Temporary Coherent Irradiance Caching for High Quality Animation Rendering

Miloslaw Smyk[1/2], Shin-ichi Kinuwaki[3]
Roman Ďurikovič[4], Karol Myszkowski[1]

[1]MPI Informatik, Germany
[2]Szczecin University of Technology, Poland
[3]The University of Aizu, Japan
[4]University of Saint Cyril and Metod, Slovakia

Motivation

- Speed up
  - Computation can be reused in subsequent frames
  - Update only changed part
- Quality
  - Reduce popping or flickering artifacts
    - Cannot be seen in static pictures

Related Work

- Shading Reusing
  - Render Cache [Walter 99, 02]
  - Shading Cache [Tole 02]
  - Temporary Coherent Irradiance Cache [Tawara 04]
- Path Reusing
  - Space Time Architecture [Havran 03]
  - Real Time Light Animation [Sbert 04]
  - **Our method belongs to this category**
- Path and Shading Reusing
  - Selective Photon Tracing [Dmitriev 02]
  - Real time photon mapping [Larsen 04]

Overview

1. Global Illumination Framework
   (with photon mapping and irradiance cache)
2. Our Contribution
   - Anchor Data Structure
   - New Animation Framework
   - Anchor and Cache Density Control
   - Extension to Glossy Surface.

- Results & Demo
- Conclusions & Future Work

Global Illumination Solution

- Most time consuming tasks are
  - Indirect illumination evaluation
    - glossy and diffuse components

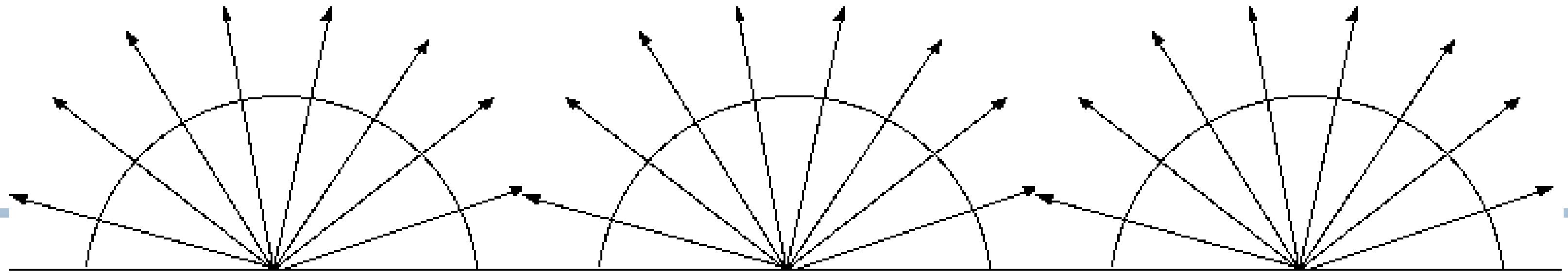
Why difficult?

- Indirect illumination coming from any direction should be considered
  - The reflectance for diffuse and glossy materials is usually smooth function influenced by only cosine factor

→ Final Gathering

Final Gathering

- Many final gathering rays traced
  - Typically 1,000-2,000 rays per sample point
- Many *k*-nearest neighbor operations for global photons is needed
  - 10 millions – 1 billion “space traverse” and “k-nearest neighbor” estimation

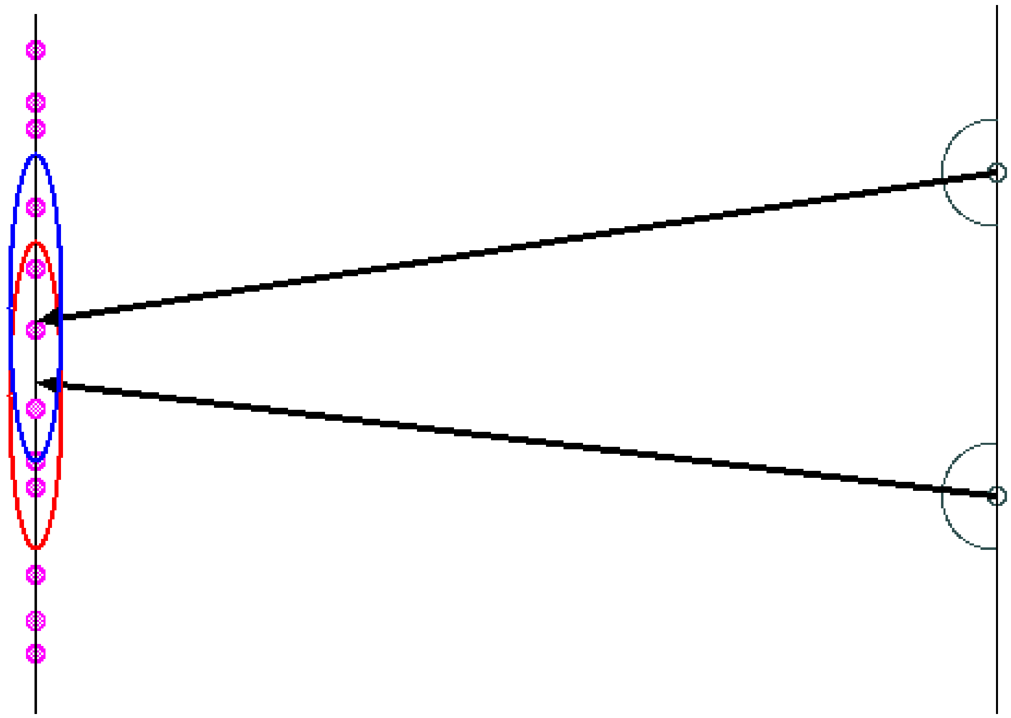


Smart Final Gathering

- To reduce the number of sample points
  →Irradiance caching [Ward 88, 92]
- To reduce the number of k-nearest neighbor
  →Precomputed Irradiance [Christensen 00]
- Still heavy …..
  →**Our anchor data structure**

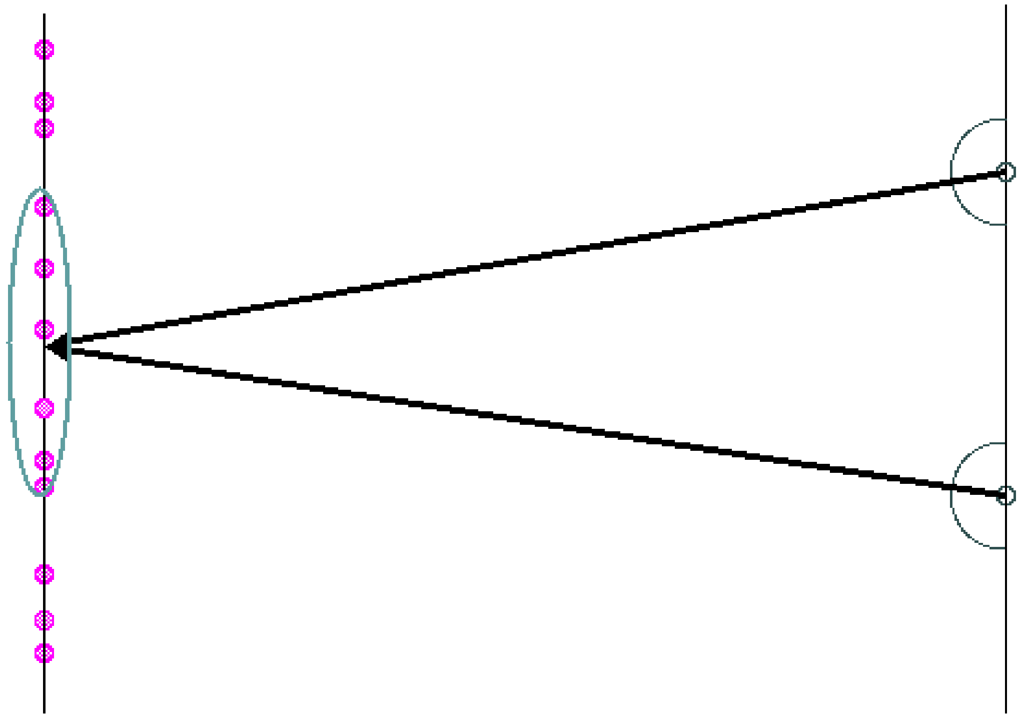
What is the Anchor?

- New type of data structure to reduce the unnecessary k-nearest neighbors search



What is the Anchor?

- These two paths share same radiance of an anchor. Anchor is “point” data structure and independent from geometry.

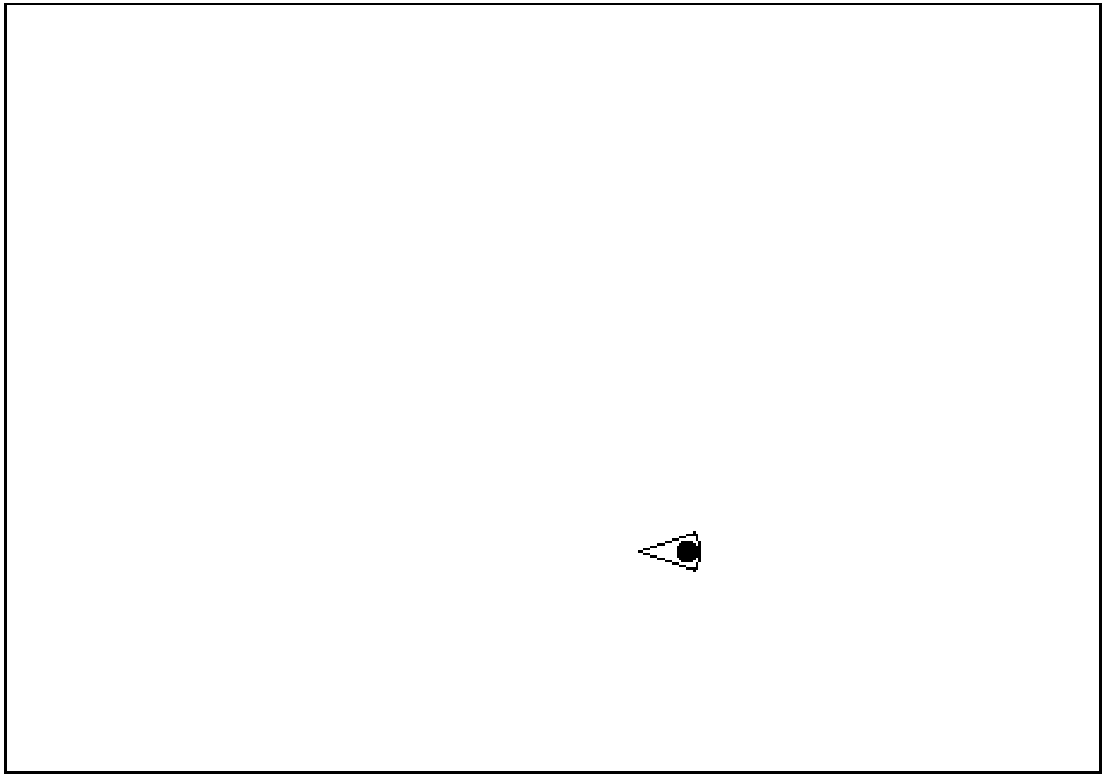


Difference Between Precomputed Irradiance and Anchor

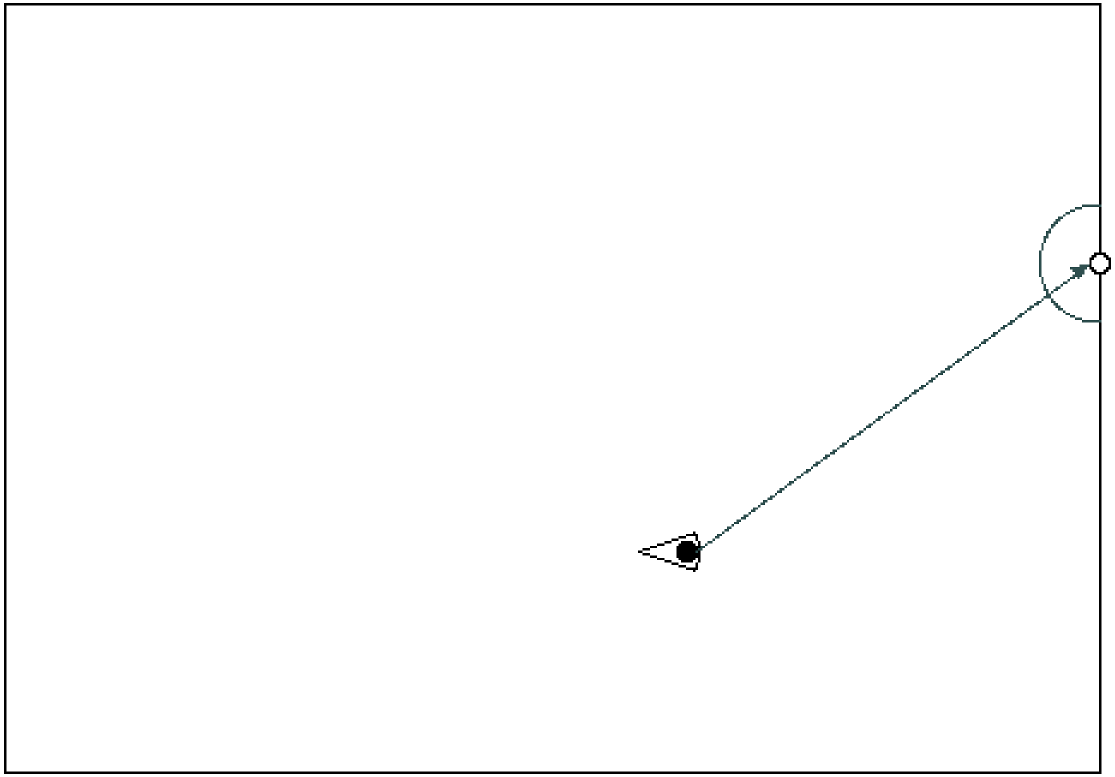
max planck institut informatik

- Very similar, but different.
- Precomputed Irradiance [Christensen 00]
  - Position → photon hit point
  - Their positions are deterministic in a single frame, but not in animation framework. (Balanced kd-tree is constructed every frame)
- Anchor
  - Position → final gather hit point
  - Their positions are not deterministic in a single frame, but deterministic in animation framework. Anchor is sequentially inserted. (unbalanced kd-tree is kept in all frames)

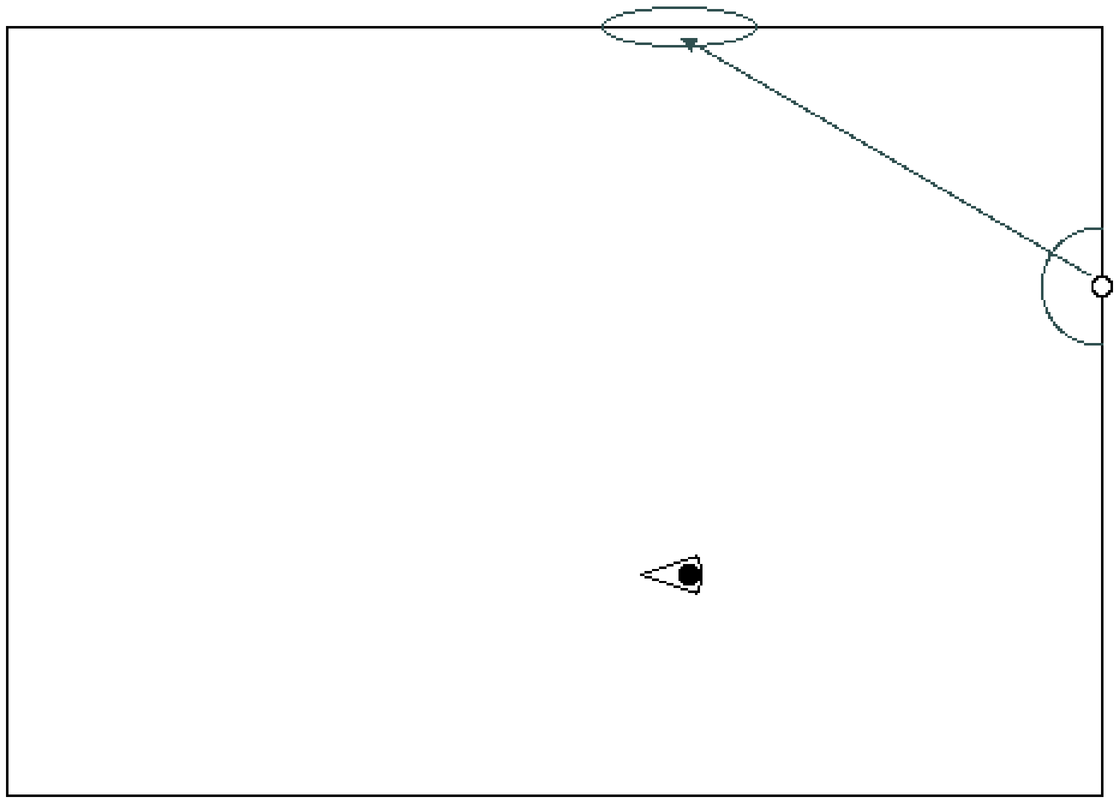
Anchor Creation Procedure



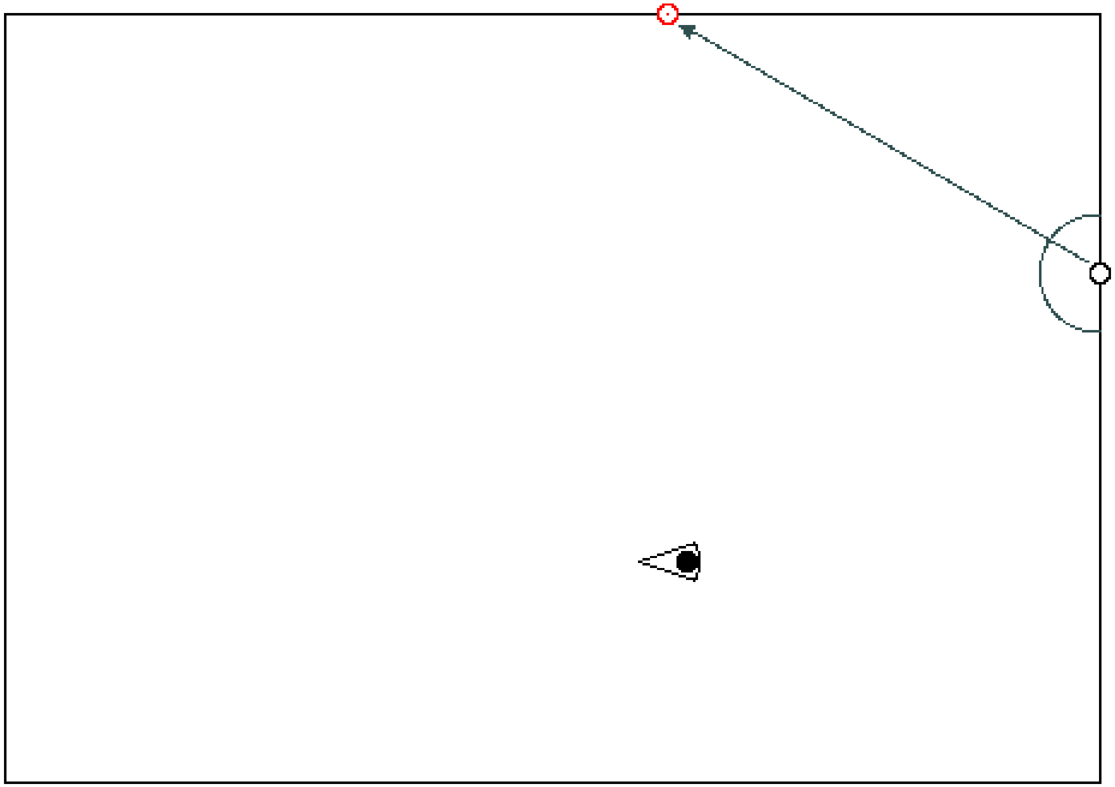
Anchor Creation Procedure



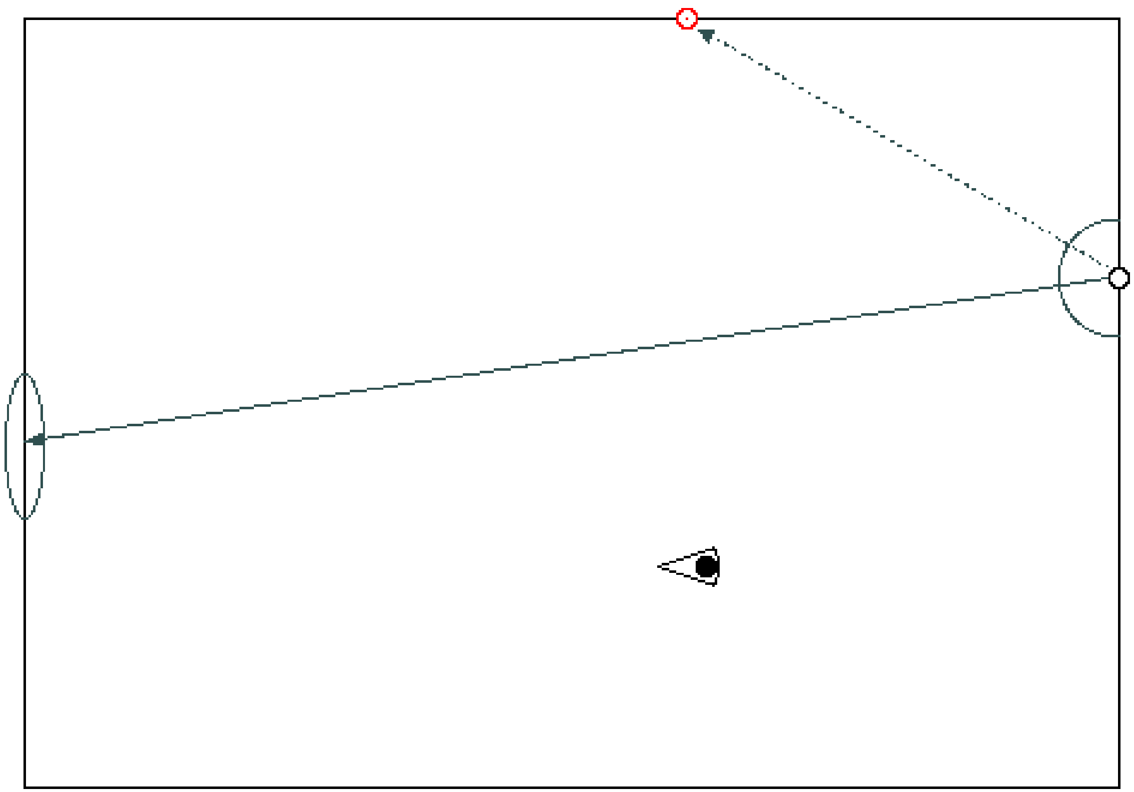
Anchor Creation Procedure



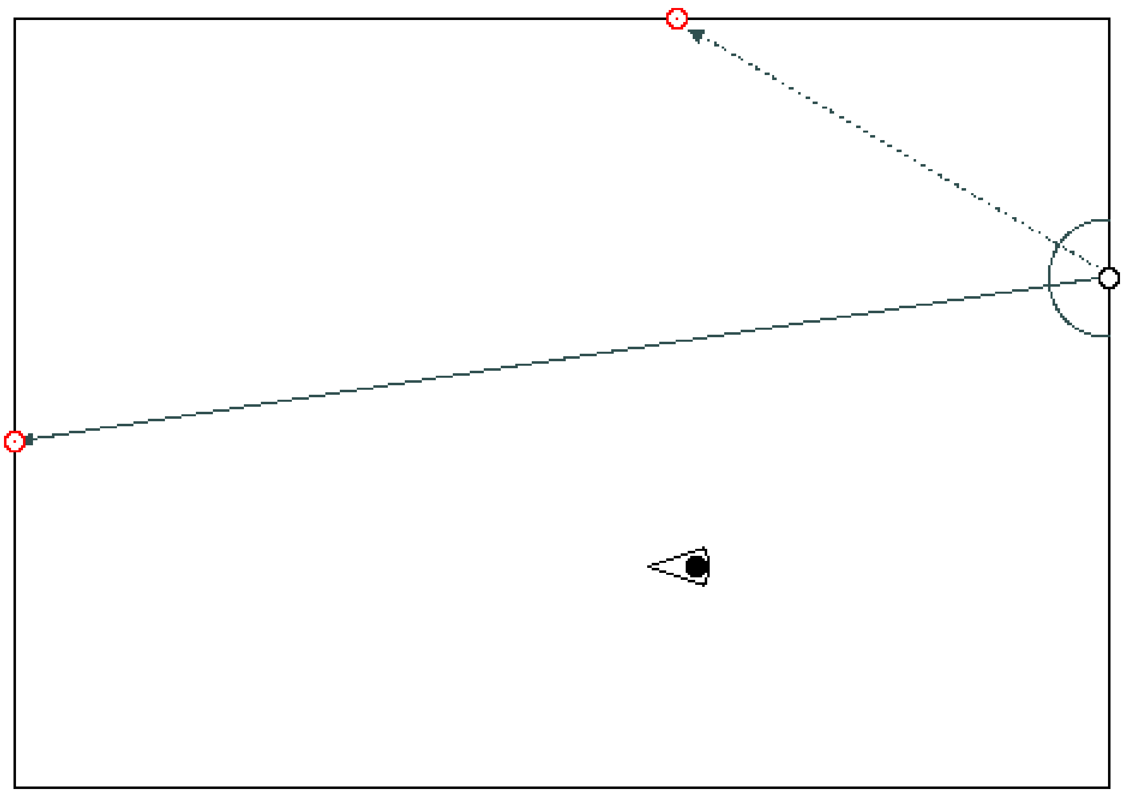
Anchor Creation Procedure



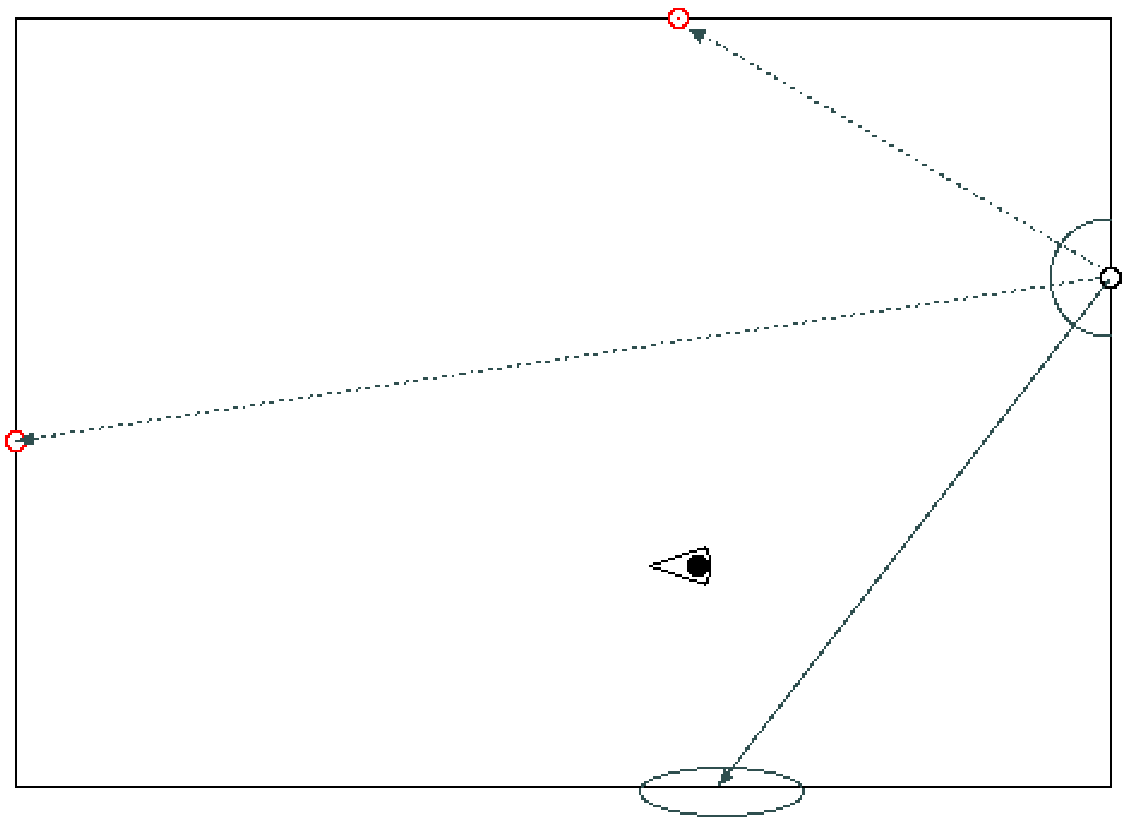
Anchor Creation Procedure



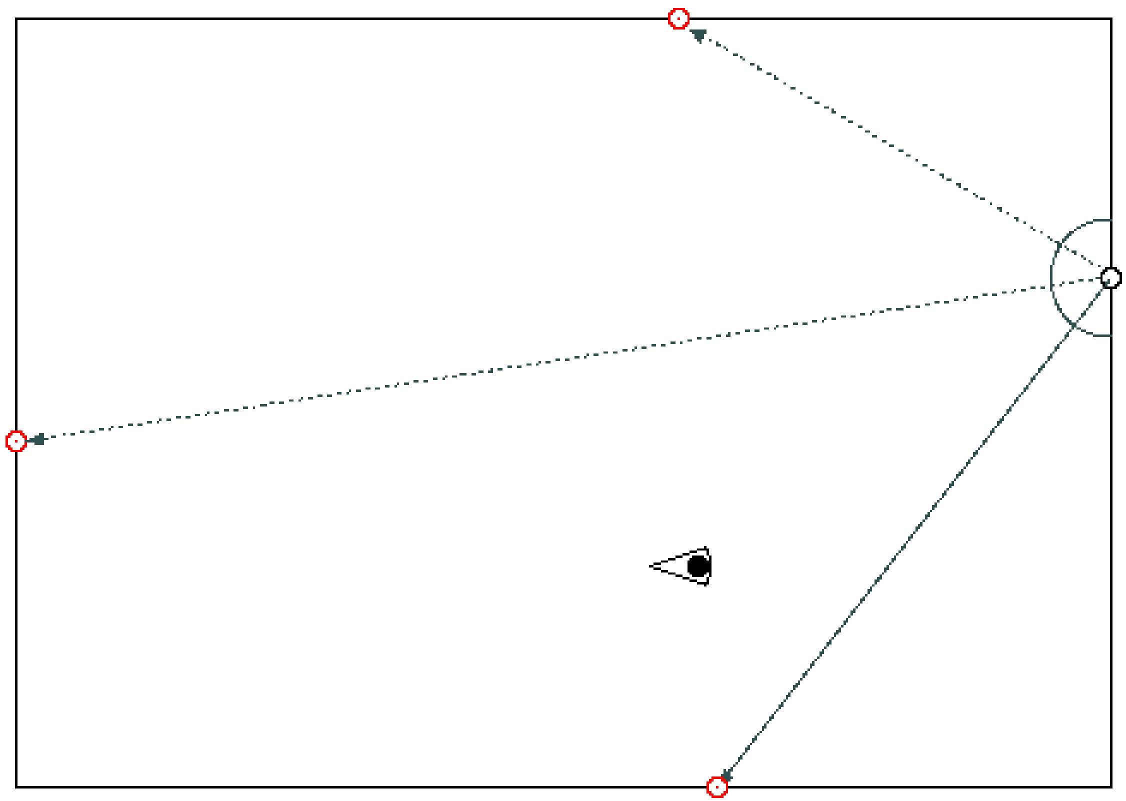
Anchor Creation Procedure



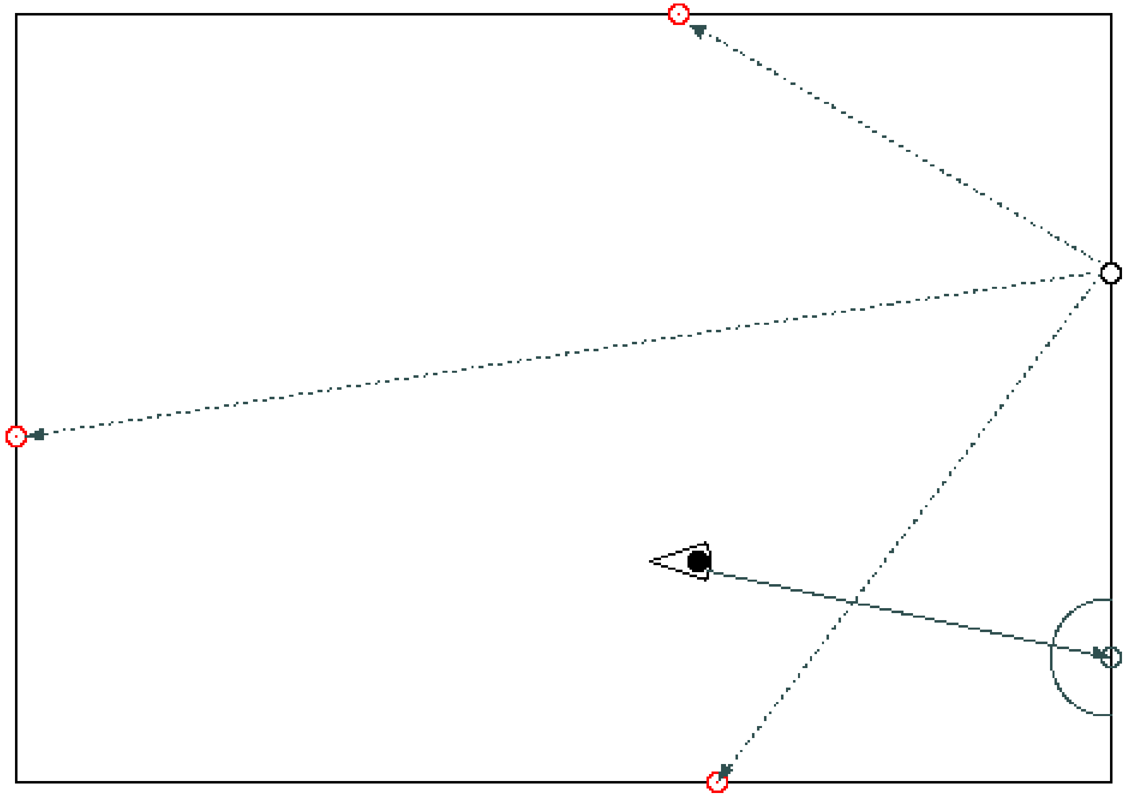
Anchor Creation Procedure



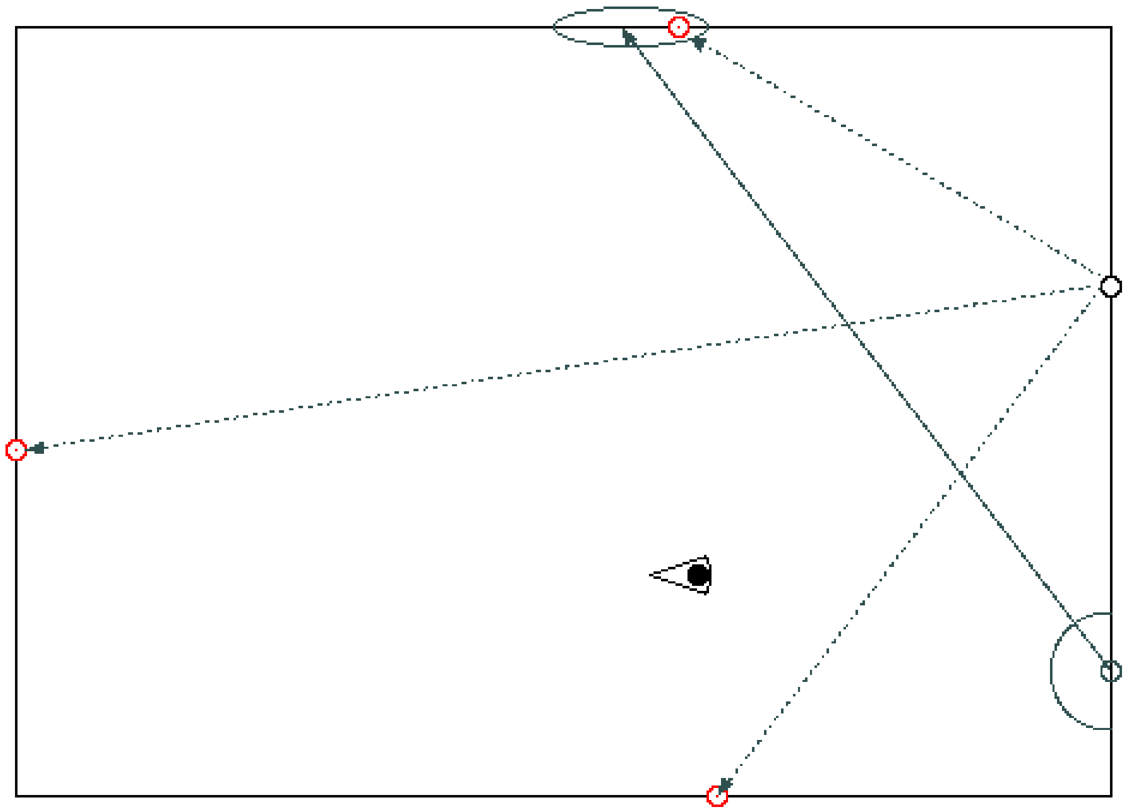
Anchor Creation Procedure



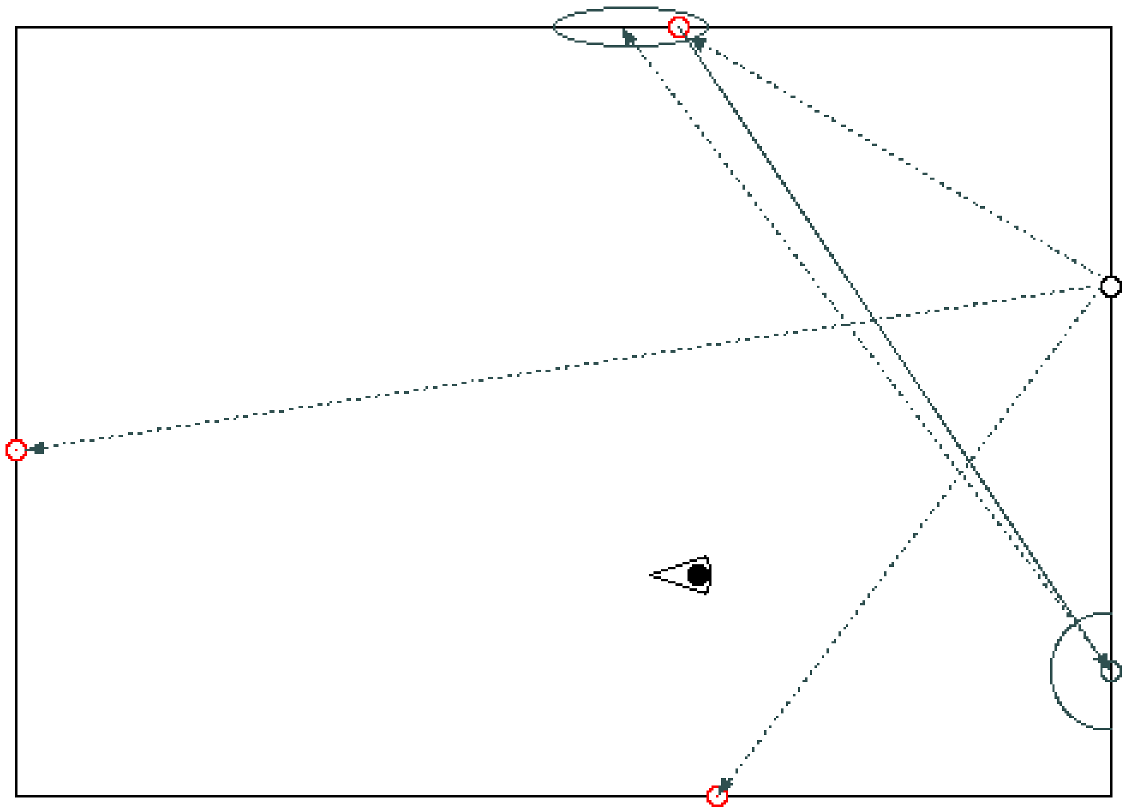
Anchor Creation Procedure



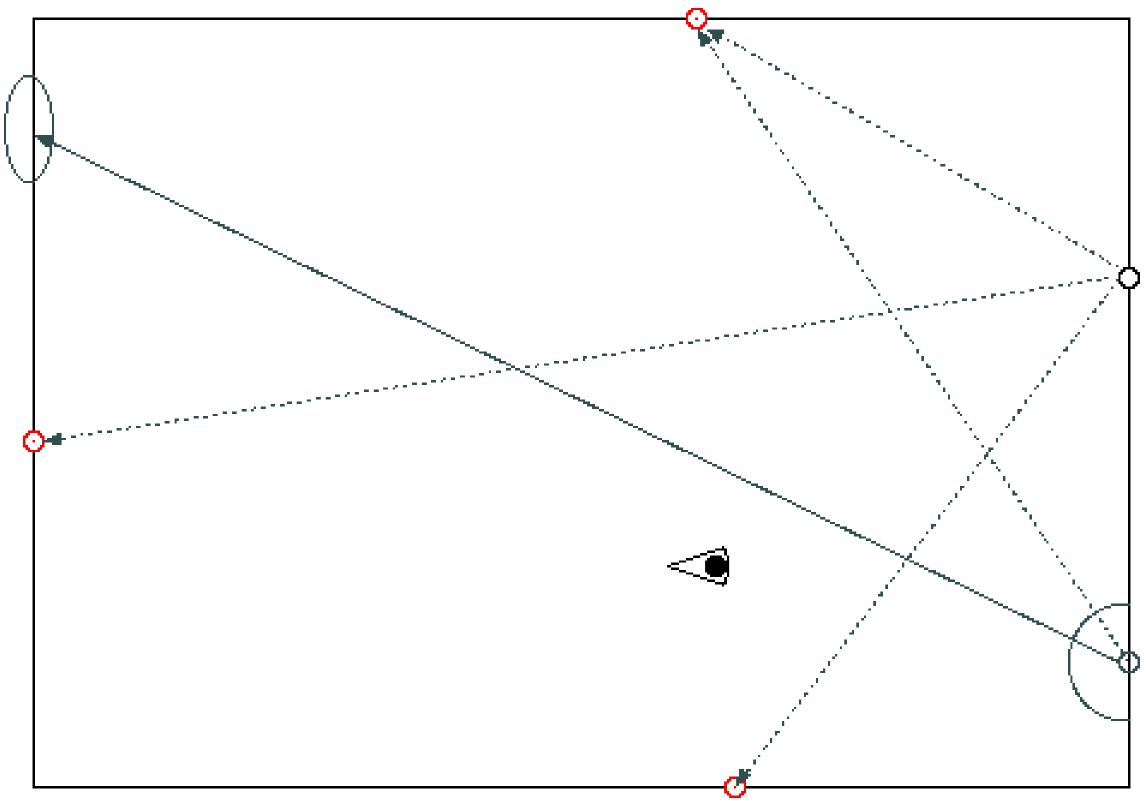
Anchor Creation Procedure



Anchor Creation Procedure

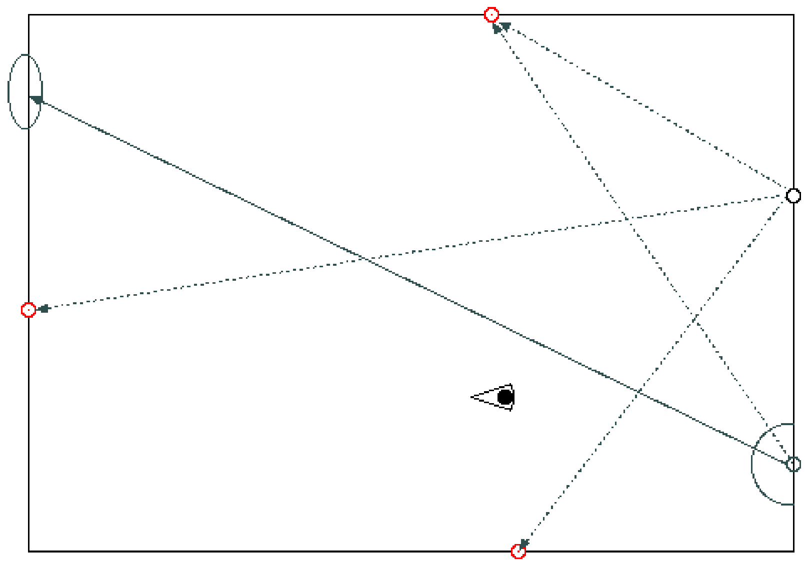


Anchor Creation Procedure



Animation framework using Anchor

- Most visibilities from “anchor” and “irradiance cache” are kept in subsequent frames
- If visibility does not change, we need not traverse space and kNN (single-NN) for control point (directly access anchor value)



Fast Detection path change

- We have two types of visibility changes between anchor and irradiance cache
  - Dynamic object comes
  - Dynamic object leaves
- We re-shot rays for these cases
- How to find them?
- Project all dynamic objects into all caches
  - Full CPU technique (e.g., Kautz 04)
  - Hemi-sphere projection using GPU (**Our choice**)

Animation framework using Anchor

max planck institut
informatik

- Photon Tracing (every frame)
- Density Estimation in all anchors (every frame)
- Cache Creation (1st frame)
- Cache Update (after 2nd frame)
  - Re-shot rays to changed path and create (if near anchor is not found) and link new anchor
  - Access directly anchor radiance value
  - Evaluate Irradiance on each cache
- Render Scene
- Proceed with the next frame and repeat I

Cost of Final Gathering per Ray (Experimental Data)

	Space traversing	Density Estimation
Brute Force	1	9
Precomputed Irradiance	1	1
Our method (in animation)	0	0

Anchor and Cache Density Control

- Each anchor requires 22 bytes
  - 500,000 anchors → 11 MB *
    - Can be easily handled
- Each strata of cache consumes 7 bytes
  - 15,000 caches 1,600rays → 168MB!!
    - Requires smart management system
- Management of such data structure is very important.

*actually, we have anchor list, anchor tree. We consume more memory.

Spurious Cache Removal

We consider three types of caches to be removed

a. Invisible Caches
   - When time stamp > threshold, delete!!

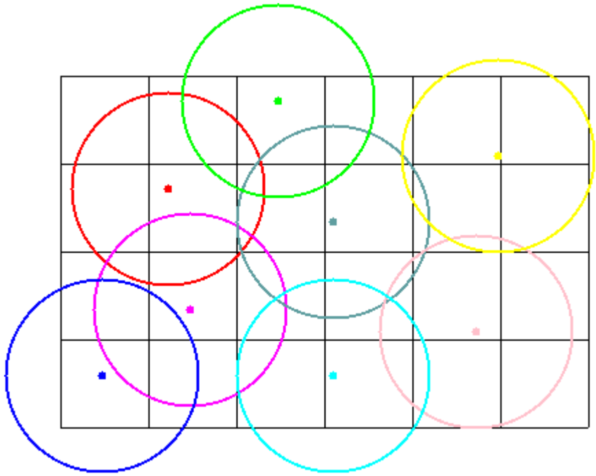
b. Perceptually unimportant cache
   - Remove little contributing caches

c. Cache on dynamic object
   a. Transform new position (rigid body only)
   b. Simply delete and recreate every frame

Perceptually Unimportant Caches

max planck institut
informatik

- For each pixel, we compute how much each cache contributes to current pixel

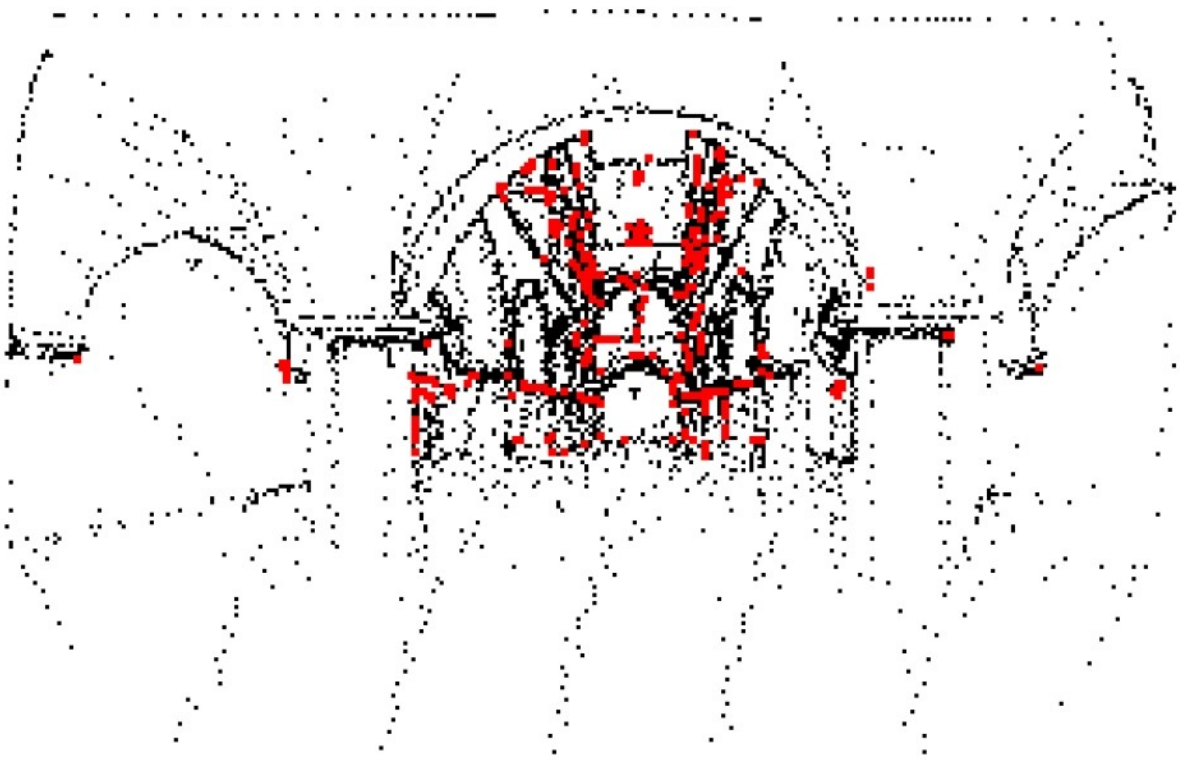


- Pixel contribution of each cache can be expressed as follows:

$$\delta_i = \sum_{pixels \in i}^{j} \frac{|P_{j,i} - \bar{P}_{j,i}|}{P_{ji}}$$

Perceptually Unimportant Caches

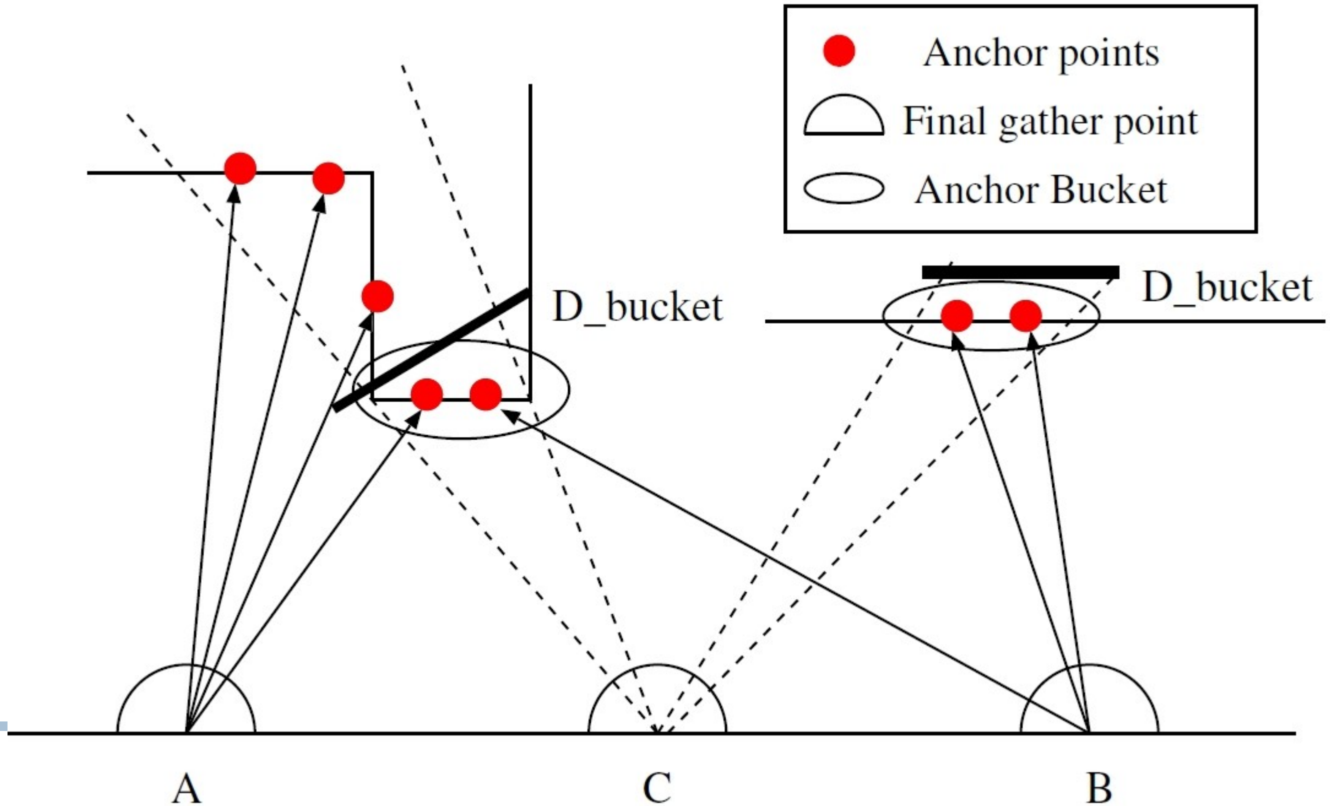
- Remove all caches that do not reach some threshold value (e.g. 5% of all caches)



Glossy Surface Extension

max planck institut
informatik

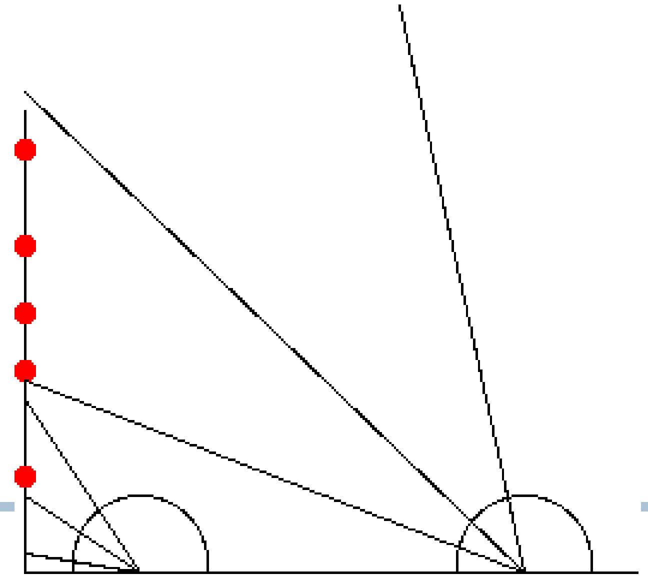
- Anchor Bucket
- Exploit projection of anchor (like Kato 02)



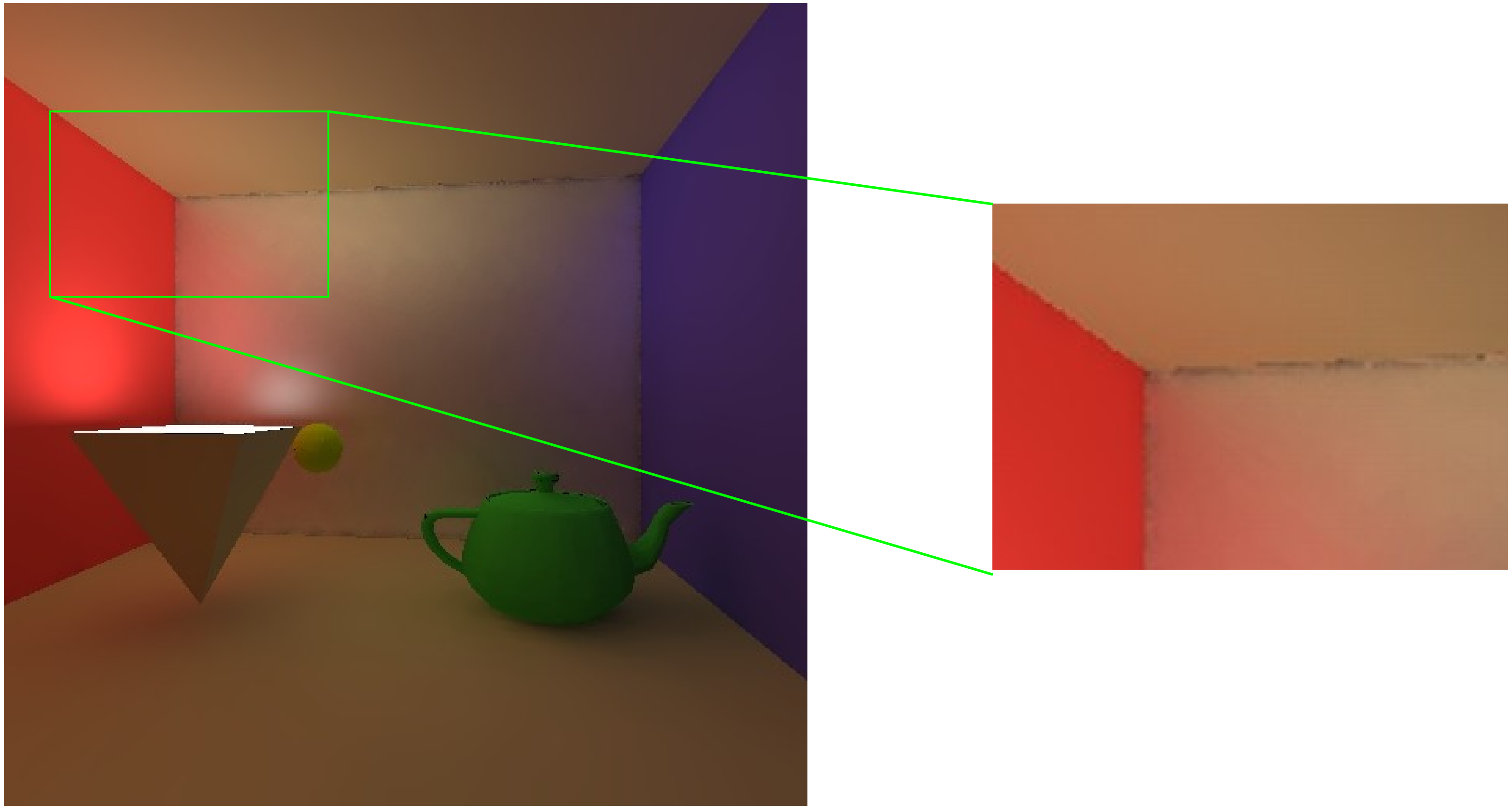
Boundary Problem (Glossy)

- In the corner, we cannot bucket an adequate number of anchor
  - This leads to darker corners

→ ray should be re-shot for such strata

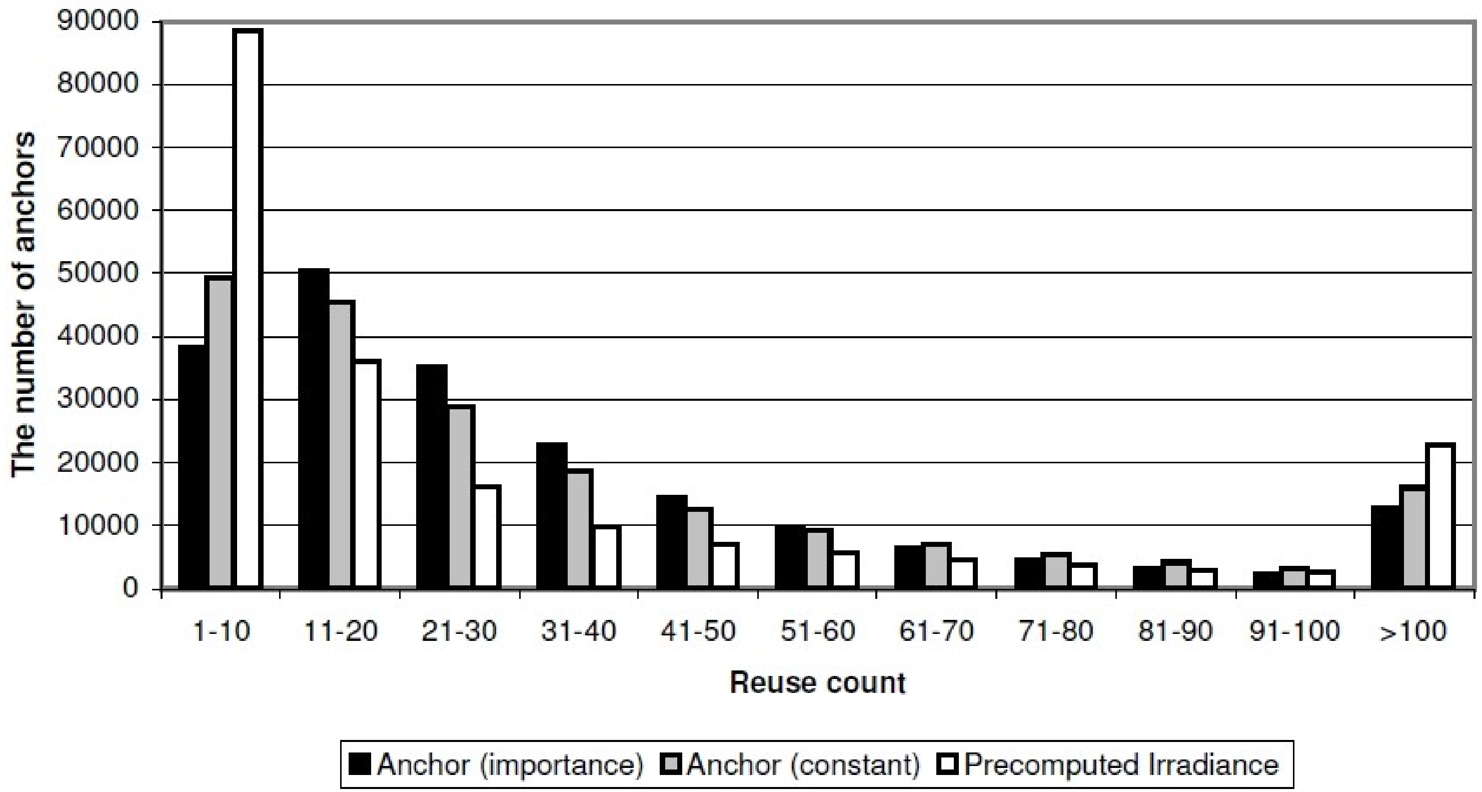


Boundary Problem (Glossy)



Result

- Reuse of Anchor vs. Precomputed Irradiance.



Result

- Computation time (average time per frame)

	Sponza		Sibenik	
	Brute Force	Proposed	Brute Force	Proposed
Photon Tracing	28s	28s	34s	34s
Density Estimation	21s	6s	23s	6s
Final Gathering	901s	115s	1040s	122s
Irradiance Interpolate	37s	41s	33s	36s
Total	987s	186s	1130s	198s

Movie (Sibenik Atrium)

Future Work

- All frequency support
  → view dependent anchor?
- Irradiance cache on dynamic object
  → some smart visibility technique
- Anchor bucket in general solution
- 1st frame speed up