Lecture 23: Image-based Modeling and Lighting

Fall 2004

Kavita Bala

Computer Science

Cornell University

Announcements

- Projects: Email short (1-2 paragraphs) update on project by next Tuesday
  - Plans till project due date

Complexity

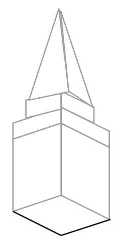
- Lighting: many lights, environment maps
  - Global illumination, shadows
- Materials: BRDFs, textures
- Geometry: Level-of-detail, point-based representations
- All: image-based rendering

Image-based Modeling

- Extract geometry + textures from pictures
  - Create a simple model
- Reuse geometry and texture to render scene from new viewpoints
- Façade system:
  - Debevec and Malik, UCB

Match Edges

- Start from photograph of building
- User creates rudimentary model with cubes
- Marks corresponding edges between model and photographs
- Enough correspondences to reconstruct camera and model parameters



Use blocks

- Easy to specify
- Fewer interactions
- Easy to determine final surface

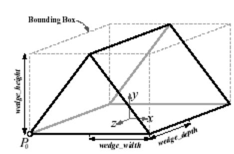


Figure 3: *A wedge block with its parameters and bounding box.*

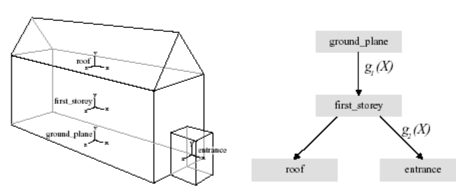


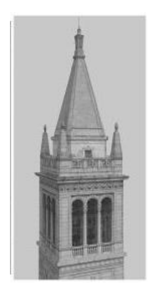
Figure 4: **(a)** *A geometric model of a simple building.* **(b)** *The model's hierarchical representation. The nodes in the tree represent parametric primitives (called* blocks*) while the links contain the spatial relationships between the blocks.*

Find best fit

- Project model representation to camera
- Create error term
- Minimize error
- Iterative method updates model representation and looks for best fit
- Also solves for camera parameters

Extract textures

- Extract textures from original photograph
- Project them onto the surface



View-dependent texture mapping

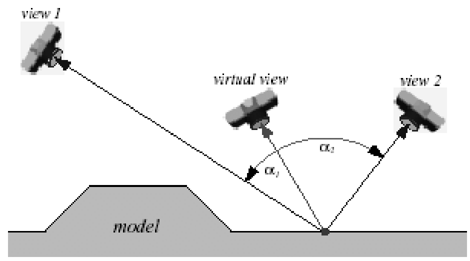
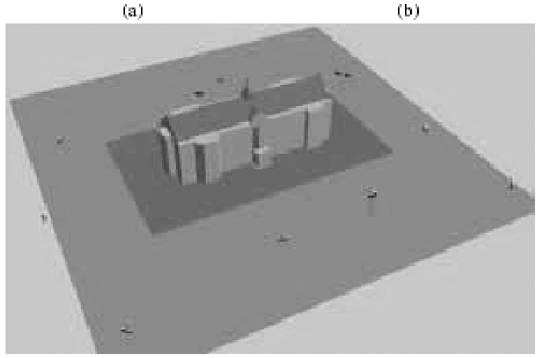


Figure 12: *The weighting function used in view-dependent texture mapping. The pixel in the virtual view corresponding to the point on the model is assigned a weighted average of the corresponding pixels in actual views 1 and 2. The weights w_1 and w_2 are inversely inversely proportional to the magnitude of angles α_1 and α_2. Alternately, more sophisticated weighting functions based on expected foreshortening and image resampling can be used.*

Example: Extract Geometry

(a) (b)



(c)

Figure 8: *The high school model, reconstructed from twelve photographs.* **(a)** *Overhead view.* **(b)** *Rear view.* **(c)** *Aerial view showing the recovered camera positions. Two nearly coincident cameras can be observed in front of the building; their photographs were taken from the second story of a building across the street.*

Example: Final Model w/ Textures

Used in the Matrix

IBM Discussion

- Cons:
  - Small geometric details not included in model
    - primitives represented by the user are limited
  - Features in textures not part of model
  - Fair amount of manual input required!

- Pros:
  - Effective and useful
    - RealViz, …
  - Open area of research

Cons of IBM

- Reasonable image quality but
  - Lighting is baked in: to undo lighting need material properties
  - Geometry is fixed



Image Based Lighting

- Real Scene

- Goal: place synthetic objects on table

© Kavita Bala, Computer Science, Cornell University

Image Based Lighting

- Capture illumination using illumination sphere

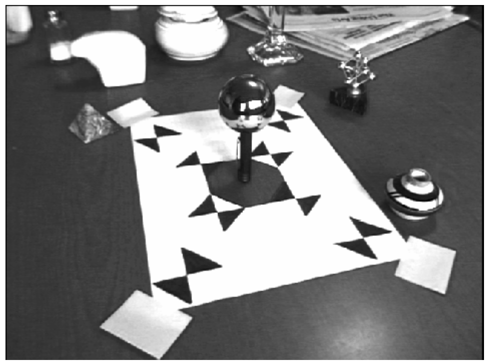


Image Based Lighting

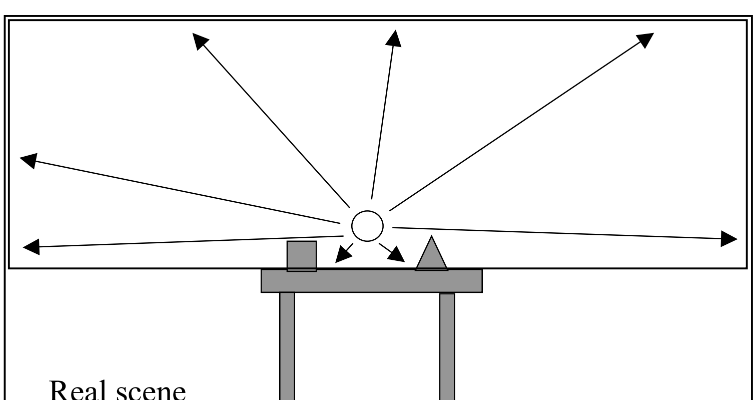
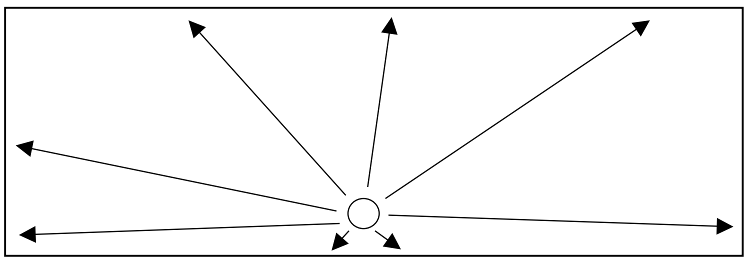
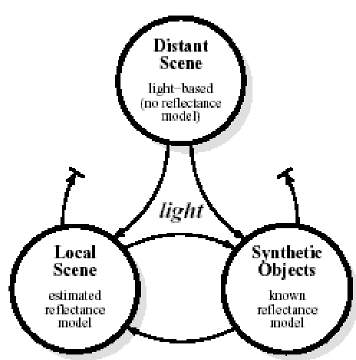


Image Based Lighting

captured illumination field



© Kavita Bala, Computer Science, Cornell University



© Kavita Bala, Computer Science, Cornell University

Image Based Lighting

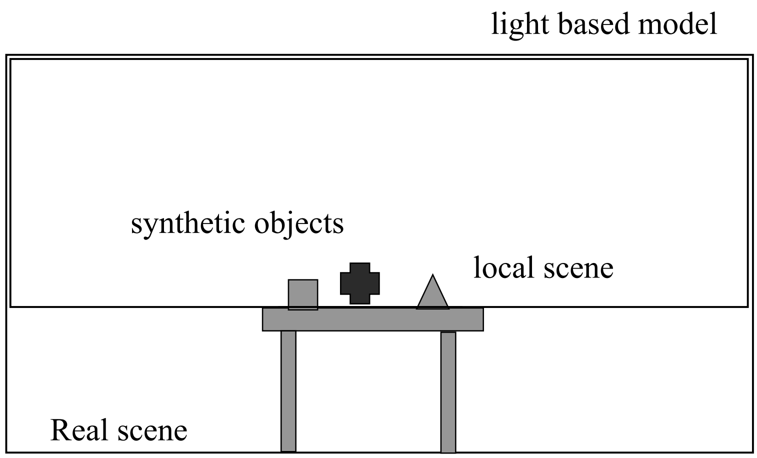


Image Based Lighting

- Use renderer - compute effects of synthetic objects on local scene

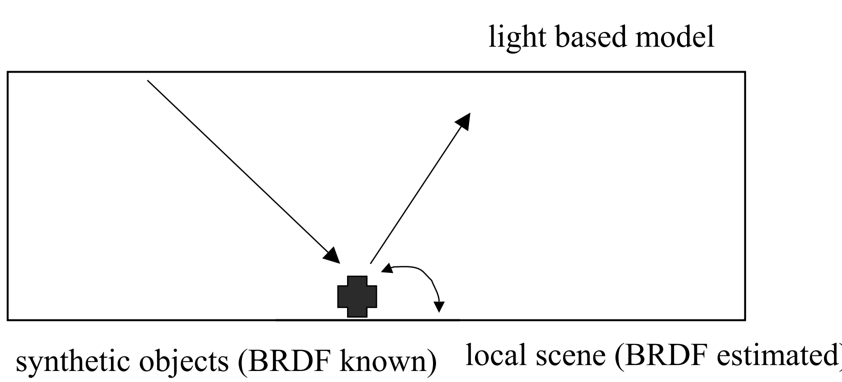


Image Based Lighting

- Render into the scene

background

Image Based Lighting

- Render synthetic objects

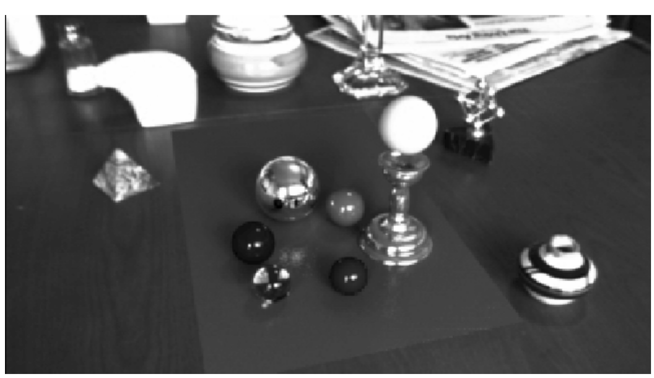


Image Based Lighting

- Effect of local scene on real scene

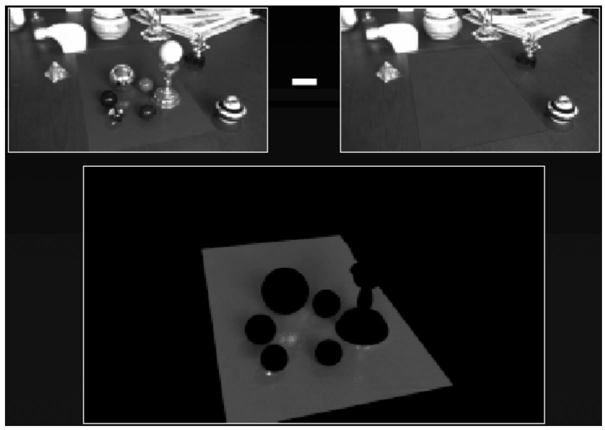


Image Based Lighting

- Add differences to image

Discussion

- Good results for special effects
- But, estimating reflectances problematic

Inverse global illumination

- Goal: recover BRDF per patch
- But account for GI in an interior environment

- Idea: solve for Ward BRDF model
  - specular parameters uniform per patch
  - diffuse interreflection handled trivially
  - specular interreflection (rare) handled by simple iterative algorithm

Handling interreflection

- In diffuse case, all is easy
  - each patch lit by sources and other patches

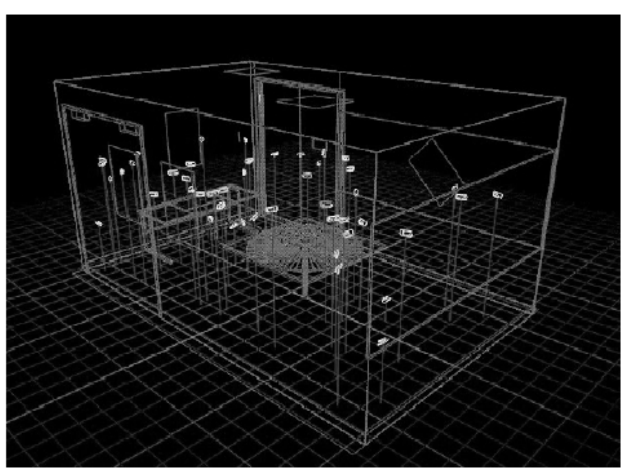
$$B_i = E_i + \rho_i \sum_j B_j F_{ij}$$

- Specular case is more tricky
  - illum. depends on specular parameters elsewhere
  - assuming diffuse dominates, iteratively solve for a correction for specular illumination
  - (note: must observe a highlight on every surface)

$$L_i = (\frac{\rho_d}{\pi} + \rho_s K(\alpha, \Theta_i)) I_i$$

IGI inputs

40 high-dynamic-range images



IGI results

Re-rendered (bottom) to match input (top)

IGI results

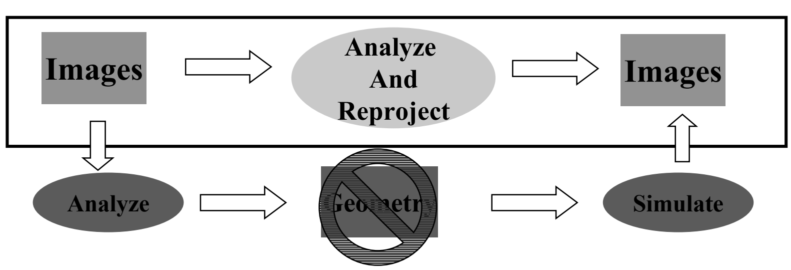
Re-rendered with new light (bottom) to match photos (top)

Summary of IBM, IBL

- Very powerful
- Many interesting research areas to be solved

Image-based Approaches

- Goal: Realism!
- Image is input *and* rendering primitive



Comparison

- No depth
  - QuickTime VR (simplest, 2D panoramas)
  - Lumigraph/Light Field (4D arrays)
- Reconstructed Depth
  - Plenoptic Modeling (2.5 D)
- IBM vs. IBR
  - Some manual user input ok
  - Simple geometry recovered with user assistance
  - Complex geometry represented as texture

Conclusions

- IBR: promising approach to handle complexity
- Benefits:
  - No labor-intensive modeling
  - High geometric and material complexity
  - Rendering time constant: proportional to image size, independent of scene complexity
- Disadvantages:
  - Quality
  - Not-quite automatic

- Exam: Everything till today