The Future of Mixed Reality

Issues in Illumination and Shadows

Charles E. Hughes

Jaakko Konttinen

Sumanta N. Pattanaik

University of Central Florida

Why Virtual Lighting

- How do you give a real flashlight to a virtual object?

- Certain effects, such as the overall luminance of the scene (such as changing the time of day), may be cheaper to implement virtually. Useful for dynamic simulation applications.

- In entertainment applications, sometimes physically incorrect lighting may be necessary to create the desired dramatic effect.

Importance of Shadows

- Shadows are an important part of lighting simulation:
  - Lights should cast shadows on real objects even though the light sources are virtual.

Importance of Shadows

- Helps with suspension of disbelief.
- Helps us understand:
  - Relative virtual-to-real object position and height in space.
  - Shape of a virtual shadow receiver’s surface.
  - Position of nearby virtual objects that are occluded or outside the field of view (virtual enemy in hiding).

Importance of Shadows

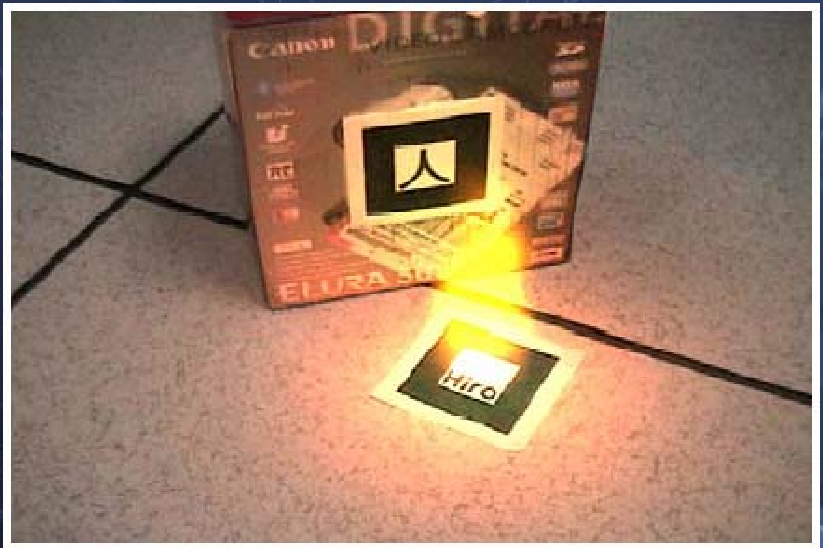
- Is this man levitating?

Importance of Shadows

- Is this man levitating?

Our Contribution

- We demonstrate two ways to inject virtual light into a scene in the context of two practical MR applications.



Virtual Fire

Virtual Flashlight

Occlusion Models

• Our algorithms make use of the occlusion models of real objects which are common in MR applications.

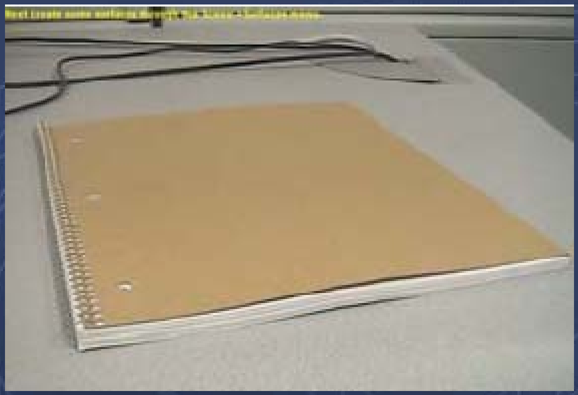
• Occlusion models describe the geometry of all real objects in the scene. They can be pre-measured or automatically generated.

• Model position relative to camera is tracked in some way.

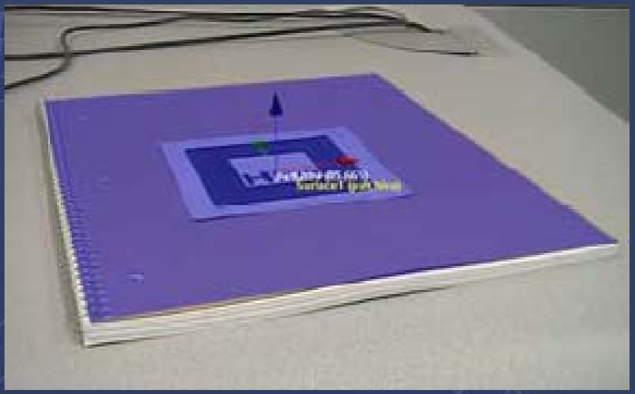
• Usually used for determining which parts of virtual objects are hidden by real objects closer to the camera.

• Having this geometry available lets us adapt many algorithms from computer graphics.

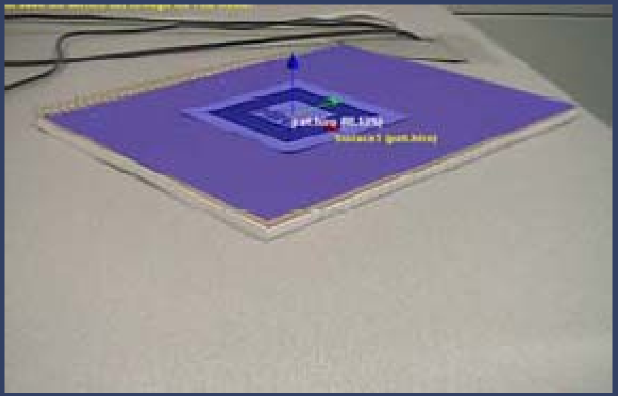
Occlusion Models



No occlusion model



With occlusion model



Tracking the model's position

- Example of the occlusion model for a notebook (approximated by a single polygon).

- Tracked by a marker.

Tracking

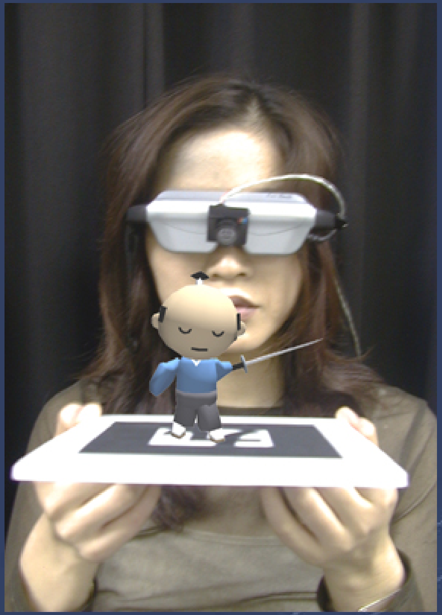
- Need to know position relative to camera of every interactive real object.
- Any object that is allowed to move independently of others must also be tracked independently.
- Several methods of tracking:
  - Sensor-based (InterSense, Polhemus, GPS)
  - Image-based (ARToolkit)

Tracking

- ***ARToolkit*** is an image-based tracker which derives camera position relative to a particular marker based on its location and tilt in the video frame.

- Developed by University of Washington's Human Interface Technology Lab.

- We chose ***ARToolkit*** for our demo for several reasons:
  - Light-weight
  - Free
  - Easy to set-up

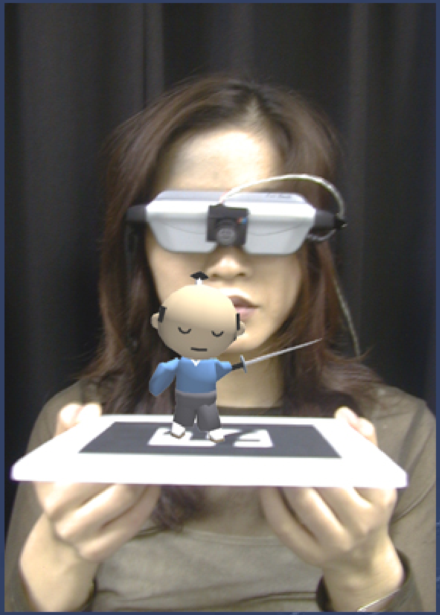


Tracking

- Not necessarily that great for larger projects:
  - Poor tracking quality of far-away objects.
  - Image-based tracking done in software = slow
  - Markers everywhere!
  - Tracked object's marker must be visible in the video frame

- Luckily the tracking problem can be easily isolated from the rest of the system and thus the method of implementation can be changed easily.

- ARToolkit: http://www.hitl.washington.edu/artoolkit/



Virtual Fire

- Our virtual fire application simulates a real environment light by computer-generated fire.

- The process works by calculating how the intensity of each pixel covered by an occlusion model is increased by the virtual flames.

Figure 3: Sequence of virtual fire brightening a real scene, from left to right: (a) Original intensity, (b) Illumination of real scene without virtual flames, (c) final scene with composited virtual flames.

Virtual Fire

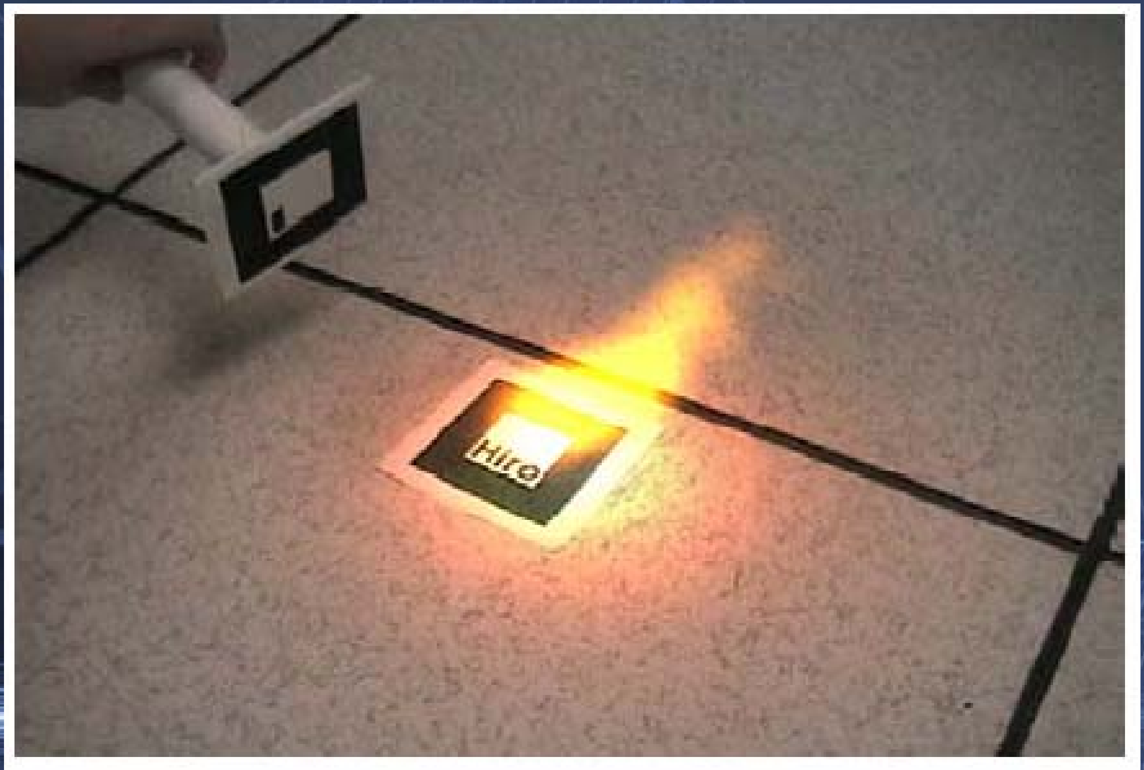
- Thanks to occlusion models, this calculation can be done with many known computer graphics lighting algorithms and can be accelerated by graphics cards. The calculation can be simple or complex depending on the requirements of the application.

- The original intensity of the pixel is scaled up based on the result of the computation. Finally, virtual flames are drawn onto the image.

Figure 3: Sequence of virtual fire brightening a real scene, from left to right: (a) Original intensity, (b) Illumination of real scene without virtual flames, (c) final scene with composited virtual flames.

Bonus: Virtual Wind

- A separate marker tracks the position of a non-existing fan. Lighting on ground shifts with flames.



Virtual Flashlight

- Artificially make a room darker, and restore it to its original intensity with a virtual flashlight.

- Done in multiple steps.

Unmodified video frame

Final processed frame

Virtual Flashlight

- **Step #1**: Scale down intensity of pixels that are not covered by tracked occlusion models by some constant factor.

Partially darkened frame

Final processed frame

Virtual Flashlight

- **Step #2**: Calculate shadows from virtual objects falling on real objects.
- Using a version of the shadow volume algorithm from computer graphics adapted for MR.

Virtual shadows on real objects

Final processed frame

Virtual Flashlight

- **Step #3**: Artificially darken occlusion model-covered pixels and restore some intensity using the virtual light.
- Spotlight algorithm from computer graphics.

Full virtual lighting applied

Final processed frame

Virtual Flashlight

- **Step #4**: Draw virtual objects and compute shadows from real and other virtual objects falling on them..

Final processed frame

Final processed frame

Virtual Flashlight

- **Step #4**: Draw virtual objects and compute shadows from real and other virtual objects falling on them..

Final processed frame

Final processed frame

Acknowledgements

- RAVES Project: Office of Naval Research / Naval Research Laboratory
- MR MOUT Project: Embedded Training for Dismounted Soldier STO at the Army Research, Development and Engineering Command (RDECOM)