# Unknown, Irregular Light Sources in Dynamic Global Illumination

**Sharat Chandran**
http://www.cse.iitb.ac.in/~sharat

(Joint work with Mayur Srivastava)
October 21, 2002

# Overview

- The nature of the problem
- Background of existing methods
  - Pre-1995 methods
  - Recent methods
- Setting for our solution
- The intuition behind the algorithm
- Some details
- Some results
- Conclusion, limitations, and prospects

# Why consider this problem?

- Sudden changes in lighting are common (If the geometry of the light sources are known in advance, existing incremental radiosity techniques can be used.)

# Why consider this problem?

- Sudden changes in lighting are common (If the geometry of the light sources are known in advance, existing incremental radiosity techniques can be used.)
- However, lighting may change unpredicatably

# Why consider this problem?

- Sudden changes in lighting are common (If the geometry of the light sources are known in advance, existing incremental radiosity techniques can be used.)
- However, lighting may change unpredicatably


# Some other situations to consider

- Partial pulling of window blinds or closing of curtains

# Some other situations to consider

- Partial pulling of window blinds or closing of curtains
- Appearance of daylight in room because of motion of parked cars

# Some other situations to consider

- Partial pulling of window blinds or closing of curtains
- Appearance of daylight in room because of motion of parked cars
- Bathing of interiors as the full moon appears past overcast clouds

# Some other situations to consider

- Partial pulling of window blinds or closing of curtains
- Appearance of daylight in room because of motion of parked cars
- Bathing of interiors as the full moon appears past overcast clouds
- Effects created in operas and theaters

# Some other situations to consider

- Partial pulling of window blinds or closing of curtains
- Appearance of daylight in room because of motion of parked cars
- Bathing of interiors as the full moon appears past overcast clouds
- Effects created in operas and theaters
- Revolving police search beams, particularly during dusk and dawn

# Some earlier work

# Some earlier work

- Precomputations *(Dorsey et al. 1991)*

# Some earlier work

- Precomputations *(Dorsey et al. 1991)*

  Problem: Run time light changes is not permissible

# Some earlier work

- Precomputations *(Dorsey et al. 1991)*
  Problem: Run time light changes is not permissible
- Creating new patch, and shooting possibly **negative** energy *(Chen 1990)*

# Some earlier work

- Precomputations *(Dorsey et al. 1991)*
  Problem: Run time light changes is not permissible
- Creating new patch, and shooting possibly **negative** energy *(Chen 1990)*
  Problem: Calls for **re-computation**

# Some earlier work

- Precomputations *(Dorsey et al. 1991)*
  Problem: Run time light changes is not permissible
- Creating new patch, and shooting possibly **negative** energy *(Chen 1990)*
  Problem: Calls for **re-computation**
- Line-space hierarchy *(Drettakis et al. 1997, Schoeffel et al. 1999, Damez 1999)*

# Some earlier work

- Precomputations *(Dorsey et al. 1991)*
  Problem: Run time light changes is not permissible
- Creating new patch, and shooting possibly **negative** energy *(Chen 1990)*
  Problem: Calls for **re-computation**
- Line-space hierarchy *(Drettakis et al. 1997, Schoeffel et al. 1999, Damez 1999)*
  Problem: Only **local** Illumination changes are handled

# Some earlier work

- Precomputations *(Dorsey et al. 1991)*
  Problem: Run time light changes is not permissible
- Creating new patch, and shooting possibly **negative** energy *(Chen 1990)*
  Problem: Calls for **re-computation**
- Line-space hierarchy *(Drettakis et al. 1997, Schoeffel et al. 1999, Damez 1999)*
  Problem: Only **local** Illumination changes are handled
- Caching *(Tole et al. 2002)*

# Some earlier work

- Precomputations *(Dorsey et al. 1991)*
  Problem: Run time light changes is not permissible
- Creating new patch, and shooting possibly **negative** energy *(Chen 1990)*
  Problem: Calls for **re-computation**
- Line-space hierarchy *(Drettakis et al. 1997, Schoeffel et al. 1999, Damez 1999)*
  Problem: Only **local** Illumination changes are handled
- Caching *(Tole et al. 2002)*
  Problem: Cache defeated by **new** patch inclusion

# How do *we* define dynamic?

- Scene geometry does not change

# How do *we* define dynamic?

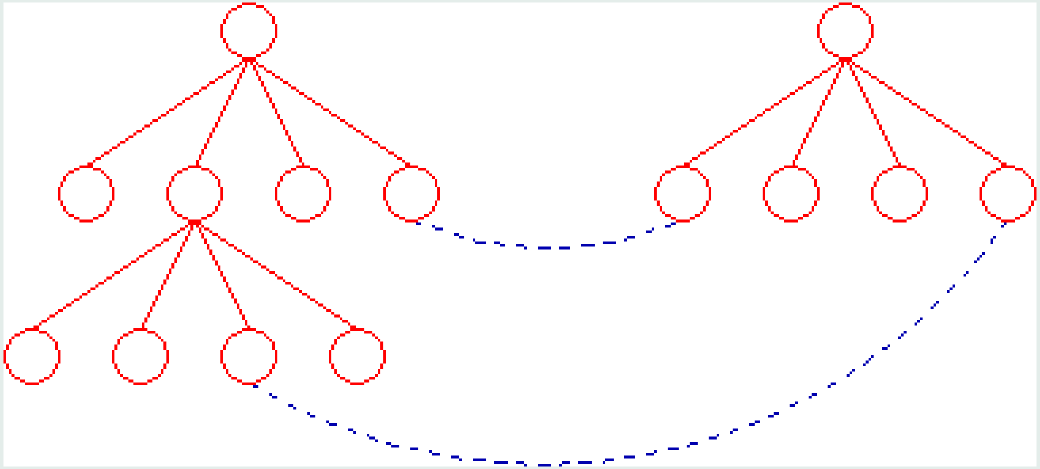
- Scene geometry does not change
- Light sources (of arbitrary shape) can be added or removed

# How do *we* define dynamic?

- Scene geometry does not change
- Light sources (of arbitrary shape) can be added or removed
- Straightforward solution
  - Introduce a new patch that models the light source, and recompute energy values
  - To turn of light source, shoot negative energy (keep the patch in memory, we may need it later on?)
  - Creation of patch and recomputing solution is time consuming.
- Our approach
  - Can we take a incremental solution for adding?
  - Can we modify the best solution out there?

# Hierarchical Radiosity

- Trade complexity for error in a disciplined way
- Computation of geometric configuration ("formfactors") is more time consuming than energy calculation
- Preprocess scenes by setting up energy exchange links.

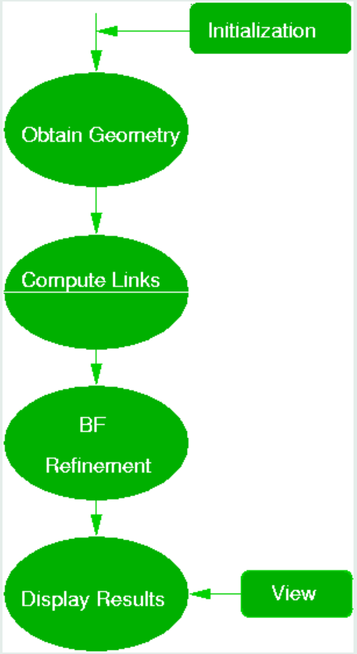


# Hierarchical Radiosity Interaction Computation

If we divide $k$ patches into $n$ elements, the total number of interactions is $O(n)$

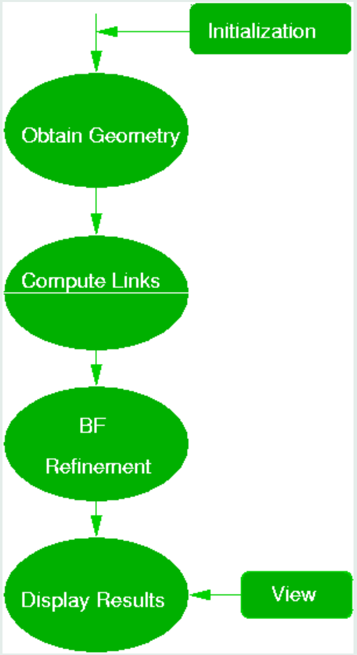
```
Refine(Patch *p, Patch *q, float Feps, float Aeps) {
  float Fpq, Fqp;
  Fpq = FormFactEstimate(p,q);  Fqp = FormFactEstimate(q,p);
  if (Fpq < Fqp && Fqp < Feps) Link (p,q);
  else {
    if (Subdiv(q, Aeps)) {
      Refine (p,q.nw,Feps,Aeps); Refine (p,q.ne,Feps,Aeps);
      Refine (p,q.sw,Feps,Aeps); Refine (p,q.se,Feps,Aeps);
    else
      Link(p,q);
    }}}
```

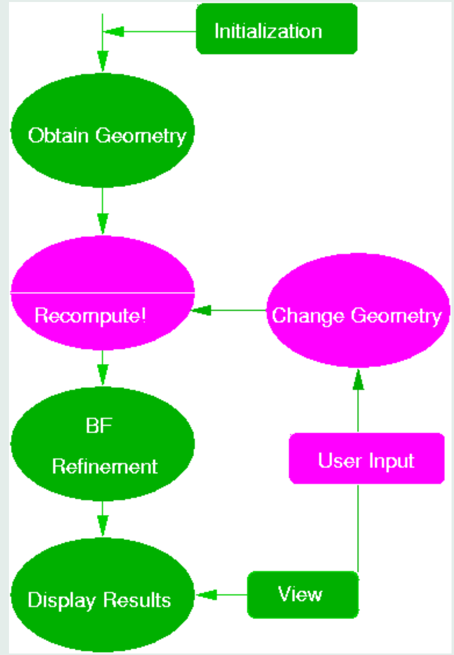
# Context of Prior Solution



Home Page
Title Page
Contents
Page 9 of 20
Go Back
Full Screen
Close
Quit

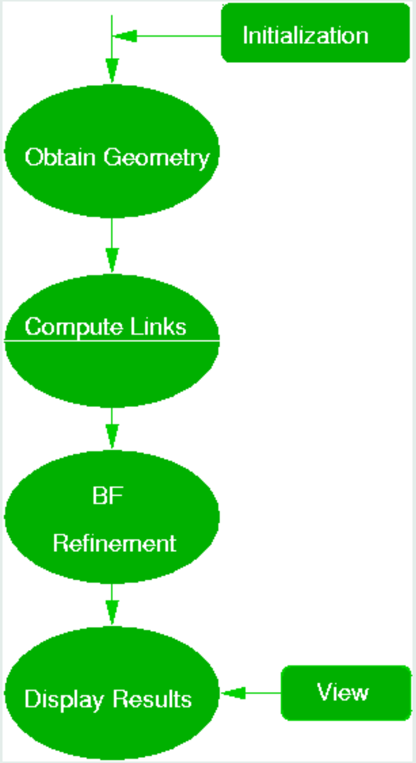
# Context of Prior Solution



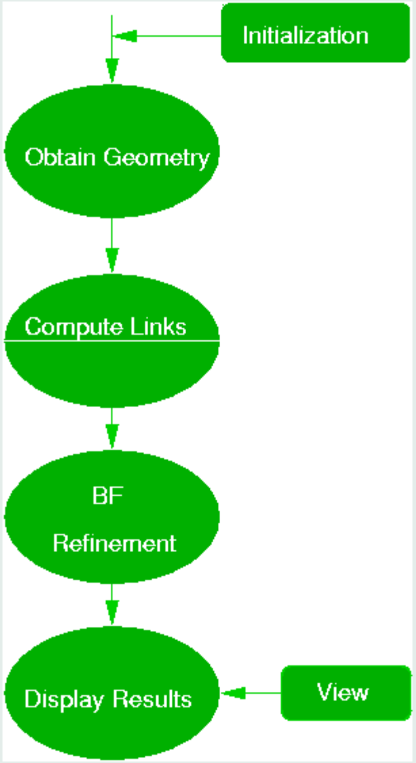



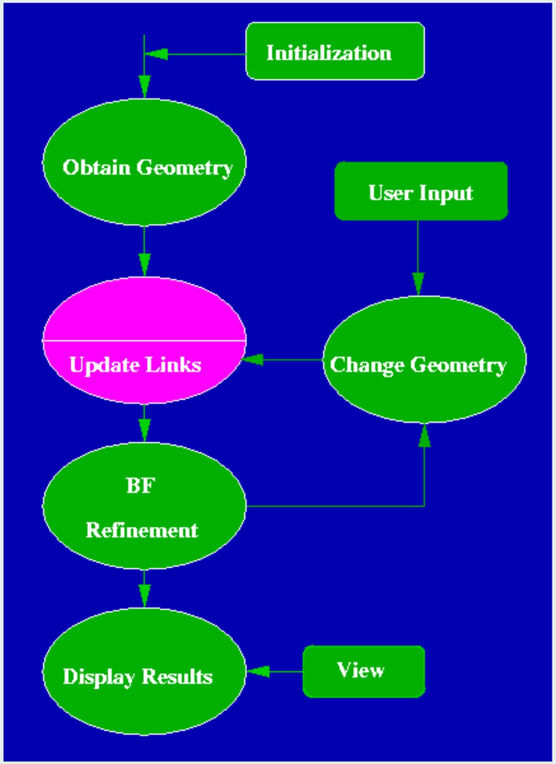

- Introduce a **new** patch modeling light source (requires computation of the set of patches the user inputted light source may impact)
- Set up links again
- Implies throwing away some of the hardwork done

# Context of *Our* Solution



# Context of *Our* Solution

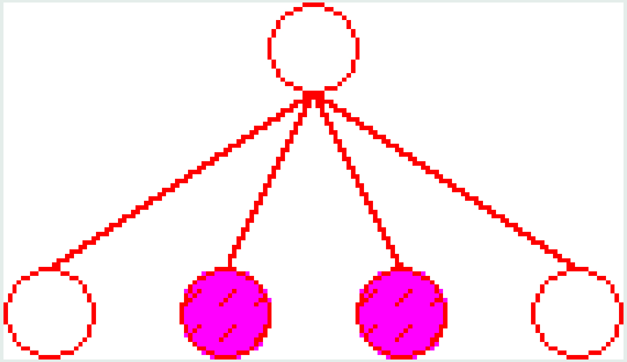



Update links, rather than recompute

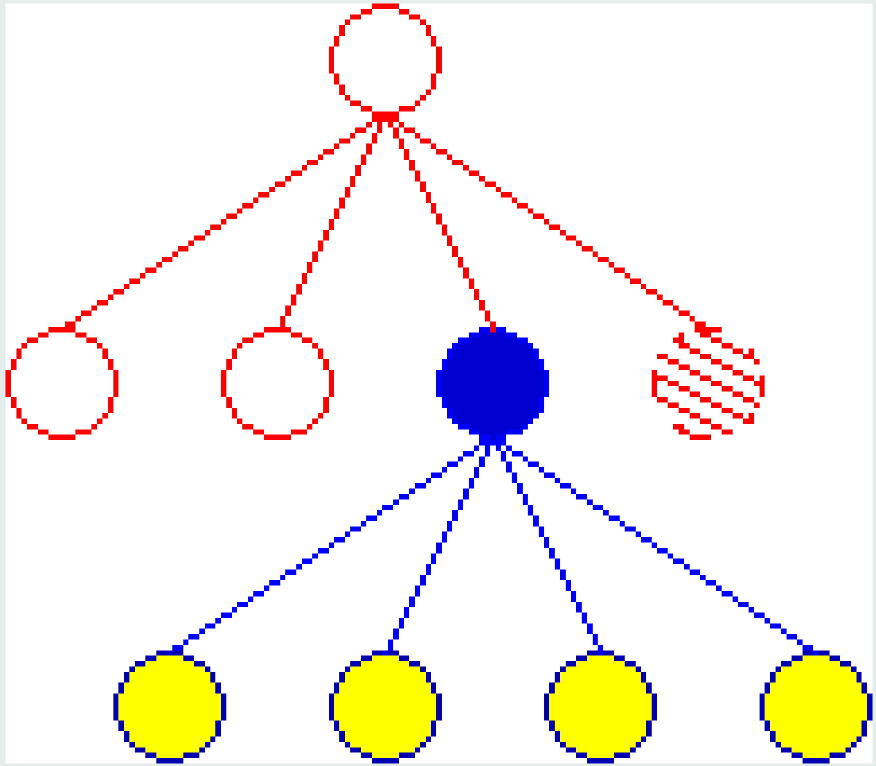
# Intuition Behind Our Solution

- HR has subdivided the geometry
- What if nodes in our tree is a part of the unknown light source?
- Straightforward to update patch properties of these nodes



# Intuition Behind Our Solution

- If the HR node is a proper subset of the light source . . .
- Expand the node
- Recursively process the node



# Adaptive Refinement: AR

- Mark the nodes that are *completey* inside the light source

# Adaptive Refinement: AR

- Mark the nodes that are *completey* inside the light source
- Sub divide nodes that partially intersects with the light source

# Adaptive Refinement: AR

- Mark the nodes that are *completey* inside the light source
- Sub divide nodes that partially intersects with the light source
- Further refine node if it is a leaf in the hierarchical radiosity computation, and recurse

```
Algorithm Intersection(LightSource R, Quadtree Q)
    if Disjoint(R,Q) then
        return NULL;
    if Contained(R,Q) then /* Q is contained in R*
        return Q;
    else for each child_i of Q do
        Intersection (R, Q.child_i);
}
```

- Assign the emissivity of the light source to the *marked* nodes.

# Adaptive Refinement: AR

- Mark the nodes that are *completey* inside the light source
- Sub divide nodes that partially intersects with the light source
- Further refine node if it is a leaf in the hierarchical radiosity computation, and recurse

```
Algorithm Intersection(LightSource R, Quadtree Q
        if Disjoint(R,Q) then
            return NULL;
        if Contained(R,Q) then /* Q is contained in R*
            return Q;
        else for each child_i of Q do
            Intersection (R, Q.child_i);
}
```

- Assign the emissivity of the light source to the *marked* nodes.
- Distribute the emissivity of a *marked* node to its descendents

# What if there is too much adaptive refinement?

AR may take considerable time when:

- Scenes are dense
- User introduced light source is not axis aligned with patch boundaries

# What if there is too much adaptive refinement?

AR may take considerable time when:

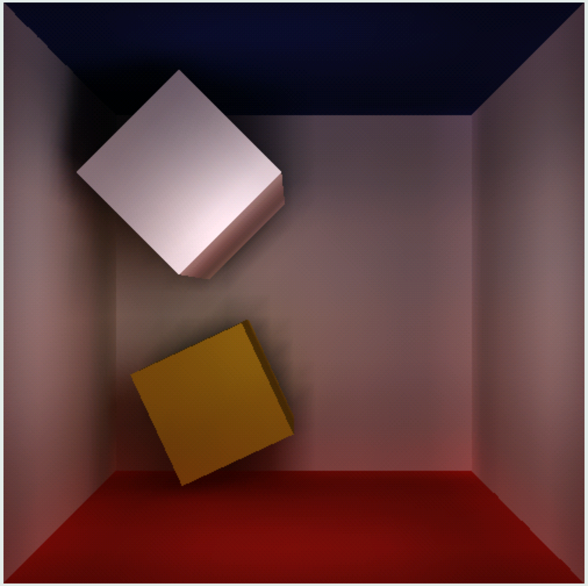
- Scenes are dense
- User introduced light source is not axis aligned with patch boundaries
- A faster solution (Fractional Emissivity(FE)
  - Do not refine the node if it is a HR leaf

# What if there is too much adaptive refinement?

AR may take considerable time when:

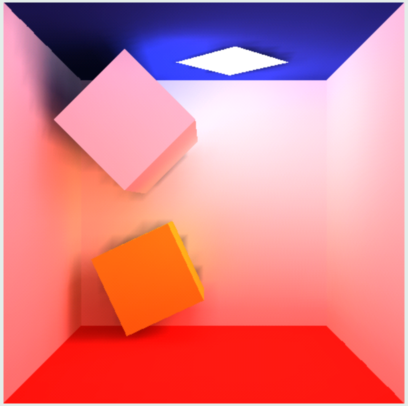
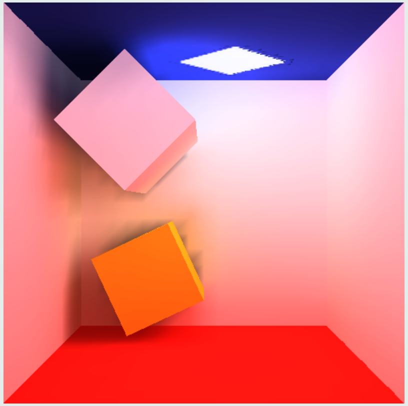
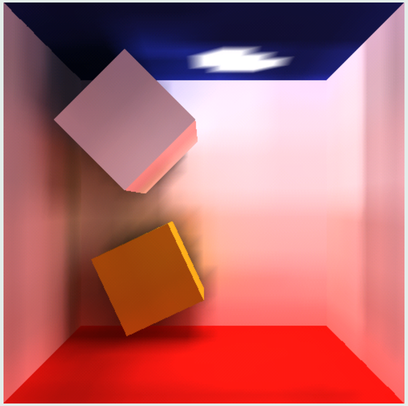
- Scenes are dense
- User introduced light source is not axis aligned with patch boundaries
- A faster solution (Fractional Emissivity(FE)
  - Do not refine the node if it is a HR leaf
  - Calculate the fractional overlap ($\triangle f$) with the light source
  - $E_{node} = \triangle f * E_{lightSource}$
  - Requires a clipping algorithm such as Sutherland-Hodgman
- Disadvantages of FE: Coarser approximation of light source

# Standard Test Scene



A pigeon's view of the Cornell Room.

# Results

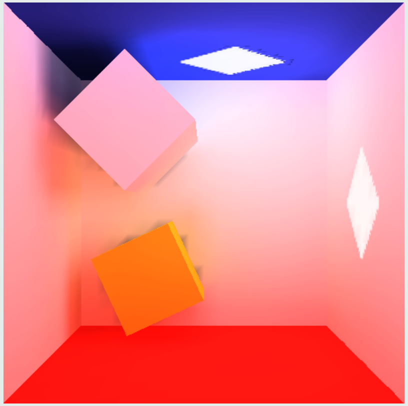




Naive method (126 sec) AR Rendering (3 sec) FE Rendering (0.8 sec)

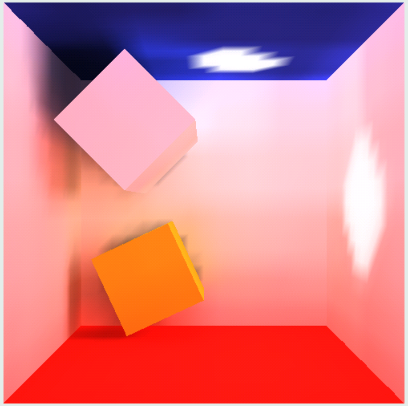
Light filters through the southern wall onto the blue wall. The resulting scene is rendered incrementally in the two methods on the right.

# Results

Naive method (172 sec)



AR Rendering (4 sec)



FE Rendering (0.8 sec)

A second light appears on the eastern wall.

Home Page

Title Page

Contents

Page 18 of 20

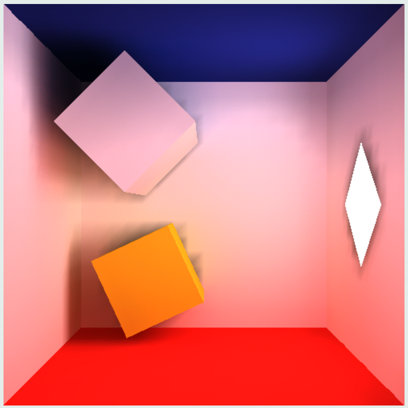
Go Back

Full Screen

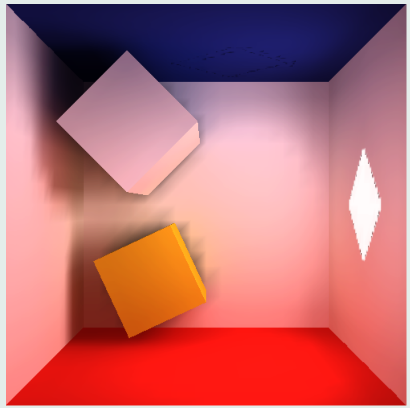
Close

Quit

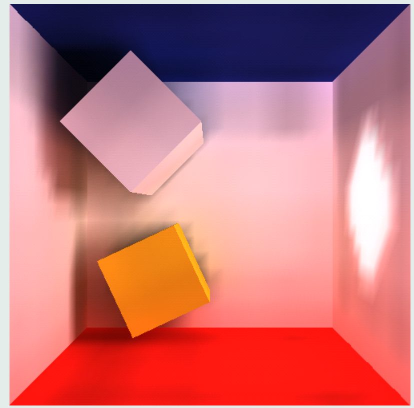
# Results



Naive method (92 sec)



AR Rendering (1.5 sec)



FE Rendering (0.8 sec)

The light on the blue wall disappears to reflect the passage of time.

# More Results

An office with two flat monitors, presumably with screen savers on.
The appearance changes because mouse motion causes the windows to break out of the screen saver mode.
The solution on the right is incrementally computed based on the illumination in the first by Algorithm AR.

# Conclusion

- Considered a sub-class of dynamic global illumination environments

# Conclusion

- Considered a sub-class of dynamic global illumination environments
- The method is fast and simple

# Conclusion

- Considered a sub-class of dynamic global illumination environments
- The method is fast and simple
- The solution can be buried in more sophisticated schemes