Real-Time Photorealistic Terrain Lighting

Naty Hoffman Kenny Mitchell
Westwood Studios, EA

Overview

- Naty Hoffman
  - Introduction
  - Foundations
  - Outdoor illumination
  - Analytical lighting method
- Kenny Mitchell
  - Video-based lighting method

Westwood STUDIOS

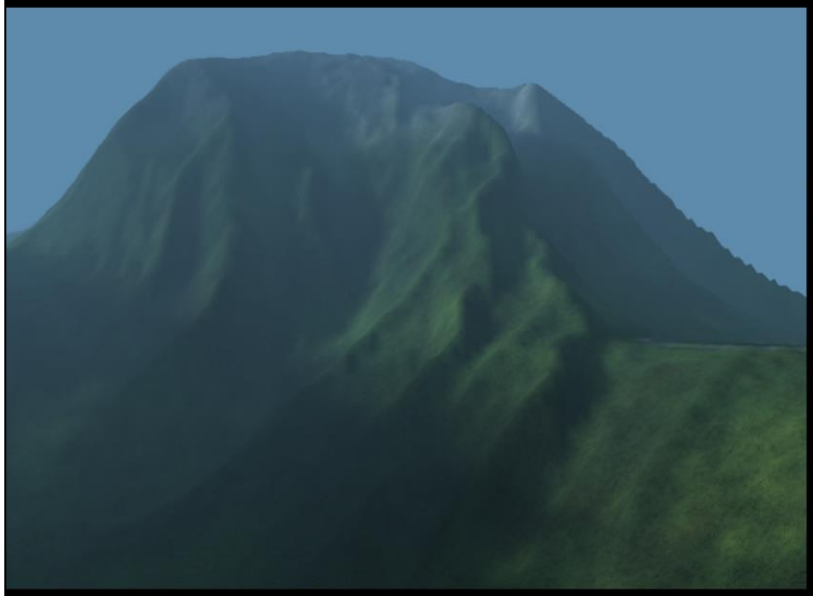
EA ELECTRONIC ARTS

Background & Analytical Lighting Method

Naty Hoffman

Introduction

- Capture the visual richness of outdoor landscapes
  - Many advances have been made in getting the geometric complexity right
  - But the simplistic directional light + ambient model is still mostly used for lighting



Westwood STUDIOS

EA ELECTRONIC ARTS

Foundations

- Radiance (L)

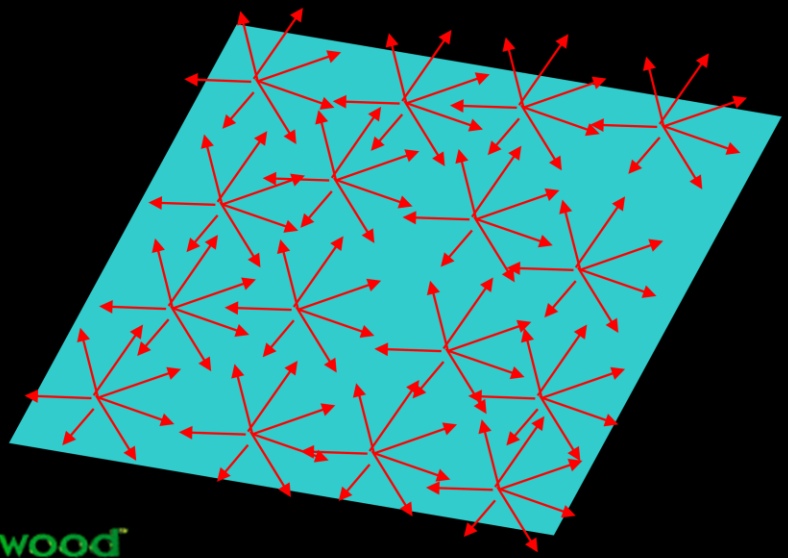
Westwood STUDIOS

ELECTRONIC ARTS

We measure light as *radiance*.

Foundations

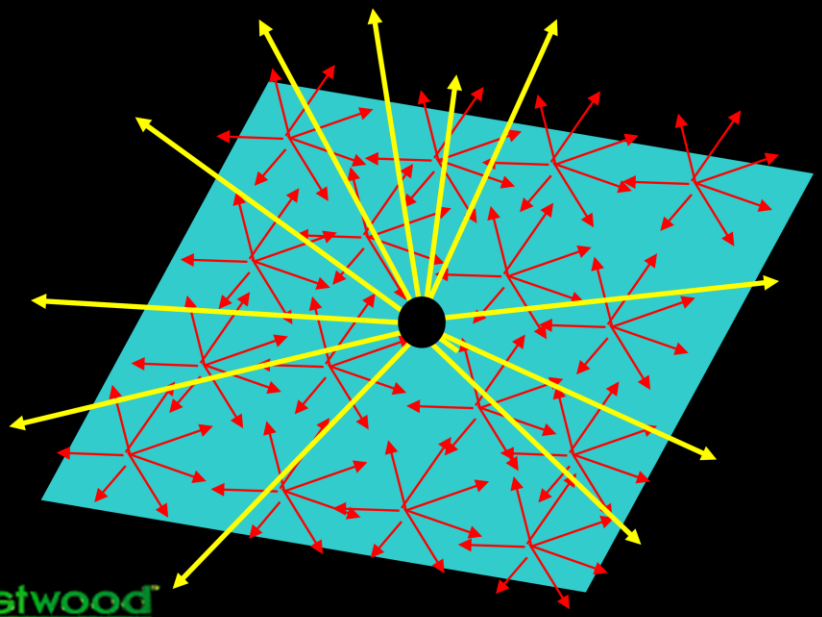
- Radiance (L)
  - Light



To illustrate, look at a light-reflecting surface. Every second a certain amount of energy exits this surface. For a given moment in time, we are interested in the power, or energy per time from the surface, as shown by the red arrows.

Foundations

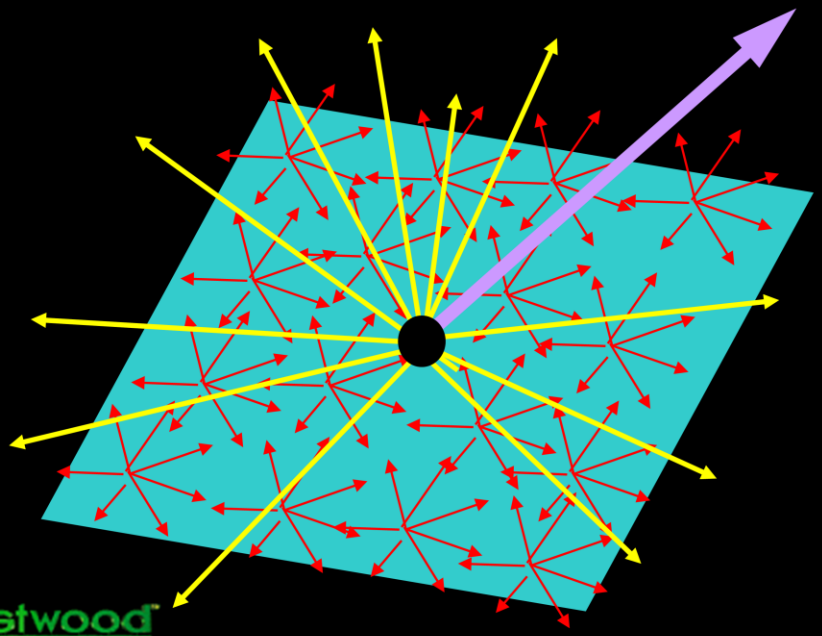
- Radiance (L)
  - Light through a given point



For shading, we care about a specific point on that surface. Since a zero-area point emits no power, we divide emitted power by area to get the power per area from that point, as shown by the yellow arrows.

Foundations

- Radiance (L)
  - Light through a given point in a given direction



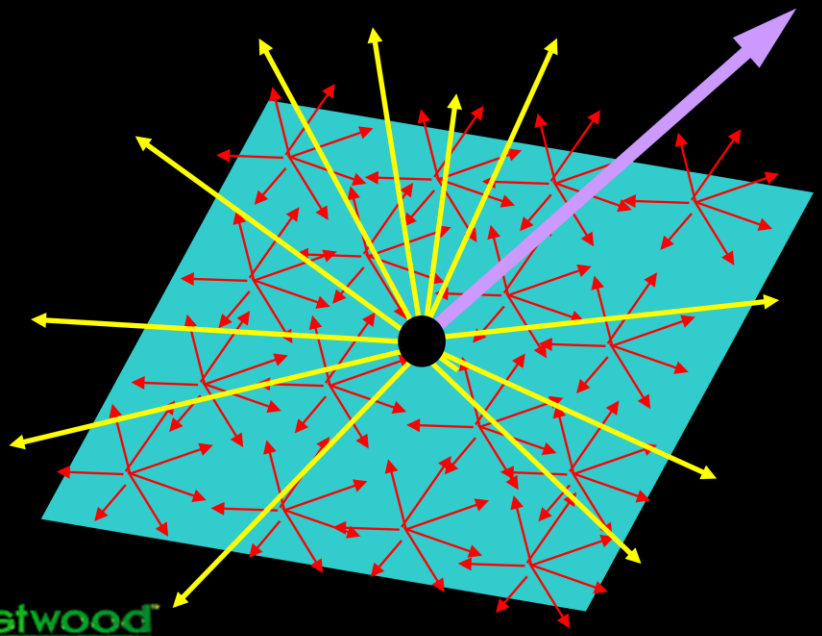
Westwood STUDIOS

ELECTRONIC ARTS

Often, we are not interested in all the light exiting a point, only the part going in a specific direction (for example, toward the eye). So we divide the power per area emitted in a set of outgoing directions by the *solid angle* of that set, to get power per area per solid angle (purple arrow).

Foundations

- Radiance (L)
  - Light through a given point in a given direction



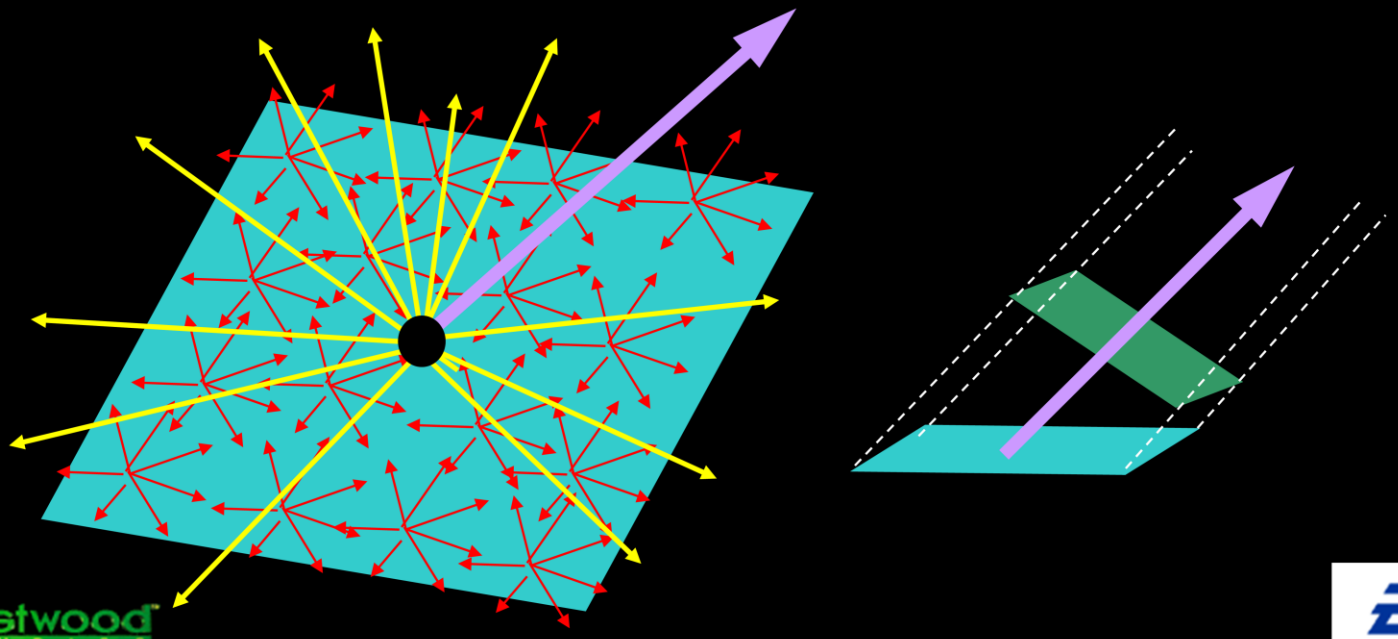
Westwood STUDIOS

ELECTRONIC ARTS

A solid angle is a 3D angle, or set of directions, defined by an area on a sphere. It is measured in *steradians*, of which there are four pi in a sphere.

Foundations

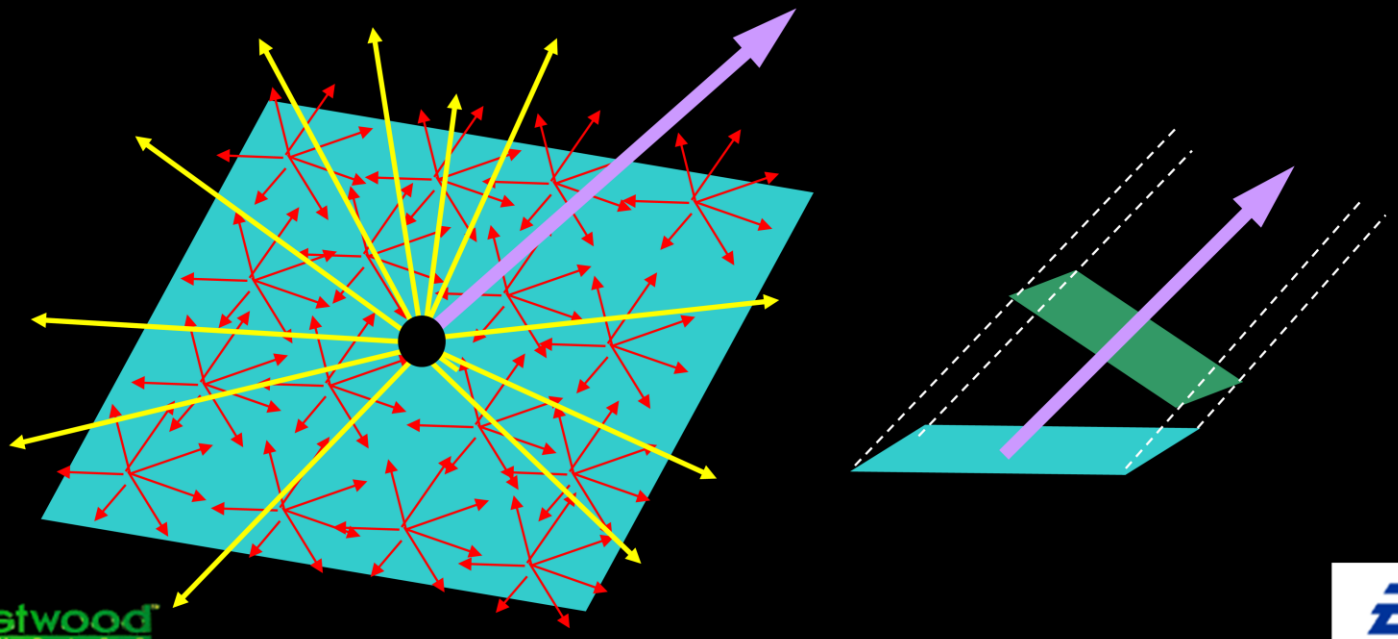
- Radiance (L)
  - Light through a given point in a given direction
  - Power per projected area per solid angle



A final detail about radiance – we do not use area, but *projected area*. When you look at a surface from an oblique angle its area appears to be smaller – this is the projected area (dark green area on the right). It is equal to the area times the cosine of the angle between the surface normal and the direction of projection.

Foundations

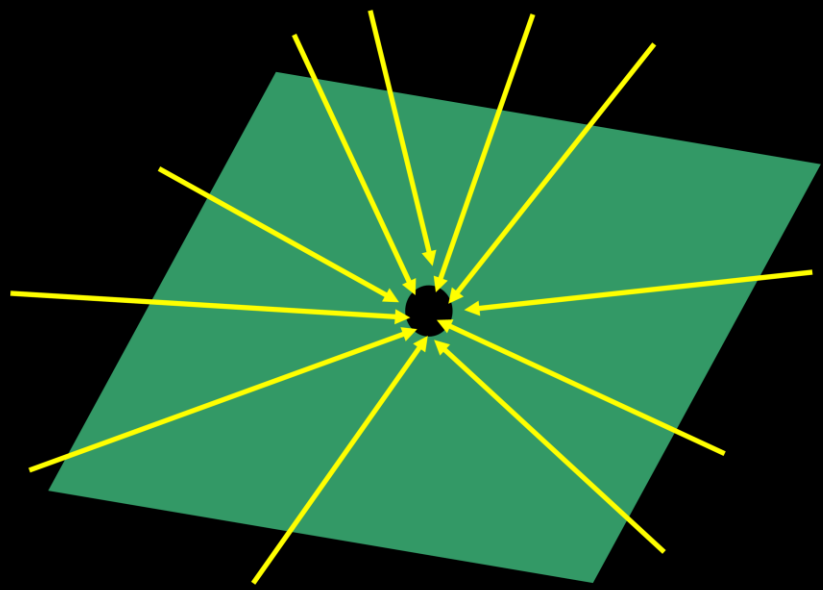
- Radiance (L)
  - Light through a given point in a given direction
  - Power per projected area per solid angle



To sum up, radiance represents the light going through a point in a given direction. Its units are Watts per square meter per steradian. When we render a scene, each pixel's intensity is dependent on the radiance through the corresponding point on the view plane towards the camera.

Foundations

- Irradiance (E)
  - Total light incoming into a surface point
  - Power per area
  - $E = \int L \cos\theta$



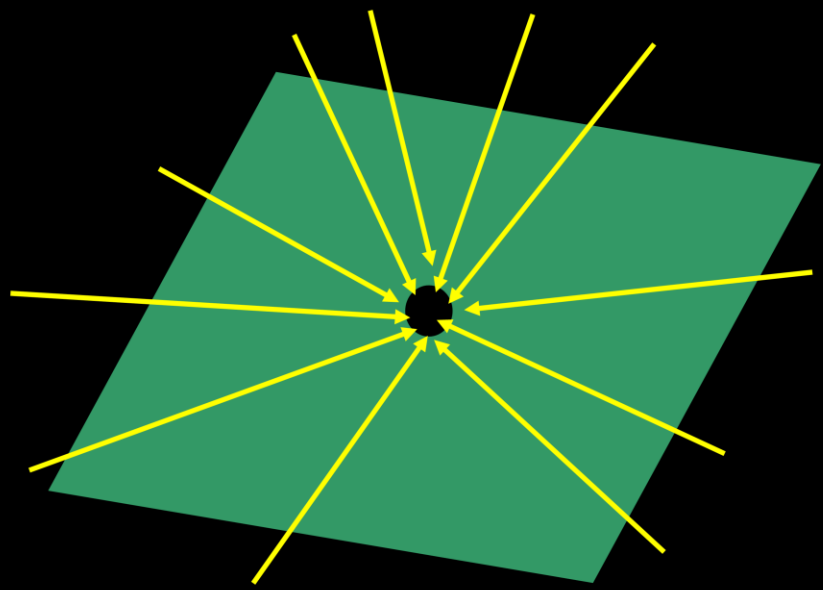
Westwood STUDIOS

ELECTRONIC ARTS

Another quantity of interest is *irradiance,* the total incoming light impinging on a point on a surface, measured as power per area. To calculate it, we total up the incoming radiance from all directions.

Foundations

- Irradiance (E)
  - Total light incoming into a surface point
  - Power per area
  - $E = \int L \cos\theta$



Westwood STUDIOS

ELECTRONIC ARTS

Since radiance is power per projected area per solid angle, and irradiance is power per area, to calculate irradiance we compute the integral of the radiance times the cosine of the angle between the surface normal and the radiance's direction. This integral is computed over all incoming directions in the hemisphere centered on the surface normal.

Foundations

- Lambertian, non-glowing surfaces
  - Outgoing radiance same in all directions and proportional to irradiance: $L = (C / \pi) E$

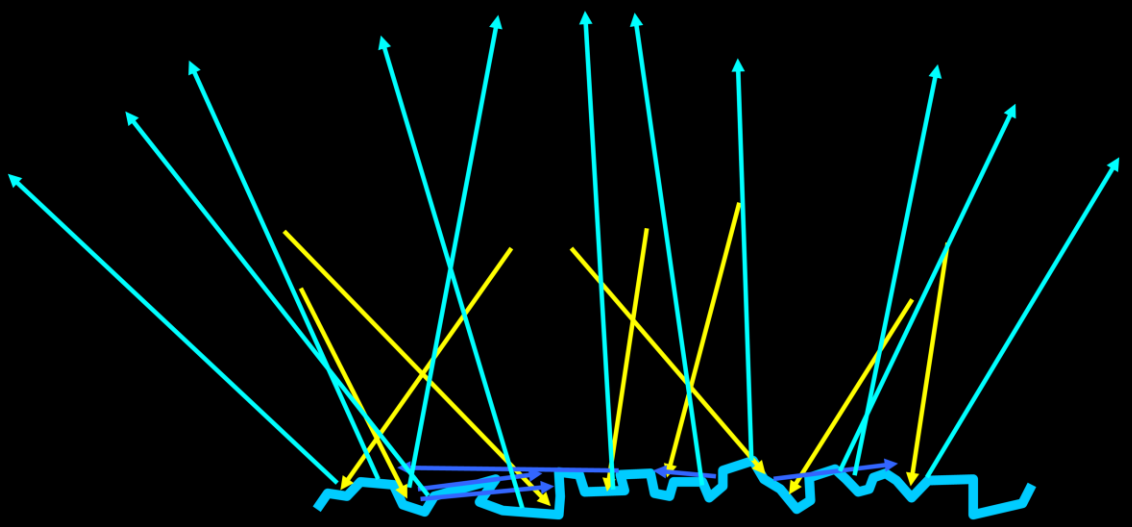
Westwood STUDIOS

ELECTRONIC ARTS

In this talk, we will assume that the surfaces we are seeing are perfectly Lambertian (diffuse), and that they don't glow. The outgoing radiance from such surfaces is the same in all directions, and is proportional to irradiance. The constant of proportionality is the diffuse color divided by pi.

Foundations

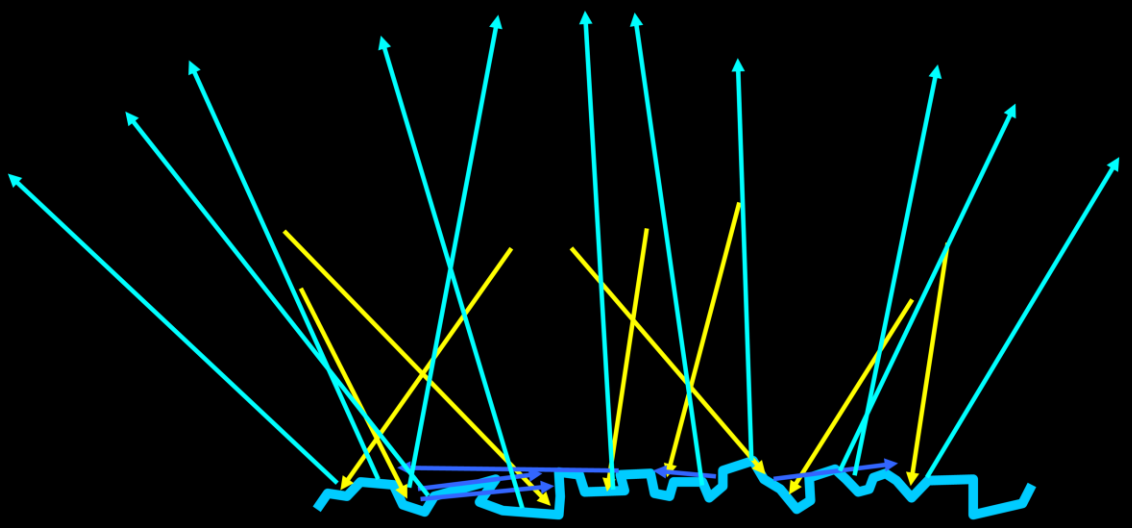
- Lambertian, non-glowing surfaces
  - Outgoing radiance same in all directions and proportional to irradiance: $L = (C / \pi) E$



Physically what is happening here is that the surface is a random collection of little microscopic mirrors pointing in all directions. Incoming light gets bounced about and mixed together, some of it is absorbed, and the remainder is reflected equally in all directions.

Foundations

- Lambertian, non-glowing surfaces
  - Outgoing radiance same in all directions and proportional to irradiance: $L = (C / \pi) E$



Westwood STUDIOS

ELECTRONIC ARTS

The physical definition of diffuse color is the ratio between incoming and outgoing energy. For non-glowing surfaces, this ratio can never be more than one. In graphics, we sample the visible spectrum at three discrete frequencies (red, green and blue) so we have three numbers between zero and one, which is the RGB color we are used to.

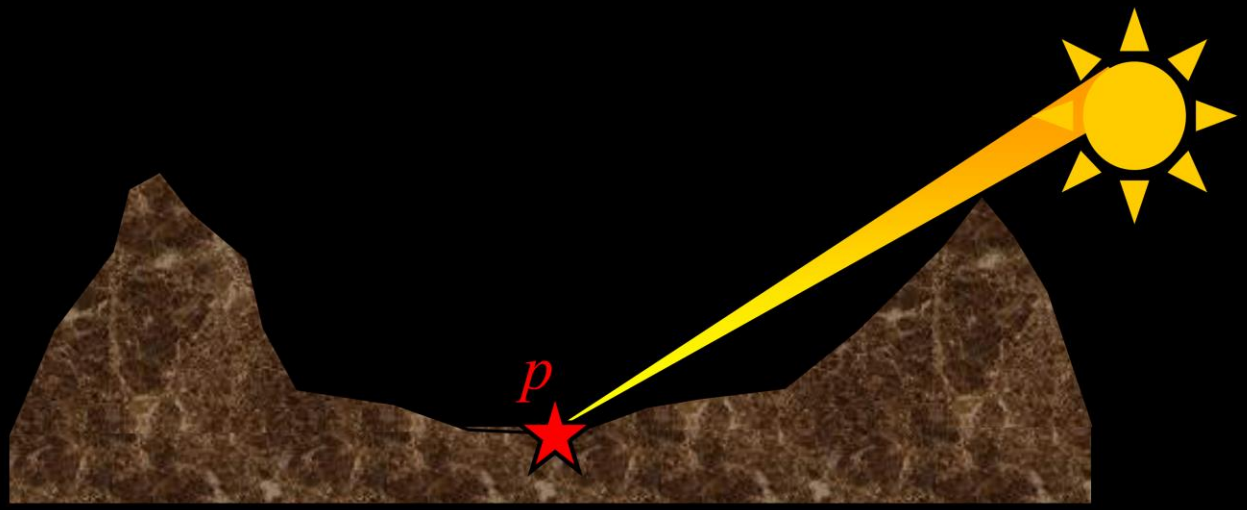
Outdoor Illumination

Westwood STUDIOS

ELECTRONIC ARTS

The outdoors are a fairly complex lighting environment. For the purposes of this lecture, I will be talking about daytime and sunlight, but the same principles apply to nighttime, moonlight, etc. We are looking at a given terrain point p (red star). The outgoing radiance from p depends on incoming radiance from all directions.

Outdoor Illumination

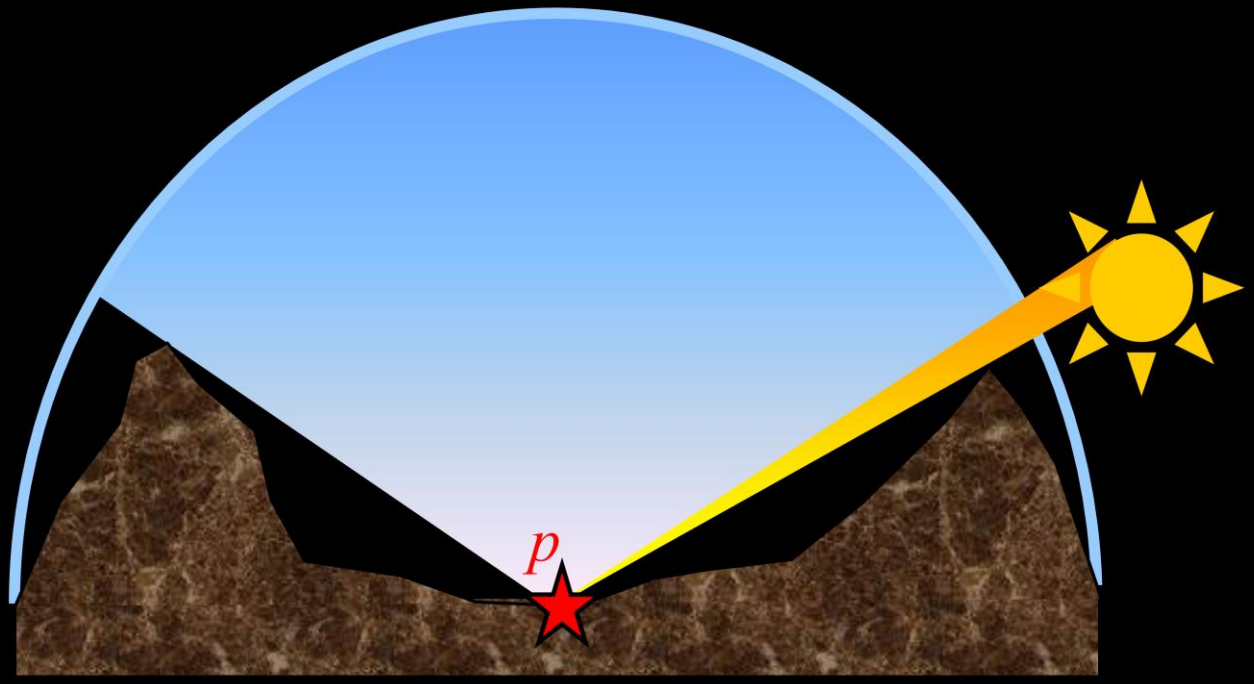


Westwood STUDIOS

ELECTRONIC ARTS

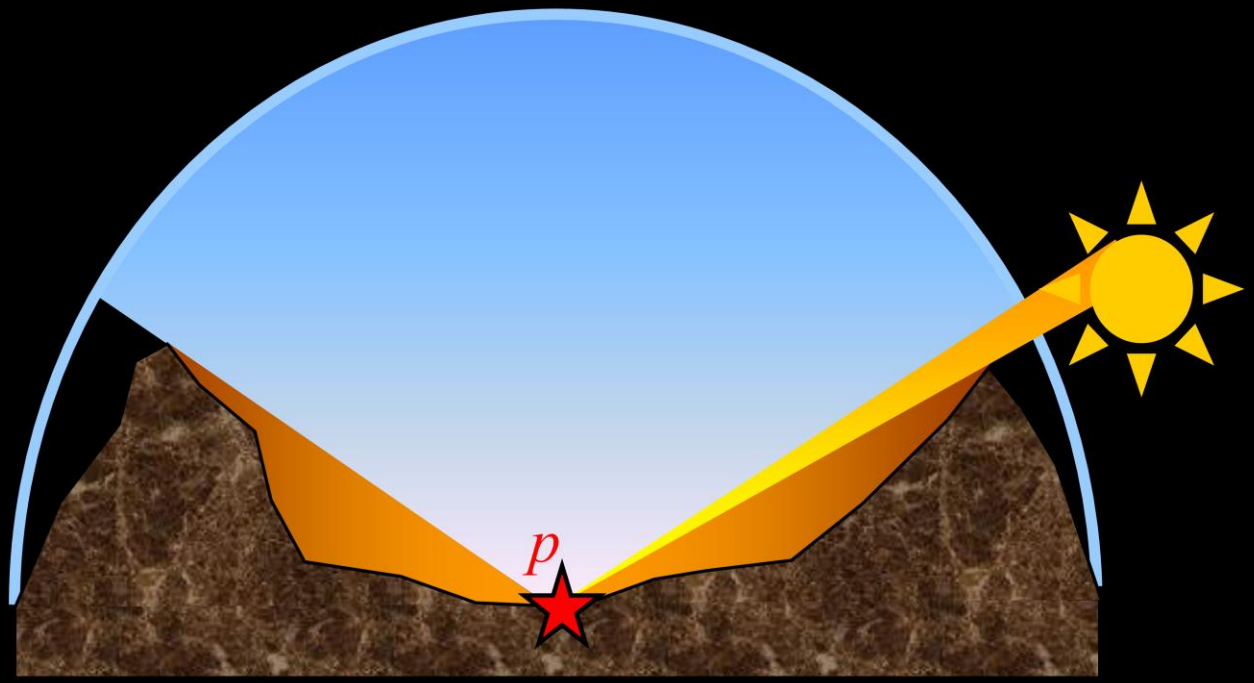
The sun is the most important source – it covers a relatively small solid angle but it has very high radiance, so its total contribution to the irradiance is high. In this example the sun is partially occluded from our point. The sun's radiance and direction both vary over time.

Outdoor Illumination



The sky is the second most important source – it covers a very large solid angle, which makes it quite different from the illuminators we are used to in real-time graphics. Sky radiance varies over time and by direction. Different terrain points can “see” different regions of sky – our example point can “see” quite a bit, but a point in the bottom of a valley would not “see” much.

Outdoor Illumination



Another source of illumination is the terrain itself, which reflects light into our point. This tends to have a relatively small contribution – it is most noticeable in regions of shadow. To fully calculate such interreflections, a global illumination algorithm such as radiosity is needed, however these are very slow. We will achieve similar results in real-time by making approximations.

Analytical Terrain Lighting

- Overview
- Similar Work
- Implementation
- Demo
- Performance
- Future Directions and Conclusion

Westwood STUDIOS

EA ELECTRONIC ARTS

Analytical Method - Overview

- Calculate lightmap at run-time
  - Recalculate, upload as lighting changes
- Separate illumination into sun & sky
  - Interreflections from sky only
  - Combine solutions for final result
- Additional speedups
  - More approximations
  - Use precomputed data
- Software implementation

Westwood STUDIOS

ELECTRONIC ARTS

We would like to include interreflections from the sun too, but this is hard to do fast. In practice, this does not hurt us much since interreflections are most noticeable in areas of shadow, where the sunlight contribution is small. We will show a software implementation first, and then show how hardware can be used to speed it up. Details on the math can be found in the proceedings.

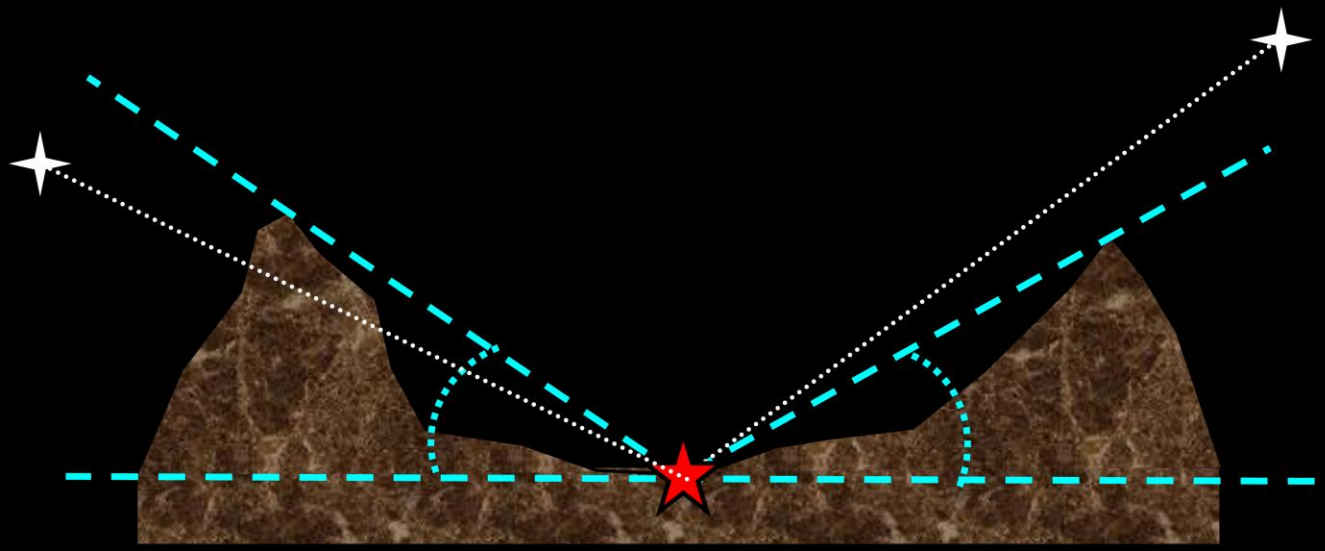
Similar Work

Westwood STUDIOS

ELECTRONIC ARTS

Horizon Maps

- Max, 1988
  - Store horizon angles in eight cardinal directions
  - Find matching angles to light source, compare to determine if in shadow



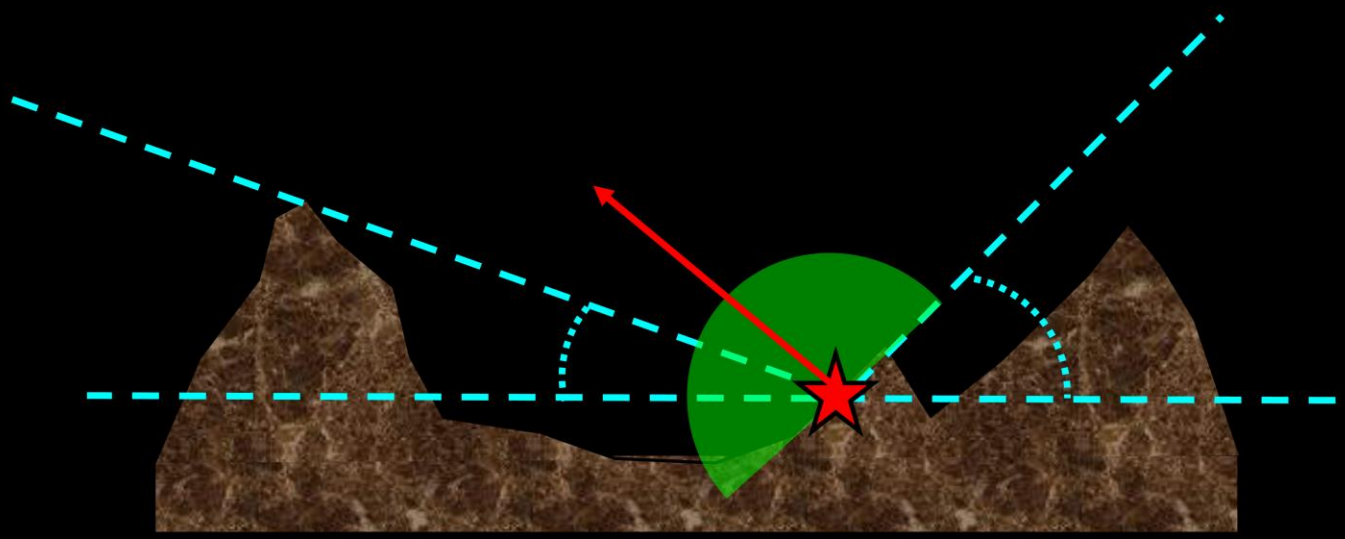
Westwood STUDIOS

ELECTRONIC ARTS

This is useful for shadowing, and can be extended to soft shadows by having a range of angles for the light source. Does not handle sky lighting or interreflections.

Note on Horizon Angles

- It is important to clip the horizon angle to the hemisphere around the surface normal



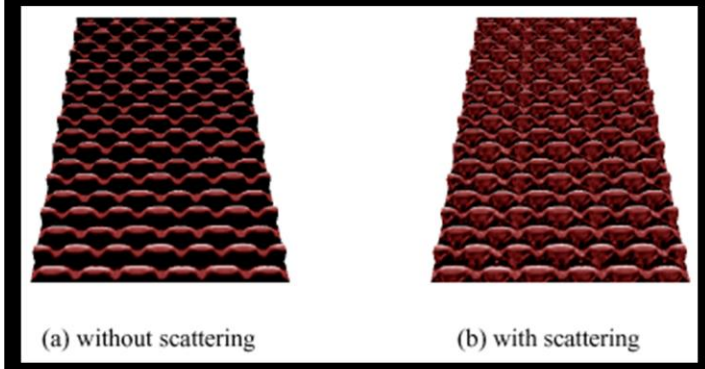
Westwood STUDIOS

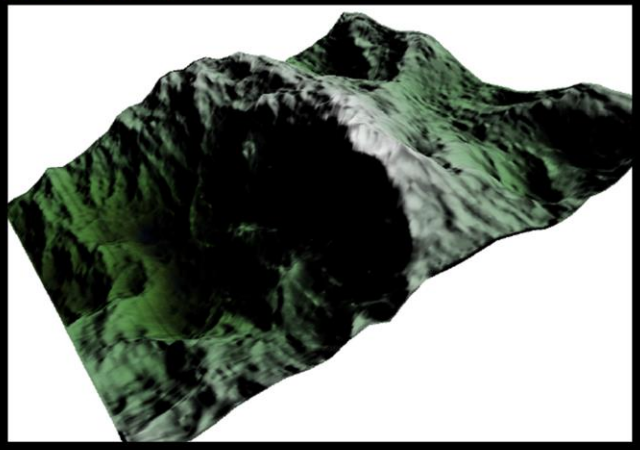
ELECTRONIC ARTS

The red arrow is the surface normal. On the left side, the horizon angle is determined by the occluding terrain but on the right side the hemisphere is unoccluded. In this case, we take the hemisphere boundary on the right side as the effective horizon angle.

Illuminating Micro Geometry

- Heidrich, Daubert, Kautz and Seidel, 2000
  - Precalculated visibility maps in a set of directions
    - From each point on the map, which other point on the map (if any) is seen in that direction
    - Used with multiple iterations to do interreflections
  - Elliptical horizon map
    - Fit the set of visible directions from each point to an ellipse parameterized by 6 numbers
    - Store ellipse parameterizations in two RGB textures
    - Use hardware to compute shadows





Westwood STUDIOS

ELECTRONIC ARTS

The visibility maps can be done in hardware that supports dependent texture reads (like GeForce 3). This handles interreflections, and can possibly handle sky light. It does require many different textures and passes, so is quite expensive. The Horizon ellipse map enables generating shadows with lights in arbitrary directions, in hardware. Does not handle soft shadows.

Interactive Horizon Mapping

- Sloan and Cohen, 2000
  - Use horizon maps in eight directions
  - But use basis maps to smoothly interpolate between horizon maps in hardware

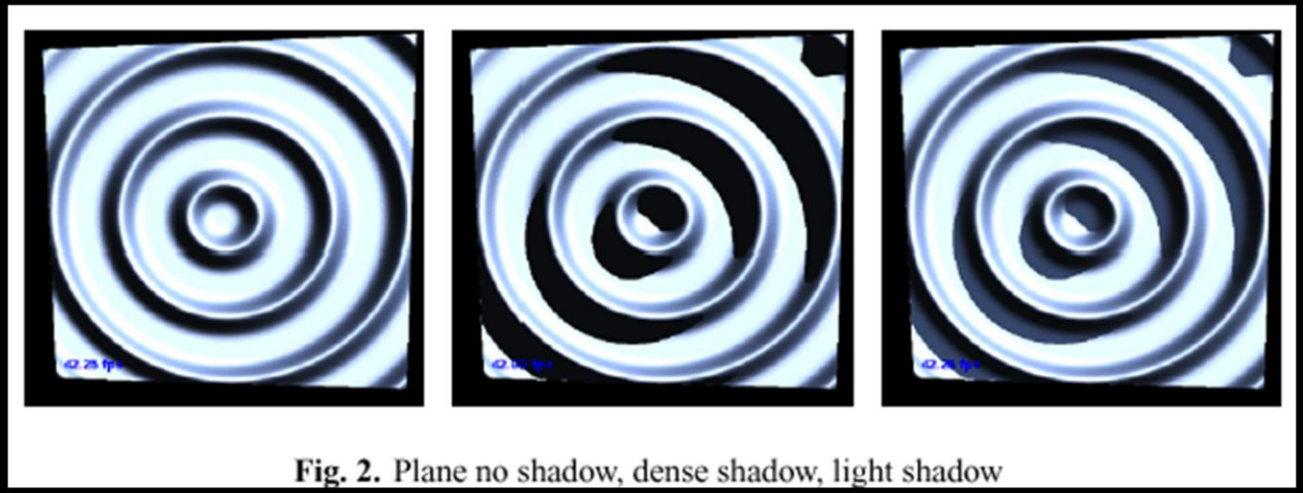


Fig. 2. Plane no shadow, dense shadow, light shadow

Westwood STUDIOS

ELECTRONIC ARTS

Produces shadows in arbitrary directions, in hardware. No soft shadows, sky light, or interreflections.

Recovery of Shape from Shading

- Stewart and Langer, 1997
  - Global illumination approximation for diffuse hemispherical illuminators
  - Assume all other points on the terrain visible from our point have the same radiance

Westwood STUDIOS

EA ELECTRONIC ARTS

Note that the sky is a diffuse hemispherical illuminator! But this work only handled uniform illuminators. This was actually a computer vision paper, not a computer graphics paper; they attacked the opposite problem of generating geometry from an image. The approximation worked for them, which is a good sign that is is accurate.

Recovery of Shape from Shading

- Stewart and Langer, 1997
  - Global illumination approximation for diffuse hemispherical illuminators
  - Assume all other points on the terrain visible from our point have the same radiance
  - Enables closed-form solution

$$Lout_{indirect} = (C / \pi) \int Lin_{indirect} \cos \theta$$

$$Lout_{indirect} = (C / \pi) \int Lout_{indirect} \cos \theta$$

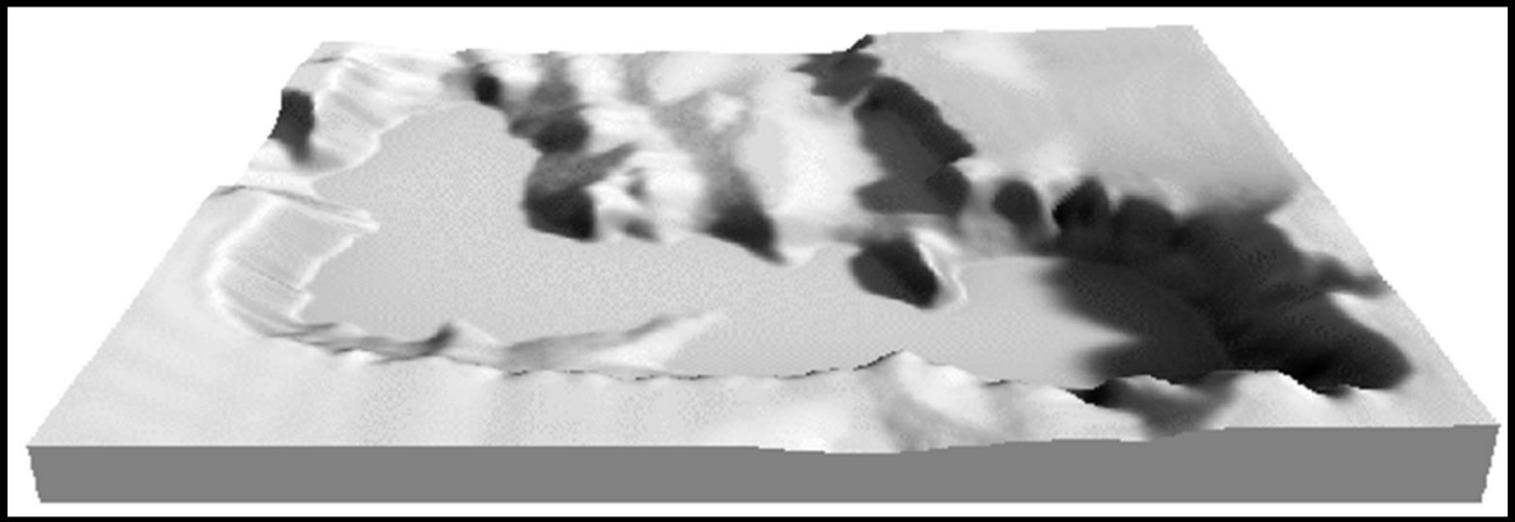
Westwood STUDIOS

ELECTRONIC ARTS

The integral is over all the incoming directions which are occluded by terrain points. Stewart and Langer used a horizon map to parameterize the incoming occluded directions, and worked out the math to calculate the radiance as a function of the horizon angles.

Fast Horizon Computation

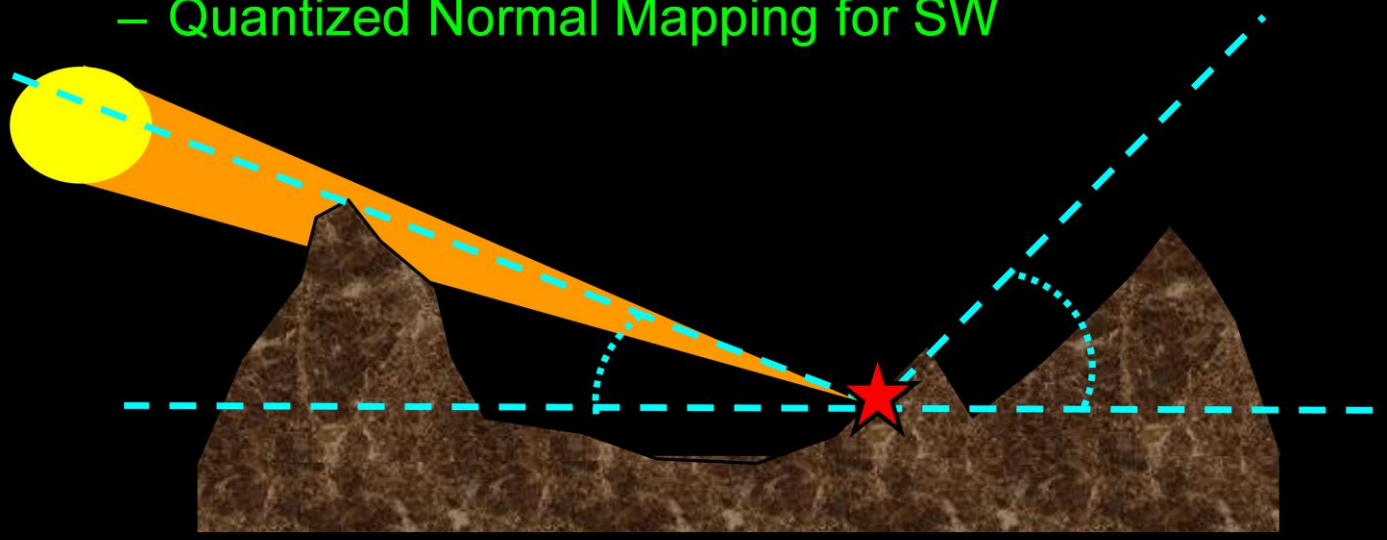
- Stewart, 1998
  - Fast computation of many horizon directions
  - Extends Stewart & Langer's 1997 work to non-uniform sky radiance



This work extended the previous paper for rendering terrains. It was combined with a fast method for calculating horizon angles in many directions for all terrain points at once. It handles shadows (and can be extended to soft shadows), sky lighting, and approximate interreflections from the sky lighting. Non-uniform sky radiance is done by dividing the sky into patches.

Analytical Method - Sunlight

- Assume sun moves in an axis-aligned plane
- Only need one pair of horizon angles
  - Quick to compute, especially since scanning along rows
- Soft shadowing using range of angles
- Normal mapping for lighting
  - Quantized Normal Mapping for SW



Westwood STUDIOS

ELECTRONIC ARTS

Normal mapping is used to calculate the sunlight contribution which is then attenuated by the shadow factor. For the software implementation, we quantize the normals into a table with 256 entries, calculate sunlight for the 256 normals once per frame, and then lookup the result by the quantized normal index. This avoids a dot product and color multiplication per lightmap texel.

Skylight

- Sky only changes radiance, not direction
- Store Skylight contribution (based on 1,1,1 sky radiance) in RGB map
- Could precompute complete global illumination solution
- Need faster computation – use Stewart & Langer's approximation instead
  - Currently uniform radiance – plan to extend to multiple sky patches
  - Calculate other 6 horizon directions approximately (scan terrain to a short distance)

Westwood STUDIOS

ELECTRONIC ARTS

We need all eight horizon angles to calculate the skylight contribution – we already have two for the shadows. We skimp a bit on calculating the other six, since they do need need to be very accurate – we scan the heightfield to a short distance when computing them.

Rendering

- Software Implementation
  - Generates lightmap in software and uploads to card
  - Uses 2nd thread to do so asynchronously

Westwood STUDIOS

ELECTRONIC ARTS

Using the second thread allows us to control how much CPU time we want to give to the lightmap update. Our current settings have this thread sleep quite often, so it takes a relatively small amount of CPU. This means that the lightmap will not update very often, but in our game the outdoor lighting changes slowly over time so that is OK.

Rendering

- Software Implementation
  - Generates lightmap in software and uploads to card
  - Uses 2nd thread to do so asynchronously
  - Combines with color map in software to save a pass at render time

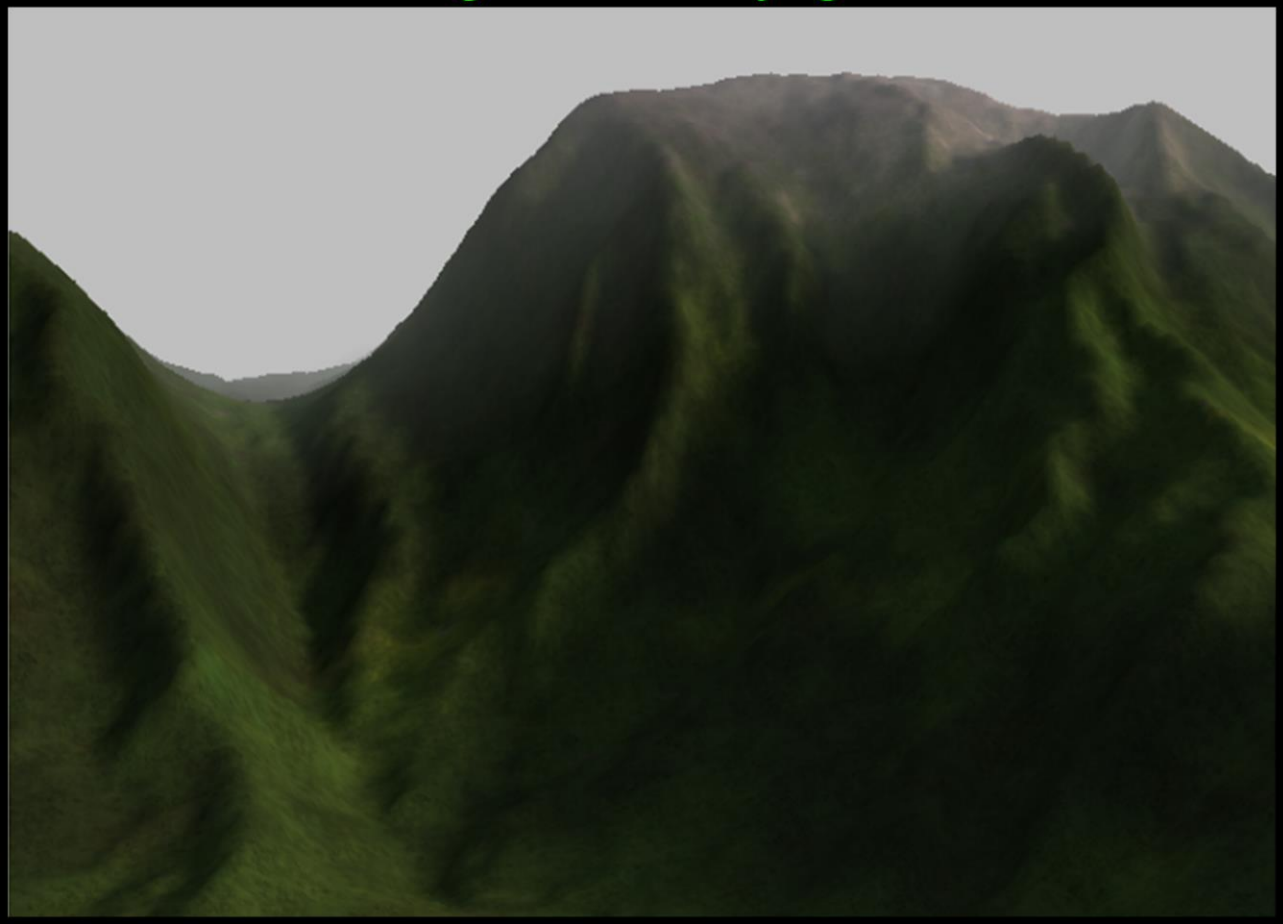
Westwood STUDIOS

ELECTRONIC ARTS

Since we are computing the lightmap in software anyway, and it is the same resolution as our color texture (we use a detail texture in addition), we multiply the two into the lightmap. This means that we can render the terrain with a single two-texture pass.

Demo

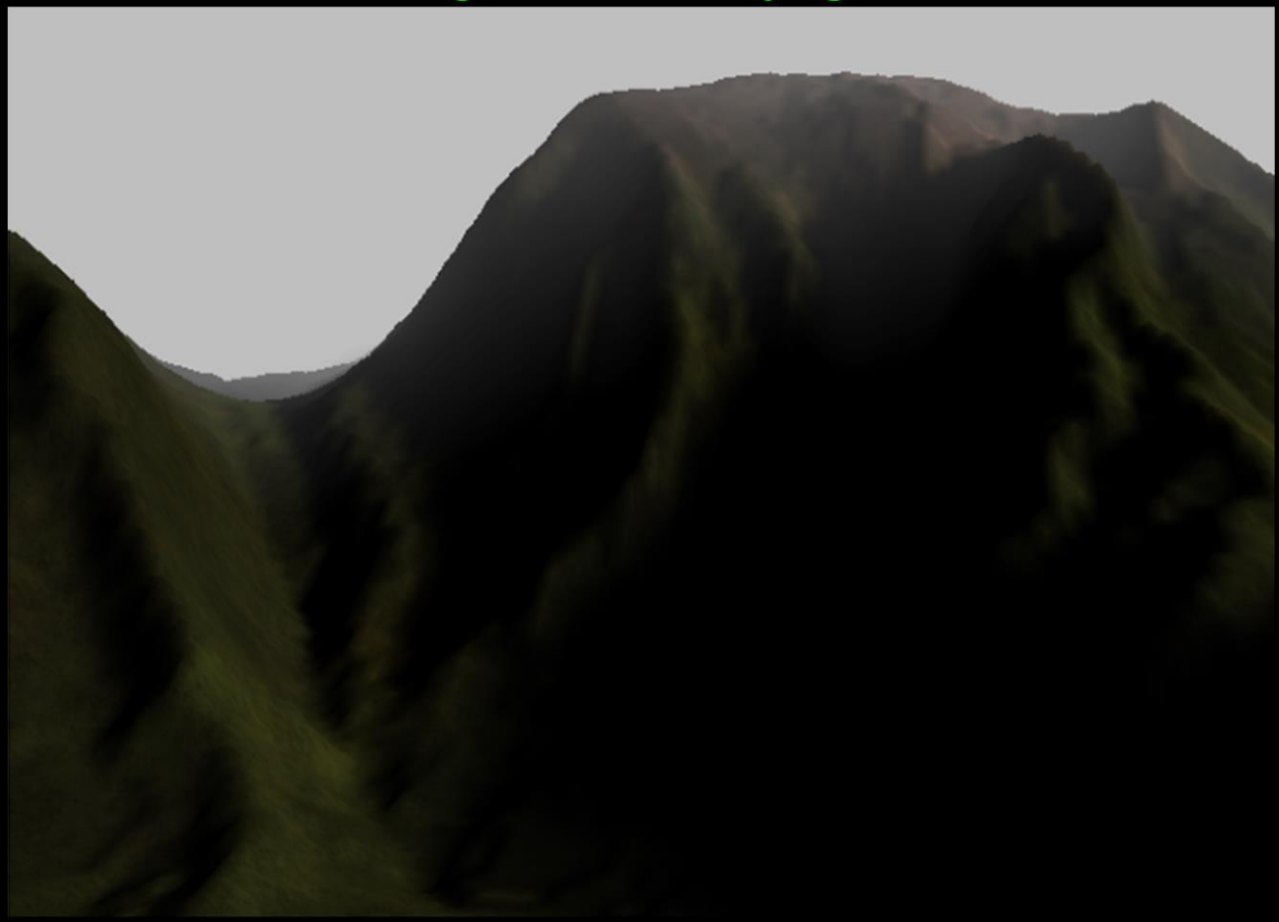
Sunlight and Skylight



Our game updates the lightmap at a relatively low rate (every few seconds) in a background thread. However, for this demo, we have a mode which speeds up time and to show this properly we put the lighting into a blocking mode where it is recalculated every frame, however this does reduce the frame rate significantly. Here we see the terrain lit by the contributions of both sun and sky.

Demo

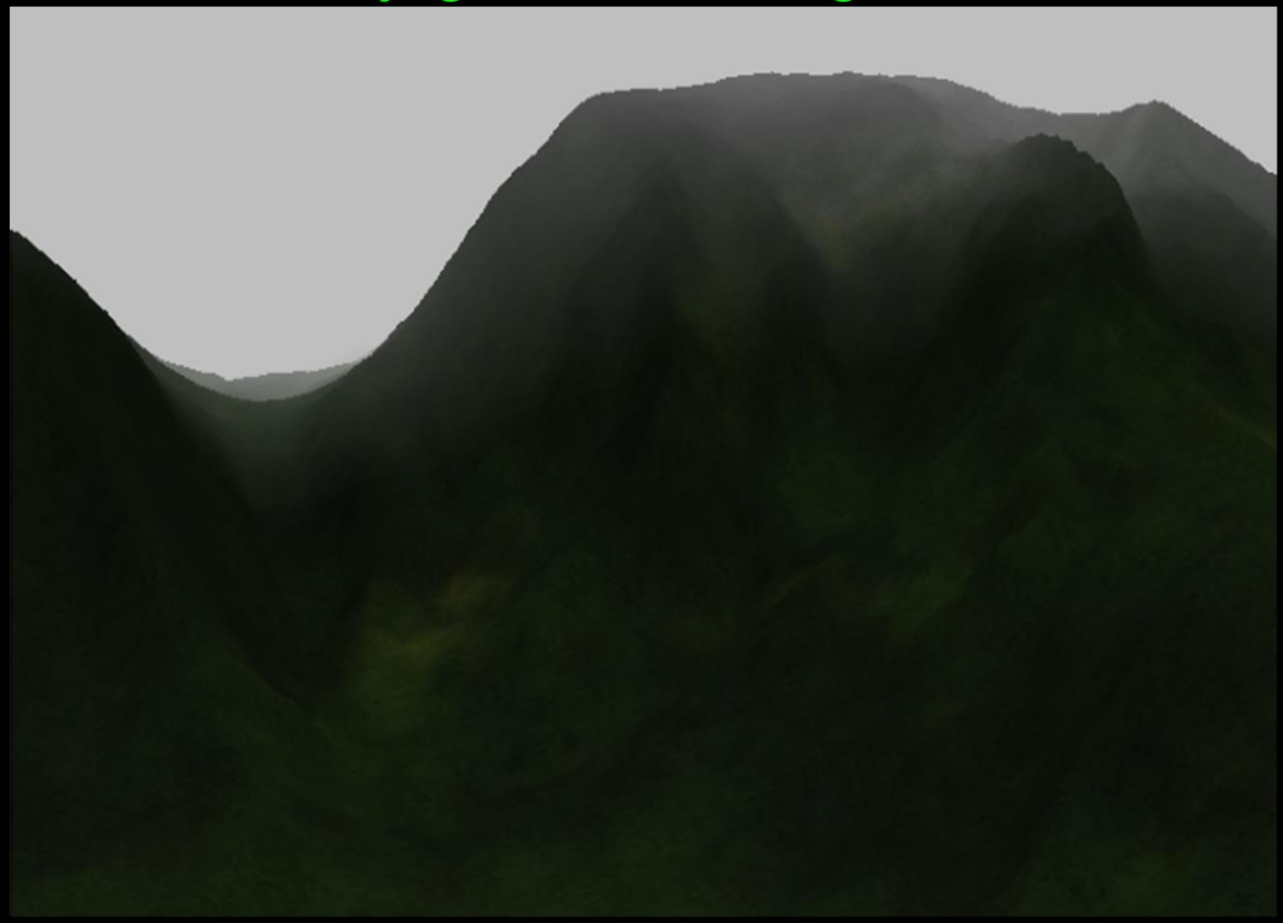
Sunlight - No Skylight



Here we have turned the sky light off – we can see that shadowed areas are now fully dark, and the lighting is less complex and realistic.

Demo

Skylight – No Sunlight



Here the sun is off, leaving only sky light; this corresponds to an overcast day. We can see that sky light produces a complex lighting effect, very different from the highly directional lighting produced by the sun. The lighting of each terrain element is determined by its amount of sky exposure rather than its orientation. This lighting feels correct for an overcast day.

Performance

- Multithreading allows the impact of the lightmap generation to be arbitrarily low.
  - Uploading still causes ‘hiccups’, we plan to amortize uploads over multiple frames to avoid this.
  - The gating factor is the speed at which the lighting changes in the game
- There is upside for future optimizations

Westwood STUDIOS

ELECTRONIC ARTS

And the game runs even better on multiprocessor machines. If lighting changes quickly and we do not give enough CPU to the second thread, the lighting will ‘pop’. The current implementation is in C++; the computations are a good fit for MMX and we plan to port to MMX assembly, which should yield large speedups. In future we could use the 128-bit integer SIMD in SSE 2.

Future Directions

- Hardware Lightmap Generation
  - Normal mapping, skylight and sharp shadows straightforward
  - Soft shadows may be possible using pixel shaders
- Cloud shadows
  - Can modulate the sun contribution with a cloud map.
  - Does not take more complex cloud effects into account – see next method!
- Interreflections from sunlight
  - Not sure how to do this yet – maybe start with 2000 paper by Heidrich et. al.

Westwood STUDIOS

ELECTRONIC ARTS

The normal mapping can be trivially done using dot-product3 blending. Sharp shadows could be done by using alpha tests. The skylight factors can be simply multiplied by the current sky color and added in. Soft shadows require a few simple math operations, so hopefully they could be computed using register combiners or pixel shaders – we plan to look into this.

Conclusions

- Tradeoffs
  - + Soft shadows, skylight, some interreflections
  - + Fast precomputation
  - + Fast runtime, low runtime memory footprint
  - – No sun interreflections, complex cloud effects
  - – Large texture uploads
- Future
  - – Speed up software algorithm
  - – Hardware implementation
  - – More features

Westwood STUDIOS

ELECTRONIC ARTS

Precomputation takes from six seconds on a PII-450, depending on how far we scan for horizon angles. It is fast enough so we could handle infrequent local terrain deformations in runtime if we wanted – we would need to only scan the rectangle affected by the change and recompute the precalculated data.

Video Based Terrain Lighting

Kenny Mitchell

Westwood STUDIOS

ELECTRONIC ARTS

Video Based Terrain Lighting

- Background
- Hardware Implementation
  - Playstation2
  - PC
- Further Directions

Westwood STUDIOS

ELECTRONIC ARTS

Background

- Video Based Illumination Maps
  - Image based lighting
  - Video textures

Westwood STUDIOS

ELECTRONIC ARTS

Image Based Lighting

- Uses images applied to geometry to yield realistic appearance
- Images can be
  - Synthetic
  - Photographs

Blinn & Newell 1976

Images from Paul Debevec's Story of Reflection Mapping
http://graphics3.isi.edu/~debevec/ReflectionMapping/

Miller & Perlin 1982

Chou & Williams 1982

Haeberli & Segal 1993

Westwood STUDIOS

ELECTRONIC ARTS

Video Textures

Schödl, Szeliski, Salesin, Essa SIGGRAPH 2000

- Uses video applied to textured geometry
- Produce continuous sequences
  - Search video for natural loops
  - Apply blended transitions

Westwood STUDIOS

ELECTRONIC ARTS

Image Based Terrain Lighting

- Apply image to height field
- Real image
  - Sample light on terrain
  - Aerial Photography
- Synthetic image
  - Terrain rendering package
    - Bryce
    - Terragen

Westwood STUDIOS

ELECTRONIC ARTS

Video Based Terrain Lighting

- Video based illumination map
  - Animated sequence of terrain light images
  - High level of realism
    - Use real light
    - Advanced pre-rendered lighting
  - Accelerated by video playback hardware
  - Combine with animated sky texture
    - Demo Video

Westwood STUDIOS

ELECTRONIC ARTS

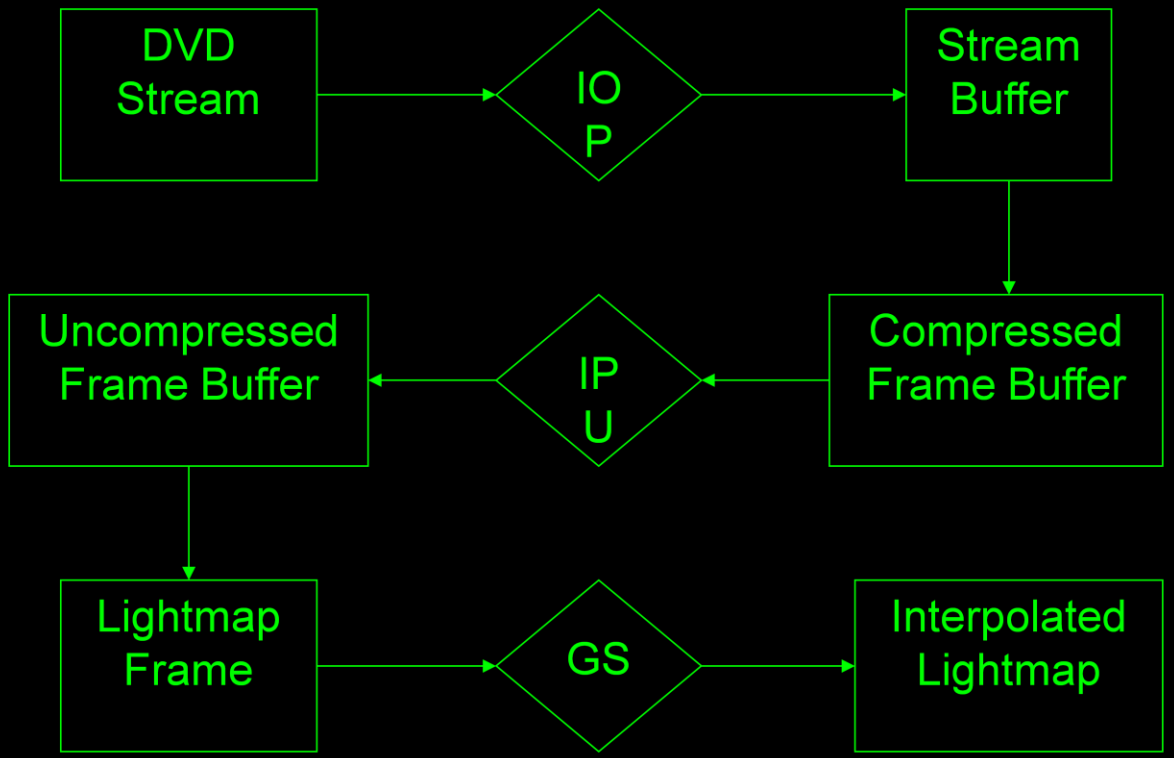
Video Demo

Westwood STUDIOS

ELECTRONIC ARTS

Playstation 2

- Hardware Implementation



Westwood STUDIOS

ELECTRONIC ARTS

Playstation 2

- Frame interpolation
  - Provides temporal anti-aliasing (motion blur)
  - Blends transitions when looping or branching video stream
  - Smoothes over hitches in animation
- Demo

Westwood STUDIOS

ELECTRONIC ARTS

PC

- Alternatives
  - DirectShow
  - Compressed textures
- Demos
  - ATI Radeon
  - NVIDIA GeForce3

Westwood STUDIOS

ELECTRONIC ARTS

PC Demo

Westwood STUDIOS

ELECTRONIC ARTS

Conclusions

- Trade-offs
  - + Unlimited lighting complexity
  - + Hardware accelerated
  - - Video generation can be time consuming
- Future
  - – HW abstraction for video textures with frame interpolation
  - – Better quality/performance terrain rendering tools

Westwood STUDIOS

ELECTRONIC ARTS

References

N.L. Max. *Horizon Mapping: Shadows for Bump-Mapped Surfaces.* The Visual Computer, 4(2):109-117, July 1988.

A.J. Stewart., M.S. Langer. *Towards Accurate Recovery of Shape from Shading under Diffuse Lighting.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(9):1020-1025, Sept. 1997

A.J. Stewart. *Fast horizon computation at all points of a terrain with visibility and shading applications*. IEEE Transactions on Visualization and Computer Graphics, 4(1):82--93, March 1998

W. Heidrich, K. Daubert, J. Kautz, and H.-P. Seidel. *Illuminating micro geometry based on precomputed visibility*. Computer Graphics(Proceedings of SIGGRAPH 2000):455-464, July 2000

P.-P. Sloan, M.F. Cohen. *Interactive Horizon Mapping.* Rendering Techniques 2000(Proceedings of Eurographics Rendering Workshop 2000): 281-298, June 2000

Westwood STUDIOS

ELECTRONIC ARTS

Thanks

- NVidia for GeForce 3 card
- ATI for PC with Radeon
- Hector Yee for analytical method demo

Westwood STUDIOS

ELECTRONIC ARTS