Stereoscopic 3D visualization on planar displays

Stefan Seipel

2013-11-15

9.15 -10.00

Omnipresence of stereo 3D

Movies Industry pushing
for modern 3D technology

Consumer electronics
industry pushes 3D TV

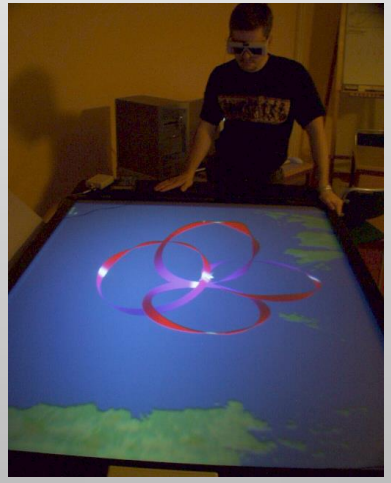
Need your own content?
You need a 3D camera!

You got a 3D camera?
Need a 3D picture frame!

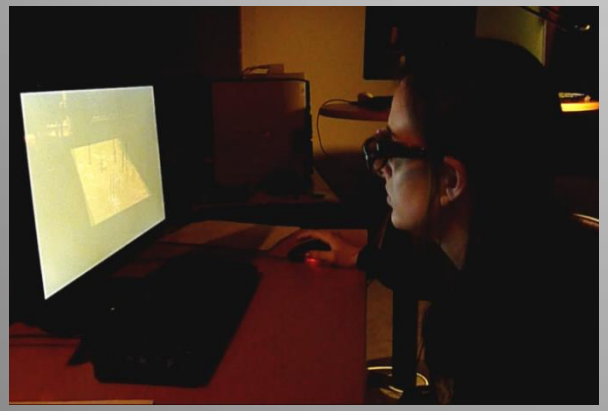
You’ve got all this?
Buy a 3D mobile phone!

Early adopters of modern stereoscopy

3D Table at HiG
GraphiX Center (2002)



G.R.A.P.H. Table
Uppsala Univ. (1999)



3D GIS, GraphiX Center, 2011

ComputerSweden

SVERIGES IT-TIDNING

ONSDAGEN DEN 20 OKTOBER 2004 NR 105

Mobilt missnöje

Handdatorer IT-chefens största bekymmer

Militären överlever med 3d

Utnyttja systemet bättre

Fel satsa på sökmotor

Linux tar tåten

Navision à la Windows

IT-säkerhet klättrar

Het höst för pc-säljare

SJ söker ny partner

"I skuggan av Microsoft pågår det ständigt inbördes affärer"

Computer Sweden,
20th of October 2004

How does it work?

- Human stereoscopic vision
- Some available techniques

How can we get the best out of it?

- Learning from other fields helps creating great content

The perceptual basis for stereoscopic displays

Viewing in the real 3D world

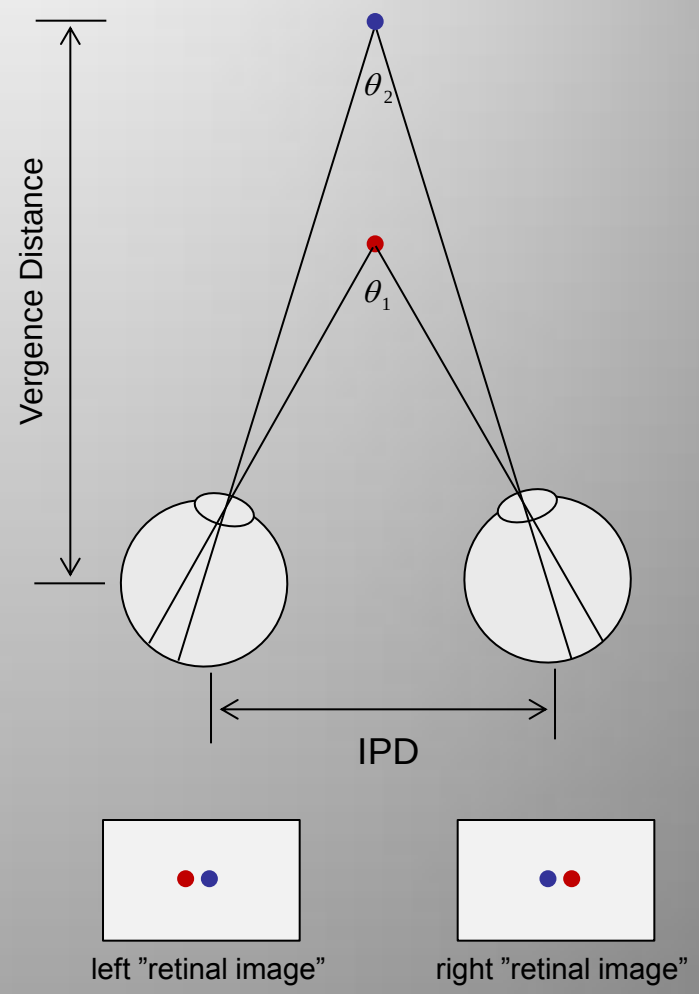
When converging on an object at some certain distance, a point at some different distance in the scene will appear on the retina with some **horizontal disparity**.

Stereopsis :merging two retinal images into one 3D image by **evaluating horizontal disparities**.

Retinal disparity (retinal image)

vs.

Stereo parallax (stereographic image)



Perceptual issues of stereoscopic displays

Viewing virtual 3D images on a 2D screen is different.

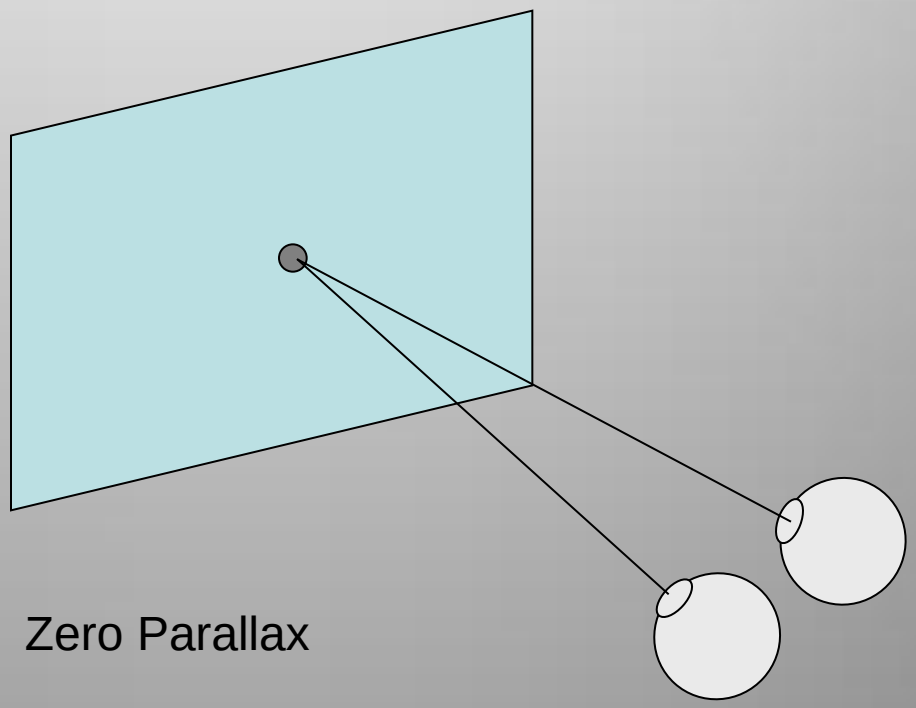
All pixels have actually the same distance to viewer!

The perceptual basis for stereoscopic displays

Parallax: Viewing on plano-stereoscopic displays

Parallax is lateral displacement of homologous points on the planar display.

Converging at a point on a display surface causes homologous points on display to have zero parallax.



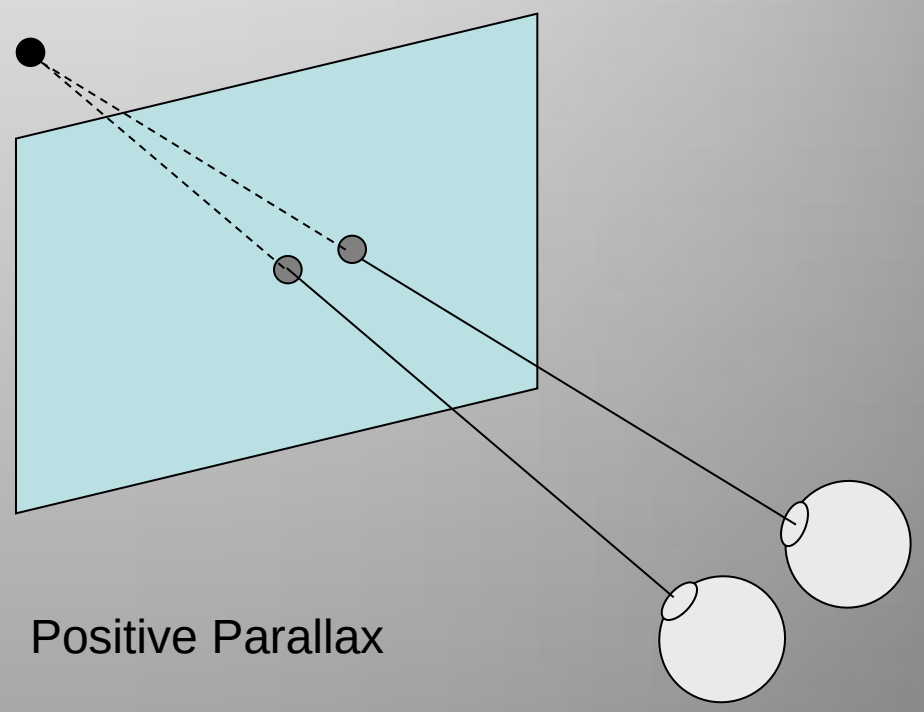
The perceptual basis for stereoscopic displays

Parallax: Viewing on plano-stereoscopic displays

Parallax is lateral displacement of homologous points on the planar display.

Converging at a **virtual point** behind the display surface causes **homologous points** on display to have **positive** parallax.

This point is said to be in **screen space**.



Positive Parallax

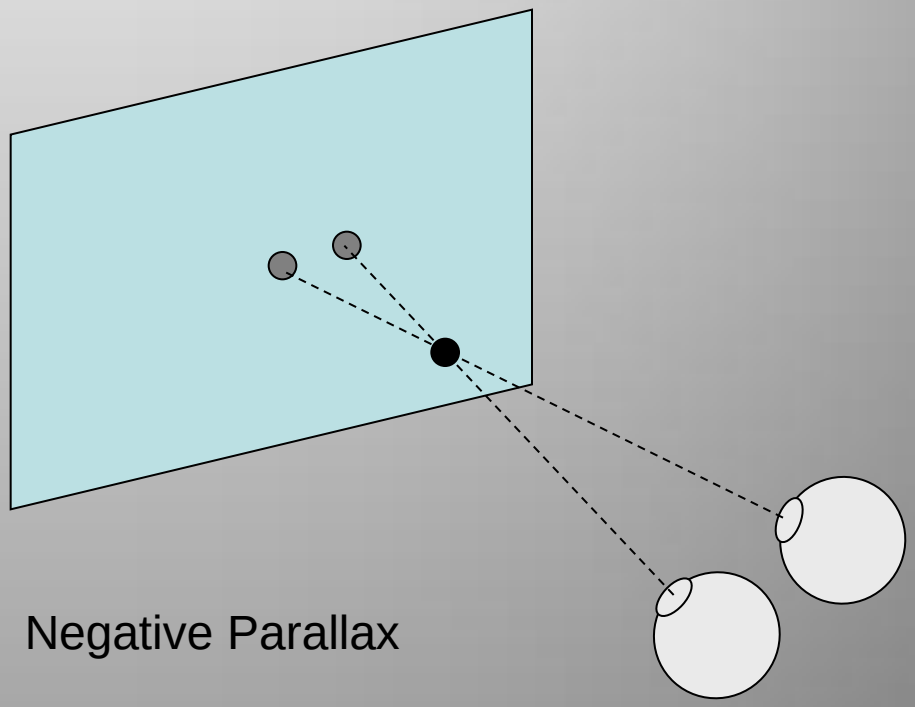
The perceptual basis for stereoscopic displays

Parallax: Viewing on plano-stereoscopic displays

Parallax is lateral displacement of homologous points on the planar display.

Converging at a **virtual point** in front of the display Surface causes **homologous points** on display to have **negative** parallax.

This point is said to be in **viewer space**.

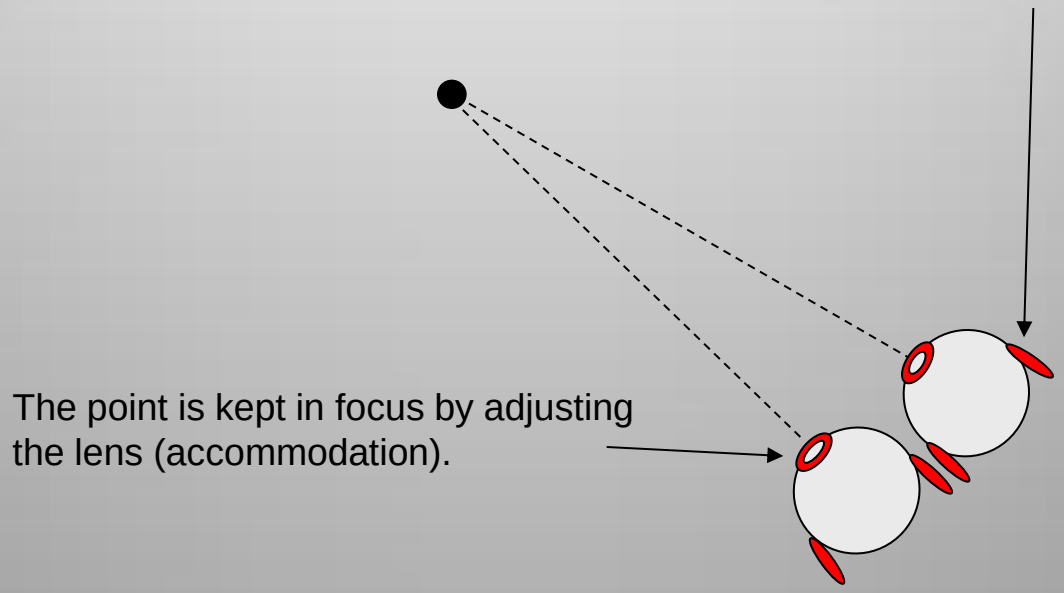


Negative Parallax

The perceptual basis for stereoscopic displays

The relationship between convergence and accomodation

Converging the eyes' axes on a real point in 3D space implies verging the eyes towards that point.
The neuro-muscular functions to control vergence are also evaluated to assess distance.



Under normal natural viewing conditions, accomodation and convergence correspond.
The correspondence of convergence and accomodation is habitual an can be voluntarily put out of function (crossing the eyes).

One dilemma of stereo-graphic 3D images

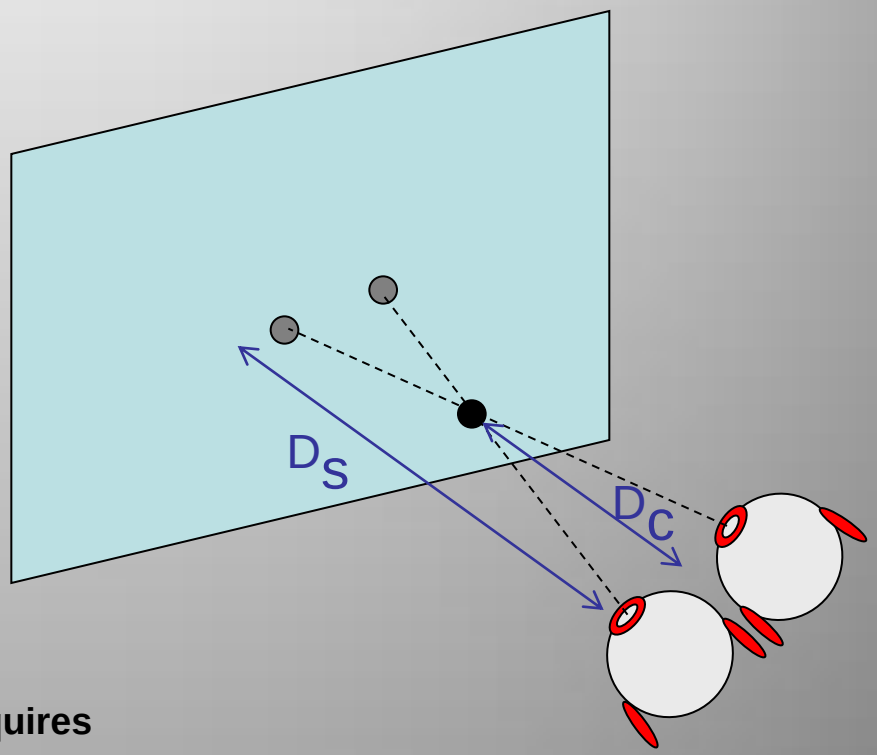
The accommodation – convergence conflict

Keeping homologous points on screen in focus requires accommodation at screen distance D_S.

Accommodation is a neuromuscular function of the ciliari muscles controlling the lens.

To support fusion of homologous points, the eyes must **converge at D_C** – the distance to the virtual point.

Convergence is a occulo-motoric function.



In natural viewing of the real world accommodation and convergence are **coupled processes**.

But:
Seeing plano-stereoscopic 3D images correctly, requires accommodation and convergence on different distances.

Pixels on screen have ”fixed” distance to viewer

Avoiding AC-Conflict

In practice comfortable stereo graphics is a compromise

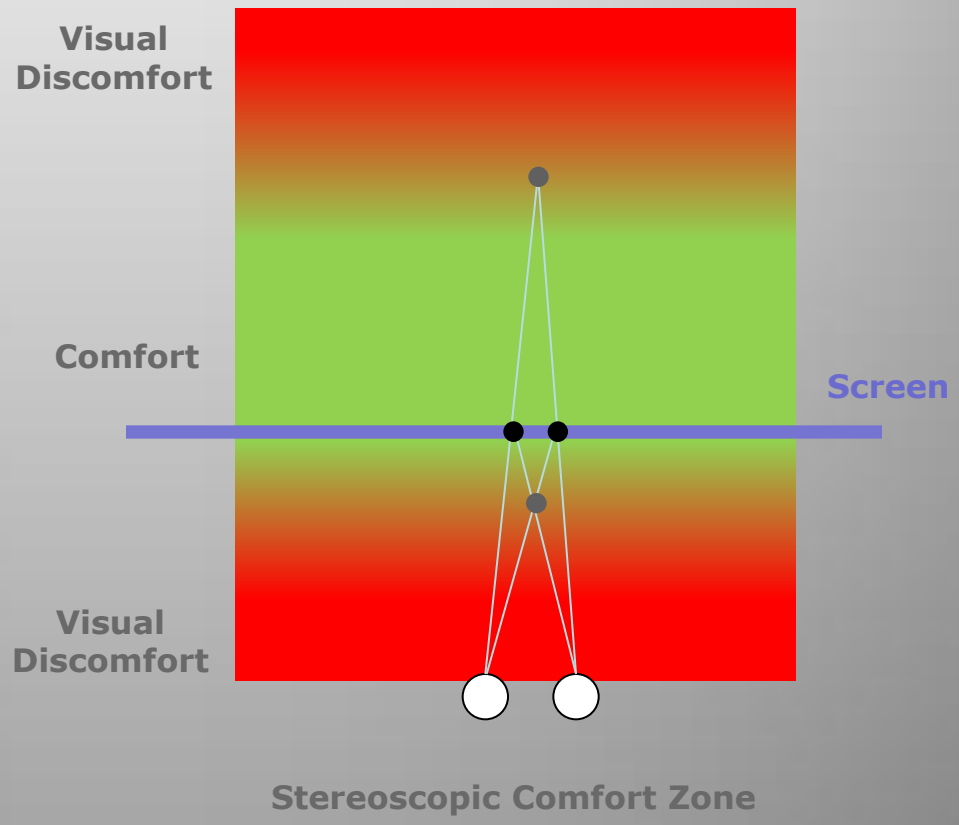
Fixed Variables:

Focal distance to screen

Maximum tolerable parallax

⇒ larger depths in screen space

⇒ limited depths in viewer space



The perceptual basis for stereoscopic displays

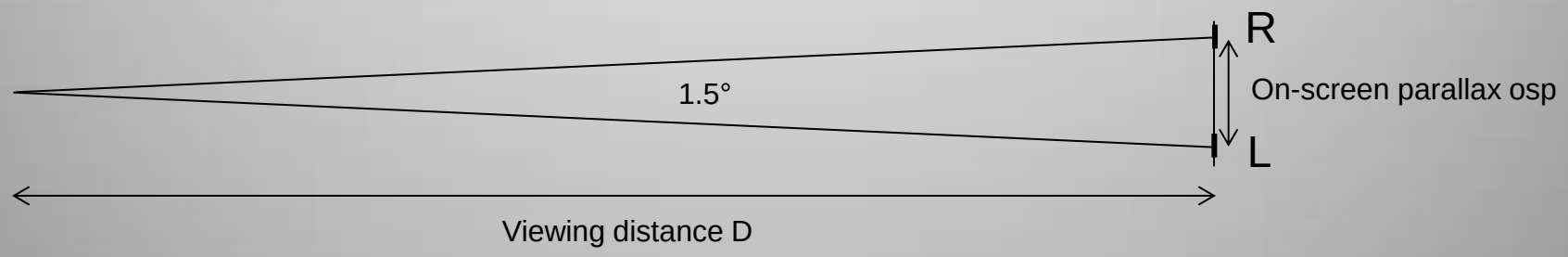
Summary:

- Regions in a stereographic image with high parallax cause retinal disparity, which is one cue for perceiving depth in a scene.
- Stereo parallax and retinal disparity are related
- The brain can fuse binocular retinal images into **one spatial** image (stereopsis)
- Equivalently, there are limits to display parallax. (e.g. 1,5° of visual angel Lipton[1991])
- Viewing non-zero parallax images on plano-stereoscopic displays infringes the accomodation/convergence relationship learned in natural viewing.

Practical issues in stereo graphics

Lipton [1991] – Parallax Control

Don’t exceed parallax values of more than 1.5° visual angle
(rather than using explicit parallax values on screen).



Practical examples for on-screen parallax values:

D [cm]	50	75	100	200	300	400
osp [cm]	1,31	1,96	2,62	5,24	7,86	10,47

Practical issues in stereo graphics

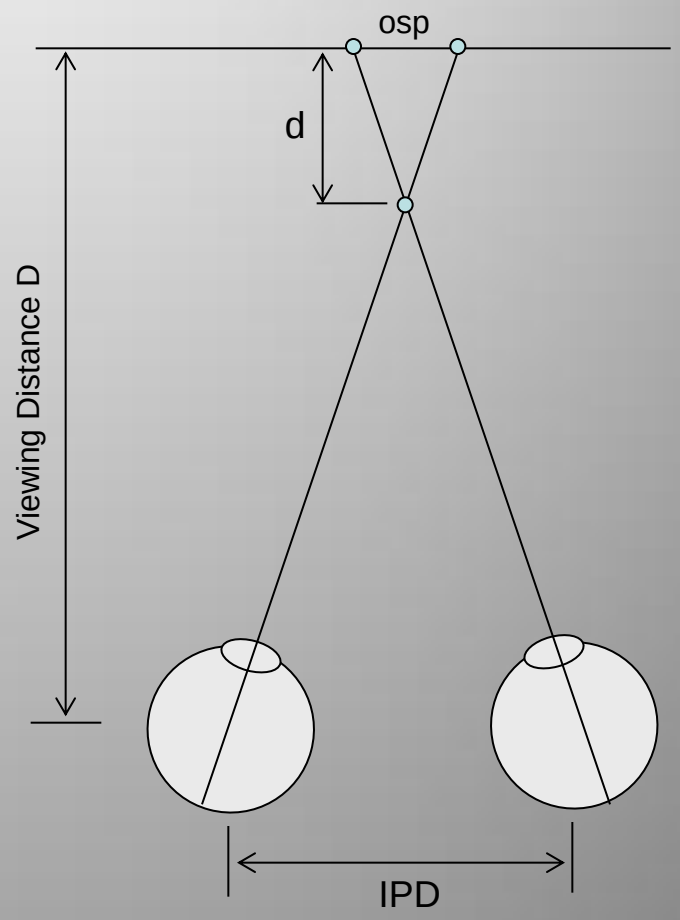
Virtual spatial depth due to observer distance and limited parallax.

Negative parallax situation:
Allowable spatial depth d towards viewer space

$$\frac{osp}{IPD} = \frac{d}{D-d}$$

...

$$d = \frac{D}{\frac{IPD}{osp} + 1}$$



Example for IPD = 6,5 cm

D [cm]	50,00	75,00	100,00	200,00	300,00	400,00
osp [cm]	1,31	1,96	2,62	5,24	7,86	10,47
d	8,38	17,40	28,72	89,24	164,17	246,83
d/D	0,17	0,23	0,29	0,45	0,55	0,62

Practical issues in stereo graphics

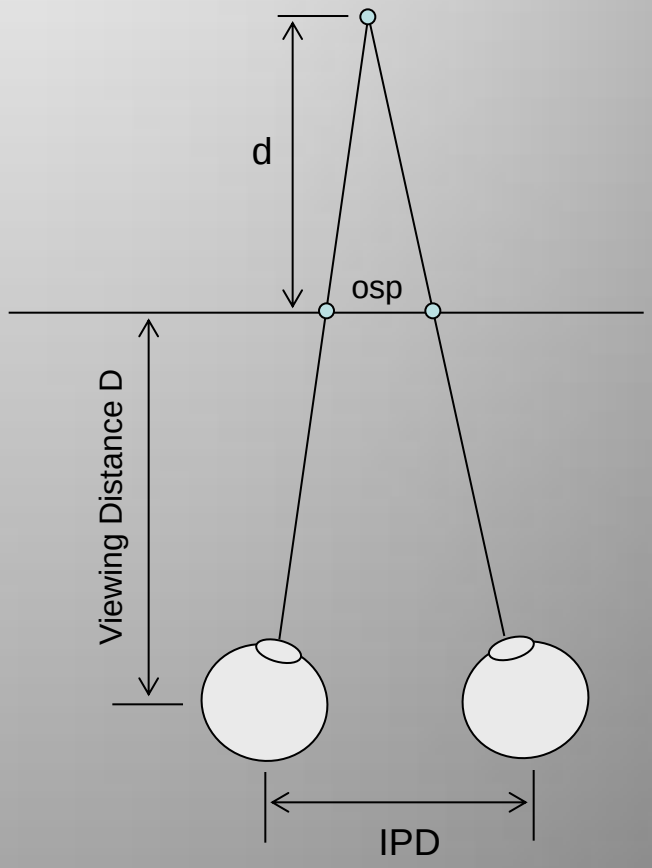
Virtual spatial depth due to observer distance and limited parallax.

Positive parallax situation:
Allowable spatial depth d into screen space

$$\frac{osp}{IPD} = \frac{d}{D+d}$$

...

$$d = \frac{D}{\frac{IPD}{osp} - 1} =$$



Example for IPD = 6,5 cm

D [cm]	50,00	75,00	100,00	200,00	220,00	245,00
osp [cm]	1,31	1,96	2,62	5,24	5,76	6,42
d	12,61	32,47	67,47	829,45	1714,79	18612,49
d/D	0,25	0,43	0,67	4,15	7,79	75,97

Practical issues in stereo graphics

Remember:

Values sometimes seen as a rule of thumb/best practice in specific situations. (Lipton's seem to work for many VR applications)

Values do not generally apply for all viewing conditions
(Desktop Viewing, Head-Mounted Displays, Cinema have different convergence distances)

Once you know the basics, composing stereoscopic images is an art, not a science. (Lenny Lipton,1991)

Practical issues in stereo graphics

Does my desktop 3D application also work in the VR theatre?

Many visualization applications are parameterized to work in stereo on desktop systems.

Example:

17” screen approx. 34 cm wide.
Application creates maximum parallax of 2 cm, which is comfortable when viewed from approx. 70-80 cm distance.

You bring your application to the computer in the VR theatre, without altering viewing parameters.
The stereo-picture is now blown up to 3 meters horizontal size i.e. 850%.

Maximum parallax on the projection screen is now 17,14 cm.

According to Lipton’s recommendation for parallax of max. 1.5°, the observer should be seated at least 6,6 meters from the screen!

Note: Viewing parameters for stereo-viewing do NOT scale linearly!!!

A general problem in computer graphics

Viewing metaphor in use is not suitable for VR and stereo 3D!

Traditionally the field of computer graphics has adopted the *metaphor* of a synthetic pin-hole camera

Projection math is then

* based on central perspective projections

* adapted to parameters such as horizontal and vertical field of view

… even worse: focal length and aspect ratio

This is manifested in industry standard APIs (e.g. OpenGL/GLUT, ”gluPerspective”)

Standard stereo render mode in vtk is based on central perspective projections

A general problem in computer graphics

What is wrong with a camera model?

1. The **scale level** in recording/synthesizing an image is not related to the spatial relations between observer and display during presentation.

   This is usually not a problem in monoscopic graphics, but in stereo it is!

   (parallax and convergence issues, discussed before)

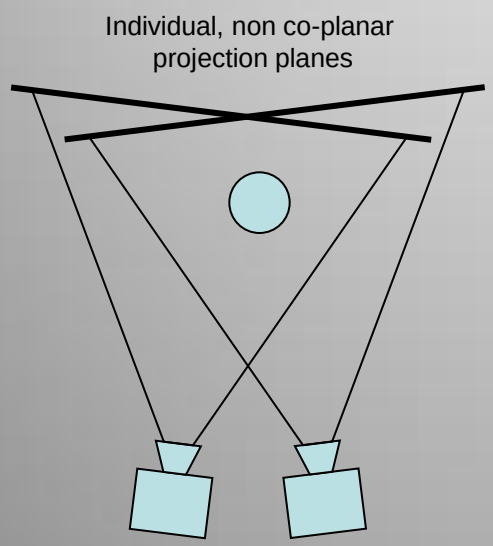
   Example:
   Hyper-stereo and excessive parallax when photographing a frog from 30 cm distance with a IPD of 6 cm and display on a 4m x 3m screen.

   What focal length is applicable for visualizing atoms and molecules?

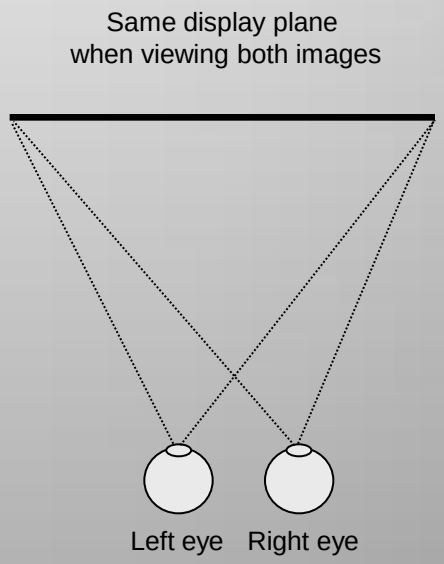
A general problem in computer graphics

What is wrong with a synthetic camera model?

2. Binocular viewing of (projected) plano-stereoscopic displays requires off-axis projected images



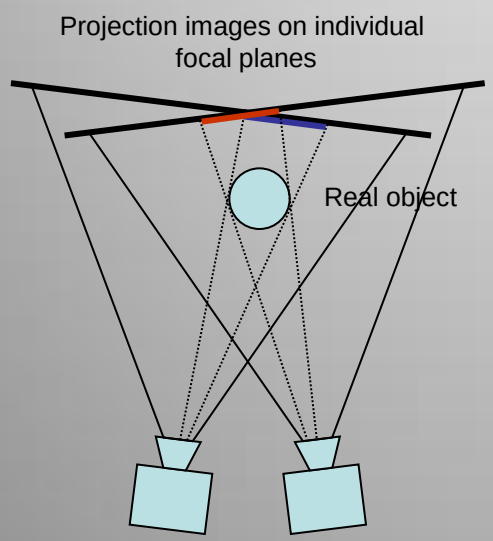
typical setup for camera based stereo



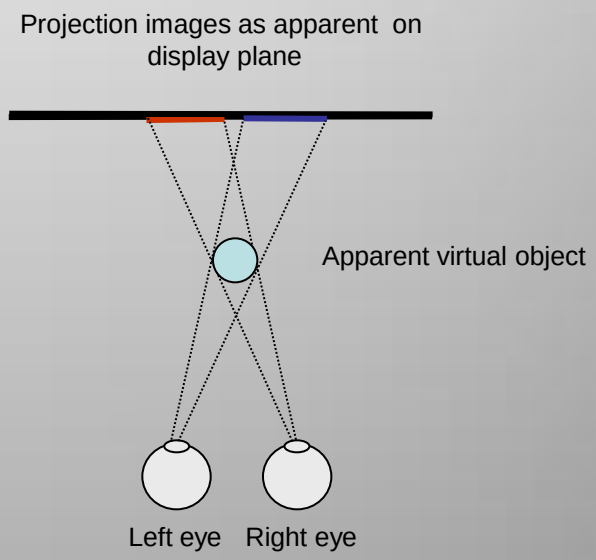
A general problem in computer graphics

What is wrong with a synthetic camera model?

2. Binocular viewing of (projected) plano-stereoscopic displays requires off-axis projected images



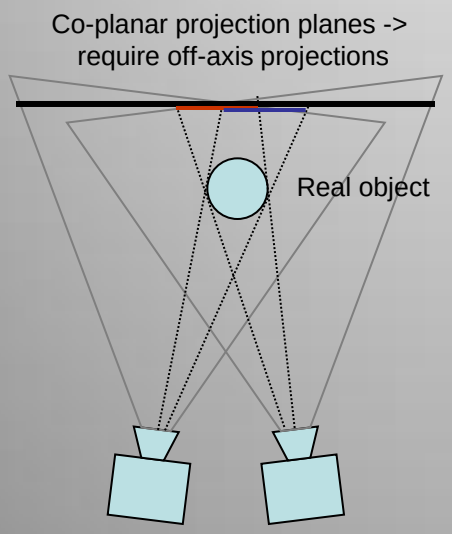
typical setup for camera based stereo



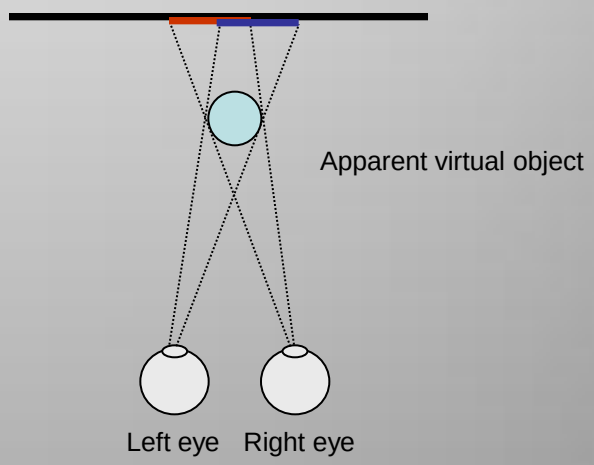
A general problem in computer graphics

What is wrong with a synthetic camera model?

2. Binocular viewing of (projected) plano-stereoscopic displays requires off-axis projected images



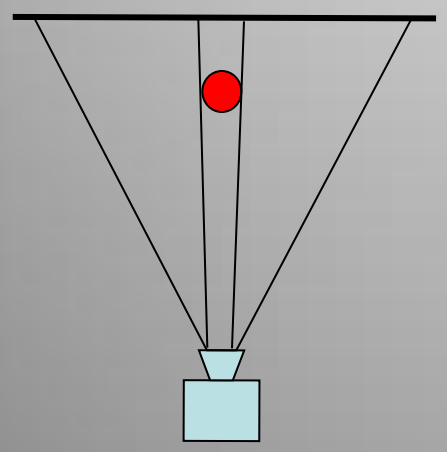
typical setup for camera based stereo



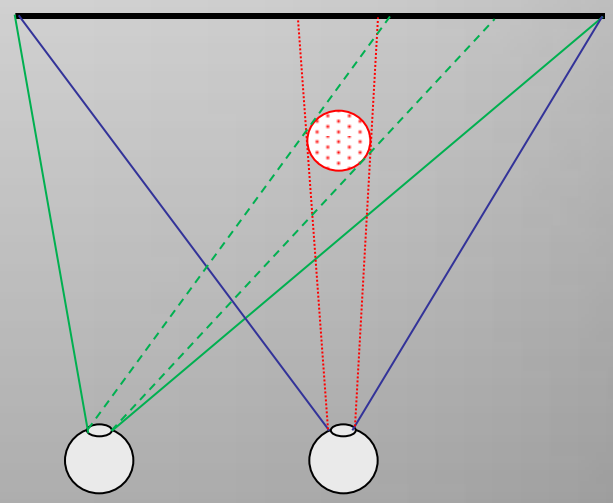
A general problem in computer graphics

What is wrong with a camera model?

3. A camera model with central perspective projection assumes the viewer to be located in front of the center of the image



Single view with centric position



”Incorrect” viewing position ”Correct” viewing position

What can we do about it?

Presentation of 3D objects on planar 2D displays requires generally an arbitrary relationship between the projection (=presentation) plane and the center of projection (=observer´s eye).

This viewing metaphor is called ”Window-on-world”, short WoW or “Fish-Tank Virtual Reality”

A WoW assumes generally an off-axis projection

The result of off-axis projection is an anamorphic image which ,viewed from the CoP, appears geometrically undistorted.

Anamorphic images are not only compelling in computer graphics…

Works of the famous artist Julian Beever

http://users.skynet.be/J.Beever/

Works of the famous artist Julian Beever

This is the center of projection

Here is what you see from the center of projection

http://www.julianbeever.net/

How can I accomplish good off-axis projections?

A metrically correct Window-on-world projection requires

a) the relation between window and observer position to be modeled at the same level of scale (same WCS)

b) requires a parameterization of an appropriate off-axis projection matrix

c) zooming of objects in the scene is accomplished through manipulation of object parameters rather than viewing parameters

A) and b) usually require that you implement the projection pipeline into your Vis-App yourself to be sure that the job gets done right.

How to parameterize an off-axis projection matrix?

Using built-in OpenGL frustum function

```
glFrustum(left,right,bottom,top,near,far);
```

(observer at origo, facing towards –z, near defines proj. plane)

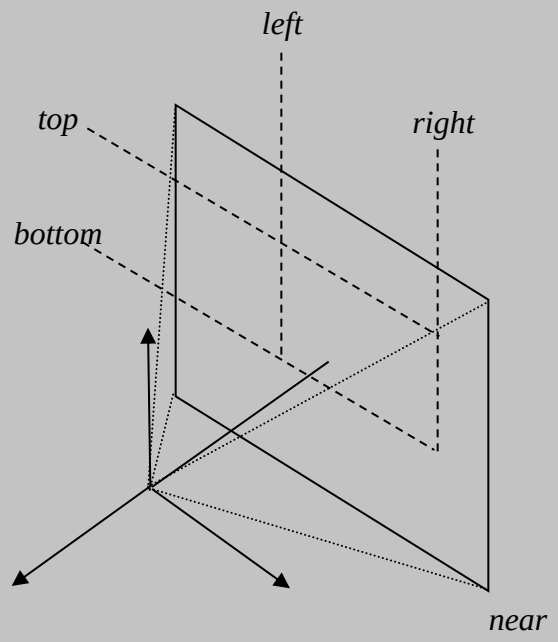
$$P = \begin{bmatrix} \frac{2 \cdot near}{right - left} & 0 & A & 0 \\ 0 & \frac{2 \cdot near}{top - bottom} & B & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$A = \frac{right + left}{right - left}$$

$$B = \frac{top + bottom}{top - bottom}$$

$$C = -\frac{far + near}{far - near}$$

$$D = -\frac{2 \cdot far \cdot near}{far - near}$$



Sabotaging a good 3D illusion

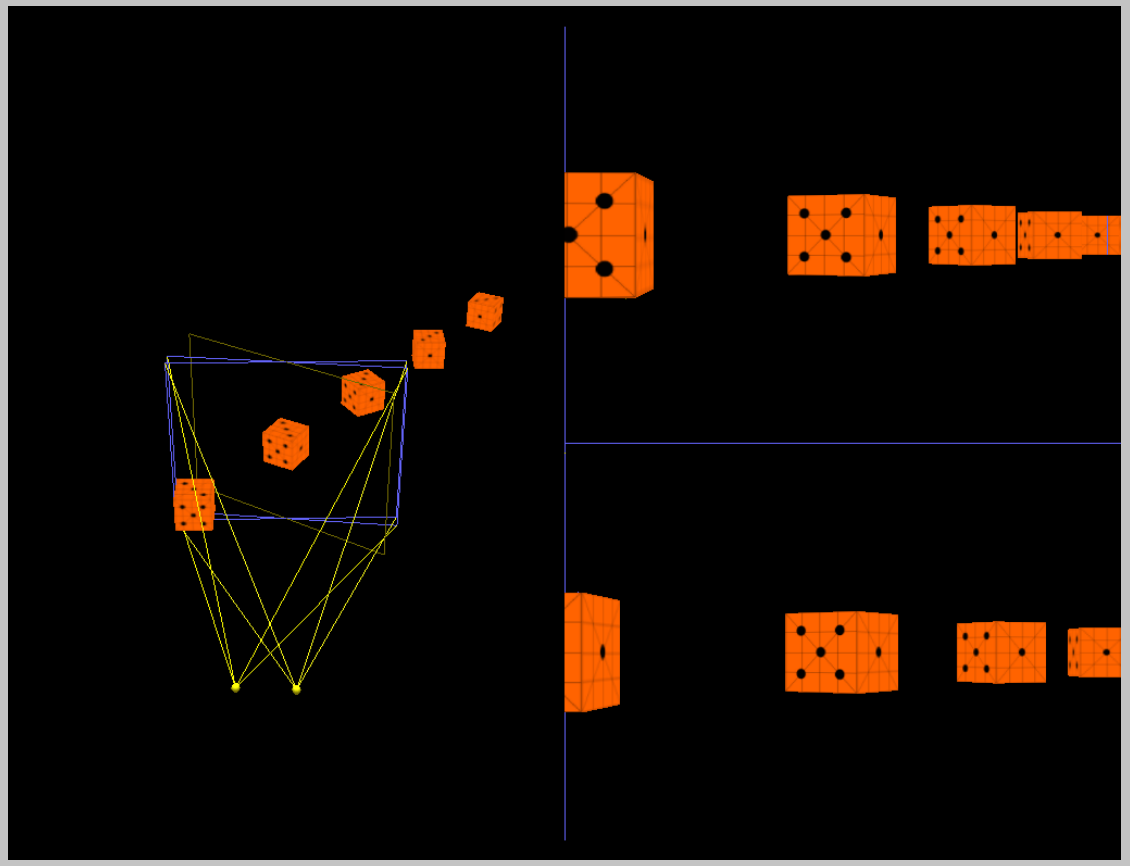
In-proper interaction between 3D content and screen surround destroys illusion:

Virtual 3D object intersects with screen borders.

Contradictious 3D cues

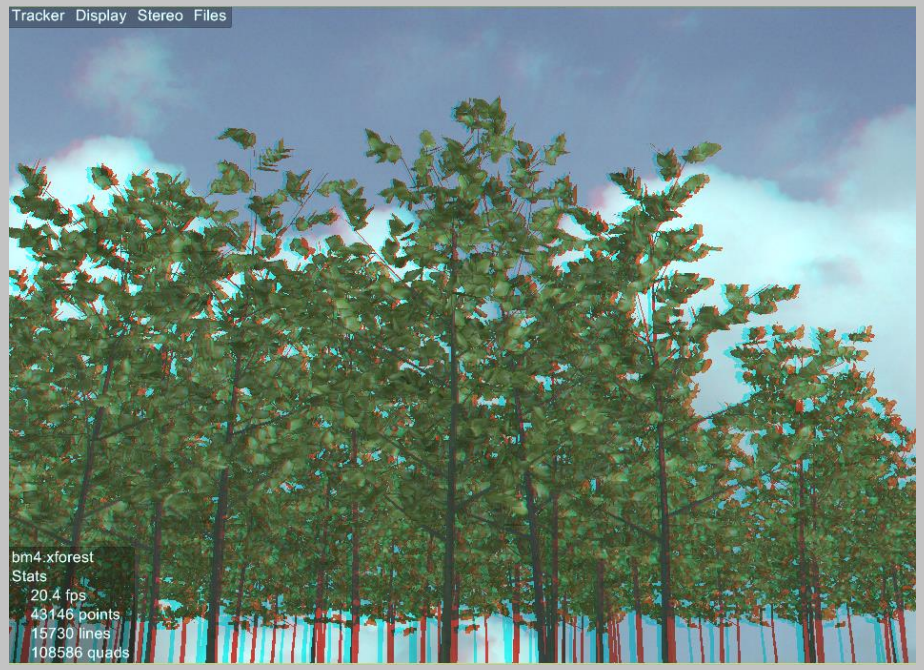
Stereo-disparity says:
“object is in front of screen”

Object occlusion by surround says:
“object is behind the screen”



Sabotaging a good 3D illusion

Giving to much freedom to the user when interacting…



…can cause hyper-parallax and diplopia

Example of dynamic perspective conditions

Alternate observer's percept

First person view with dynamic perspective

Summary: Cues for 3D image generation?

Correct stereo is only one of many spatial cues

- Correct parallax (software)
- Projector adjustment/image alignment (hardware)
- Scaling conditions (sw + hw)
- Interference with surround

Utilize shadows, shading

Exploit motion parallax

- Object rotation creates strong optic flow

Keep noise or small spatial features/structures in e.g. 3D images

- objects with lack of spatial freq. ? -> use textures

Try to use dynamic perspective conditions

Stereo-rendering in vtk

Classes involved in stereo rendering in vtk:

vtkRenderWindow

- contains a flag that indicates stereo-rendering
- responsible for the output of stereo image pairs on screen
- offers a variety of image multiplexing modes

vtkRenderer

- contains a flag that indicates stereo-rendering
- manages rendering several passes
- does not contain any else relevant information for stereo rendering

vtkCamera

- contains actual viewing and projection matrices
- standard built-in support for stereo viewing
- allows ”digging deeper” , internals poorly documented

Stereo-rendering in vtk

Example of methods to control stereo parameters in vtk:

vtk:RenderWindow

::StereoCapableWindowOn()
::StereoRenderOn()
::SetStereoType()
::SetStereoTypeToAnaglyph()
::SetStereoTypeToInterlaced()
::SetStereoTypeToCheckerboard()
::SetStereoTypeToCrystalEyes()

vtkCamera

::stereo	”flag indicating stereo rendering”
::SetEyeAngle(double)	”set separation between eyes in degrees”
::SetUseOffAxisProjection(int)	”off axis viewing frustum is used”
::ComputeOffAxisProjectionFrustum()	
::SetEyeSeparation(double)	”used for OA projections, default 0.06”
…	

Stereo-rendering in vtk

```
# This program demonstrates how VTK can be used
# to render a text
# The user can also interact with the text by
# using the mouse

# load VTK
import vtk

# Create a Text source and set the text
text = vtk.vtkTextSource()
text.SetText("UPPMAX")
text.SetForegroundColor(0.6,0.2,0.2)

# Create a mapper and set the Text source as
# input
textMapper = vtk.vtkPolyDataMapper()
textMapper.SetInputConnection(text.GetOutputPor
t())

# Create an actor and set the mapper as input
textActor = vtk.vtkActor()
textActor.SetMapper(textMapper)

# Create a renderer
ren = vtk.vtkRenderer()

# Assign the actor to the renderer
ren.AddActor(textActor)
```

```
# Create a rendering window
renWin = vtk.vtkRenderWindow()

# Add the renderer to the window
renWin.AddRenderer(ren)

# Set the name of the window (this is optional)
renWin.SetWindowName("Hello World!")

# Enable stereo rendering
renWin.StereoCapableWindowOn()
renWin.SetStereoTypeToInterlaced()
renWin.StereoRenderOn()

# Make sure that we can interact with the
# application
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renWin)

# Initialze and start the application
iren.Initialize()
iren.Start()
```

Related literature and references

Additional reading for this lecture:
http://www.cs.unc.edu/Research/stc/FAQs/Stereo/stereo-handbook.pdf

Research community in stereoscopic/3D displays
http://www.stereoscopic.org/2013/index.html