2

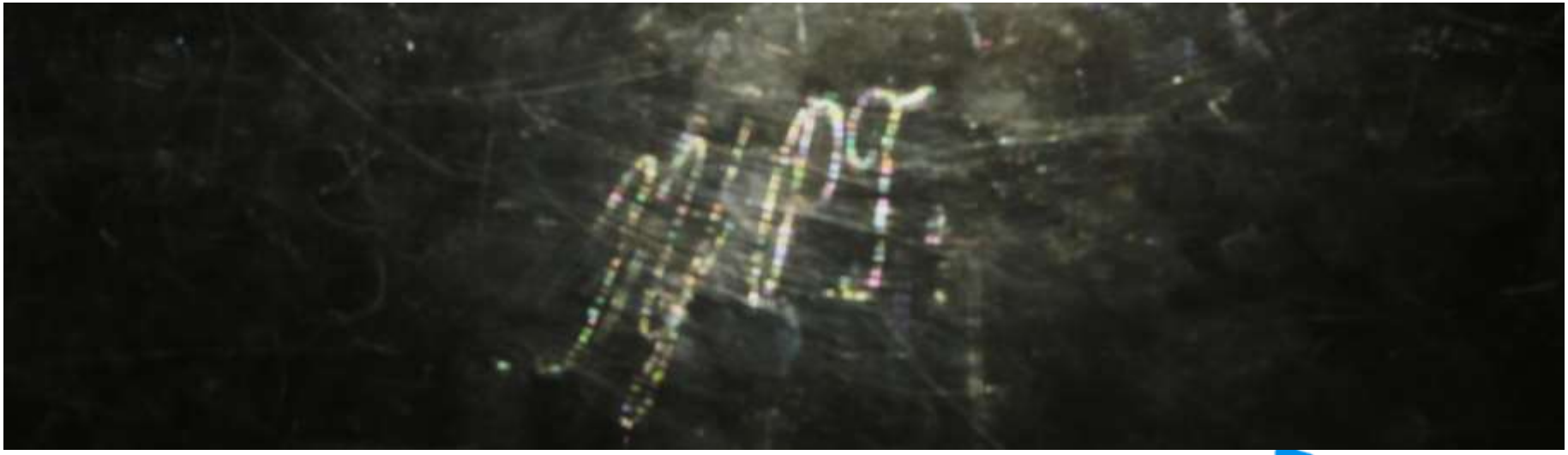
Hologram

Mário Lipovský

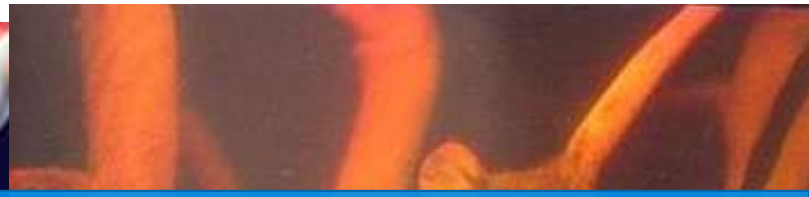
Task

It is argued that a hologram can be hand made by scratching a piece of plastic.

Produce such a 'hologram' with the letters 'IYPT' and investigate how it works.



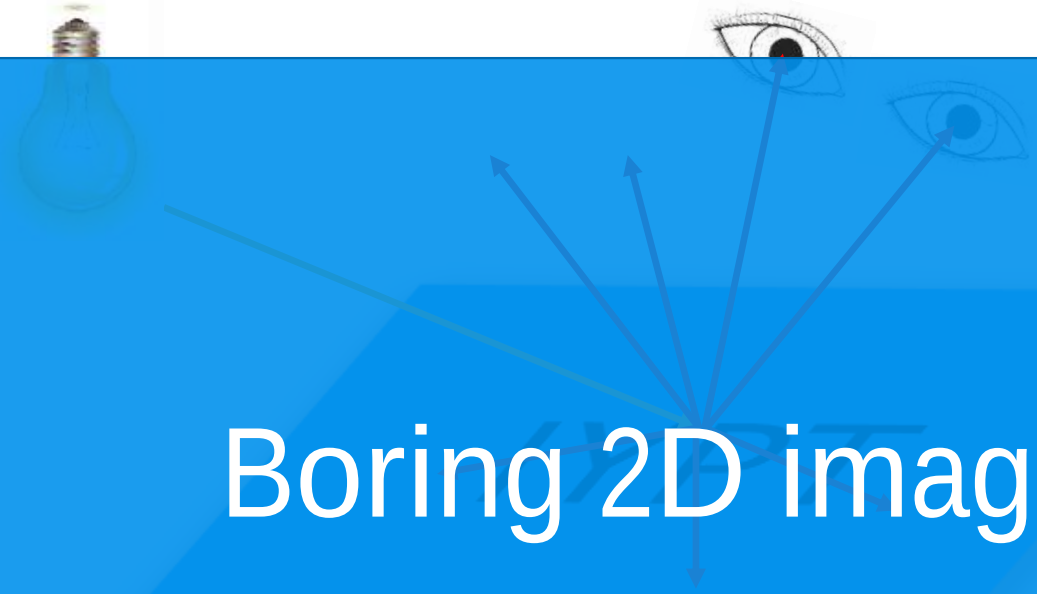
Real hologram - 3D image stored in 2D



This is not our case

What is scratched “hologram” in this task?

Simple scratch on the plastic



Boring 2D image

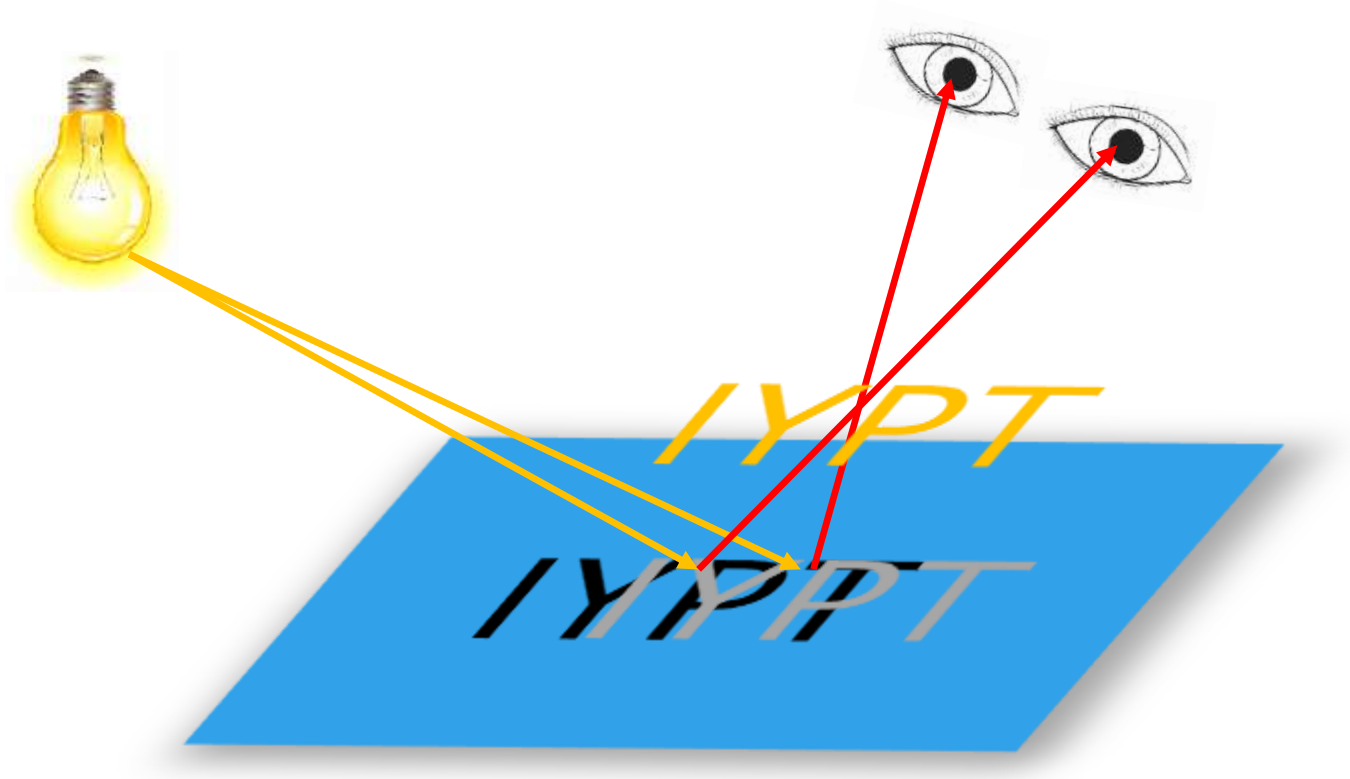
Light is scattered to all directions

Both eyes see the image at the same position

Whole image visible on the surface

"Hologram" = advanced image

If 2 eyes see 2 different images



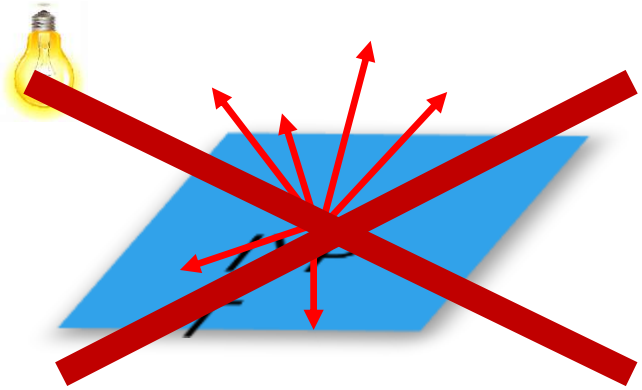
Final image is visible over/under the surface

"Hologram" = advanced image

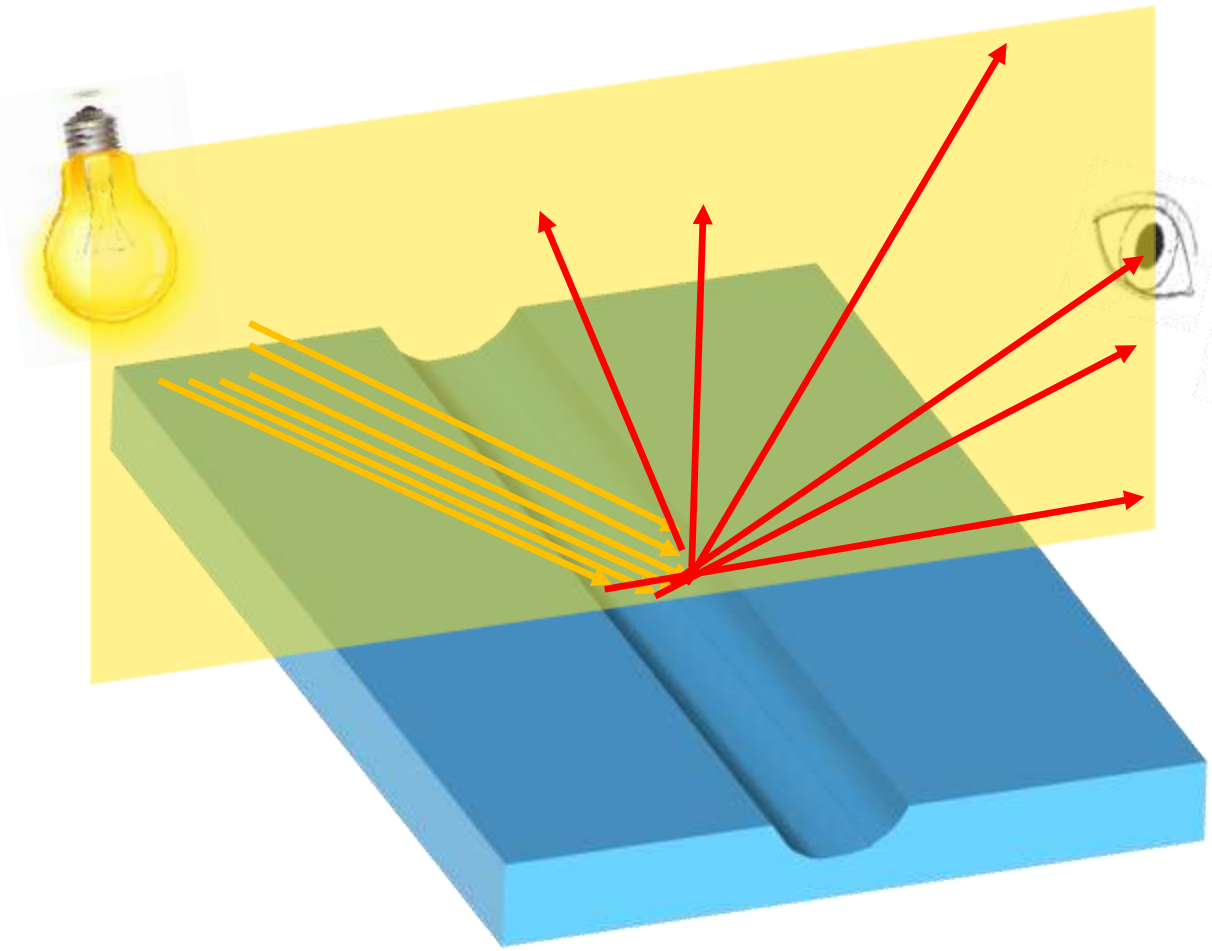
2 eyes see 2 different images

How can this be achieved?

Light can't be scattered to all directions



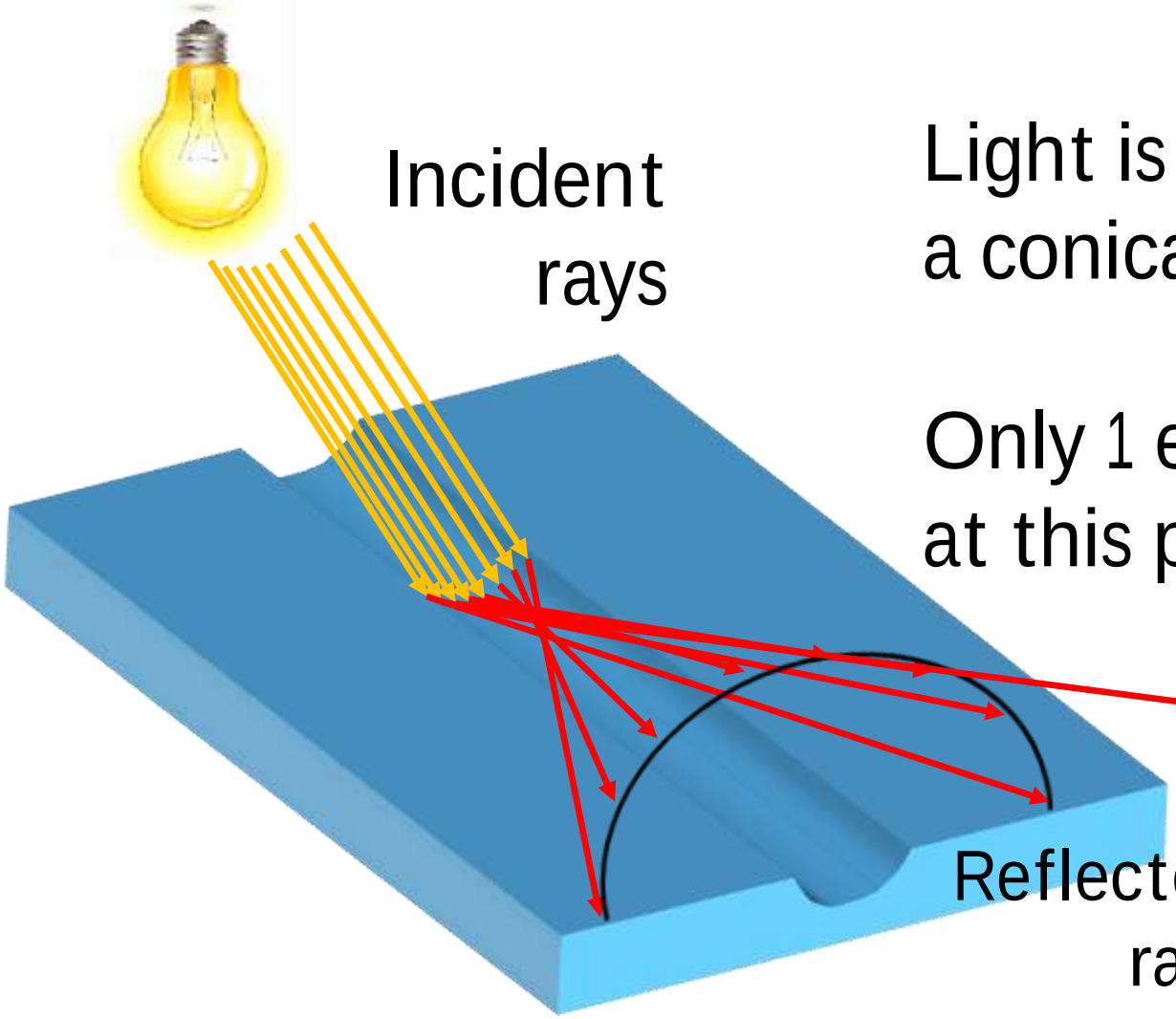
Smooth scratch – planar scatter



Light is scattered in only 1 plane

Only 1 eye can see the image at this position

Smooth scratch – conical scatter



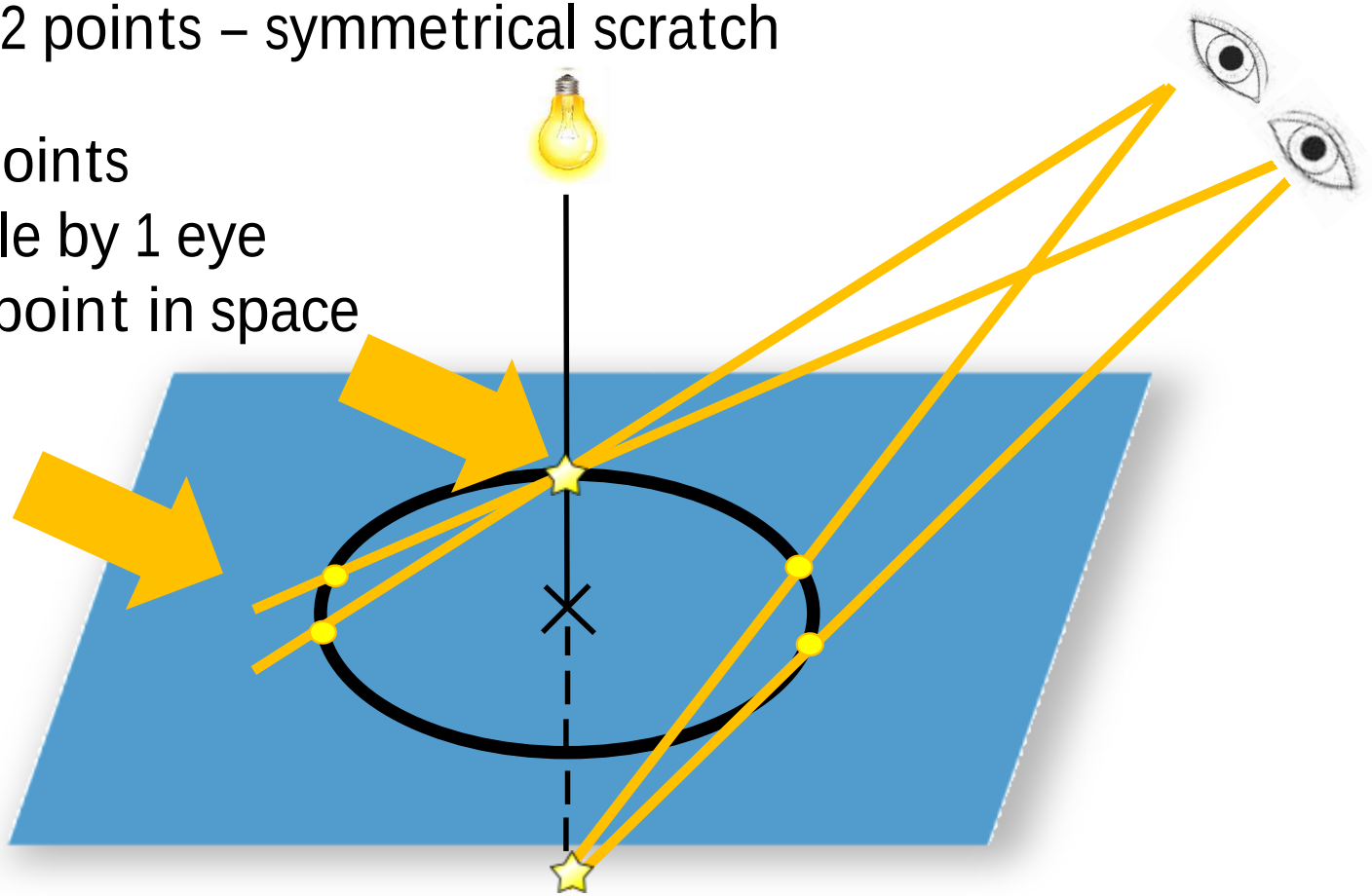
Light is reflected into a conical shape

Only 1 eye can see point at this position

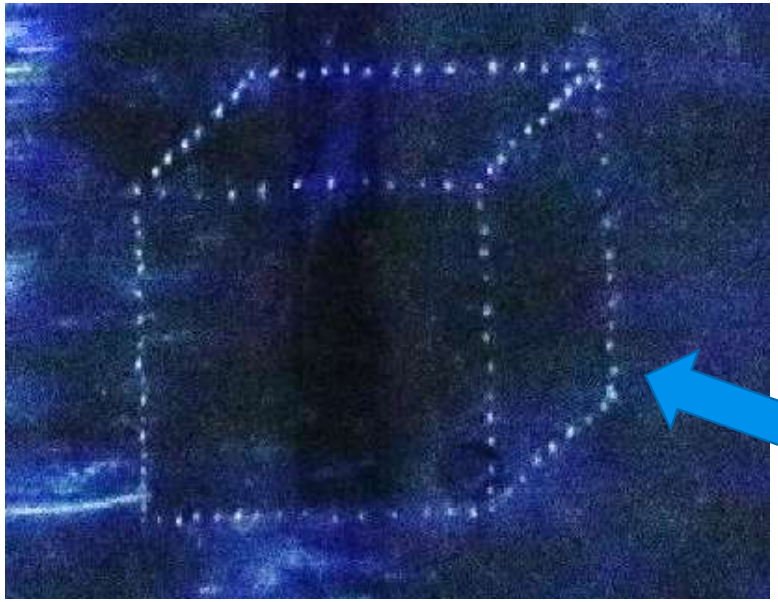
Circular scratch behaviour

Eye can see 2 points – symmetrical scratch

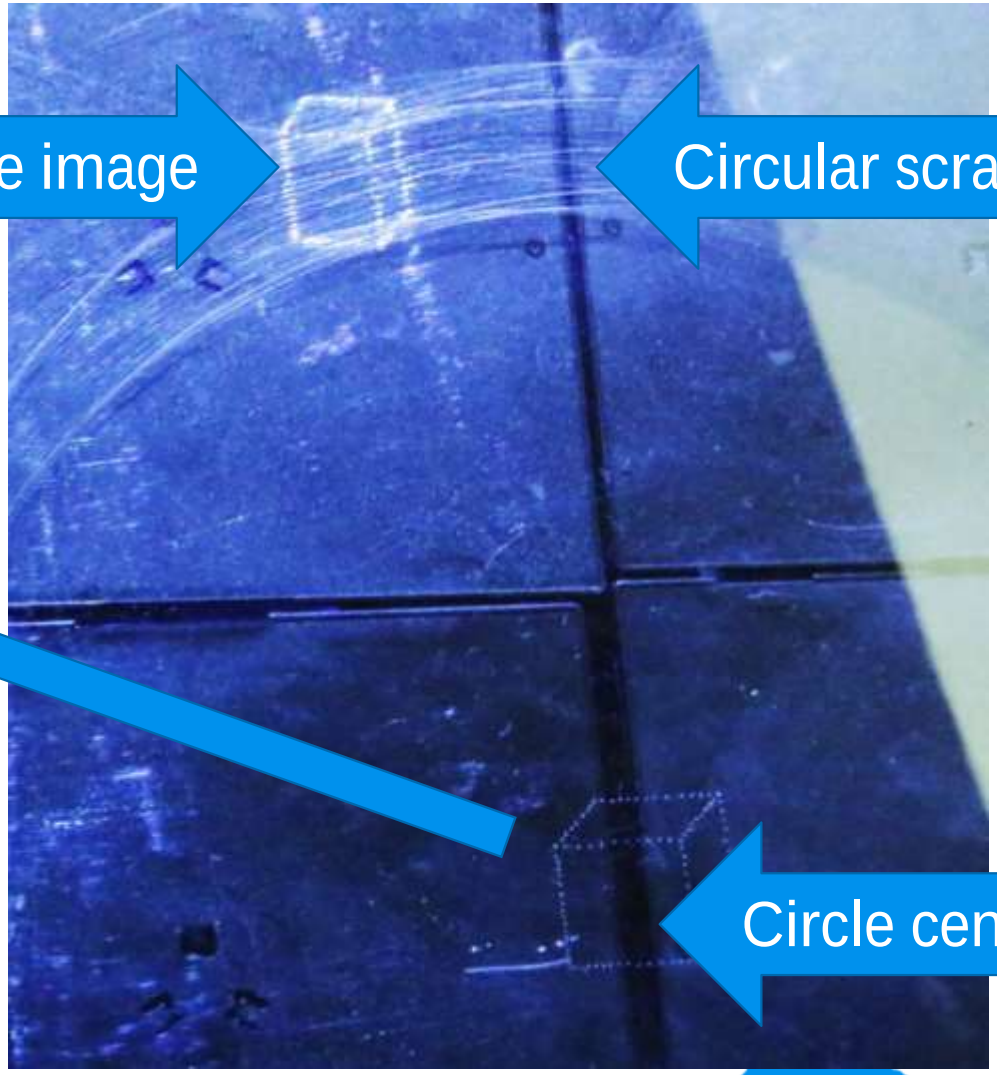
2 different points
each visible by 1 eye
form a point in space



Hologram of 2D view of cube



More points of reflection
= reflected line

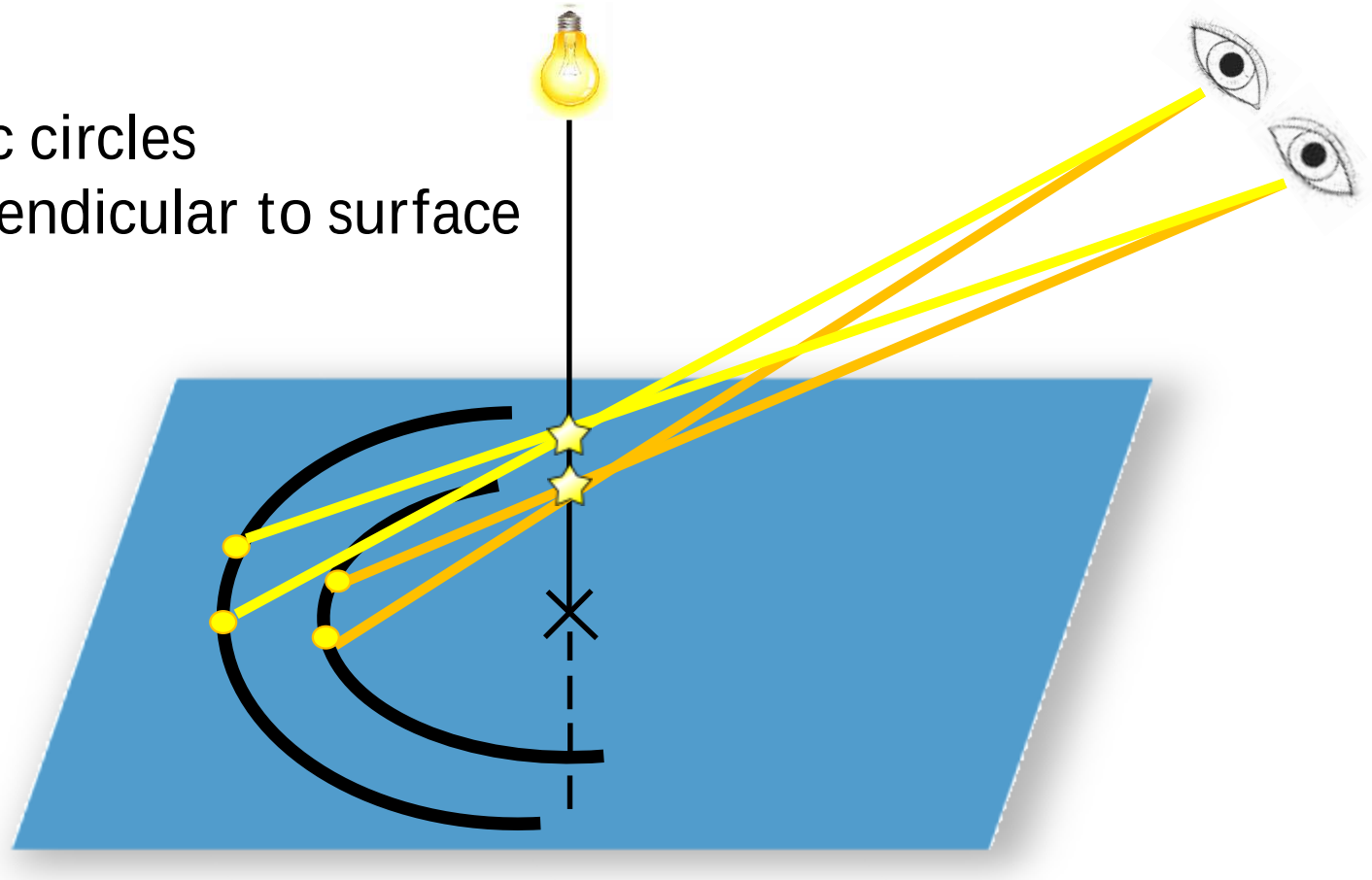


2D HOLOGRAM IS EASY TO CREATE

LET'S CREATE 3D HOLOGRAM

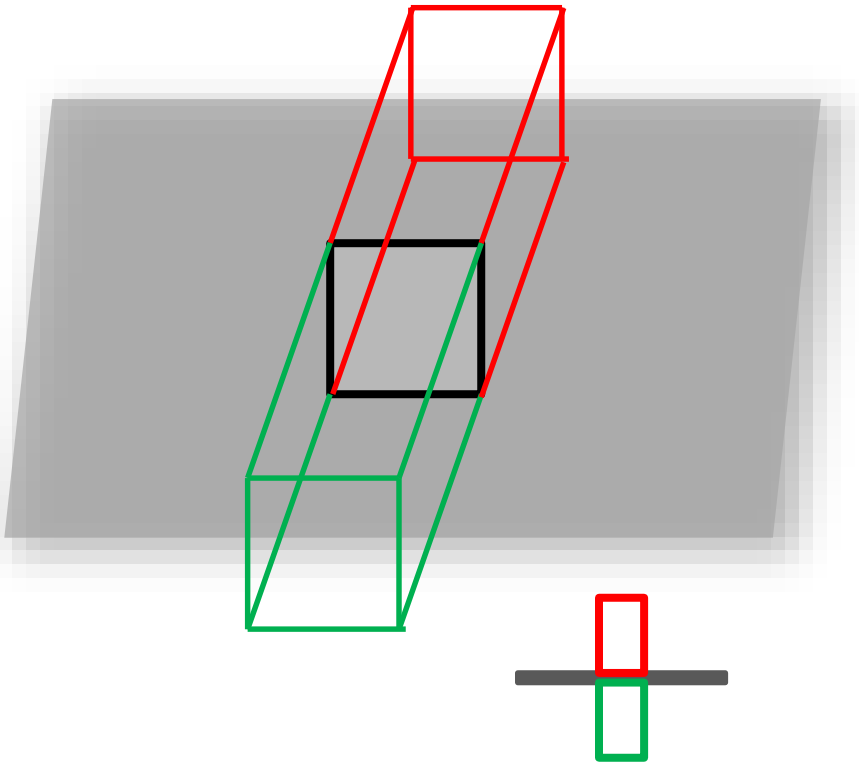
Depth of point vs. radius of circle

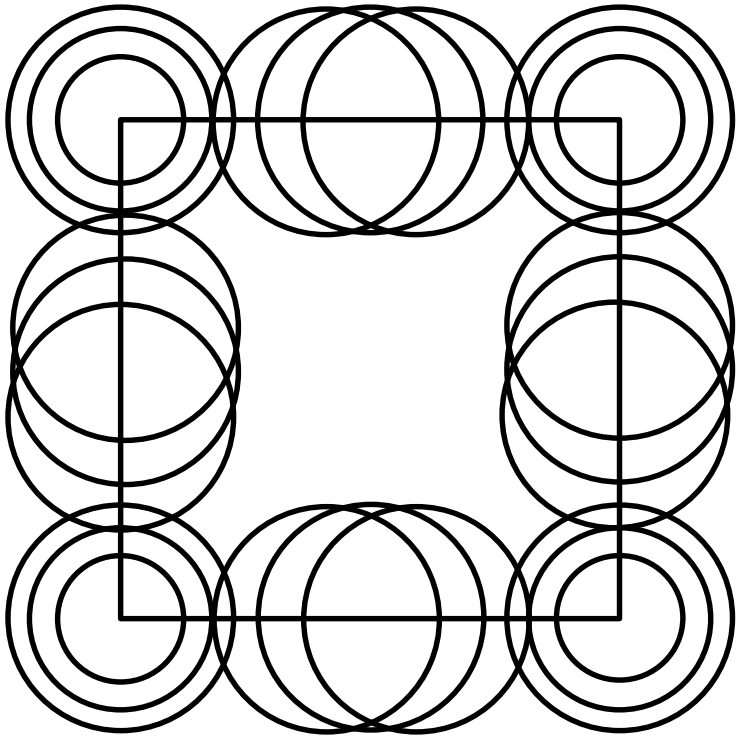
Concentric circles
= line perpendicular to surface



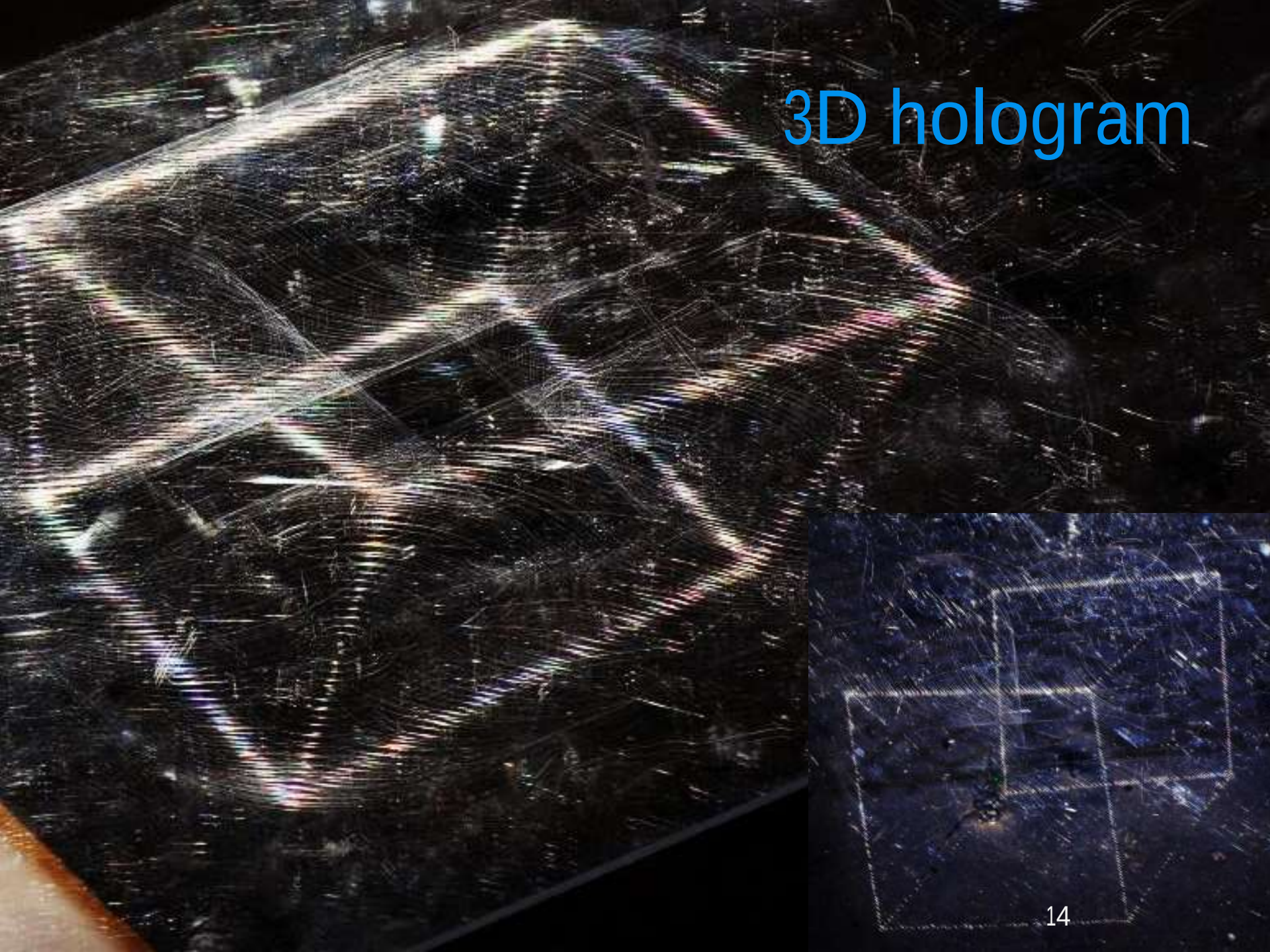
Creating 3D hologram

1) Sketch
2) Concentric circles = perpendicular
3) Constant radius = parallel



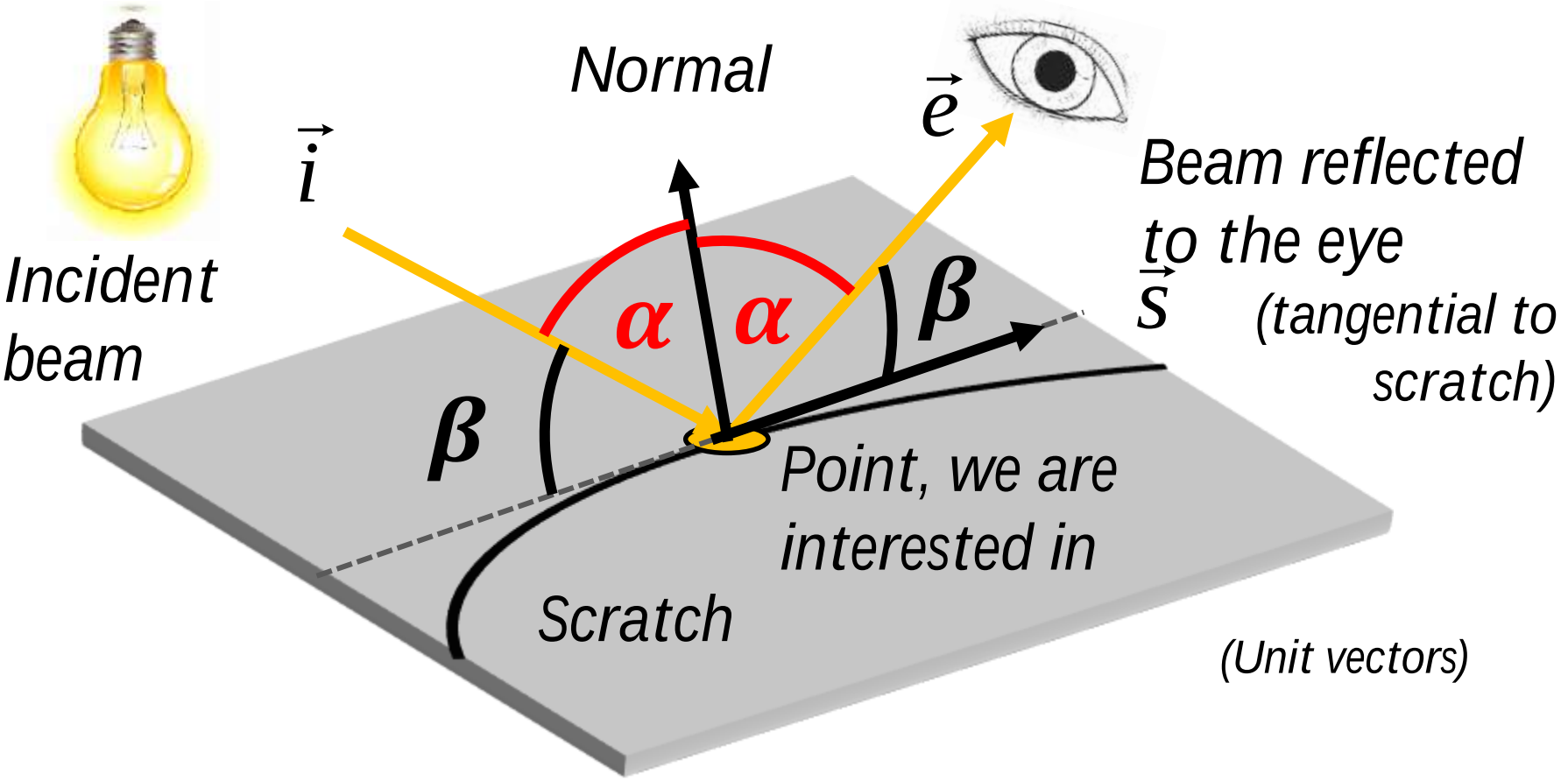


3D hologram



EXACT CALCULATION OF REFLECTION POINTS

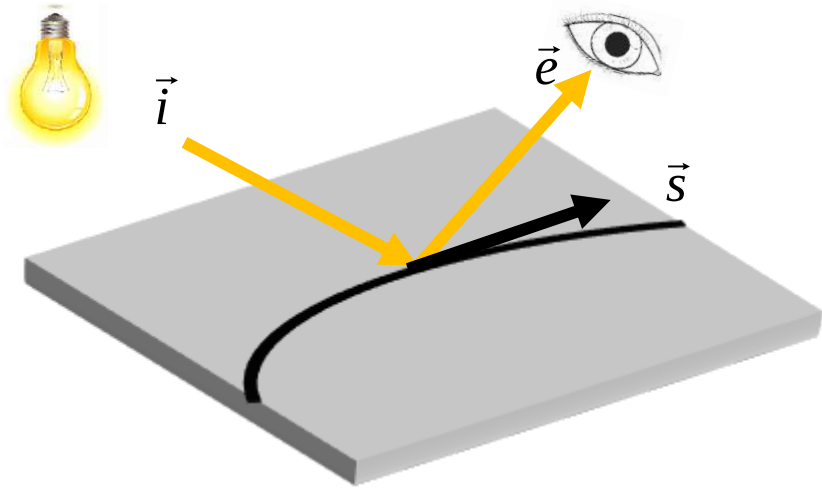
Geometry of the beam



Condition of shining: $\vec{i} \circ \vec{s} = \vec{e} \circ \vec{s}$

Condition of reflection

$$\vec{i} \circ \vec{s} = \vec{e} \circ \vec{s}$$



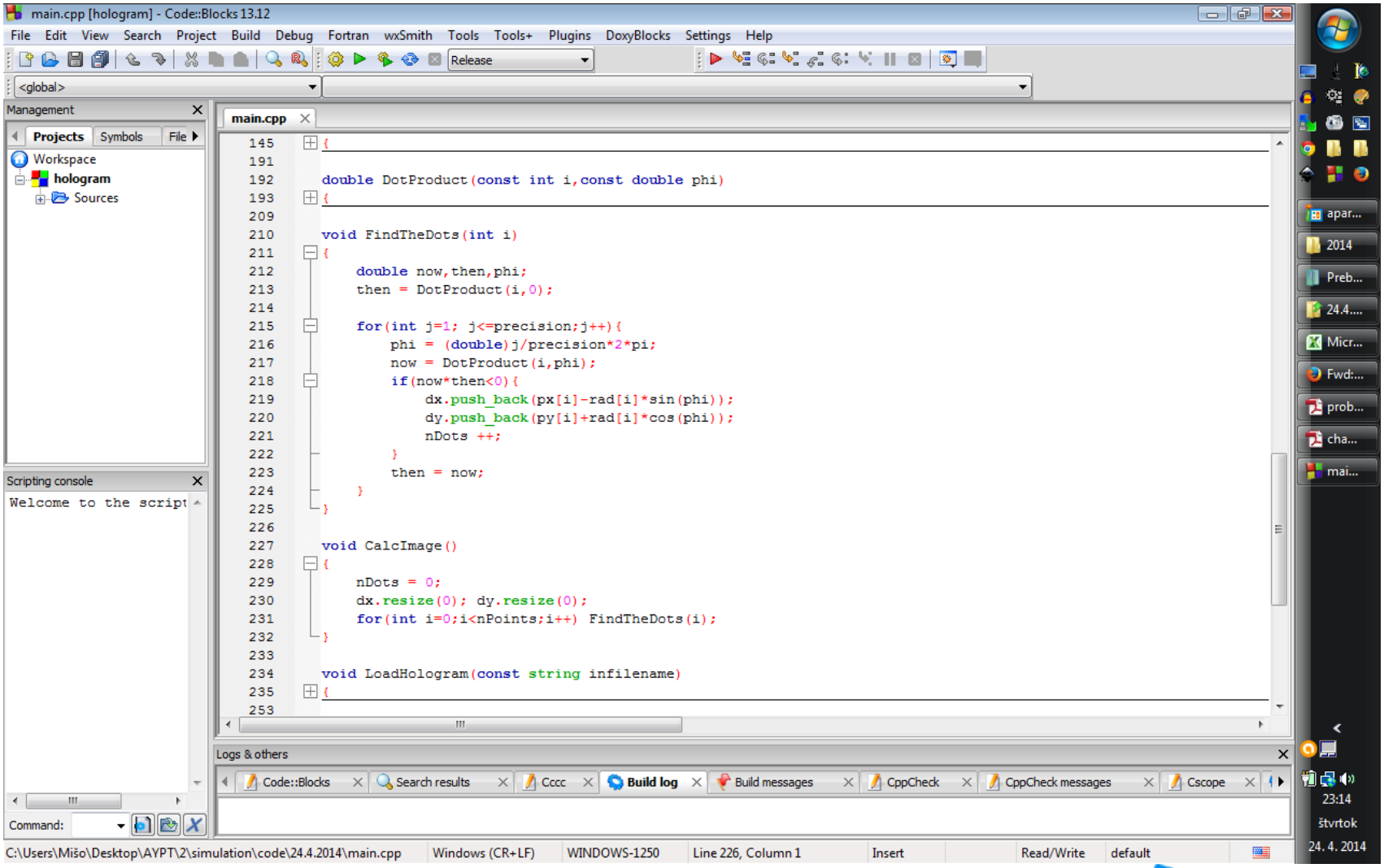
Which points on a circle fulfill this condition?

Too difficult to solve analytically

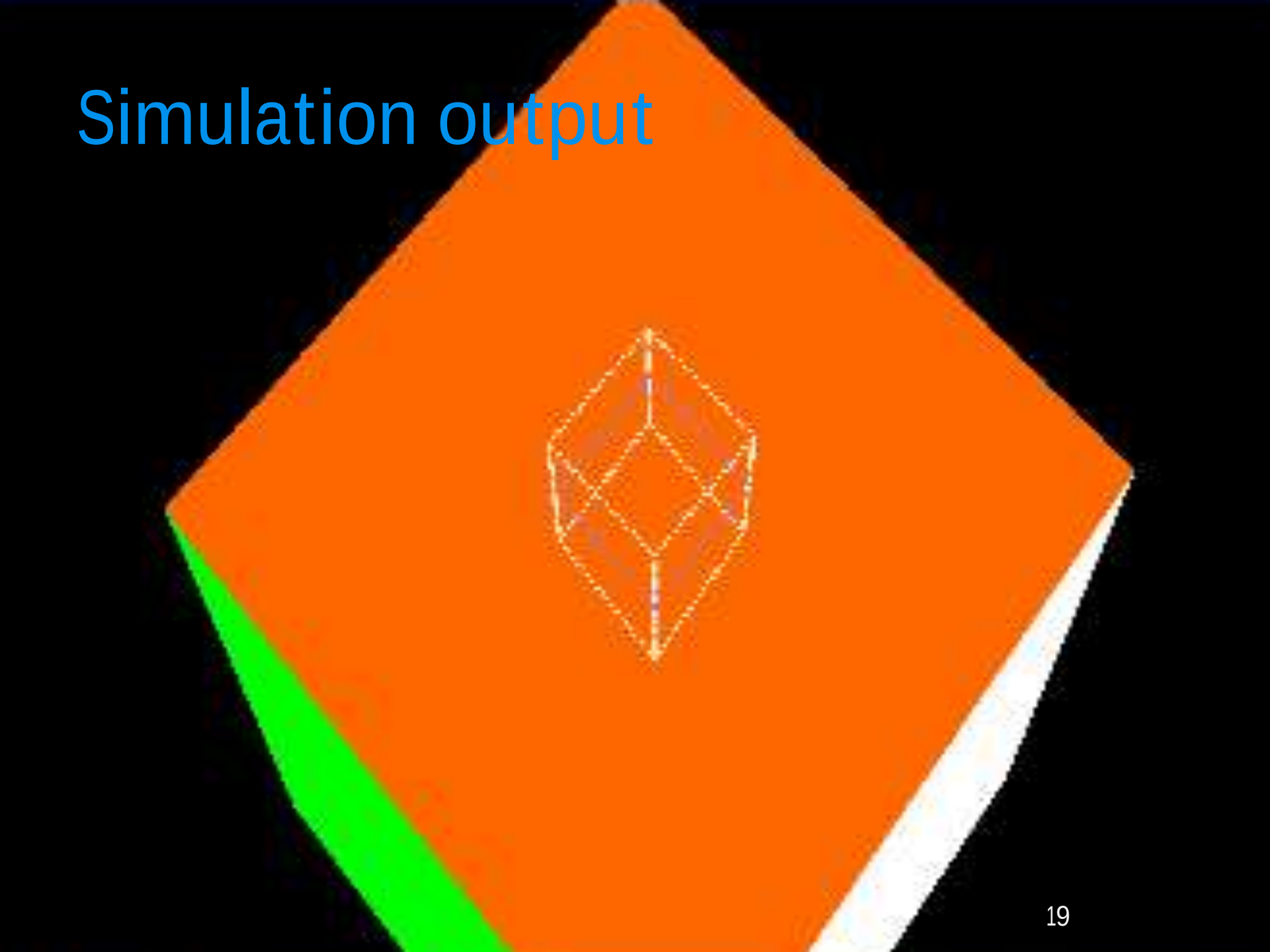
Simulation

- C++, OpenGL 3D graphics
- Numerically checks the condition
  - for 600+ points on each circle

Simulation


main.cpp [hologram] - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Release
<global>
Management
Projects Symbols File
Workspace
hologram
Sources
Scripting console
Welcome to the script
Command:
main.cpp
145 {
191
192 double DotProduct(const int i,const double phi)
193 {
209
210 void FindTheDots(int i)
211 {
212 double now,then,phi;
213 then = DotProduct(i,0);
214
215 for(int j=1; j<=precision;j++){
216 phi = (double)j/precision*2*pi;
217 now = DotProduct(i,phi);
218 if(now*then<0){
219 dx.push_back(px[i]-rad[i]*sin(phi));
220 dy.push_back(py[i]+rad[i]*cos(phi));
221 nDots ++;
222 }
223 then = now;
224 }
225 }
226
227 void CalcImage()
228 {
229 nDots = 0;
230 dx.resize(0); dy.resize(0);
231 for(int i=0;i<nPoints;i++) FindTheDots(i);
232 }
233
234 void LoadHologram(const string infilename)
235 {
253
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck
CppCheck messages
Cscope
C:\Users\Mišo\Desktop\AYPT\2\simulation\code\24.4.2014\main.cpp
Windows (CR+LF)
WINDOWS-1250
Line 226, Column 1
Insert
Read/Write
default
23:14
štvrtok
24. 4. 2014

Simulation output

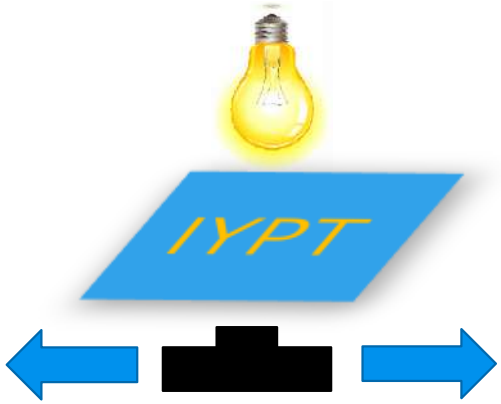


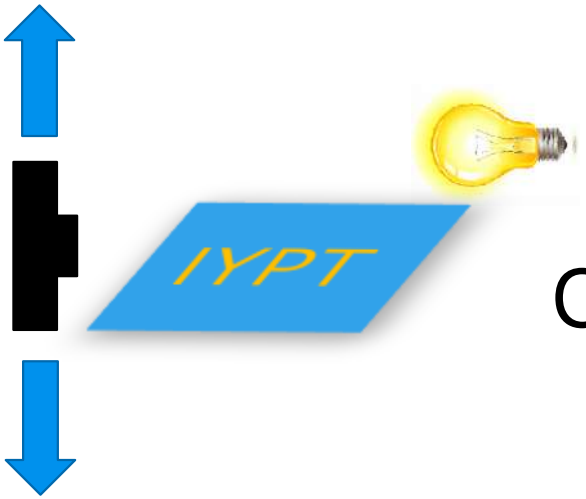
EXPERIMENTAL VERIFICATION OF THEORY

Experimental conditions

Light and hologram - stay at the same place

Camera moves horizontally





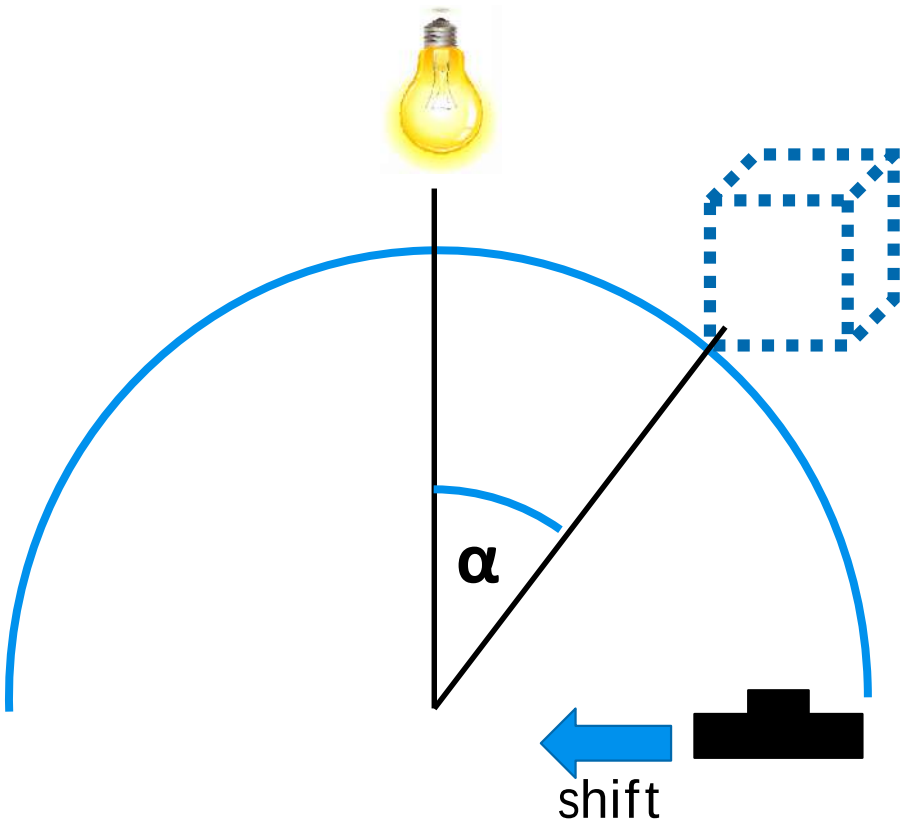
Camera moves vertically

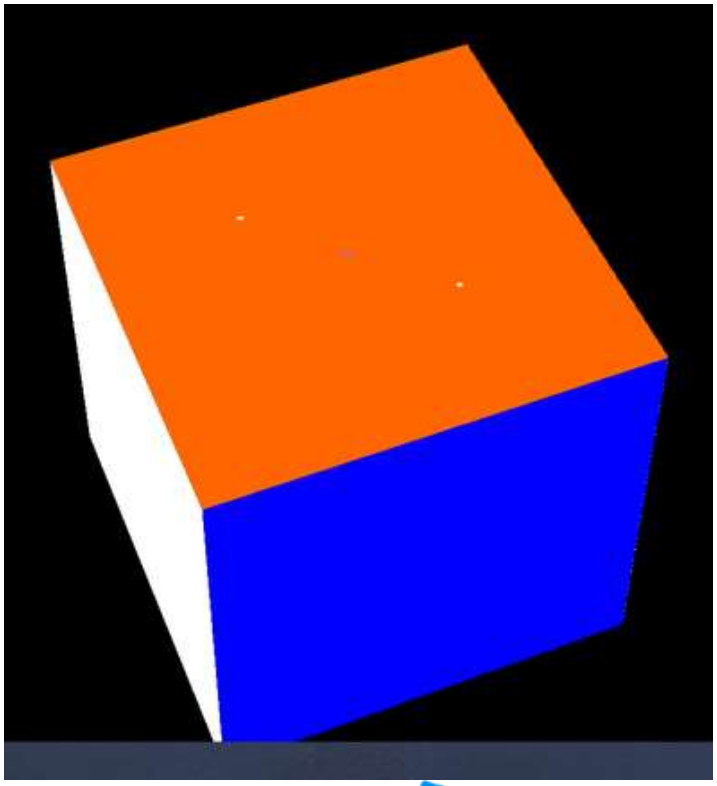
Horizontal camera movement

Angular position of 1 point

vs.

Horizontal displacement of camera



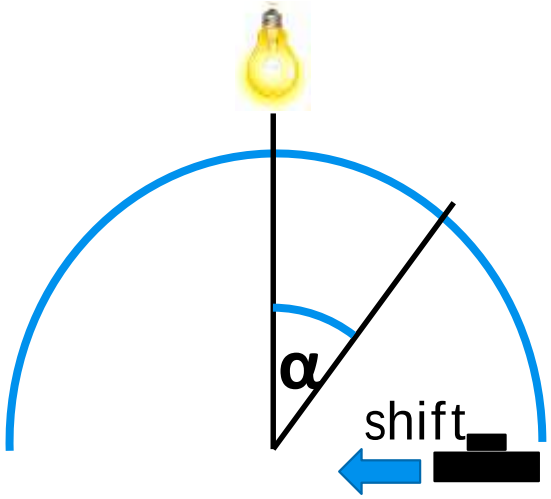


Experimental setup

Camera shift

Hologram shift

Horizontal movement - simulation



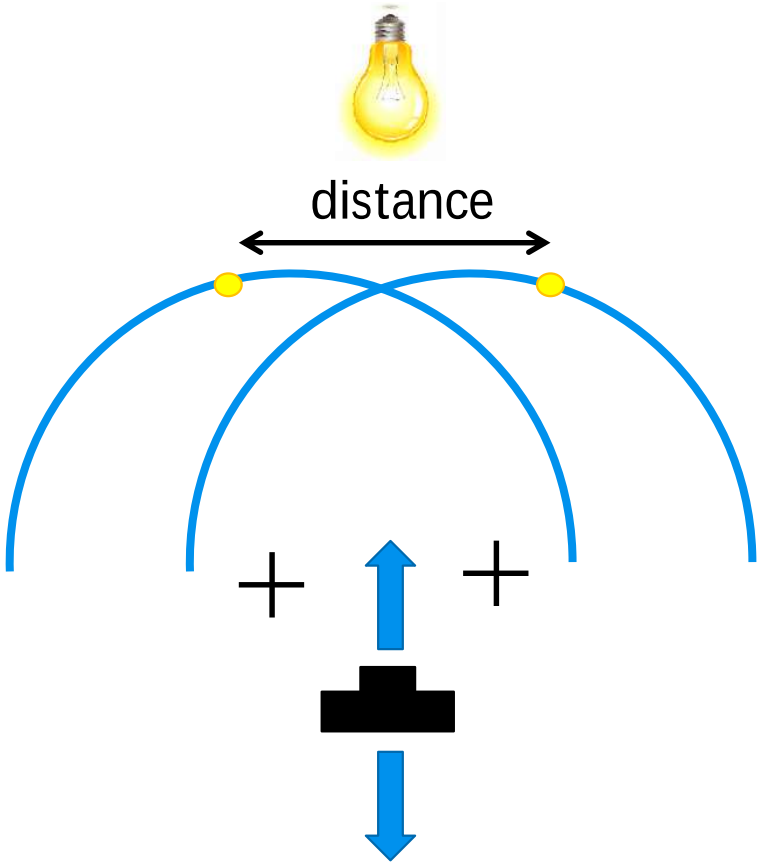
Simulation vs. experiment

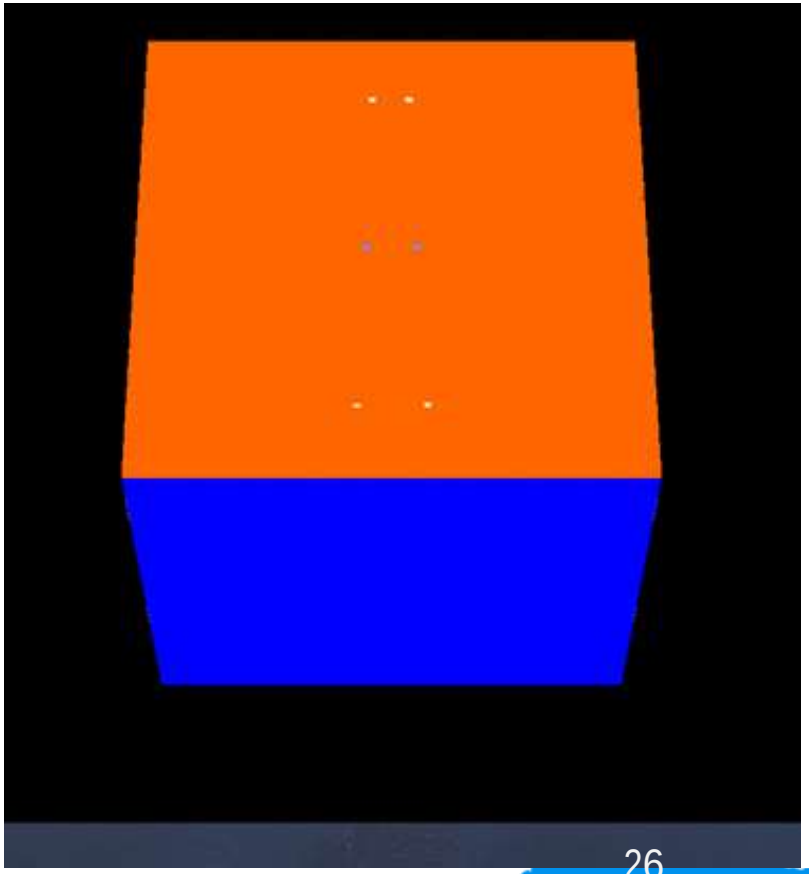
Vertical camera movement

Distance of 2 points

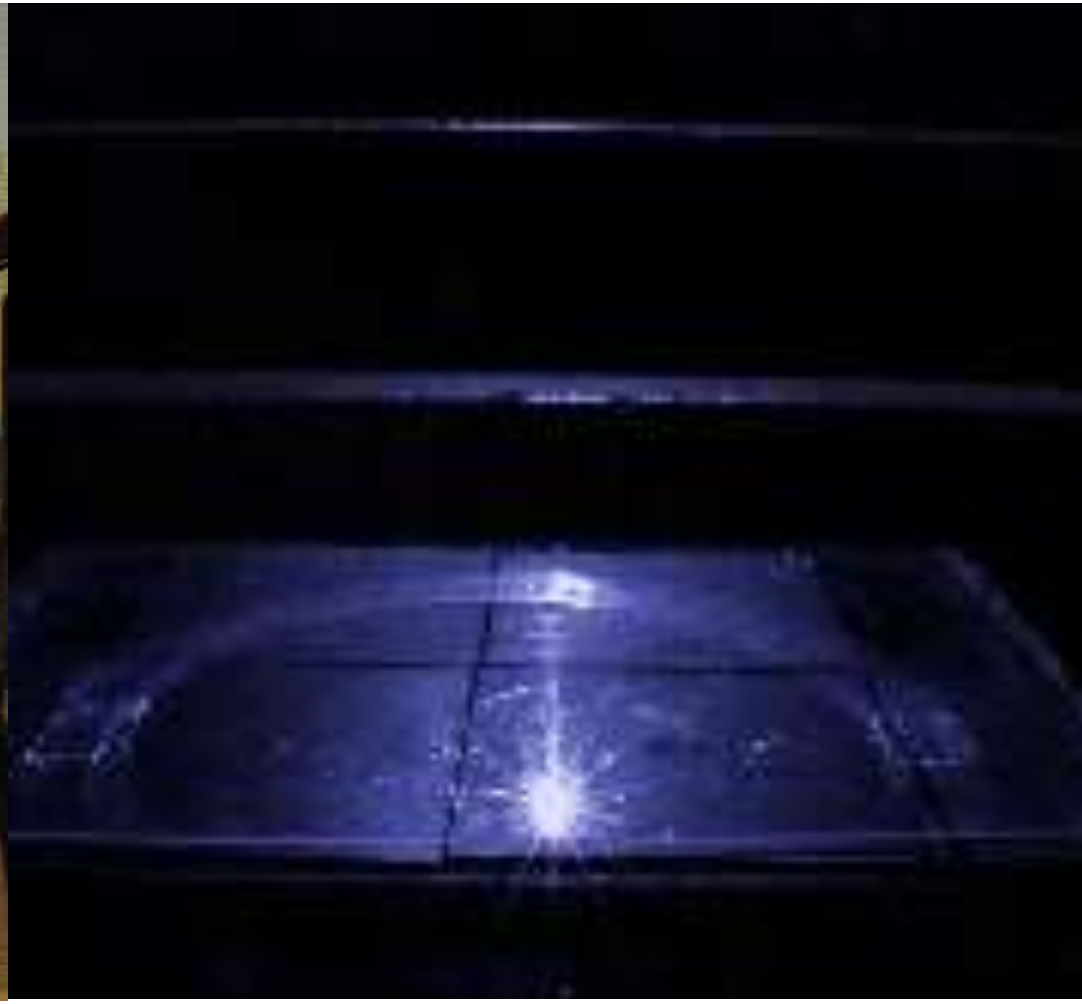
vs.

Vertical displacement of camera

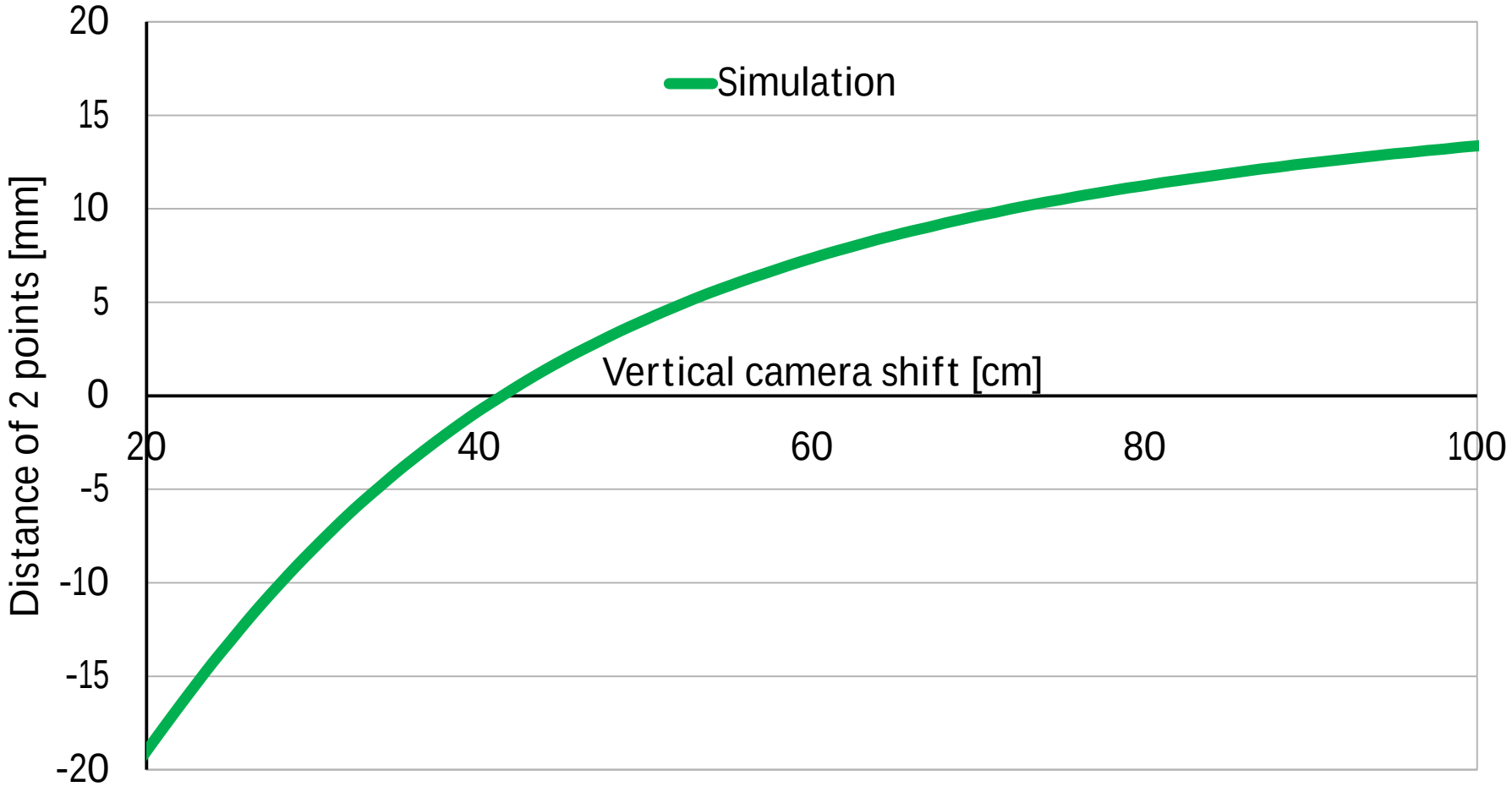




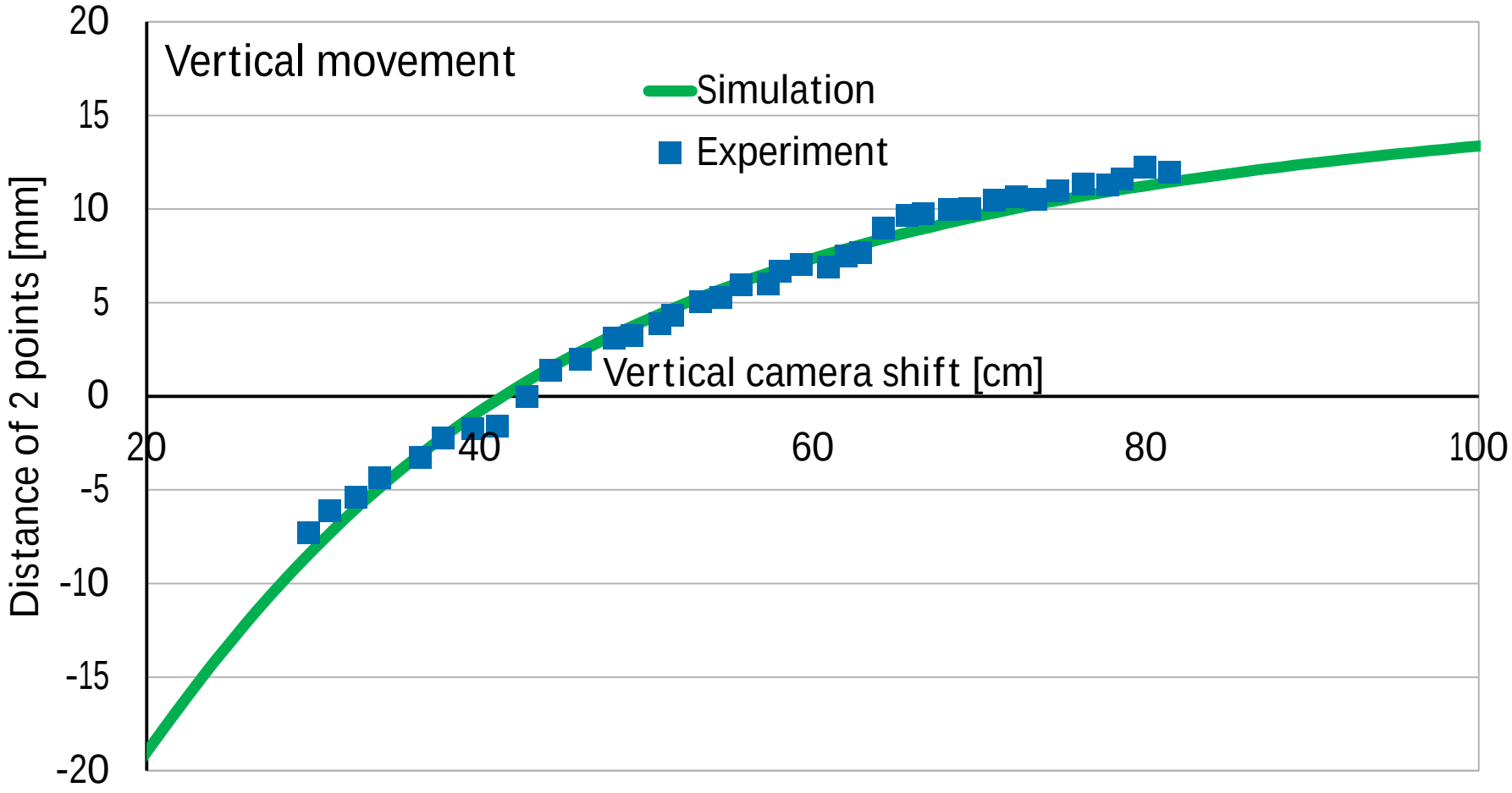
Experimental setup



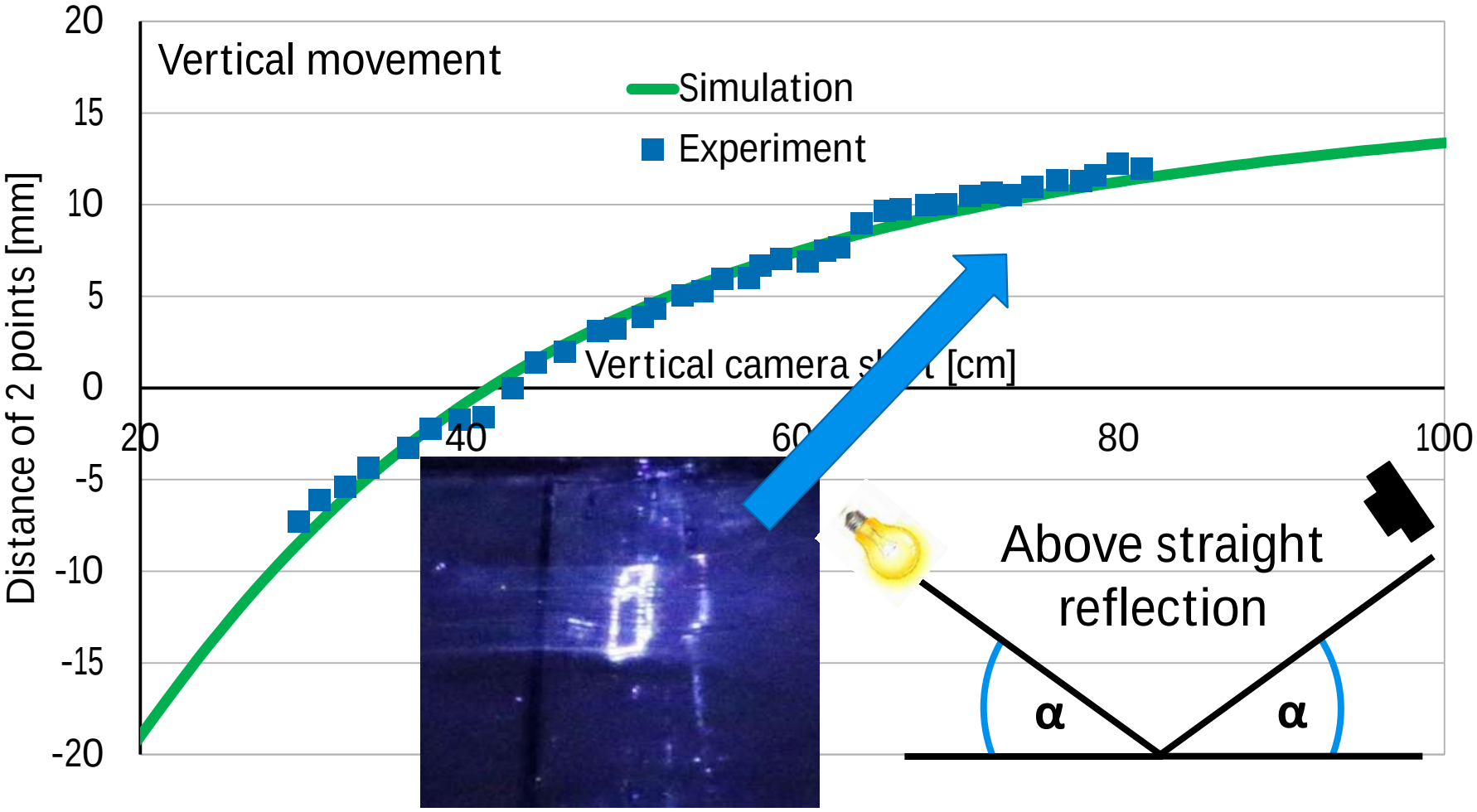
Vertical movement - simulation



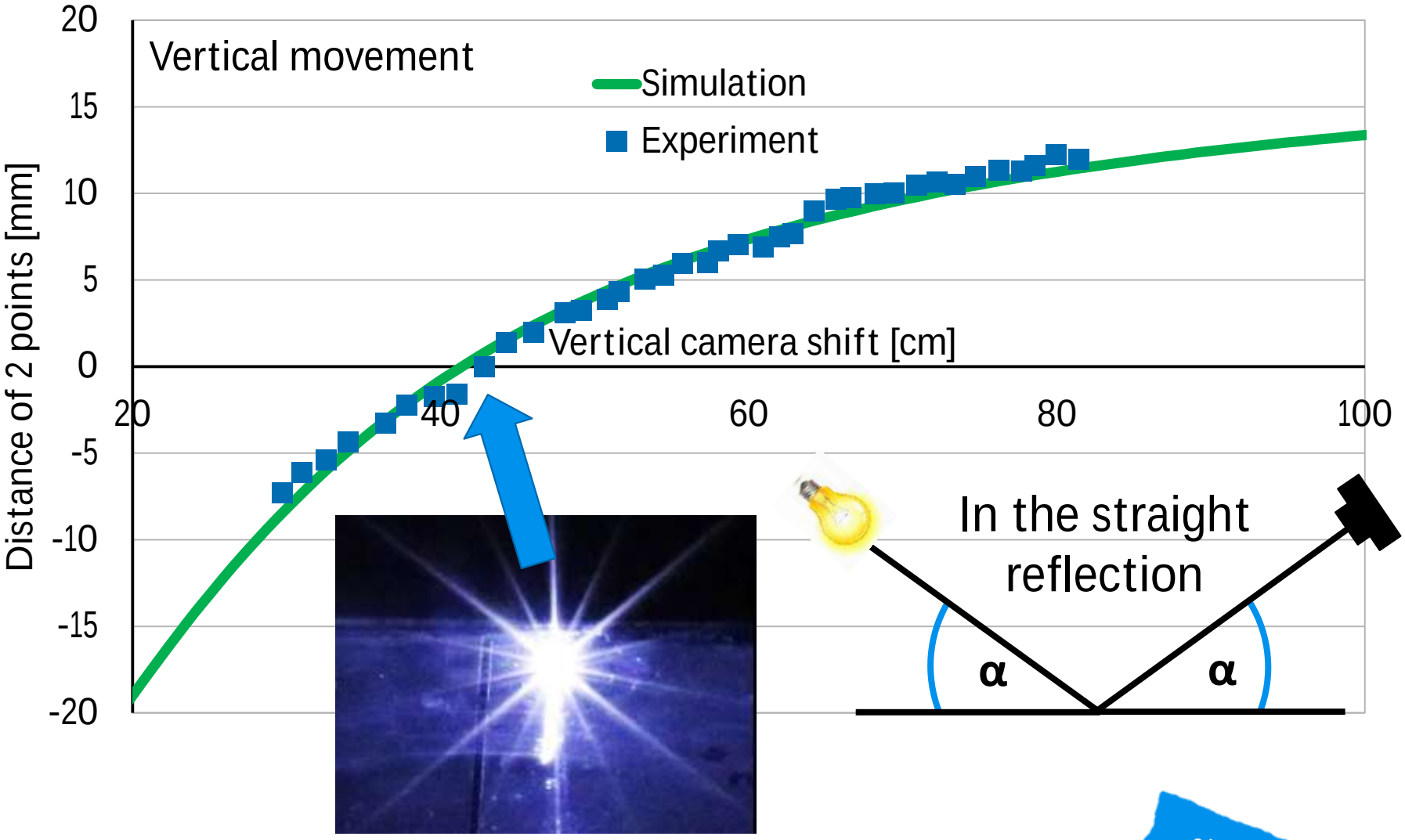
Simulation vs. experiment



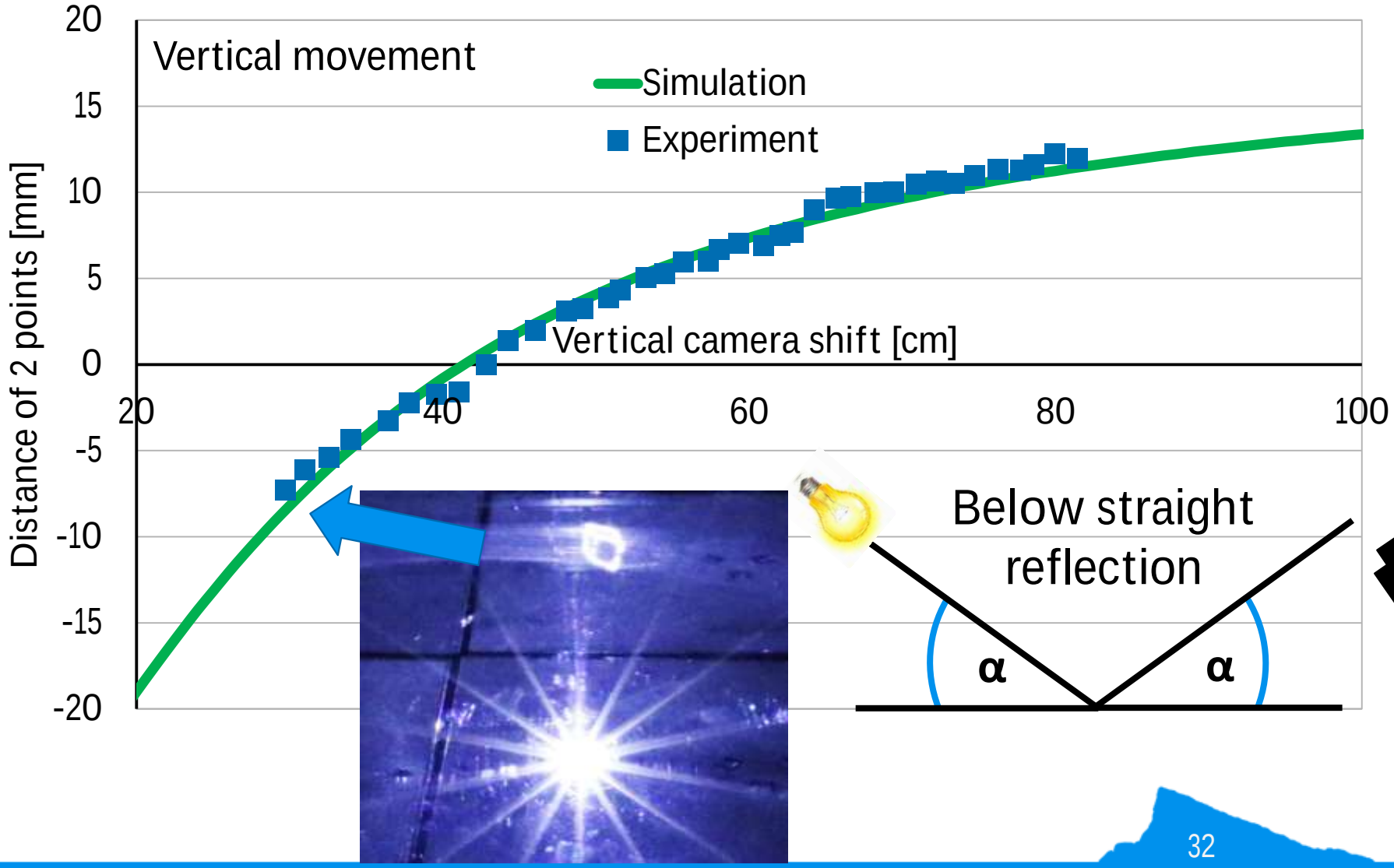
Simulation vs. experiment



Simulation vs. experiment



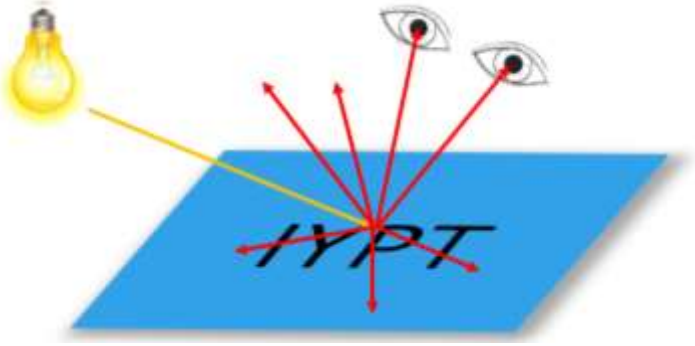
Simulation vs. experiment



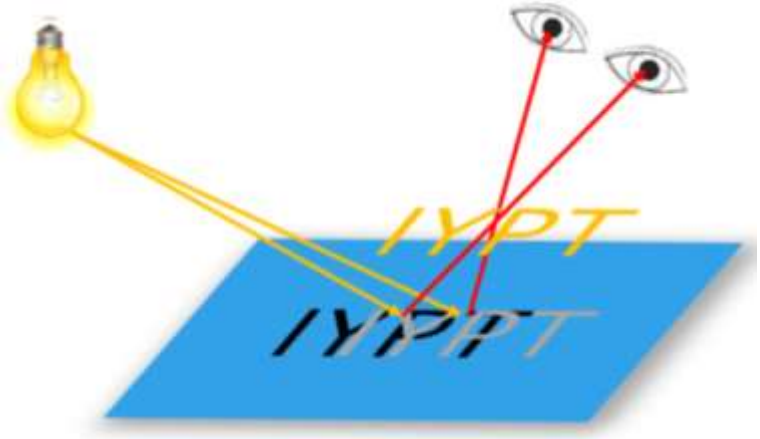
CONCLUSION

Conclusion – simple mechanism

Simple scratch = boring image

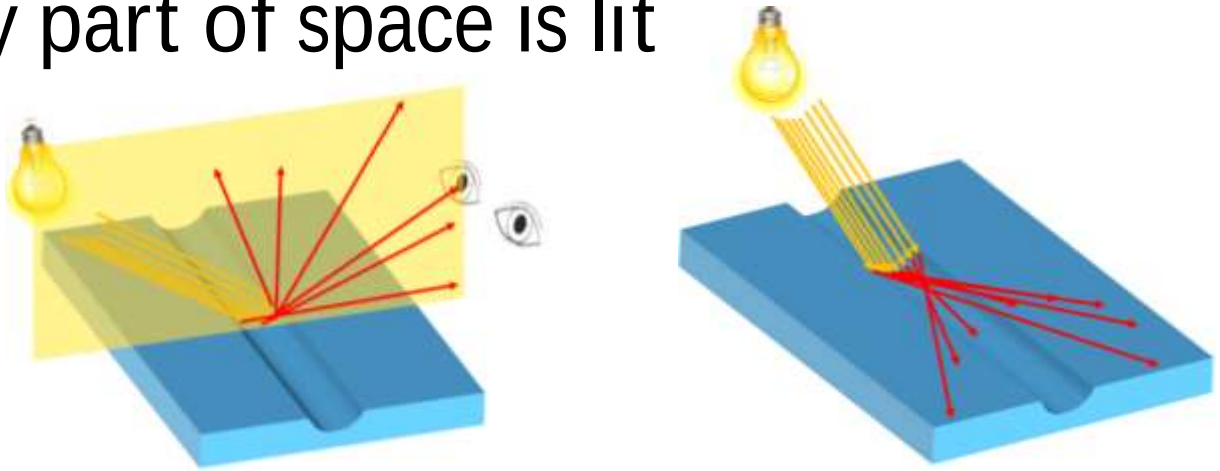


"Hologram" = image visible under/over the surface

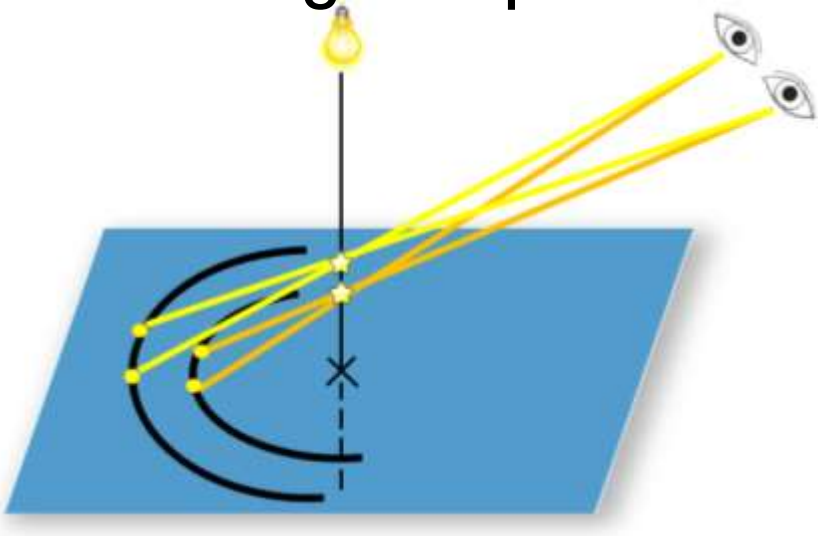


Conclusion – simple mechanism

Scratches = only part of space is lit

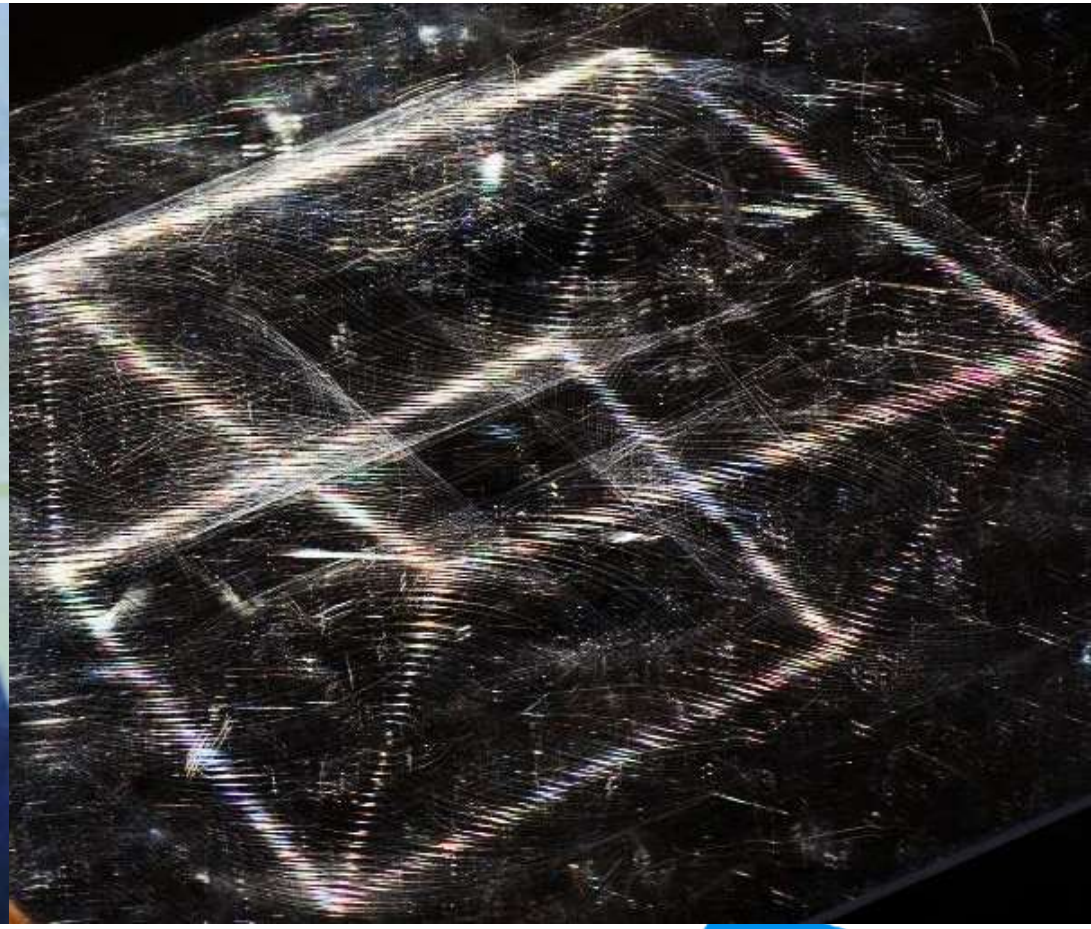


Eyes see 2 different images = points in space

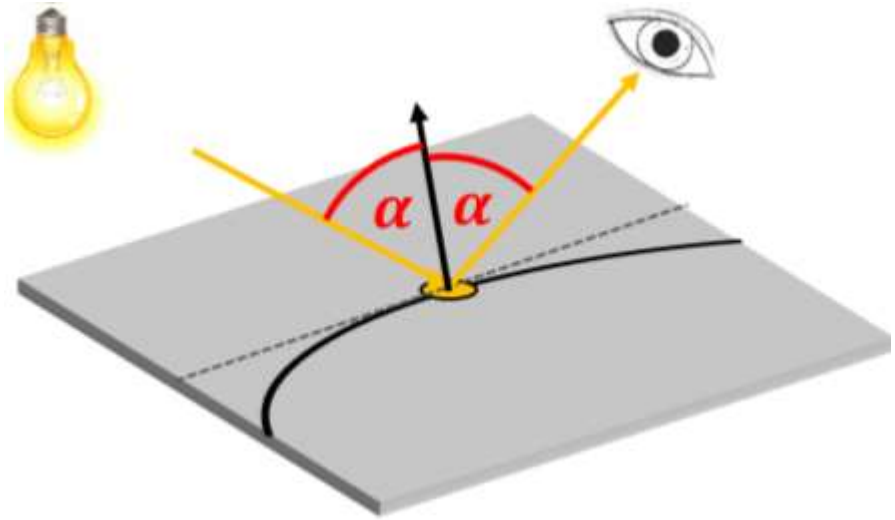


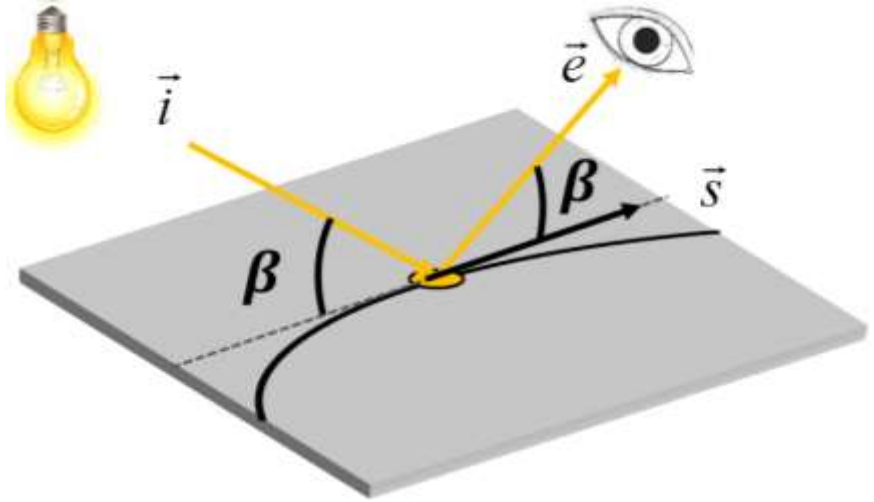
Conclusion – simple mechanism

Working 2D & 3D holograms



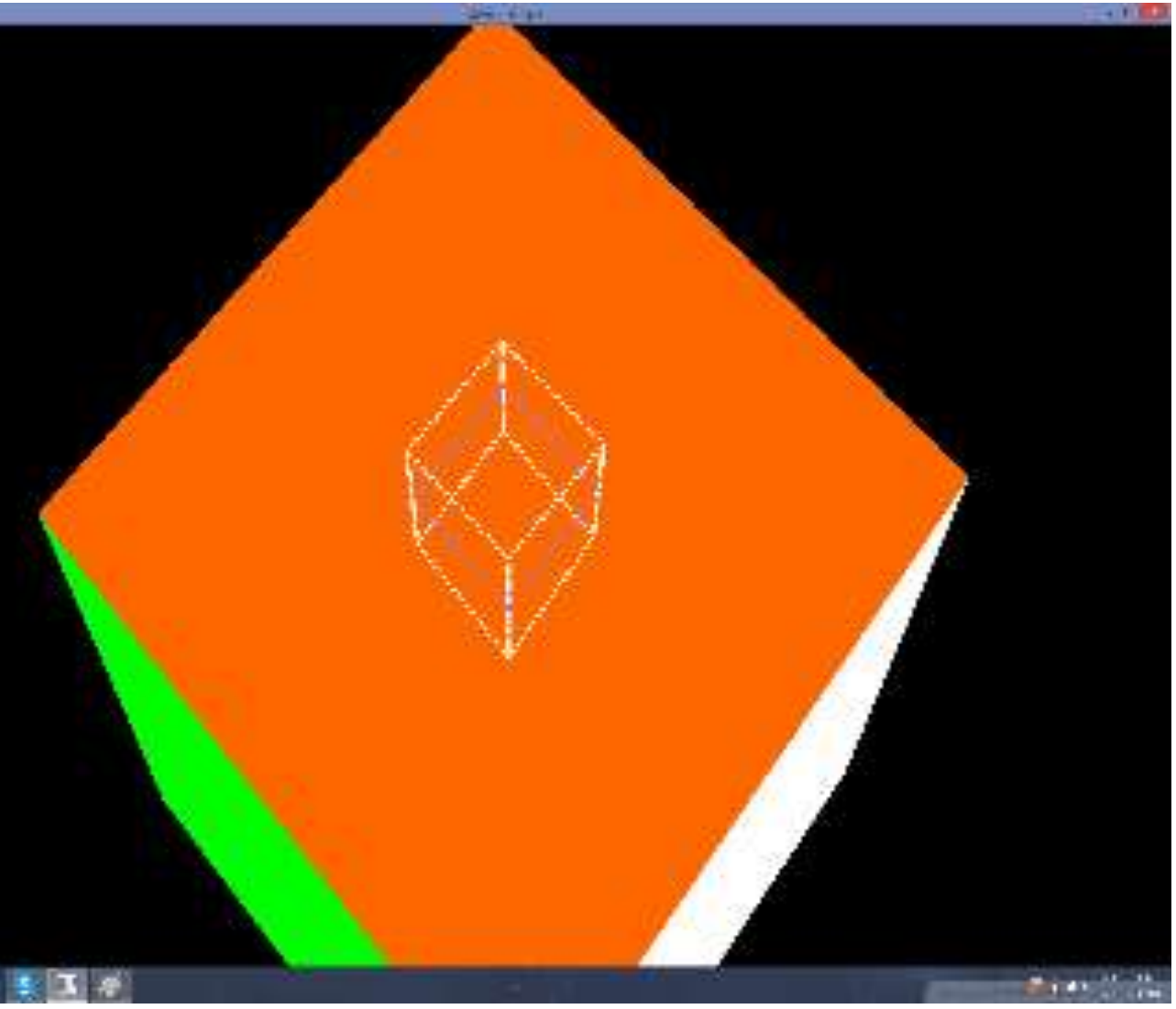
Theory = geometrical optics

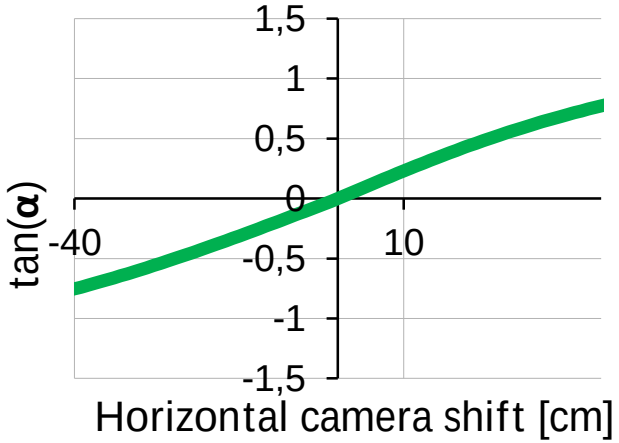


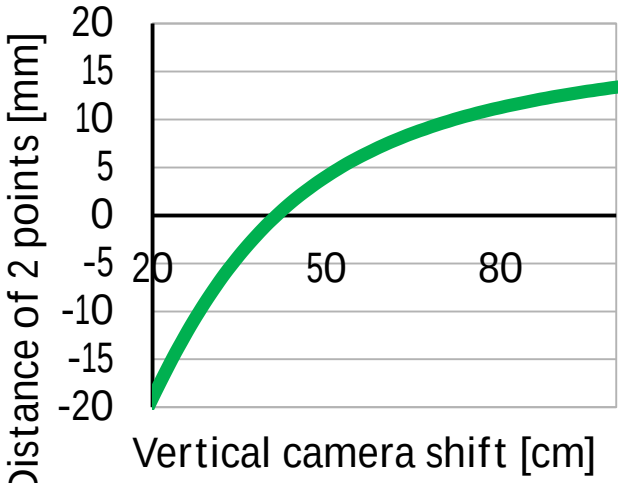


Law of reflection → $\vec{i} \circ \vec{s} = \vec{e} \circ \vec{s}$

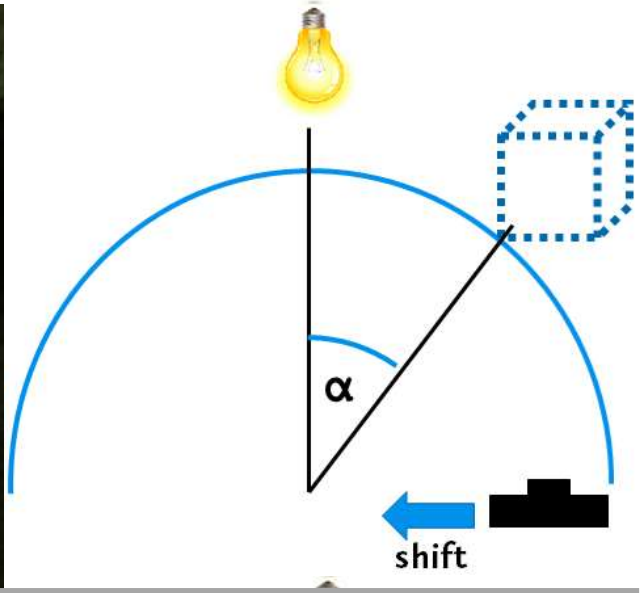
Simulation = practical use of theory

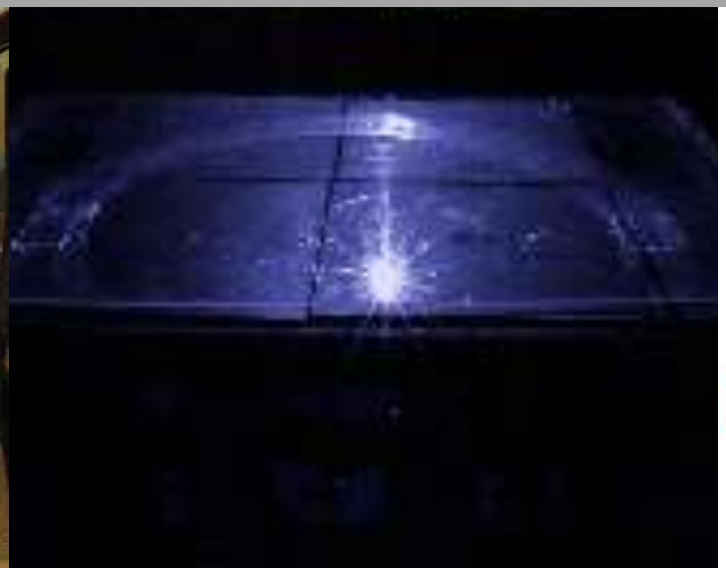


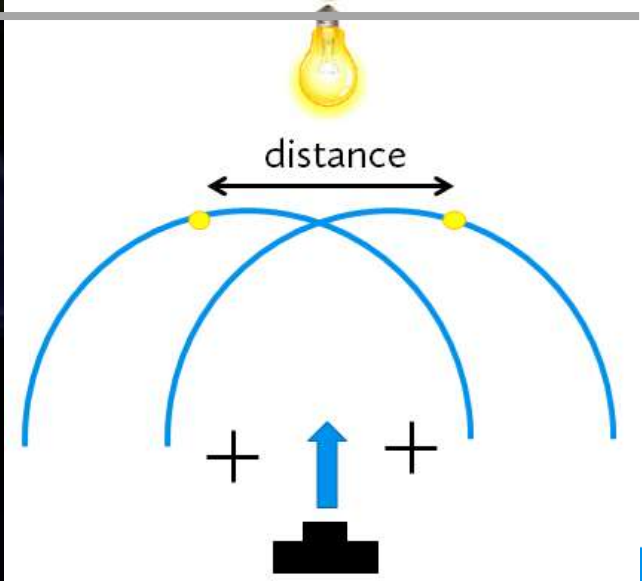




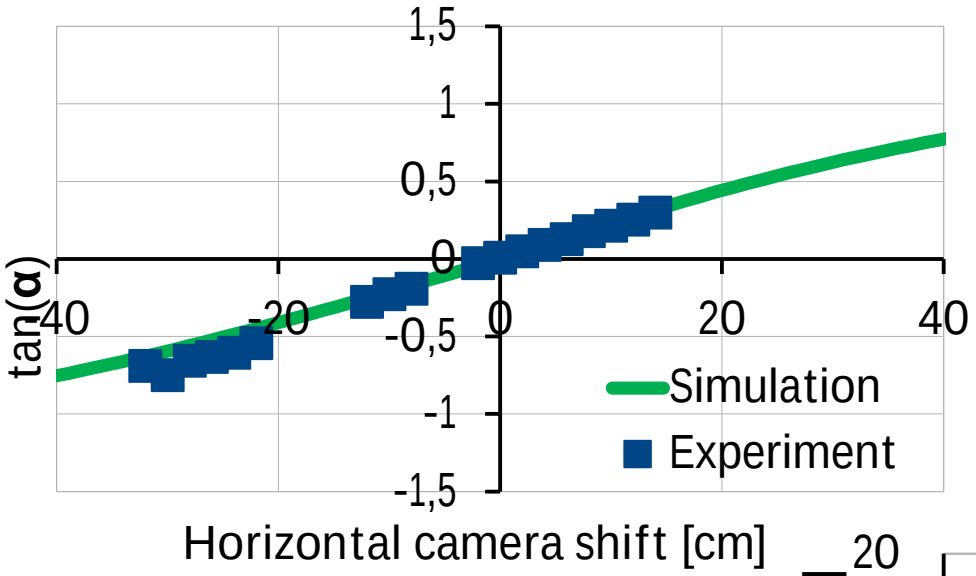
Conclusion - Experiments

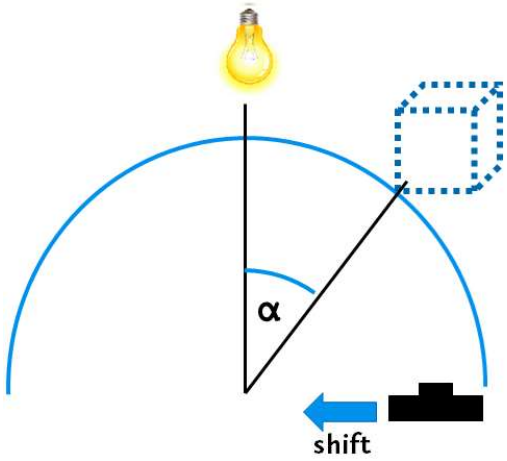


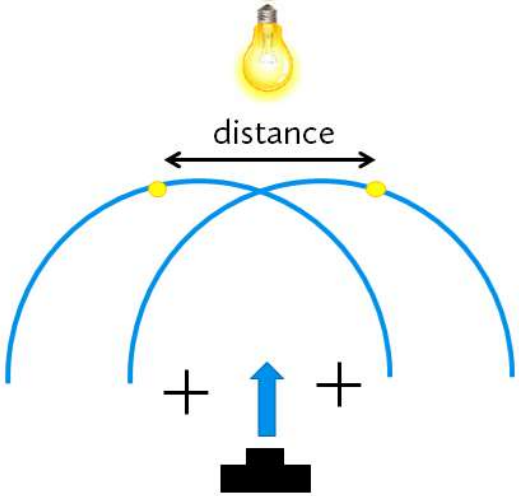


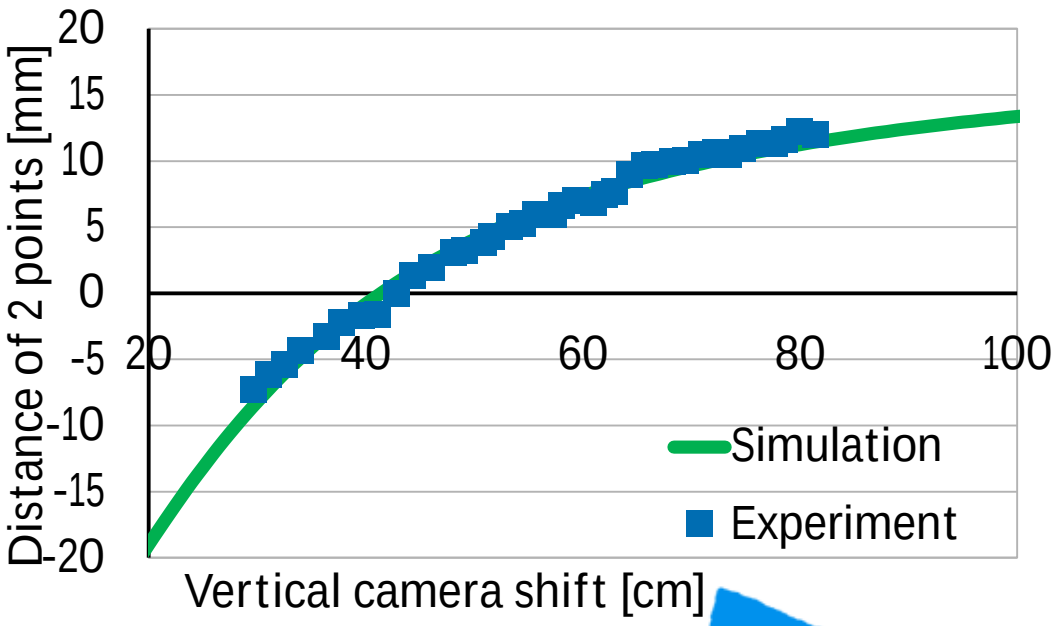


Experimental verification of theory









Thank you for your attention

APPENDICES

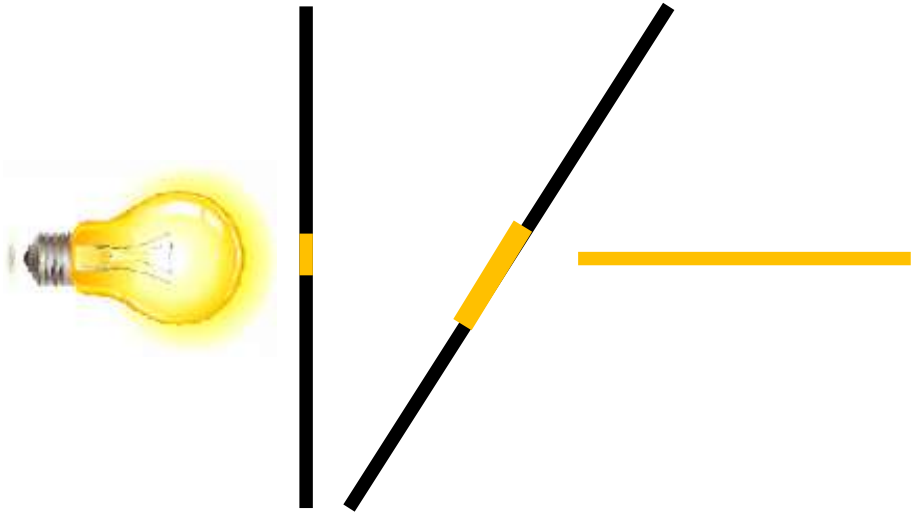
More than 1 image visible?

more points on circle may reflect light

Circular scratch

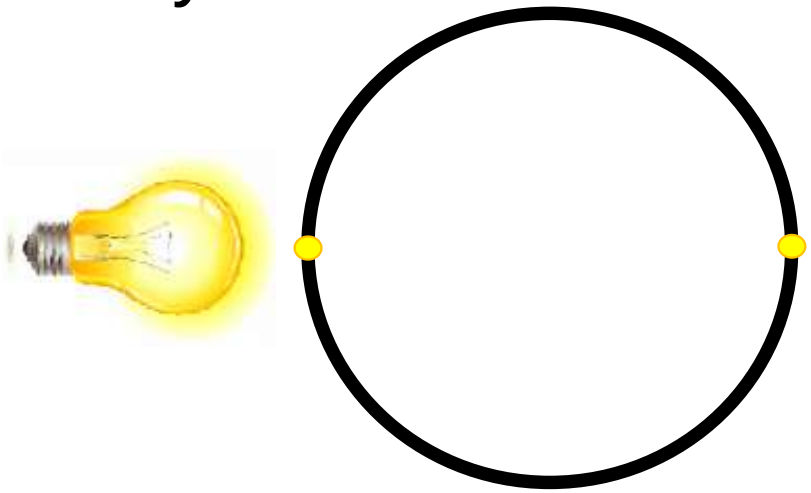
Line

- Line segments visible

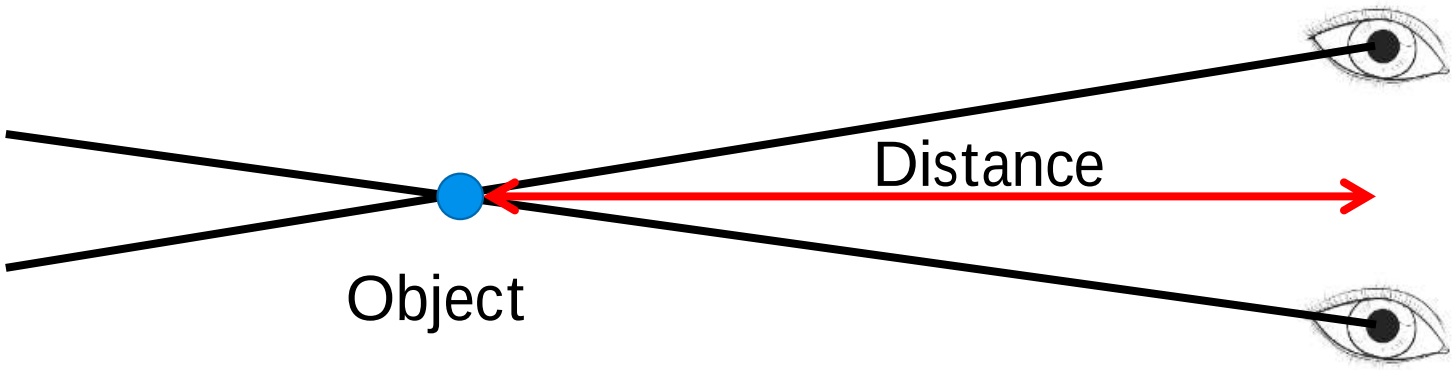


Circle

- Small segments always visible



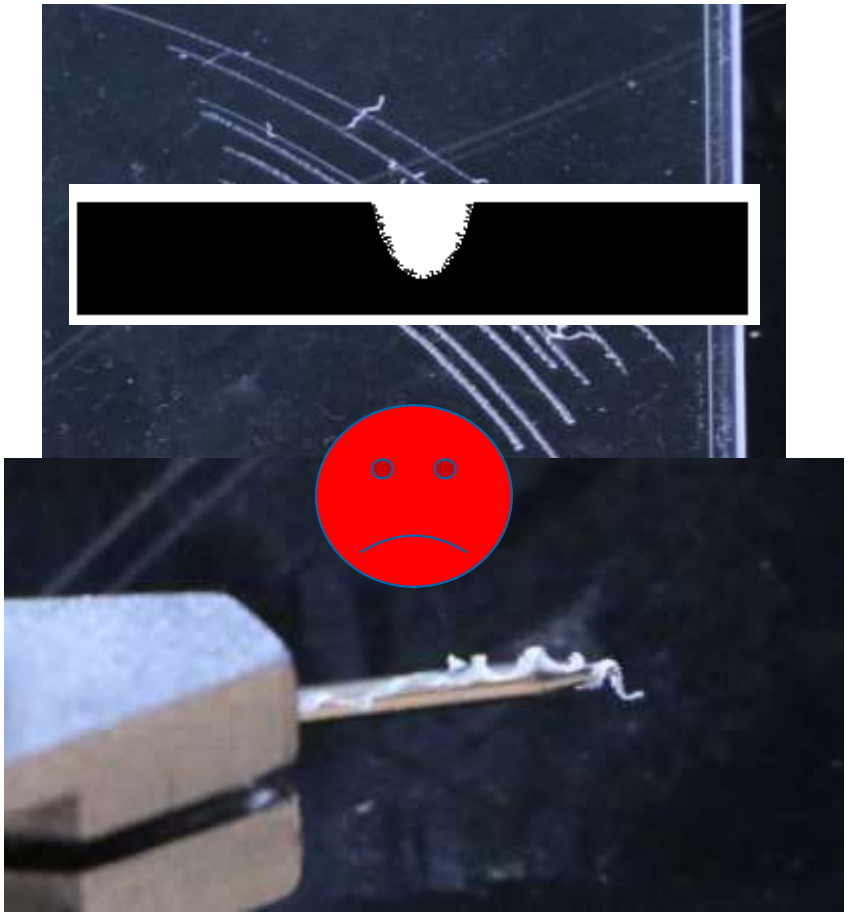
Human 3D vision mechanism

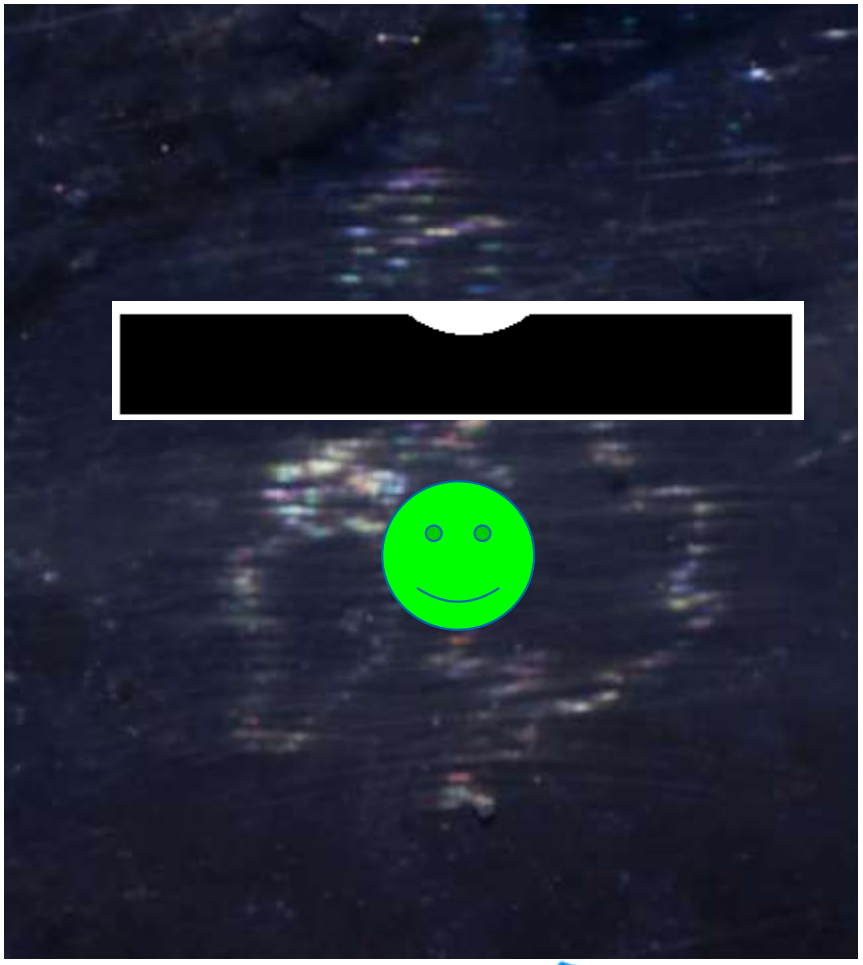


Position of shining point on circle

$$\tan\alpha = -\frac{\left(x_{light} - x_{point}\right)D_{eye} - \left(x_{eye} - x_{point}\right)D_{light}}{\left(y_{light} - y_{point}\right)D_{eye} - \left(y_{eye} - y_{point}\right)D_{light}}$$

Scratch on the plastic



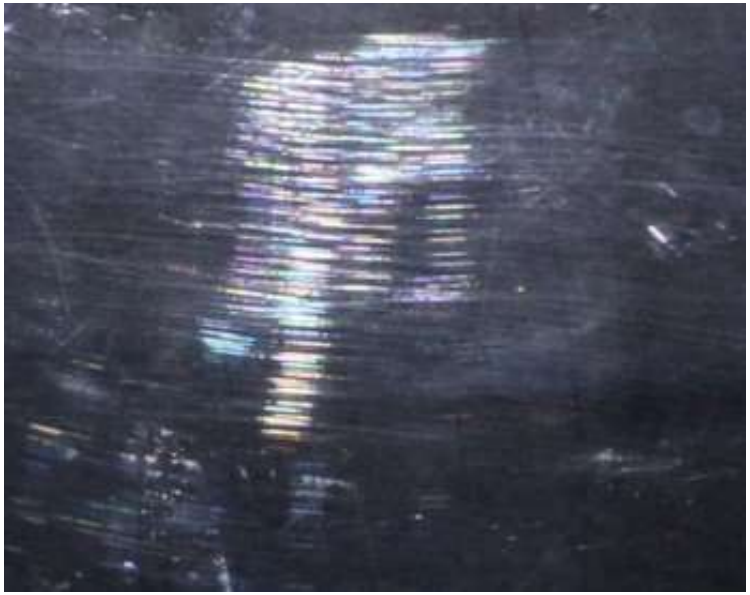


Radius of the circle sratch

We see section of a circle → radius ∝ length of the shining part

Small radius 13cm

Large radius 30cm



Distance vs. radius

Distance is propotional to radius

We can create straight line through space