheduled to be published in the coming weeks. It combines astrobiology, remote satellite imaging, spectral sensors, plant photopigments, and yes, computer graphics techniques to implement virtual spectroradiometers and spectral rendering in virtual environments.

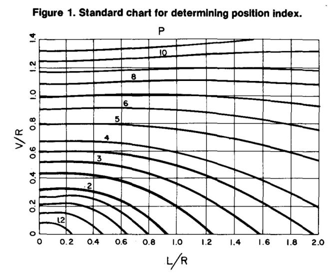
Computer graphics is involved, and indeed is essential, to this sort of work, but it is only the beginning.

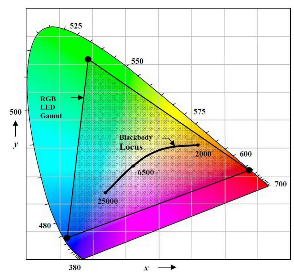
Beyond Computer Games

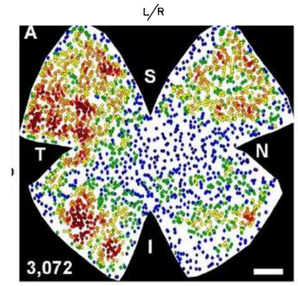
- Chronobiology
  - Circadian rhythms
  - Blue light hazard
  - Human-centric lighting











One more example. Chronobiology is the science of circadian rhythms in animals and plants, including of course humans. These circadian rhythms are synchronized ("entrained") with the daily cycle of light and dark, and are of particular interest to architects and lighting designers working on building designs that promote good health in their occupants.

Developing lighting design tools is a particular challenge here, as it involves everything from lamp spectra and colorimetry to the distribution of intrinsically photosensitive retinal ganglion cells (ipRGCs) in the retina and how the human brain and body responds to them.

Beyond Computer Games

- SunTracker Technologies – past and current projects
  - Acoustic radiosity for concert hall design
  - Predictive daylighting for real-time lighting control
  - Machine learning for weather prediction related to horticultural lighting
  - Importance-driven mesh simplification for radiosity applications
  - Neural network classification of spectral power distributions
  - Distributed Southwell iteration on multicore computers
  - Spectral modeling (eigencolors) for radiosity applications
  - Analytic BTDF and BRDF modeling
  - Radiosity-based VR using WebGL
  - Real-time dynamic lighting using radiosity solutions
  - Light transport using SVD and neural networks

There is, of course, much more. This is a list of some of the projects that we have worked on in my company over the past four years. Some of them are related to specific products under development, but the majority have been speculative … and they have all come from a quarter century-old book on the radiosity method.

Beyond Computer Games

SunTracker Technologies

- There is more – much more – to computer graphics than computer games
- Computer graphics is not a discipline – *it is an engineering tool*
- Many fields need people who understand computer graphics; the challenge is for **<u>you</u>** to learn these fields and apply your knowledge and expertise

To conclude, I need to emphasize that this presentation has focused on the radiosity method mostly because it provides a framework to show what is possible beyond computer games.

Computer graphics should not be viewed as a discipline , let alone as a tool for computer games development. It should instead be viewed as an engineering tool that can be applied in many different disciplines.

Albert Einstein made a comment in 1921 that is as relevant today as it was then: “The value of a college education is not the learning of many facts, but the training of the mind to think.” You have learned about computer graphics; it is now time to begin thinking.

Thank you for your time.