HoloVizio 3D Light-Field Displays

Péter Tamás Kovács

Holografika

MPEG FTV Seminar
8th July, 2014
Sapporo, Japan

Holografika

- Hungarian company, active in the holographic, opto-electronic and 3D display technologies
- Manufacturing next generation holographic displays and developing other 3D related technologies (3D acquisition, 3D conversion, rendering, 3D applications) with proprietary IP
- Experienced team for optical design, hw & sw, etc.
- Patent portfolio
- Red Herring Top 100 Europe in 2005, WEF Technology Pioneer company of 2006, shortlisted for World Technology Award 2006, nominee for the EU ICT Prize in 2007, silver Best Exhibit Award at 2008 ICT Lyon, One To Watch @ GPU Technology Conference 2014

What is possible?

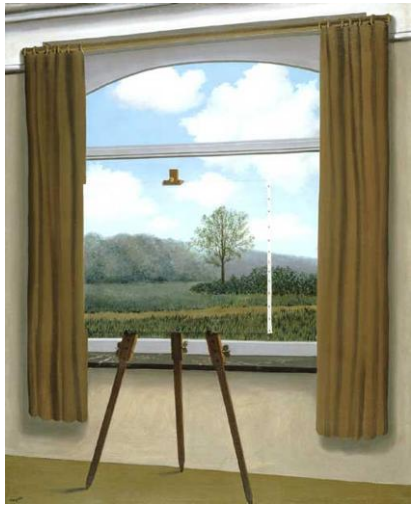
- *„We can make miracles immediately, for the impossible you should wait a few days...”*

- Law of physics – *today...*
  - Is it impossible or just difficult?
  - Easy self-check to look like an expert: “C*ould the light beams reach my eyes, as shown?”*
  - No one ever saw a light ray making a turn in free space – “*Where did those light beams come from?”*

The Window

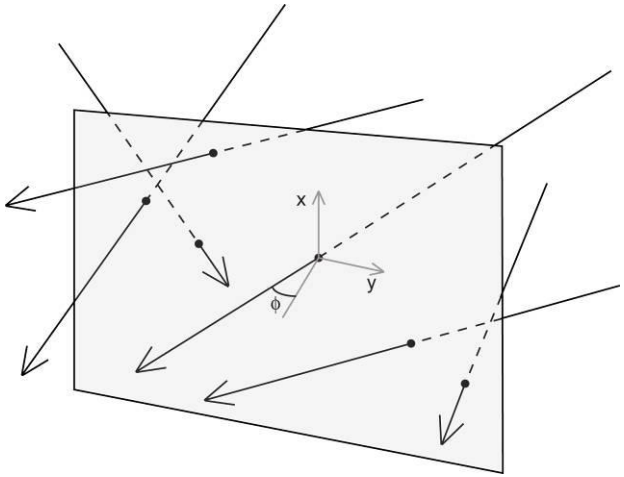
- The goal of displaying is to provide perfect representation of real/synthetic scenes
  - Life-like view
  - The ultimate display will be like a window...

- True 3D displaying - reconstructing the light-field as present in the natural view
  - Producing light beams with the same parameters the human perception is capable to process: direction, position, intensity, color (but polarization, phase)
  - „…Let the display work, not the brain…”



Light Field

- General representation of 3D information that considers a 3D scene as the collection of light rays that are emitted or reflected from 3D scene points. (M. Levoy, and P. Hanrahan, “Light Field Rendering”, 1996.)
- The visible light beams are described with respect to a reference surface (screen) using the light beams’ intersection with the surface and angle.
- The LF is defined as a function of position (2 or 3 parameters) and direction (2 parameters): L(x,y,[z],Θ,Φ)

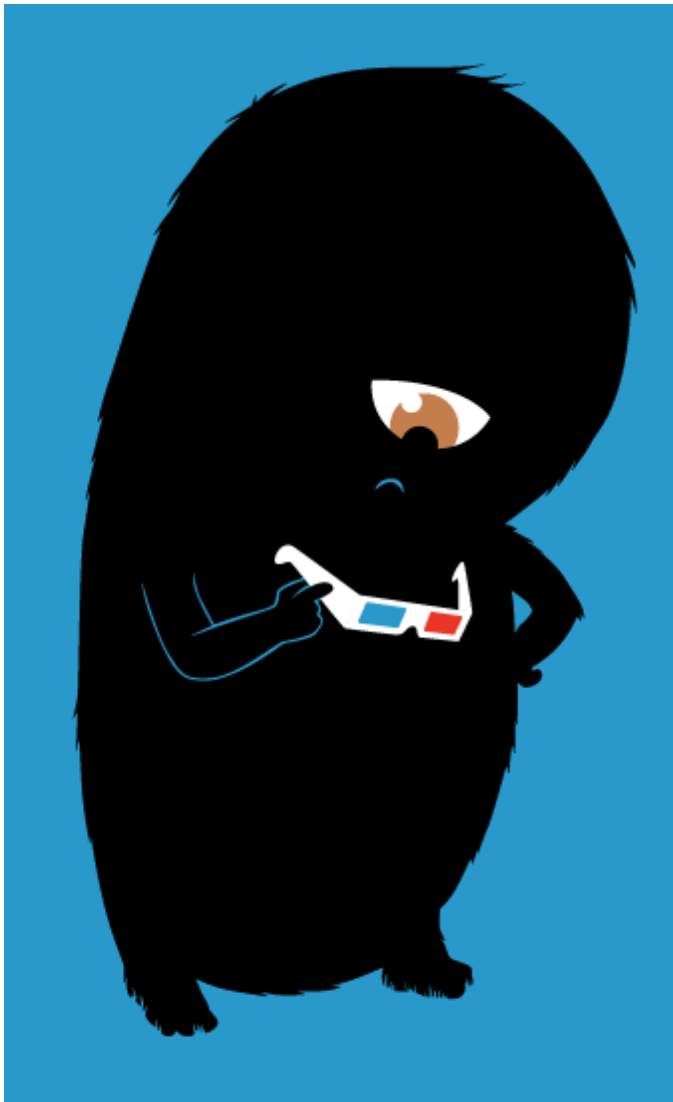


Overview of 3D technologies

- Stereoscopic (2x)
  - Sweet point, viewer dependent view, tracking for motion parallax
- Multiview (5-9x and up)
  - Jumpy image, limited are repeated FOV, small depth budget
- Integral Imaging (16-32x and up)
  - Multiview with vertical parallax, resolution issues
  - Directions at the expense of resolution - for systems based on 2D panels, except time multiplexing: multiview: 1/n; integral: $1/n^2$
- Volumetric (20 slices and up / time sequential)
  - The addressed slices/planes/spatial positions enabled by the projection technology, moving components
  - Multi-layer solutions: depth resolution discreet
- Light-field (64x and up for continuous parallax)
  - Horizontal parallax now, possible to do full parallax
- Holographic displays (~10^6x and up)
  - Pure holographic solutions: laboratory models, small image size, slow updates, viewing angle limited

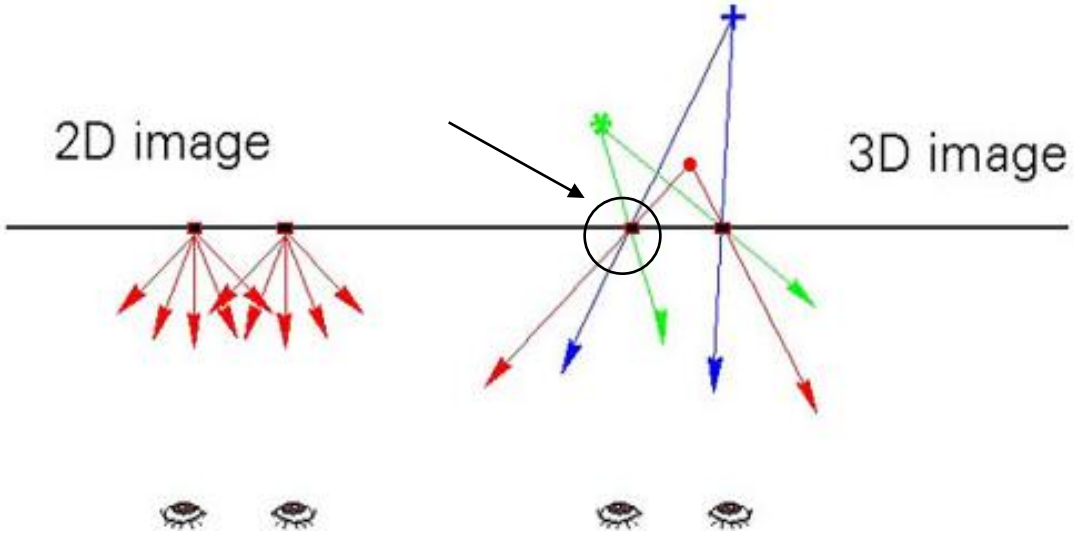
Autostereoscopic displays

- Autostereoscopic displays provide 3D perception without the need for wearing special glasses or other head-gear, as separation of left / right image is implemented using optical or lens raster techniques inside or directly above the screen surface
  - Also known as
    - Glasses-free 3D
    - Naked Eye 3D
    - Natural 3D
    - True 3D



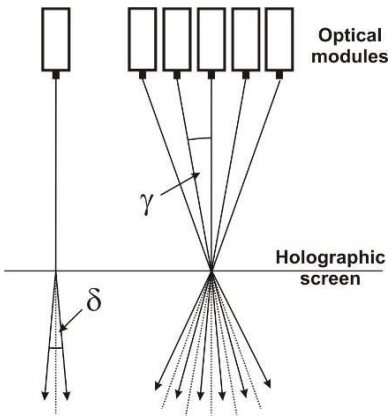
Direction selective light emission

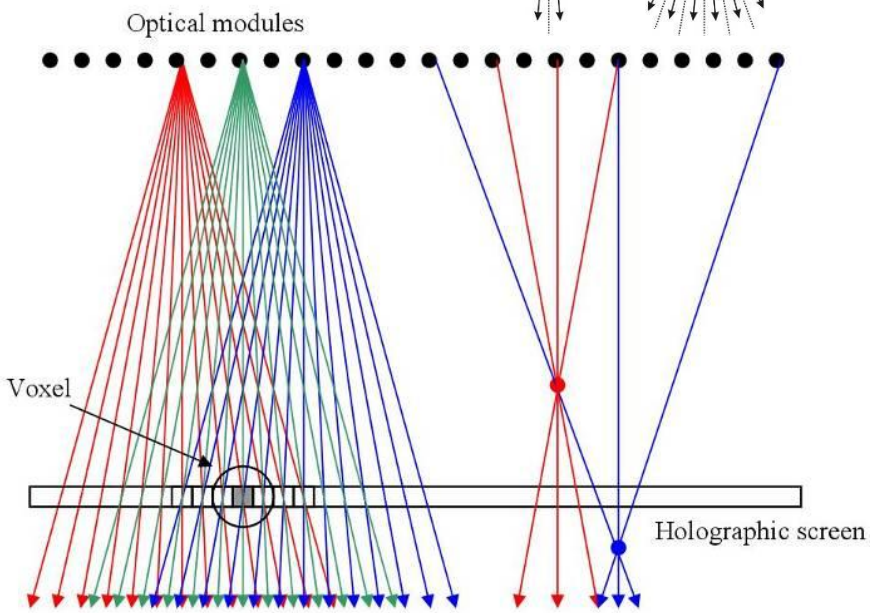
- Direction selective light emission
  - Common for all systems having a screen, conforming with current displaying conventions (also true for the outer surface of volumetric systems)
  - Not true for glasses-based, as separation is achieved in a different way
- General approach: to create a light emitting surface, where we are able to emit different light beams from each point in a controlled way



Light-field displays

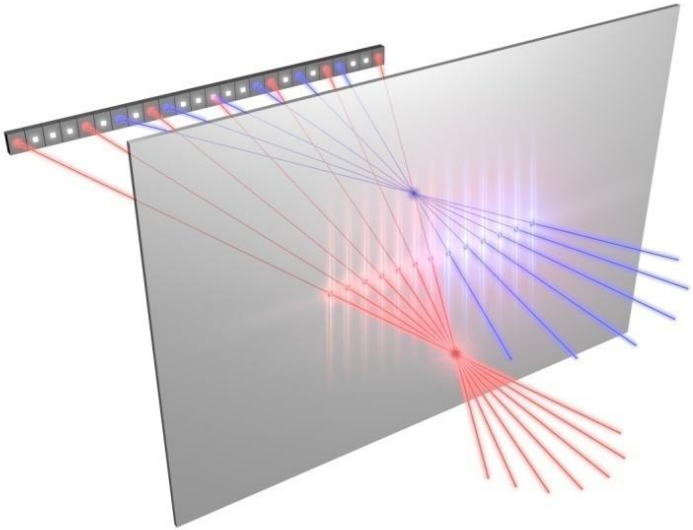
- Projection-based Light-field displays
- Projection modules
  - Project light beams
- Light beams
  - Hit the screen under various directions
- Holographic screen
  - Lets light rays pass through in their original direction
  - Without mixing colors
  - Direction selective
- “Pixels” may have different color when observed from different directions
- Points appear where light rays cross
- In front of the screen, or behind the screen

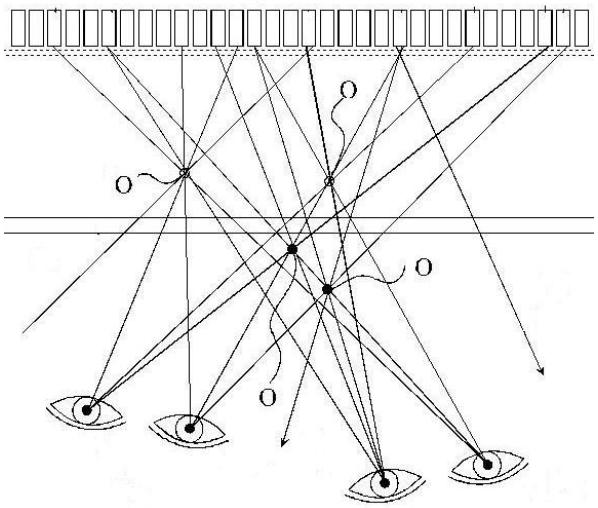




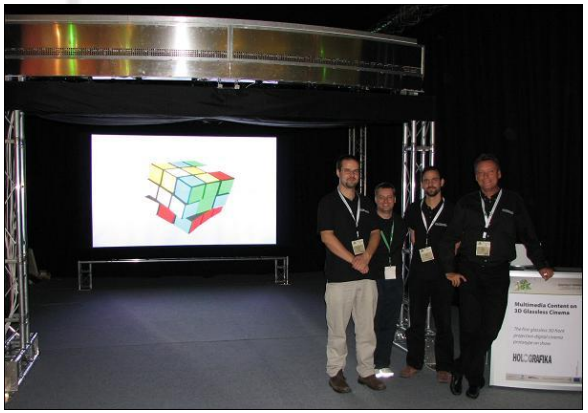
Light-field displays

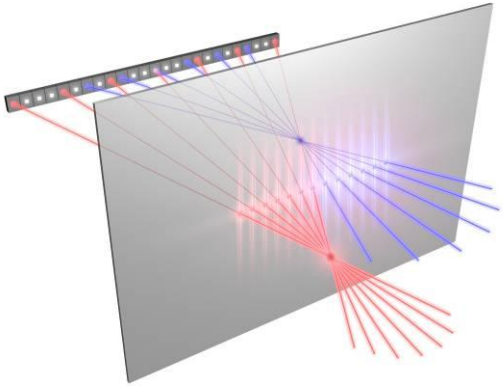
- Benefits
  - Multi-user
  - All angles valid inside in FOV
  - All viewing distances are optimal viewing distances
  - Stereoscopic vision everywhere
- Emission angle geometry determined
  - The projected module image not a 2D view of the final 3D image
  - Each view of the 3D image comes from multiple modules
  - Continuous motion parallax - no discrete border between views

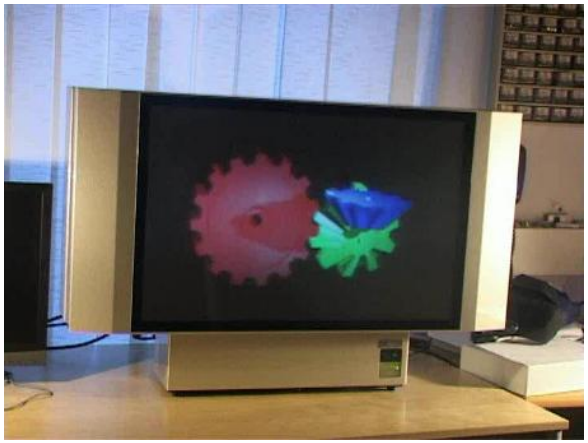


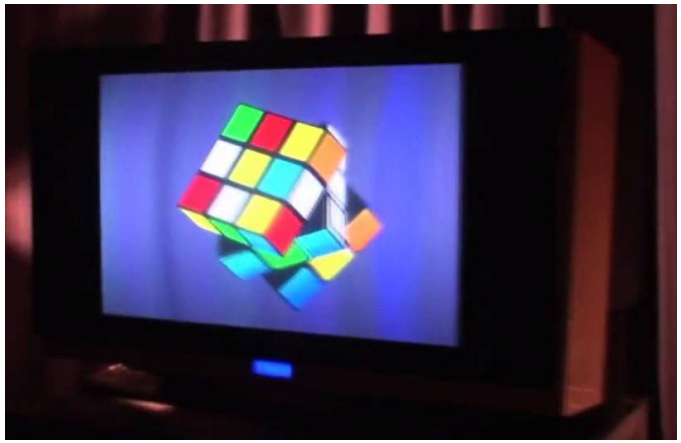


HoloVizio Light-field displays







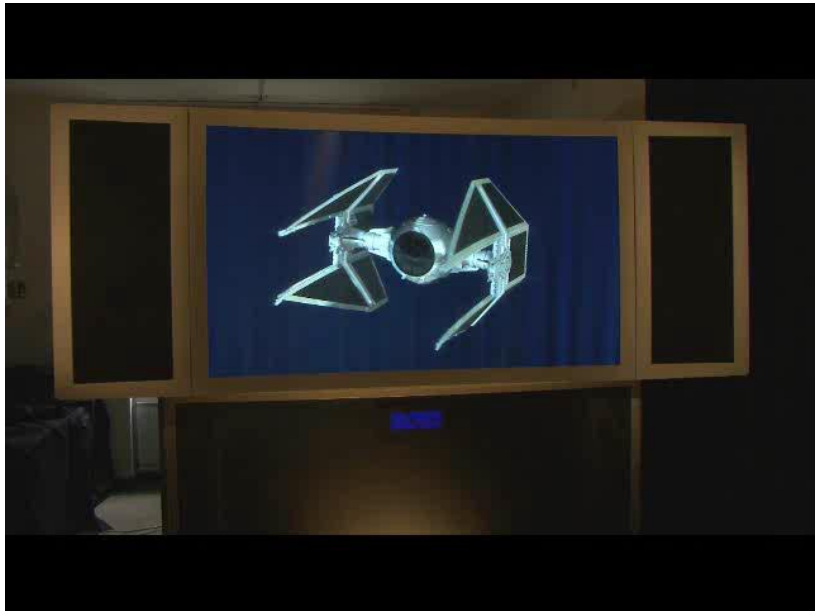


HoloVizio displays

- The large-scale HoloVizio system
  - **HoloVizio 721 RC**
    - Earlier: 640RC, 720RC

    73 Mpixel, 72”, (16:9)

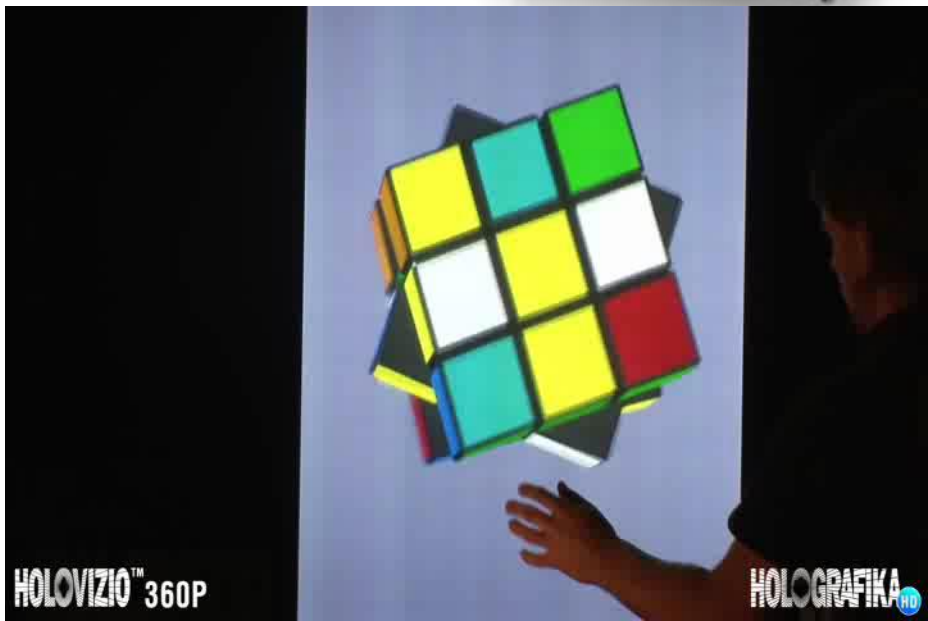
  - 70 degrees FOV
  - Angular resolution 0.9 degrees
  - 2D equivalent image resolution 1280x768
  - LED colors
  - Gigabit Ethernet input
  - Control system+PC based render cluster



HoloVizio displays

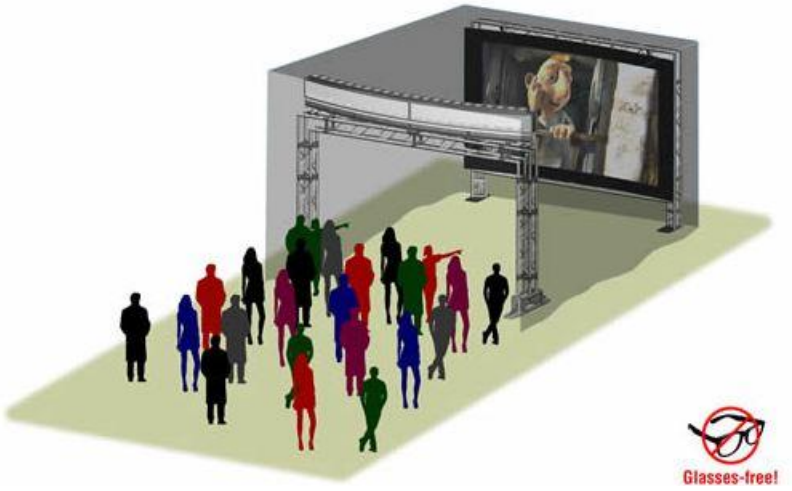
- The digital signage HoloVizio kiosk system
  - **HoloVizio 361H**
    - Earlier: 240P, 360P

    33 Mpixel, 52”
  - 50 degrees FOV
  - 2D equivalent image resolution 1280 x 720
  - LED colors
  - Gigabit Ethernet input
  - Control system + 3 PC
  - Only 0.5 m deep!



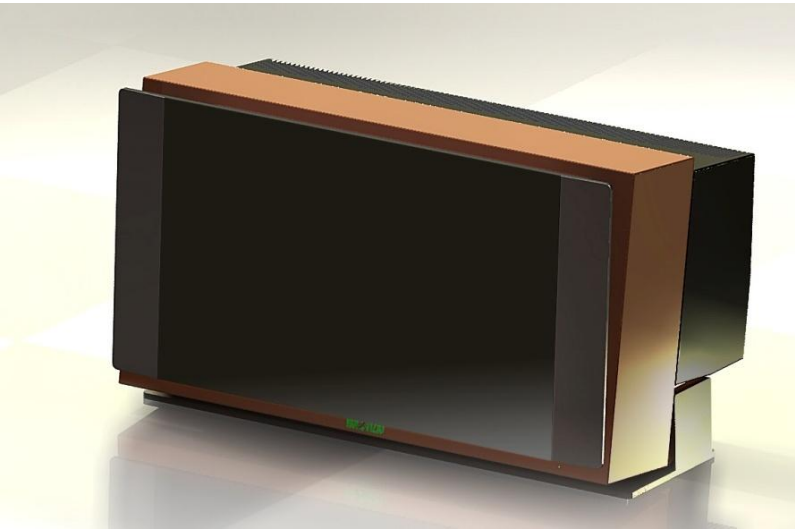
HoloVizio displays

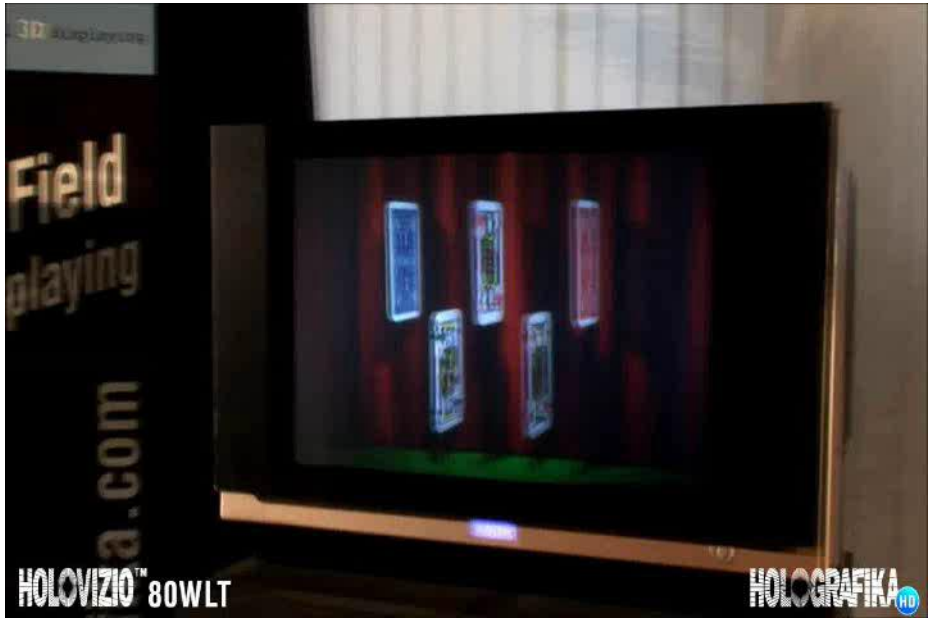
- The world first glasses-free 3D cinema system
  - **HoloVizio C80**
  - 3,5 m reflective holoscreen (140”)
  - 63 Mpixels
  - LED based 3D projection unit
  - Exceptional 1500 Cd/m^2 brightness
  - 40 degrees FOV
  - PC based render cluster
  - No perspective distortion
  - 2D compatible
  - Fitting cinema rooms, 3D simulators
  - Also with portable mechanics for event rental



HoloVizio displays

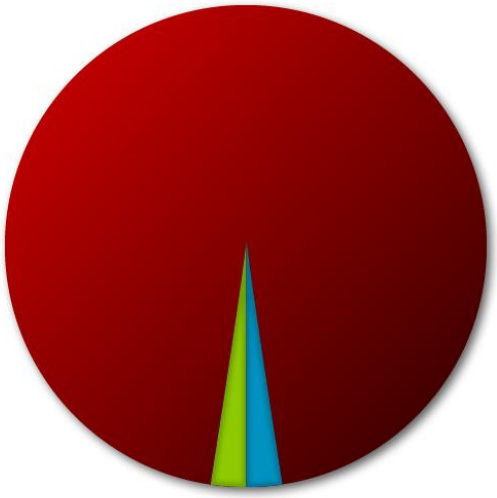
- The full-angle HoloVizio monitor
  - **HoloVizio 80WLT**
    - Earlier: 96ND, 128WD, 128WLD
  - 78 Mpixel, 30” (16:10)
  - ~180 degrees FOV
  - Total freedom in 3D experinece
  - No invalid zones
  - No repeated views
  - 2D equivalent image resolution 1280 x 768 (WXGA)
  - LED colors
  - Multiple DVI inputs





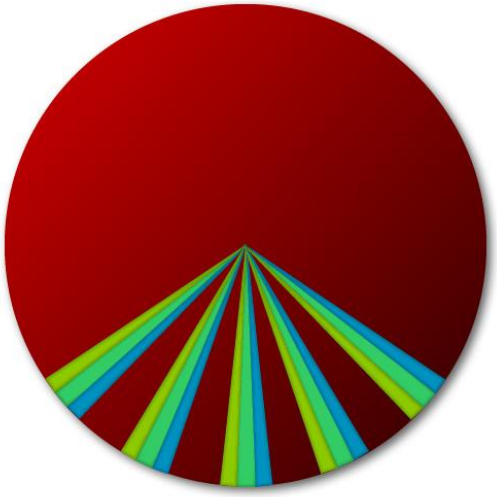
Viewing angles of 3D display technologies

2 view
Stereoscopic

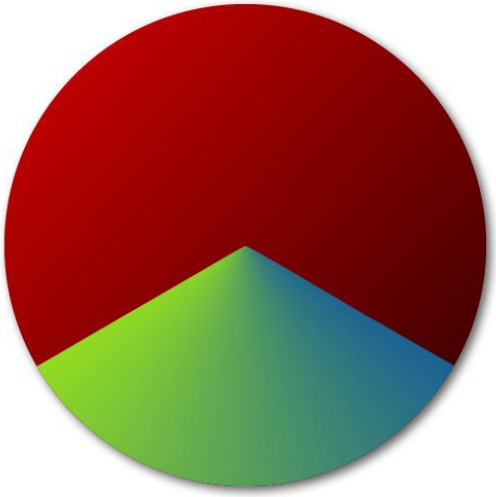


6-8°

Multiview



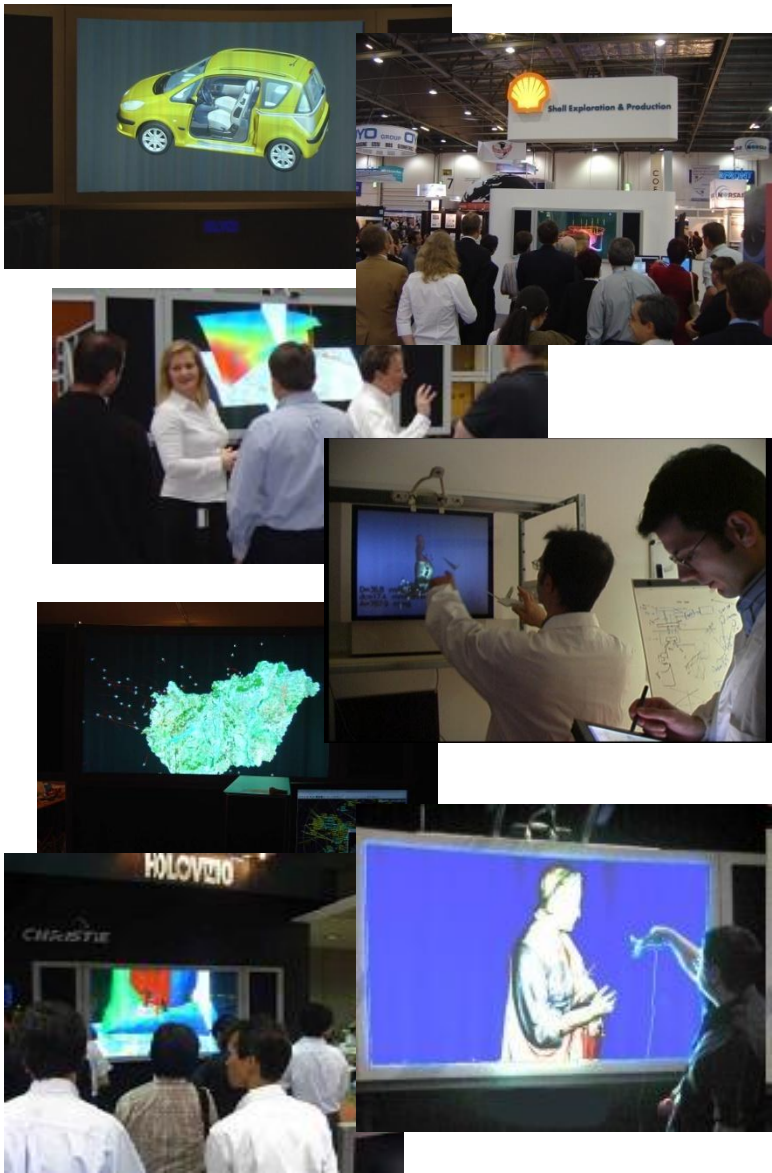
HoloVizio



70-180°

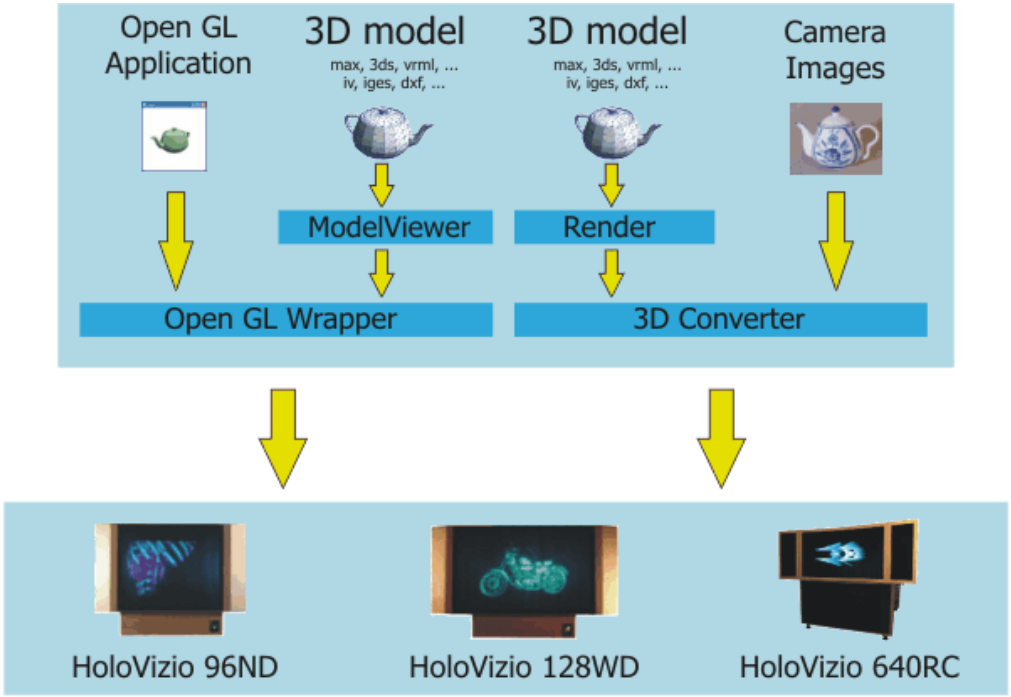
Applications

- Oil&gas – exploration, geological data visualization
- Medical – surgical training
- Museums, Edutainment, Entertainment, Theme parks
- Military, Security, Simulation, Air traffic control
- Event rental
- Scientific visualization, VR
- CAD – automotive, molecular
- 3D Telepresence
- 3DTV



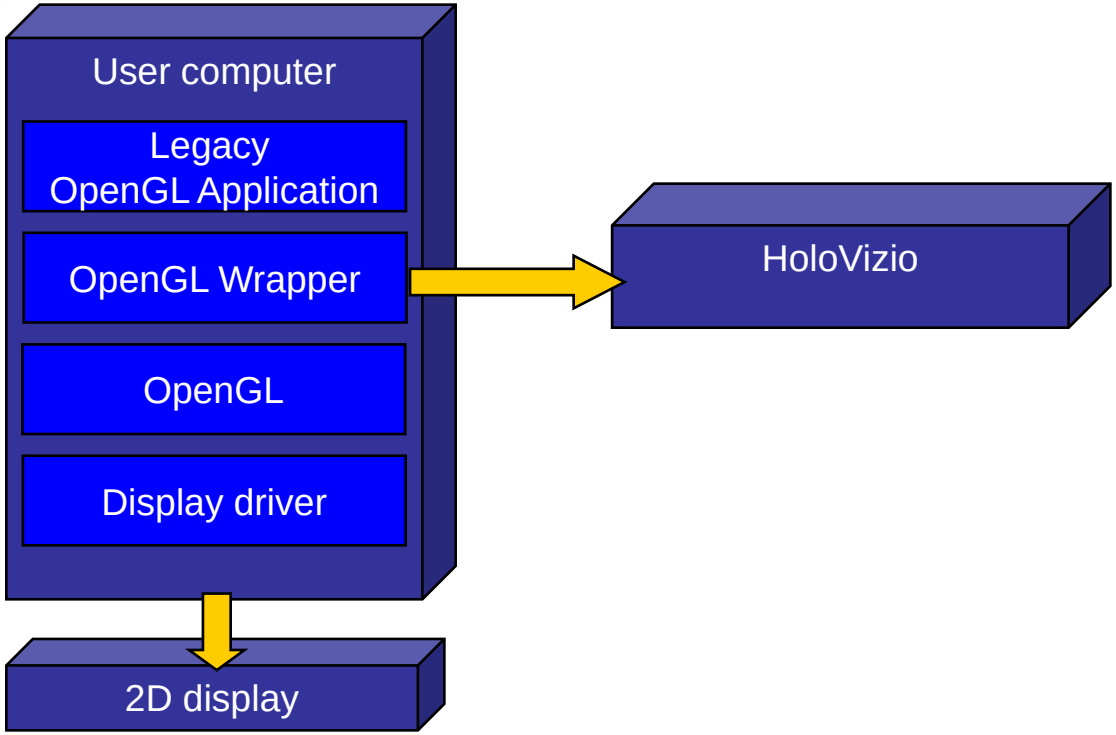
HoloVizio software system

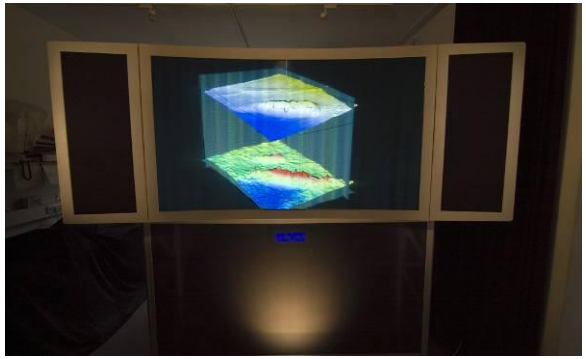
- Integrate existing applications / display 3D models in real-time
  - Interactive applications
- Display pre-rendered 3D models
  - High quality images
- Display real-life 3D scenes
  - Light field streaming

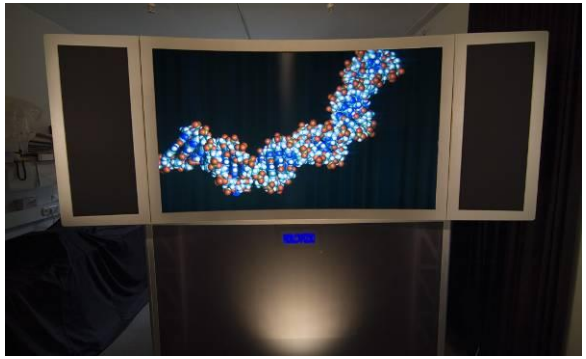


Integrate Existing Apps

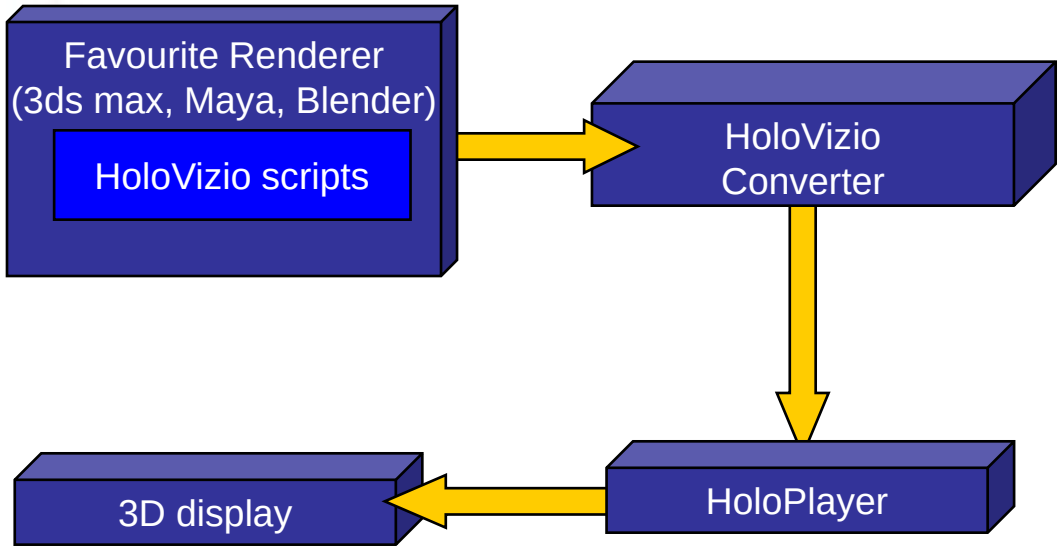
- OpenGL wrapper for HoloVizio display systems







Display Pre-Rendered 3D



Live 3D transmission

- Linear camera array prototype
  - 27 cameras (more is better)
  - 640x480 at 15 FPS
  - 960x720 at 10 FPS
- Camera calibration
  - Mapping of camera pixels to light rays
- Transmission
  - Camera streams
- Rendering
  - Ray interpolation
  - Captured rays to displayed rays
- Display
  - Any HoloVizio display

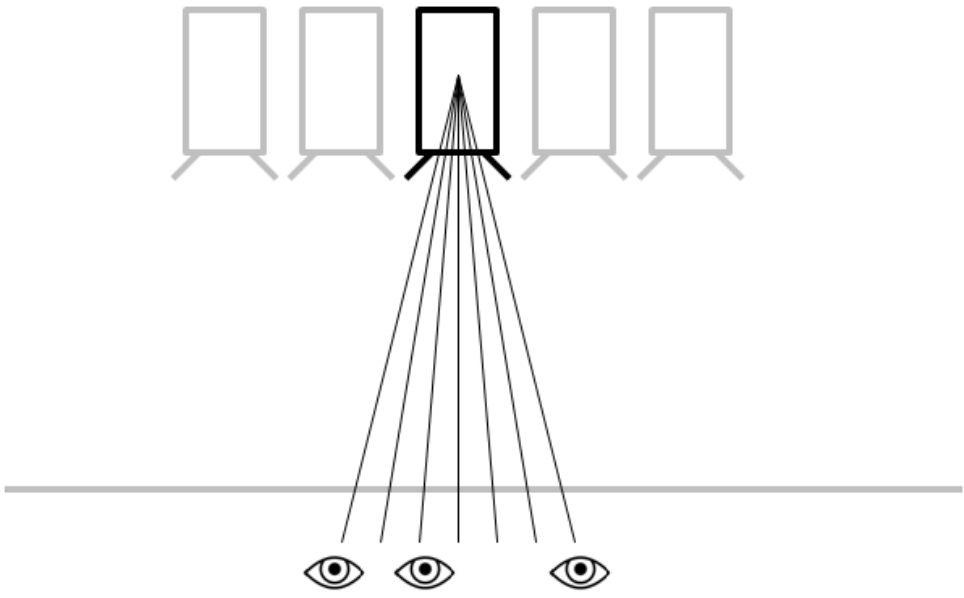
Image-based LF applications

- Several image-based (i.e. non geometry-based) applications
  - 3D video playback
  - Light-field 3DTV
  - 3D Telepresence
- These require
  - Compression and decompression of 3D LF videos
  - Up to 100 Mpixels (today, more to come)
  - In real-time
  - Using reasonable hardware resources
- How can these be supported by compression?
  - Checked MVC - finalized long enough to have real-time implementations(?)

Kovács, P. T., Nagy, Zs., Barsi, A., Adhikarla, V. K., Bregović, R., "**Overview of the Applicability of H.264/MVC for Real-Time Light-Field Applications**", at **3DTV-Con 2014**

Light-field displays

- The projected image not a 2D *view* of the 3D scene
- Light rays emitted by a single projection module are spread over screen positions and viewing directions, thus cannot be seen from a single viewing position
- A projection module shows a *LF slice*



Distributed rendering

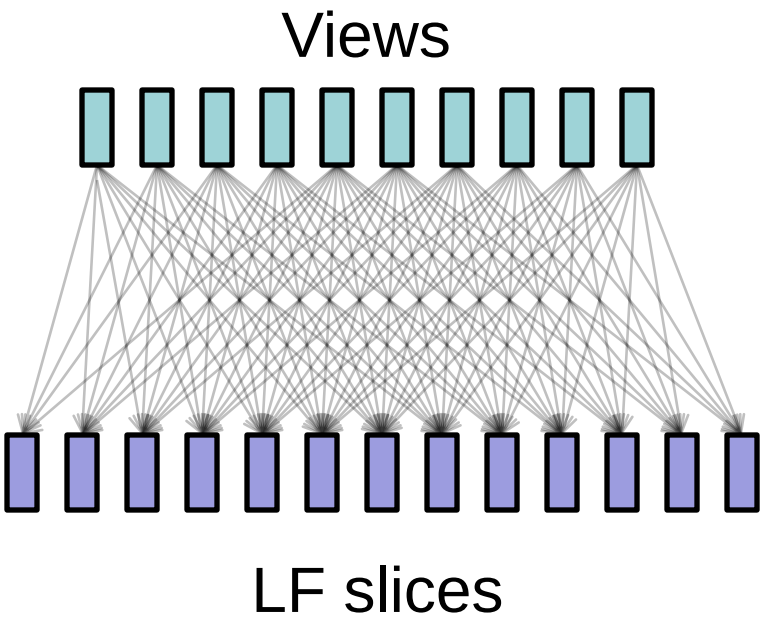
- Distributed LF generation
  - High number of pixels / light rays
  - Many inputs to feed directly
- Distributed optical system – distributed rendering
- GPU enabled rendering clusters provide scalability
  - GPU outputs correspond to projection modules
  - One output → one LF slice

    OR
  - One output → adjacent LF slices
  - One rendering node → adjacent LF slices

LF conversion

- Input: many 2D views
- Output: LF slices in display-specific arrangement
- One LF slice is composed of pixels from many views

  and backwards
- One 2D view contributes to many LF slices

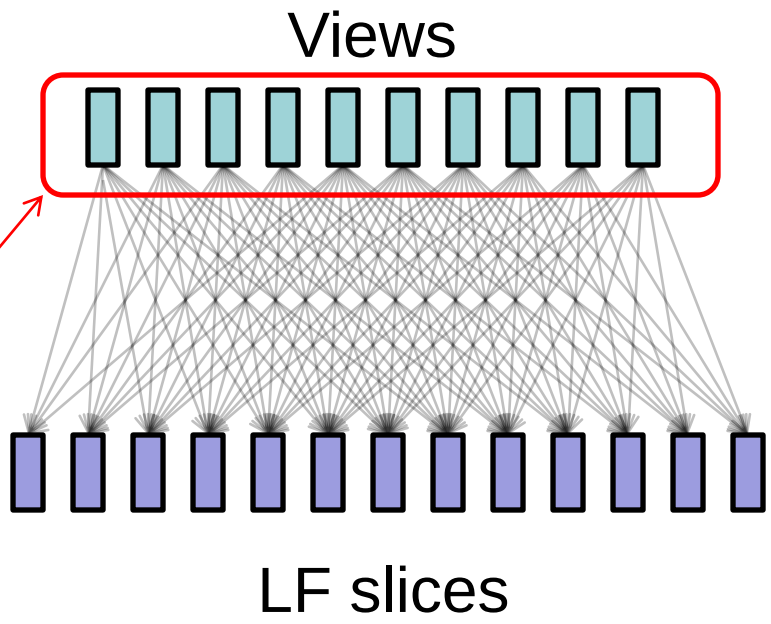


LF conversion

- Input: many 2D views
- Output: LF slices in display-specific arrangement
- One LF slice is composed of pixels from many views

  and backwards
- One 2D view contributes to many LF slices

- Display independent LF

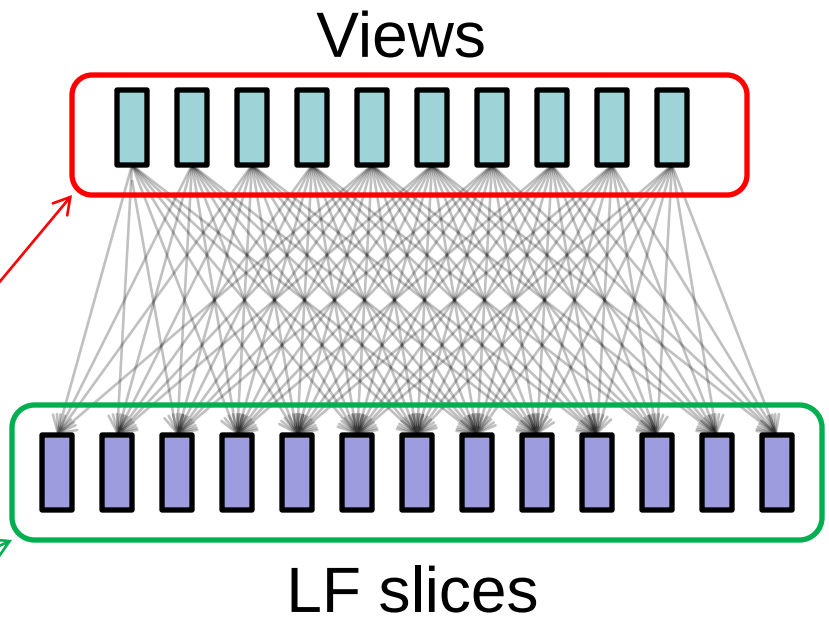


LF conversion

- Input: many 2D views
- Output: LF slices in display-specific arrangement
- One LF slice is composed of pixels from many views

  and backwards
- One 2D view contributes to many LF slices

- Display independent LF
- Display specific LF



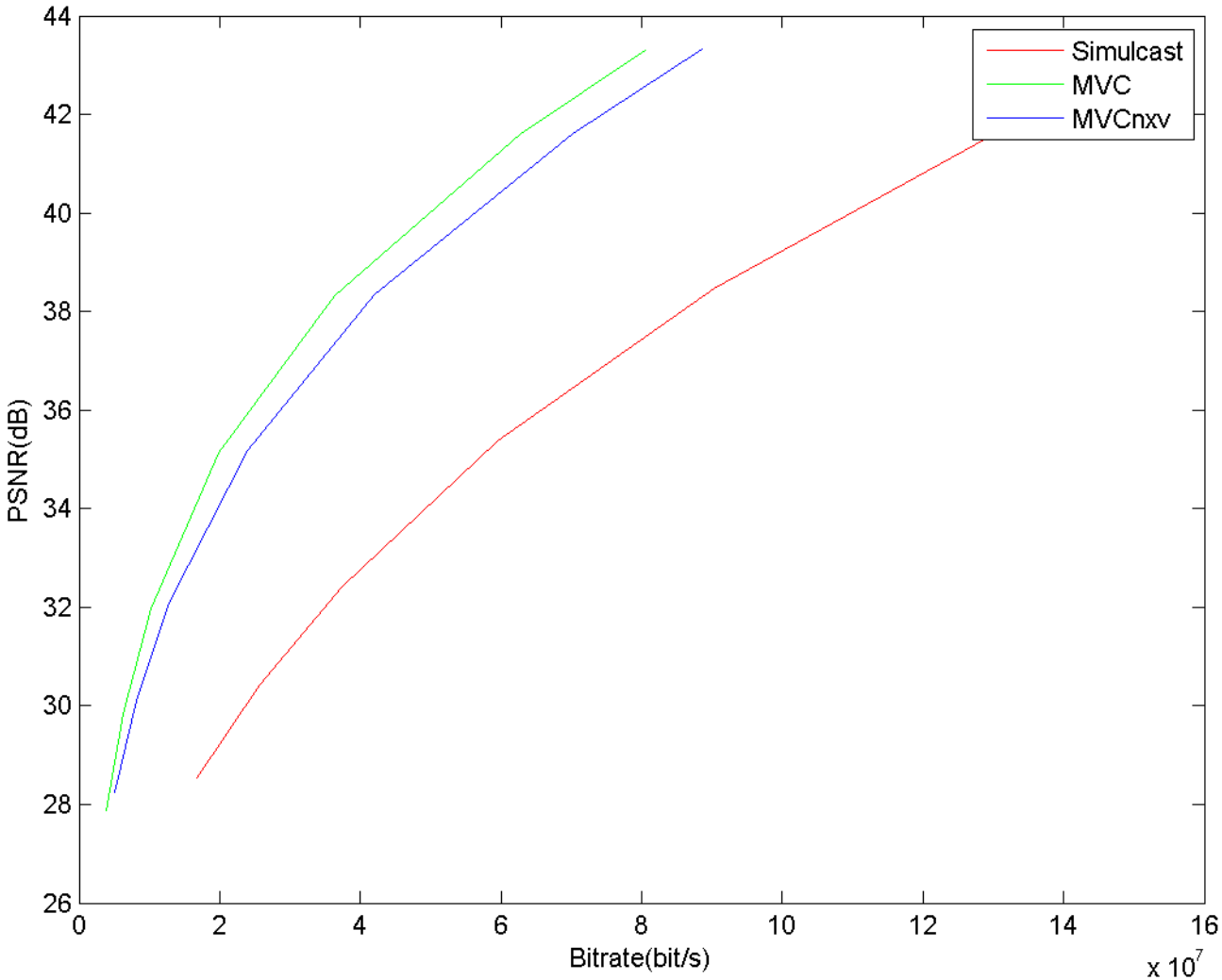
Use case requirements

- Playback of pre-processed LF content
  - Stored in display-specific LF
  - LF slices correspond to the layout of the distributed rendering and projection system
  - Example: 80 x WXGA LF slices
  - Visualized directly on the targeted display
- Broadcast LF video transmission
  - Target different displays
  - Different Field Of View
  - Stored in display independent LF (views)
  - Needs to be converted to display-specific LF at the end user in real-time

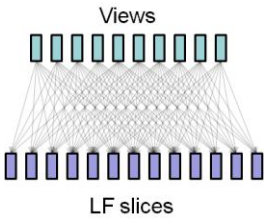
Processing display-specific LF

- Simplest solution is to simulcode LF slices (e.g. 80x1)
  - Decode them in parallel on rendering nodes independently
  - Decoding e.g. 8 times H.264 is possible on a multi-core CPU
  - This approach works today
  - Does not exploit similarities between LF slices
- Compress all LF slices into a single MVC stream (e.g. 1x80)
  - Single processing node should decompress the whole stream
  - Larger Decoded Picture Buffer is needed → more memory
  - Decoding complexity increases with number of views
  - Decompression in real-time is too resource intensive
- Compress N adjacent LF slices (e.g. 10x8)
  - Reasonable tradeoff that could map well to the distributed rendering cluster if there was a real-time implementation

Processing display-specific LF



Processing display-independent LF



- Processing node needs to
  - Decompress views
  - Perform LF conversion
- Not all views are required by a single node
  - But many
- Number of views used depends on display's FOV
  - How many views we need to decompress for the whole display?
  - Simulated LF displays with a range of FOVs

FOV (degrees)	27	38	48	59	69	79	89
No. views used	42	44	46	48	50	52	54

  - Decompressing this many views is still prohibitive
  - How to reduce decoding workload?
  - Not all pixels are used from the views

View

Nodes 4,5,6

Nodes 0,5,9

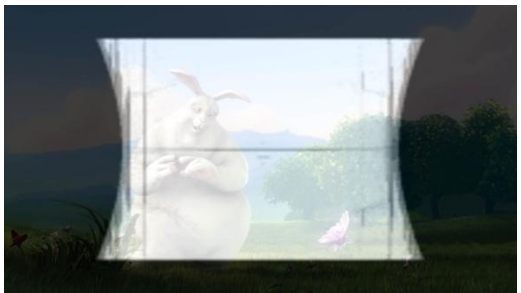
Processing display-independent LF

- Not all pixels are used from the views
- Why decode pixels that are left unused?
  - Why not just transmit parts that are actually used?
- Used area increases with FOV
  - Still, large portions could be skipped
  - To support different FOVs, in-view scalability is needed
- Slicing the image vertically could do it
  - With a specific granularity
  - By losing some coding gain
  - If the encoder / decoder supports this kind of slicing (or rotate the images)
  - If the decoder supports skipping slices

27º FOV

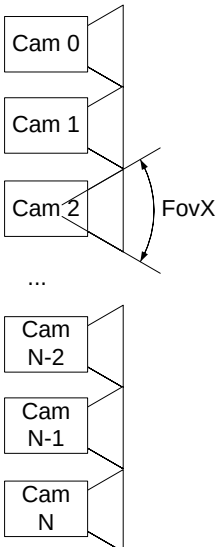
59º FOV

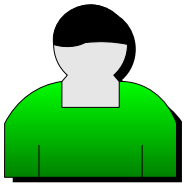
89º FOV

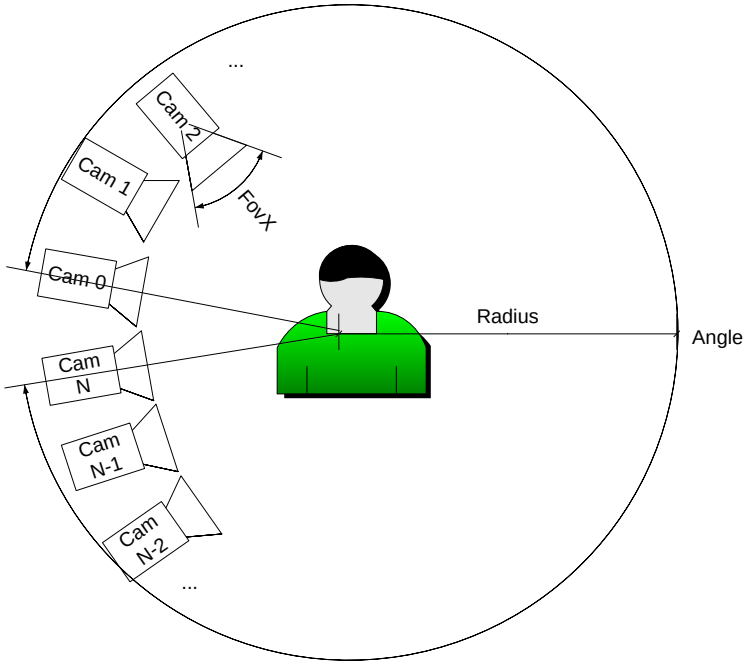


Nonlinear camera setups

- A linear setup cannot capture the scene from all around
  - Think of the 180º FOV 3D display (80WLT)
- MVC is designed for parallel linear camera setups
  - Inter-view similarity is less with arc setups
- With wide-angle LF displays, compressing views from arc rigs is important







MVC implementations

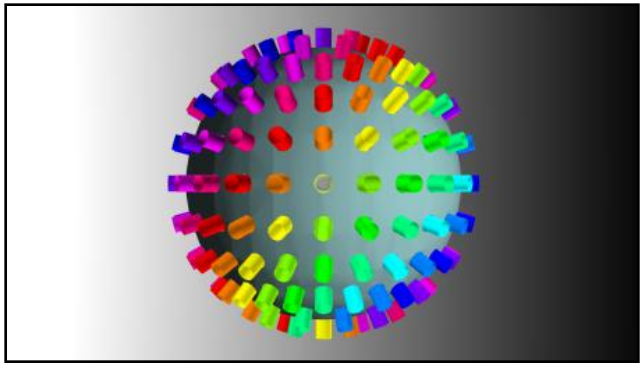
	MVC	Real-time	Arbitrary slicing
JM	2 views	X	√
JMVC	N views	X	√
x264	X	√	X
NVENC	2 views	√	X
MainConcept MVC/3D	Encoding 2, decoding 10	√	?
IP cores / HW	2 views	√	?

Conclusions, ongoing and future work

- Displays
  - 3D cinema with N*100 directions
  - Higher resolution sources (towards gigapixel displays)
- Video compression
  - Desired functionality
    - Encode views of different resolution (parts of views)
    - Decode views partially, starting from the scene center
    - Nonlinear (arc) multi-view coding tools
  - Add MVC functionality to x264
    - To be able to exploit MVC in real-time
    - NVENC rather goes towards HEVC
  - Develop in-between representation with better localization of rays
    - Something between views and LF slices?
  - Investigate 3D HEVC
    - Even more unlikely to have real-time implementations (looking at the future)
    - Experiments and subjective tests with Orange Labs are ongoing

Relation with MPEG

- Past inputs in collaboration with HHI and Huawei Germany
- P-membership request just submitted
- Plan to contribute test materials
  - Big Buck Bunny rendered from 91 views, 45º arc setup, HDR, 2 video sequences
  - Hedgehog test pattern from 180º arc



Questions?

p.kovacs@holografika.com
www.holografika.com

Videos in this presentation

- Some of the videos embedded in this presentation can be seen on YouTube:
  - https://www.youtube.com/watch?v=WqtFS7dcvC0
  - https://www.youtube.com/watch?v=sw_LI8J-AlU
  - https://www.youtube.com/watch?v=3oX-aOJMX34
  - https://www.youtube.com/watch?v=PlAqXNnQQtk