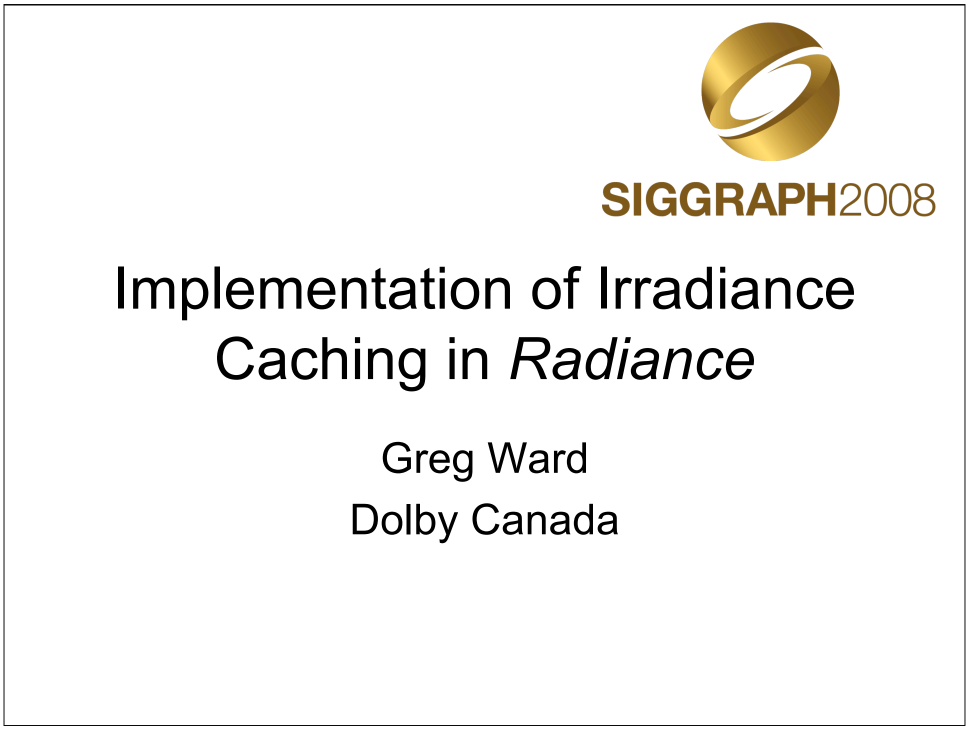
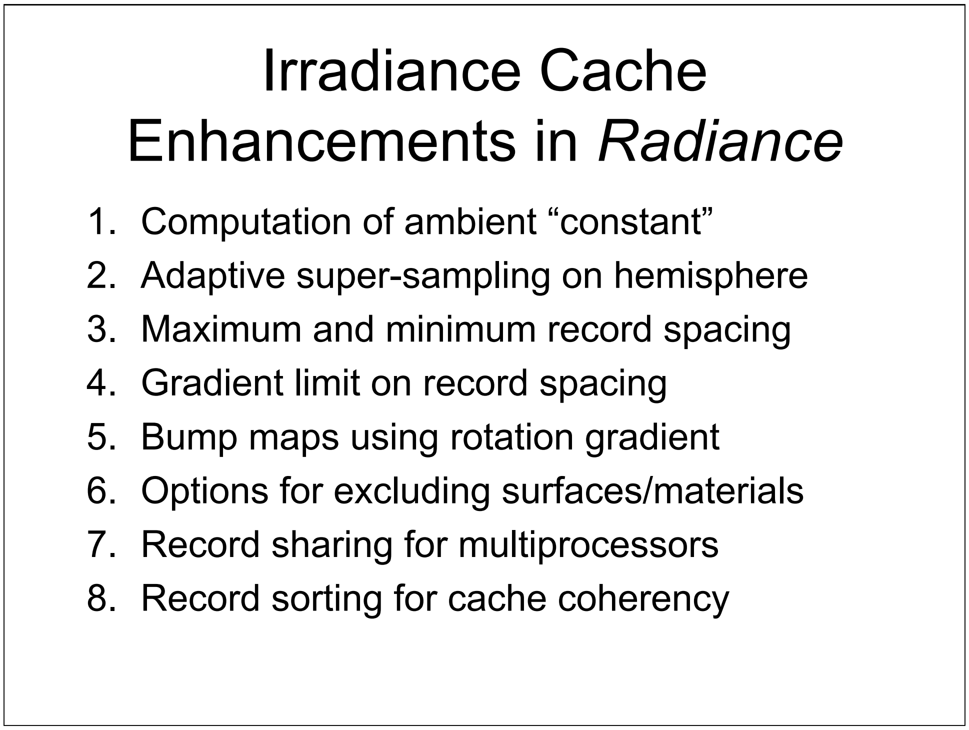


SIGGRAPH2008

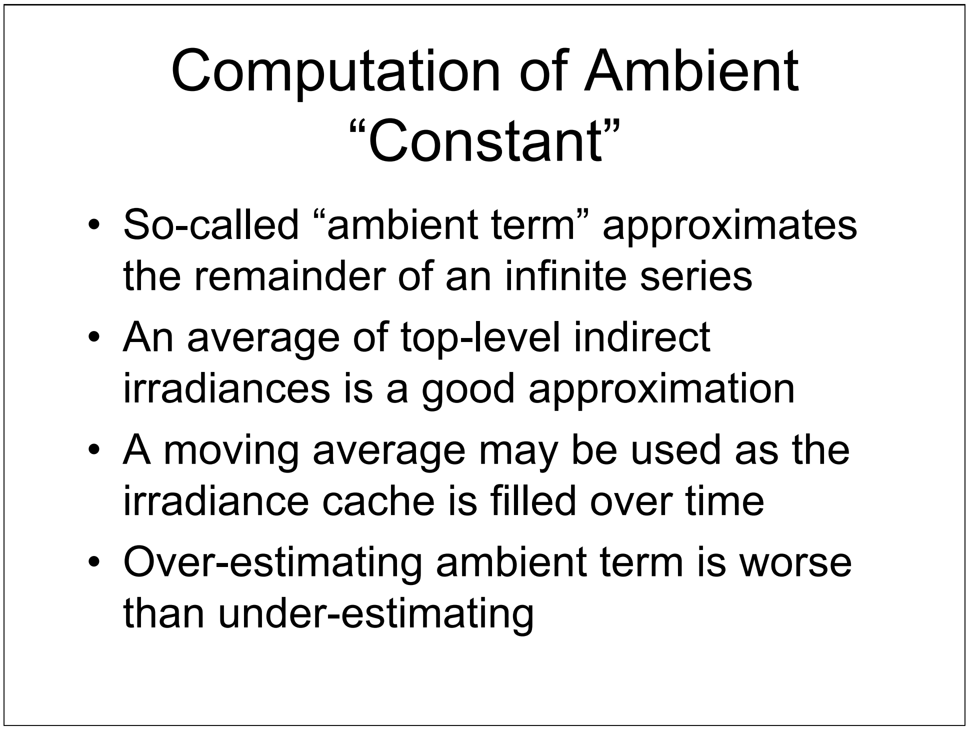
Implementation of Irradiance Caching in *Radiance*

Greg Ward
Dolby Canada



Irradiance Cache Enhancements in *Radiance*

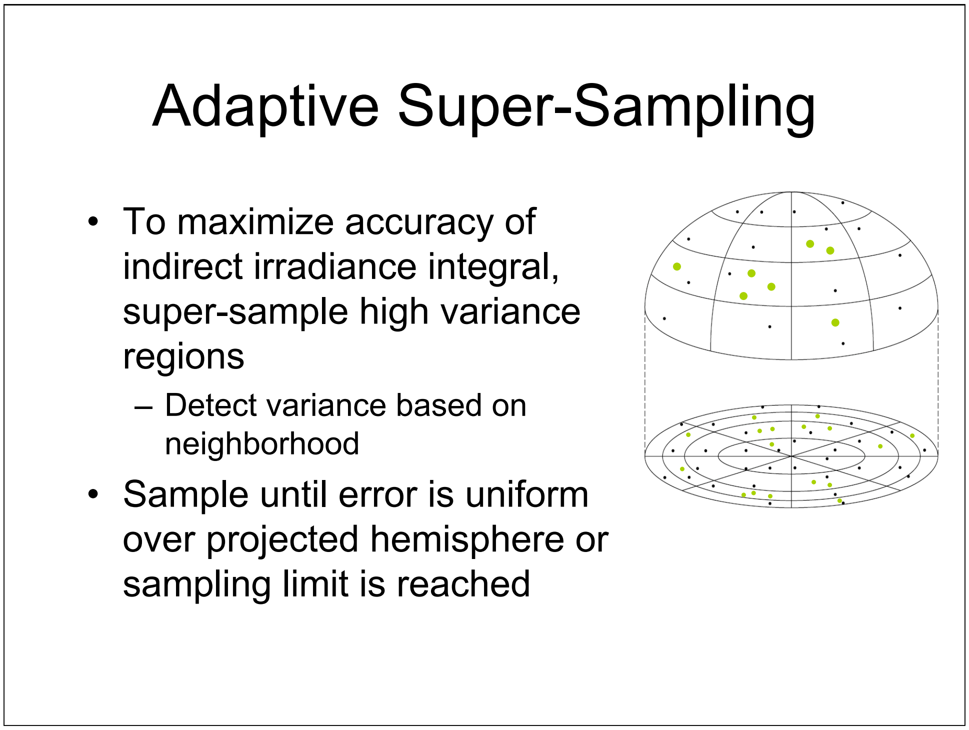
1. Computation of ambient “constant”
2. Adaptive super-sampling on hemisphere
3. Maximum and minimum record spacing
4. Gradient limit on record spacing
5. Bump maps using rotation gradient
6. Options for excluding surfaces/materials
7. Record sharing for multiprocessors
8. Record sorting for cache coherency



Computation of Ambient "Constant"

- So-called "ambient term" approximates the remainder of an infinite series
- An average of top-level indirect irradiances is a good approximation
- A moving average may be used as the irradiance cache is filled over time
- Over-estimating ambient term is worse than under-estimating

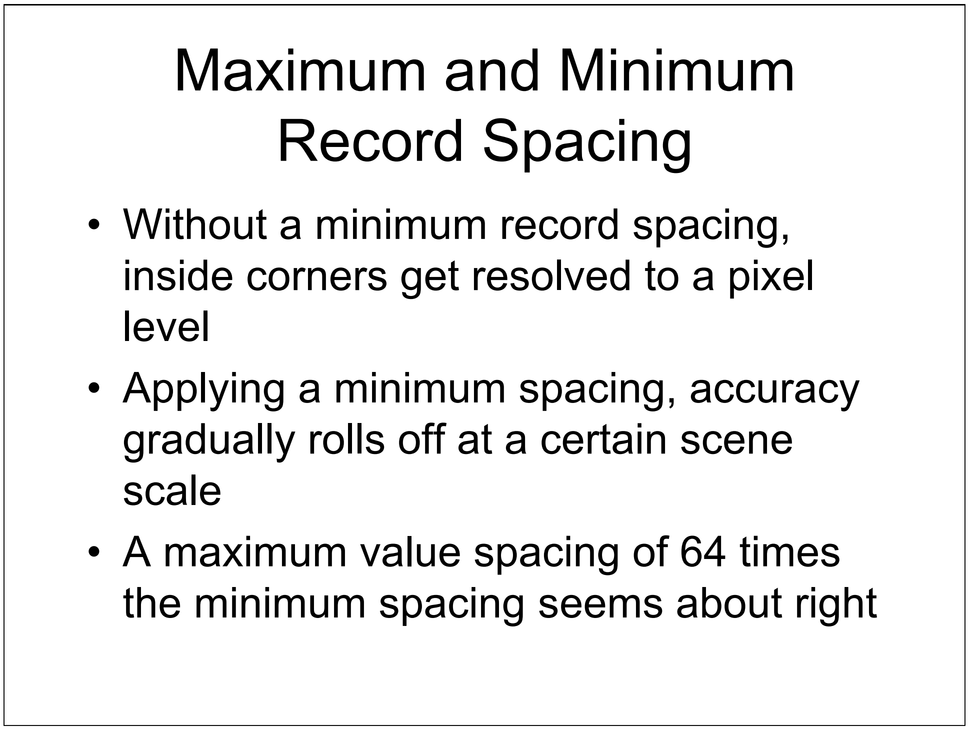
The -aw option in Radiance controls this function. As indirect values are collected, they overtake the initial, user-specified ambient term. It's one of those features that seemed like a good idea at the time, but in practice it doesn't get much use.



Adaptive Super-Sampling

- To maximize accuracy of indirect irradiance integral, super-sample high variance regions
  - Detect variance based on neighborhood
- Sample until error is uniform over projected hemisphere or sampling limit is reached

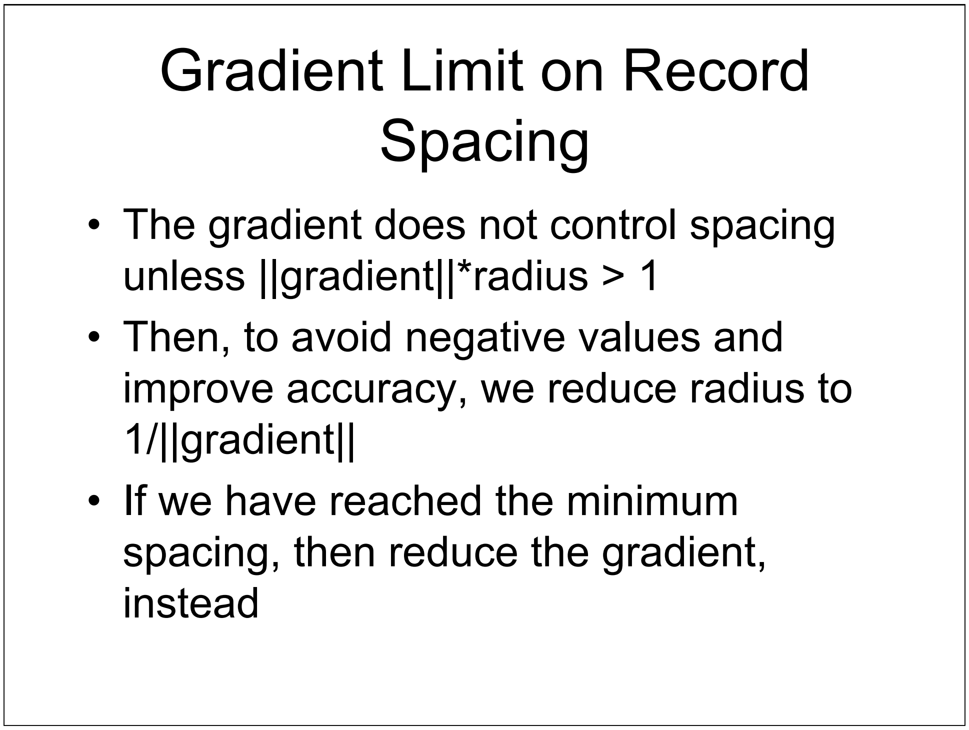
Controlled by the -as option in Radiance, this often-used optimization is especially effective in bright, daylighted interiors.



Maximum and Minimum Record Spacing

- Without a minimum record spacing, inside corners get resolved to a pixel level
- Applying a minimum spacing, accuracy gradually rolls off at a certain scene scale
- A maximum value spacing of 64 times the minimum spacing seems about right

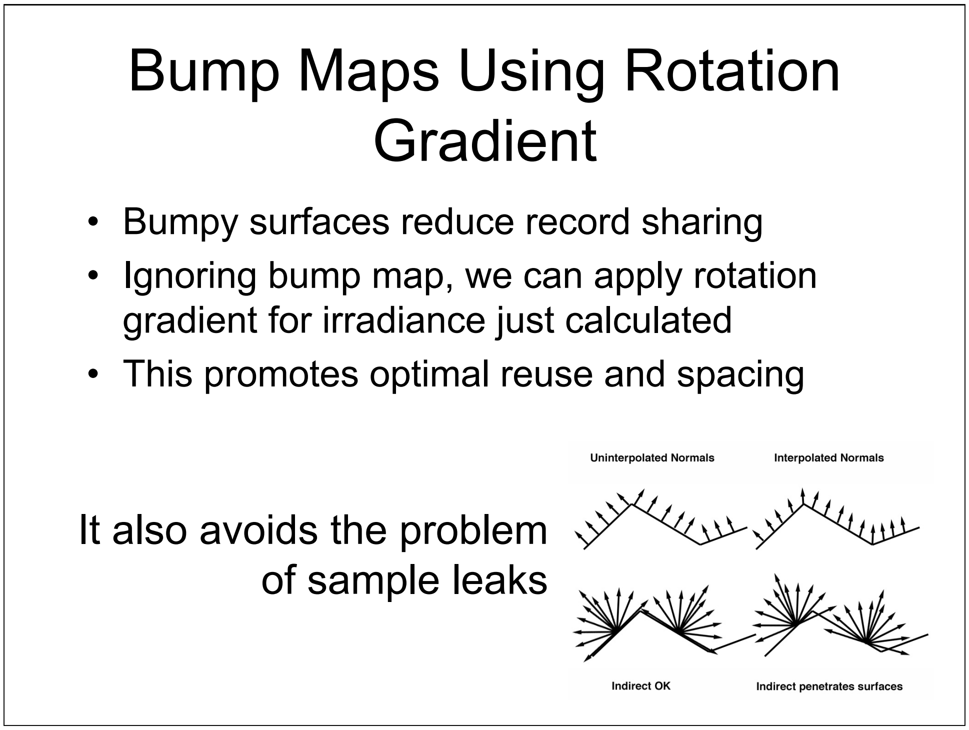
If available, ray pixel size may also be used to adjust record spacing, though this would tend to undermine view independence.



Gradient Limit on Record Spacing

- The gradient does not control spacing unless ||gradient||*radius > 1
- Then, to avoid negative values and improve accuracy, we reduce radius to 1/||gradient||
- If we have reached the minimum spacing, then reduce the gradient, instead

It is important *not* to use the gradient to determine record spacing in general, as we cannot know what the gradient is before sampling, and we don’t know how often to sample if the gradient dictates spacing. It is better to use a conservative metric that doesn’t depend on actual scene radiances, which may or may not behave as expected.



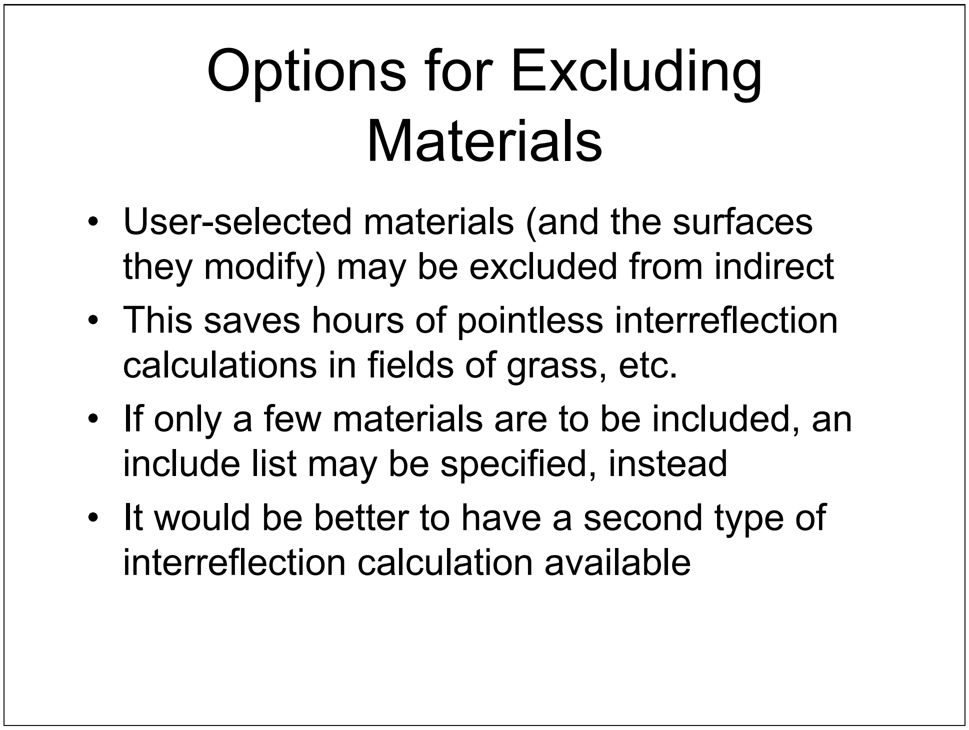
Bump Maps Using Rotation Gradient

- Bumpy surfaces reduce record sharing
- Ignoring bump map, we can apply rotation gradient for irradiance just calculated
- This promotes optimal reuse and spacing

It also avoids the problem of sample leaks

Uninterpolated Normals
Interpolated Normals
Indirect OK
Indirect penetrates surfaces

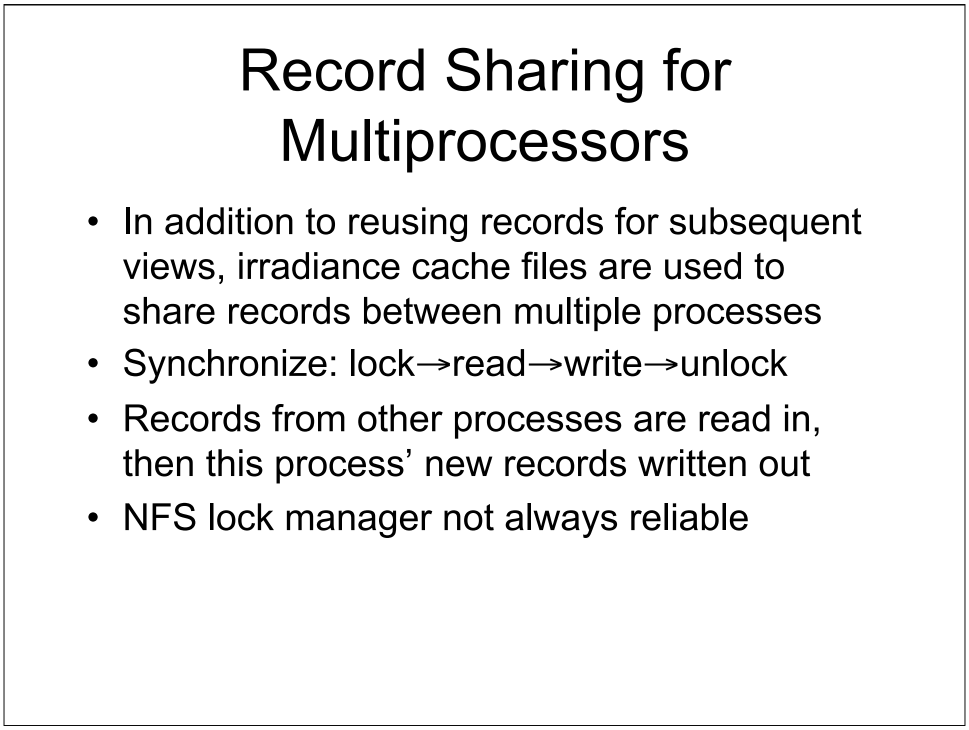
No-cost addition to irradiance gradient calculation -- just apply the perturbed surface normals on the final interpolation.



Options for Excluding Materials

- User-selected materials (and the surfaces they modify) may be excluded from indirect
- This saves hours of pointless interreflection calculations in fields of grass, etc.
- If only a few materials are to be included, an include list may be specified, instead
- It would be better to have a second type of interreflection calculation available

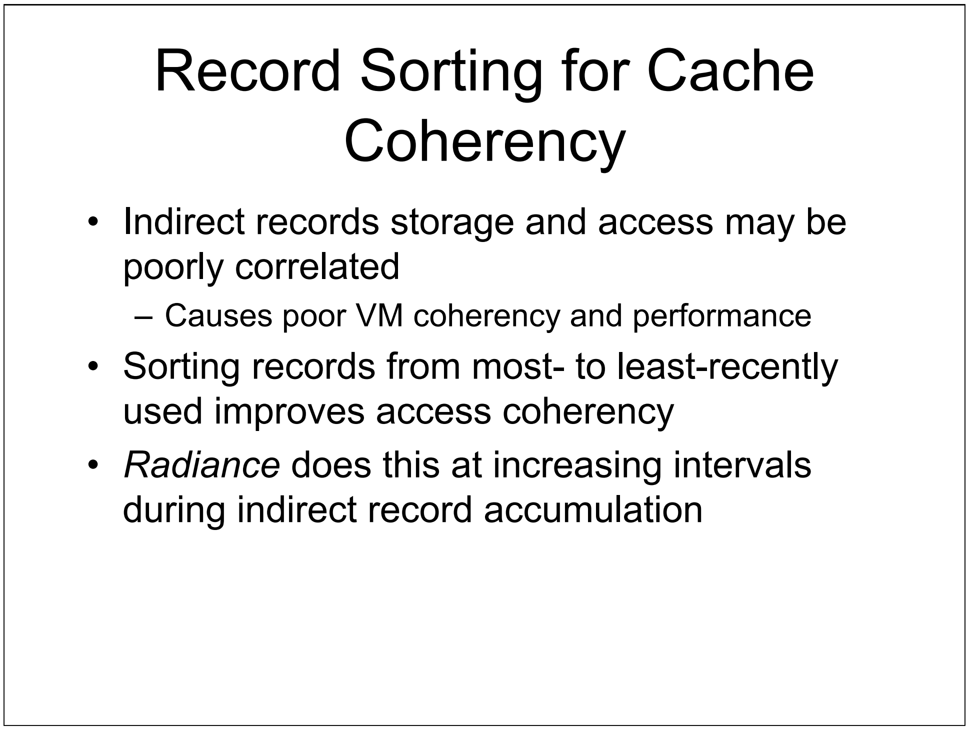
The -ae option excludes a single material (and its assigns), whereas -aE excludes all materials listed in a file. The -ai and -aI options may be used to specify included materials, instead.



Record Sharing for Multiprocessors

- In addition to reusing records for subsequent views, irradiance cache files are used to share records between multiple processes
- Synchronize: lock→read→write→unlock
- Records from other processes are read in, then this process' new records written out
- NFS lock manager not always reliable

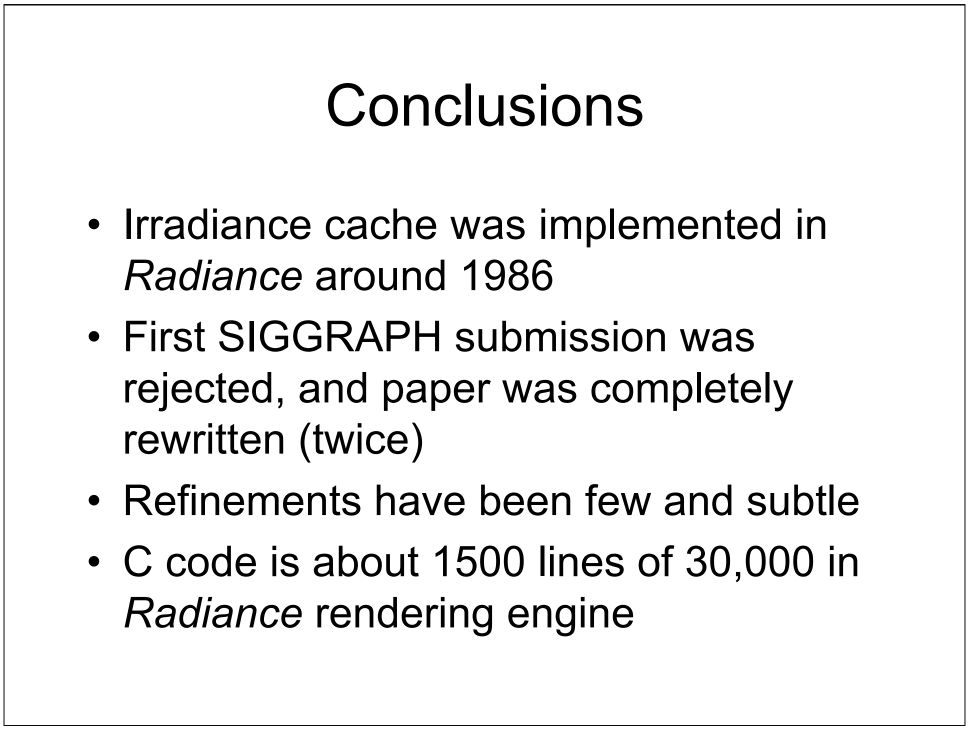
This technique works with a fixed buffer size of about 13 records up to 10 processors or so, then a larger buffer works better. At some point, I would like to implement a client-server or broadcast model to reduce overhead and avoid problems with NFS.



Record Sorting for Cache Coherency

- Indirect records storage and access may be poorly correlated
  - Causes poor VM coherency and performance
- Sorting records from most- to least-recently used improves access coherency
- *Radiance* does this at increasing intervals during indirect record accumulation

There are additional caveats in cases where indirect values are being shared. Since a field is stored in each record for when it was last used, and changing this would undermine value sharing, we turn this optimization off when more than a third of the records are being shared across processes. The biggest savings occur in processes that exceed physical memory, which causes VM thrashing.



Conclusions

- Irradiance cache was implemented in *Radiance* around 1986
- First SIGGRAPH submission was rejected, and paper was completely rewritten (twice)
- Refinements have been few and subtle
- C code is about 1500 lines of 30,000 in *Radiance* rendering engine