Lecture #13

Virtual Reality – Head Mounted Displays

Computer Graphics

Winter term 2020/21

Marc Stamminger

Up to now…

- Desktop Screen + GPU, 30-60 fps

Mobile Displays

- today: mobile, position-aware display devices
  - rendering can be done on high-end desktop

Head-mounted displays

navigation systems

augmented reality

HMDs: Head Mounted Displays

- Earliest real stereo display type
  - separate screens for left and right eye
    → stereo vision
- Advantages:
  - Large field of view
  - very good immersion
  - affordable
  - simple installation
- Disadvantages:
  - heavy, not comfortable
  - image distortions
  - environment is locked out
  - usage of controls difficult (mouse, keyboard, …)
  - single user only

Boom

- HMD mounted to boom
- Advantage w.r.t. HMD
  - larger resolution
  - easier to take on and off
  - less heavy
  - tracking by boom itself
- Disadvantage
  - smaller range of action
  - one hand needed
  - inertia
  - lower immersion

HMD

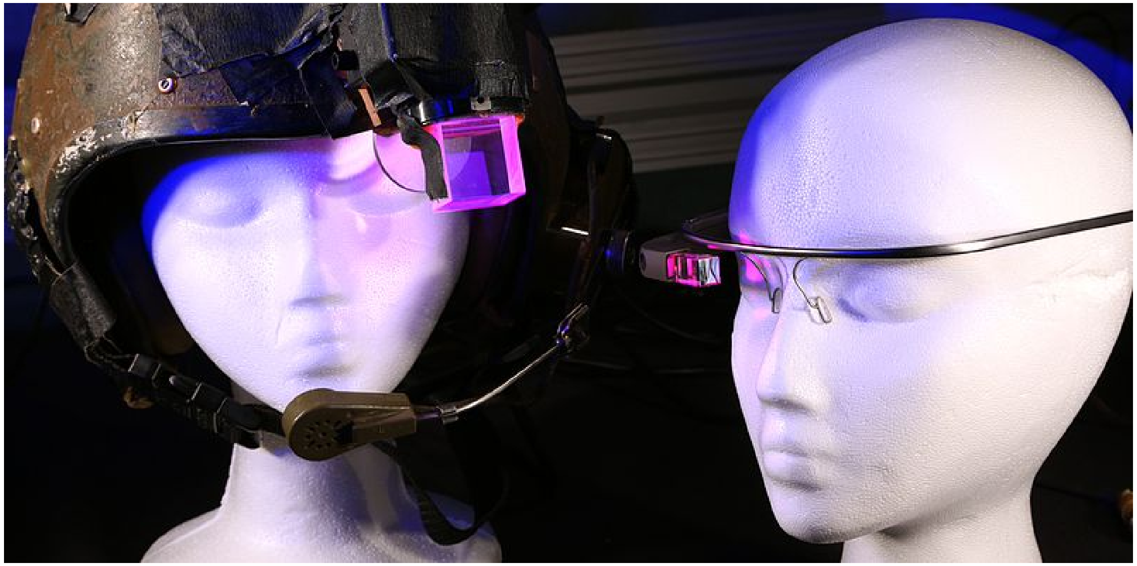
- First Head Mounted Display: Ivan Sutherland **1968!**
  - In fact, a see-through display
  - Tracking via boom or with ultrasound

Ivan Sutherland, 1968

https://www.youtube.com/watch?v=NtwZXGprxag

See-through Displays

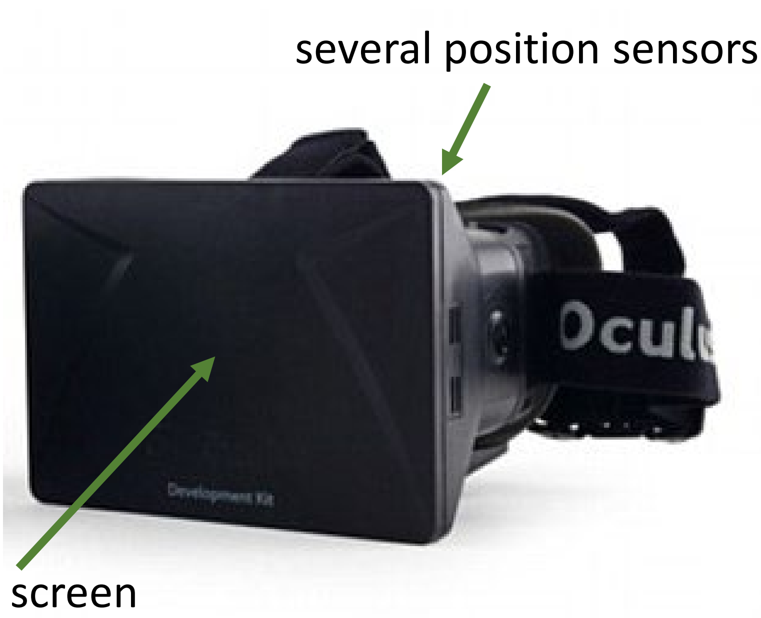
- Microsoft Hololens (left, 2016), Google Glasses (right, 2013)
- also not a new idea: Mann's Digital Eye Glass (center, 1980)



Oculus Rift

- (early) oculus rift
  - stereo rendering
  - position sensors track head movement and adapt view position accordingly
    → user can "look around" in virtual world

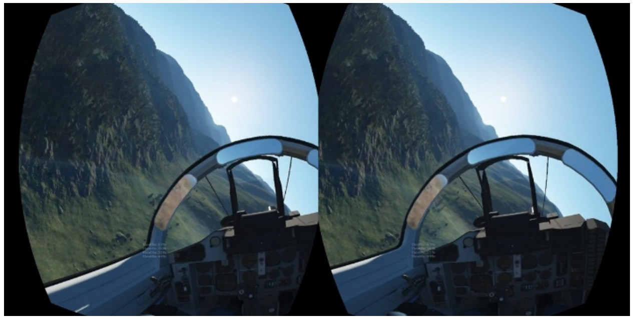
several position sensors



single screen

lenses

screen shot



left eye right eye

Successors

- HTC Vive

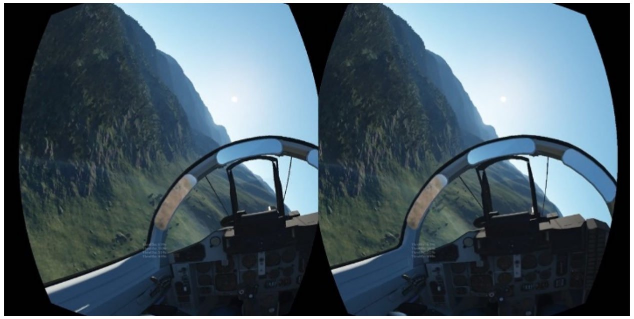
Samsung Gear VR

- Google Cardboard

Head Mounted Displays

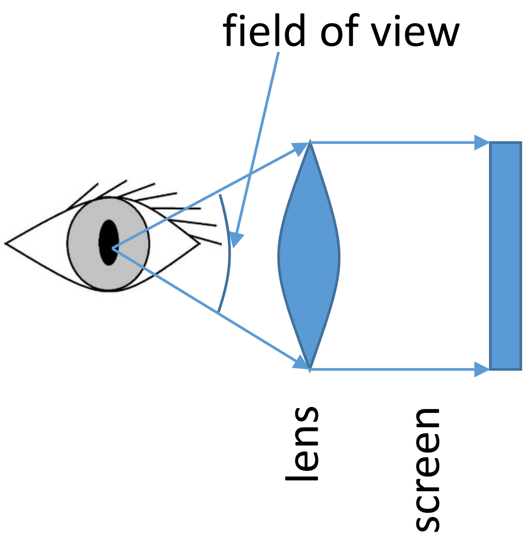
- in this lecture
  - correction of lens distortion
  - stereo rendering

- next lecture
  - tracking
  - latency



HMD Lenses

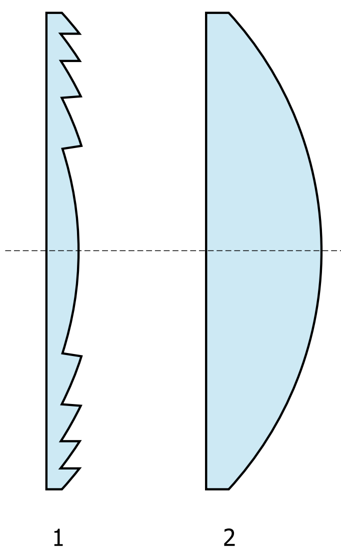
- let us see a sharp image of the screen, despite its closeness (~10 cm)
- increase the field of view (important for immersion)



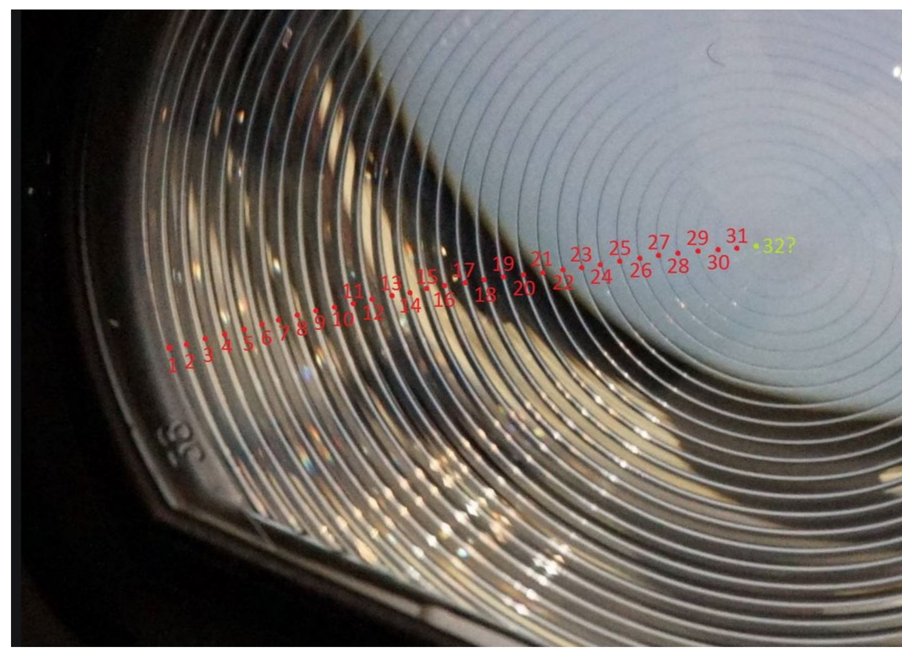
HMD Lenses

- Thick Lens (1)
  - thick, heavy
  - strong distortion
- Fresnel Lenses (2)
  - flat, less/no distortion
  - worse optical properties: lens flare / god rays

thick lens



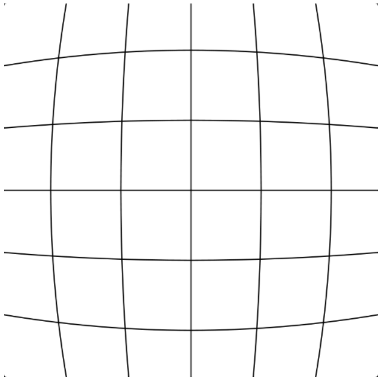
wikipedia.org

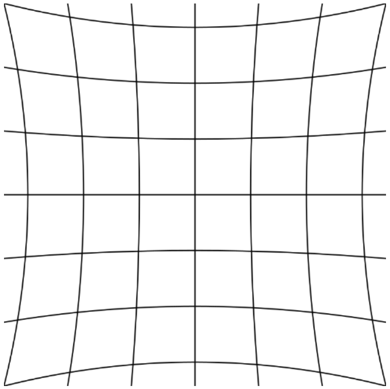


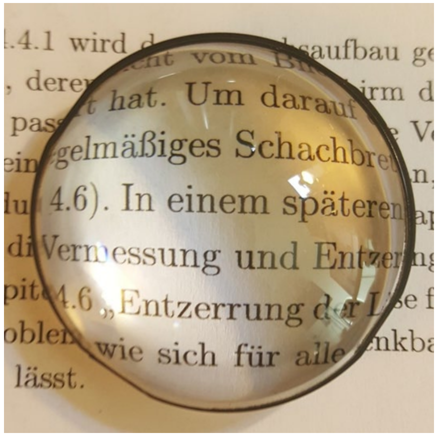
reddit.com

Lens Distortion

- In particular for thick lenses:
  Barrel Distortion / Pincushion distortion



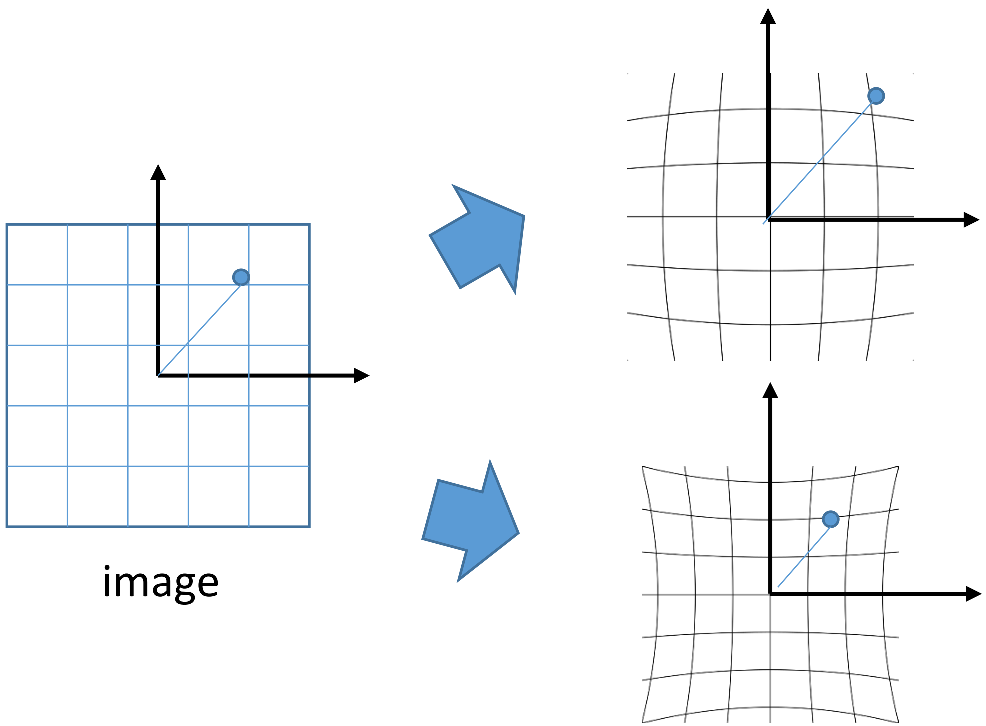




"Barrel distortion" by WolfWings - Own work. Licensed under Public Domain via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:Barrel_distortion.svg#/media/File:Barrel_distortion.svg

Lens Distortion

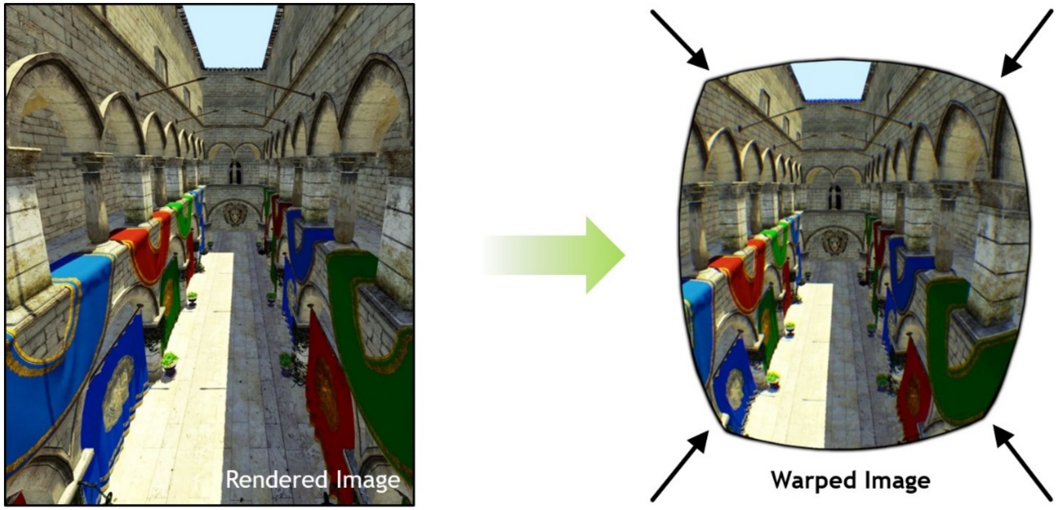
- barrel and pincushion distortion
  - go to polar coordinates (r, ϕ)
  - scale r by distortion function $d(r)$
  - Brown's distortion model: $d(r) = 1 + K_1 r^2 + K_2 r^4 + \cdots$



$$r \mathrel{*}= d(r)$$

Lens Distortion

- HMD lenses generate pincushion distortion
  → virtually increased field of view
  → distorted image
- undo distortion
  → undistorted image
  → still increased field of view
- inverse pincushion distortion = barrel distortion



nvidia.com

Lens Distortion

- How to do the undistortion?
  - vertex shader: move vertices of triangles accordingly
    → simple to implement and efficient, but:
    - also triangle edges should get bent accordingly
    - if only vertices are moved, triangle edges remain straight
    - no problem for small triangles, but for large triangles artifacts can become visible

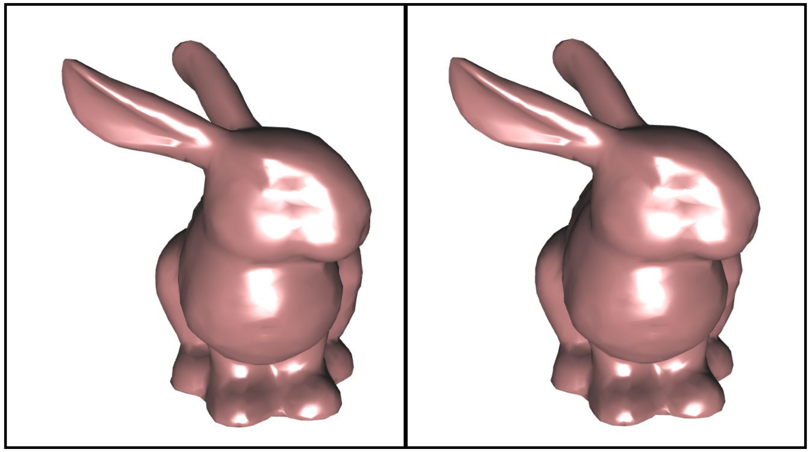
  - more general solution:
    undistort image in a second render pass using the fragment shader
    → render undistorted image, big enough, to texture
    → then rerender texture with proper distortion

Lens Distortion

- Algorithm
  - Render Scene to texture
  - bind texture
  - render quad covering entire screen with fragment shader:
    (K1 and K2 are global parameters that come from a device-dependent calibration)

```
// input uv is coordinate in screen from [0,1]^2
// subtract distortion center
uv -= center;
// compute r^2
r2 = uv[0]*uv[0] + uv[1]*uv[1];
// compute distortion
distortion = 1.0 + K1*r2 + K2*r2*r2 + …
// apply distortion
uv *= distortion;
// add distortion center back
uv += center;
// read texture
gl_FragColor = tex2D(image,uv);
```

Stereo Rendering



- Display shows two slightly different images, one for left, one for right eye
- this allows for **stereo rendering**
- both views are rendered with slightly displaced view point:
  → **stereo parallax**: slightly different views to a scene by two eyes

- Stereo parallax is only one depth cue
- Obviously, also with one eye alone depth perception is possible
  → several depth cues, parallax is only one of them

Stereo Rendering

Further depth cues

- motion parallax:
  head is constantly moving,
  nearby objects move faster
  in image than distant ones
- (b) color / saturation:
  very distant objects get blueish
- (c) shading: information about
  shape and thus depth
- (a) size: distant objects are smaller
  than close ones
- shadows (d): give hints on the
  relative position
- focus: eye lens accommodates
  → accommodation is depth cue

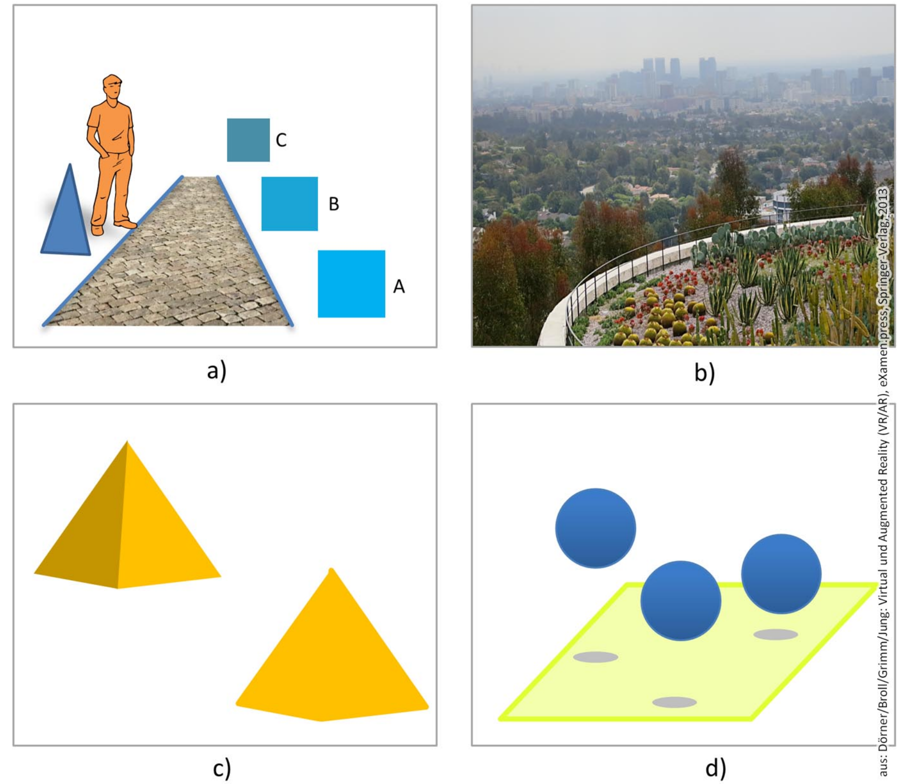


Abb. 2.3 Beispiele für Tiefenhinweise

Figures on this and the following pages: **Virtual und Augmented Reality (VR / AR)** herausgegeben von Ralf Dörner, Wolfgang Broll, Paul Grimm und Bernhard Jung, erschienen 2013 in der Reihe eXamen.press des Springer Verlags

Stereo Rendering

- With an HMD, we can generate
  - stereo parallax (next slides)
  - a) – d)
  - motion parallax
    requires fast head tracking
- but not:
  - focus !
  - our eyes accommodate to the displays distance, not the distance of the objects
  - usually, lenses move the display distance to about 2m
  - bad experience if focus depth and stereo parallax differ too much!

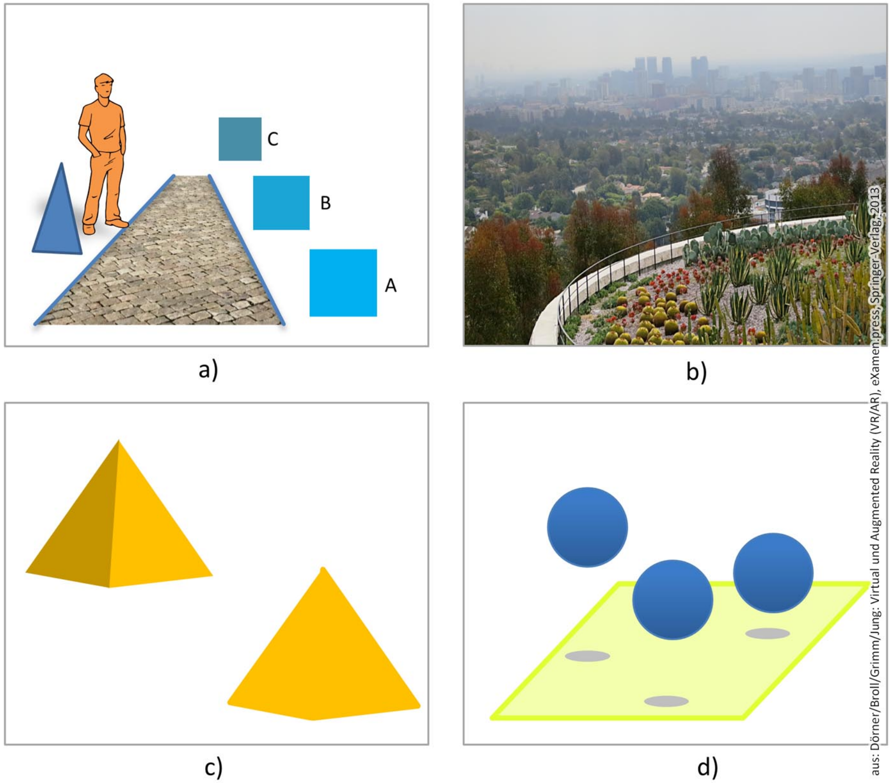
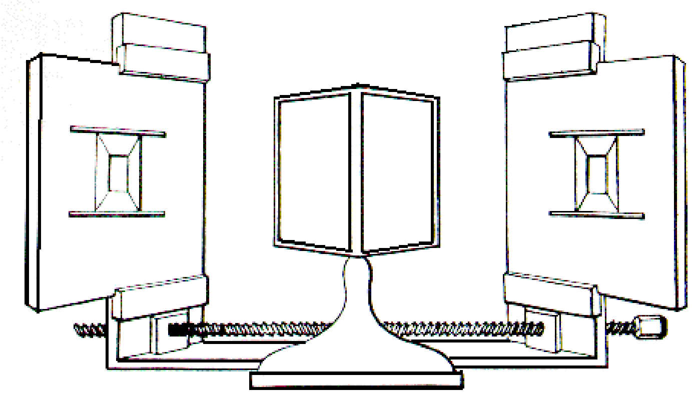


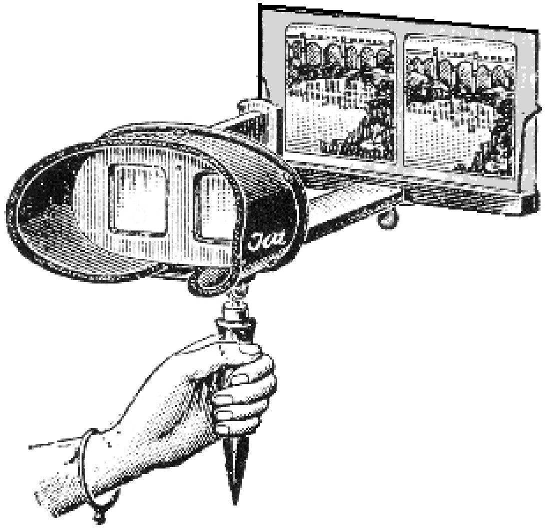
Abb. 2.3 Beispiele für Tiefenhinweise

aus: Dörner/Broll/Grimm/Jung: Virtual und Augmented Reality (VR/AR), eXamen.press, Springer-Verlag, 2013

History of Stereo Images

- Euklid (4. century B.C.)
- Sir Charles Wheatstone (1838)
- 1860: 1 Million Stereoscopes sold
- 50‘s:





Stereo Rendering Devices

- Other than HMD
- In Cinemas / on TVs:
  - Two images on one screen
  - Image separation using polarization filters / shutters / color filters

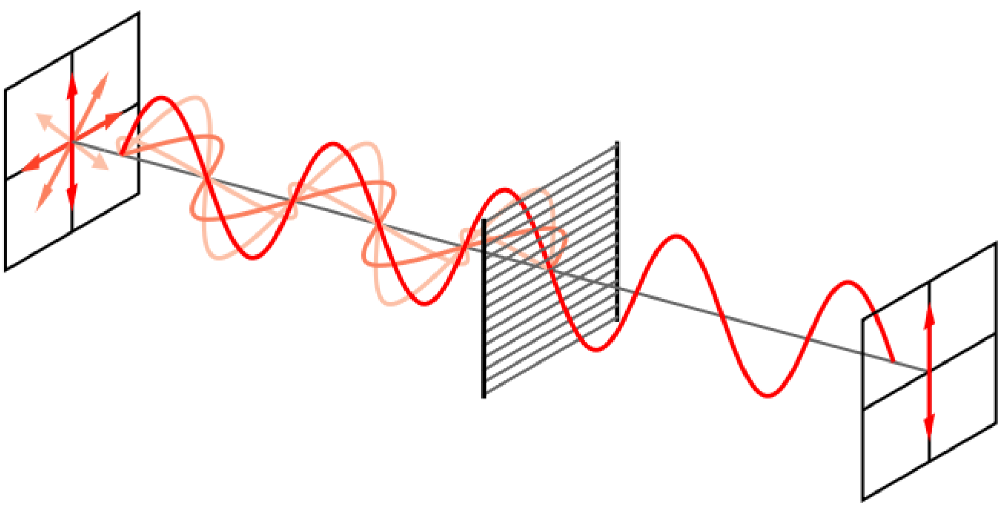
Stereo Rendering Devices

- Image separation using shutter glasses
  - display shows image for left and right eye in even and uneven frames
  - an infrared signal sends this information
  - shutter glasses turn “other” eye black
  - Image gets darker, but each eye sees only its own image
  - double frame rate required

Stereo Rendering Devices

- Image separation using polarization filters
  - Image for left and right eye are displayed on top of each other, but with different polarization
    → polarization filters
  - Glasses with corresponding polarization filters separate images

- → two projectors required



https://de.wikipedia.org/wiki/3D-Polarisationssystem

Stereo Rendering Devices

- Projector based systems

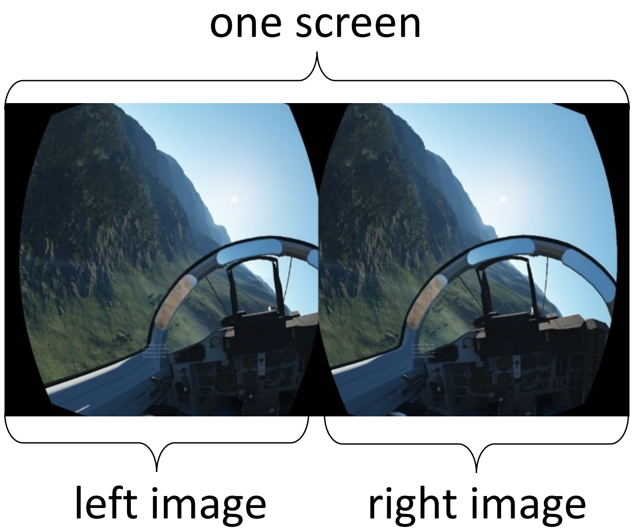
Stereo Rendering Devices

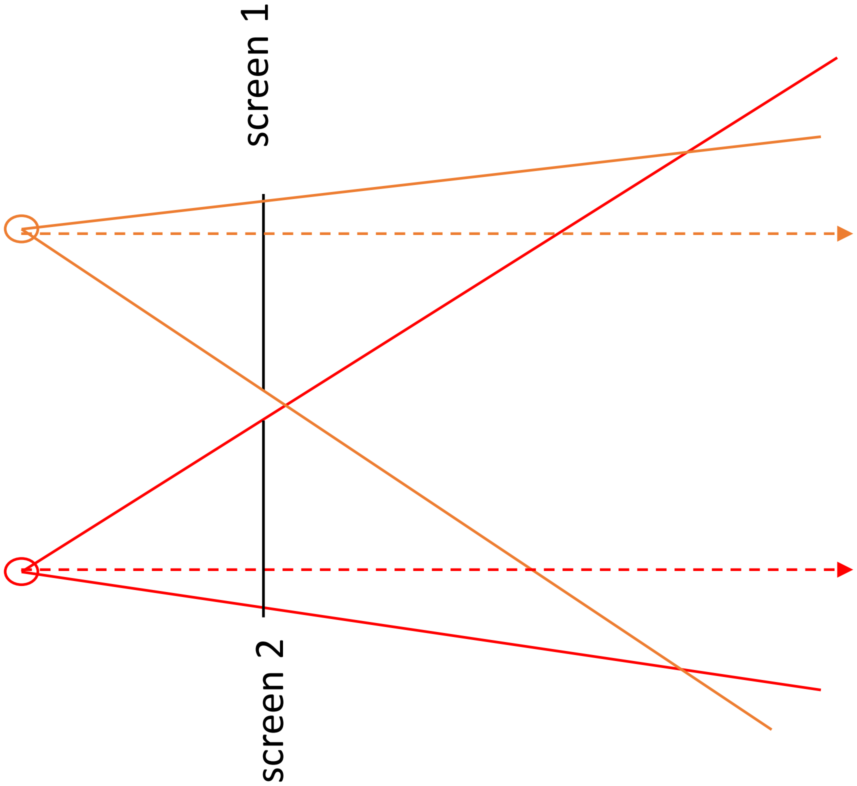
- Infitec-Glasses
  - projectors use different frequency bands for red, green and blue
  - glasses filter out proper image

infitec.net

Stereo Rendering Devices

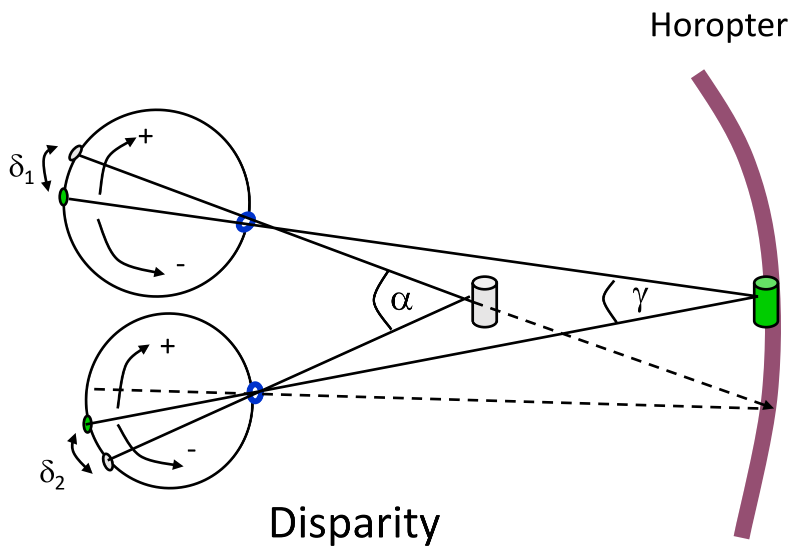
- For an HMD, the two eyes' images are separated spatially
- Often on the same screen: left eye sees left half, right one right half





Stereo Rendering

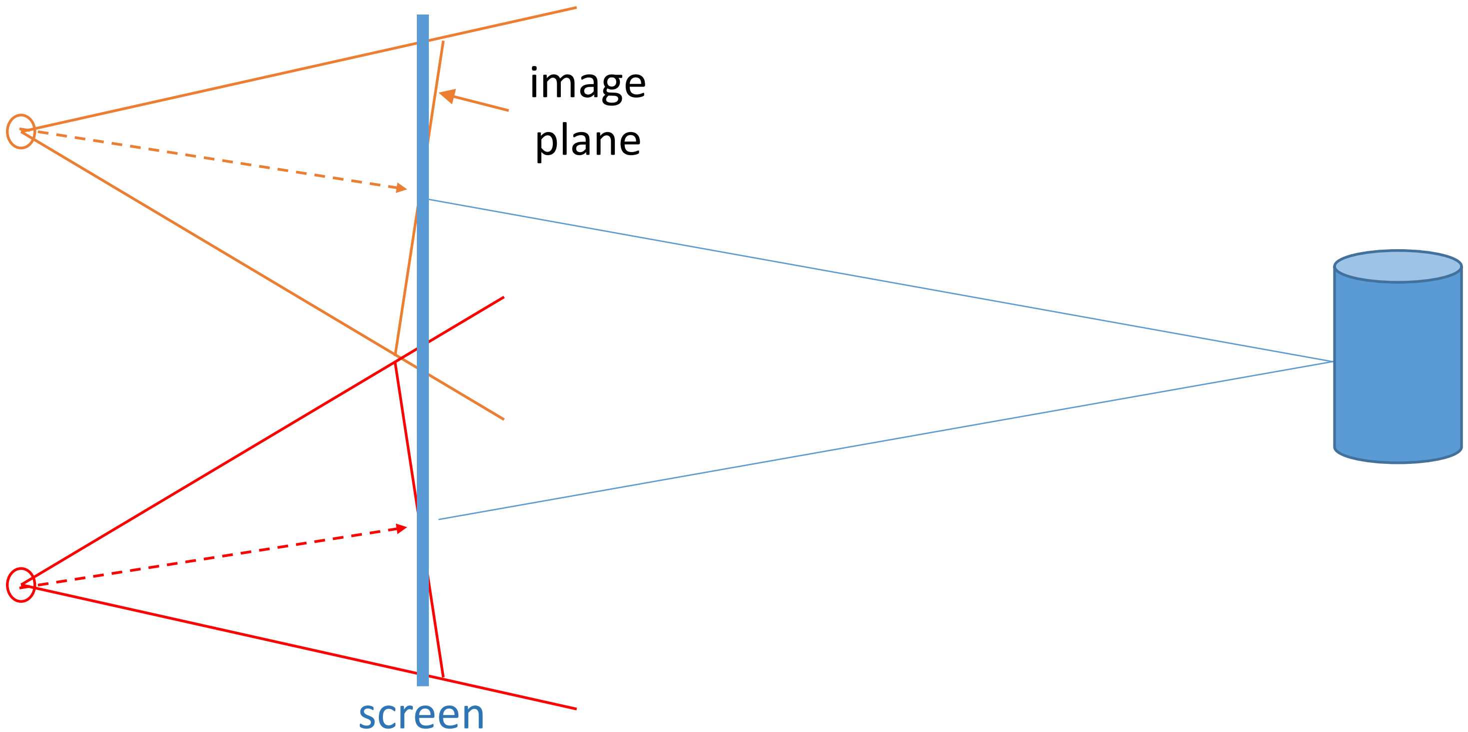
- Important depth cue, but only one of several
- only works for nearby objects (few meters)
- Disparity in the eye = $\delta_2 - \delta_1 = \gamma - \alpha$
  - Horopter = points with same depth as focused object
- Vergence: eyes are rotated, so that parallax of focused object is minimized



Disparity

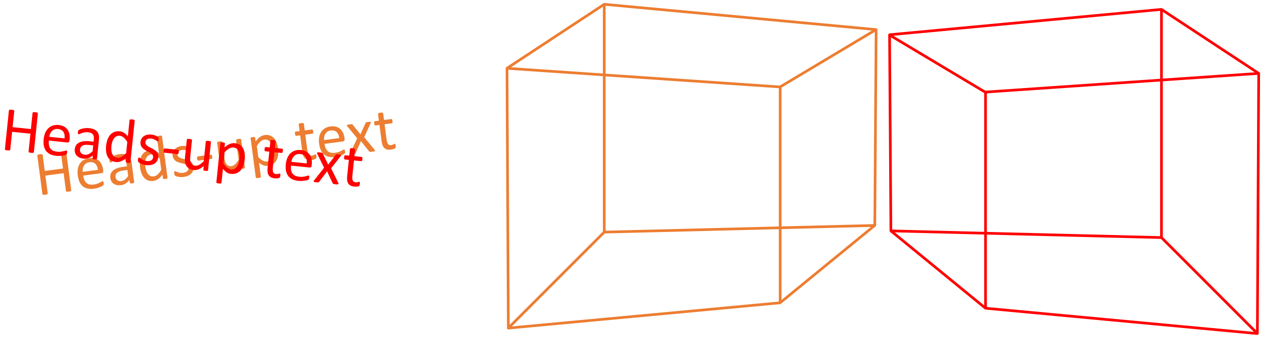
Stereo Rendering

- How to generate the two images ?
- Solution 1: toe-in stereo
  - render two images with slightly offset view points (about 10cm)
  - let main view directions coincide in focus plane
  - image plane and screen not perfectly aligned → image error



Stereo Rendering

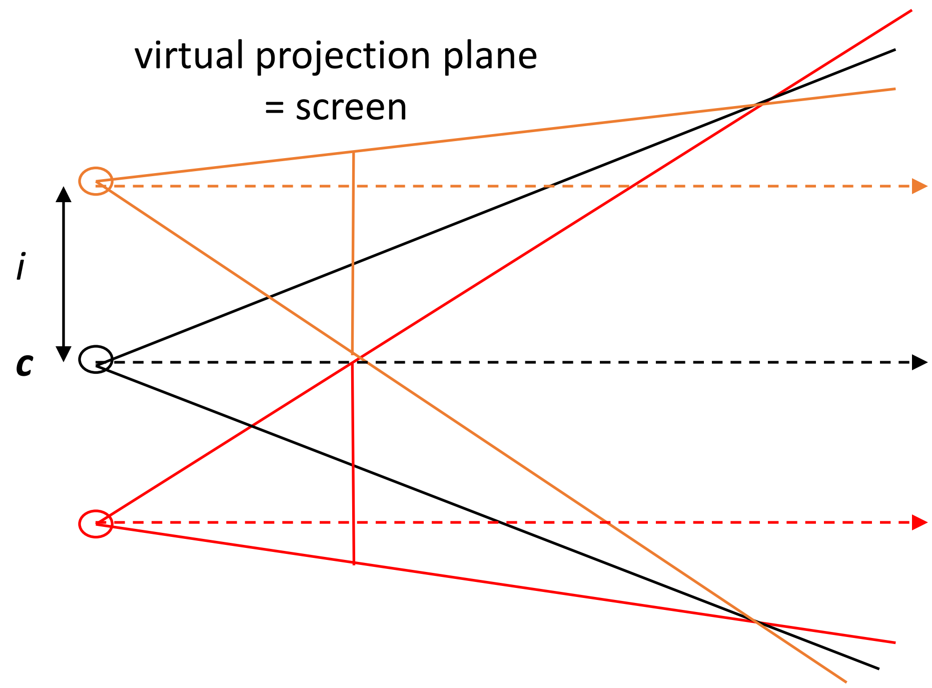
- toe-in stereo:
  - does not describe geometry correctly: vertical parallax
  - where is focus plane? We might want to adapt it…



  - see also „Good Stereo vs. Bad Stereo“

Stereo Rendering

- correct solution: skewed view frusta
  - view frusta are skewed inwards
  - proper stereo rendering only in overlap of view frusta

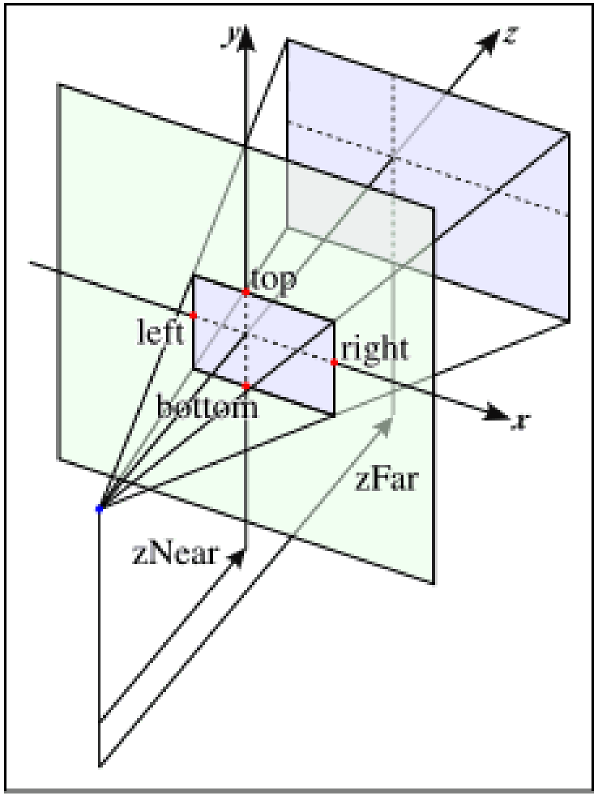


Stereo Rendering

- So we need skewed view frusta

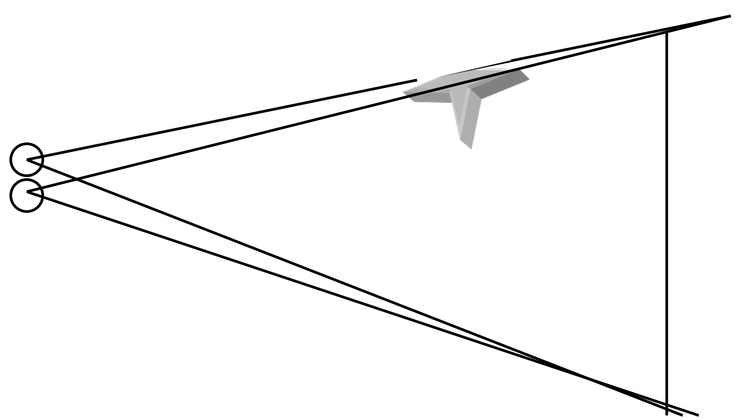
```
glFrustum(left,right,bottom,top,zNear,zFar)
```

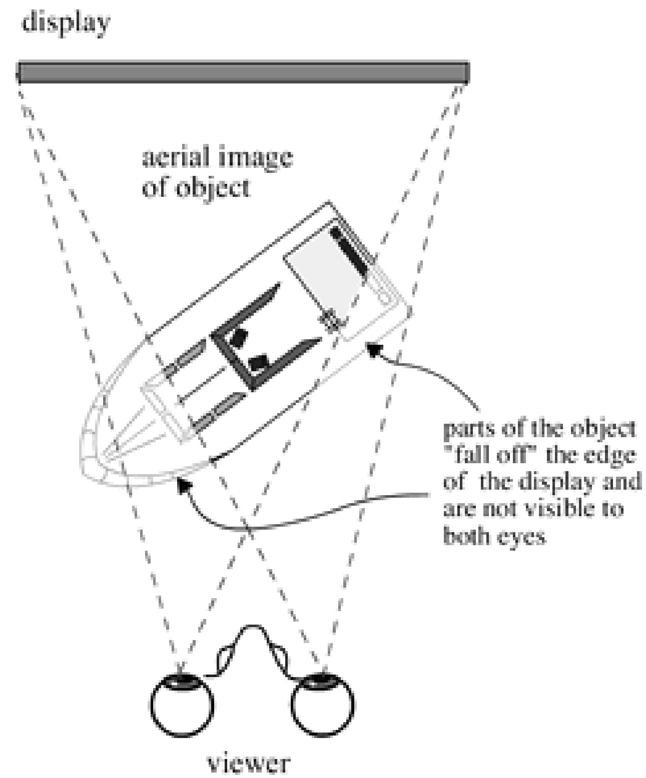
- Problem:
  frustum parameters depend on geometry of HMD and on lens
- need to be calibrated
- need to be considered together with lens undistortion
- usually done by SDK…



Stereo Rendering

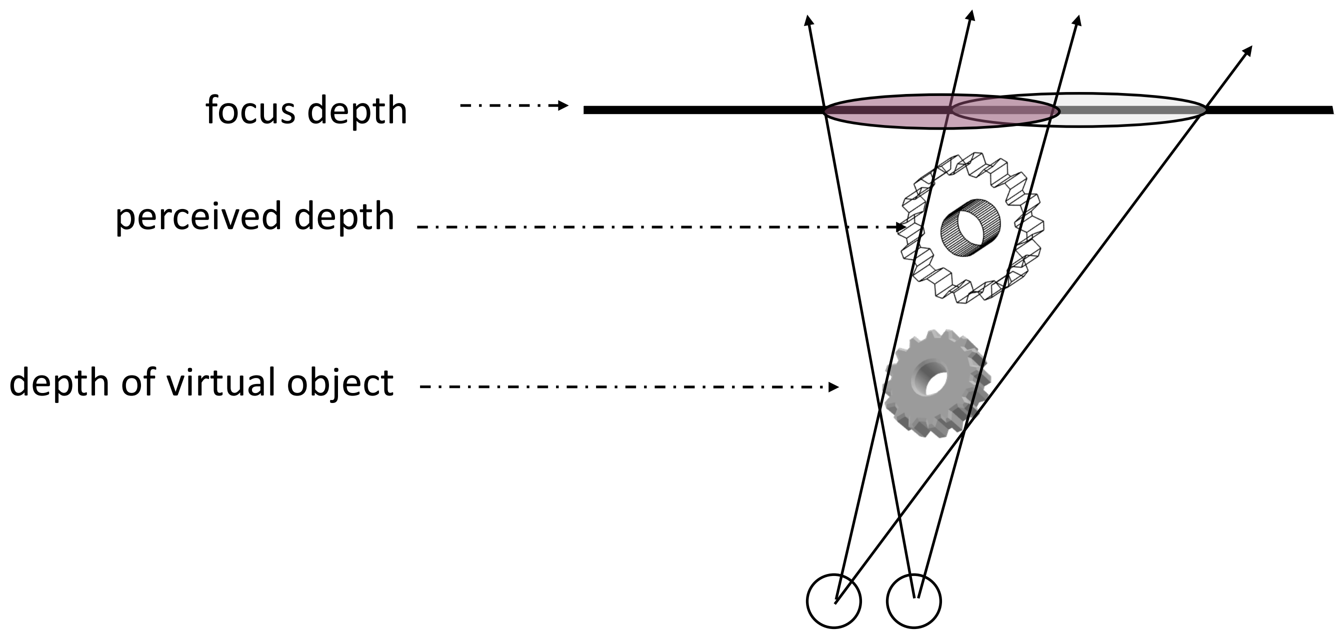
- Problem 1: Stereo violation
  - for close objects: parts of an object are seen by only one eye





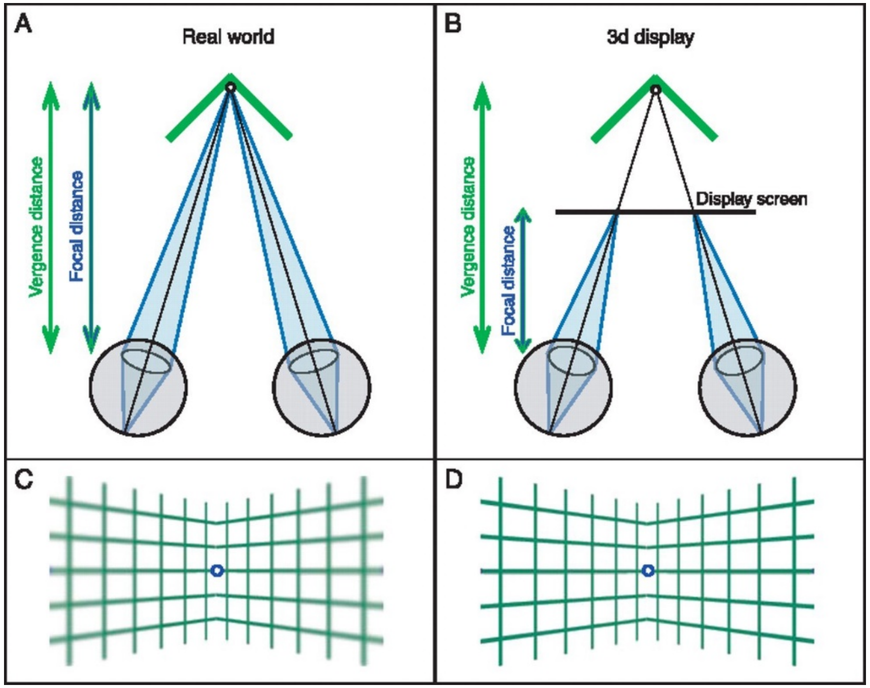
Stereo Rendering

- Problem 2: depth from parallax and focus depth can be different
  → **Vergence accommodation conflict**
- Big problem for HMDs, difficult to solve technically, no real solution yet



Stereo Rendering

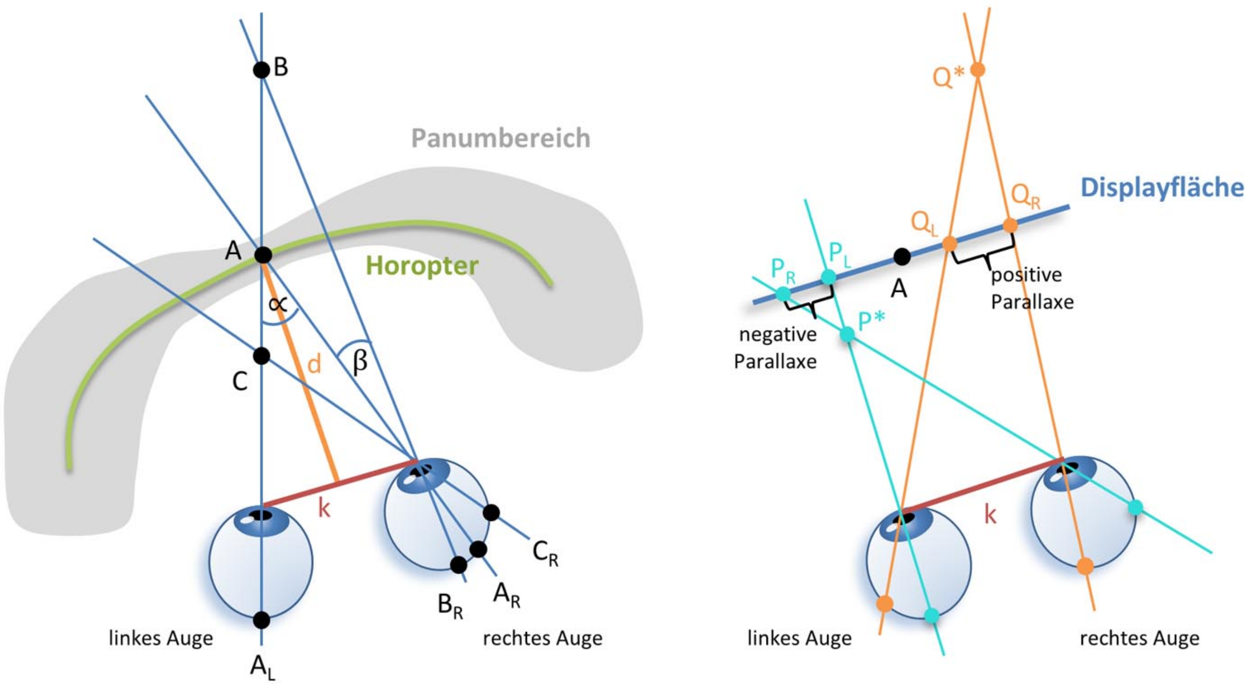
- Vergence accommodation conflict



https://medium.com/vrinflux-dot-com/vergence-accommodation-conflict-is-a-bitch-here-s-how-to-design-around-it-87dab1a7d9ba

Stereo Rendering

- **Horopter**: regions, where focus depth and zero parallax coincide → "perfect" stereo
- **panum region**: region where difference is not noticeable
- Try to keep scene within panum region → not always possible



aus: Dörner/Broll/Grimm/Jung: Virtual und Augmented Reality (VR/AR) eXamen.press, Springer-Verlag, 2013

Abb. 2.2 a) Stereopsis b) Manipulation der Stereopsis mit einem Stereodisplay

Augmented Reality Headsets

- display is „transparent“
- image is blended in using semi-transparent mirrors
- can be used to augment the real environment
  → augmented reality

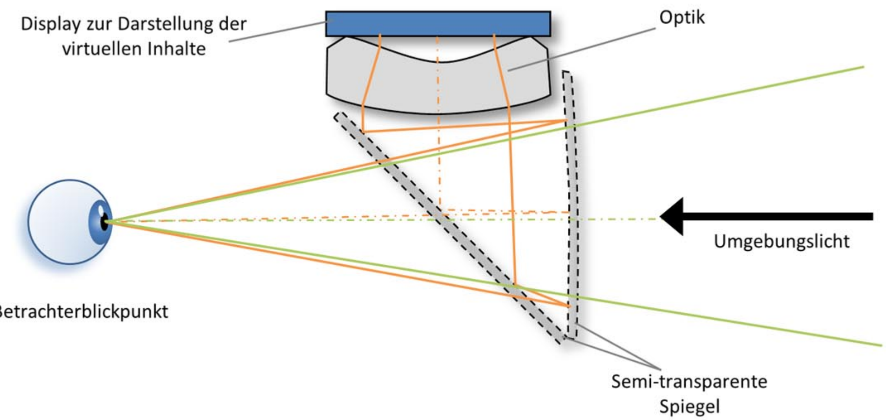
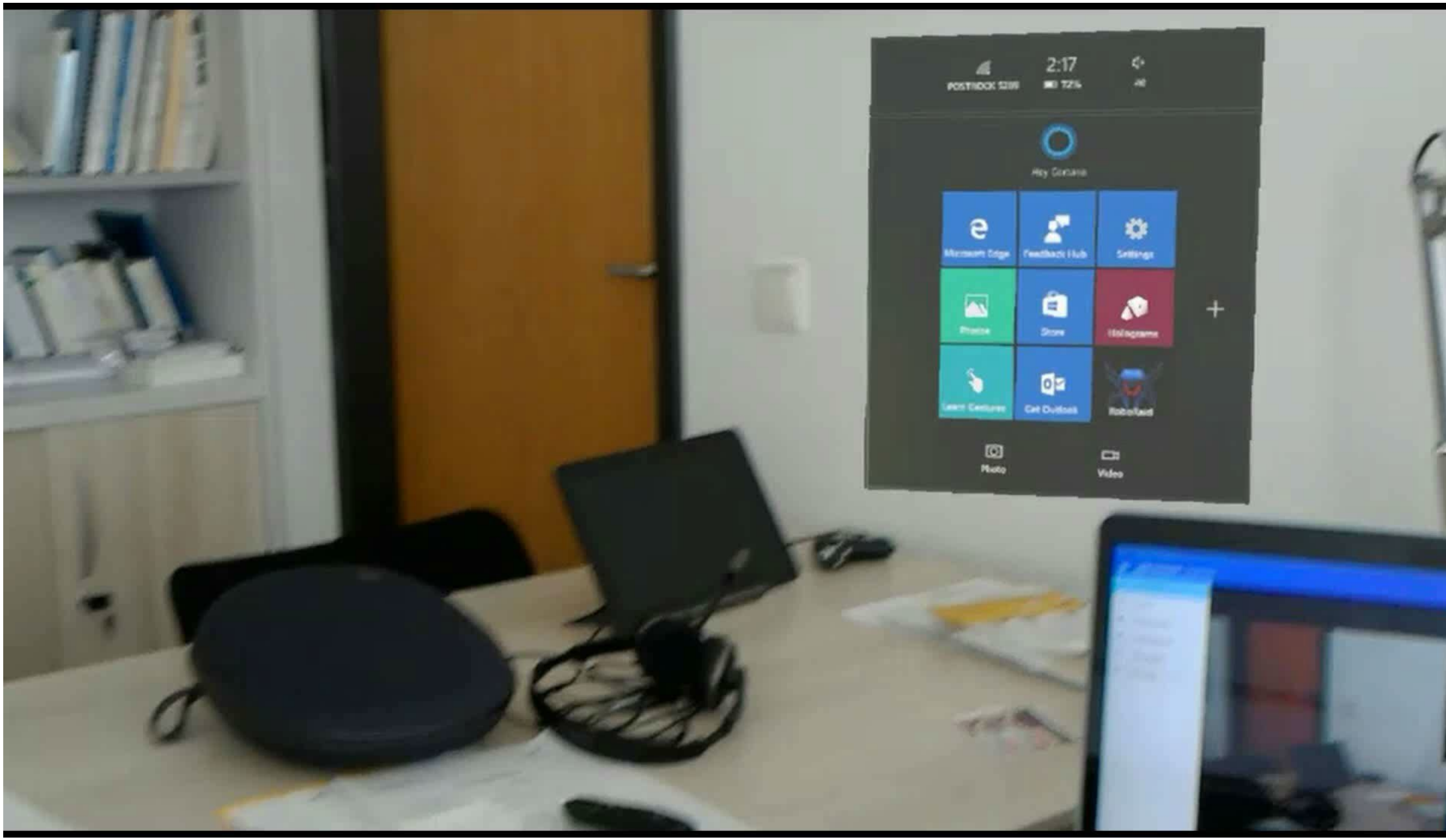


Abb. 8.30 Funktionsweise optischer See-Through-Displays mit semi-transparenten Spiegeln

aus: Dörner/Broll/Grimm/Jung
Virtual und Augmented Reality (VR/AR)
eXamen.press, Springer-Verlag, 2013

Augmented Reality Headsets



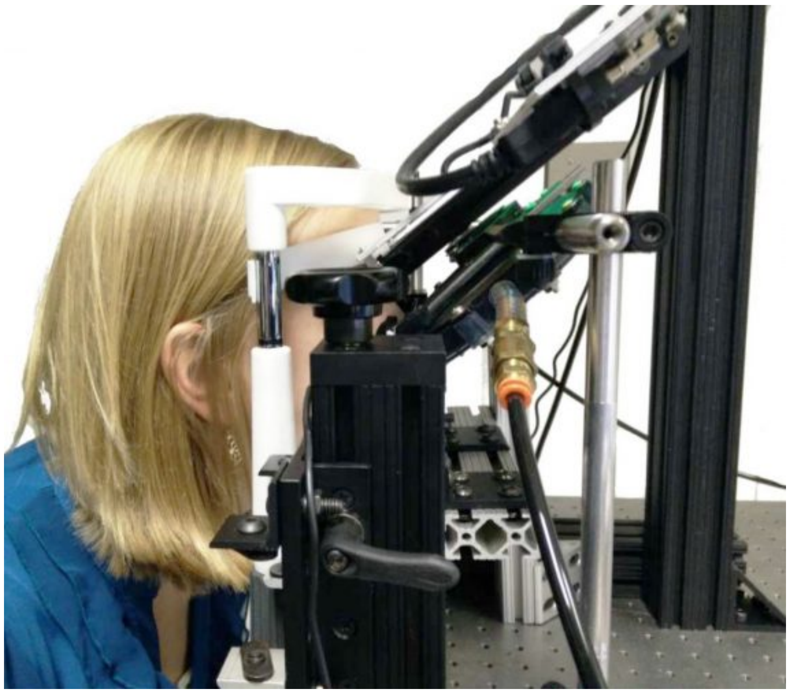
Augmented Reality Headsets

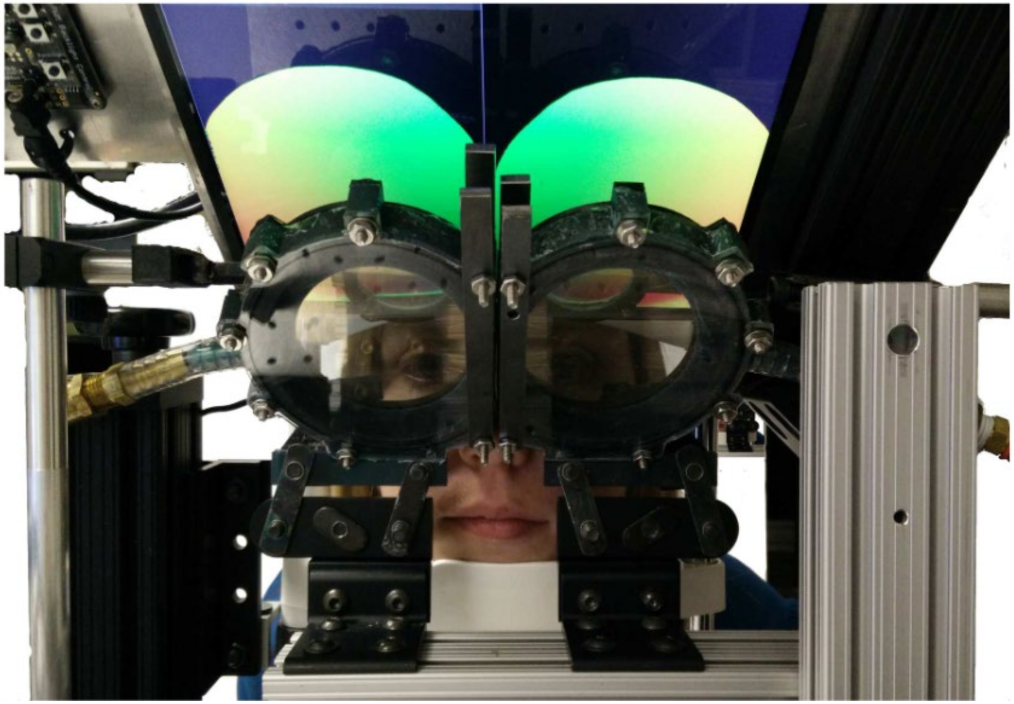
Challenges:

- virtual image only blended over
  → real world always visible behind
  → no opaque virtual objects
- tracking very important, otherwise virtual and real content don't fit
- virtual content and real world should fit together:
  - similar lighting
  - virtual objects should be occluded by real world objects
- vergence – accommodation conflict problem even worse:
  - two objects next to each other, one real one virtual, but with different focus
    → if eye accommodates to #1, #2 is out of focus, and vice versa

Vergence-Accomodation Conflict

- Dunn et al., TVCG 2017





Vergence-Accomodation Conflict

- Magic Leap

magicleap.com

HMDs - Challenges

- Tracking:
  - Fast, comfortable, cheap tracking of headset
  - ideally also in larger scale: room, building, outdoor, ...
  - → **next lecture**

- Low Latency:
  - Head movement should immediately result in an adaptation of the image
  - Multiple stages contribute to this latency
    → **next lecture**

- Vergence-Accomocation-Conflict:
  - largely unsolved. First devices available, but still lacking
  - lot of research

Late research results

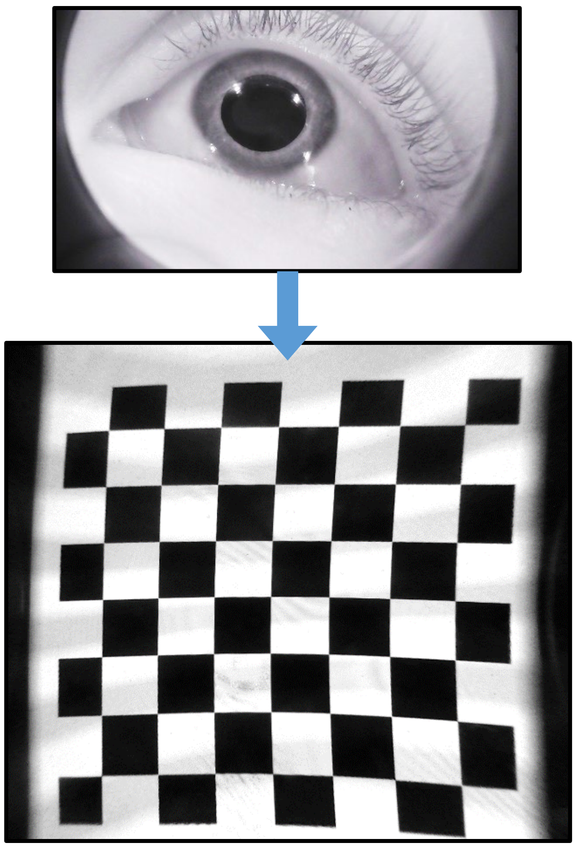
- Martschinke et al. (2019) "Gaze-Dependent Distortion Correction for Thick Lenses in HMDs“ (https://ieeexplore.ieee.org/document/8798107 via university IP)

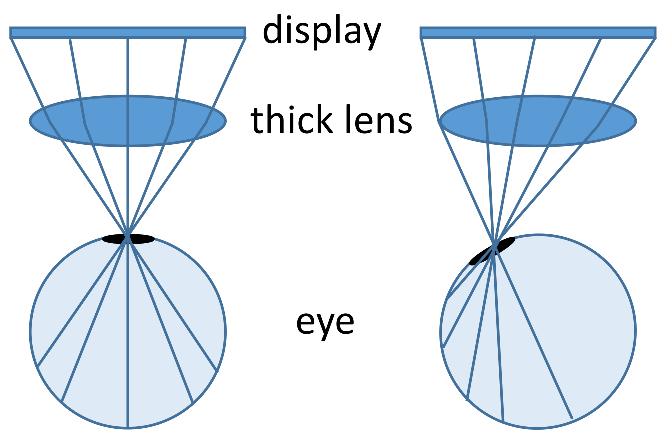
- Fink et al. (2019) “Hybrid Mono-Stereo-Rendering” (https://ieeexplore.ieee.org/document/8798283 via university IP)

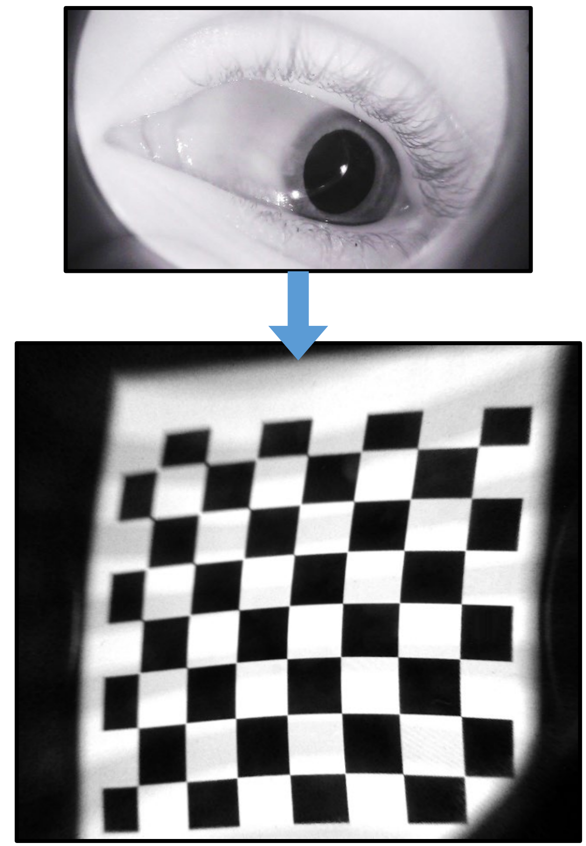
- Franke et al. (2020) “Time-Warped Foveated Rendering for Virtual Reality Headsets” (https://onlinelibrary.wiley.com/doi/10.1111/cgf.14176)

these papers are not relevant for the exam

Gaze-dependent distortion removal



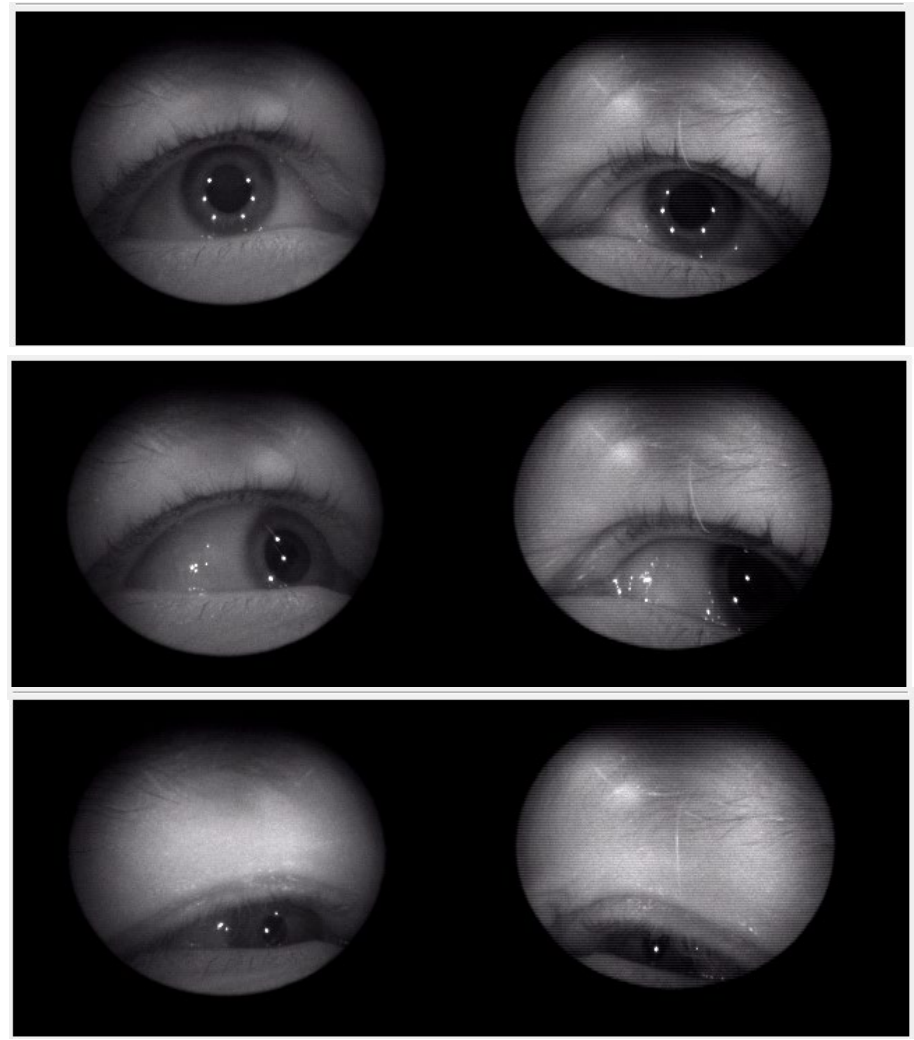




→ breaks immersion?
→ causes motion sickness?

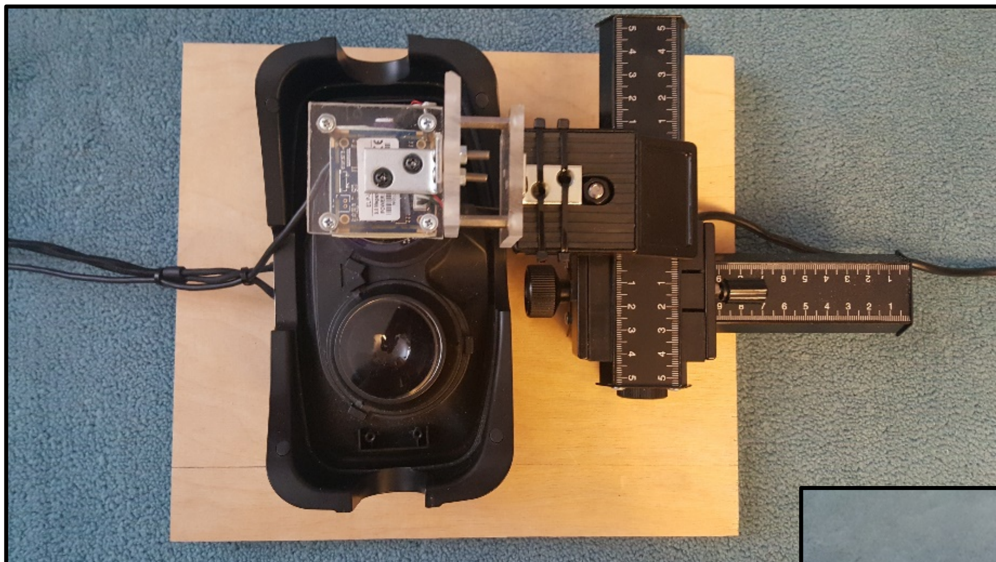
Gaze-dependent distortion removal

- Eye tracking

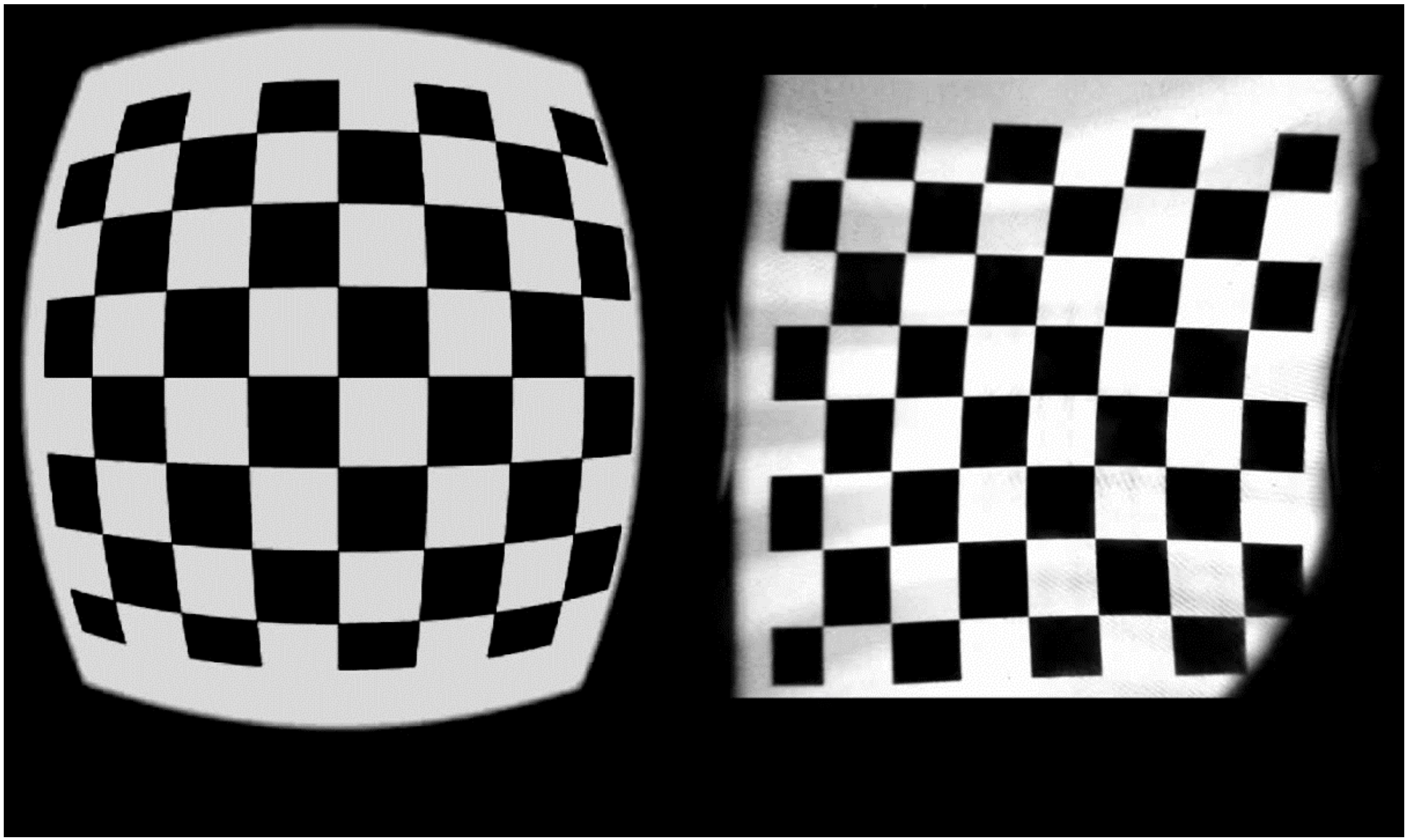


Gaze-dependent distortion removal

- Measuring distortion



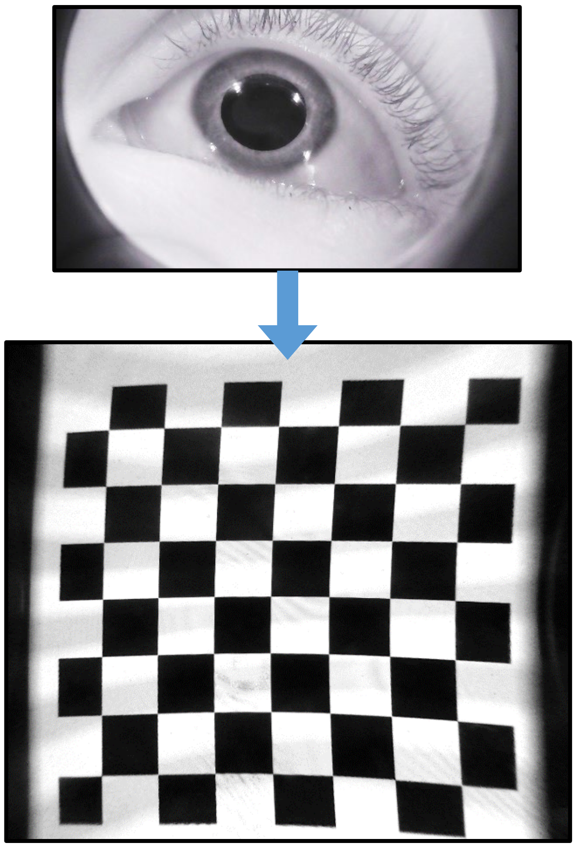
Gaze-dependent distortion removal

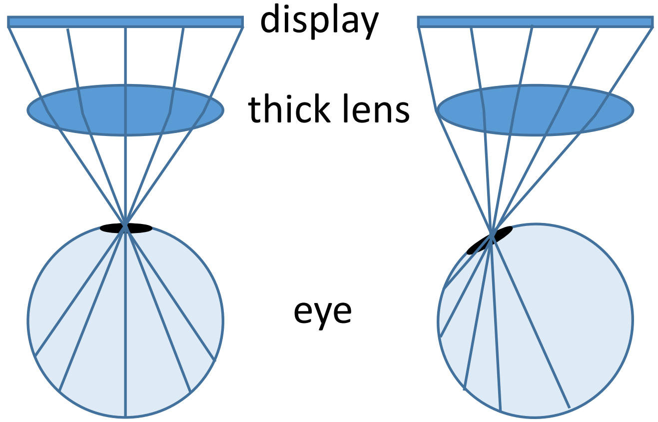


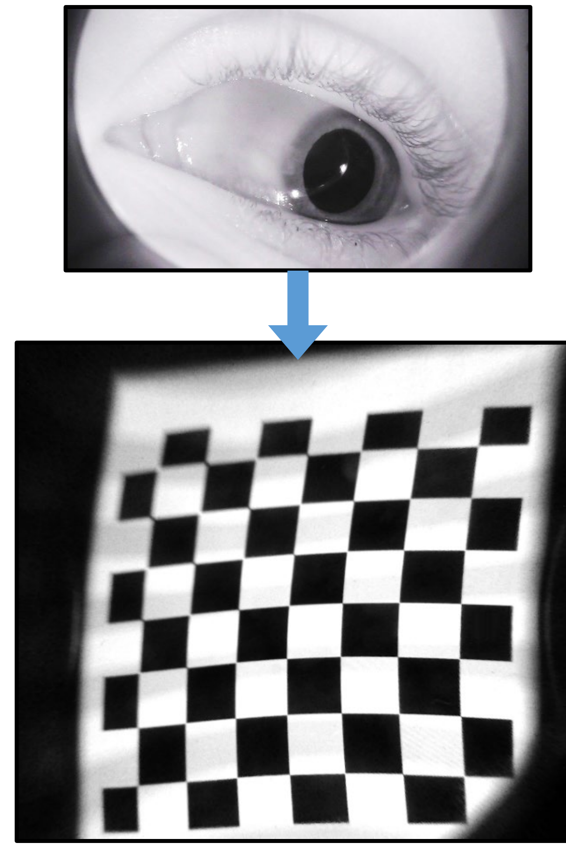
Display

Bild für Beobachter

Gaze-dependent distortion removal





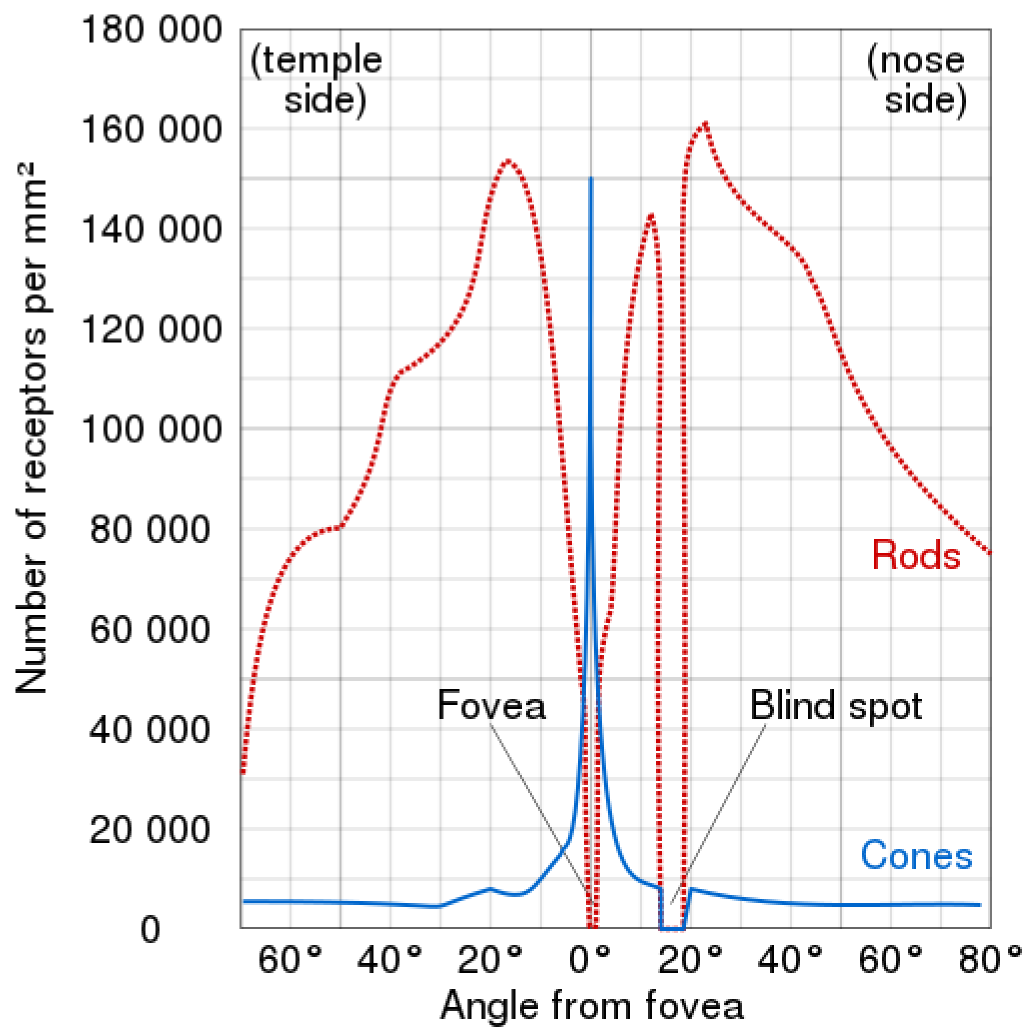


→ breaks immersion?
→ causes motion sickness?

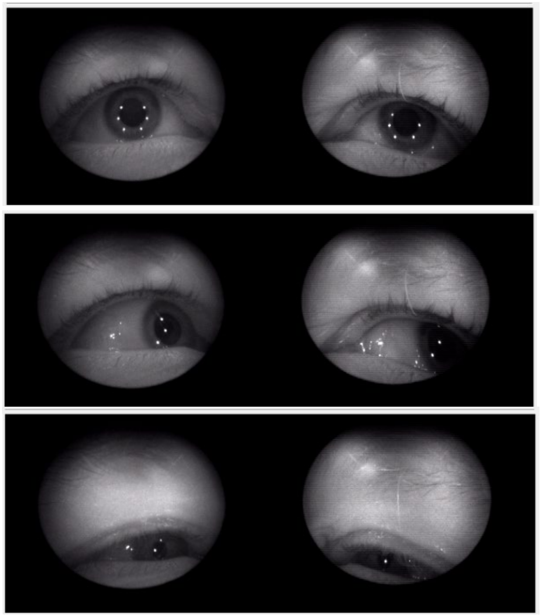
Hybrid Mono-Stereo Rendering

Foveated Rendering

- Fovea: central regions of retina where resolution is maximal
- Periphery (outside fovea): much lower resolution, less detail, but more sensitive to temporal changes (movements)



Foveated Rendering



- HMDs with eye tracking: current gaze is known

Foveated Rendering:

- focus on rendering quality in foveal region
- avoid flickering (**aliasing!**) in periphery

Foveated Rendering