The Problem

We want to put a CG object in this room

- Temporally varying lighting
- Glossy surfaces
- Mirror surfaces
- Diffuse surfaces
- Spatially varying lighting
- Good luck getting occlusion right for the rug and the plant!
- Are we allowed to put light probes on the table? ...how about a laser scanner?

Okay, now do it all in real time!
Background: The Rendering Equation

- Describes how light travels inside a scene
- Doesn’t handle subsurface scattering, phosphorescence, and fluorescence
- Still the foundation of almost all photorealistic rendering algorithms
  - Think ray tracing, radiosity, path tracing, photon mapping...
  - Without this, your CG movies wouldn’t look anywhere near as nice.
- Often solved using Monte Carlo techniques

*Finding Dory* was primarily rendered with path tracing!
**Rendering Equation: The Long Version**

\[ L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\omega_i \in \Omega^+} L_i(x, -\omega_i) f_r(x, \omega_i, \omega_o)(\omega_i \cdot n) d\omega_i \]

- \( L_e(x, \omega_o) \) is the radiance at surface point \( x \) in direction \( \omega_o \).
- \( \Omega^+ \) is the + hemisphere of all incoming directions to \( x \). \( -\omega_i \) is one of those directions.
- \( L_i(x, -\omega_i) \) represents the amount of light emitted from material at point \( x \) (zero unless object is light source).
- \( f_r(x, \omega_i, \omega_o) \) represents the BRDF term.
- \((\omega_i \cdot n)\) accounts for diminished intensity per unit area as a function of angle to the normal.
Rendering Equation: The Short Version

Output radiance in this direction = Emitted radiance + Incoming radiance from all directions that is reflected according to...

\[ \int \left( \text{the BDRF} \right) \, d\omega \]

and the surface normal
Modifying the Rendering Equation

• Many ways to separate the parts of the rendering equation for either ease of understanding or more efficient rendering

• Paper separates the indirect illumination into the real and virtual components
  – You need to render the local (CG model) scene twice in order to determine how the original image will be affected by the virtual object (think shadows, caustics...)

Note that the “+” here is a simplification. The actual technique uses masks and is more of a replace operation.
Light Probes and IBL

- How do we make the CG object look like a part of the original scene?
  - Need to render it with lighting that looks as similar as possible to the scene

- How do we measure the original lighting? Light probe(s)!
  - Take HDR photos of a mirrored (specular) and/or matte (diffuse) sphere to capture the incident lighting at the location where the sphere is placed
    - This assumes all light is at infinity!
  - More light probes = more information about spatially varying lighting

- IBL: Use those pictures as an environment map around your CG objects when rendering
  - Use the map to emit light rays and check color after bouncing off an object.
Light Probe Variations

• Temporally varying
  – Pro: You can capture light that changes over time!
  – Con: More difficult set up and processing

• Dense (spatially varying)
  – Pro: Decently robust and accurate
  – Con: Getting all those samples is invasive, and may not even be feasible for every scene

• Sparse (spatially varying)
  – Pro: About the same physical and perceptual accuracy as dense capture, but much less difficulty involved in getting samples – fast and inexpensive set up compared to dense
  – Con: ...At the cost of processing difficulty and robustness
    • Assuming Lambertian objects = not so great for scenes with lots of high-specular surfaces

This is what a Plenopter looks like, for the curious
Temporal IBL

• How is it done?
  – No standard set up; usually some video camera + light probes combo

• When would you use this?
  – Any environment where lighting changes over time
  – Example: inserting a CG object into augmented reality

An AR application using temporal IBL from Calian et al. (2013)
Dense Capture

• How is it done?
  – Take lots and lots of radiance samples (HDR images) at small intervals (in a 1D path, 2D box, or 3D cube) and interpolate between them when doing IBL.
  – (Does not use scene geometry)

• When would you use this?
  – You have an environment with spatially varying lighting where you can set up capture equipment
  – You don’t want to or can’t capture models of the scene (like if there are glass objects)

A good example of light varying across a 1D path from Unger et al. (2007)
Sparse Capture

• How is it done?
  – Use only a few light probes, but supplement with assumptions about the scene (lambertian surfaces, rough scene geometry)

• When would you use this?
  – Faster, easier capture = good for scenes that change often or where you can’t use invasive capture equipment (Film or stage sets, cultural or archeologically important sites, etc.)

ILM moved to sparse capture with *Iron Man 2* (2010)
Left: HDR capture of the set. Right: Adding the CGI. Note the reference spheres in the corner!
Explicit Geometry

• More complex scene requires more complex measurements
• How it’s done:
  – Get a detailed geometric model of the scene
    • Can do by modelling it yourself, by using a laser scanner, by using computer vision techniques...
    • All the fun things we’ve discussed in previous weeks!
  – Capture the lighting – can vary between a few HDR environment maps, or may be done as part of the same process as getting the model
  – Project the lighting info from the maps back onto the geometry
• Pros:
  – Difficult spatially varying effects (sharp shadows, light shafts, parallax) are automatic!
  – Robust to different types of scenes – indoor/outdoor, spatially varying, specular surfaces...
  – The images produced using this technique are the most physically and perceptually accurate of those covered in the paper!
• Cons:
  – Remember how dense capture was difficult to capture and process? This can be even worse.

More renders of the Parthenon from Devebec et al. (2004)
There’s also this cool video!
Explicit Geometry

• When would you use this?
  – When you want the most photorealistic result possible
  – When you already have a model of the environment or set, or have a scanner available
    • 3D scanning (or use of multiple photos) to create virtual objects is common both in films and videogames

Filming in 48 fps stereo makes inaccuracies more noticeable.
As a result, Weta Digital used LIDAR scans and light probes on every single set for *The Hobbit: An Unexpected Journey* (2012).
Estimated Lighting Conditions

- Sometimes, measuring the original lighting with light probes just isn’t possible
  - For example, when modifying old images or videos
- These are techniques that make the best of what little information we do have
  - Be it estimating the existing lighting from non-light probe objects
  - Or taking advantage of quirks of human vision to get physically inaccurate but perceptually plausible results.
    - The eye and brain care more about certain errors than others.

Don’t have an environment map? Fake it!
Left: Khan et al. Perceptual technique.
Right: Devebec et al. IBL.
Estimated Lighting Variations

• Overall
  – Pros: Absolutely minimal set up (all you need is an LDR image); can be more perceptually plausible than IBL, with none of the hassle of light probes
  – Cons: Can be less reliable than more physically accurate methods; some methods produce illumination estimates that are hard to edit
  – If your application is meant to estimate physics of light transport, you’re out of luck

• Implicit light probes
  – Pro: Output is an environment map, so it can be used with other IBL and light probe techniques; some variants can also be done in real time!
  – Cons: often requires assumptions about the scene (like about scene geometry, or Lambertian surfaces); still not as good as a normal light probe

• Outdoor Illumination
  – Pro: Specialized for outdoor scenes, so it produces good results for them
  – Con: Specialized for outdoor scenes, so it can’t do anything else
    • And even outdoors, it still isn’t as accurate as measured lighting conditions

Implicit Light Probes

• How is it done?
  – Take known objects (faces, eyeballs or scenes with measured depth) and use that known information to help calculate illumination an environment map

• When would you use this?
  – You want to use IBL and/or environment maps, but regular light probes are not an option
    • Example: AR, where the lighting may be changing frequently and the user won’t have a light probe, but you can use their face / eyes or RGB-D data
  – Editing legacy photos and still images

Replacing faces in *Roman Holiday*, again from Nishino et al. (2004)
Implicit Light Probes in Real Time

- Implicit light probes can be used to estimate lighting in real time (including temporal variation!)
  - Survey mentions two papers that do this: Knorr et al. (2014) and Gruber et al. (2014)
- Both papers use Radiance Transfer to speed up rendering and spherical harmonics to do the lighting estimation
- Both demonstrate their algorithms with AR applications
- Otherwise very different
  - Knorr et al. estimates lighting by learning what different lighting conditions on a face look like
  - Gruber et al. uses RGB-D data from a Kinect to update knowledge of the scene geometry

**Using human faces to estimate lighting for AR** in Knorr et al (2014)

Estimated Lighting Variations
Outdoor Illumination

• How is it done?
  – Estimate the light coming from the sky, including the position of the sun, to make an environment map (and directional light for the sun)

• When would you use this?
  – You want to add an object to an outdoor scene and you don’t have a way to measure the light

Lalonde et al. (2009) suggest architects could use their technique to show off a planned building in a variety of environments

Estimated Lighting Variations
Interactive Differential Rendering

- Not the same thing as additive differential rendering
  - Additive: Render twice and composite the new object in the scene (what we talked about at the start)
  - Interactive: One render pass

- How it’s done:
  - Modifying photon mapping
    - Keep track of which photons bounce through virtual objects
    - Irradiance caching
  - Modifying other techniques (environmental lighting, radiosity)

- Pro: Eliminates the inefficiency of rendering much of the scene the same way twice
  - Save on computation effort and decrease render time!

- Con: No unified technique, and the proposed methods have limitations
  - Often only works on diffuse materials

Top: Standard photon mapping (cs224)  
Bottom: Real photo (left) compared to Grosh et al. (2005) (right)
## Summary of Techniques

<table>
<thead>
<tr>
<th>Category</th>
<th>Best</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Capture time and effort</td>
<td>Perceptual</td>
<td>Explicit geometry</td>
</tr>
<tr>
<td>2. Processing time and effort</td>
<td>Standard and temporal IBL, but Dense capture and Perceptual are close</td>
<td>Explicit geometry</td>
</tr>
<tr>
<td>3. Robustness and generality</td>
<td>Explicit geometry</td>
<td>Implicit light probes and perceptual</td>
</tr>
<tr>
<td>4. Physical accuracy</td>
<td>Explicit geometry</td>
<td>Implicit light probes, perceptual, standard IBL</td>
</tr>
<tr>
<td>5. Perceptual accuracy</td>
<td>Explicit geometry</td>
<td>Implicit light probes and standard IBL</td>
</tr>
</tbody>
</table>

- High cost, but high quality output: measured lighting techniques
  - Especially explicit geometry
- Low cost, but lower quality output: estimated lighting techniques
  - Especially perceptual and implicit light probes
Biggest Problems for Mixed Reality

• Best rendering techniques are still expensive
  – This will improve with time – better algorithms and better GPUs
  – Key for mixed reality is development specifically for mobile devices

• Best measuring equipment is invasive and/or requires more processing power
  – Improved computer vision algorithms could mitigate this

• Many techniques not very robust, even for common scenes
  – Remember all the difficult things in the average living room picture?
  – Techniques that work on 2D images may not always be convincing in stereo

Two different furniture placing apps on Google’s Project Tango

Why are the virtual objects unconvincing?
Overall Future Work

• Improvements to spatially varying lighting
  – Especially in more robust estimation of material properties with less invasive capture equipment
• Improve robustness for complex illumination environments and specular illumination
• HDR will become more available, as will converting from LDR to HDR
  – It’s already kind of available on mobile – lots of free cell phone apps for tone mapping!
• Better rendering methods will become faster and cheaper
• Create techniques for inserting into legacy video
• Improve algorithms for inserting into legacy images, improve accessibility and ease of use for average consumer
  – Perceptual methods seem like a good way forward here
  – Stay tuned!
Automatic Scene Recognition for 3D Object Composition
Motivation

- Cinematography (controlled)
- Mixed Reality (variable)
Method

• Goal: Given RGB Image, estimate 3D scene
• *Estimated lighting conditions* method
Method

- Goal: Given RGB Image, estimate 3D scene
- *Estimated lighting conditions* method
Method

• Goal: Given RGB Image, estimate 3D scene
• *Estimated lighting conditions* method

Vanishing point method
Method

• Goal: Given RGB Image, estimate 3D scene
• *Estimated lighting conditions* method
Method

• Goal: Given RGB Image, estimate 3D scene
• *Estimated lighting conditions* method

- Vanishing point method
- Depth Inference
- Color Retinex
Method

- Goal: Given RGB Image, estimate 3D scene
- *Estimated lighting conditions* method

Vanishing point method

Depth Inference

Color Retinex

In-view and out-of-view illumination estimation
Method

• Goal: Given RGB Image, estimate 3D scene
• *Estimated lighting conditions* method
Method

- Place a 3D object in the approximated scene
- *Additive differentiable render*
Depth Inference

- We have camera parameters
  - Geometry can be constructed from depth
- Depth, materials, and lights are all still unknown
  - Many, many combinations create our RGB input
  - Simplifying constraint, assume a Manhattan World
    - All normals align with one of three axes
Depth Inference

\[ \arg \min_D E_{geom}(D) = \sum_{i \in \text{pixels}} E_t(D_i) + \lambda_m E_m(N(D)) + \lambda_o E_o(N(D)) + \lambda_3 E_{3s}(N(D)) \]

D is a depth map

N(D) is a normal map, formed by D
Depth Inference

\[
\text{argmin}_{D} \ E_{geom}(D) = \sum_{i \in \text{pixels}} E_t(D_i) + \lambda_m E_m(N(D)) + \lambda_o E_o(N(D)) + \lambda_3s E_{3s}(N(D))
\]

To compute \( E_t(D_i) \):
1. match input with RGB-D candidates from database
2. warp candidate to align with input’s features
3. sum difference in \( D \) and warped candidate’s depth
Depth Inference

\[
\arg\min_D E_{geom}(D) = \sum_{i \in \text{pixels}} E_t(D_i) + \lambda_m E_m(N(D)) + \lambda_o E_o(N(D)) + \lambda_3 s E_3 s(N(D))
\]

\(E_m(N(D))\) is small if normals are one of three axes.
Depth Inference

\[
\arg\min_D E_{geom}(D) = \sum_{i \in \text{pixels}} E_t(D_i) + \lambda_m E_m(N(D)) + \lambda_o E_o(N(D)) + \lambda_{3s} E_{3s}(N(D))
\]

\(E_o(N(D))\) “constraints the orientation of planar surfaces.”
Depth Inference

\[
\arg\min_D E_{geom}(D) = \sum_{i \in \text{pixels}} E_t(D_i) + \lambda_m E_m(N(D)) + \\
\lambda_o E_o(N(D)) + \lambda_{3s} E_{3s}(N(D))
\]

\( E_{3s}(N(D)) \) discourages local variance in normals, and therefore in \( D \).
Depth Inference

\[
\arg\min_{\mathbf{D}} E_{geom}(\mathbf{D}) = \sum_{i \in \text{pixels}} E_t(\mathbf{D}_i) + \lambda_m E_m(N(\mathbf{D})) + \\
\lambda_o E_o(N(\mathbf{D})) + \lambda_{3s} E_{3s}(N(\mathbf{D}))
\]

Solve this to get our depth map \( \mathbf{D} \).
Light Inference

In-view Positions

Step 1: per-pixel binary classifier
Light Inference
In-view Positions

Step 1: per-pixel binary classifier

Step 2: inverse project into 3D space

\[ \mathbf{X} = D(x, y)K^{-1}[x, y, 1]^T \]

\( \mathbf{X} \) is final light position

\( D \) is depth at pixel \((x, y)\)

\( K^{-1} \) is inverse projection matrix
Light Inference
Out-of-view Positions

Assumption: if two pictures look similar, their lighting environments are also similar.
Light Inference

Out-of-view Positions

For feature vectors $F_i$ and $F_j$ of images $i, j$

$$S_{i,j} = w^T F_i \cdot w^T F_j$$

where $S_{i,j}$ denotes similarity and $w$ is a learned feature weight vector.

Pick top $k$ most similar to input image.
Light Inference

Intensities

For each light source $e_k$,

Render scene $\Rightarrow R(e_k)$

Given input image $I$,

$$Q(w, \gamma) = \sum_{i \in \text{pixels}} \left\| I_i - \left[ \sum_{k \in \text{sources}} w_k R_i(e_k) \right] \right\|^\gamma$$

$Q$ is distance between input image and 3D scene. Choose $w, \gamma$ to minimize $Q$. 
Light Inference
Intensities

While we’re at it…

\[ P(w) = \sum_{k \in \text{sources}} \left[ \|w_k\|_1 + w_k \sum_{i \in \text{pixels}} \|\nabla R_i(e_k)\| \right] \]

Minimize \( P \) to discourage:

– Too many lights
– Lights that take long to converge in rendering
Light Inference

Intensities

Given $\gamma_0, \lambda_P, \lambda_\gamma$. We want to minimize

$$\arg\min_{w, \gamma} Q(w, \gamma) + \lambda_P P(w) + \lambda_\gamma \|\gamma - \gamma_0\|_2$$

s.t. $w_k \geq 0 \ \forall k, \ \gamma > 0$.

Solved with active set method.

This gives us:

- Light intensities $w$
- Camera response function $\gamma$
Additive Differential Rendering

1. Model or estimate scene
2. Render
   - Scene \((R_0)\)
   - Scene with addition \((R_A)\)
3. Blend
   - \(R_A, \text{ (Input } + R_A - R_0\)"
   - Mix with mask
Results
Results
Evaluation

• Conducted a user study
• Users shown two images
  – Ground truth
  – Composited image
• Asked to pick real one
• Users selected the composited image about 35% of the time
  – 50% would be ideal
Thoughts

• Limits of assuming a Manhattan World
  – Can only place objects on surfaces aligned a dominant axis
  – Less successful in complex environments (outdoor scenes)

• Try to infer lights and materials first
  – Can now estimate RGB output in scene at a depth
    • Alternative constraint on depth solution space
    • Incorporate this in energy function
Thoughts

• Limits of assuming Lambertian materials
  – Realism of re-rendering
  – Accuracy of light intensity inference
    • Due to inaccurate $R(e_k)$ in $Q$ term

• Instead, try semantic analysis
  – Certain objects are more likely to have certain BRDFs
  – Could provide arbitrary BRDFs
    • Reflective and transparent objects
Thoughts

• Out-of-view light position inference
  – Avoid ranking every environment in dataset
  – Put dataset in a feature space KD-tree, then search

• Light intensity inference
  – Light Intensity is a global feature
  – Render from each light at low resolution
    • Compare to down-sampled input image
Thoughts

• Not the method of choice for physical accuracy
• Scene inference enables other post-processing effects
  – Depth of field
  – Lighting and material adjustments
  – Minor geometry warping
• Autonomy lends itself to real-time applications
  – Snapchat
  – Augmented Reality HMD
Thoughts

• Too slow for real-time, how does Snapchat do it?
• Likely a specialized method that assumes a face
• Likely estimates facial geometry, enabling:
  – Warping of facial features
  – Modification of face “material”
• Lighting may be inferred from intensity distribution of face
  – Might assume one point light