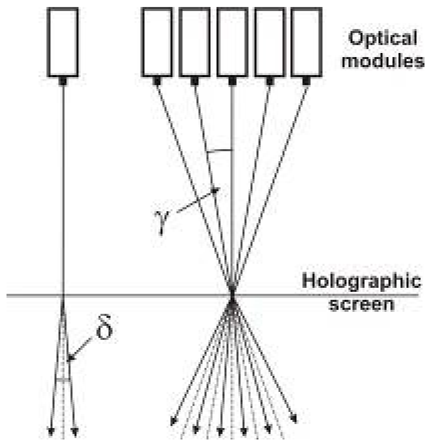
Multi everything:
parallelism in all shapes and sizes in HoloVizio light field display systems

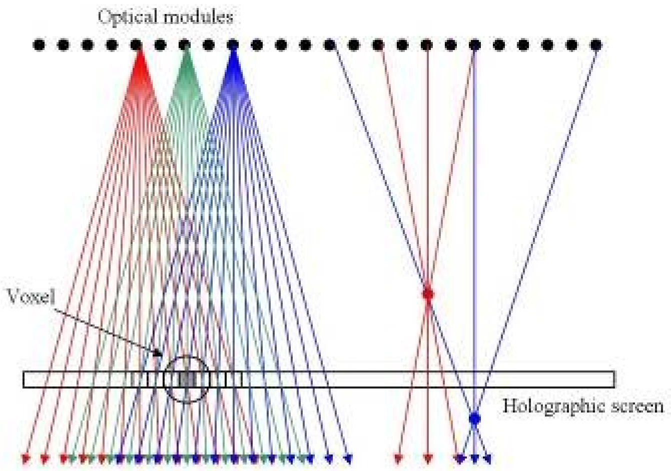
Attila Barsi
Holografika Ltd.

GPU Day 2016 - THE FUTURE OF MANY-CORE COMPUTING IN SCIENCE
03. 05. 2016. Budapest

The HoloVizio System

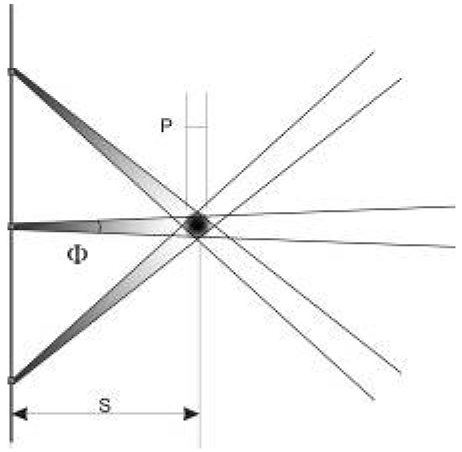
- Optical modules
  - Project light beams to hit the points of a screen with multiple beams under various angles of incidence
  - The exit angle depends only on the relative position of the given screen point and the relevant modules
- Holographic screen
  - Direction selective property with angularly dependent diffusion characteristics
  - The screen diffusion angle d is equal to the angle g between the neighboring modules
- Emission angle geometry determined
  - No optical road-blocks like at Fresnel, or lenticular lenses

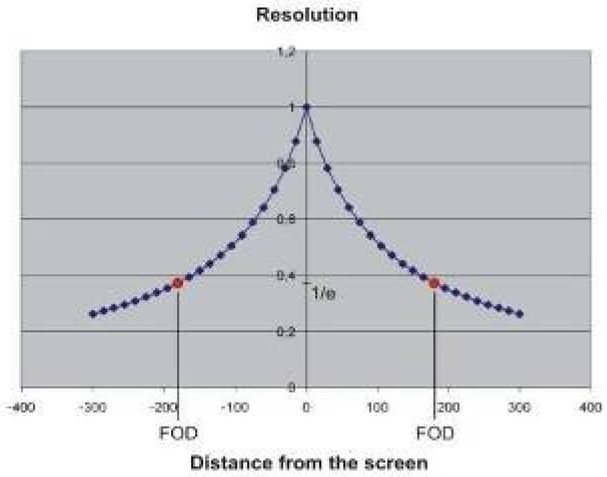




The HoloVizio System

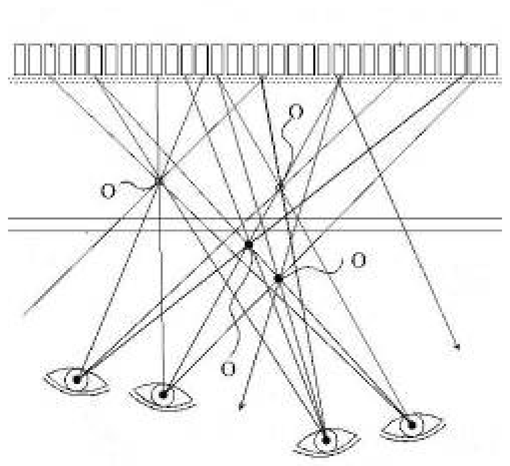
- FOV
  - Controlled FOV determined by the module optics and arrangement
  - Wider FOV, more complexity; the freedom given to the viewer is the challenge
  - No favorable zone in the FOV, viewers can freely move in the whole FOV area
- Angular resolution / FOD
  - Large number of light beams can be emitted from the same small screen pixel area
  - The angular resolution determine the FOD
  - The smallest feature (voxel) the display can reconstruct is the function of the angular resolution and the distance from the screen ($p = p_o + s * \tan \Phi$)
  - The achievable resolution is decreasing with the distance from the screen

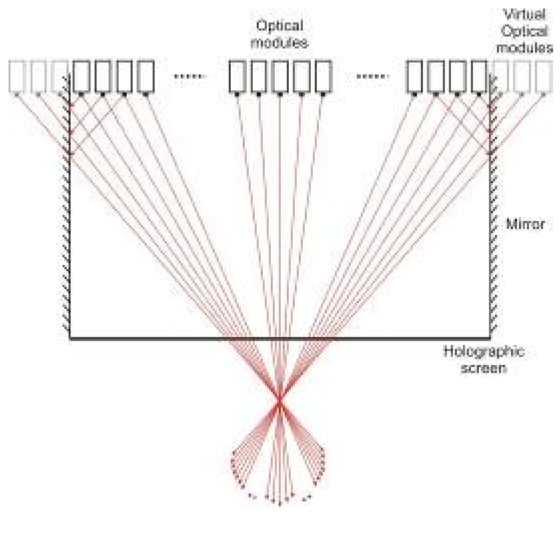




The HoloVizio System

- Light field reconstruction instead of views
- Specific distributed image organization
  - A module is not associated to a direction
    - The projected module image not a 2D view of the final 3D image
  - Each view of the 3D image comes from more modules
    - The number of contributing modules does not change over the FOV (no point in the FOV where only one single module image would be seen, like at multiview)
  - Distributed changes within the 3D image on different image areas
    - Smooth and continuous transition, no single border occur between views
    - Continuous motion parallax

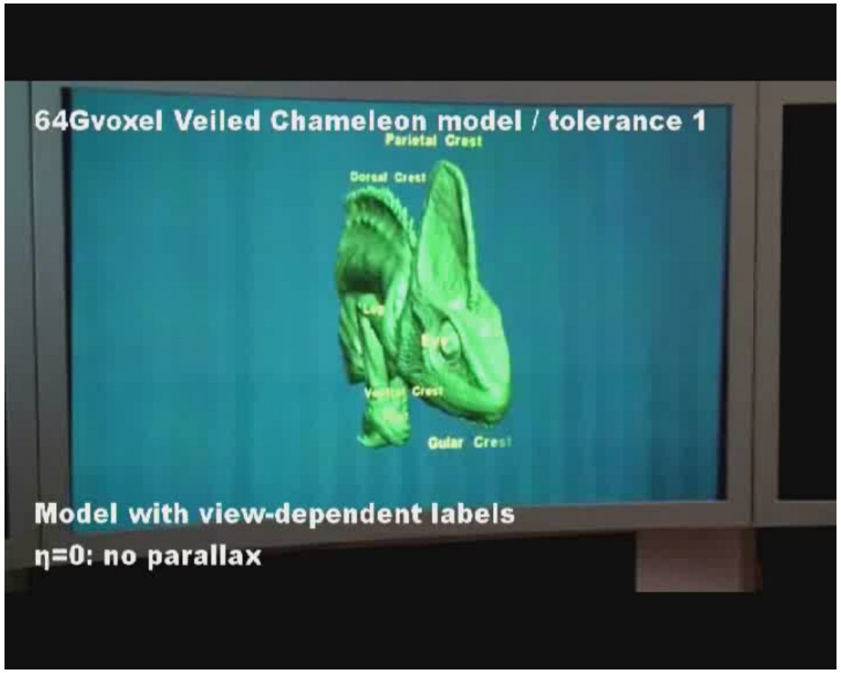




HOLOGRAFIKA

HoloVizio Displays

- Large-scale HoloVizio Systems
  - HoloVizio 722RC
    - 72”, 34.5 Mpixel, 16:9
    - 50-70° FOV, 0.9° Φ
    - Input: Gigabit Ethernet
    - PC-based render cluster
    - LED based projection engines
      - Better colours
      - More uniform image
      - Less maintenance


64Gvoxel Veiled Chameleon model / tolerance 1
Parietal Crest
Dorsal Crest
Gular Crest
Model with view-dependent labels
η=0: no parallax

HoloVizio Displays

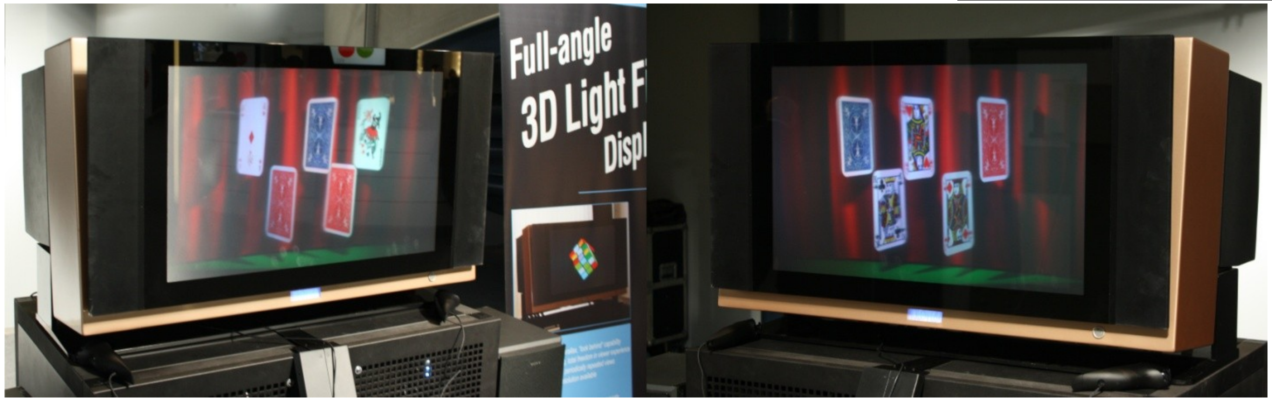
- The full angle HoloVizio monitor

- **HoloVizio 80WLT**
- 78 Mpixel, 30" (16:10)
- 180 degrees FOV
- Total freedom 3D experience,
  no invalid zones,
  no repeated views

- 2D equivalent image resolution 1280 x 768 (WXGA)
- LED colors
- Multiple DVI inputs

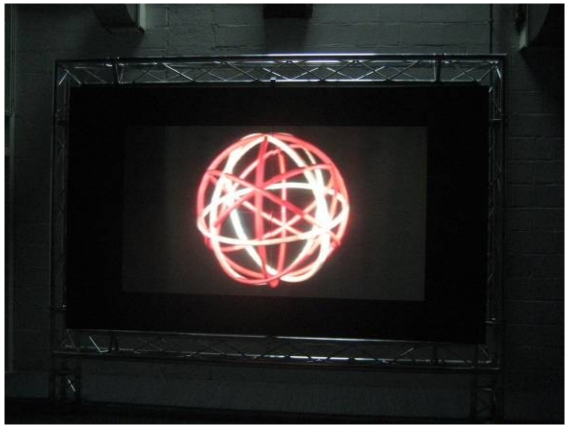
HOLOVIZIO™ 80WLT

Full-angle
3D Displaying



HoloVizio Displays

- The world first glasses-free 3D cinema system:
- HoloVizio C80
  - 3,5 m dia Holoscreen (140”)
  - No glasses, no optical contradiction
  - LED based 3D projection unit
  - Exceptional 1500 Cd/m2 brightness
  - 40 degrees FOV
  - 2D compatible
  - Fitting cinema rooms, 3D simulators



Benefits

- View with naked eye
  - No glasses
  - No tethers
  - No tracking
  - No positioning
  - No image jumps
- Wide Field Of View
- Virtual object position
  - Behind screen
  - In front of screen
- Look behind objects
- Multi-user collaboration
- 3D movie is more like a theatrical performace

In short, none of these!

She’s got the right idea!

Light field rendering

- Challenges
  - Do everything that you would do on a 2D display and more on up to 80 megapixels
  - Make everything multiplatform (Windows/Linux)
  - Drive up to 80 graphics outputs (DVI, HDMI, DP)
  - Synchronize all renderers
  - Do a non-linear projection that matches the optics
  - Rendering compared to 2D on our worst case cluster:
    - Render 5120* 2880 pixels compared to the usual case of 1920*1080 on a single GPU!
    - Render 48 times the geometry!
    - Do this on three GPUs for a single PC (CPU load!)
  - Most post process effects are not working out of the box due to the non-linear nature of the primary rays
  - Using commercial renderers is out of the question (pinhole camera model and linear screen space is in the [illegible] of every renderer)

Use cases

- Proxy OpenGL library
  - For existing applications, without plugin support or closed rendering methods due to IPR
- Light field video player
  - To allow cluster reconfiguration, video is stored to allow decompression per optical module
  - Calibration
- 2D image array to light field conversion
  - Allows rendering from existing engines (CAD/CAM)
  - Allows rendering from real cameras
- ClusteredRenderer for everything else
  - Native applications
  - Plugins for existing applications
  - Raytracer

Network level parallelism

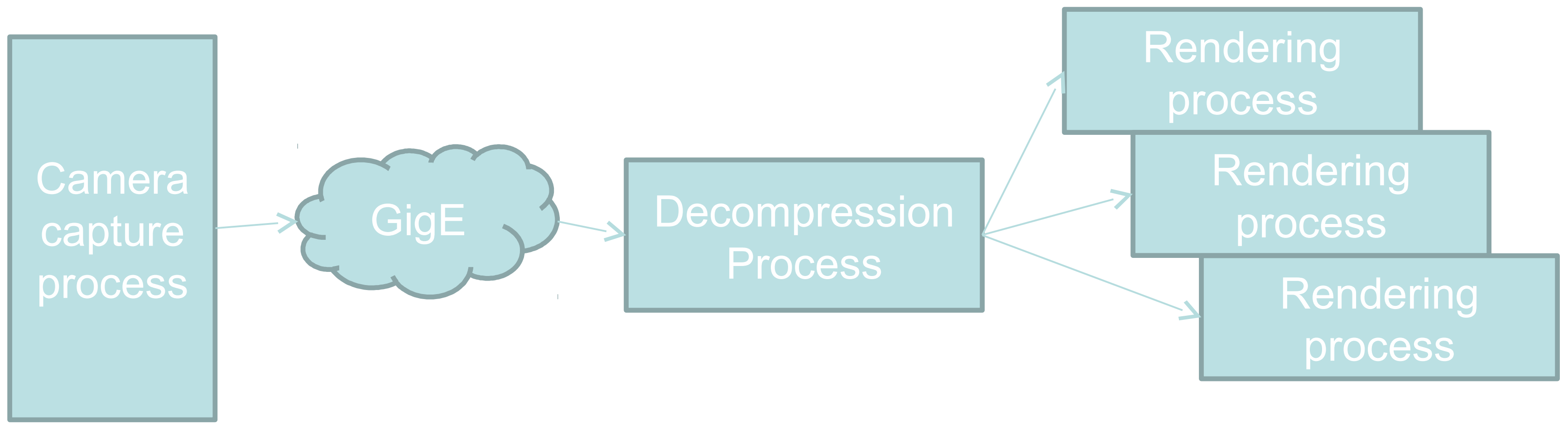
- All displays have multiple inputs (2-80!)
- Currently a maximum of 4 outputs can be driven simultaneously from a single NVIDIA GPU and 6 from an AMD one
- A single PC can house a maximum of 4 dGPUs (3 if you consider heat buildup)
- GPU clusters are a must
- Using custom Fedora Linux OS as server and nodes

We use every type of network from IB to 40 GbE

Cluster runs on GPUs and junk food

Process level parallelism

- X screens are per GPU due to acceleration model (no Mosaic SLI on GeForce). This way all GPUs render at full screen.
- OpenGL context switches between threads are slow (glXMakeCurrent() calls). Solution: one process for rendering per GPU. Completely scalable.
- Additional processes are mostly used for conversion tasks from compressed images (single decompress in a process, shared memory copy to all renderers).



Multithreaded parallelism

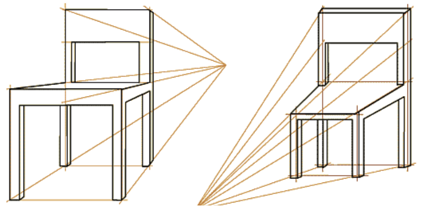
- Video playback
  - Multithreaded decompression via FFMPEG
  - Each video stream has it’s own decompressor thread
  - Rendering on separate thread
  - Download to GPU via pixel unpack buffers (See https://www.opengl.org/wiki/Pixel_Buffer_Object#Downloads)
  - Buffered to avoid speed inconsistencies
  - Imagine debugging video seek for 40 video streams
- Multi camera MJPEG decode
  - TurboJPEG decompressor

Massively multithreaded parallelism

- Real-time rendering
  - OpenGL 4.x AZDO forward and deferred renderer
  - Two slit projection. Linear approximation of real optical system.
  - Needs calibration post effect.
  - Uses a combination of viewport arrays and instancing for fast rendering.
  - Materials from bindless textures and shader buffer objects.
  - Shadow mapping.
  - Order independent transparency.
  - Physically based rendering (ongoing research).
  - HDR tone mapping (ongoing research).

Two-slit projection

- Reverse perspective in X, normal perspective in Y
- Projection is not a single 4x4 matrix
- Projection parameters are stored in UBOs for speed
- Instance ID indexes projection
- Viewport arrays are used to send pixels to the right image part
- Projection is correct per vertex
- Tessellate or use high res models to avoid incorrect projection


Perspective and reverse perspective

Reverse perspective, lines converge towards the viewer

Culling

- Display geometry’s observable area can be represented with 6 planes
- Cutting becomes generic for 2D and 3D cases
- Use compute shader to cut geometry for the display
- After cutting, every mesh is in the same buffer. Single draw call rendering for the complete scene.

Materials and shadows

- Bindless textures (best thing ever!)
- Everything is in one SSBO.
- Eye positions are per pixel (read from texture)
- Materials are evaluated in world space
- Also useful for ray tracing.
- PCF shadows are rendered into 2D texture arrays (directional and spot lights) and cube map arrays (point lights).
- Single pass for each texture type with instancing and texture array layering.
- Cascaded rendering for directional lights is supported. Light cascades are calculated for the double frustum of the display.

```
struct MaterialDescriptor
{
  uint64_t ambientMap;
  uint64_t diffuseMap; //128 bits
  uint64_t specularMap;
  uint64_t shininessMap; //256 bits
  uint64_t normalMap;
  float shininess;
  float roughness; //384 bits
  vec3 ambient;
  float reflectionCoefficient; //512 bits
  vec4 diffuse; //640 bits
  vec3 specular;
  float padding; //768 bits
}
layout(std430,binding = 4) readonly buffer MaterialBuffer
{
  MaterialDescriptor materialDescriptors[];
};
```

Order independent transparency

- Render fragments into a large fragment buffer with image load/store
- GPU linked list implementation
- Sort fragments in a compute pass
- Render opaque first, disable depth writes, but keep depth test to throw away fragments masked by opaque
- If OIT is on, the compute shader will separate scene geometry by material transparency

```
layout(std430,binding = 0) coherent buffer Abuffer
{
  uvec4 imgABuffer[];
}
uniform int aBufferSize;
layout(r32ui,binding=4) uniform coherent uimage2D imgListHead;
layout(offset=0,binding=0) uniform atomic_uint counter;
//Somewhere later in main()…
uint idx = atomicCounterIncrement(counter)+1u;
uvec2 fragment = uvec2(packUnorm4x8(finalColor),floatBitsToUint(gl_FragCoord.z));
if(idx < uint(aBufferSize))
{
  uint prev = imageAtomicExchange(imgListHead,ivec2(gl_FragCoord.xy),idx);
  imgABuffer[idx] = uvec4(fragment,gl_SampleID,prev);
}
```

Tone mapping

- Converts HDR image to SDR image
- Based on Erik Reinhard’s work.
- Global and local variant
- Uses image atomics to calculate SAT (very fast)

Ray tracing

- GPU Ray tracing
  - Ray generation (implemented in OpenGL, OpenCL, NVIDIA OptiX). Complete optical model. Best solution for light field rendering.
  - Accelerators BVH, KD-Tree.
  - Ray tracing (NVIDIA OptiX, OpenGL version in R&D).
  - Still needs to be faster

HOLOGRAFIKA

Light field conversion

- Converts from 2D image array to light field
- Size of accelerator table that indexes input depends on the number of output pixels
- Source can be as large as maximum texture size
- Allows high quality conversion to light field video

Be there in 3D!

Telepresence with
HoloVizio 3D display

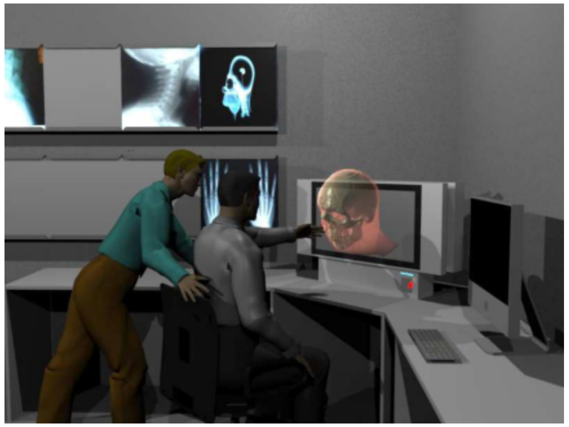
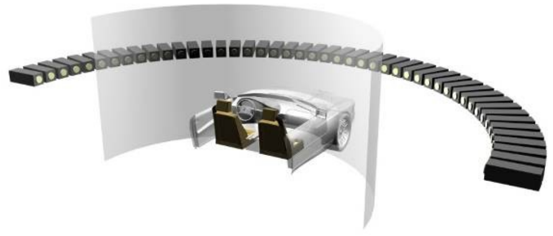
2009

Tl;dr on OpenGL rendering

- Use the graphics pipeline for free goodies not available in compute
  - Depth & Stencil buffer for cheap comparisons (early Z out is really powerful)
  - Blending for min, max, missing multichannel atomics and much more
  - Bindless textures and arrays for flexible data structures
  - Permanently mapped buffers
  - Instancing and viewport arrays when you need to run different algorithms on the same data
  - Use indirect rendering and compute to avoid readbacks to CPU
- OpenGL compute
  - Does not require interop. No need to go the OpenCL or Cuda route
  - Don’t forget restrict
  - Don’t forget to specify readonly or writeonly
  - Use atomics instead of barrier & friends if you can do it (much cheaper)
  - Only caveat, you have to specify the warp size in advance. (Why, Khronos, why???)
- AZDO can boost your performance to very good levels without having to go to Vulkan levels of complications
- If at all possible, avoid readbacks to CPU at all costs
- Deferred rendering is no longer necessary (finally)

R&D-current & future work

- 3D Telepresence
  - Natural communication
  - Eye contact, directive gestures
- Towards Gigapixel displays
  - 100Mpixel today, high pixel count systems with parallel distributed approach
- Large-scale 3D visualization
  - Scalable
  - Glasses-free CAVE-s, no edges, no corners
- 3D Internet connected TV
  - End-to-end 3D video systems through direct Internet connection
  - Two-ways communication, dynamic 3D models, personalized programs with local 3D render
- 3D formats
  - Scalable and generic 3DTV representation format
    - 3D Light field representation
  - Inherent 3D compression technologies

HOLOGRAFIKA

Current research projects and training networks

- ETN-FPI - European Training Network on Full Parallax Imaging
- QoE-Net - Innovative Quality of Experience Management in Emerging Multimedia Services
- OptIntegral - Advertisement displays manufactured by hybrid in-mould integration

european training network on full parallax imaging

QoENet MSCA-ITN Network

OPTintegral

Questions?

Acknowledgements:
The Holografika Team http://www.holografika.com/
All the good people @ NVIDIA, AMD, and Khronos
Computer Graphics Group @ BME,
especially Dr. László Szirmay-Kalos and Dr. László Szécsi; thanks for teaching me so much about GPUs & rendering