SIGGRAPH2007

Implementation Details

SIGGRAPH2007

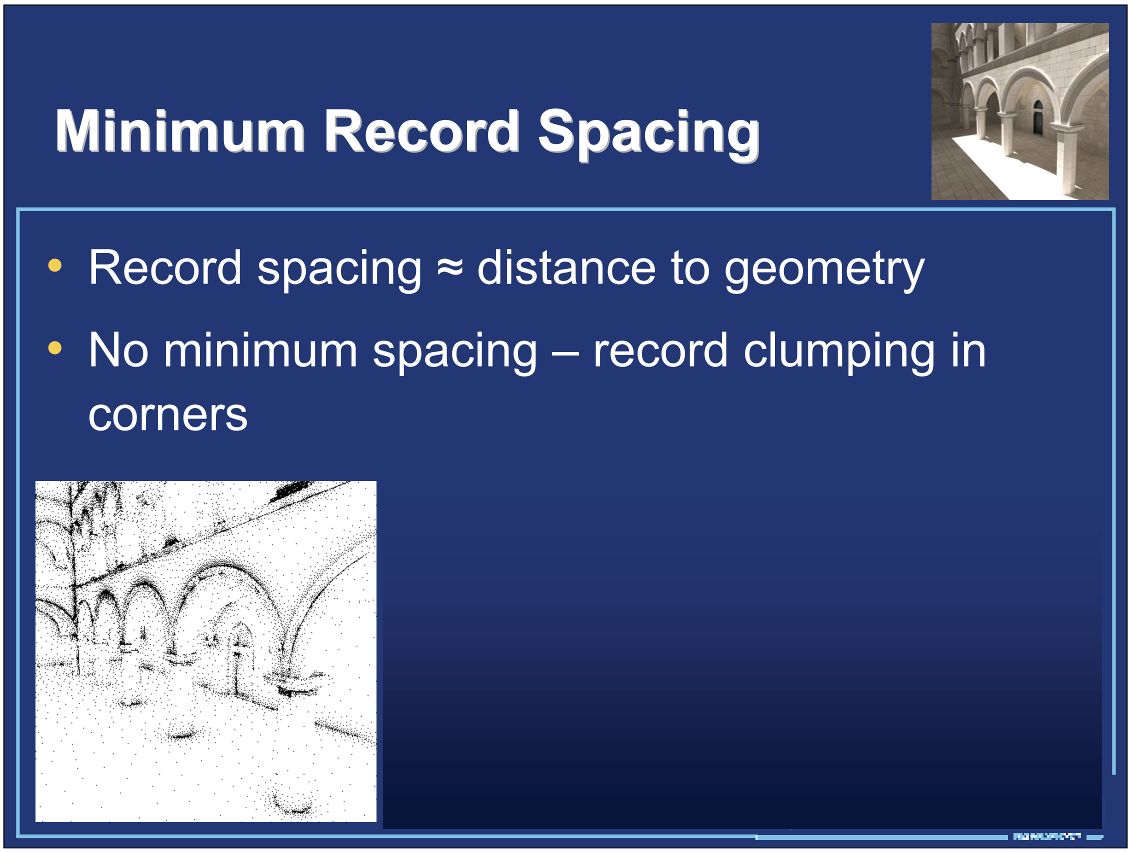
Jaroslav Křivánek

ČVUT v Praze – CTU Prague

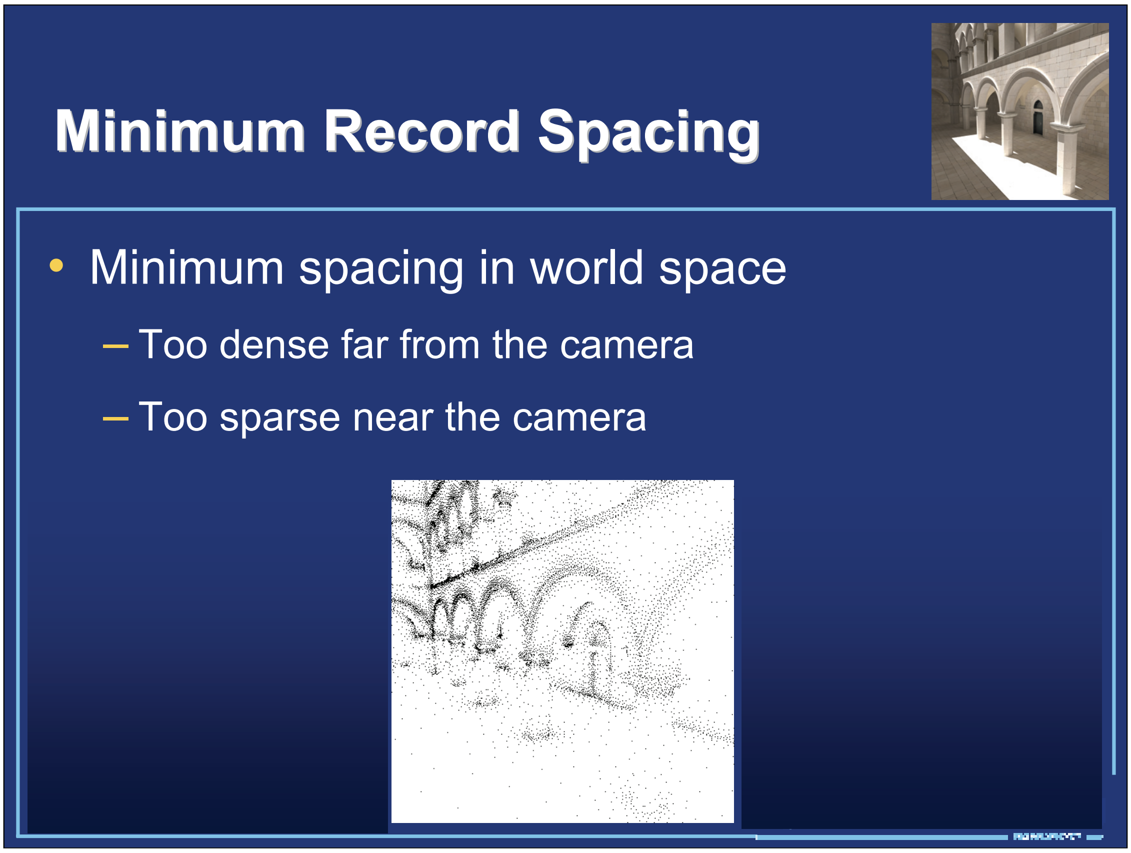
In this part of the course, we will discuss a number of tricks that make irradiance caching a reliable algorithm. At the first sight, they might not make much sense, but it is quite difficult to get irradiance caching produce artifact-free images without using these tricks.

Implementation Details

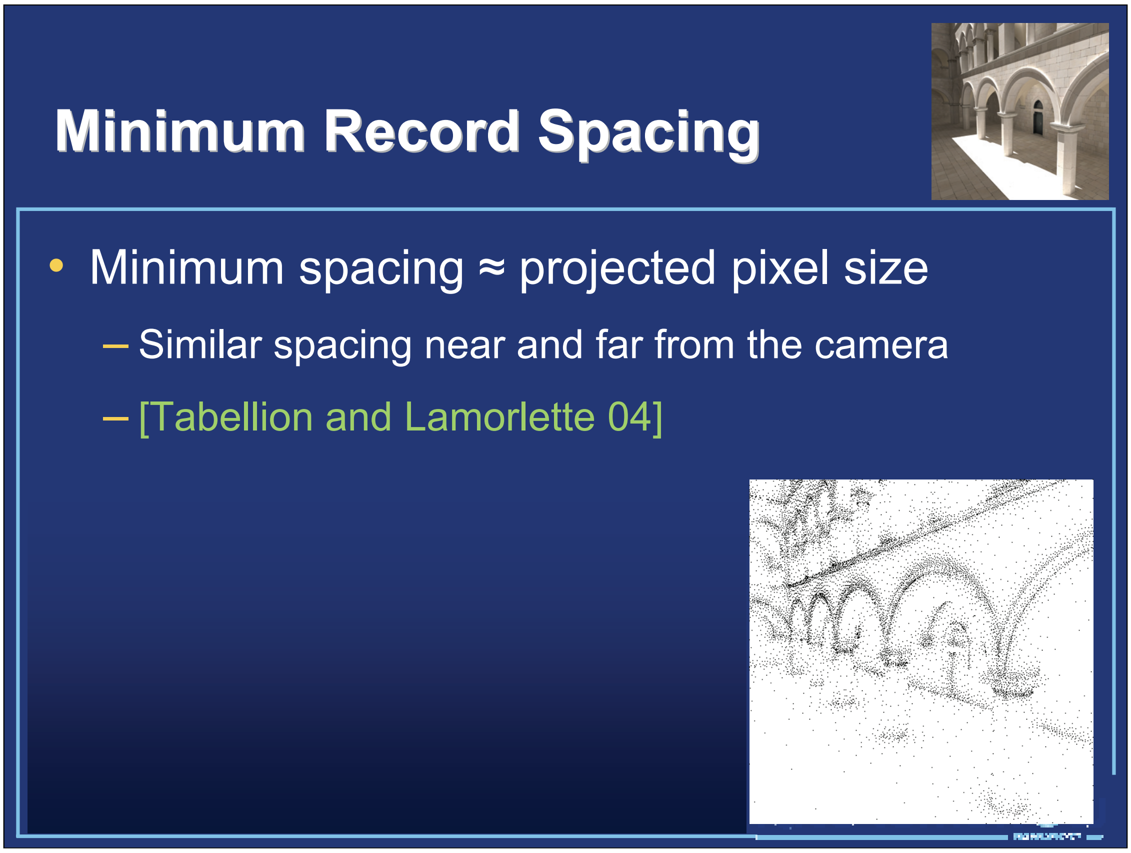
- Minimum record spacing
- Missing small geometry
- Neighbor clamping
- Ray leaking
- Weighting function
- Image sampling



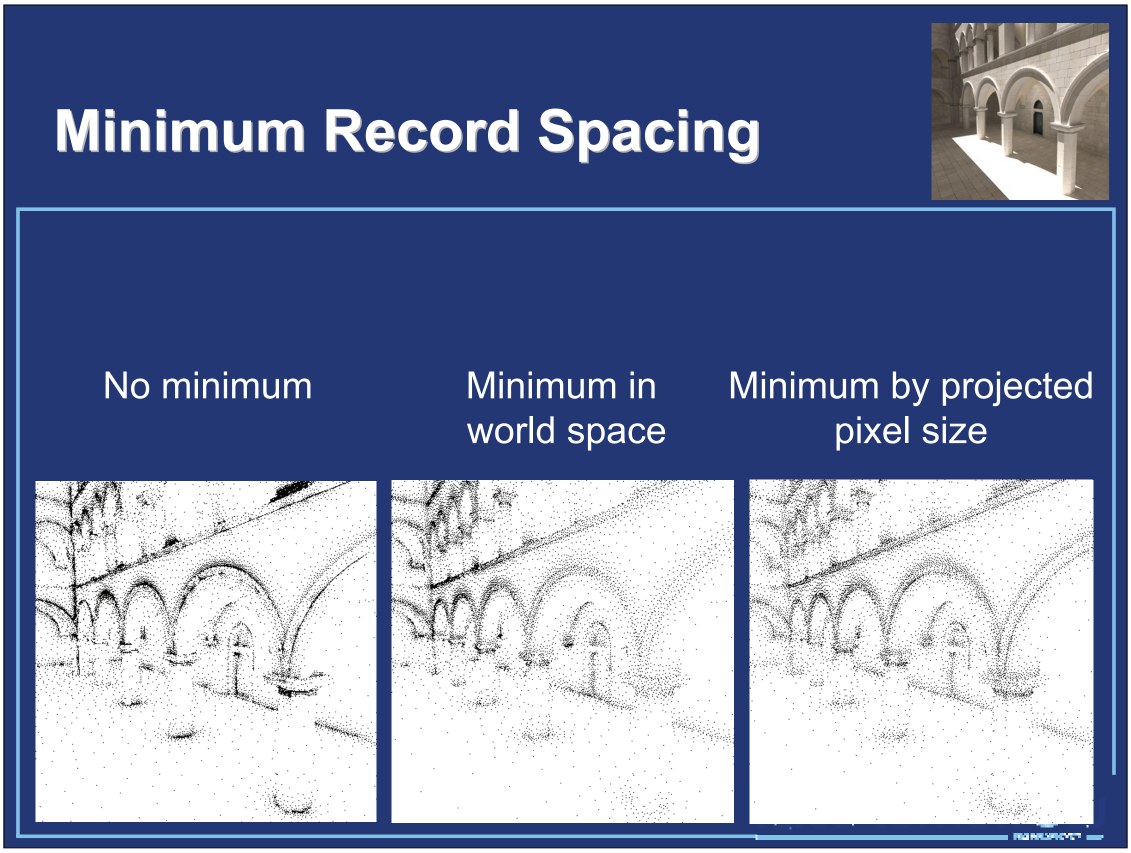
Remember that the spacing of irradiance records is given by the mean distance to the neighboring geometry (and also by the object curvature, which we disregard in this discussion). If you do not impose any minimum limit on the spacing, irradiance caching will spend most of the time generating too many records around edges and corners. To avoid this problem, it is a good idea to impose a minimum distance between the records. This can be done by setting some minimum threshold R_{min} on the R_i value of a record.



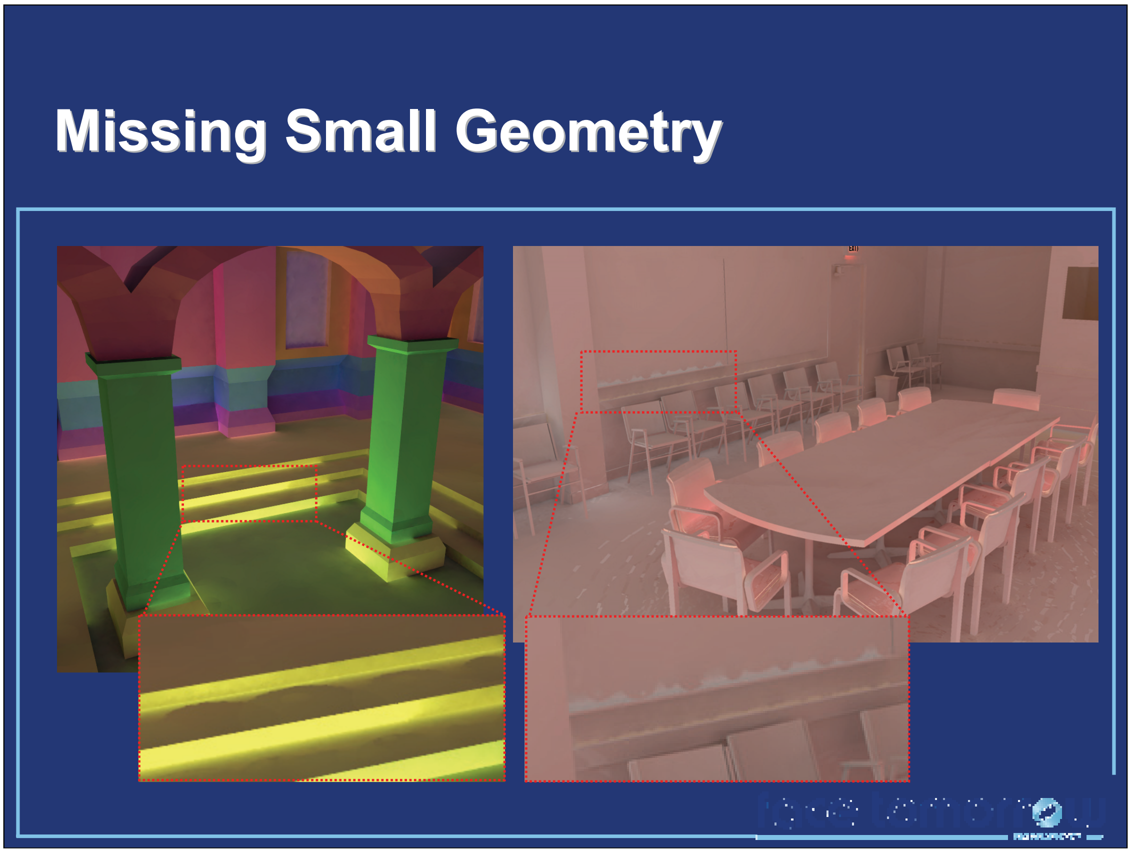
One possibility is to limit the spacing in world space, by fixing the threshold, R_{min}, to the same value all over the scene. In *Radiance*, R_{min} is specified as a fraction of the scene size. This way of limiting the record spacing tends to generate too few radiance cache records near the camera and too many records far away.



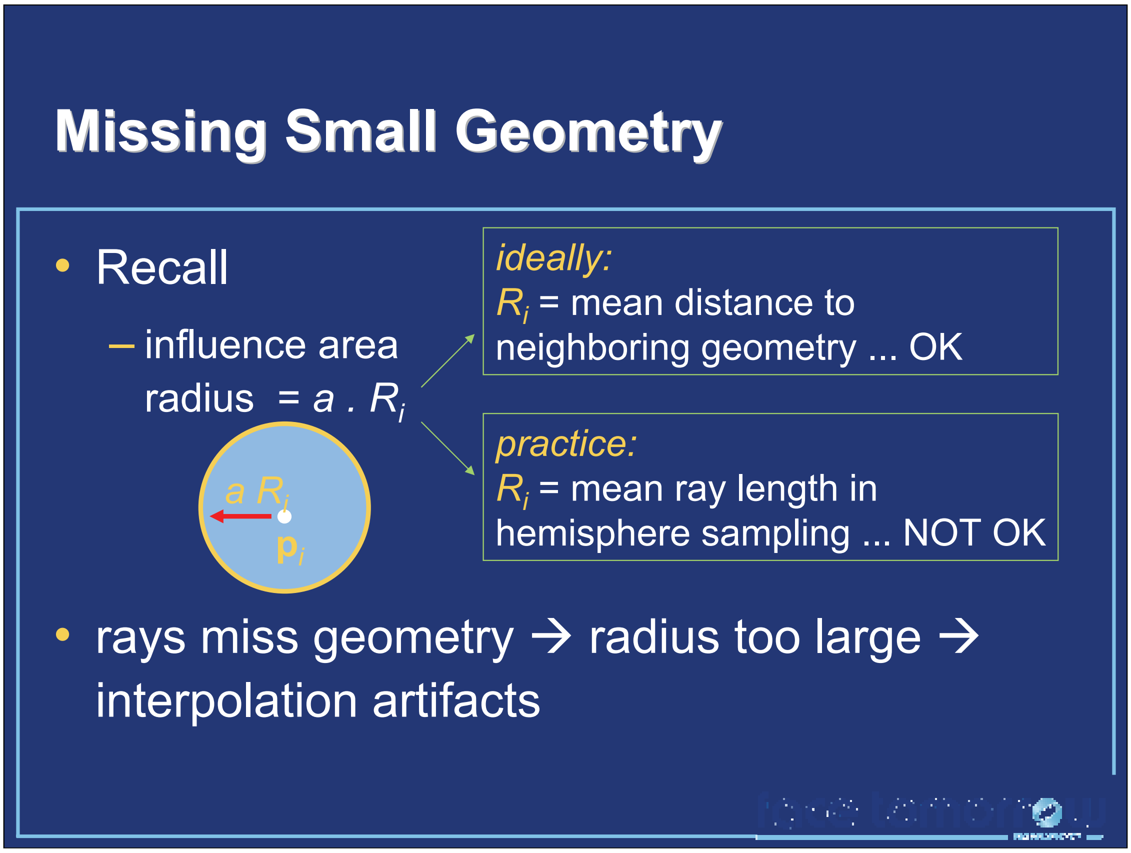
A better idea, proposed by Tabellion and Lamorlette [2004] is to use a multiple of the projected pixel size for the threshold R_{min}. Good values for R_{min} range between 1.5x and 3x the projected pixel size.



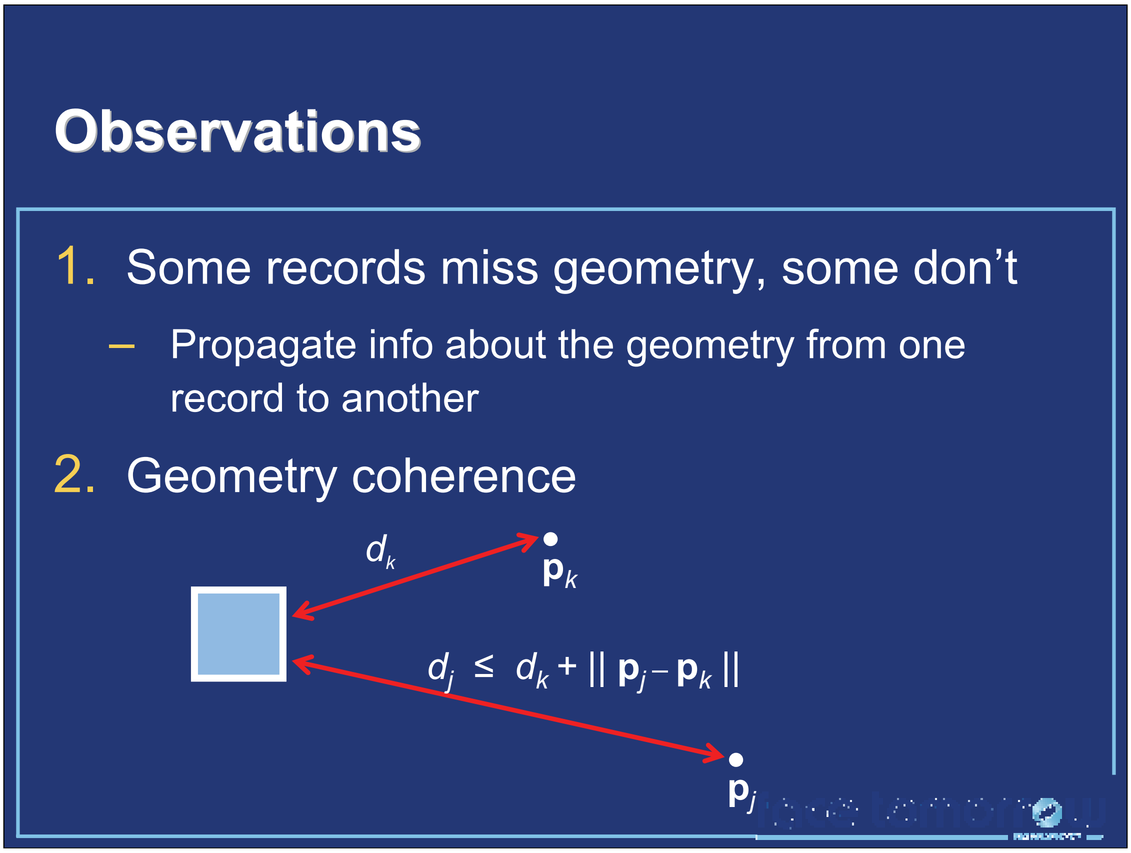
Especially for exterior scenes, it is also important to limit the maximum value of R_i. In *Radiance*, this maximum is 64 times the minimum. Tabellion and Lamorlette [2004] use the maximum of 10x the projected pixel size.



A common problem in irradiance caching is that rays in the stochastic hemisphere sampling miss geometry features in the scene. This can produce clearly visible image artifacts.



Ideally, record spacing would be determined by the mean distance to the neighboring geometry. In practice, this distance is determined as the mean of the ray lengths in hemisphere sampling. If rays miss some geometry, the resulting mean distance is overestimated and we can see discontinuities in the resulting images due to interpolation.



A reliable way of detecting the over-estimated mean distance due to missing geometry in hemisphere sampling is based on two observations.

1. Because only few record usually suffer from the overestimated mean distance, we could use the distance estimate at other records to rectify the overestimated distance.
2. Distances obey the triangle inequality. If one record, $\mathbf{p}_k$, is at the distance d_k from some geometry feature, then another record, $\mathbf{p}_j$, cannot be farther from this geometry feature than $d_k + \| \mathbf{p}_j - \mathbf{p}_k \|$.

If we replace d_k and d_j by the mean distance R_k, R_j, we can detect suspicious cases by comparing R_k, R_j to the distance between the two records to verify if the triangle inequality holds. Strictly speaking, this should only work if R_k, R_j, was the distance to the *nearest* geometry feature, but in practice this works fine even for *mean* distance. We call this heuristic 'neighbor clamping'.

Neighbor Clamping

- Upon addition of a new record, *j*, in the cache:
  - **for** *k* in nearby records
    - $R_j = \min\{ R_j , R_k + \| \mathbf{p}_j - \mathbf{p}_k \| \}$

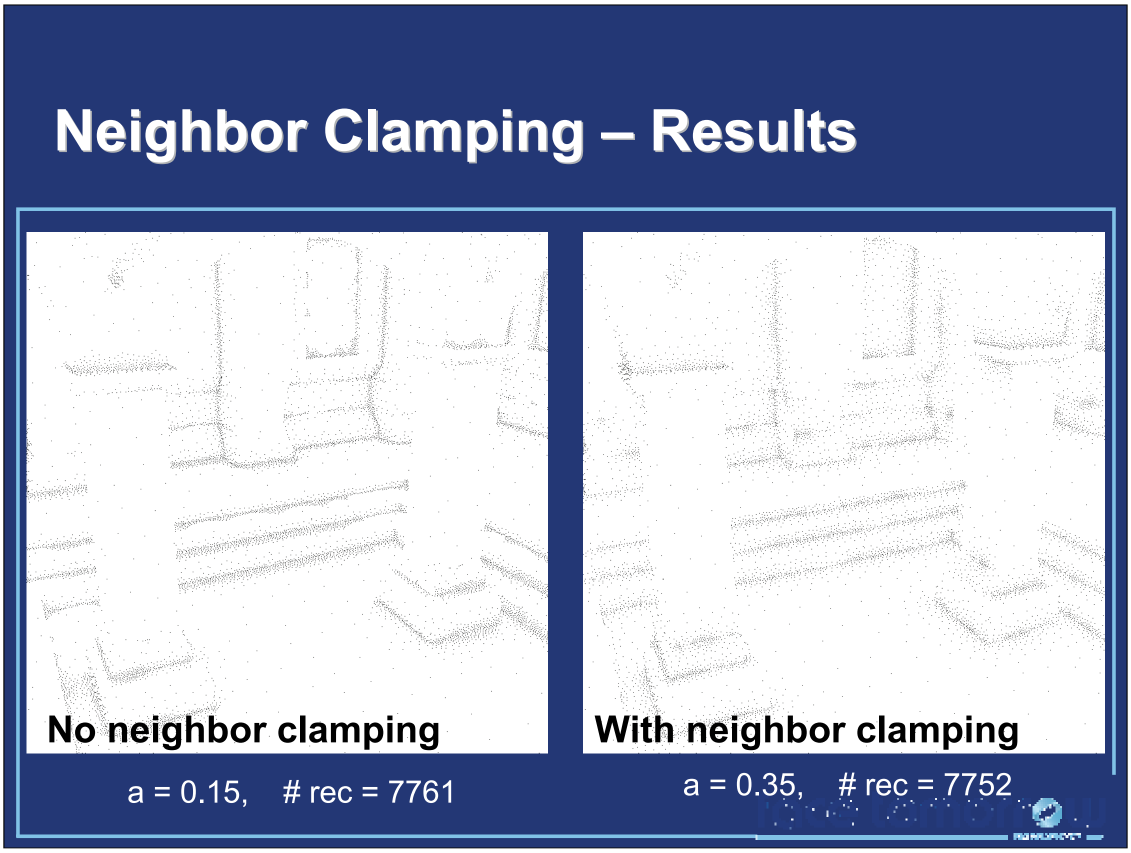
  Clamp new record's *R* by its neighbors' *R*

  - **for** *k* in nearby records
    - $R_k = \min\{ R_k, R_j + \| \mathbf{p}_j - \mathbf{p}_k \| \}$

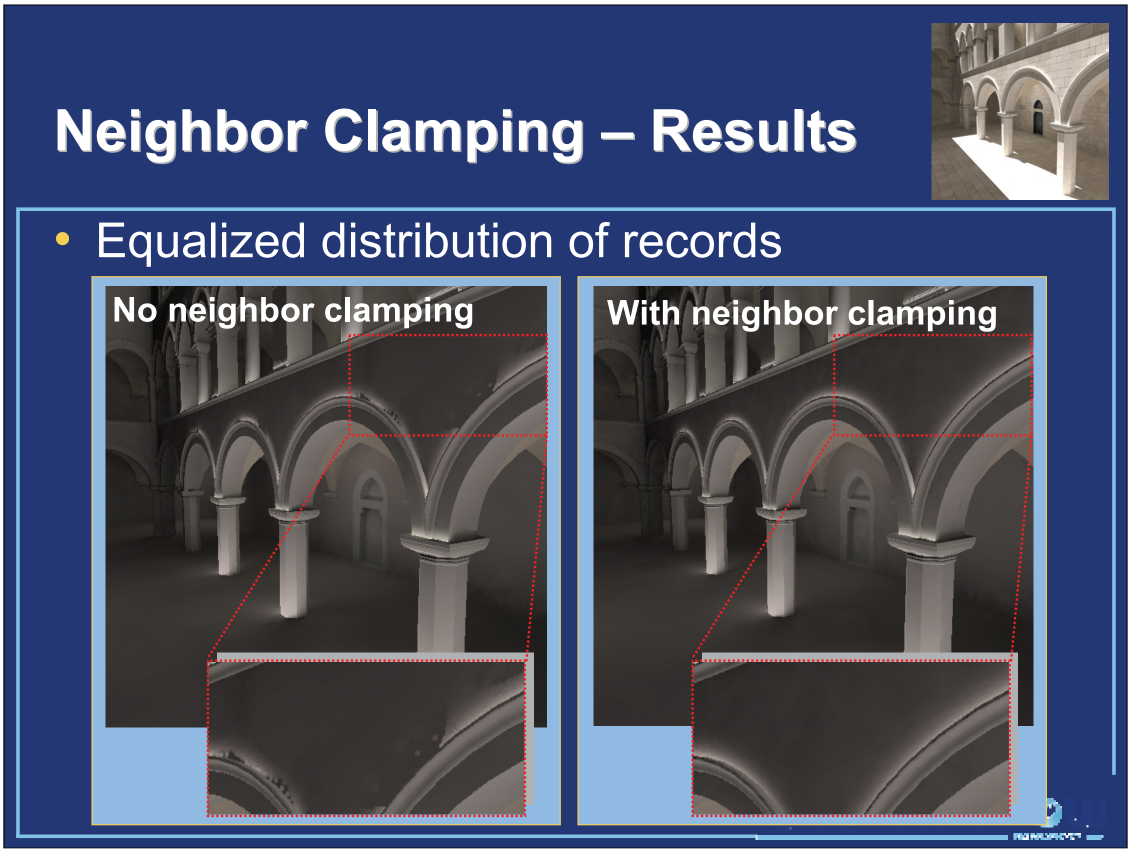
  Clamp neighbors' *R* by the new record's *R*

Here is how we proceed in practice. When a new record, *j*, is added to the cache, we first locate all existing records whose area of influence overlap with the area of influence of the record being added. (That is to say, all records *k*, such that $\| \mathbf{p}_j - \mathbf{p}_k \| < R_j + R_k$). Then for all those records, we clamp the R_j value of the new record: $R_j = \min\{ R_j , R_k + \| \mathbf{p}_j - \mathbf{p}_k \| \}$. This enforces the triangle inequality. After that, we use the clamped R_j value of the new record to clamp the R_k values of the nearby records. This second step enforces the *transitivity* of triangle inequality.

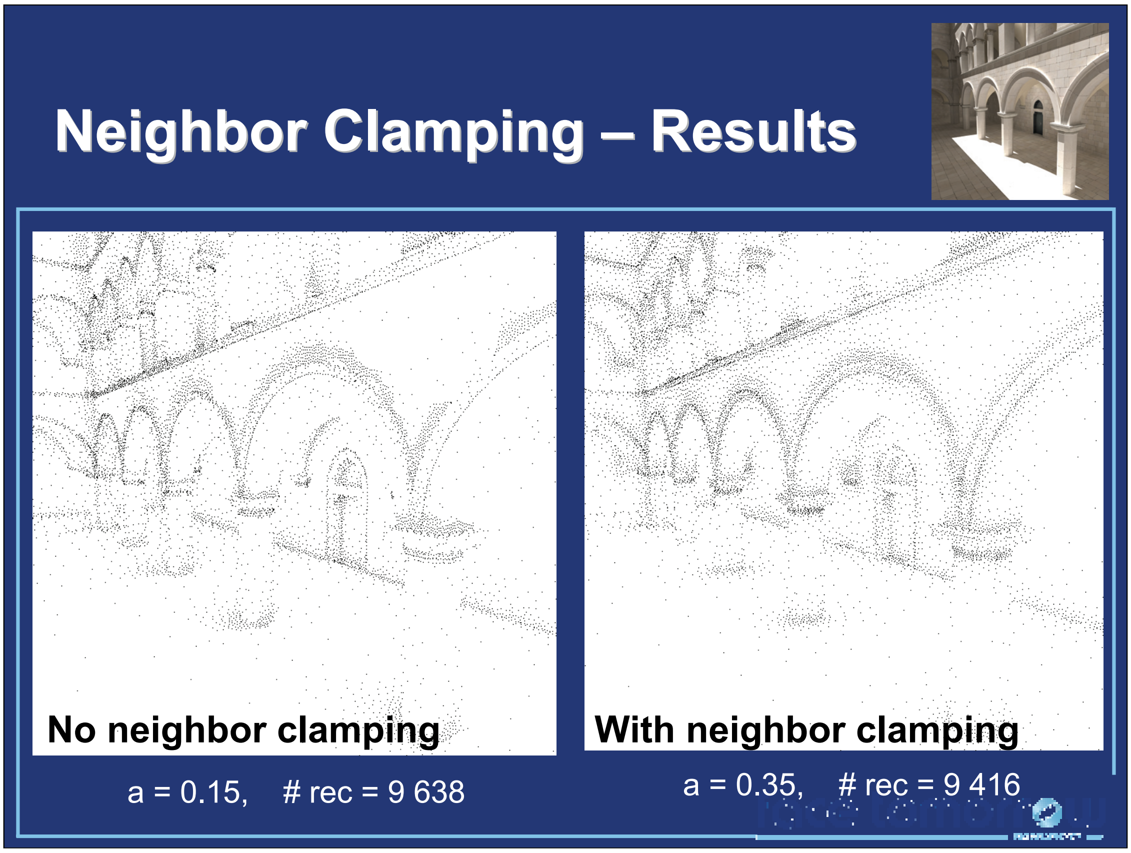
And voilà, the artifacts due to the over-estimated mean distance are gone.



With neighbor clamping, the spacing between the records is equalized. It falls off gradually as we move away from the geometry features.

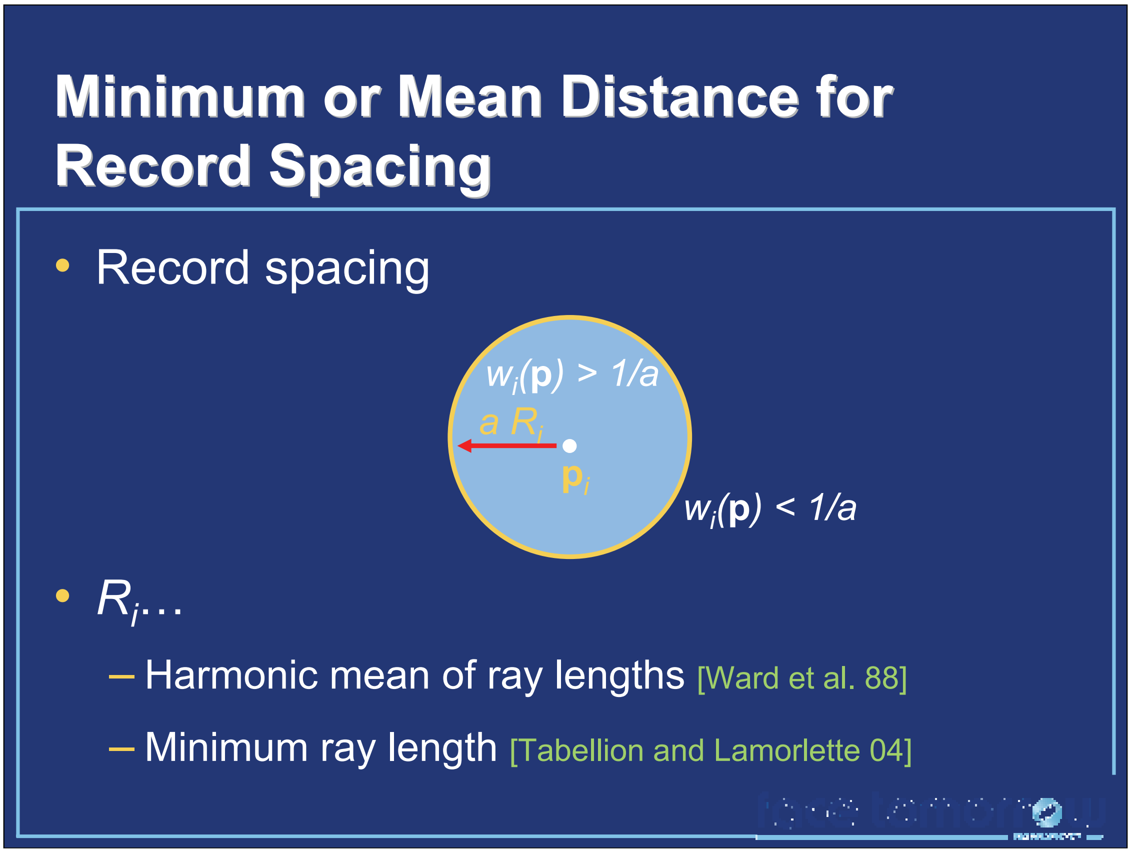


In the Sponza atrium scene, the benefit of neighbor clamping is even more apparent.

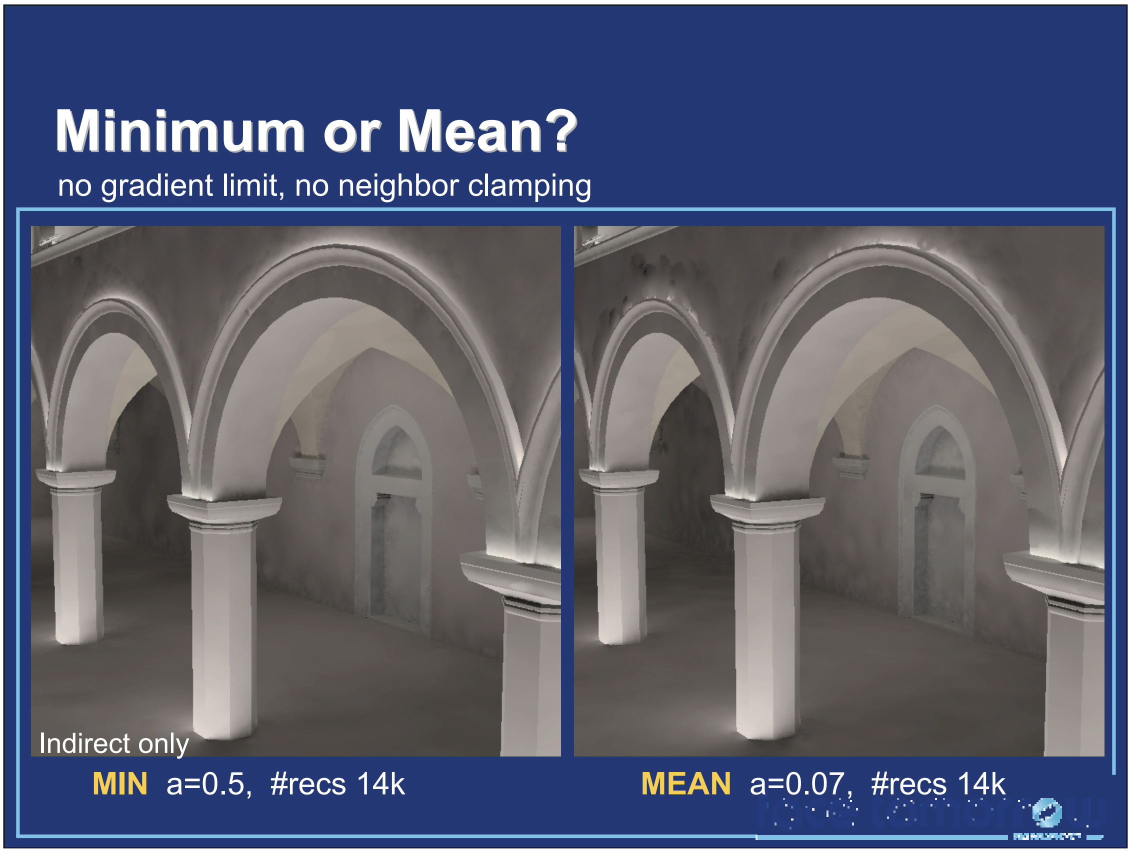


Record spacing is equalized and indirect illumination is properly sampled around the cornices above the arches.

To conclude, irradiance caching produces image artifacts when the mean distance to geometry is overestimated. Neighbor clamping reliably detects and corrects the overestimated mean distance, thereby suppressing these artifacts.



In the irradiance caching implementation in *Radiance* [Ward et al. 1988], the record spacing is based upon the *mean* distance to neighboring geometry. As shown on the previous slide, this tends to miss some geometry features. We have proposed neighbor clamping to resolve these problems at EGSR in 2006 [Křivánek et al. 2006]. In 2004, Tabellion and Lamorlette [2004] used the minimum distance instead of the mean distance to resolve these problems. Let us see how irradiance caching behaves when using the minimum and mean distance.

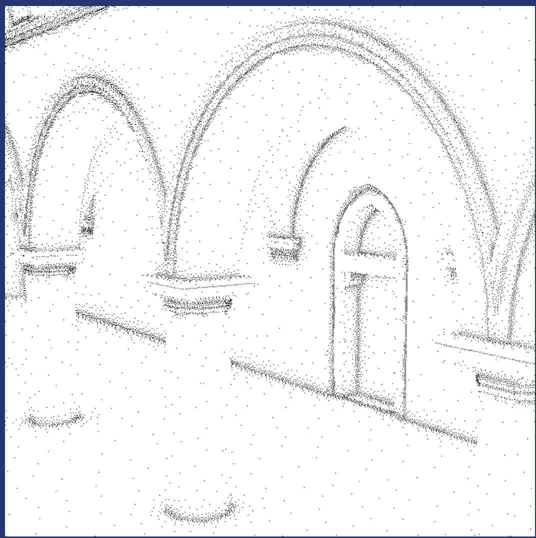


First, when there is *no* gradient limit on record spacing and neighbor clamping is *not* used, then the minimum distance indeed produces much better images than the mean distance.

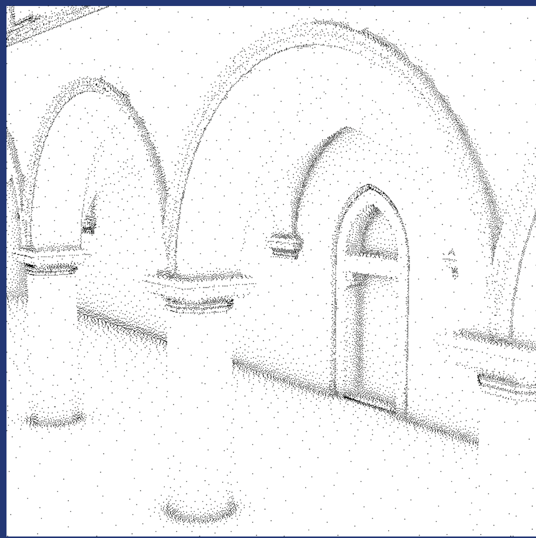
(See Greg Ward's slides on Radiance implementation for more information about the gradient limit on record spacing.)

Minimum or Mean?

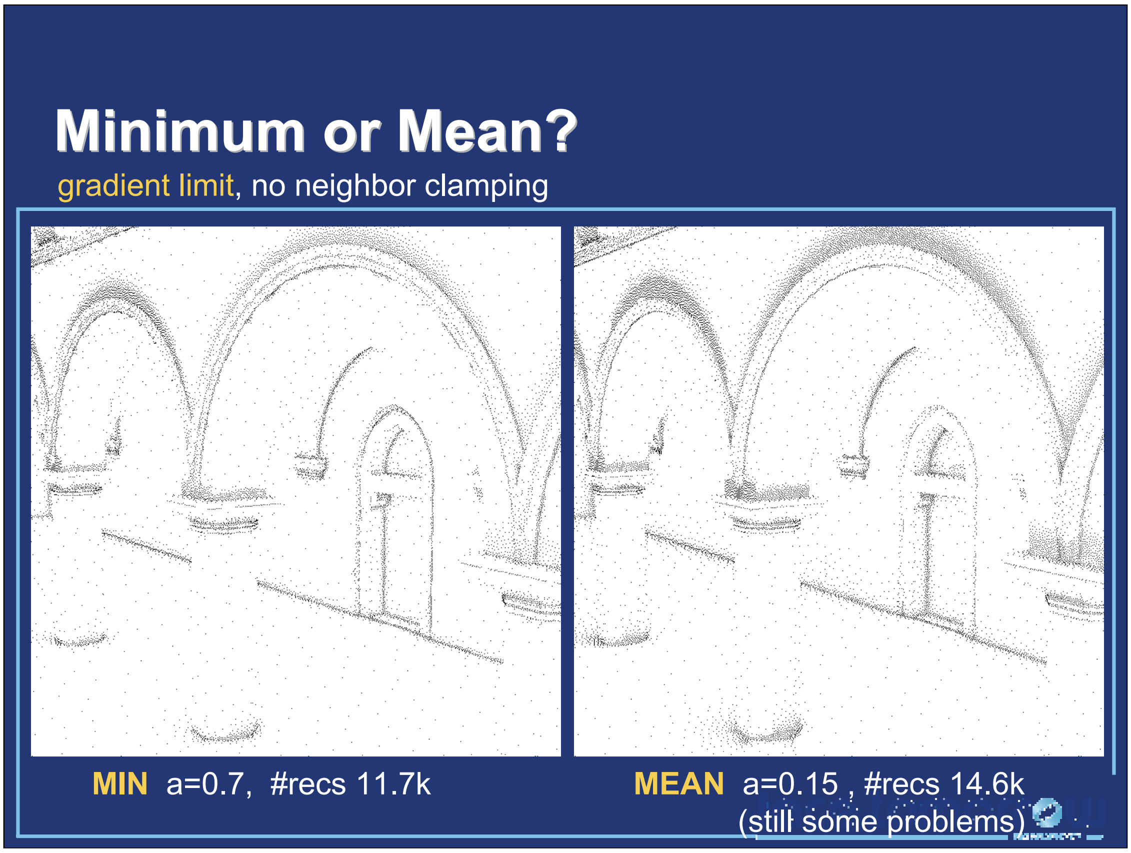
no gradient limit, no neighbor clamping



MIN a=0.5, #recs 14k

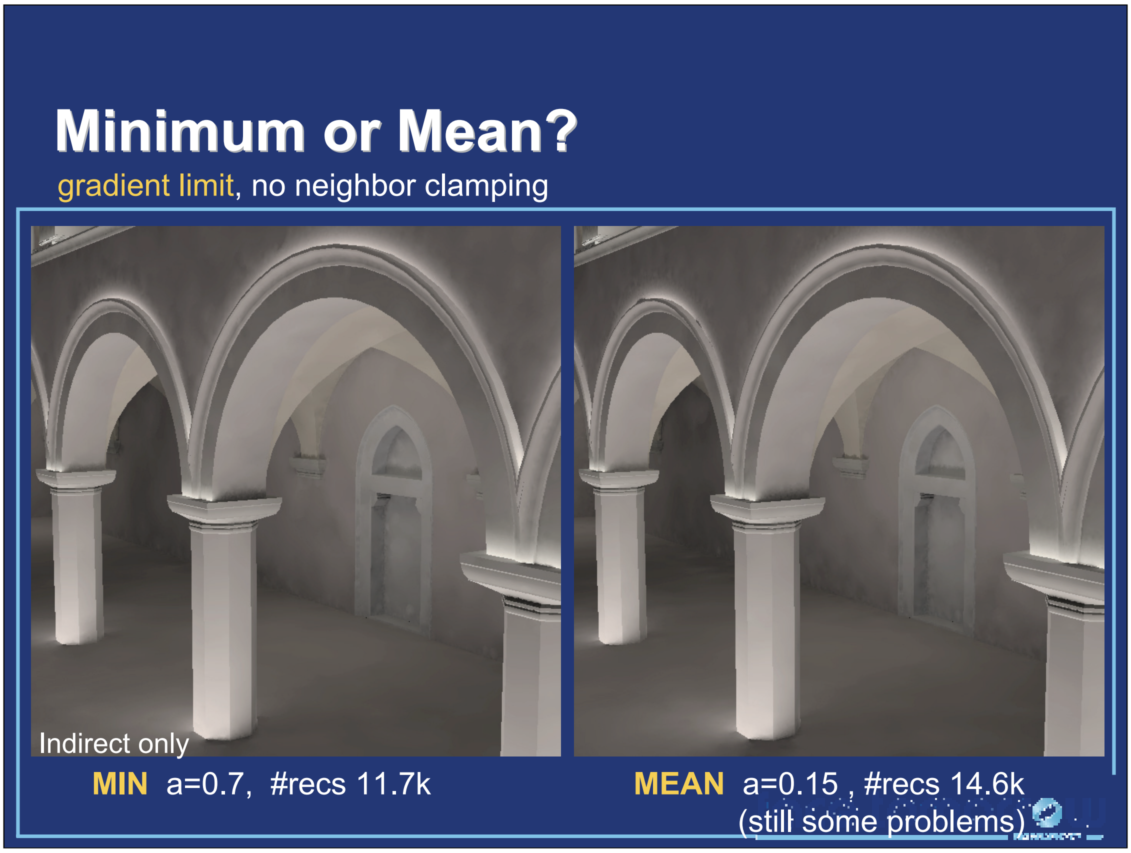


MEAN a=0.07, #recs 14k

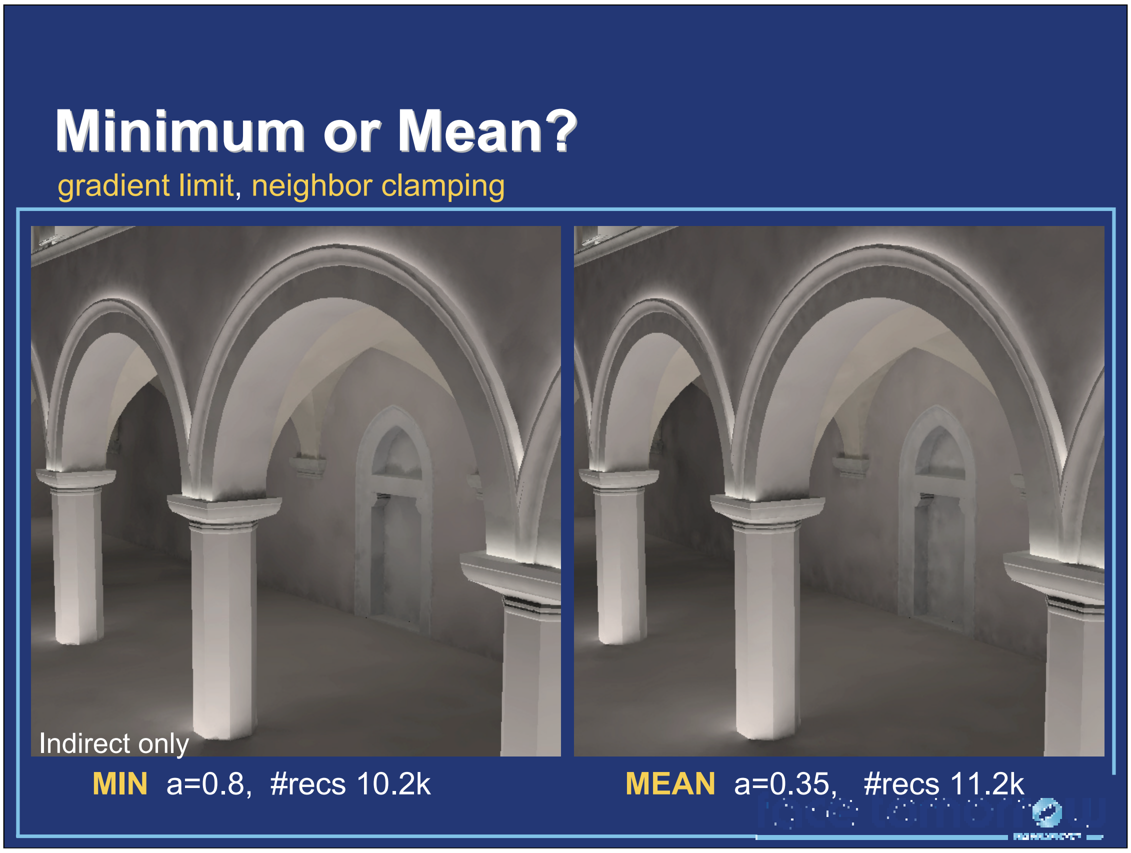


If we limit record spacing by the translational gradient, we get much more records in the high-gradient areas around the cornices.

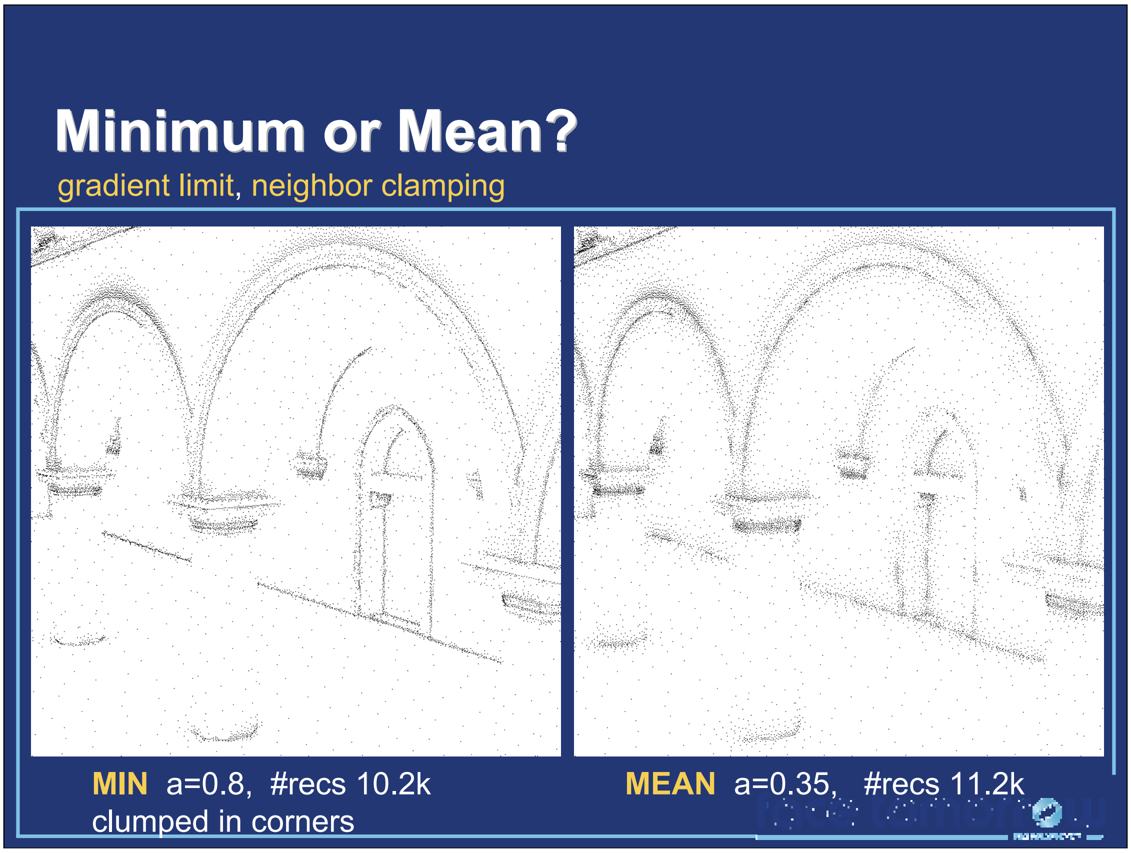
This is also a nice example of the gradient limit on record spacing.



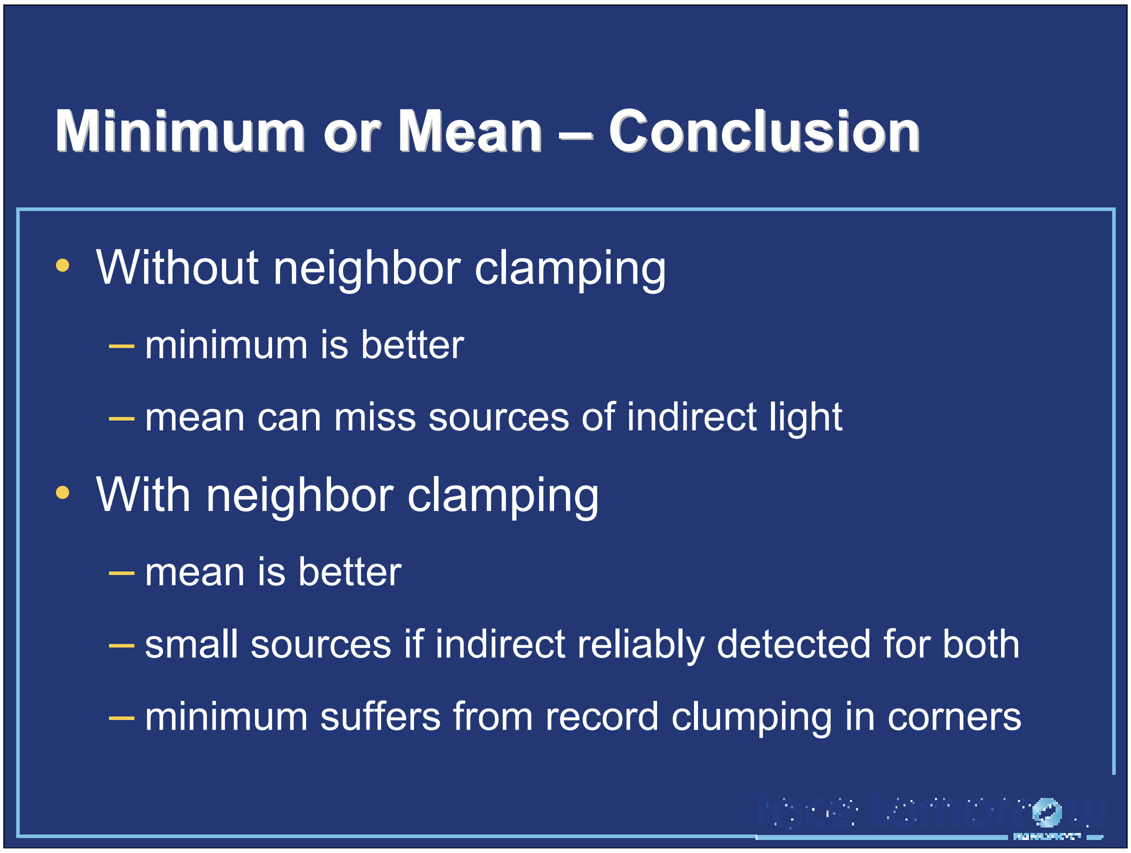
But still, even with the gradient limited spacing, using the mean distance leaves some artifacts around the cornices.



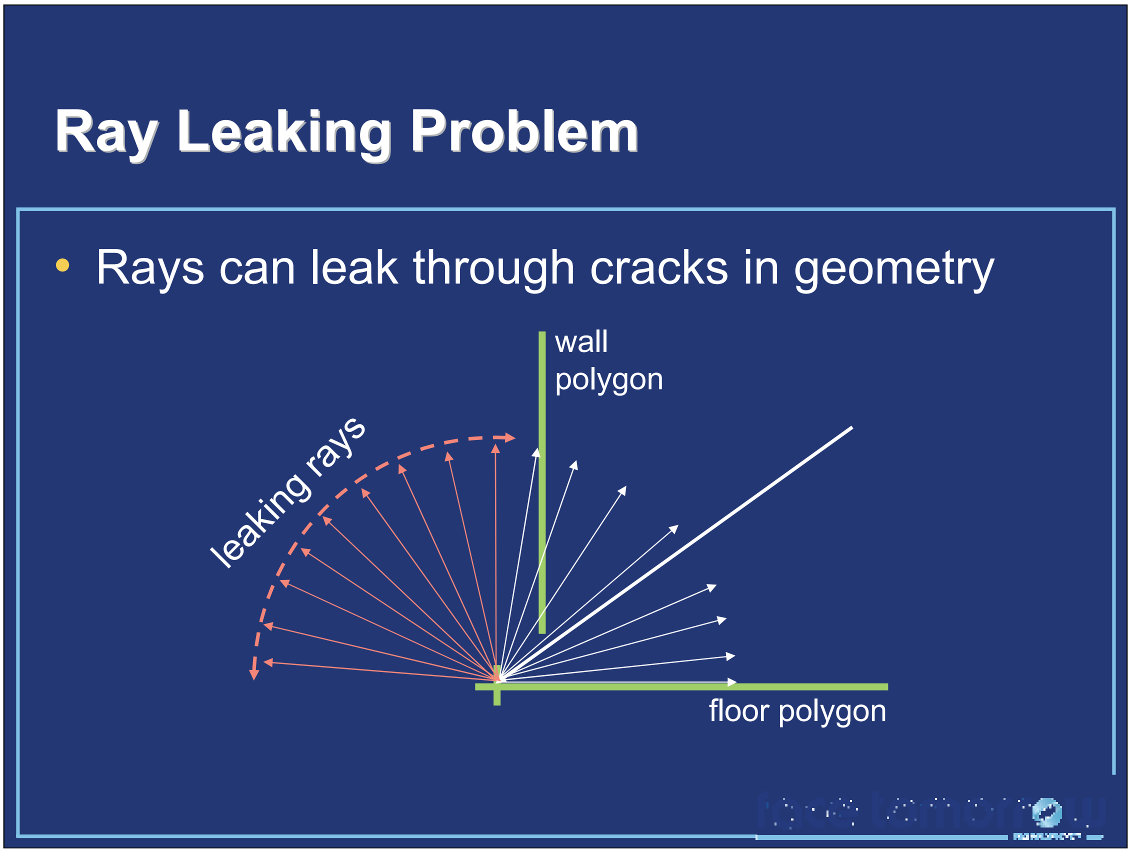
However, when we turn on neighbor clamping, the artifacts are gone.



Looking at the record distribution, we see that with the mean distance, record spacing falls of gradually as we move away from the geometry, whereas with the minimum distance, records tend to concentrate in the corners.



In our experience, the gradual falloff of the spacing is desirable. When gradient limit on record spacing and neighbor clamping is used, then the mean distance produces better images with the same number of records than the minimum distance.

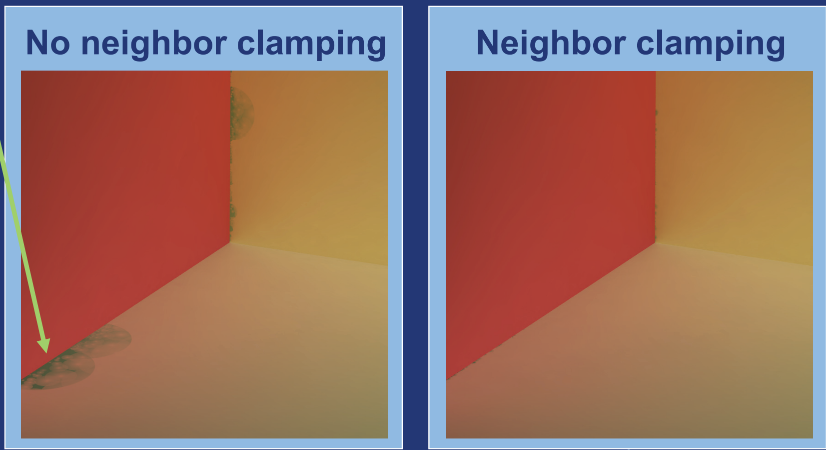


Another serious problem of irradiance caching occurs when handling scenes with small cracks between polygons. Such scenes are quite common in practice – either because the scene is not well modeled or because of a limited numerical precision in the exported scene.

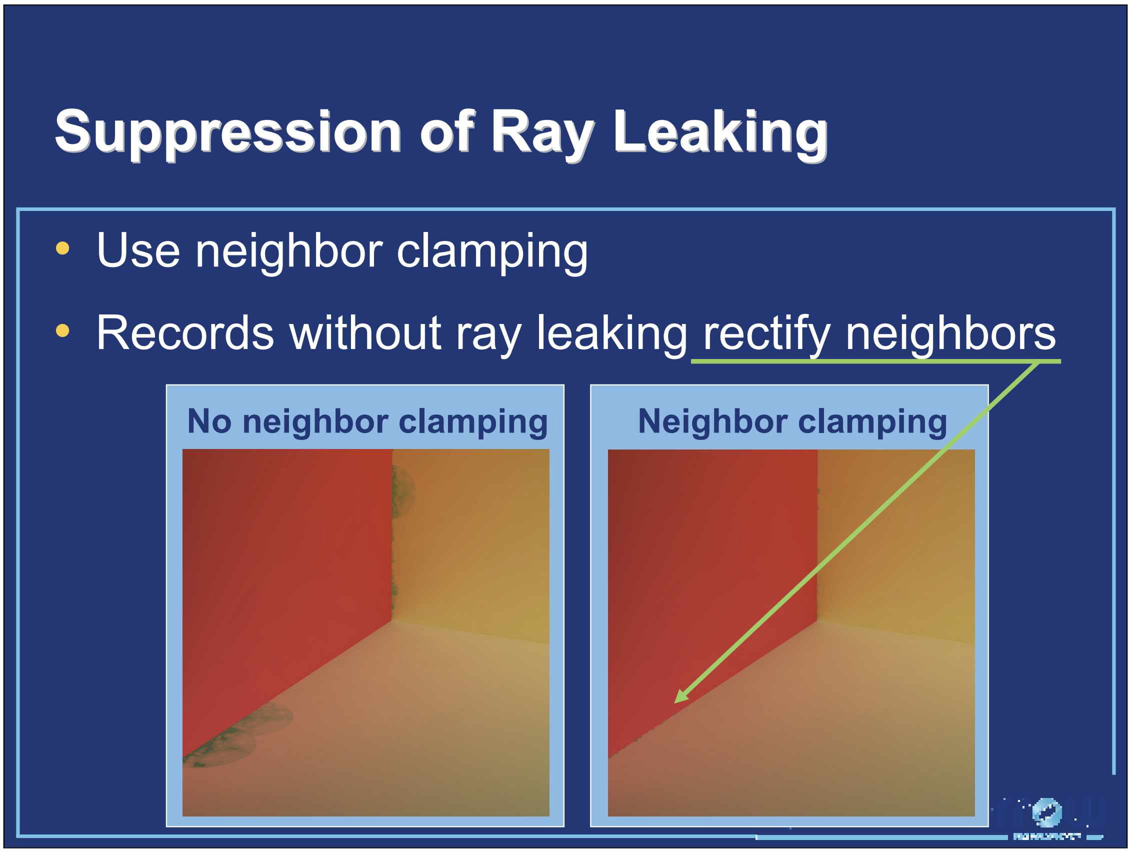
If a primary ray happens to hit such a crack, then most of the secondary rays used in hemisphere sampling “leak” through the crack. As a result, the irradiance estimate is completely wrong and, more seriously, the mean distance is greatly overestimated. (If there were no ray leaking, all those leaking rays would actually be very short.)

Consequences of Ray Leaking

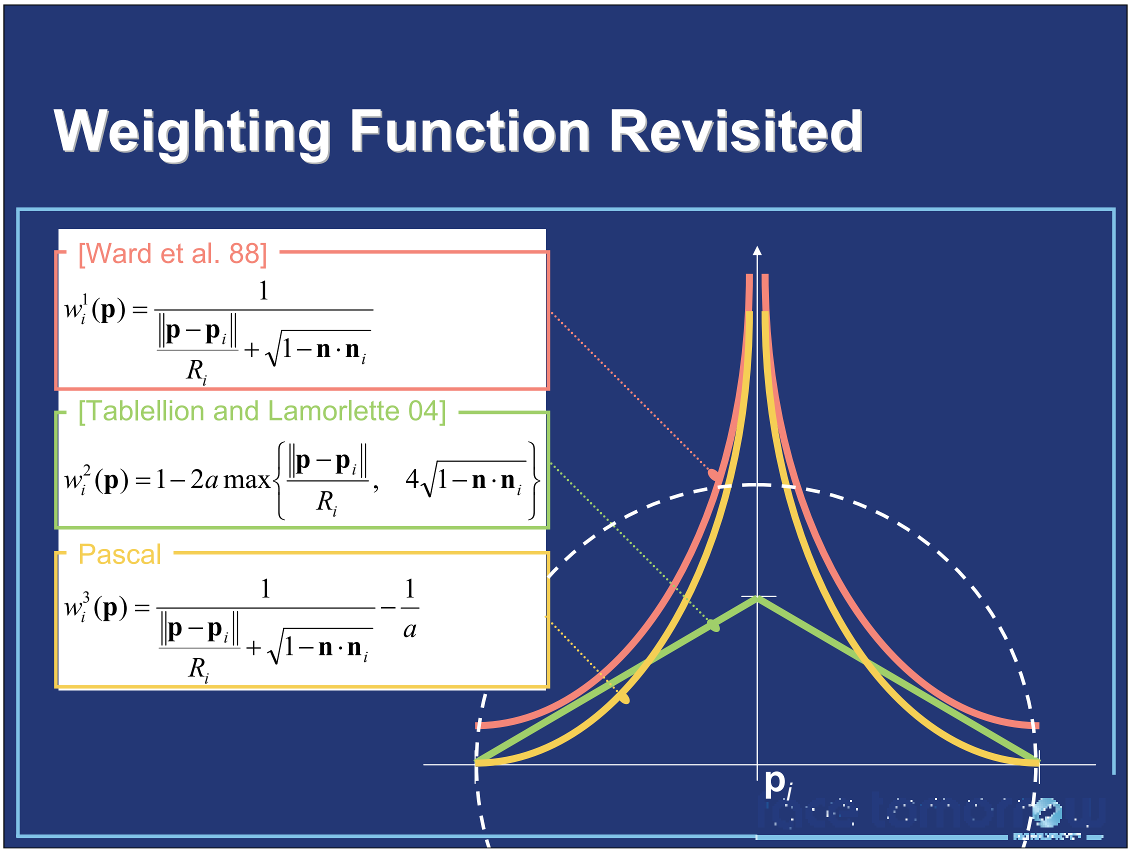
- Wrong illumination estimate
- Extrapolated over large area



The incorrect irradiance estimate is then extrapolated over a large area.

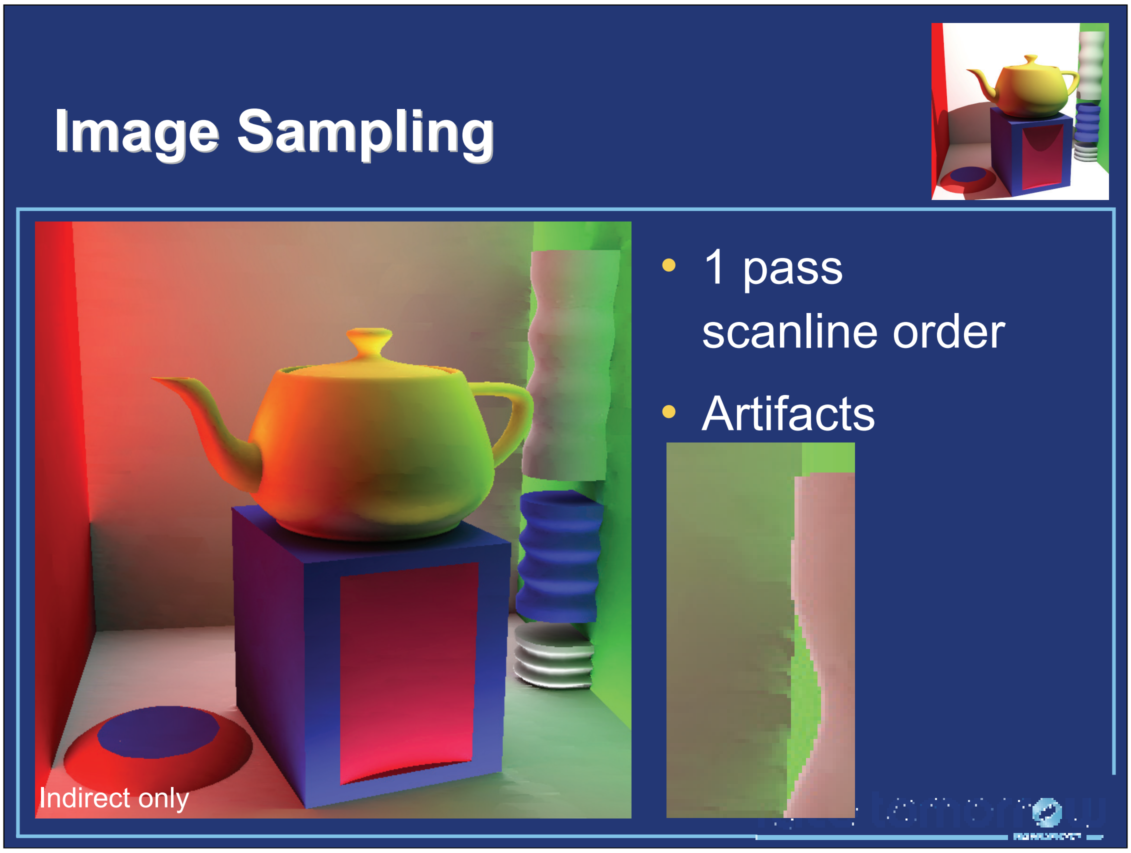


A partial remedy is quite simple – just turning on neighbor clamping. Neighbor clamping detects and rectifies overestimated mean distance, so the wrong irradiance estimate is not extrapolated over such a large area. However, neighbor clamping cannot do anything about the wrong irradiance estimate.

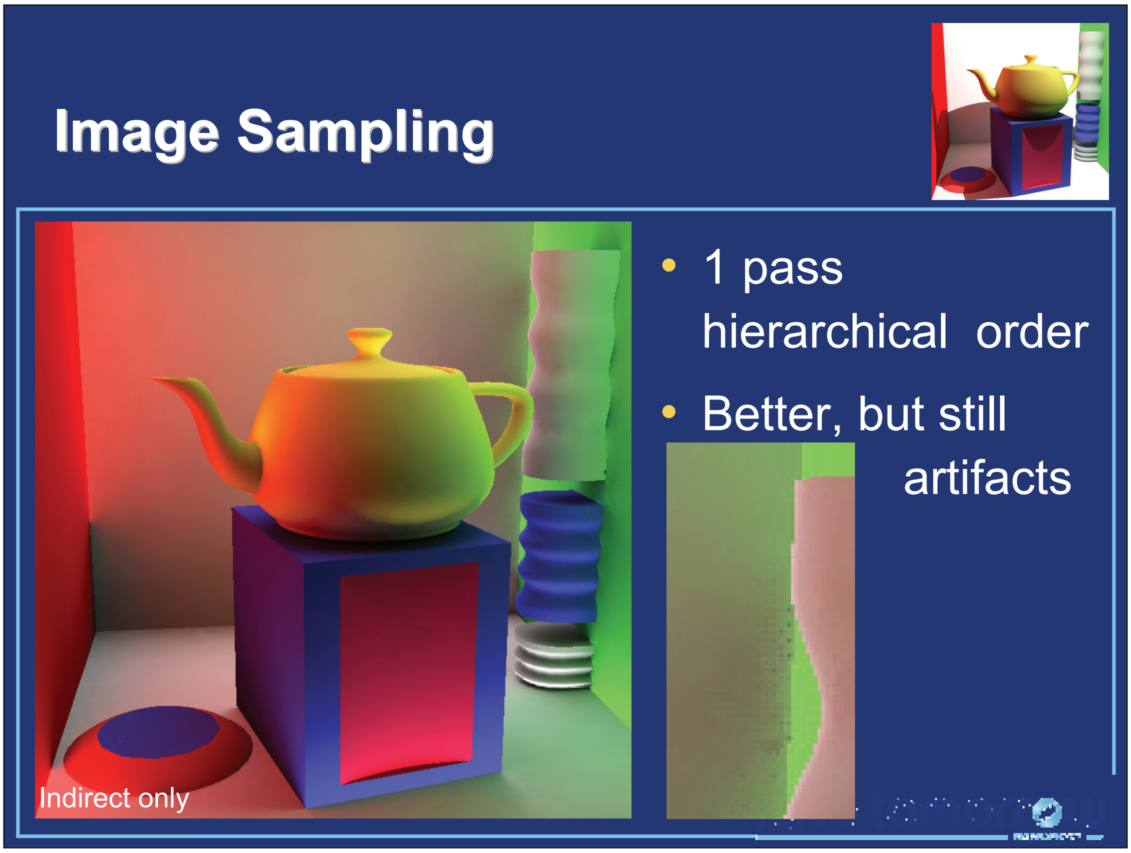


Moving on to another topic---the weighting function used in the weighted average in irradiance interpolation.

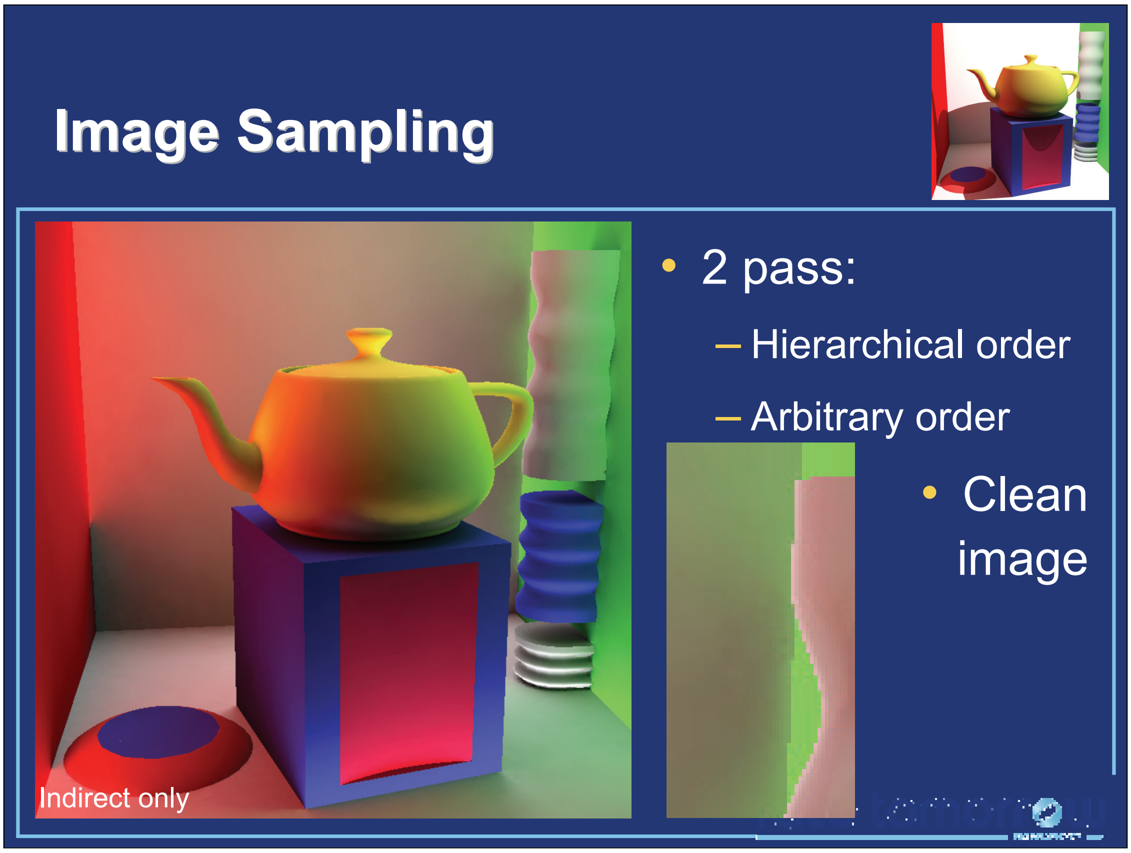
The original weighting function proposed by Ward et al. [1988] has two undesirable properties. First, it goes to infinity when the distance between the point of interpolation **p** and the location of a record, $\mathbf{p}_i$, goes to zero. Second, there is a discontinuity at the border of the influence area of a record. This tends to produce some visible seams in the images. One solution, used in *Radiance*, is to randomize the acceptance of a record for interpolation. In our experience, a better way is to make the weight function zero at the border of the influence area. Two possibilities for this are shown on the slide. The image quality does not depend much on which of them is used.



Lazy evaluation of irradiance values is a great feature of irradiance caching that makes the algorithm very flexible. However, if not used carefully, it has a negative impact on the image quality. This slide shows the kind of artifacts you may expect when generating image pixels in the scanline order, adding new irradiance values lazily as needed.



Using hierarchical image traversal instead of the scanline order improves the image quality (and, actually, decreases the number of records needed to cover the whole image), but image artifacts still remain.



In our experience, the best solution is a two pass traversal of the image. In the first pass, the irradiance cache is filled so that all pixels are covered, but no image is generated. In the second pass, arbitrary pixel traversal can then be used to generate the image.

Summary of Implementation Details

- New record:
  - Sample Hemisphere

    (returns irradiance, gradients, mean/min distance R)
  - $R = \min \{ R, 1/||\mathrm{grad}_t|| \}$ // limit R by gradient
  - $R' = \max \{ \min \{ R, R_{\mathrm{max}} \}, R_{\mathrm{min}} \}$ // clamp R between R_{min} and R_{max}
  - if(R < R') grad *= R/R' // limit gradient by R
  - Neighbor clamping // use R instead of R' here
  - Insert into cache

This slide summarizes the actions taken to add a new irradiance value into the cache. First, we sample the hemisphere by casting a number of secondary rays. This gives the irradiance estimate, the translational and the rotational gradients, and the estimate of the mean (or minimum) distance to the neighboring geometry, *R*. We then limit the value of *R* by the translational gradient. After that we clamp *R* by the minimum and maximum threshold (determined from the projected pixel size). This produces the clamped value *R*'. If *R* was increased by this clamping, we then decrease the gradient magnitude accordingly, in order to avoid negative values in extrapolation. The next step is neighbor clamping. For its correct functionality, it is essential to use the original, unclamped value of *R* (not *R*'). Finally, we insert the new record into the cache.