JÜLICH
FORSCHUNGSZENTRUM

Introduction to Computer-Based Holography

October 29, 2013 | Carsten Karbach, Jülich Supercomputing Centre (JSC)

Why computergenerated holography?

Source: `www.starwars-union.de`

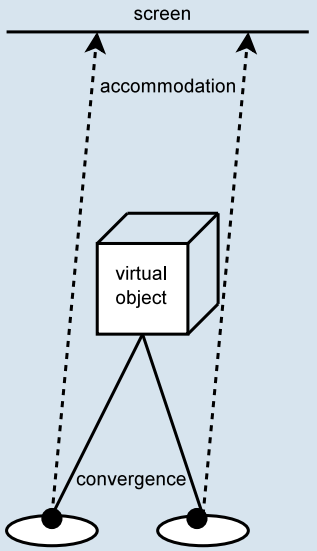
Applications

- **3D Scientific Visualization**
  - **Natural** 3D viewing, focus on different depths
  - **Correct perspective** for multiple viewers
  - No special glasses required
- Holographic optical elements

Problems with Stereoscopy

Accommodation/Convergence conflict

- **Accommodation**:
  change power of lens
  according to object distance
  - near object $\rightarrow$ thick lens
  - far object $\rightarrow$ thin lens
- **Convergence**:
  eye rotation to the object



Same perspective for all

- Optimized view for single viewer location
- All viewers receive identical perspective

Challenges for CGH

Software

- Efficient, parallelized implementation
- $O(N^2 * log(N))$ FLOP per image in real time, with $N =$ px count of one edge, example $N = 10000 \Rightarrow 10^8$
- Multiple 2D Fourier Transformations for each image

Hardware

- Display with extremely **high resolution** required (100$*$ HDTV resolution)
- Pixel distance in magnitude of light waves

Content

Part I: Holography

October 29, 2013 | Carsten Karbach, Jülich Supercomputing Centre (JSC)

JÜLICH FORSCHUNGSZENTRUM

Holography

Definition

- Record entire light properties reflected from an object
- Store **intensity** and **phase** information
- Reconstruction of light field of the recorded object

History

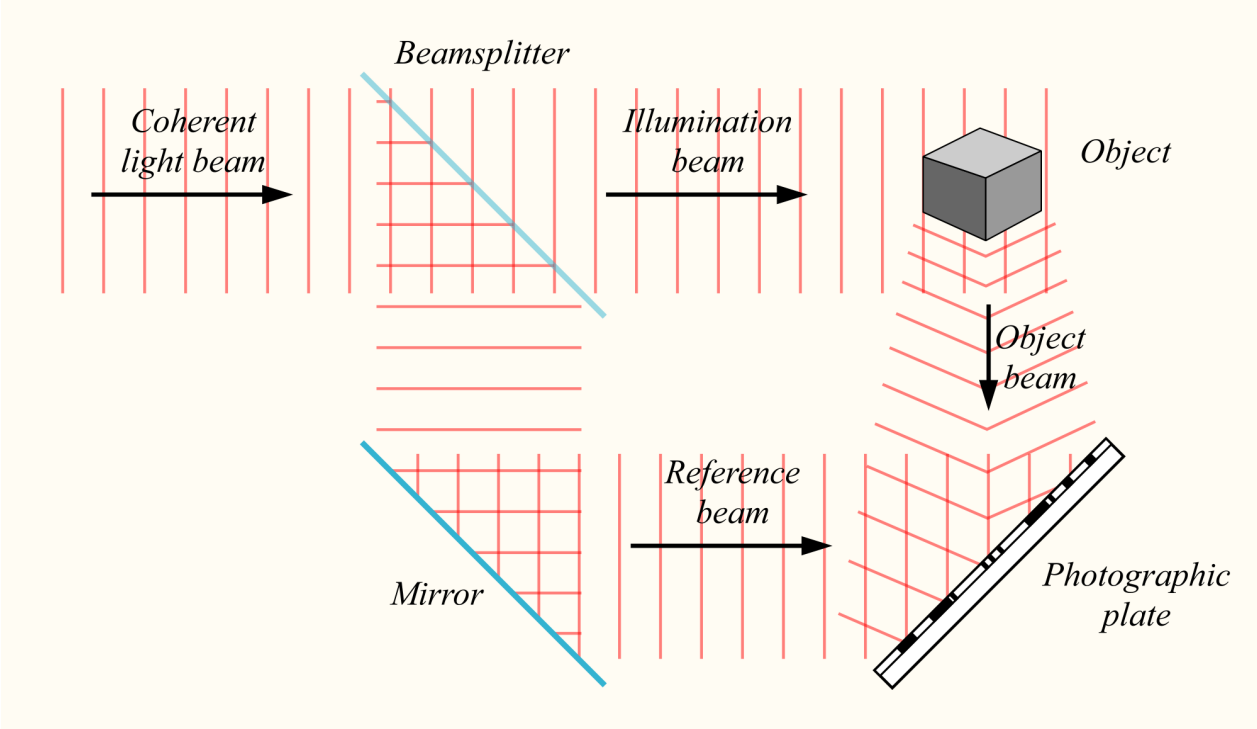
1948 Dennis Gabor generates first hologram

1959 Improved hologram quality by Leith and Upatnieks

1960 Invention of laser, requirement for holograms

1966 First implemented algorithm for computergenerated hologram

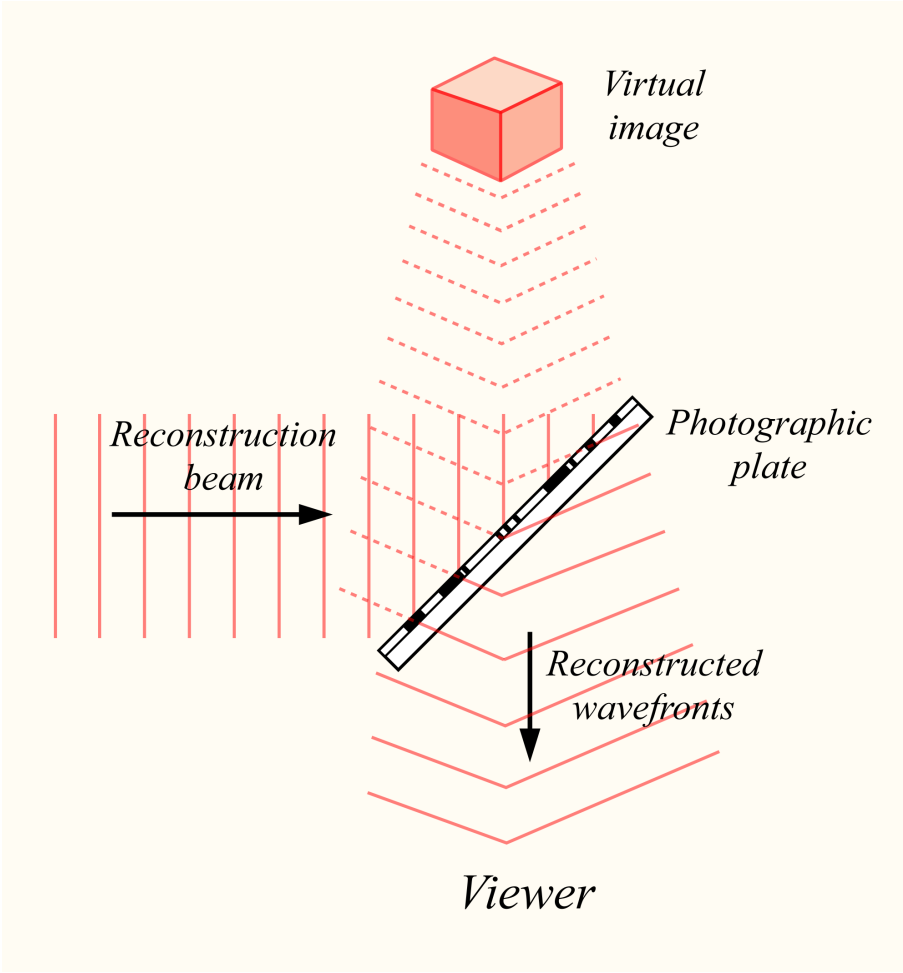
Hologram Recording



Source: `http://en.wikipedia.org/wiki/Holography`

- **Interference** of reference and object beam
- **Coherent** laser light generates standing waves
- Phase and amplitude of light are recorded

Hologram Reconstruction

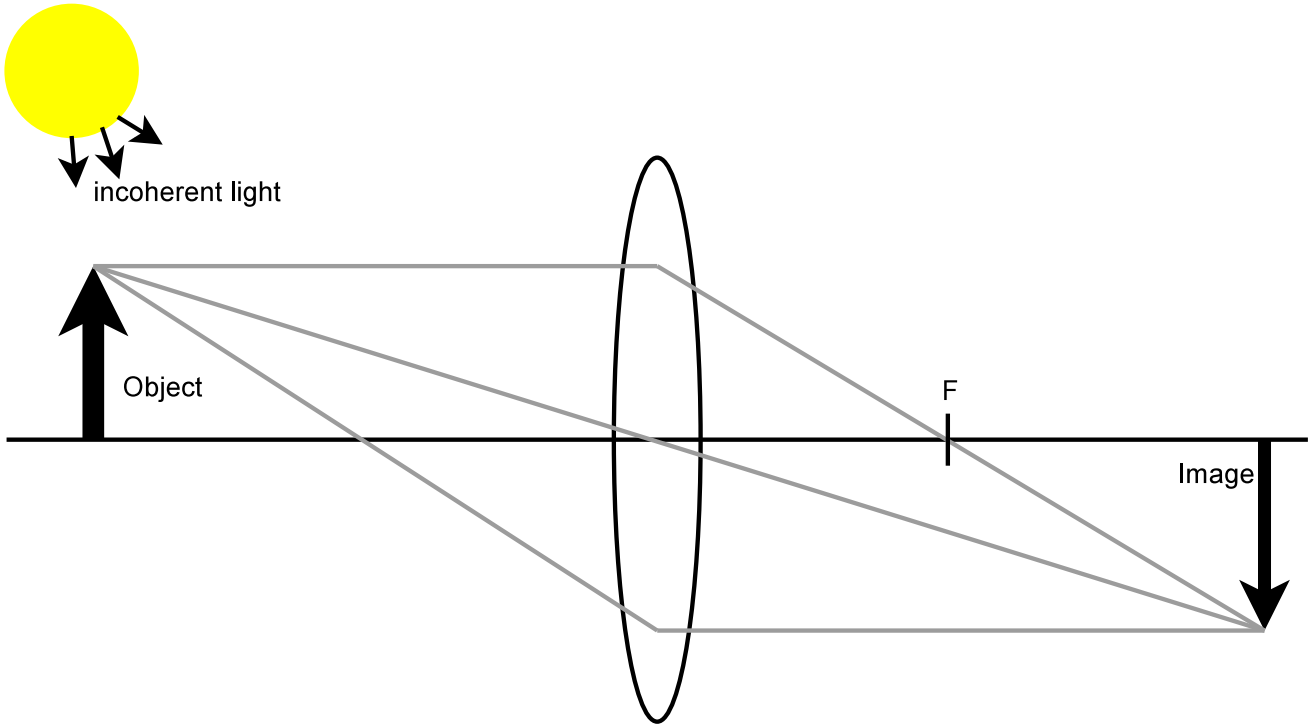


- Reconstruction beam is identical to reference beam
- **Diffraction** of reference beam by hologram
- **Virtual image** appears at position of recorded object
- **Real image** appears in front of the hologram

Source: `http://en.wikipedia.org/wiki/Holography`

JÜLICH FORSCHUNGSZENTRUM

Comparison Photography

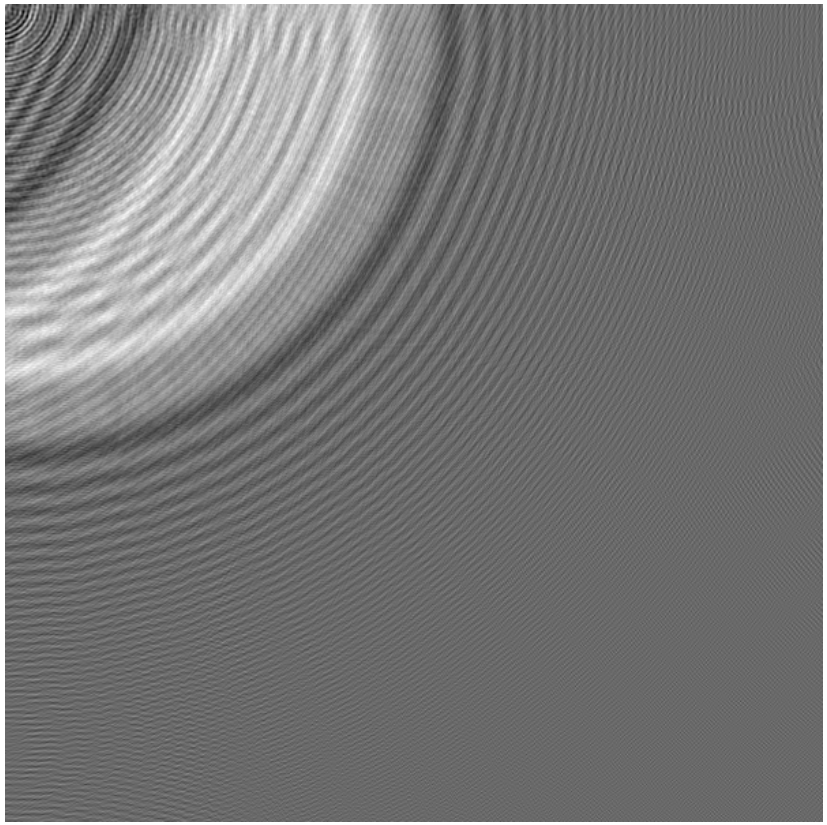


- Incoherent light source
- Lens required for focusing the object

JÜLICH
FORSCHUNGSZENTRUM

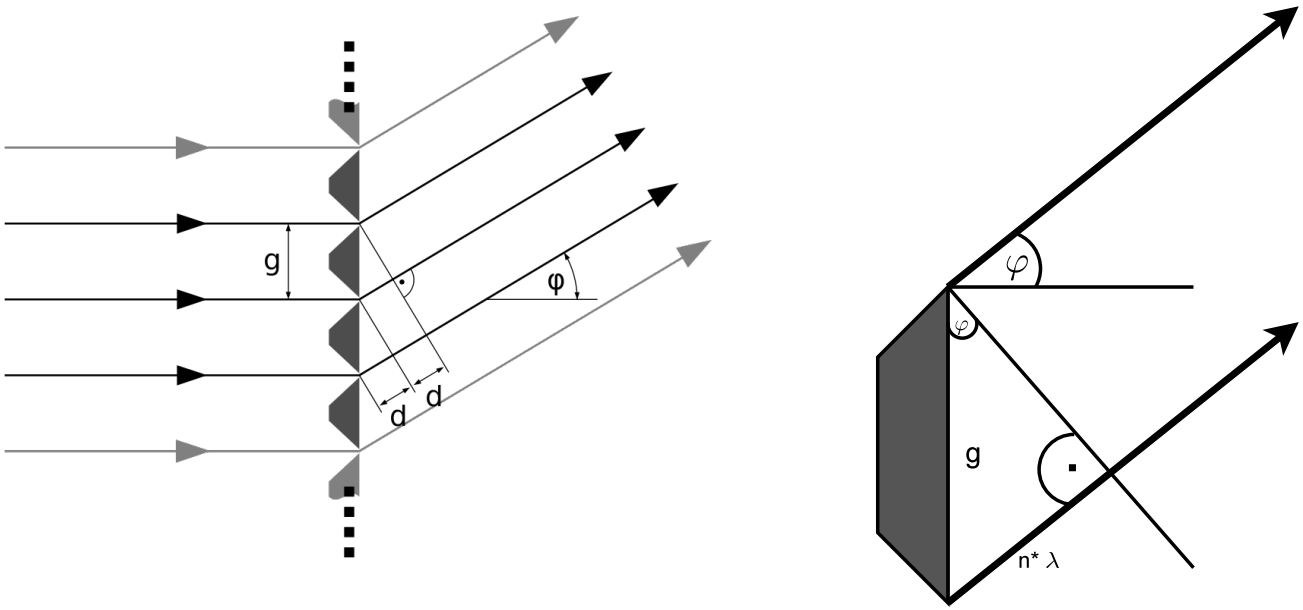
Example hologram

Recorded object



Calculated fringe pattern

Diffraction grating



Source: `http://de.wikipedia.org/wiki/Optisches_Gitter`

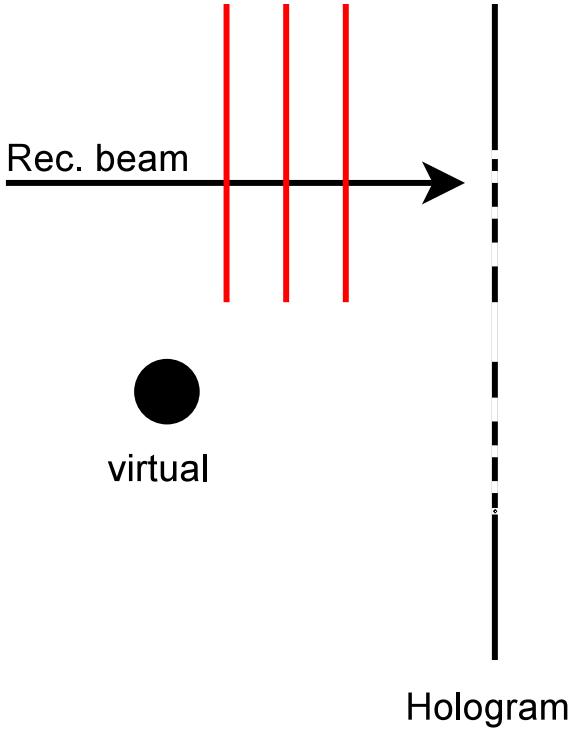
- Use of **Huygens-Fresnel** principle leads to generation of plane wave
- Direction of plane wave calculated by $|sin(\varphi_n)| = \frac{n\lambda}{g}$

  $\Rightarrow$ diffraction grating generates plane waves in directions

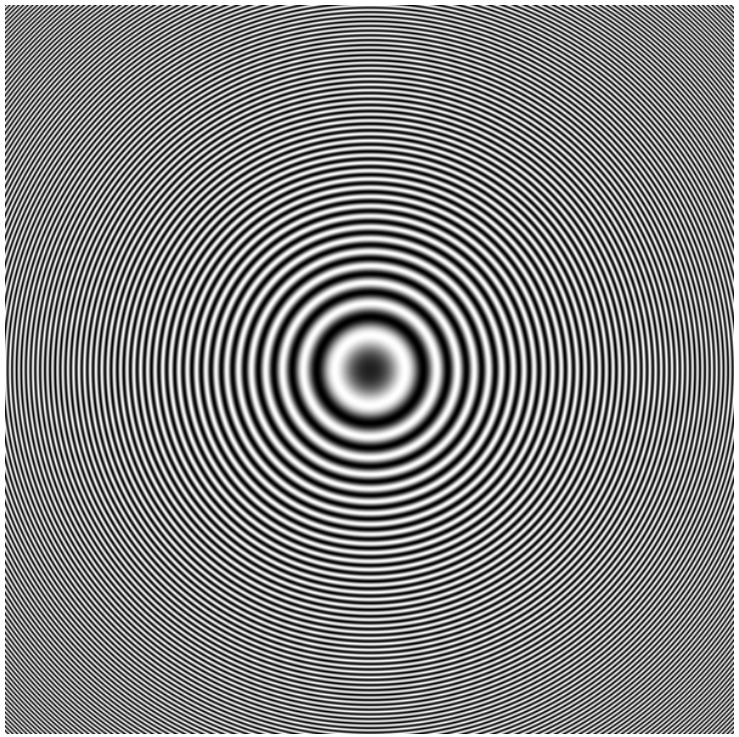
$$\varphi 1, \varphi -1, \varphi 2, \varphi -2, \cdots$$

Hologram of one point

Setup

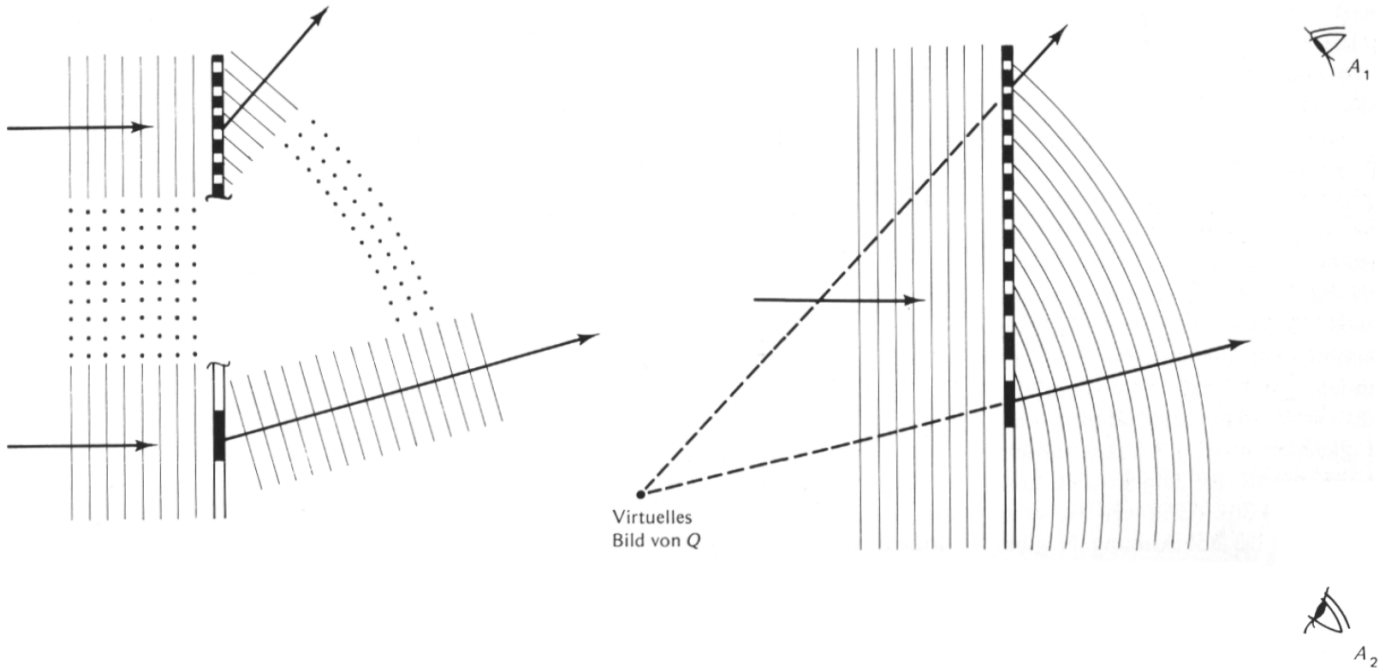


Hologram



JÜLICH FORSCHUNGSZENTRUM

Reconstruction of virtual point

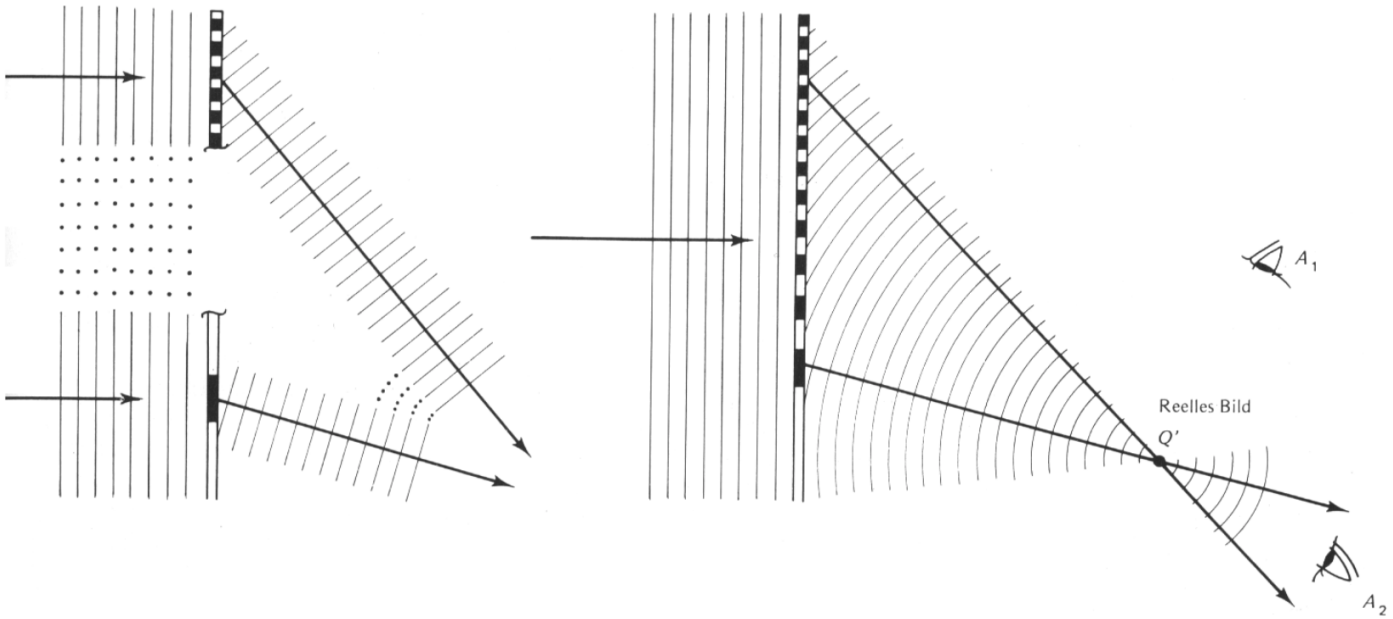


Source: `http://www.holografie.com/Fouquier.pdf`

- Assumption: Hologram is **superposition** of many diffraction gratings
- Each diffraction grating generates plane wave in given direction (here defined by φ_1)

JÜLICH FORSCHUNGSZENTRUM

Reconstruction of real point



Source: `http://www.holografie.com/Fouquier.pdf`

- Diffraction gratings also produce plane waves in φ_{-1} direction
- These plane waves form the real image

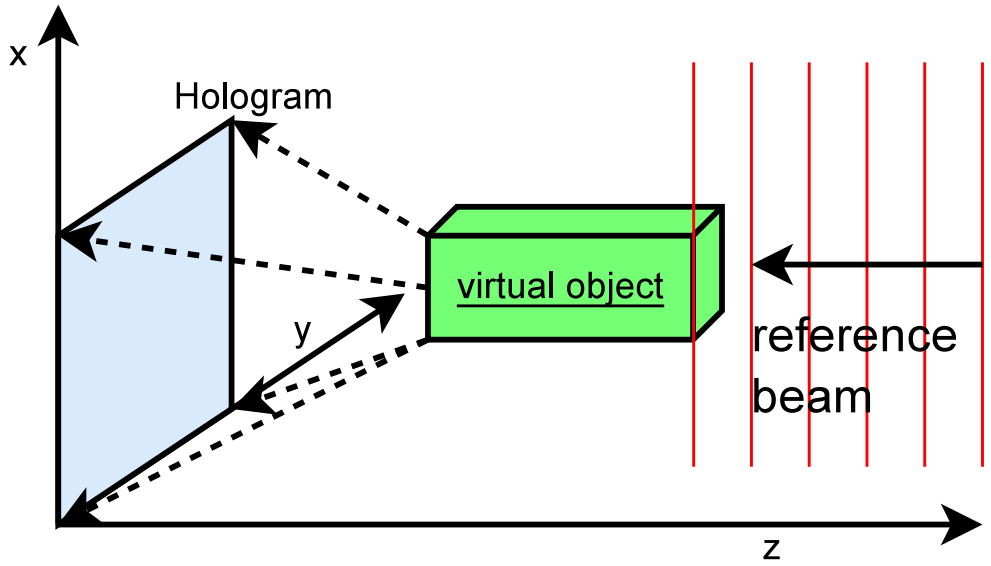
Part II: Computergenerated holograms

October 29, 2013 | Carsten Karbach, Jülich Supercomputing Centre (JSC)

JÜLICH FORSCHUNGSZENTRUM

Computergenerated holograms

- Calculate fringe pattern on hologram plate
- Create holograms of virtual objects

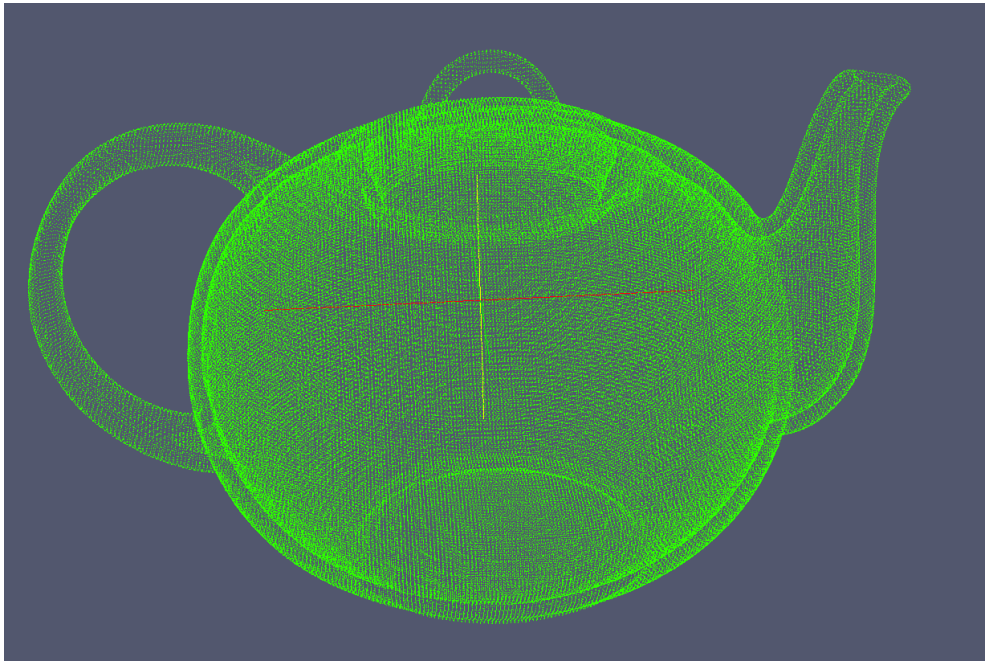


Instruction

1. Describe virtual object
2. Calculate light propagation
3. Produce hardware hologram
4. Hologram reconstruction

JÜLICH FORSCHUNGSZENTRUM

Modeling of virtual scene

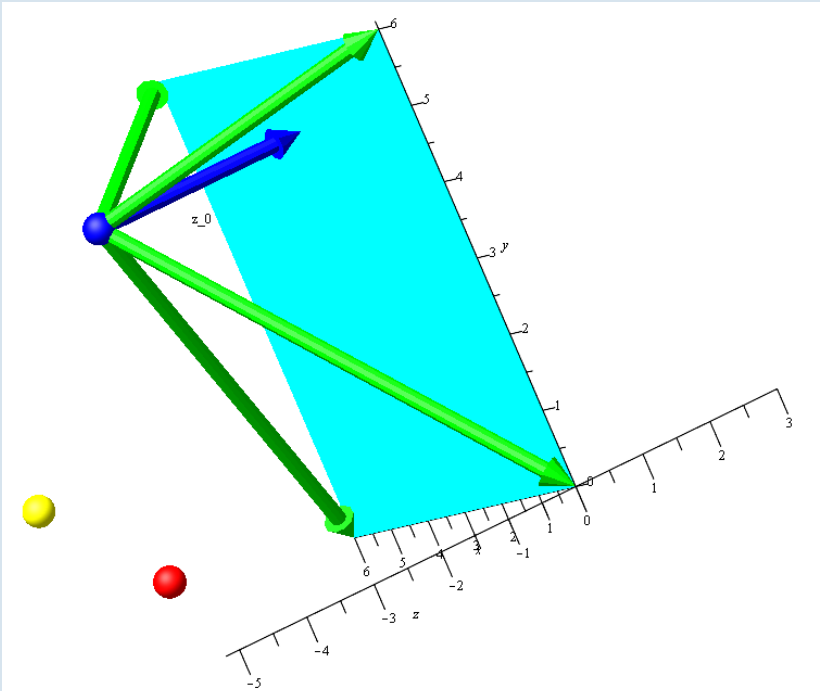


- Discretize virtual scene with points
- Use each point as source of a **sphere wave**
- Electric field of sphere wave is $A * \frac{e^{i\left(\frac{2\pi r}{\lambda}+\phi_0\right)}}{r}$ with $r =$ distance from source to hologram pixel

JÜLICH FORSCHUNGSZENTRUM

Calculate CGH

Setting



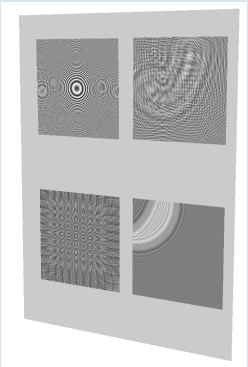
Calculation

$$
\begin{aligned}
P_l &:= \text{source of sphere wave } l \\
p_{lm} &:= \textit{pixel}(l, m) \text{ on hologram plane} \\
U_l(P) &= \frac{e^{i\frac{2\pi r}{\lambda}}}{r},\ r = |P - P_l| \\
R(P) &= \text{reference beam} = e^{i\frac{2\pi\langle\vec{k}, P - R_0\rangle}{\lambda}} \\
H(p_{lm}) &= \text{electric field on hologram plane} \\
&= R(p_{lm}) + \sum_{i=1}^{n} U_l(p_{lm})
\end{aligned}
$$

JÜLICH FORSCHUNGSZENTRUM

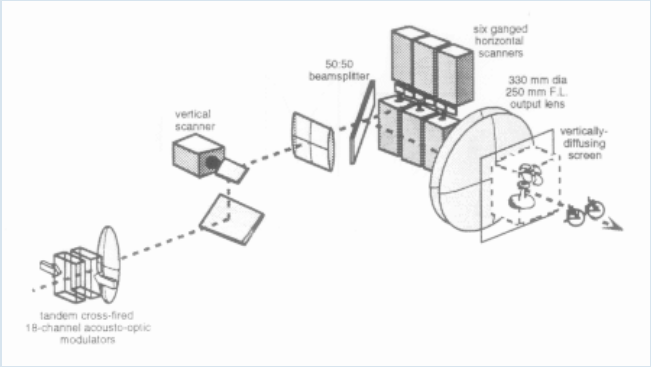
Reconstruction of CGH

Transparency film

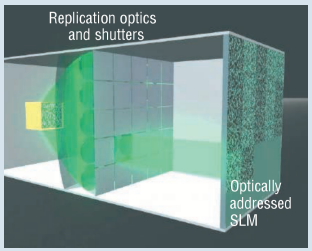


Print CGH with laser printer

Holovideo system

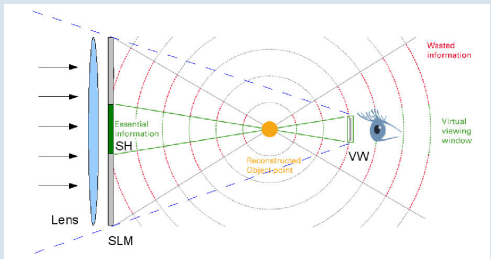


Active Tiling



Modular SLM* with 10^8 pixels

SeeReal



Subhologram

* Spatial light modulator

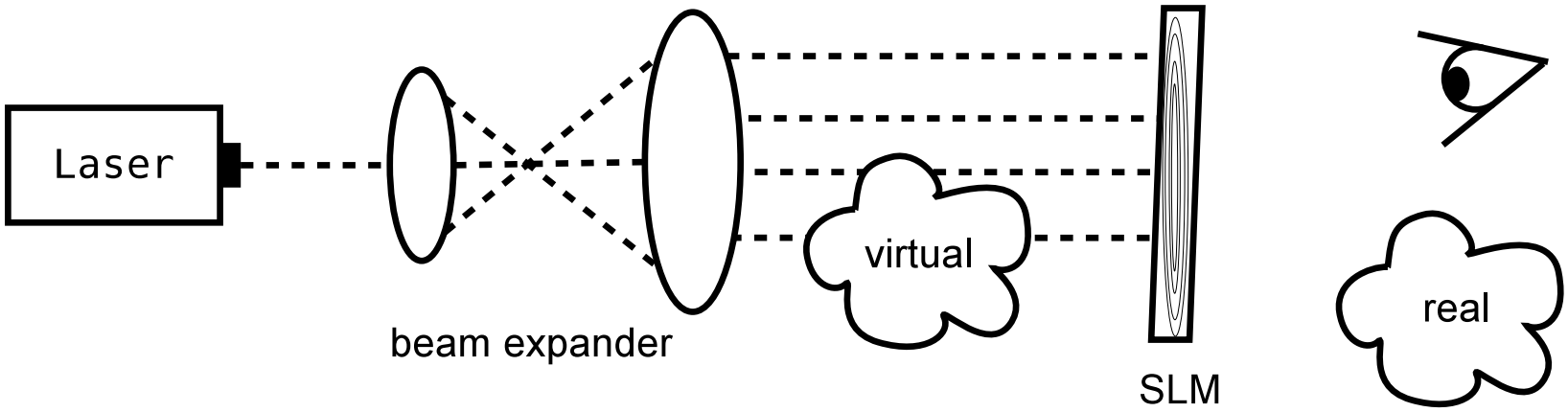
JÜLICH FORSCHUNGSZENTRUM

Image sources

- Holovideo: `http://xenia.media.mit.edu/~lucente/holo/36MB/persp.gif`
- Active tiling: Computer-Generated Holography as a Generic Display Technology, C. Slinger et al.
- SeeReal: Large holographic displays as an alternative to stereoscopic displays, R. Häussler et al.

JÜLICH FORSCHUNGSZENTRUM

Reconstruction setup



- Laser might be replaced with **LED**
- Light source is required to be **coherent**
- Virtual image visible behind SLM,
  real image in front of SLM

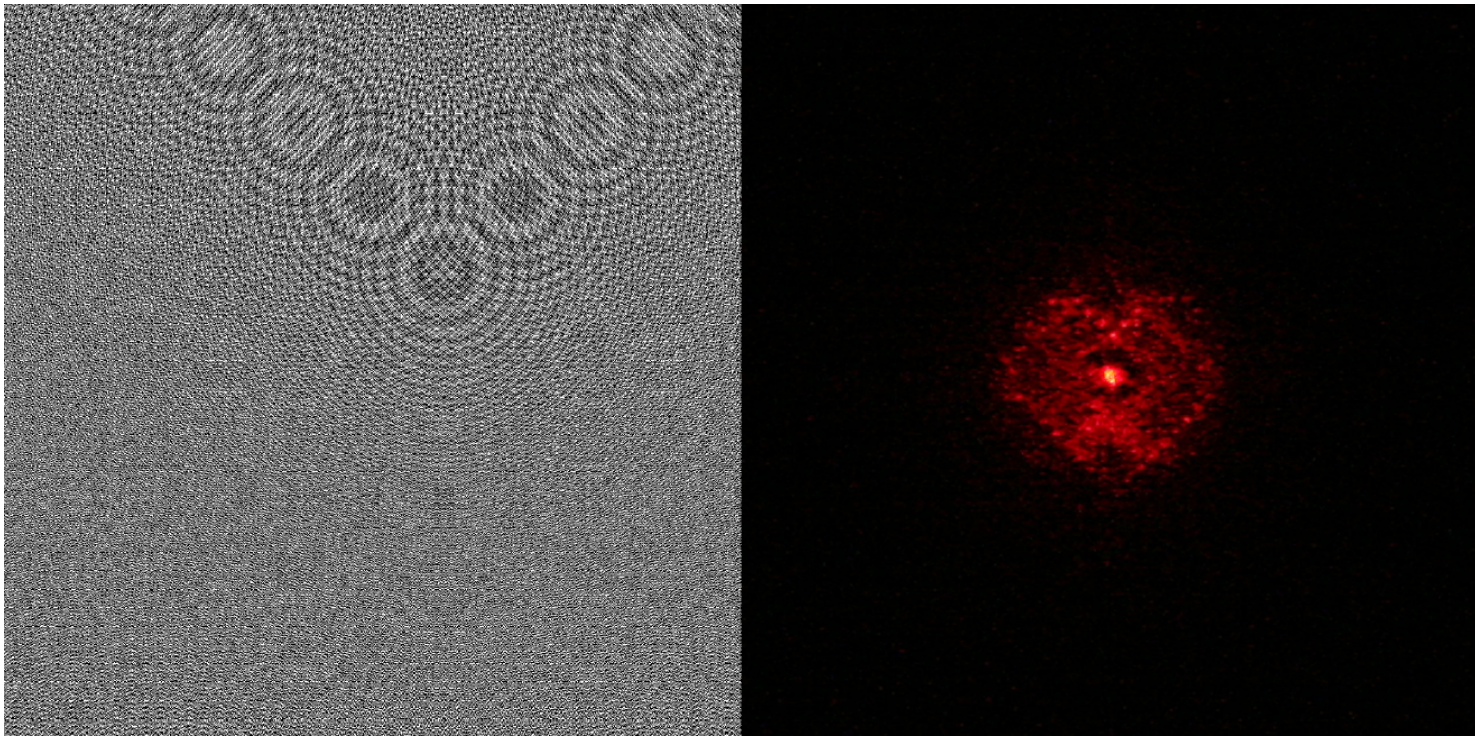
Part III: CGH examples

October 29, 2013 | Carsten Karbach, Jülich Supercomputing Centre (JSC)

JÜLICH FORSCHUNGSZENTRUM

Transparency film

- CGH printed with **laser printer**, reconstructed with **laser pointer**

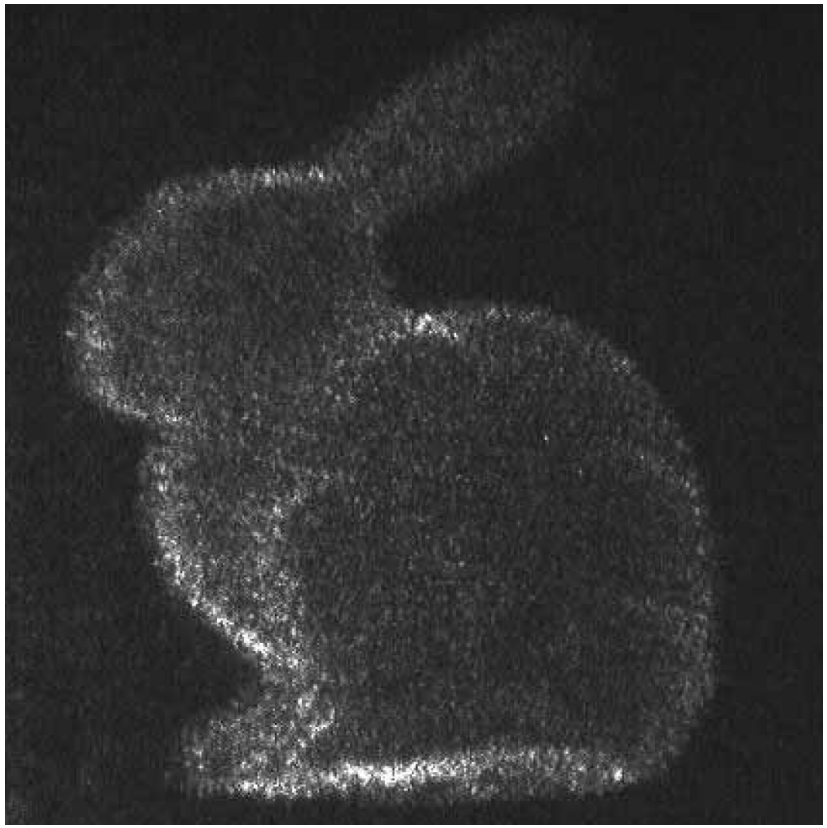


Source: Computer Generated Holography, J. Wendt

JÜLICH FORSCHUNGSZENTRUM

CGH with GPUs

- CGH calculated with **graphic hardware**, reconstruction with Brillian 1080 SLM

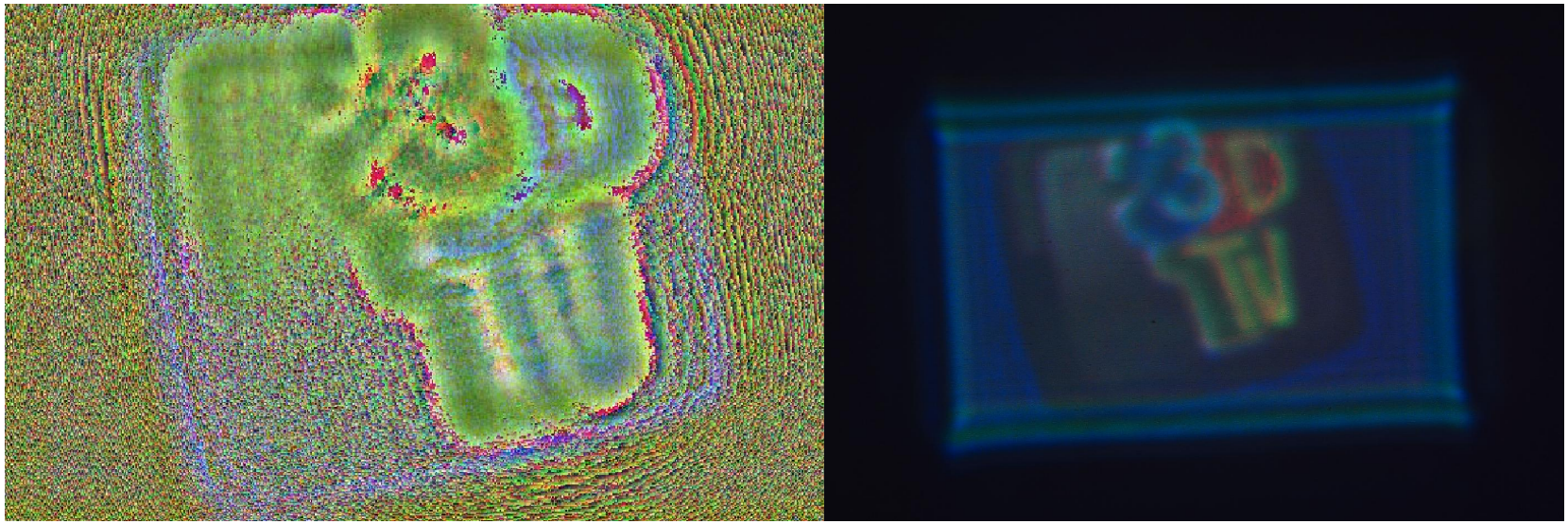


Source: Computer generated holography using parallel commodity graphics hardware, L. Ahrenberg et al.

JÜLICH FORSCHUNGSZENTRUM

Full color CGH

- Three layer CGH,
  reconstructed with **LEDs** and HoloEye SLM

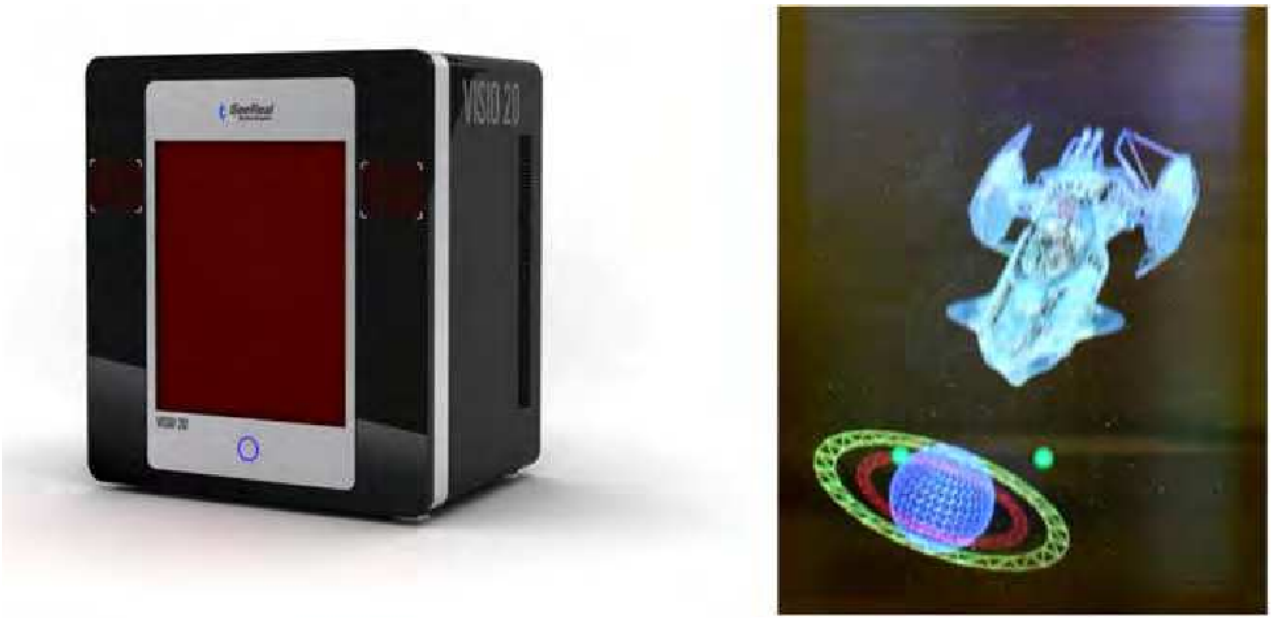


Source: Color Holographic Reconstruction Using Multiple SLMs and LED Illumination, F. Yaras et al.

JÜLICH FORSCHUNGSZENTRUM

SeeReal prototype

- CGH using sub holograms and eye tracking



Source: Holographic 3-D Displays – Electro-holography within the Grasp of Commercialization, S. Reichelt et al.

JÜLICH FORSCHUNGSZENTRUM

Summary

- Holography uses light **interference** and **diffraction** for reconstruction of a 3D-scene
- Promising visualization method
- Modeling concept of holography with sphere waves
- CGH calculates **interference pattern** of virtual scene

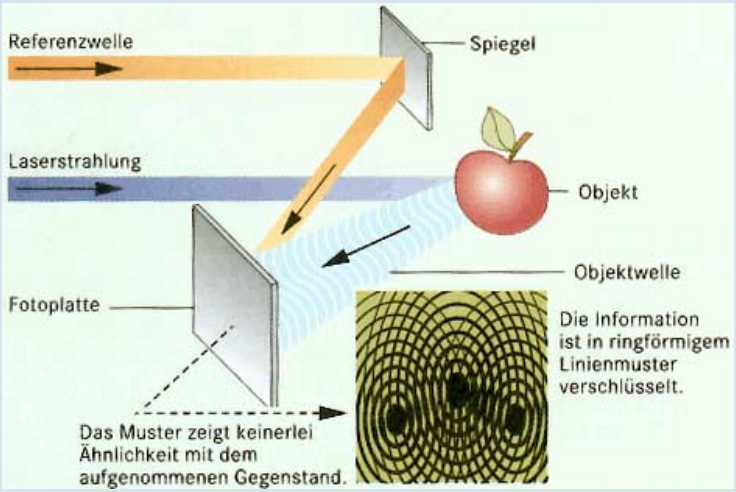
JÜLICH
FORSCHUNGSZENTRUM

Computer-Based Holography – Calculation

October 29, 2013 | Carsten Karbach

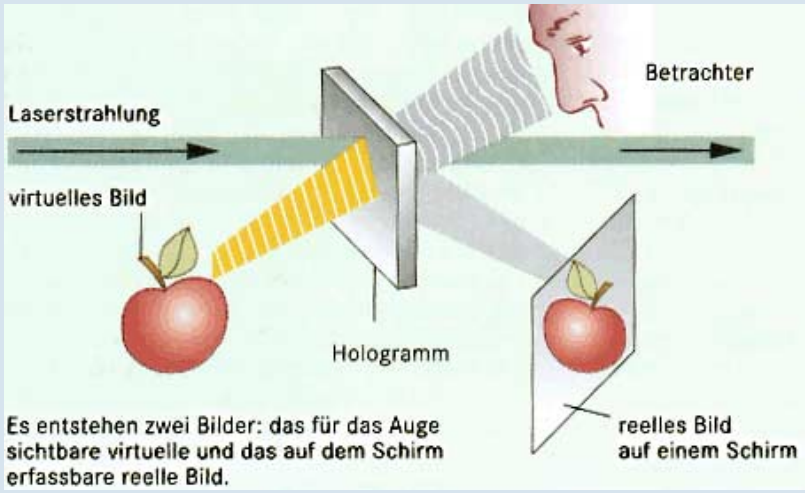
Holography

Recording



- **Interference** of reference and object beam
- Coherent light causes standing waves
- **Phase** and amplitude of light wave are stored

Reconstruction



- **Diffraction** of reconstruction beam on hologram generates object beam
- Reconstruction beam preferably identical to reference beam

Source: `http://www.hobby-output.info/inhalt/hologramm/erklaerung.htm`

Content

Part I: Simple mathematical derivation

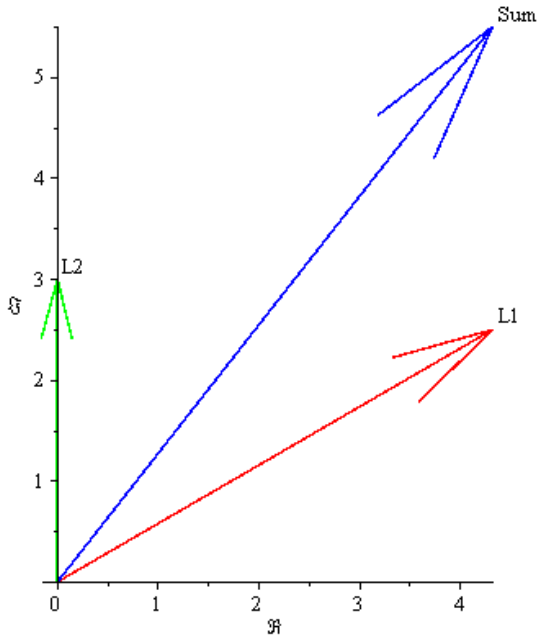
October 29, 2013 | Carsten Karbach

JÜLICH FORSCHUNGSZENTRUM

Modeling of light waves

Complex light

- Light wave defined by **electric field** $E(t)$
- $E(t) = A\,cos(\omega\,t + \phi)$
- Light has amplitude A and phase ϕ
- Complex notation $E(t) = A\,exp(i\,\phi)$ simplifies calculations
- Interference: $A_1 e^{i\phi_1} + A_2 e^{i\phi_2}$
- Intensity: $I = |E(t)|^2 = E(t) * \overline{E}(t)$



JÜLICH FORSCHUNGSZENTRUM

Hologram – Derivation I

- Interference of object wave and plane wave of reference beam
- $|H|^2$ is **intensity** stored in hologram

$$
\begin{aligned}
\text{Reference beam} : R &= A * e^{i\phi} \\
\text{Object beam} : O &= B * e^{i\sigma} \\
\text{Interference} : H &= R + O \\
\Rightarrow |H|^2 &= (R + O) * (\overline{R} + \overline{O}) \\
&= |R|^2 + |O|^2 + O * \overline{R} + R * \overline{O}
\end{aligned}
$$

JÜLICH FORSCHUNGSZENTRUM

Hologram –Derivation II

- Assumption: coherent light

$$
\begin{aligned}
R * |H|^2 &= R * \left(|R|^2 + |O|^2 + O * \overline{R} + R * \overline{O} \right) \\
&= R * (|R|^2 + |O|^2) + |R|^2 * O + R^2 * \overline{O} \\
&= \text{reference beam} + A^2 * O + R^2 * \overline{O} \\
&= \text{reference beam} + \text{virtual image} + \text{real image}
\end{aligned}
$$

- Diffraction of R by interference pattern $|H|^2$
- Virtual image looks identical to recorded object
- Real image is distracting overlap

JÜLICH FORSCHUNGSZENTRUM

Influence of coherent light

- For incoherent light the phases ϕ and σ vary statistically
- For coherent light the phase difference of both light sources $\phi - \sigma$ is **constant**

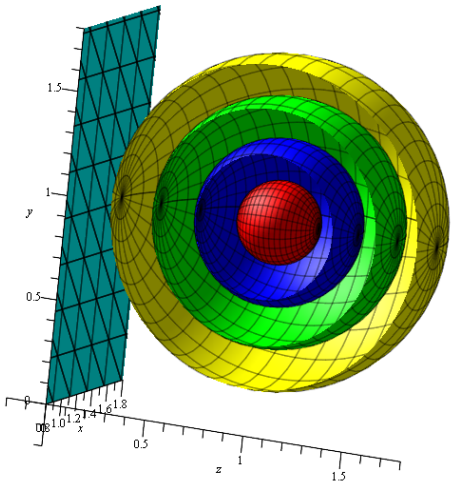
$$
\begin{aligned}
O * \overline{R} + R * \overline{O} &= A\,B\,(e^{i\,(\sigma-\phi)} + e^{-i\,(\sigma-\phi)}) \\
&= A\,B\,(cos(\sigma-\phi) + i\,sin(\sigma-\phi) + cos(\sigma-\phi) - i\,sin(\sigma-\phi)) \\
&= A\,B\,(2\,cos(\sigma-\phi)) \\
&= \begin{cases} 0, & \text{for incoherent light} \\ A\,B\,(2\,cos(\sigma-\phi)), & \text{for coherent light} \end{cases}
\end{aligned}
$$

- With incoherent light the intensity formula is $|H|^2 = |R|^2 + |O|^2$
- Only the intensities of R and O add up, no interference occurs

JÜLICH FORSCHUNGSZENTRUM

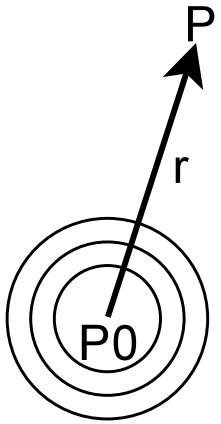
Sphere wave

- Light wave emitted from a point source P_0
- Uniform propagation of light in **all directions**
- Used for scene points



Formula

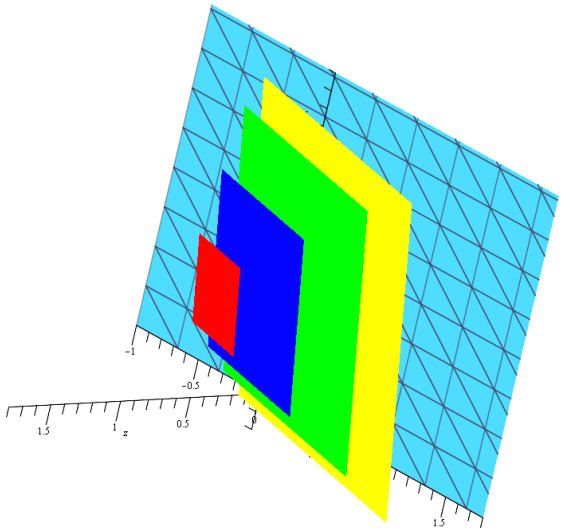
$$U_k(P) = A * \frac{e^{i\left(\frac{2\pi r}{\lambda}+\phi_0\right)}}{r}$$
$$r = |P - P_0|$$



JÜLICH FORSCHUNGSZENTRUM

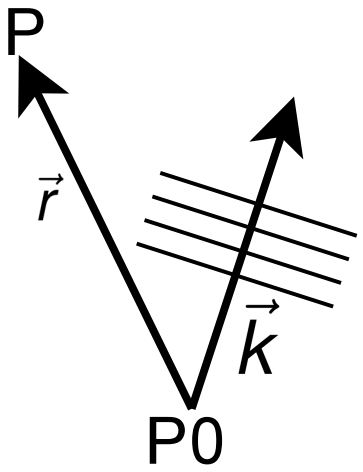
Plane wave

- Light propagates with direction $\vec{k}$
- Identical phase in each plane perpendicular to $\vec{k}$
- Used for reference beam



Formula

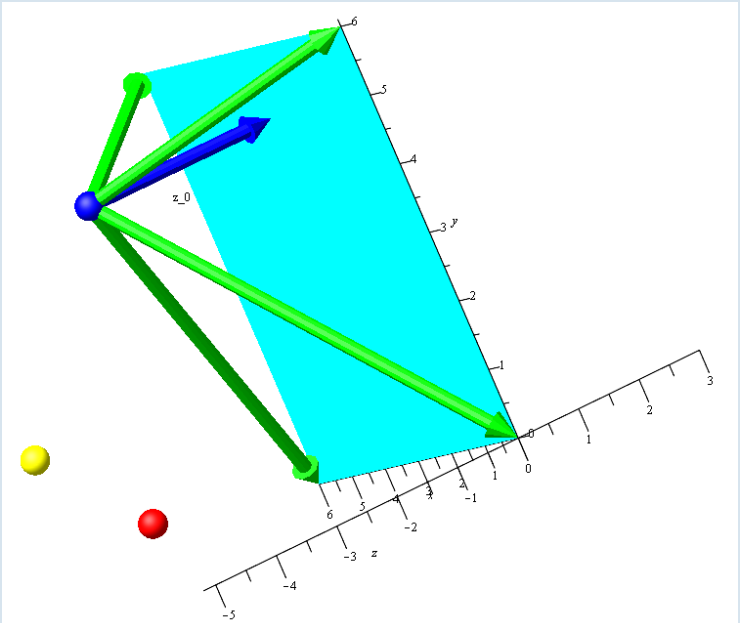
$$U_e(P) = A * e^{i\left(\frac{2\pi\langle\vec{k},\vec{r}\rangle}{\lambda}+\phi_0\right)}$$
$$\vec{r} = \vec{P} - \vec{P}_0$$



JÜLICH FORSCHUNGSZENTRUM

Calculate CGHs

Setting



Calculation

$$
\begin{aligned}
P_l &:= \text{source of sphere wave } l \\
p_{lm} &:= \textit{pixel}(l, m) \text{ on hologram plane} \\
U_l(P) &= h_l \frac{e^{i\frac{2\pi r}{\lambda}}}{r},\ r = |P - P_l|,\ h_l = amp \\
R(P) &= \text{reference beam} = e^{i\frac{2\pi\langle \vec{k}, \vec{P} - \vec{P}_0 \rangle}{\lambda}} \\
H(p_{lm}) &= \text{electric field on hologram plane} \\
&= R(p_{lm}) + \sum_{l=1}^{n} U_l(p_{lm})
\end{aligned}
$$

$\Rightarrow$ Calculate $|H(p_{lm})|^2$

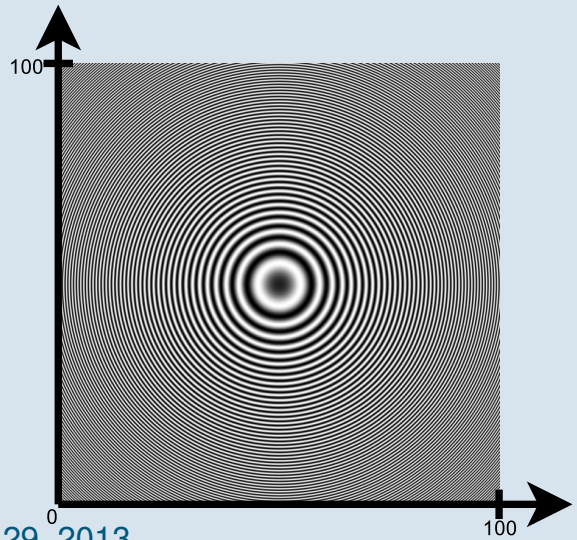
for each pixel on hologram plane

JÜLICH FORSCHUNGSZENTRUM

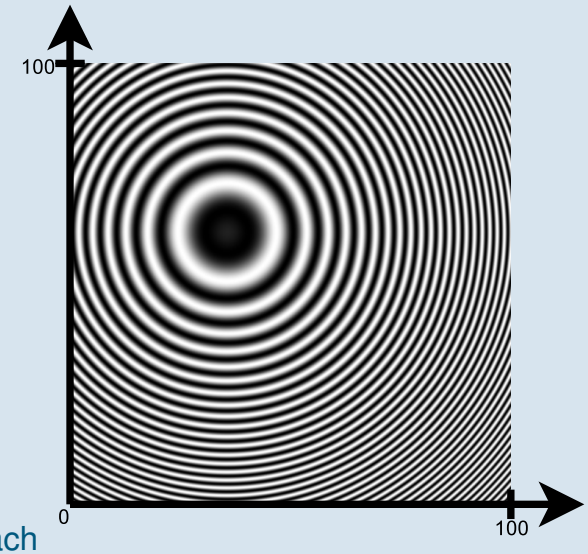
Encoding a 3D-vector in a hologram plane

- Each P_l emits a sphere wave
- Hologram of a single sphere wave generates **Fresnel zone plate**
- x and y define the center of the zone plate
- z is encoded in the ring distances

$P_l = (50, 50, 3)$



$P_l = (20, 70, 10)$



Part II: Light propagation

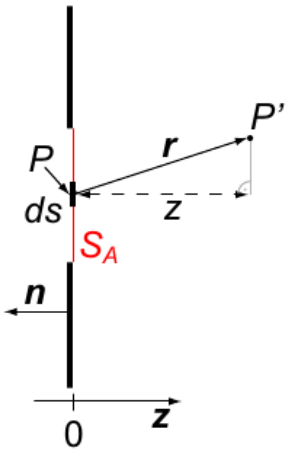
October 29, 2013 | Carsten Karbach

JÜLICH FORSCHUNGSZENTRUM

Light propagation

Problem

- Calculate light contribution of point P to P'
- Fresnel-Kirchhoff: $U(P') = \frac{1}{i\lambda} \iint_{S_A} U(P) \frac{e^{ikr}}{r} \frac{1}{2} (1 + \frac{\langle \vec{r}, -\vec{n} \rangle}{r}) ds$,
  with $\vec{n}$ = normal to plane S_A, $k = \frac{2\pi}{\lambda}$
- Direct calculation is expensive (2D integral for each point P')



Solutions

1. **Fresnel** approximation for near field
2. **Fraunhofer** approximation for far field
3. Use of **Fourier** transformation

Source: *Optische Fouriertransformation*, Westfälische Wilhelms-Universität Münster

JÜLICH FORSCHUNGSZENTRUM

Comparison to simple derivation

Fresnel-Kirchhoff

$$U(P') = \frac{1}{i\lambda} \iint_{S_A} U(P) \frac{e^{ikr}}{r} \frac{1}{2} (1 + \frac{\langle \vec{r}, -\vec{n} \rangle}{r}) ds$$

Discretization of S_A into n points $\Rightarrow$

$$U(P') \approx \frac{1}{i\lambda} \sum_{l=1}^{n} U(P_l) \frac{e^{ikr}}{r} \frac{1}{2} \left(1 + \frac{\langle \vec{r}, -\vec{n} \rangle}{r}\right)$$

$-\vec{n} \approx \frac{\vec{r}}{r}$, true for $z \gg x^2 + y^2 \Rightarrow$

$$U(P') \approx \frac{1}{i\lambda} \sum_{l=1}^{n} U(P_l) \frac{e^{ikr}}{r}$$

Start a sphere wave from each P_l on S_A

Sphere waves

$$H(p_{lm}) = R(p_{lm}) + \sum_{l=1}^{n} U_l(p_{lm}) =$$

$$R(p_{lm}) + \sum_{l=1}^{n} h_l \frac{e^{ikr}}{r}$$

Start a sphere wave from each scene point P_l

- Both approaches use **Huygens-Fresnel** principle

Fresnel approximation

Source: *Optische Fouriertransformation*, Westfälische Wilhelms-Universität Münster

- $|x|, |y|, |x'|, |y'| \ll z$
- $\vec{r} = P' - P = \begin{pmatrix} x_h \\ y_h \\ z \end{pmatrix} - \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$
- $|\vec{r}| \overset{(*)}{\approx} z + \frac{(x_h - x)^2}{2z} + \frac{(y_h - y)^2}{2z}$ in phase
- $|\vec{r}| \approx z$ in amplitude, inaccuracy uncritical

$\Rightarrow$ $U(P') = \frac{1}{i\lambda z} e^{ikz} \iint_{S_A} U(P) e^{\frac{ik}{2z}((x_h - x)^2 + (y_h - y)^2)} dxdy$

$(*)$ Approximation for $|\vec{r}|$

- $\sqrt{1+x} = 1 + \frac{x}{2} + O(x^2)$, Taylor series eval. at $x = 0$
- $|\vec{r}| = \sqrt{z^2 + (x_h - x)^2 + (y_h - y)^2} = z\sqrt{1 + \frac{(x_h - x)^2}{z^2} + \frac{(y_h - y)^2}{z^2}}$
- Use $\frac{(x_h - x)^2}{z^2} + \frac{(y_h - y)^2}{z^2}$ as x in Taylor series

$\Rightarrow$ $|\vec{r}| \approx z(1 + \frac{(x_h - x)^2}{2z^2} + \frac{(y_h - y)^2}{2z^2})$

JÜLICH FORSCHUNGSZENTRUM

Fraunhofer approximation

Source: *Optische Fouriertransformation*, Westfälische Wilhelms-Universität Münster

- With $z \gg x^2 + y^2 \Rightarrow |\vec{r}| = z + \frac{x_h^2+y_h^2}{2z} - \frac{x_h x+y_h y}{z} + \frac{x^2+y^2}{2z}$
  $\approx z + \frac{x_h^2+y_h^2}{2z} - \frac{x_h x+y_h y}{z}$, for phase
- $U(P') = \frac{1}{i\lambda z} e^{ikz} e^{\frac{ik}{2z}(x_h^2+y_h^2)} \iint_{S_A} U(P) e^{\frac{-ik}{z}(x_h x+y_h y)} dxdy$
- Advantage: $\iint_{S_A} U(P) e^{\frac{-ik}{z}(x_h x+y_h y)} dxdy = FT(U)(\frac{x_h}{z\lambda}, \frac{y_h}{z\lambda})$
- Complexity of calculation is proportional to
  2D Fourier transformation

JÜLICH FORSCHUNGSZENTRUM

Use of Fourier transformation

Source: *Optische Fouriertransformation*, Westfälische Wilhelms-Universität Münster

- 2D FT: $F(u,v) = \int_{-\infty}^{\infty}\int f(x,y) * e^{-i2\pi\,(x\,u+y\,v)}\,dxdy$
- Goal: use FFT in hologram generation
- Start with Fraunhofer approximation:

  $\iint_{S_A} U(P) e^{\frac{-ik}{z}(x_h x + y_h y)}\,dxdy =$

  $\iint_{S_A} U(P(x,y)) e^{-i2\pi(\frac{x_h}{z\lambda}x + \frac{y_h}{z\lambda}y)}\,dxdy = F(U)(\frac{x_h}{z\lambda}, \frac{y_h}{z\lambda})$
- With more effort FT can also be used for Fresnel approximation

JÜLICH FORSCHUNGSZENTRUM

FT for Fresnel

Source: *Optische Fouriertransformation*, Westfälische Wilhelms-Universität Münster

$$\iint_{S_A} U(P) e^{\frac{ik}{2z}((x_h-x)^2+(y_h-y)^2)} dxdy =$$

$$\iint_{S_A} U(P) e^{\frac{i2\pi}{\lambda 2z}(x_h^2-2xx_h+x^2+y_h^2\ldots)} dxdy =$$

$$e^{i\frac{\pi}{\lambda z}(x_h^2+y_h^2)} \iint_{S_A} U(P) e^{\frac{i\pi}{\lambda z}(x^2+y^2)} e^{-i\frac{2\pi}{z\lambda}(xx_h+yy_h)} dxdy =$$

$$e^{i\frac{\pi}{\lambda z}(x_h^2+y_h^2)} F\Big(U(P) e^{\frac{i\pi}{\lambda z}(x^2+y^2)}\Big)\Big(\frac{x_h}{z\lambda}, \frac{y_h}{z\lambda}\Big)$$

- Expanding the formula, and extracting the parts depending on x and y leads to a formula using Fourier transformation

JÜLICH FORSCHUNGSZENTRUM

Numerical reconstruction

- **Inverse operation** to hologram generation
- Calculate original light wave from a hologram
- Input is either a CGH or
  a digital hologram captured with a CCD
- Allows to calculate amplitude
  and phase of source light wave
- Calculation of light wave front at arbitrary location

JÜLICH FORSCHUNGSZENTRUM

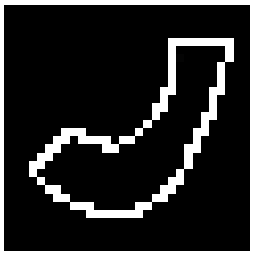
Numerical reconstruction – Idea

- Almost identical to normal hologram calculation!
- **Diffraction** of reference beam on hologram
- Use Fresnel-Kirchhoff or Fresnel-approximation:
  $U(x,y,z) = \frac{1}{i\lambda z} e^{i\frac{2\pi z}{\lambda}} \iint_{S_A} h(x_h,y_h) * e^{i\frac{\pi}{\lambda z}((x-x_h)^2+(y-y_h)^2)} dx_h dy_h$
- Reconstruction of original wave in any plane parallel to hologram plane
- Discretization of integral leads to: $U(x,y,z) = \frac{1}{i\lambda z} e^{i\frac{2\pi z}{\lambda}} \sum_{k=0}^{k=N-1} \sum_{l=0}^{l=M-1} h(x_k,y_l) * e^{i\frac{\pi}{\lambda z}((x-x_k)^2+(y-y_l)^2)}$

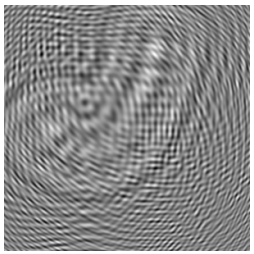
JÜLICH FORSCHUNGSZENTRUM

Numerical reconstruction – Example

1. Choose original object



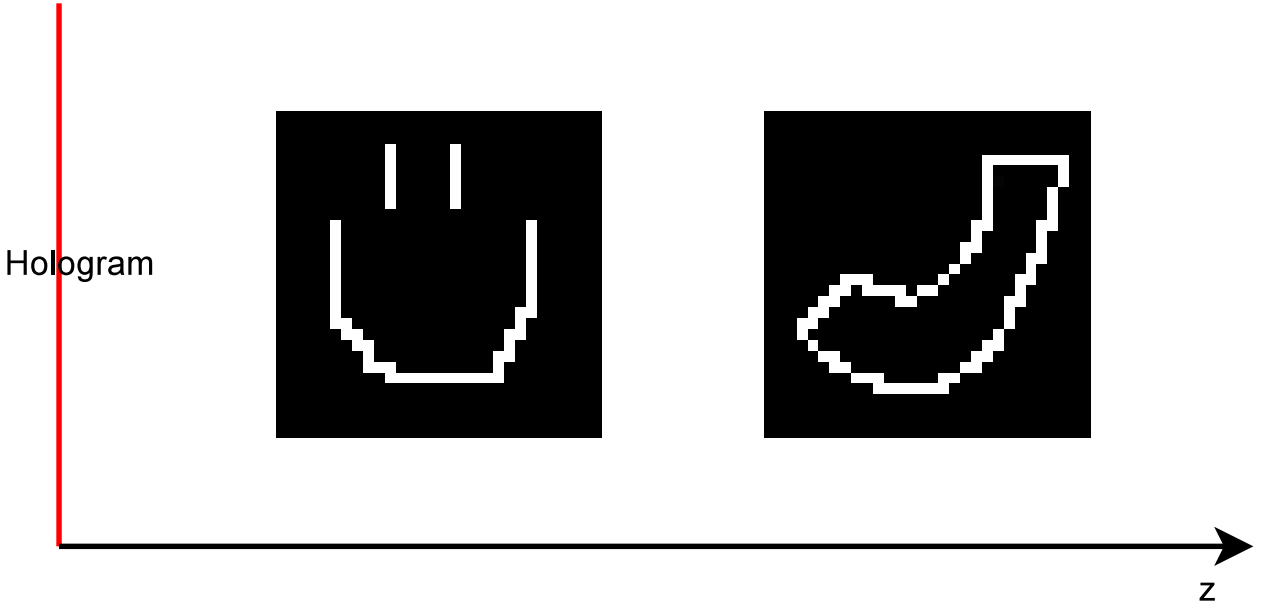
2. Calculate CGH e.g. via Fresnel-approximation



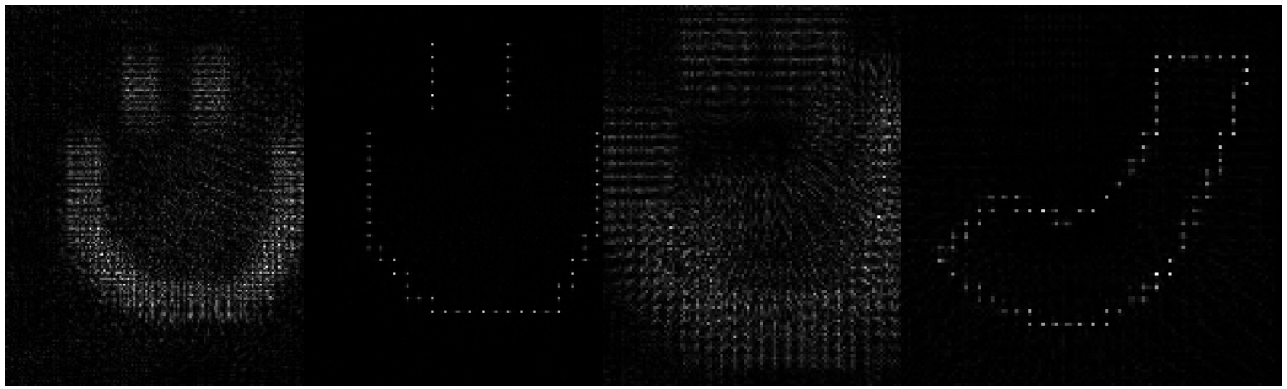
3. Use CGH amplitude $h(x_h, y_h)$ and reconstruct

JÜLICH
FORSCHUNGSZENTRUM

Numerical reconstruction 3D example



Reconstruction:



JÜLICH
FORSCHUNGSZENTRUM

Part III: Software design

October 29, 2013 | Carsten Karbach

JÜLICH FORSCHUNGSZENTRUM

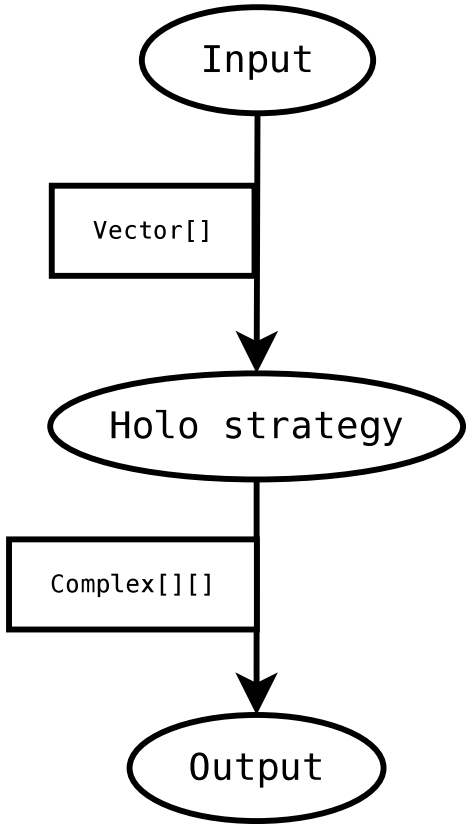
Algorithm

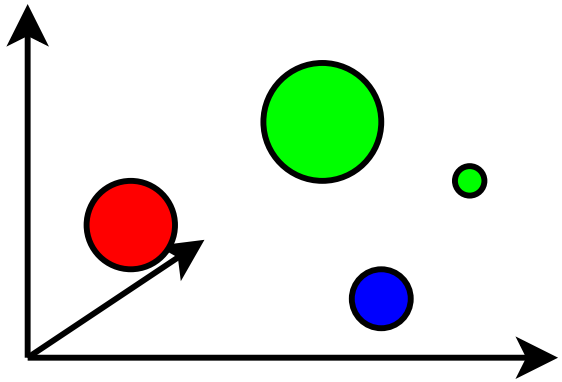
```
Complex* generateHologram(points, dim)

result = new Complex[dim*dim]
for(y=0; y < dim, y++)
    for(x=0; x < dim; x++)
        platePoint(x/pixelPerMeterX,y/pixelPerMeterY,0.0)
        for(i=0; i<points.size(); i++)
            point = points[i]
            distance = platePoint.getDistance(point), phase = 2*PI/lambda*distance
            amplitude = point.getAmplitude()/distance
            contribution.setReal( amplitude*cos(phase) )
            contribution.setImag( amplitude*sin(phase) )
            result[y*dim+x] += contribution
return result
```

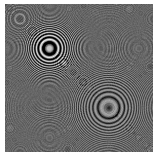
- Use of simple derivation, sphere waves
- Input:
  - **points**: source points
  - **dim**: pixels for width and height
- Output is array of Complex values
- Fresnel/Fraunhofer approximation can also be used here

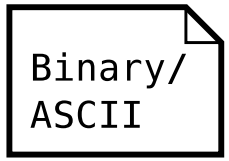
CGH application





Raycast / Fresnel / Fraunhofer
Fourier





JÜLICH FORSCHUNGSZENTRUM

CGH application – remarks

- Comparable with **VTK visualization pipeline**
- Input is a list of 3D vectors for **point sources**
  - generated
  - from file
  - 2D image as source
- **Strategies** for CGH calculation:
  - Raycast: Sum of sphere waves
  - Fresnel: Raycast using Fresnel approximation, avoids square roots
  - Fraunhofer: Raycast with Fraunhofer approximation
  - *Fourier*: Discretize volume into planes, run 2D DFT for each plane
- **Output** 2D image, binary / XML file

JÜLICH FORSCHUNGSZENTRUM

Parallelization

Approaches

1. **Domain** decomposition of 2D hologram
2. Decomposition of **scene points**, global sum
3. **Time** decomposition for dynamic images

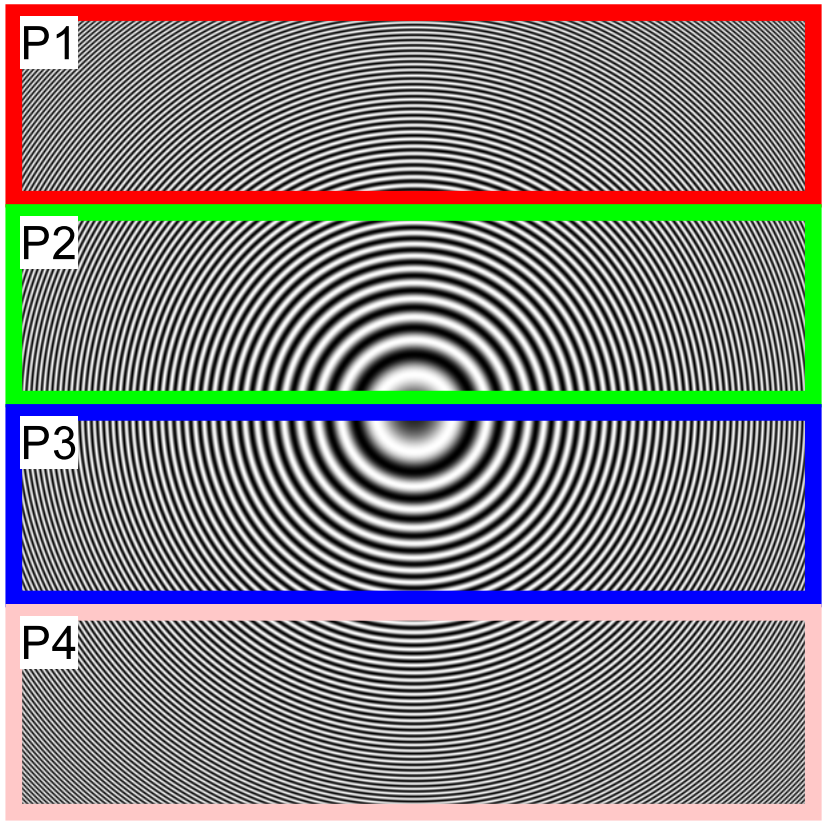
Implementation

- Java prototype for testing
- C++ with MPI/CUDA implementation of approach 1 as efficient solution

JÜLICH FORSCHUNGSZENTRUM

Domain decomposition

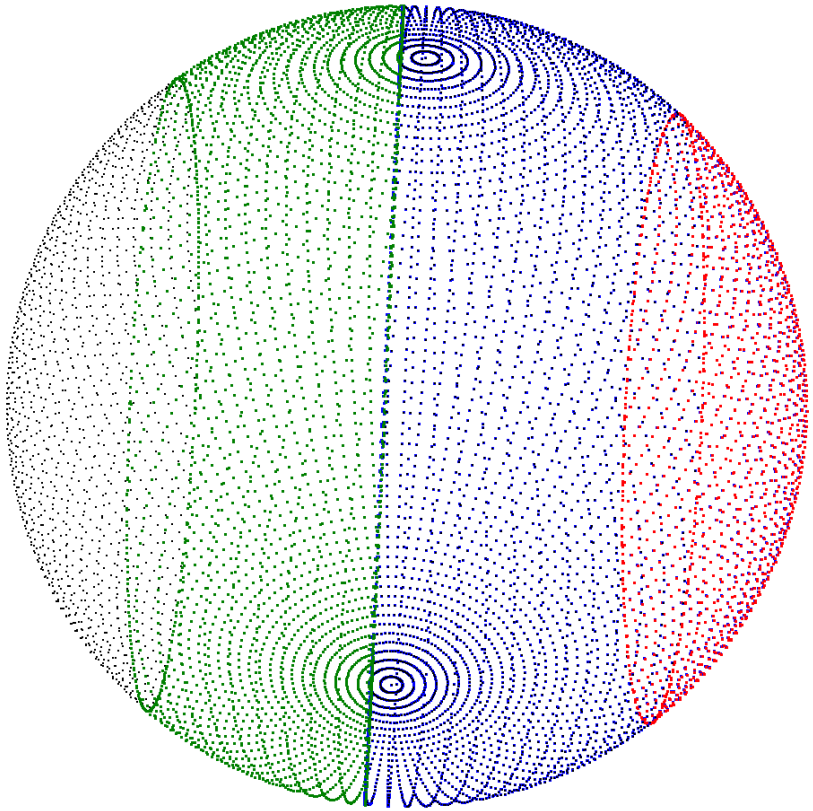
- Distribute image space uniformly on all CPUs
- Embarrassingly parallel, no communication required
- Target: interactive frame rates, short response times



JÜLICH FORSCHUNGSZENTRUM

Decomposition of scene points

- Distribute scene points across the CPUs
- **Global sum** of all contribution parts afterwards



JÜLICH FORSCHUNGSZENTRUM

OpenGL adapter

1. Get transformed vertices from the underlying visualization application
2. Optionally add more vertices by **interpolation**
3. Use these vertices as source points in hologram calculation
4. Render **interference pattern** onto screen
5. Use the screen as spatial light modulator

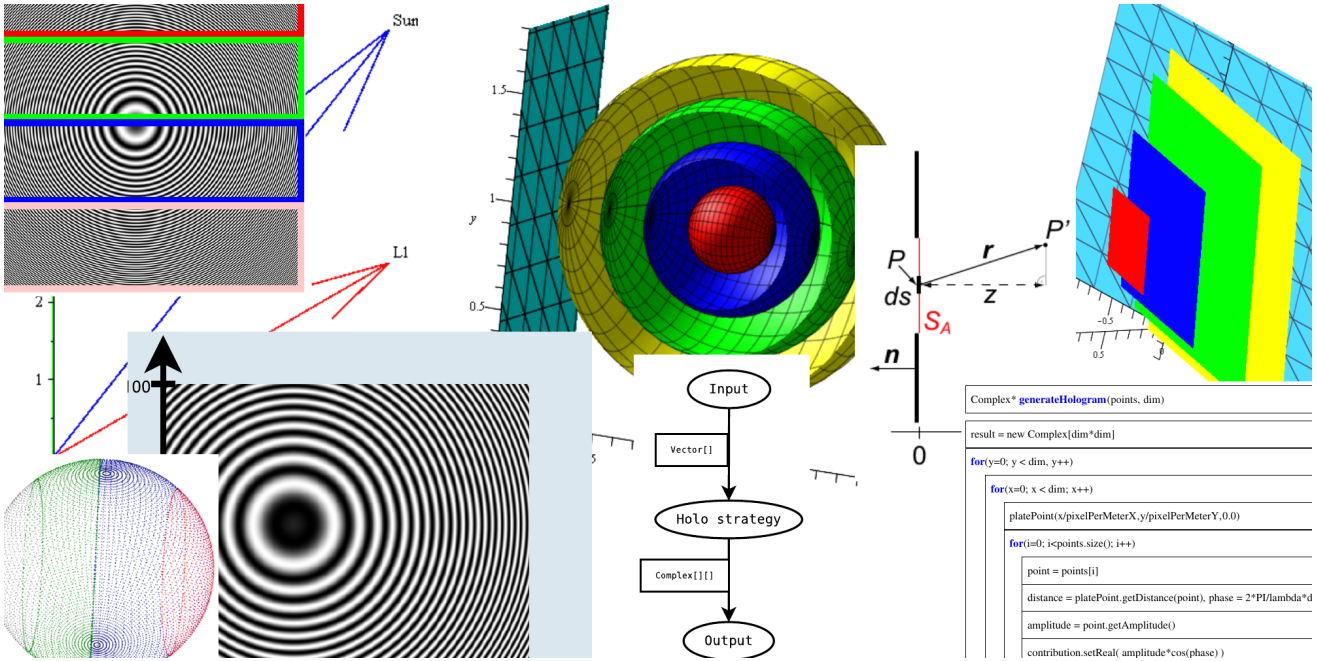
Remarks

- Simple adaption of existing visualization tools to CGH
- New input type for CGH application

Part IV: Outlook

October 29, 2013 | Carsten Karbach

Summary



- Using complex notation for light waves allows simple **mathematical derivation** of holographic principles
- **Fresnel-Kirchhoff** diffraction integral calculates light propagation, solved with approximations and FT
- Parallelized prototype developed using sphere waves as light sources

JÜLICH FORSCHUNGSZENTRUM

Outlook

- Performance analysis of parallel implementations for CGH generation
- Requirement for reconstruction: **affordable SLM**
- Evaluation of hologram calculation using **FFT**
- Integration of CGH rendering into existing visualization software (OpenGL adapter)