ETN-FPI TS3 “Plenoptic Sensing”

Efficient Depth-Compensated Interpolation for Full Parallax Displays

Reinhard Koch and Daniel Jung

Multimedia Information Processing
Institute of Computer Science
Christian-Albrechts-University of Kiel

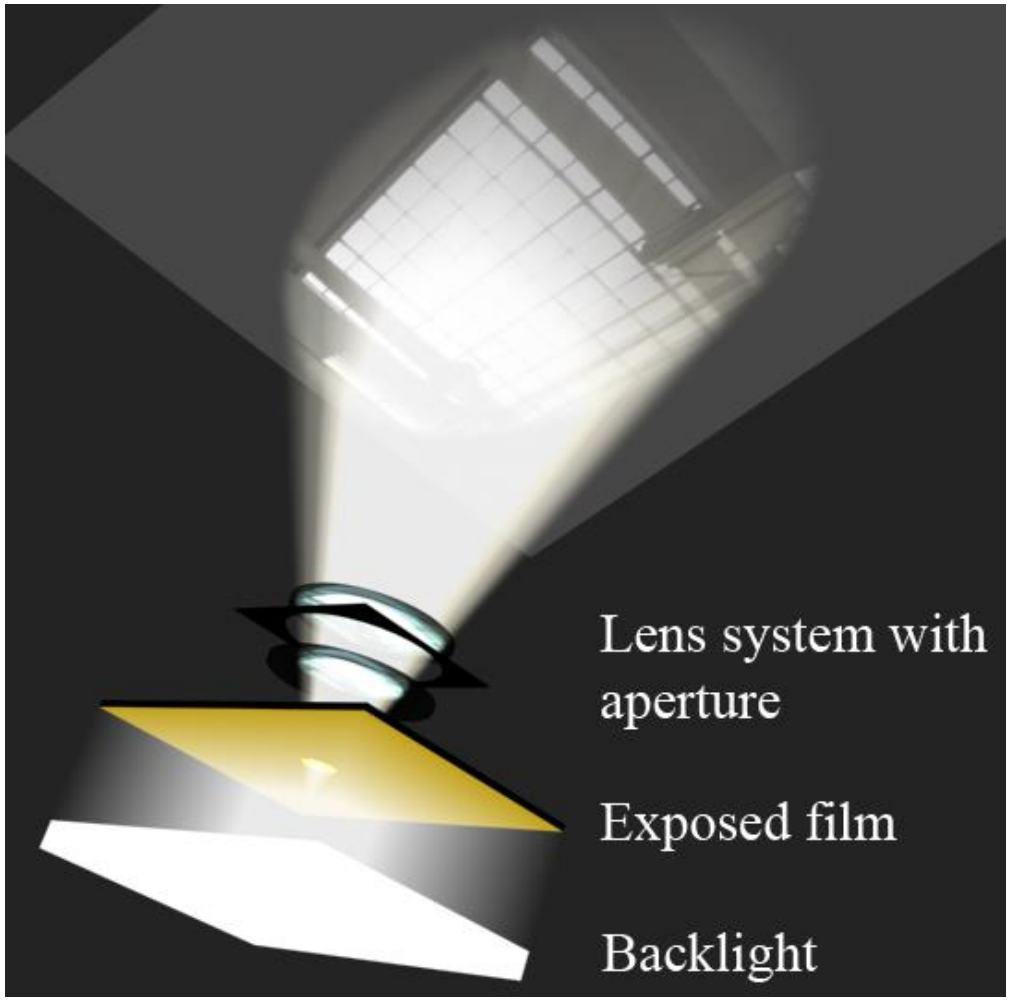
Large-scale Lightfield Display

- Full parallax autostereoscopic display
- 230,000 pixel/m²
- One lens system for each pixel (Diameter 2mm)
- 205.880 different views per lens

Picture of a prototype display (Size 1m²)

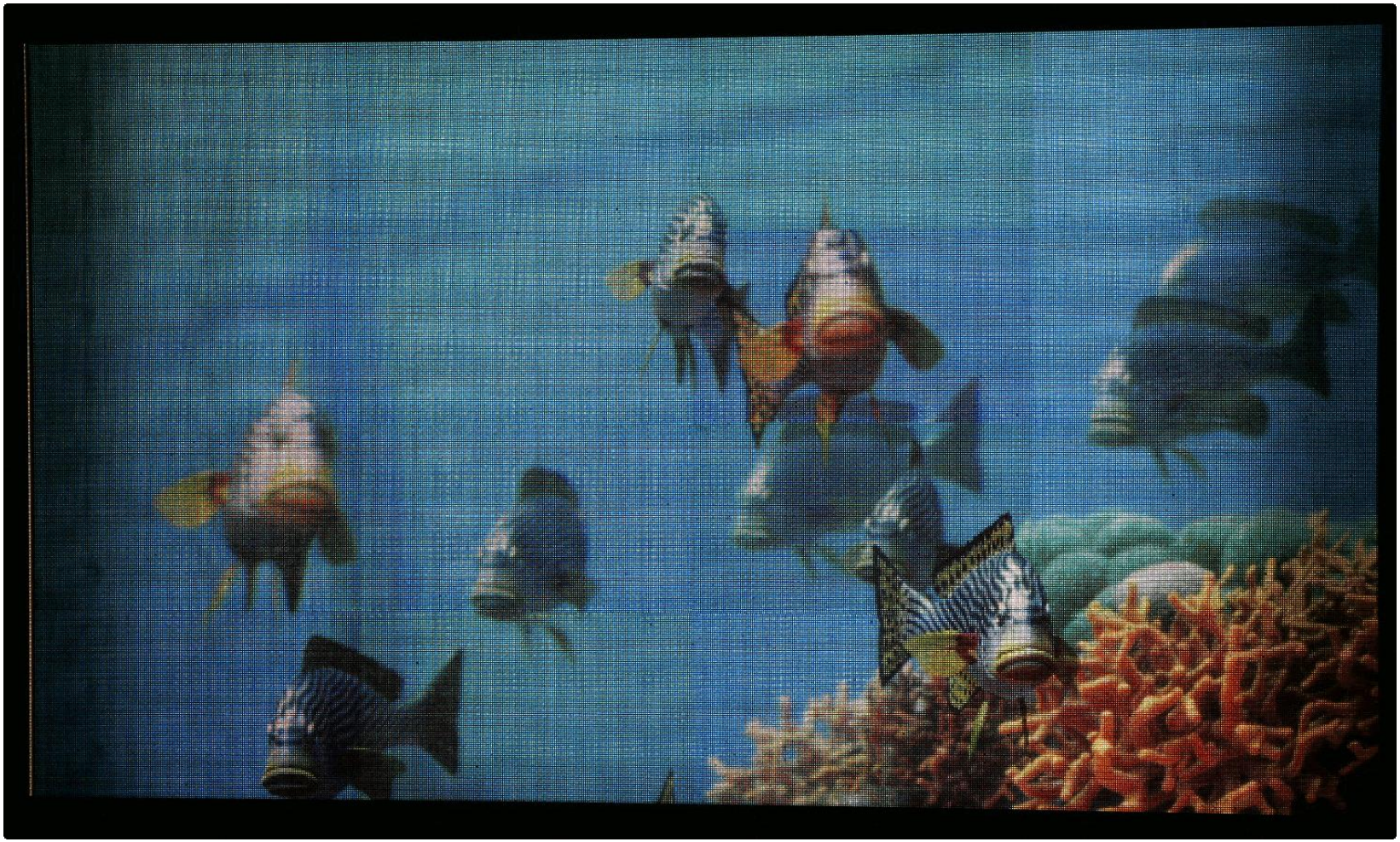
Lens System for view-dependent pixel

- Lens system works like a projector
- Film exposed with lens images
- Circular field of view of 40 degrees
- High angular resolution (0.08 ° per ray)
- One lens image per pixel with up to 512² micropixel



Sketch of one pixel lens system

Prototype Display



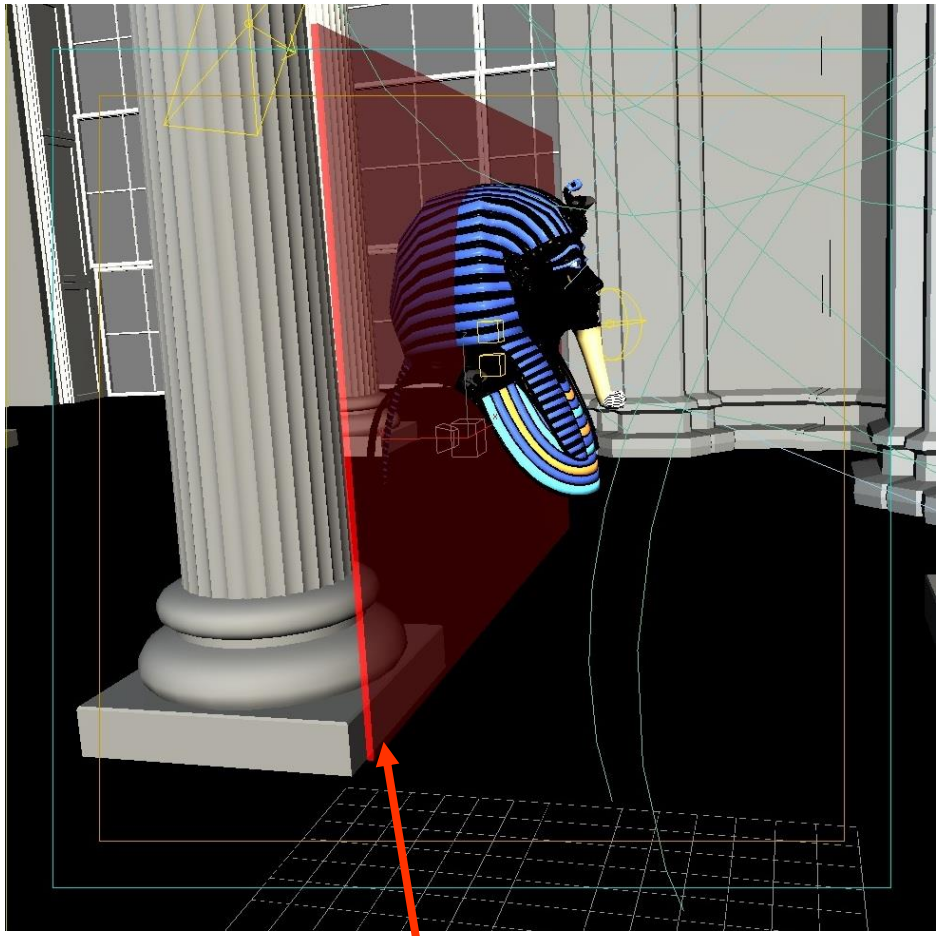
Snapshots of a prototype display captured by a photo camera

Rendering of lens images

- Modeling tool: 3ds max
- Rendering with VRay plug-in

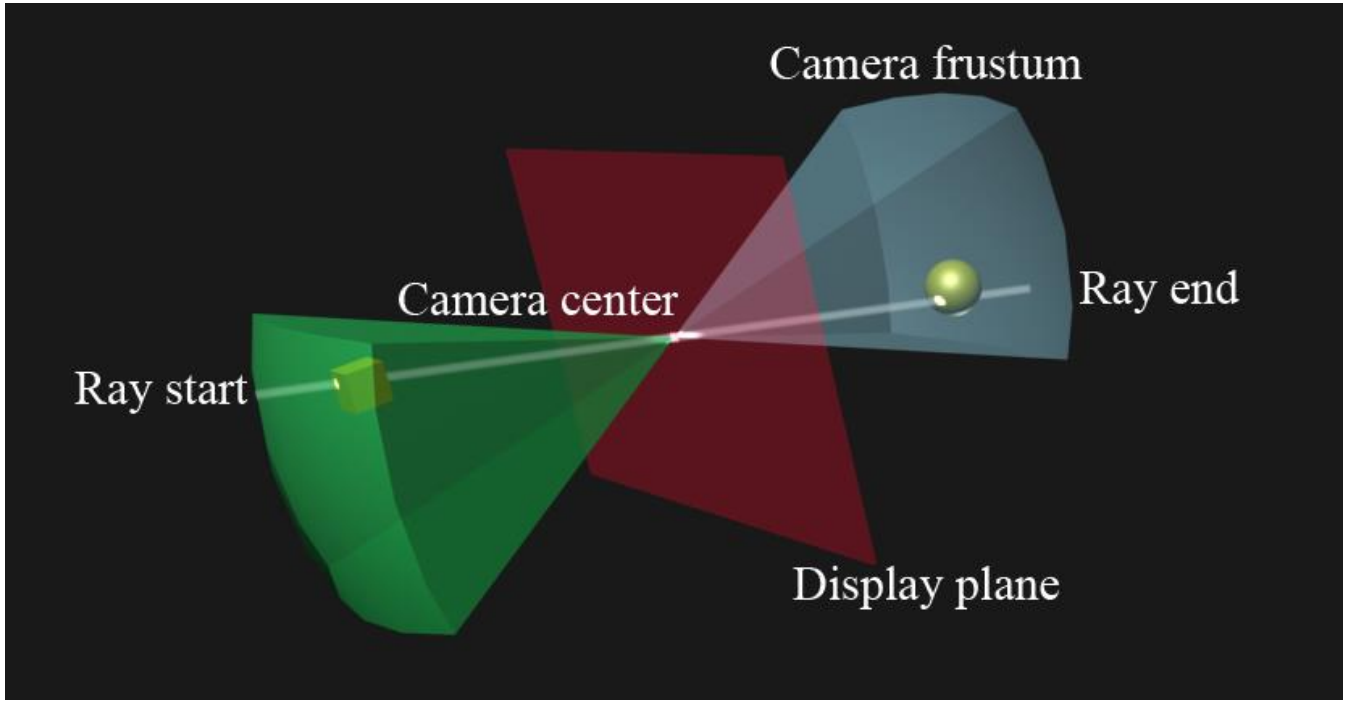
Rendering:

- Set camera into lens positions
- Ray trace lens images (1 image per pixel)



Screenshot of artificial scene in 3ds max

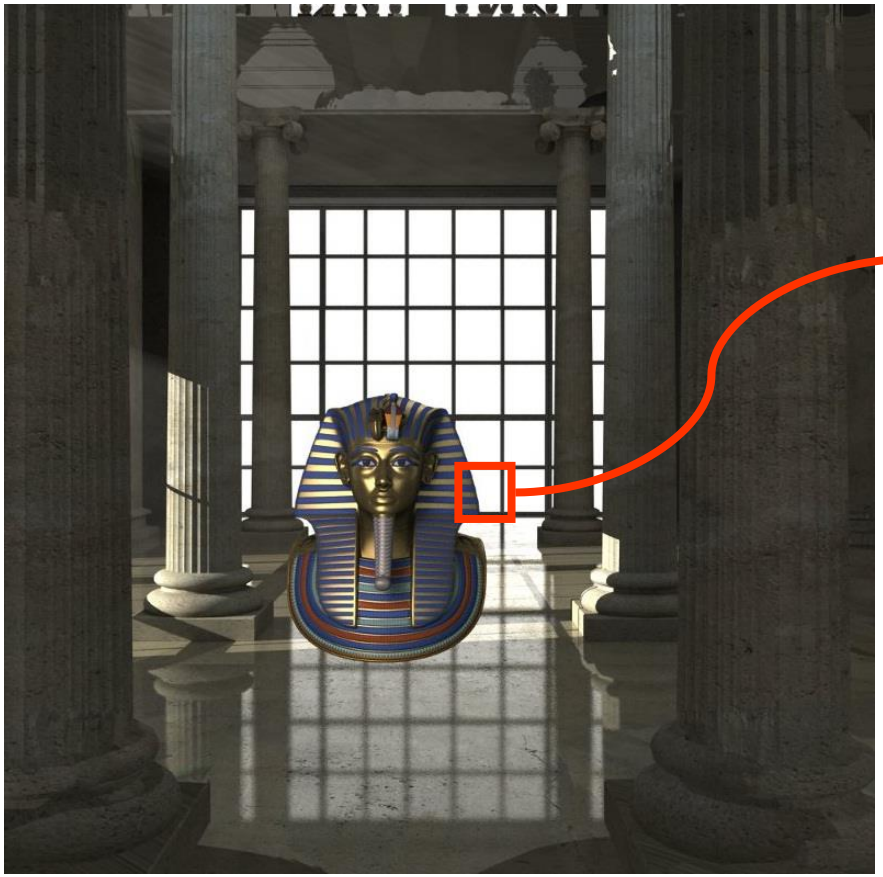
Rendering of lens images



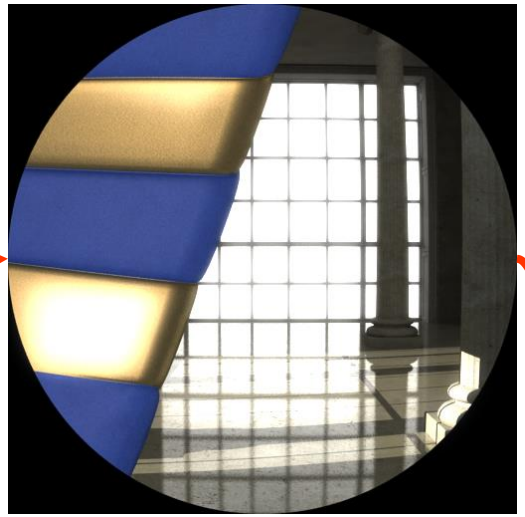
Lens Image:

- circular image 40 °
- Rays intersect on both sides of display plane
- Full ray tracing for each lens image

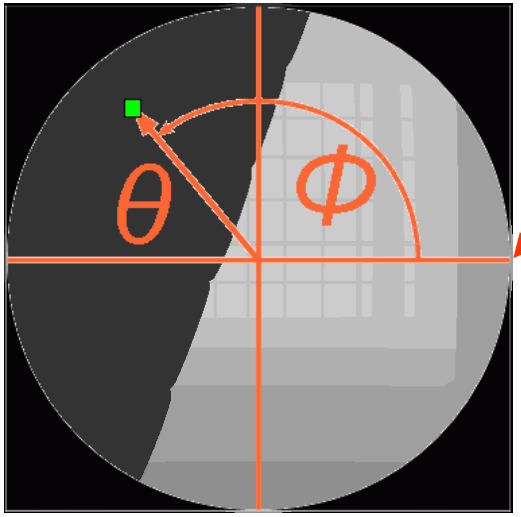
Rendering of lens images



Rendered overview with 3ds max



Color lens Image
(expensive)



Depth lens Image
(cheap)

Render Challenges

- One image per lens system (230.000 images/m^2)
- 205 GByte/m^2
- Rendering takes several months on single workstation
- Images are very similar


Goal:

- Acceleration of rendering
- Interpolate few key images
- No observable loss in quality


Lens images

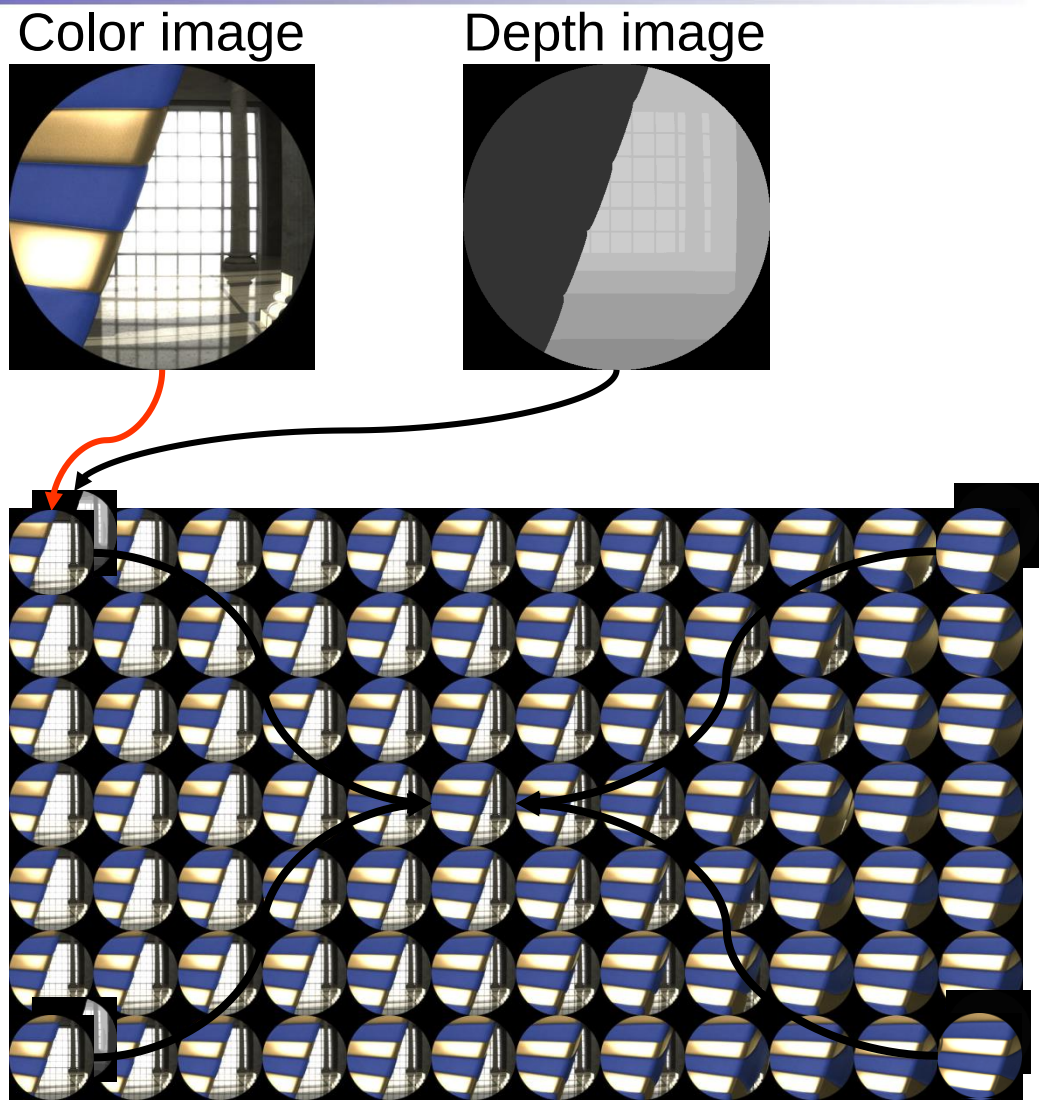
Approach

- Select key image

At every key image:

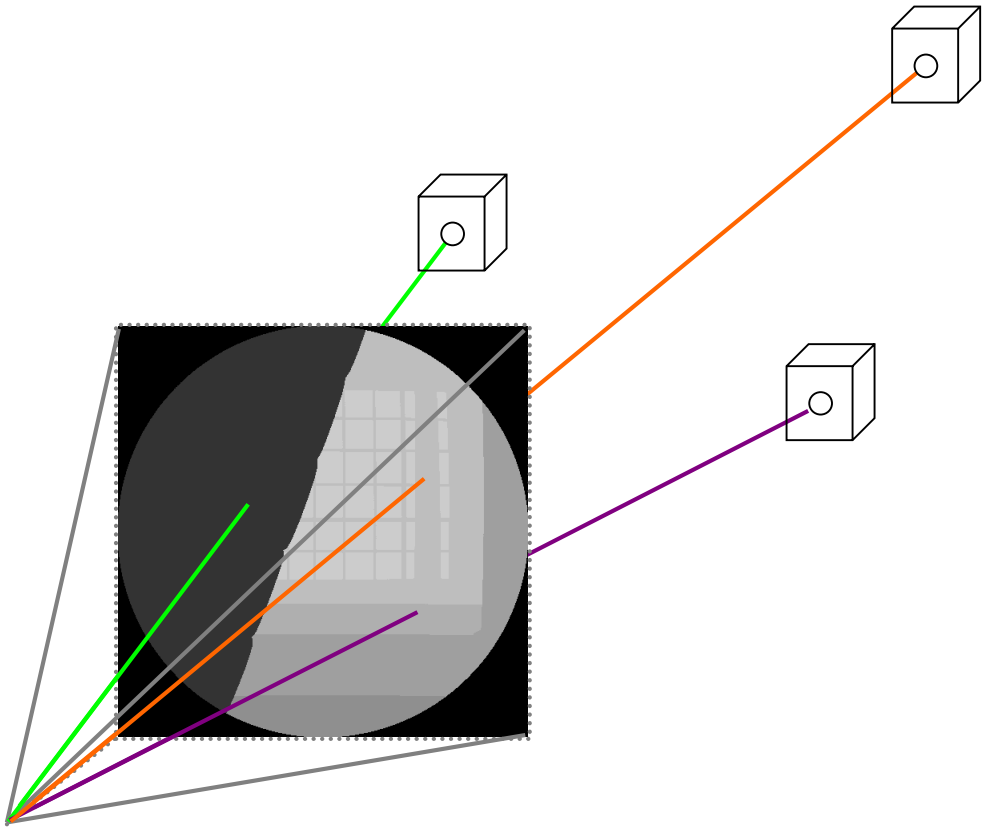
- Color image
- Depth image

- Depth-based interpolation of all other images

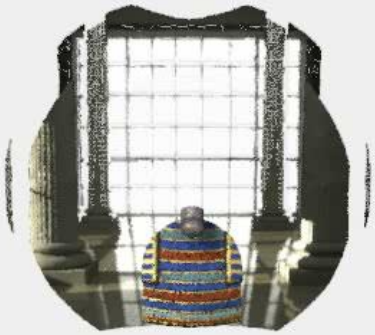


Interpolation Geometry: Octtree

- Set camera to lens position
- 1 Viewing ray for every depth pixel
- Generate 3d points from depth image
- Insert points into octree



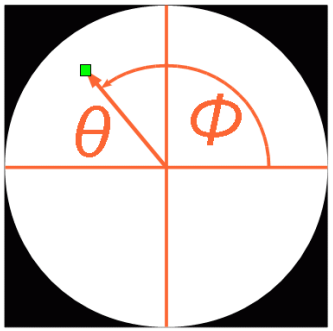
Video: Geometry



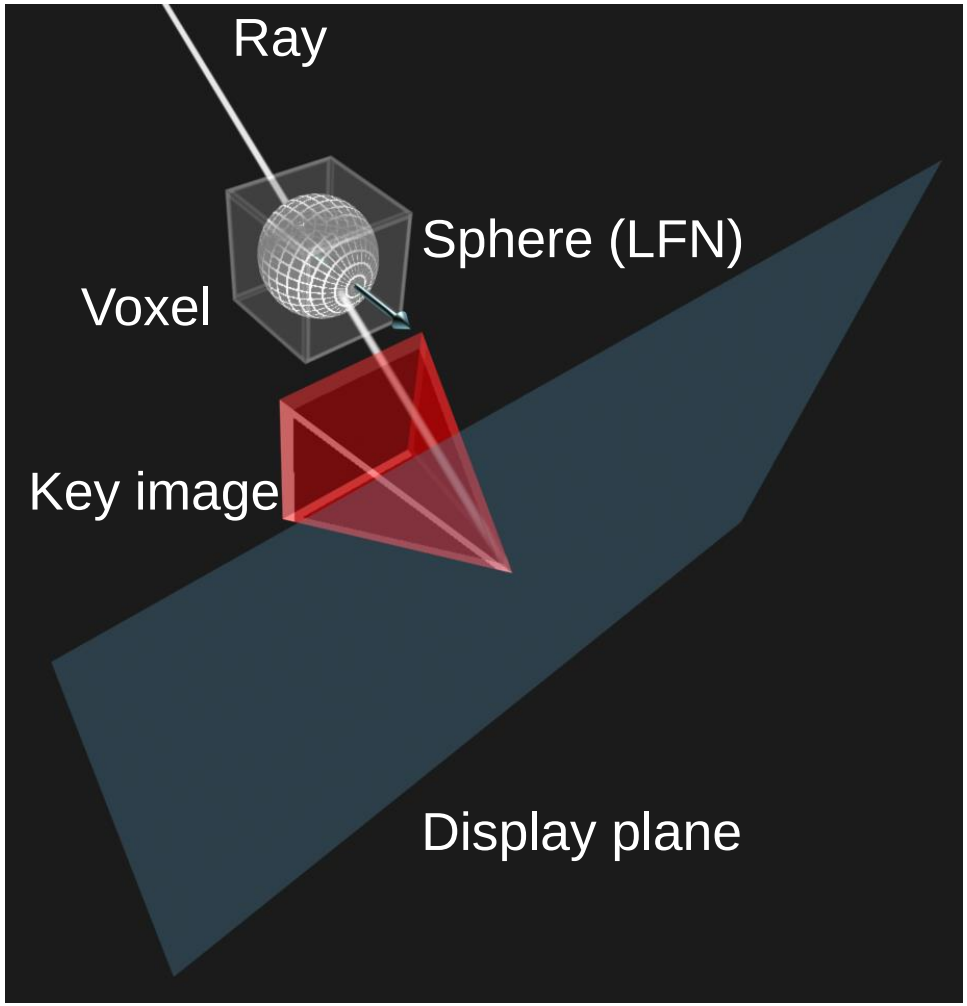
Color Information: Quadtree in Octree

- Ray cast for every pixel
- Intersection with voxel
- Sphere at every voxel position: Lightfield Node (LFN)
- Orientation orthogonal to display plane

Spherical image



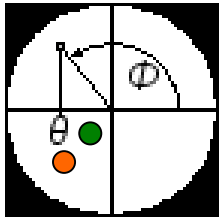
- Data structure for sparse images: quadtree

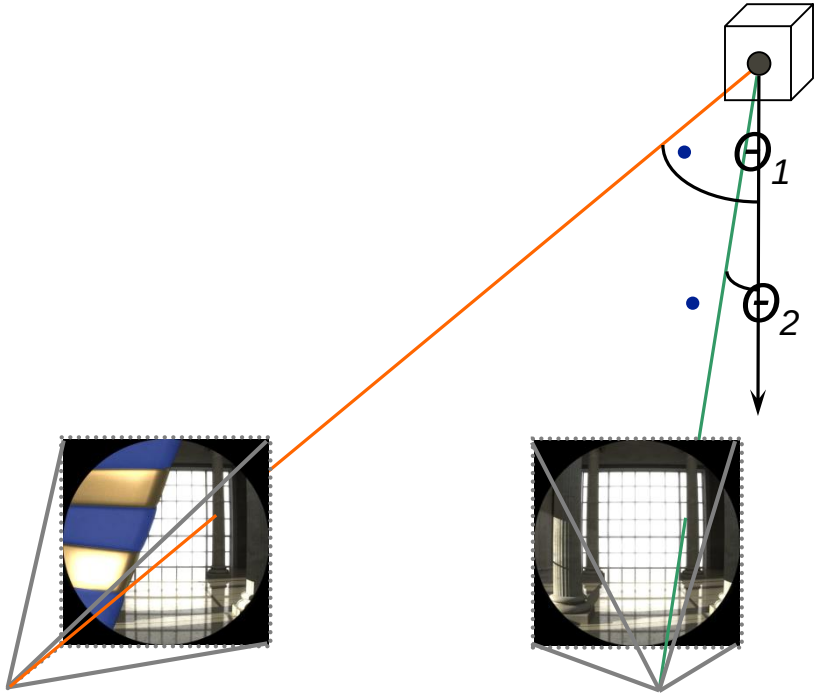


Sketch: view dependent color

View Dependent Color Information

- For diffuse surfaces only one color is added to the 3d point
- For reflective surfaces more color information has to be added
- Keep it simple!



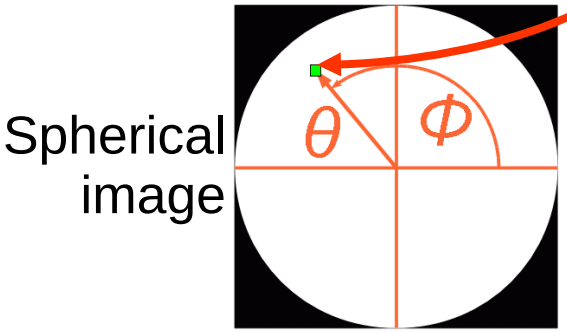


- Viewing rays intersect with the same voxel
- Projection of camera center into spherical image
- Elimination of redundant color
- Discard all empty nodes

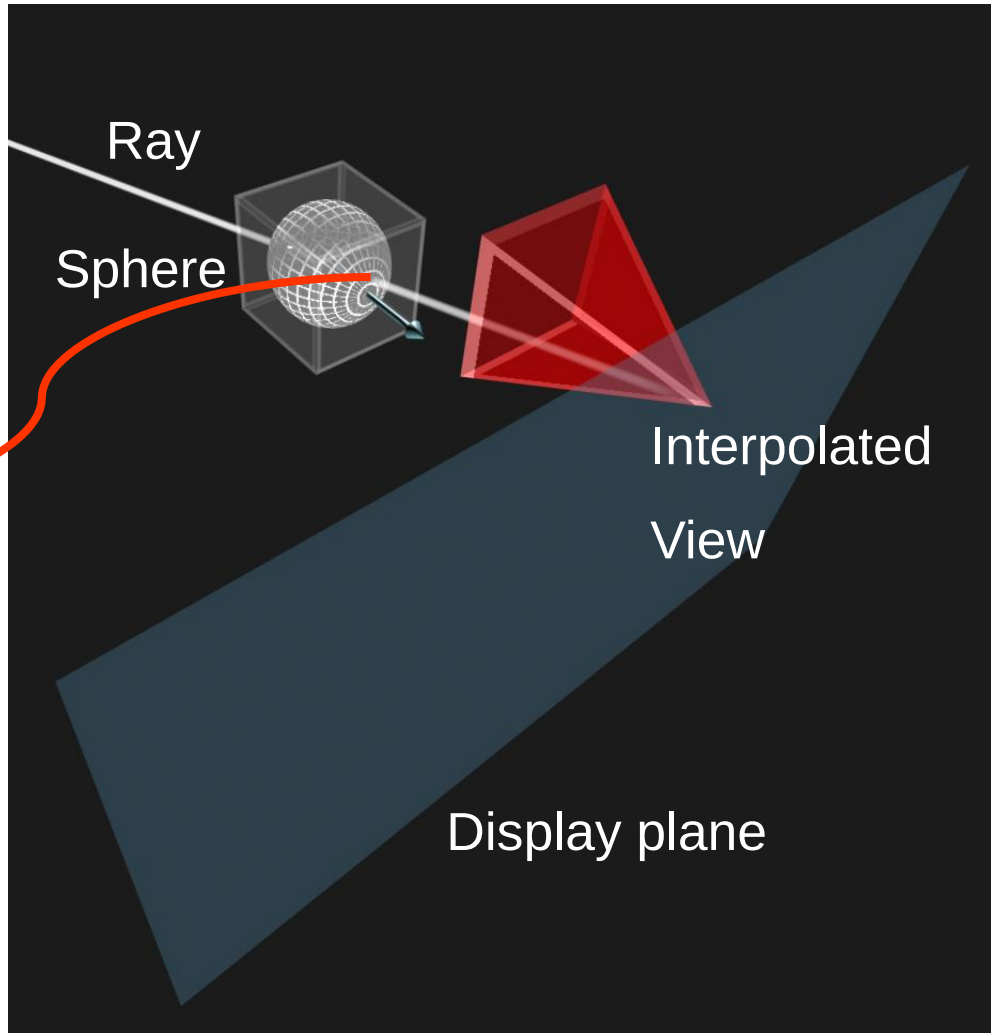
- Key image A
- Key image B

View interpolation

- Ray cast for every pixel
- Intersection with voxel
- Insert interpolated color into lens image



- Parallel rendering due to read-only data access



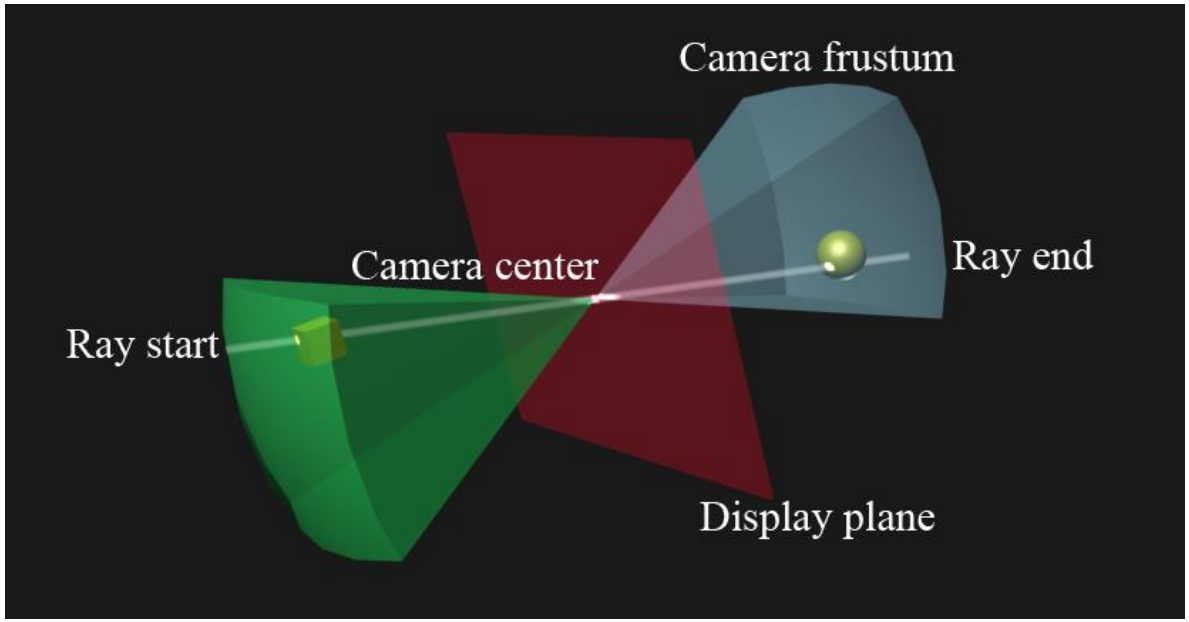
Sketch: rendering

Parallelization

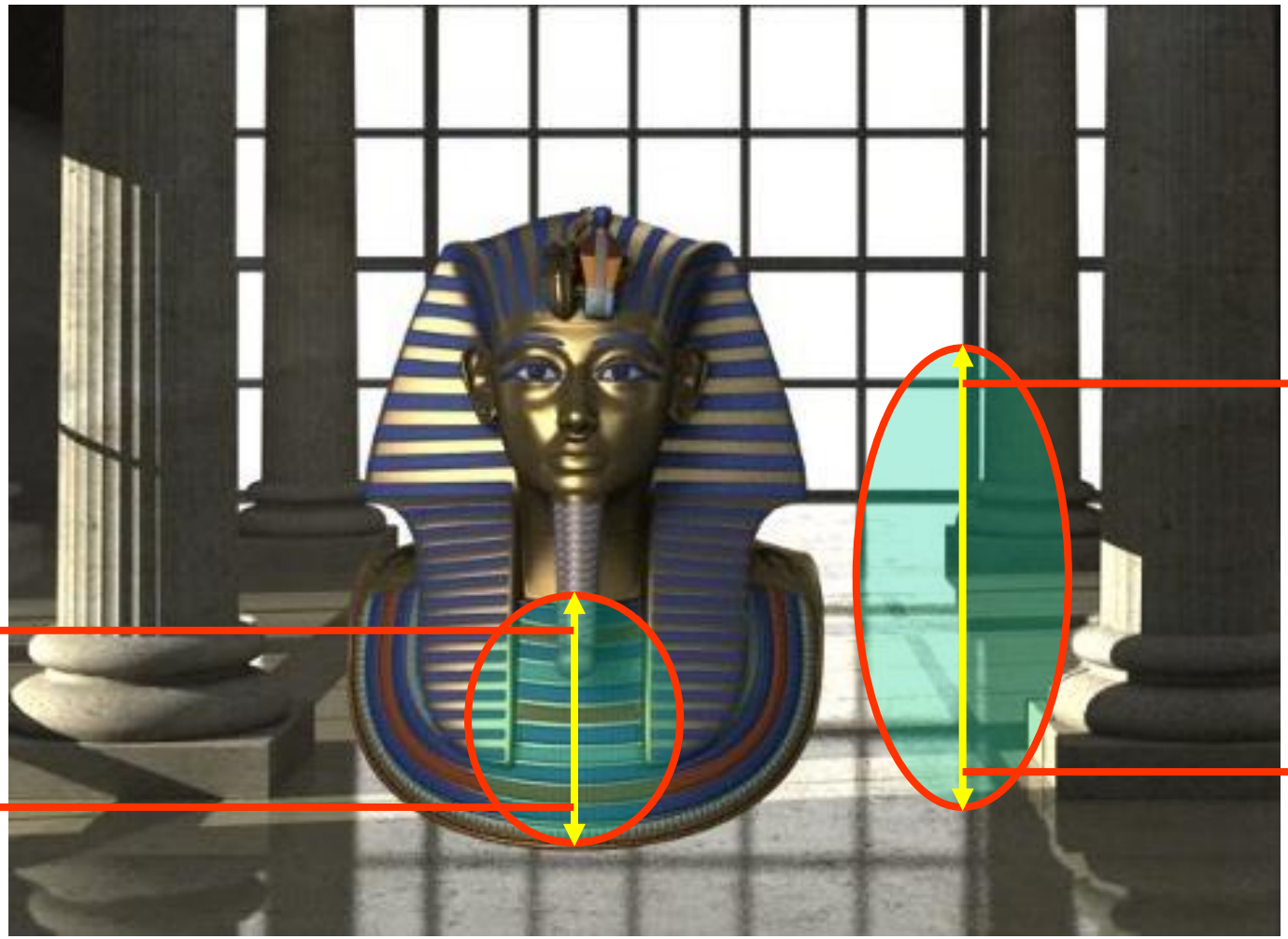
- During rendering access is read only
- Well suited for parallelization
- Parallelization for multi-core CPUs with OpenMP
- Parallelization on the GPU

Parallelization on the GPU

- Accelerate ray cast on GPU
- 2-pass rendering before/behind display plane
- Rendering with inverted z test for objects in front
- Angular color interpolation on the CPU

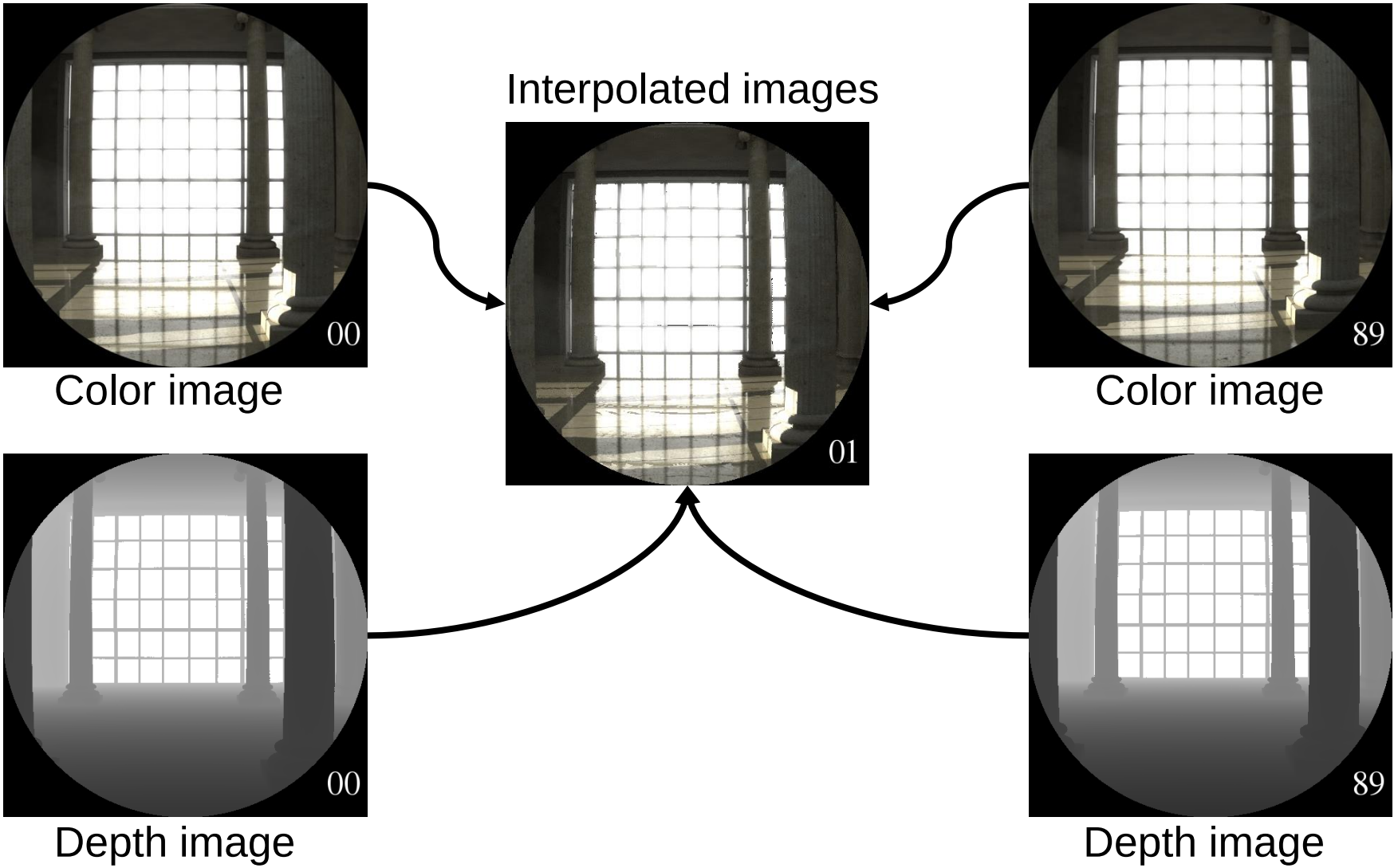


Interpolation Results

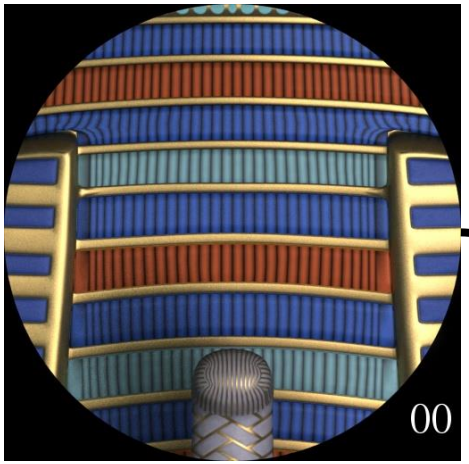


Ray traced overview image (3ds Max)

Results: Scene Background

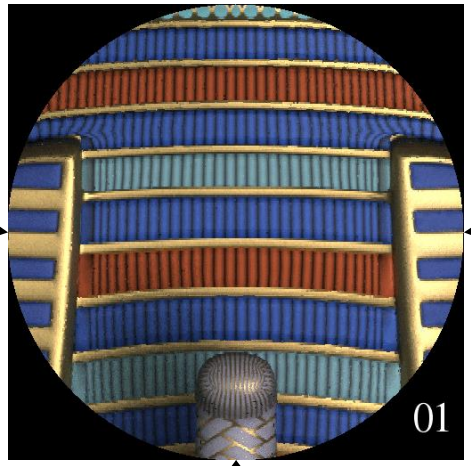


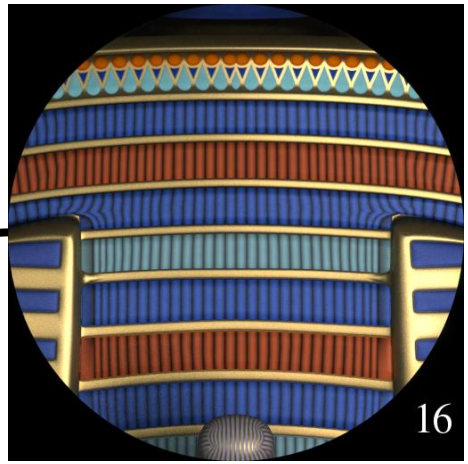
Results: Scene Foreground



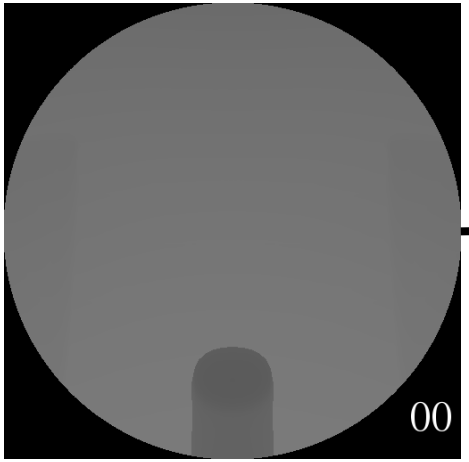
Color image

Interpolated images

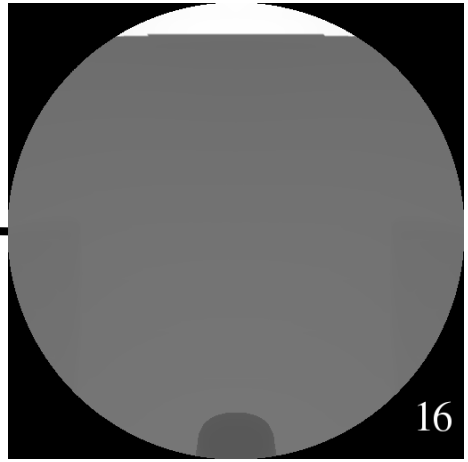




Color image

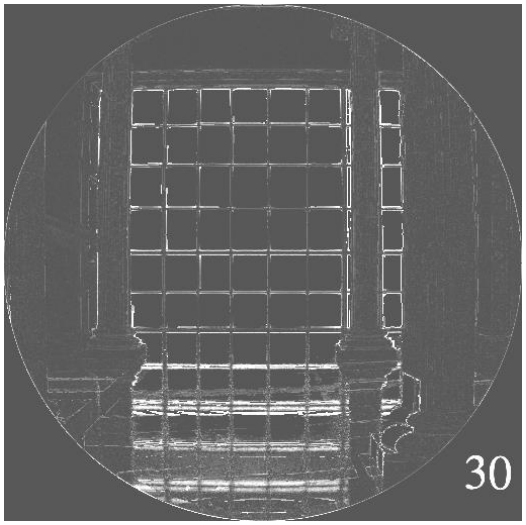


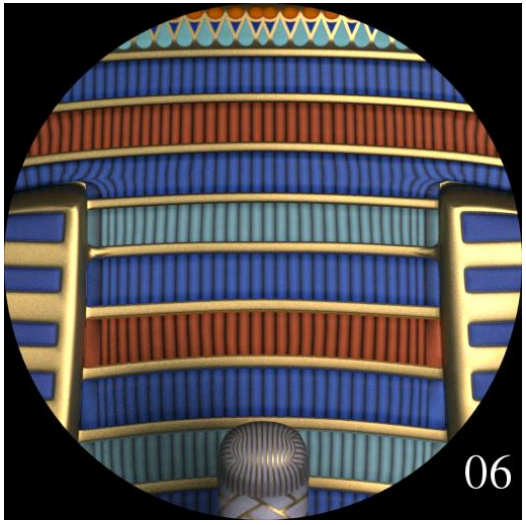
Depth image

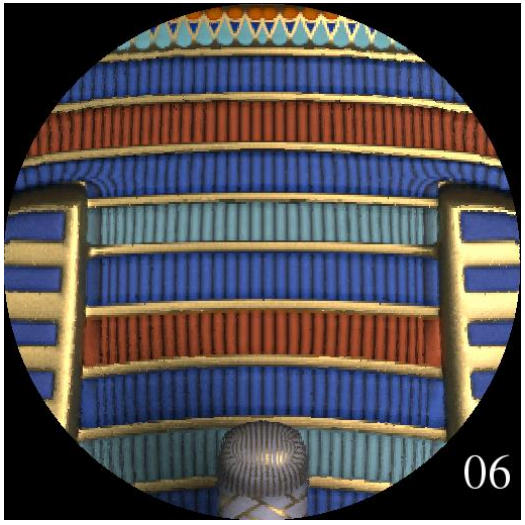


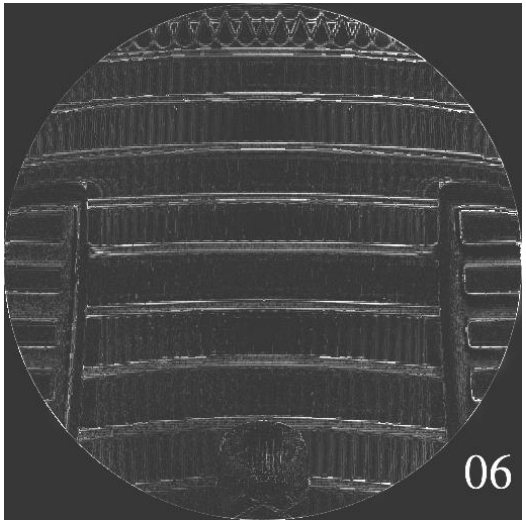
Depth image

Results: Interpolation



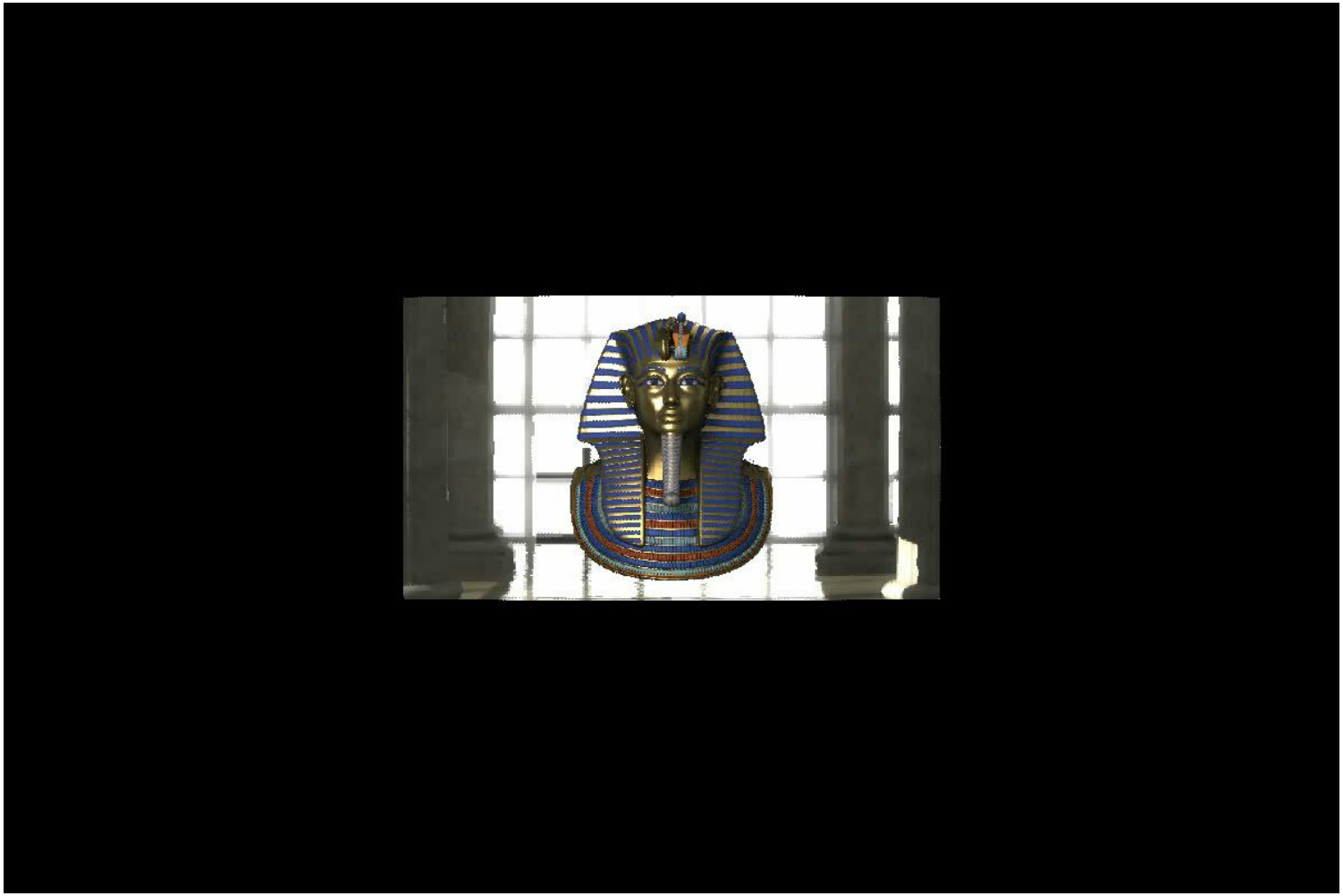






Ground truth images | Interpolated images | Difference images

Results: Simulated Display Animation



Simulated impression of the display from interpolated images

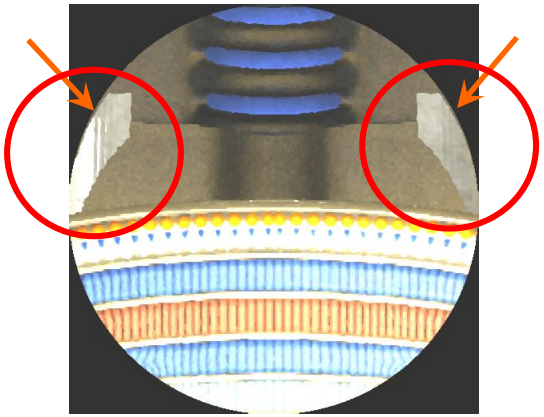
Adaptive Sampling

Generate all views:

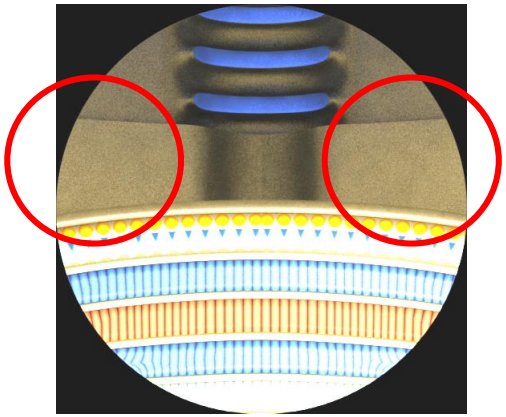
No interpolation

+ Perfect Quality

- Too expensive



Subsample 1/16²



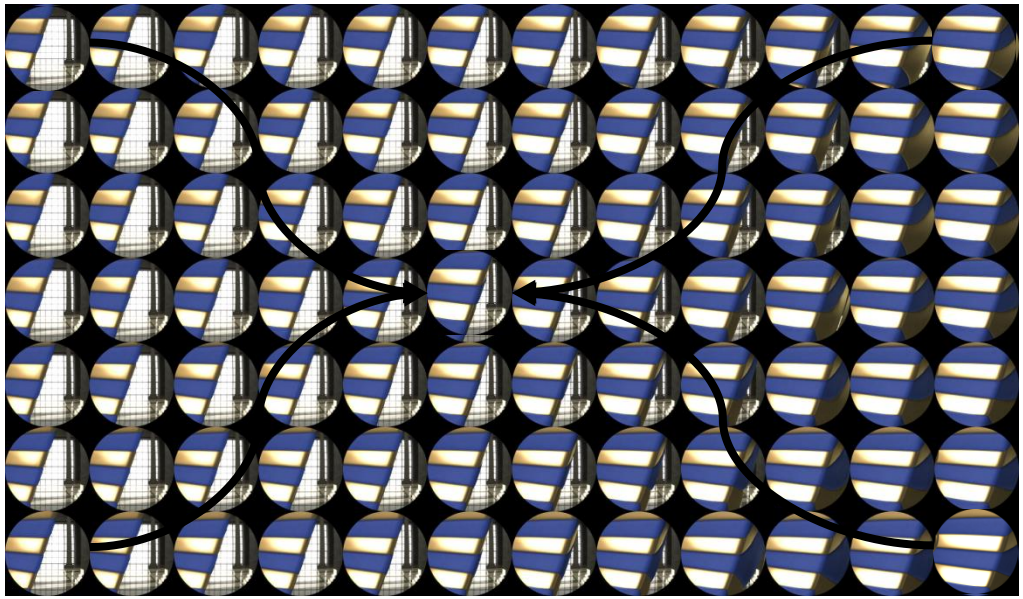
Perfect

Regular Subsampling:

Interpolation

+ e.g. 16² = Cheap

-Danger of missing details

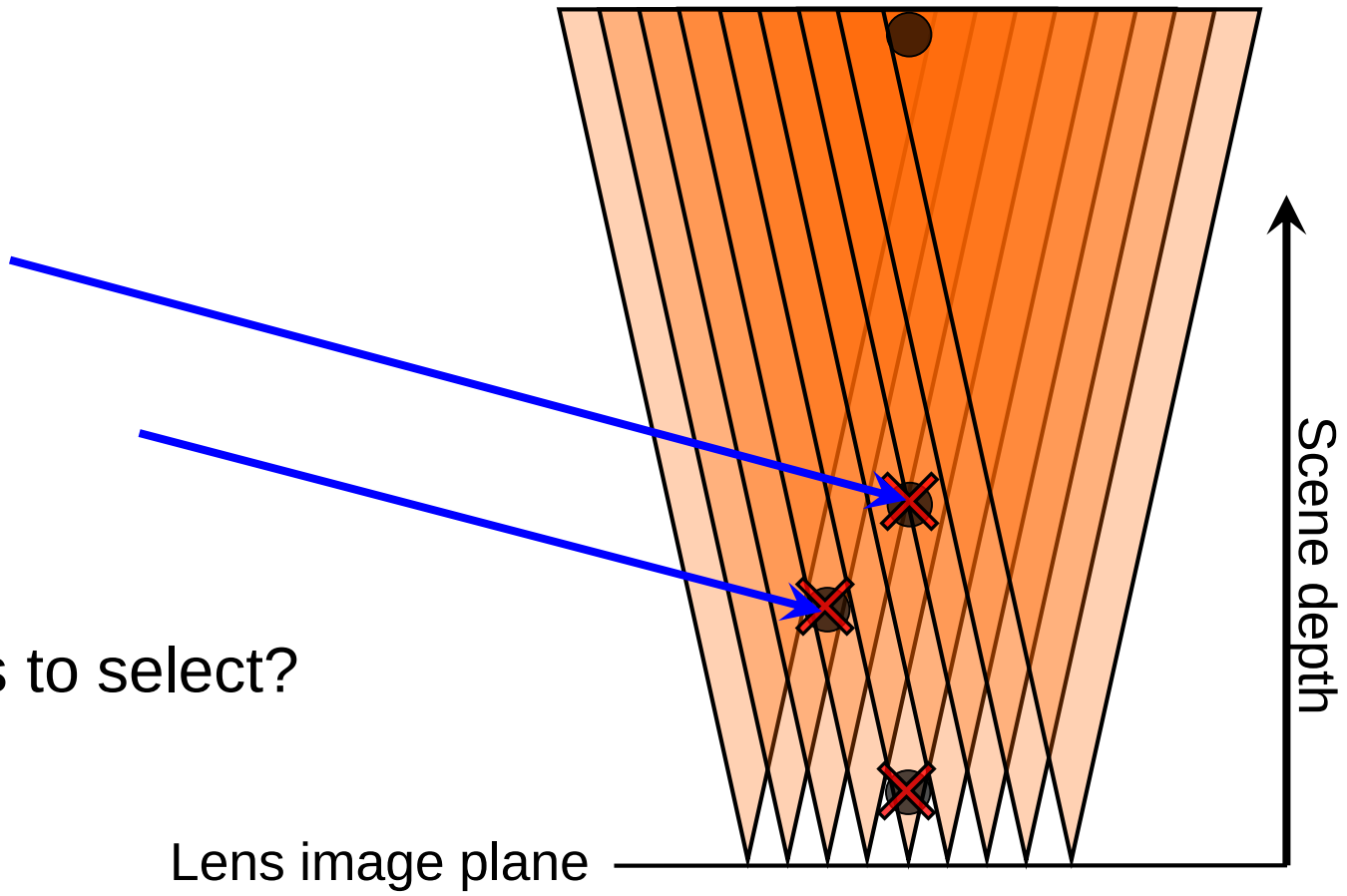


Adaptive Sampling

Reason for missing details:

1. Perfect
2. 2 x subsample
3. 4 x subsample
   - occlusion
   - outside of field of view
4. 9 x subsample

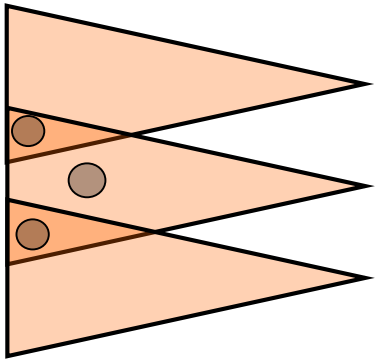
➢ Which images to select?



Adaptive Sampling

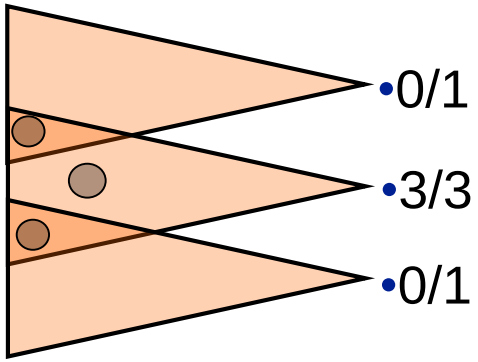
Solution – Analyse scene before interpolation

1. Geometry from depth images
2. Scene coverage analysis



Optimize:

1. Evaluate importance of lens image w.r.t. scene coverage
2. Compute minimal set of input images
3. Render image list with descending importance

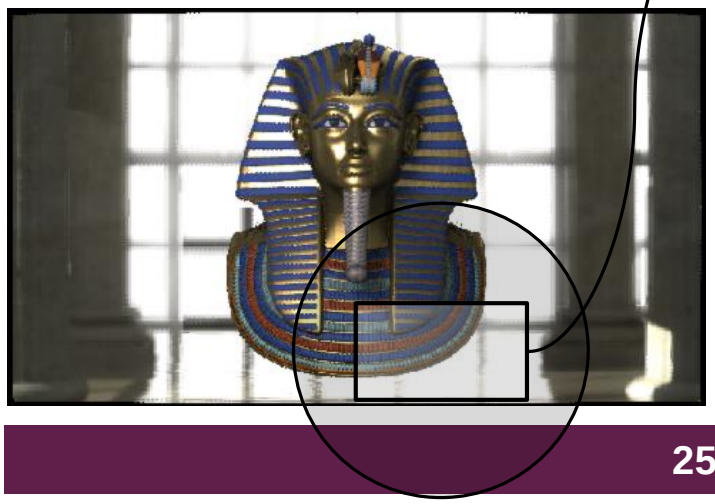


Adaptive Sampling

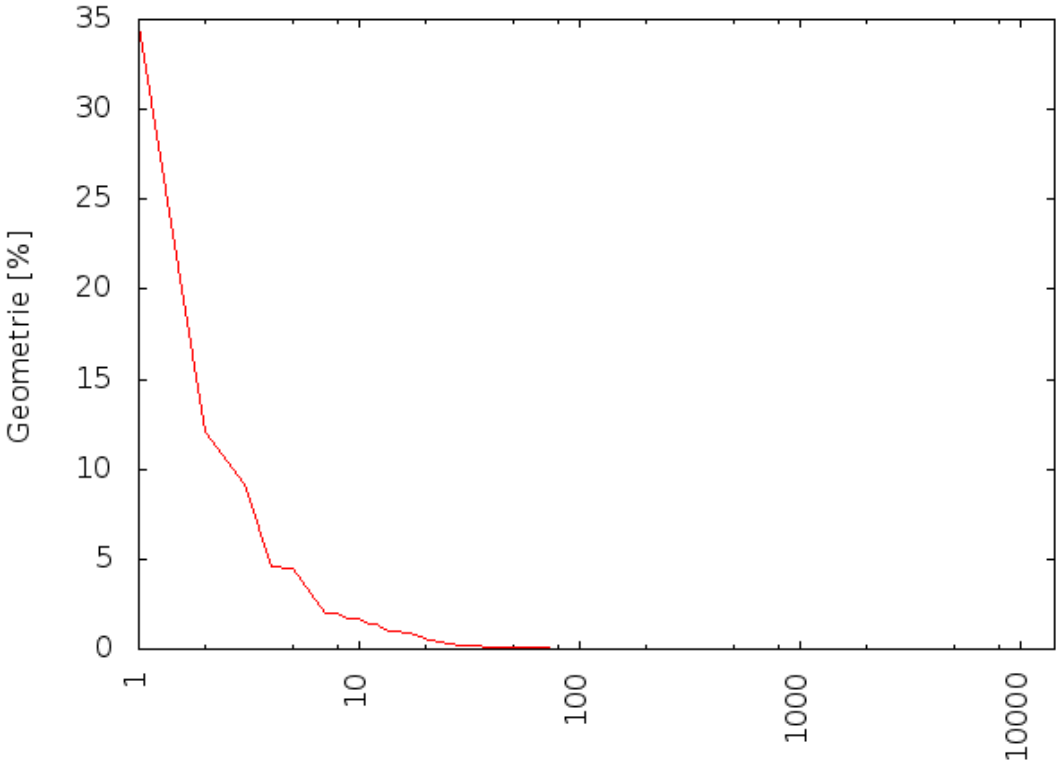
Analysis:

Display part with
14.400 images

Display



#images vs. scene innovation in %



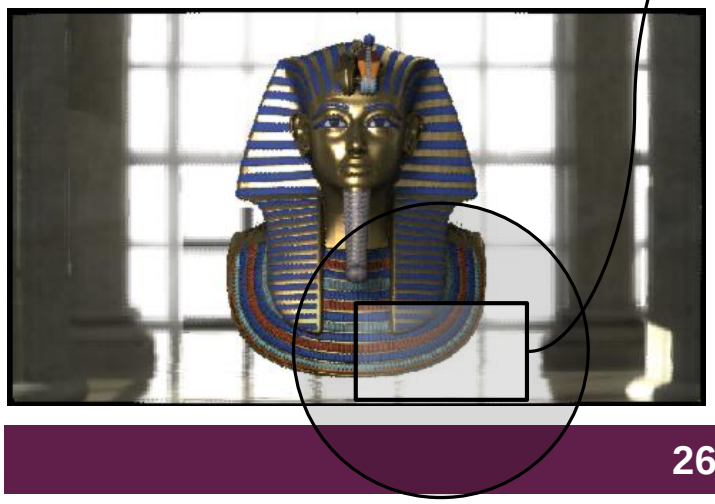
images, sorted by importance (log scale)

Adaptive Sampling

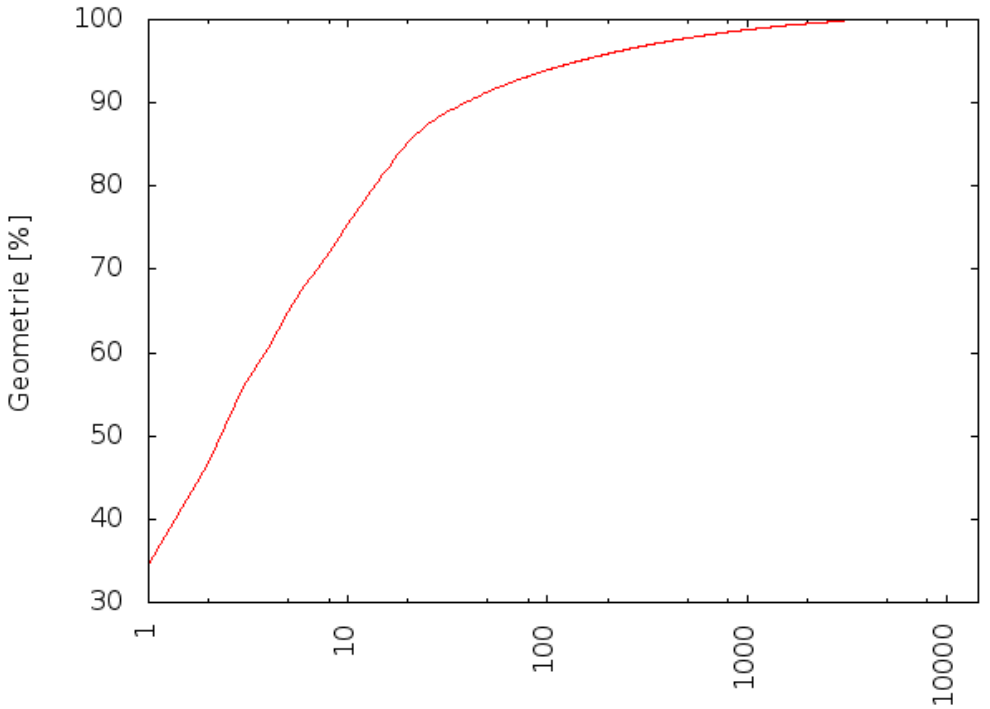
Analysis:

Display part with
14.400 images

Display



#images vs. Scene coverage in %



images, sorted by importance (log scale)

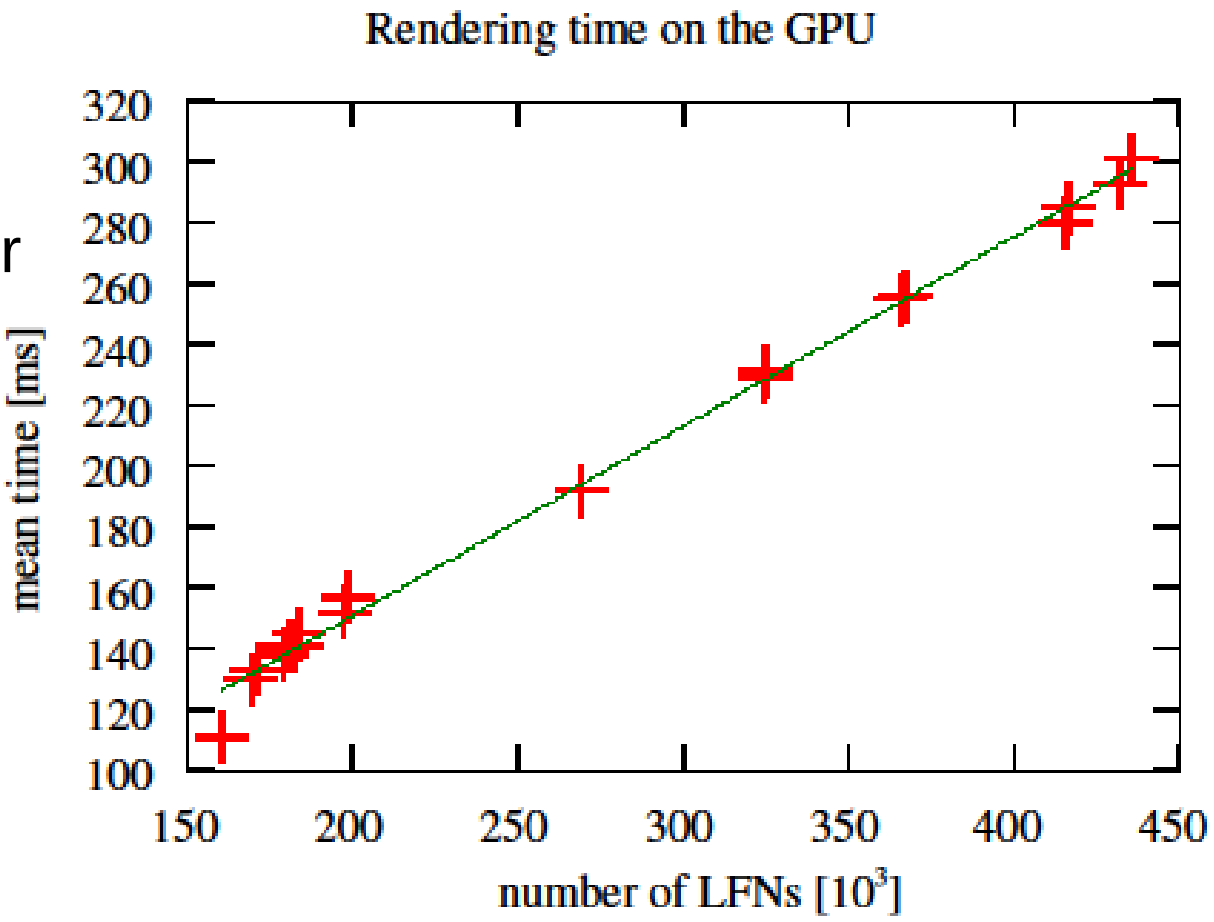
Storage Compression

	Number of images	[GB]	Percent
All images	230400	205	100
Color images	1232	1.10	0.54
Depth images	1232	0.74	0.36

- Overall compression factor of 111
- Reduce traffic over network
- Reduce storage space
- Missing images must be interpolated

Runtime Measurements

- Rendering time depends mainly on the number of LFNs
- Linear in the number of LFNs
- Average # of color samples per LFN between 2 and 7



Runtime Measurements

Ground Truth	[s]	Acceleration	Acc. Par.
3ds Max (VRay)	84,01	1,00	-
Interpolation	**[s]**	**Acceleration**	**Acc. Par.**
CPU*	1,96	43,90	1.00
OpenMP*	0,42	200,97	4.68
GPU**	0,16	535,07	12.47
Interpol. (incl. key frames)	**[s]**	**Acceleration**	**Acc. Par.**
GPU**	0,61	137,37	-

*Intel Core i7 CPU 920 @ 2,67 GHz

all images are 512x512 pixel

** NVIDIA GeForce GTX 285

Conclusion

- Rendering algorithm for structured input images
- Efficient data structures for sparse data and parallelization
- Accelerated rendering on the GPU with a speed up factor 100 – 500 (from 7.5 month to 1.5 days per m^2)
- Lossy data compression factor > 100 (from 205 GB to 1,85 GB)

Ongoing and Future Work

- Adaptive, scene dependent key image selection
- Detection, estimation and parametric representation of highlights
- Future challenge: How to produce full-parallax A0-size video with 24 frames/second ? (render time for 1 minute uncompressed rendering: 700 years and 285 Tbyte !)

Project Partners

realeyes

EUROMEDIAHOUSE
FOTOCENTRUM · DATASERVICE · ADVERTISING

IPM
Institut
Physikalische
Messtechnik

viaoptic

SOUL | PIX

C | A | U
CHILONIENSIS * SIGILLUM ACADEMIAE HOLSATORUM PAX OPTIMA RERUM A.D. 1665

Kleinhempel

IPT
Institut
Produktionstechnologie

MEUSER OPTIK GmbH
your partner for optical components

AutoPan Photo Lab Equipment