CSC496: Deep learning in computer vision

Prof. Bei Xiao

Lecture 3: Image formation, Linear filtering

Slides mostly from
MIT 6.819 Lecture 2

Today's class

- Virtual box installation
- Image Formation with pin-hole camera; Lens
- Python/Numpy primer (finish the exercises in the tutorial)

- Homework 1: A simple vision problem: Due Wed September 18th.
- Homework 2: Pin-hole camera (out next week). Might need a real camera (however crappy). Library might be able to lend you one. Cell phone camera also works!
- https://www.instructables.com/id/Pointless-Object-Cell-Phone-Camera-Obscura/

Learning outcomes

- Be familiar using common computing tools to process, manipulating, extract information from images. Being able to describe basic image formation, measurements, and analysis.
- Understand the geometric relationship between 2d image and the 3D world.
- Be familiar with basic machine learning methods in context of computer vision.
- Gained experiences with one or two common deep learning frameworks (e.g. Tensor flow, Keras, PyTorch) and GPU computing (CUDA).
- Grasp the principles of state-of-the-art deep neural networks; and
- Developed practical skills to build computer vision applications

When do we get to deep learning?

- In about two weeks, September 26. Please be patient! And yes, you will learn A LOT about deep learning. But some foundation is very important!!

- Before that we need to cover some basics in vision:
  - 1. Image formation
  - 2. Linear filter (signal processing)
  - 3. Multiscale Pyramids
  - 4. Statistical Modeling of Images
  - 5. Get hands wet with Python/numpy/tensorflow

Today's class

- Image Formation with pin-hole camera
- Linear filters
- Virtual box installation
- Python/Numpy primer (finish the exercises in the tutorial)

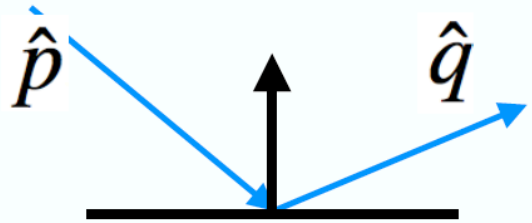
- Homework 1: A simple vision problem: Due Wed September 18th.

Imaging

- How do we form an image with pin hole?
- How do we form an image on lenses?

Light striking a surface

- Bidirectional reflectance distribution function

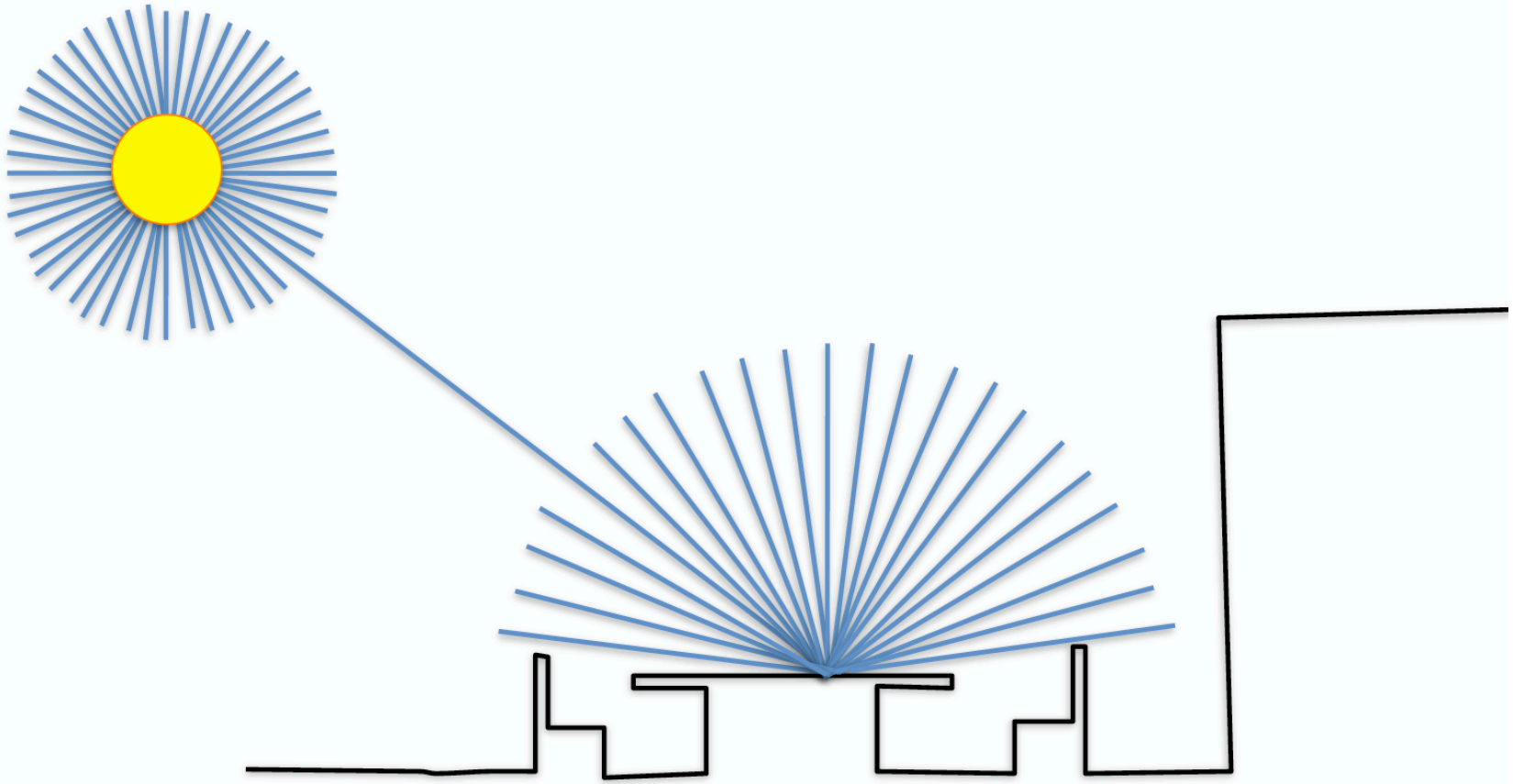


$$I_{\text{out}} = F(I_{\text{in}}, \hat{n}, \lambda, \hat{p}, \hat{q})$$

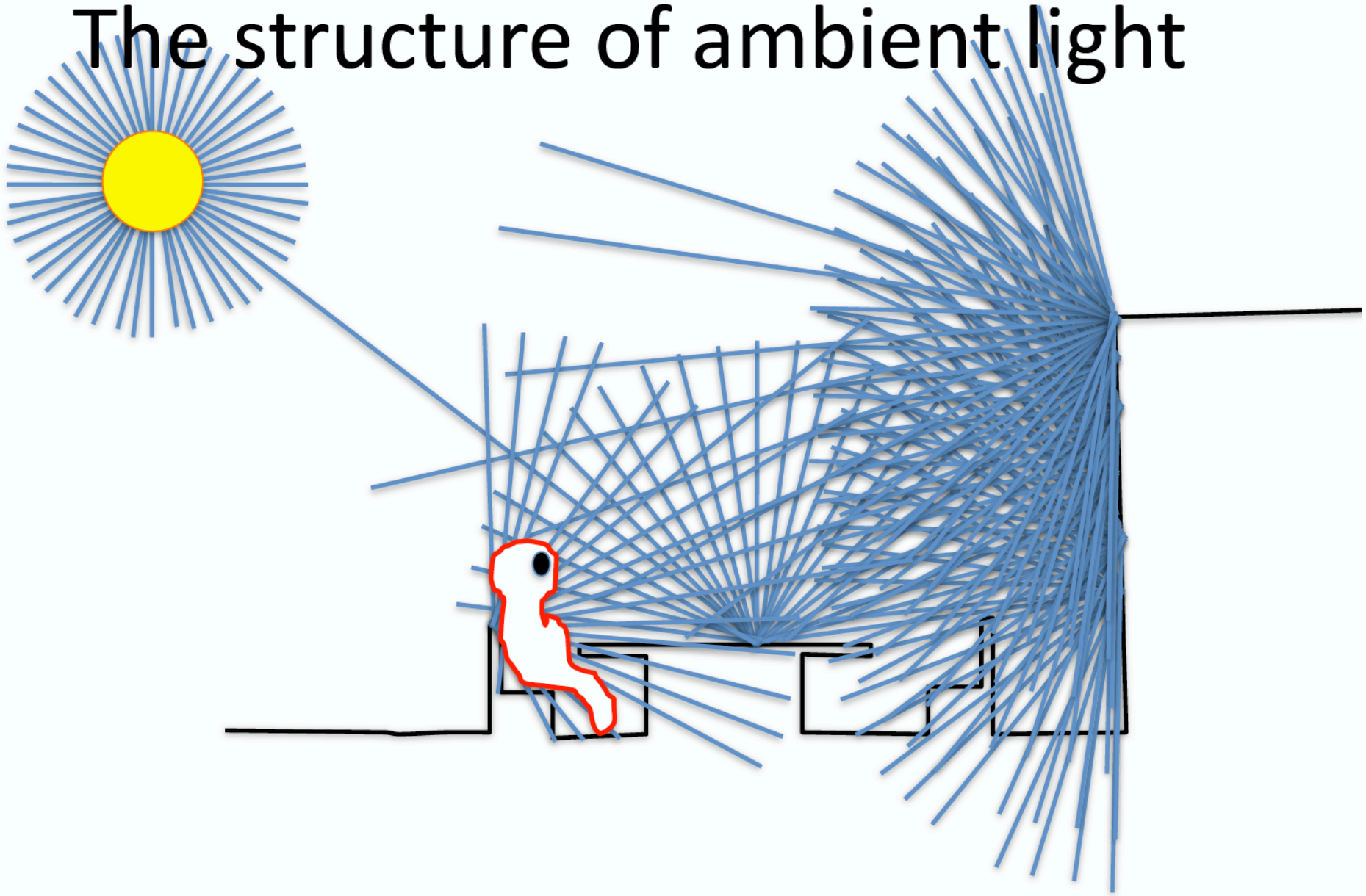
- For a Lambertian surface:

$$I_{\text{out}} = F_L(I_{\text{in}}, \hat{n}, \hat{p}) = AI_{\text{in}}(\lambda)\cos(\hat{n} \cdot \hat{p})$$

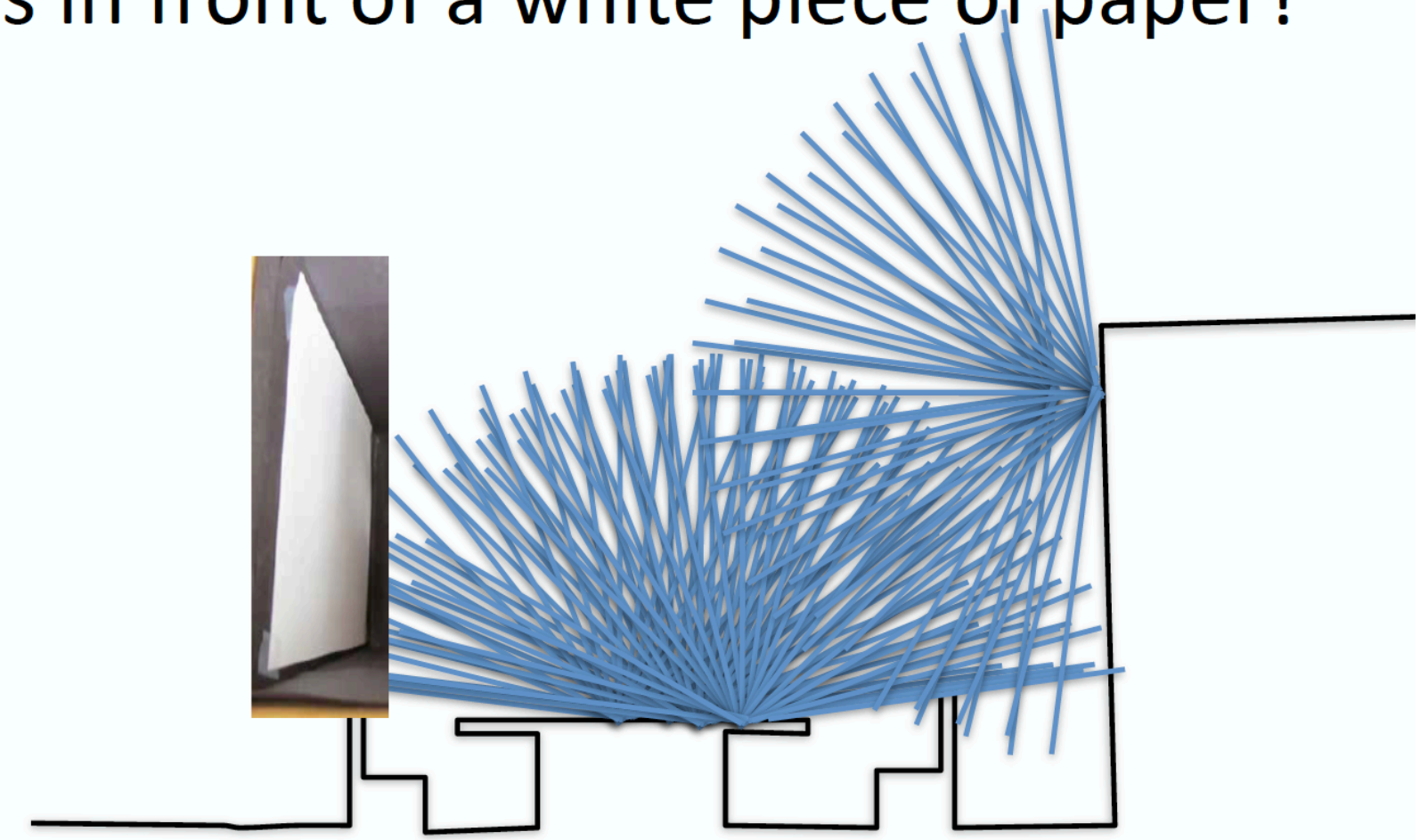
The structure of ambient light



The structure of ambient light

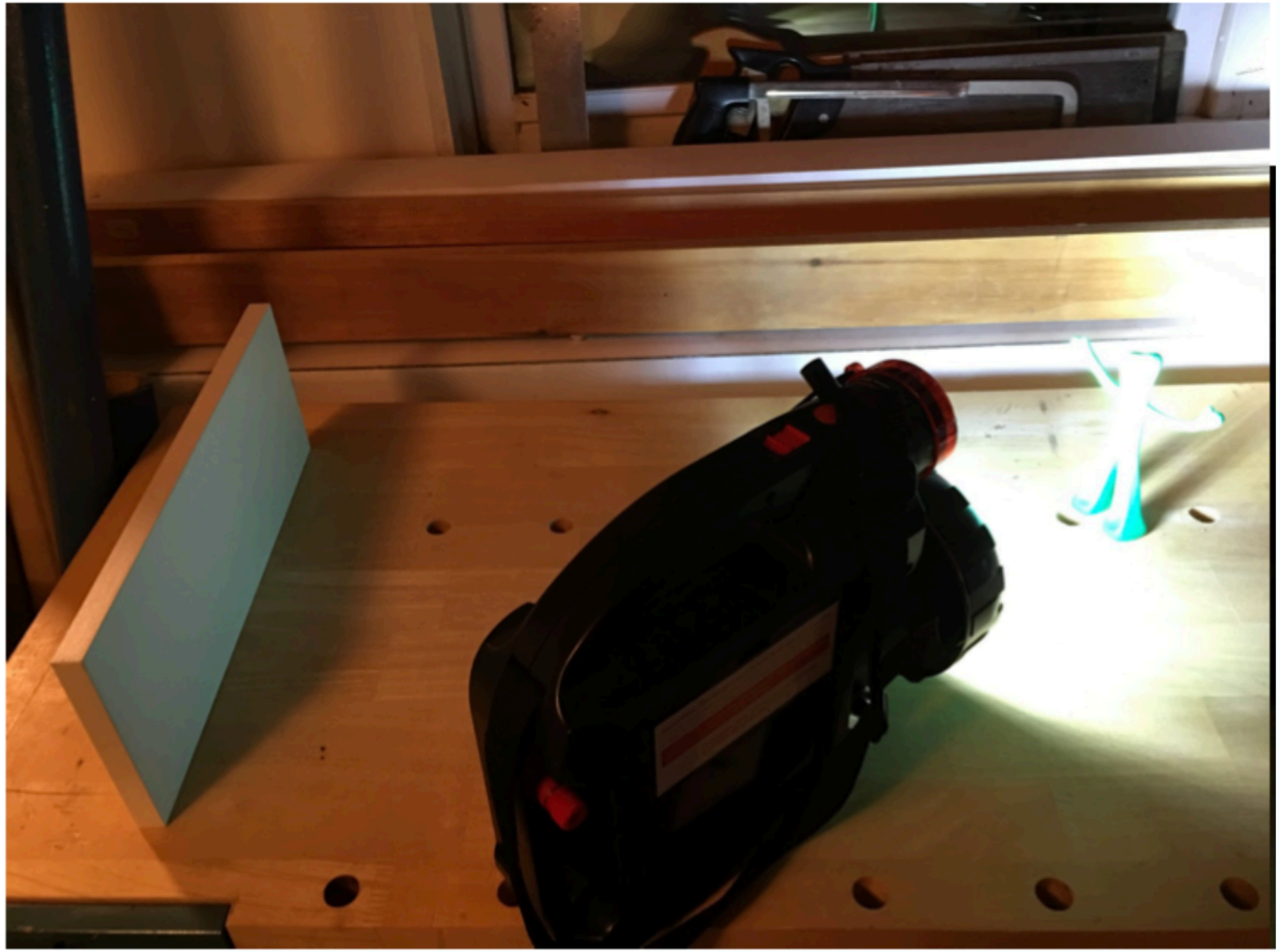


Why don’t we generate an image when an object is in front of a white piece of paper?



Why is there no picture appearing on the paper?

Let’s check, do we get an image?



Let’s check, do we get an image? No

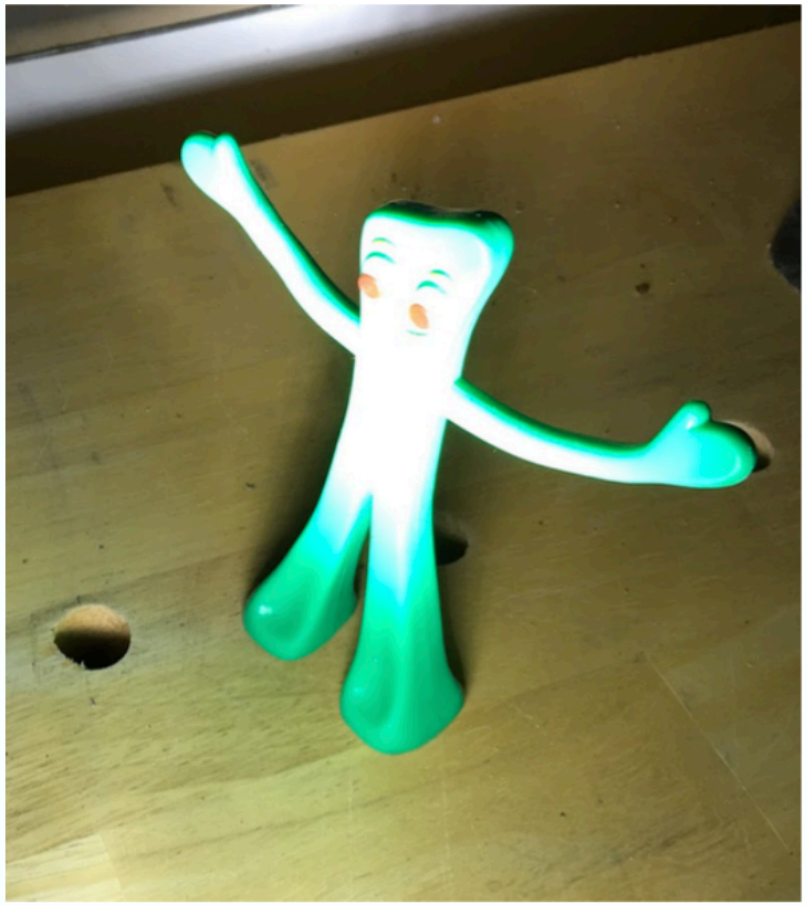
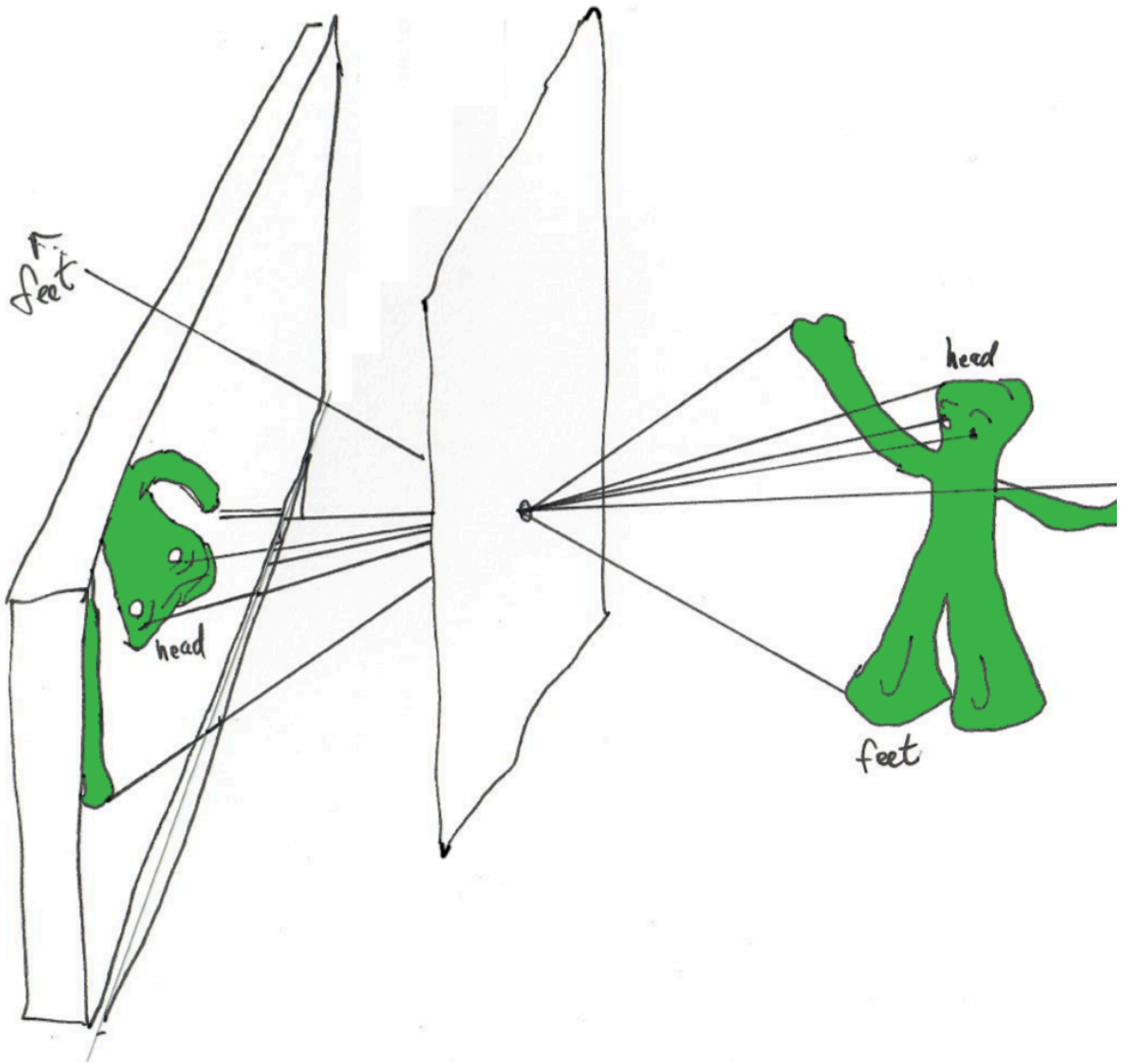
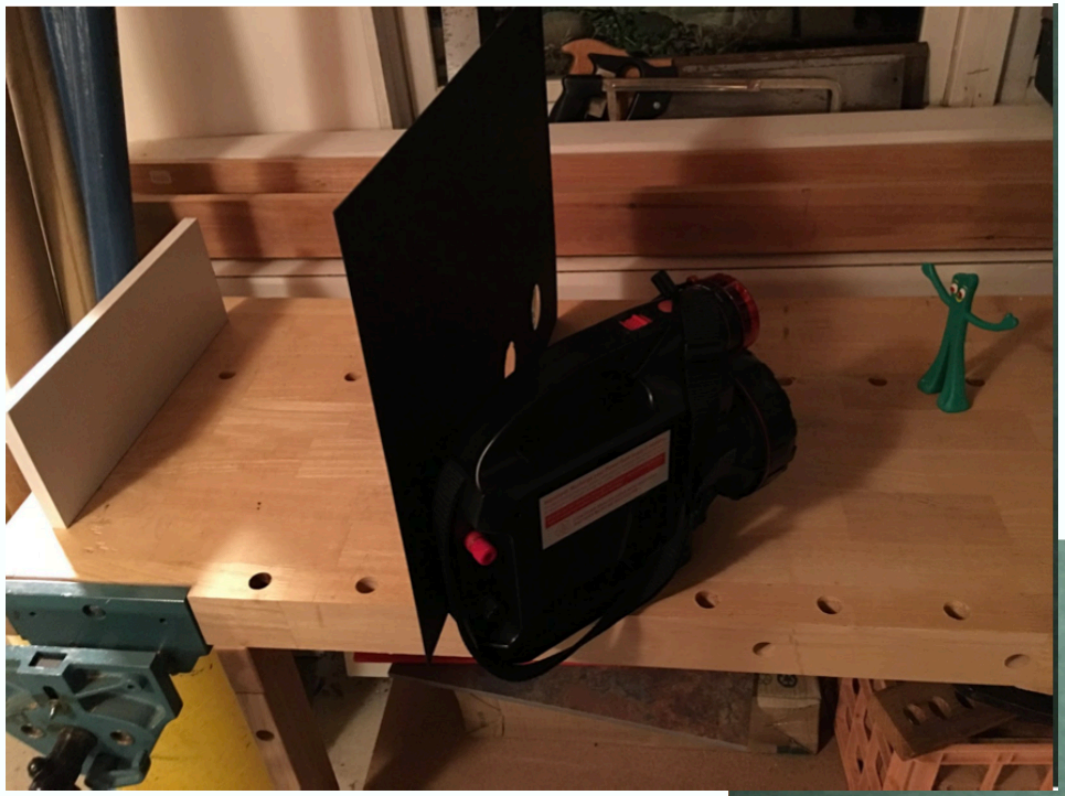
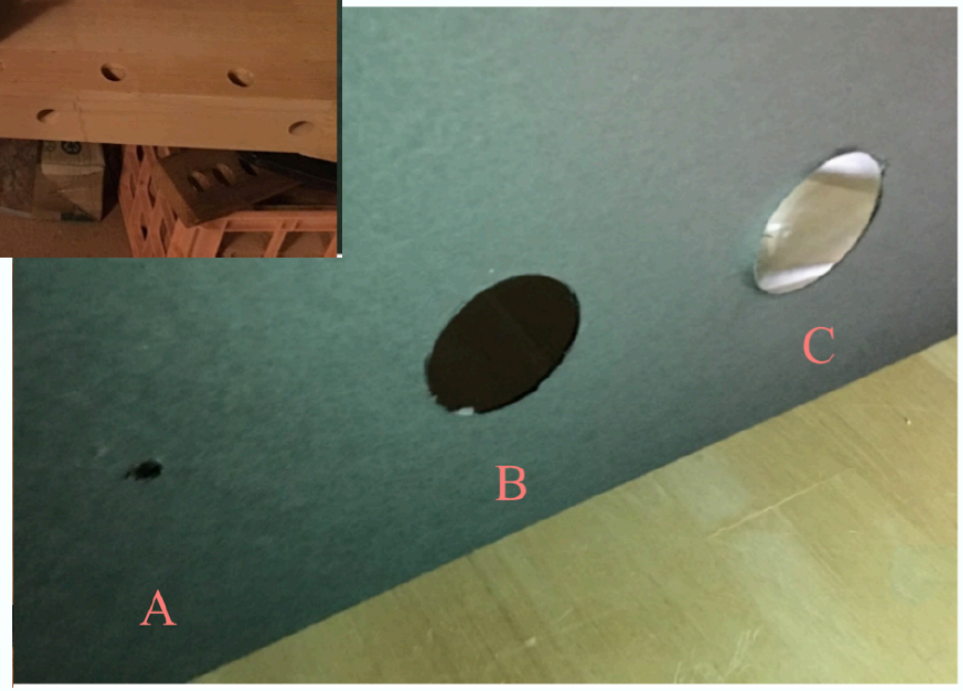


image is inverted



Let’s try putting different occluders in between





light on wall past pinhole

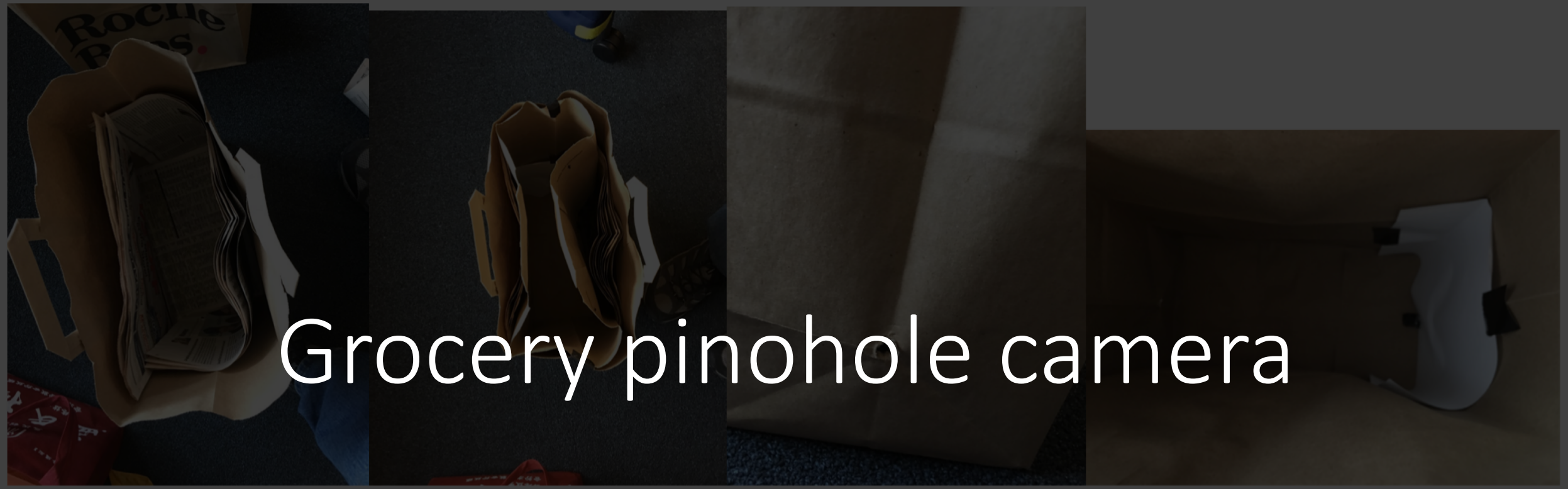
Ways to make a pin-hole camera

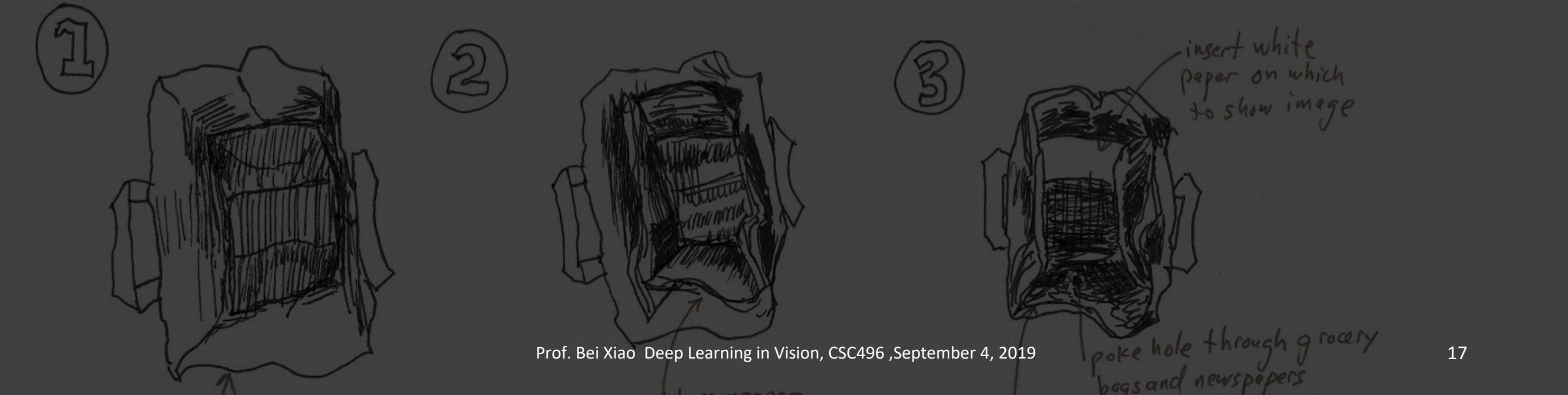
- Method 1:
  - Pick a box and poke a hole and remove a side for viewing
- Method 2:
  - The second is a paper bag, padded to be opaque, with a hole in it, inside of which one sticks his head.

Needed elements:

- An aperture to let light through, mechanisms to block stray light, projection screen, and some method to view or record the image on the projection screen (a camera)

Grocery pinohole camera





grocery bag pinhole camera

Grocery bag pinhole camera

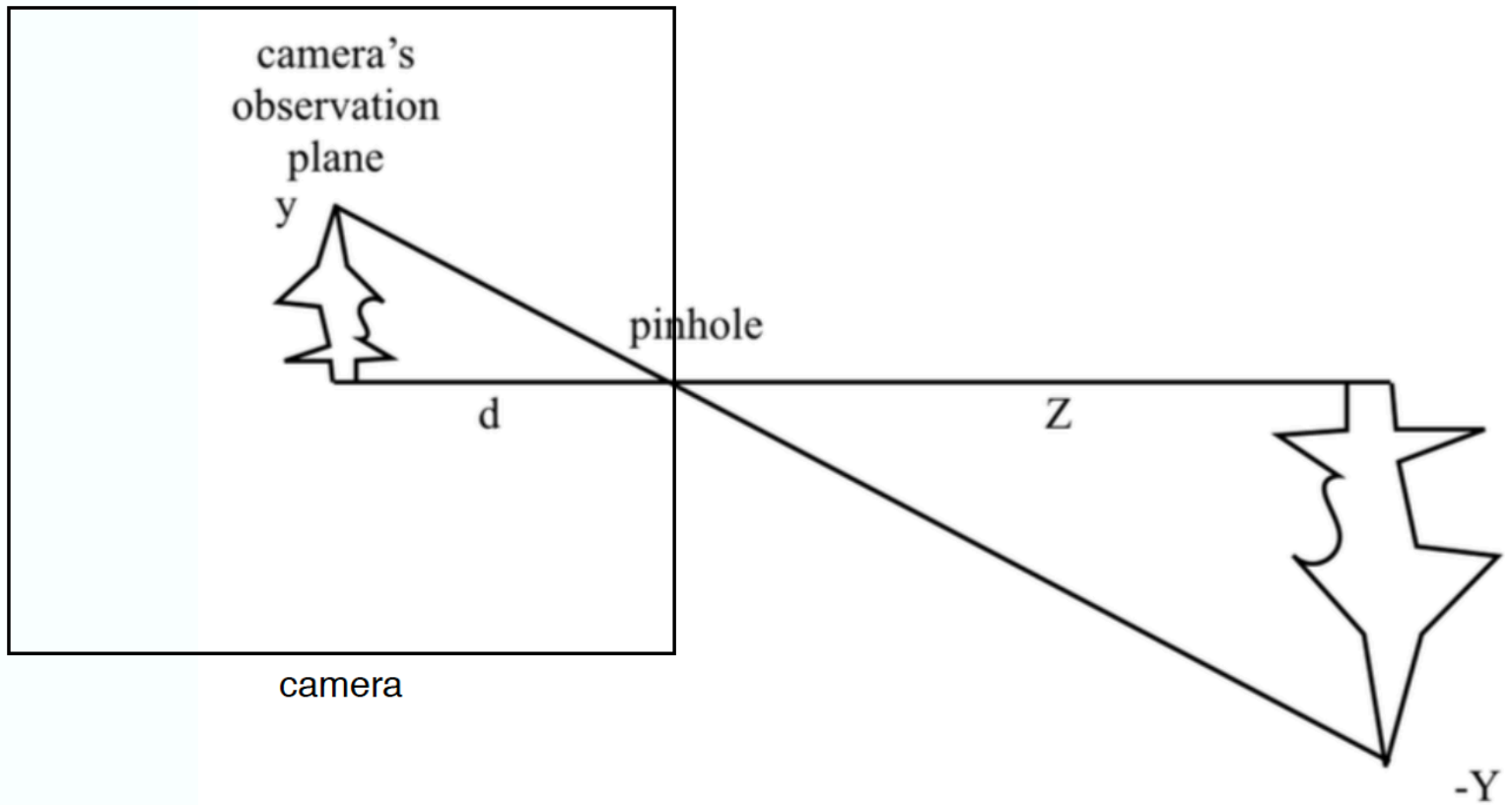
View outside the bag:

https://www.youtube.com/watch?v=FZyCFxsyx8o

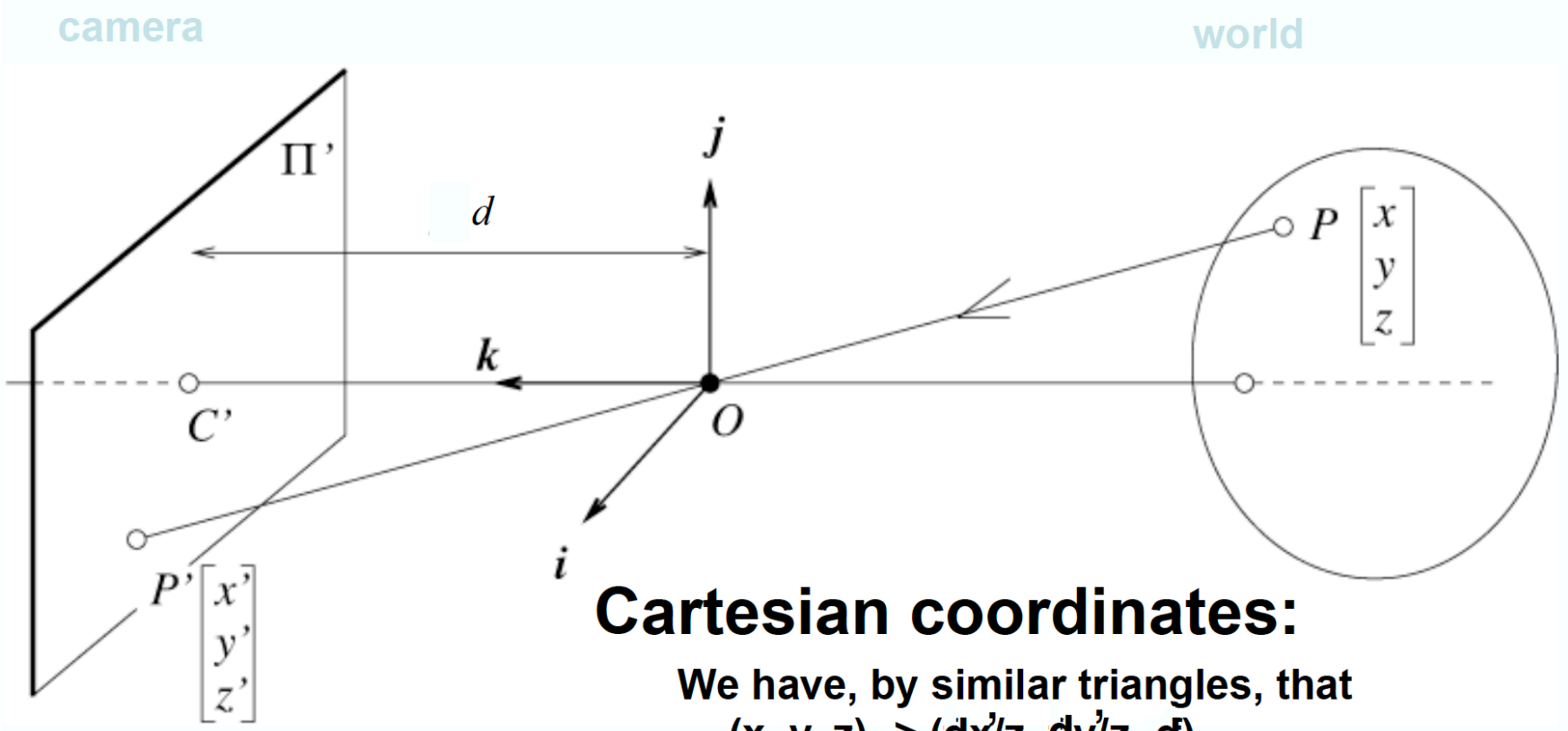
View inside the bag:

https://www.youtube.com/watch?v=-rhZaAM3F44&feature=youtu.be

Perspective projection



Perspective projection



Cartesian coordinates:

We have, by similar triangles, that
(x, y, z) -> (dx'/z, dy'/z, d)

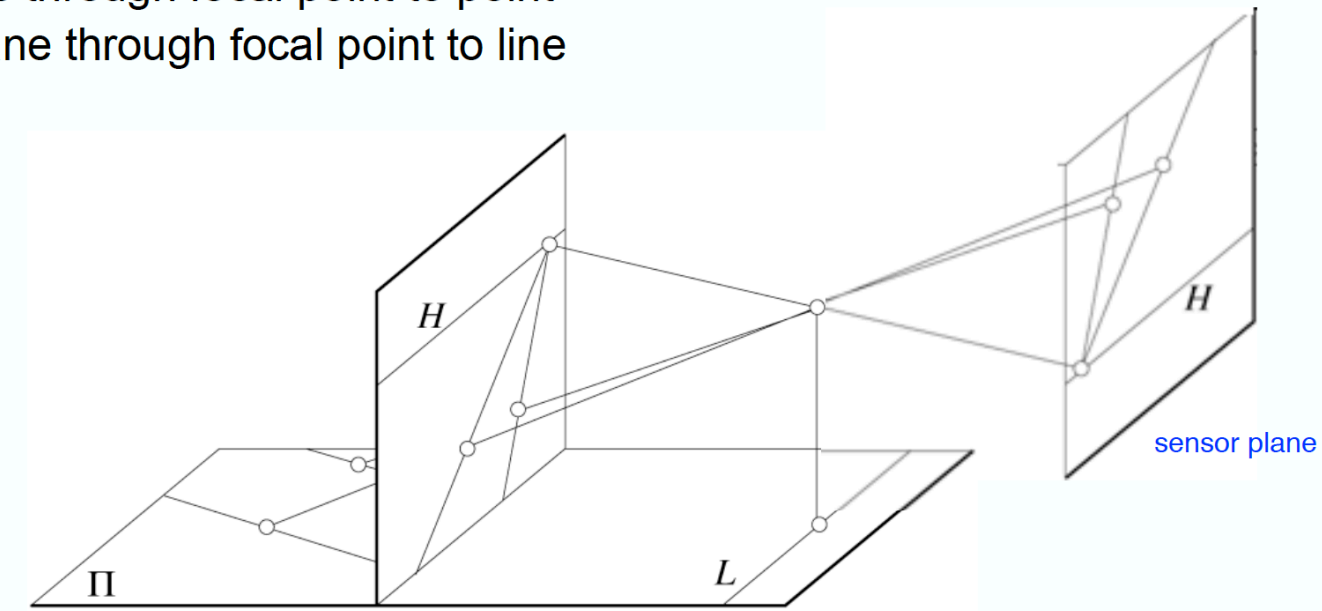
Ignore the third coordinate, and get

$$(x,y,z) \rightarrow (d\frac{x'}{z}, d\frac{y'}{z})$$

Figure from: Forsyth and Ponce.

Geometric properties of projection

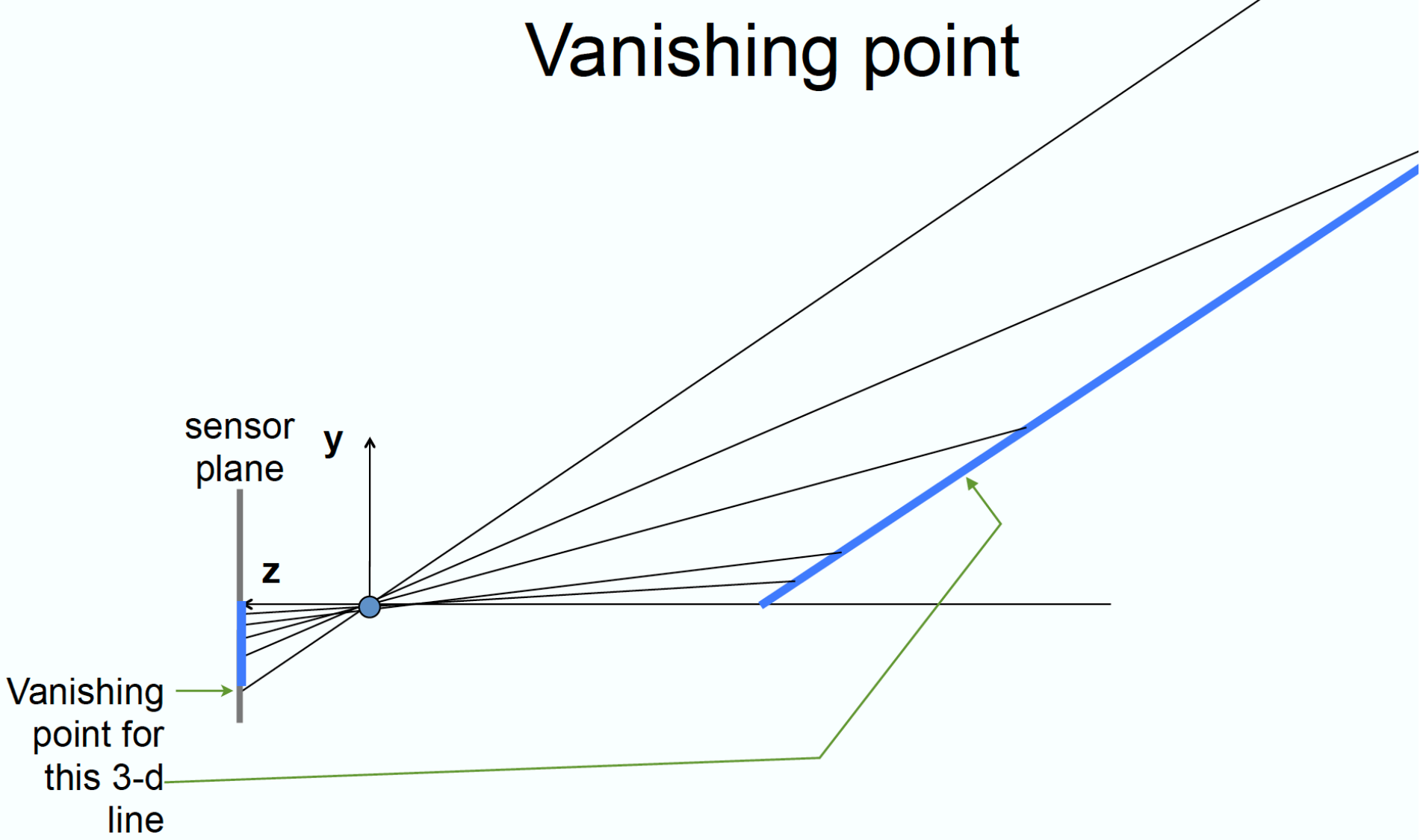
- Points go to points
- Lines go to lines
- Planes go to whole image or half-planes.
- Polygons go to polygons
- Degenerate cases
  - line through focal point to point
  - plane through focal point to line



you can also draw the sensor plane here, for simpler visualization

Figure from: Forsyth and Ponce.

Vanishing point



Line in 3-space

$$x(t) = x_0 + at$$
$$y(t) = y_0 + bt$$
$$z(t) = z_0 + ct$$

Perspective projection of that line

$$x'(t) = \frac{fx}{z} = \frac{f(x_0 + at)}{z_0 + ct}$$
$$y'(t) = \frac{fy}{z} = \frac{f(y_0 + bt)}{z_0 + ct}$$

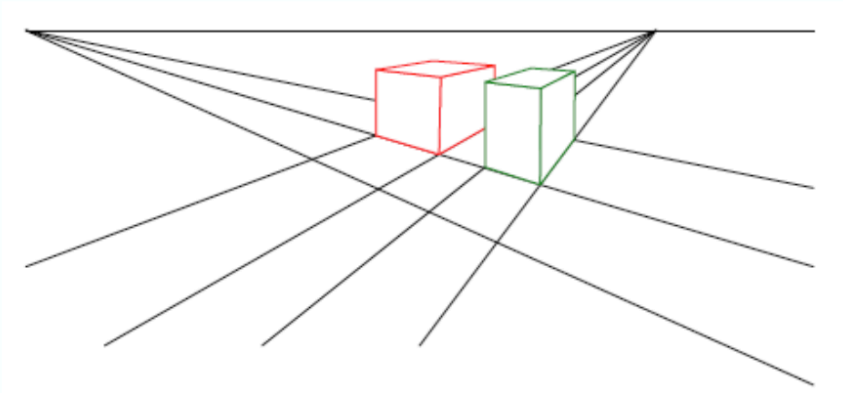
In the limit as $t \rightarrow \pm\infty$ **we have (for** $c \neq 0$**):**

$$x'(t) \longrightarrow \frac{fa}{c}$$
$$y'(t) \longrightarrow \frac{fb}{c}$$

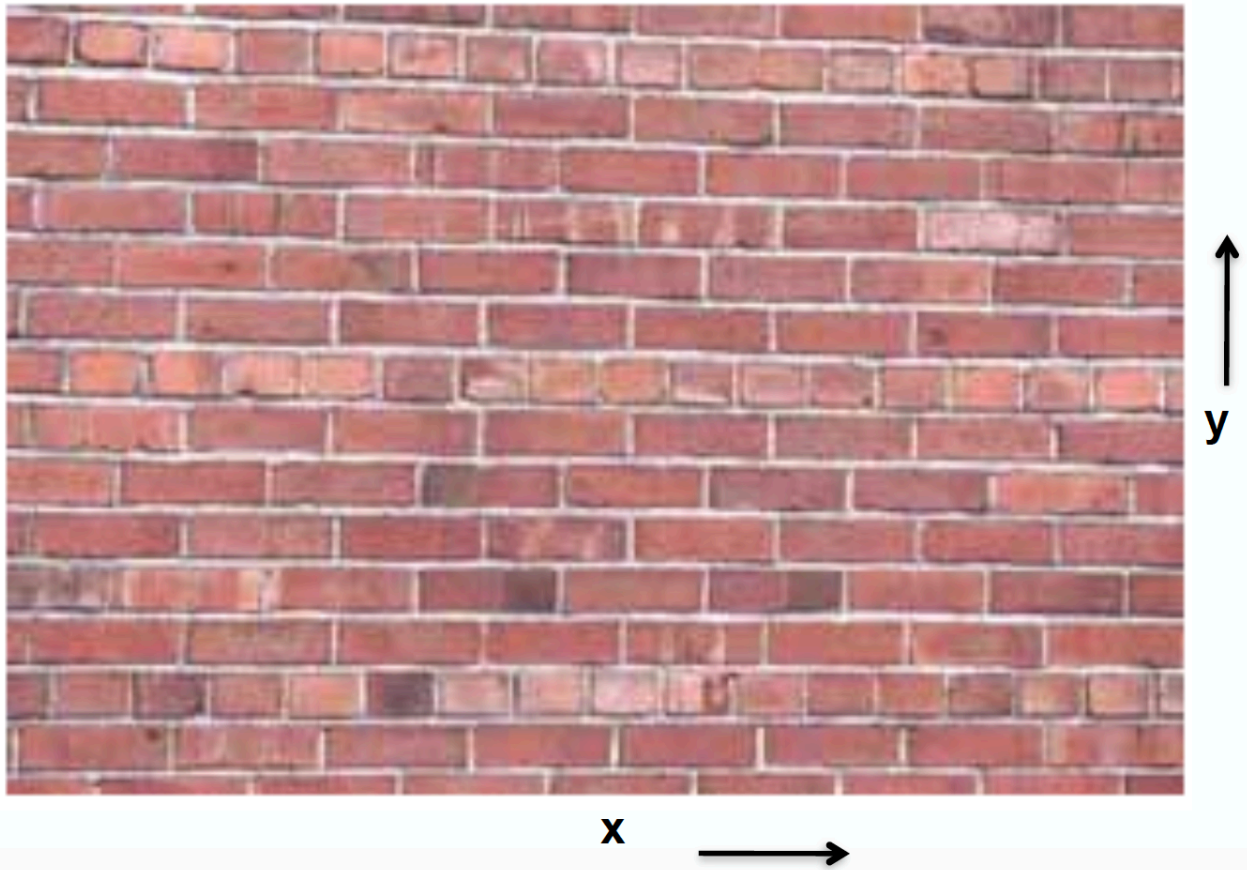
This tells us that any set of parallel lines (same a, b, c parameters) project to the same point (called the vanishing point).

Vanishing points

- Each set of parallel lines (=direction) meets at a different point
  - The *vanishing point* for this direction
- Sets of parallel lines on the same plane lead to *collinear* vanishing points.
  - The line is called the *horizon* for that plane



What if you photograph a brick wall head-on?



Brick wall line in 3-space

$$x(t) = x_0 + at$$

$$y(t) = y_0$$

$$z(t) = z_0$$

Perspective projection of that line

$$x'(t) = \frac{f \cdot (x_0 + at)}{z_0}$$

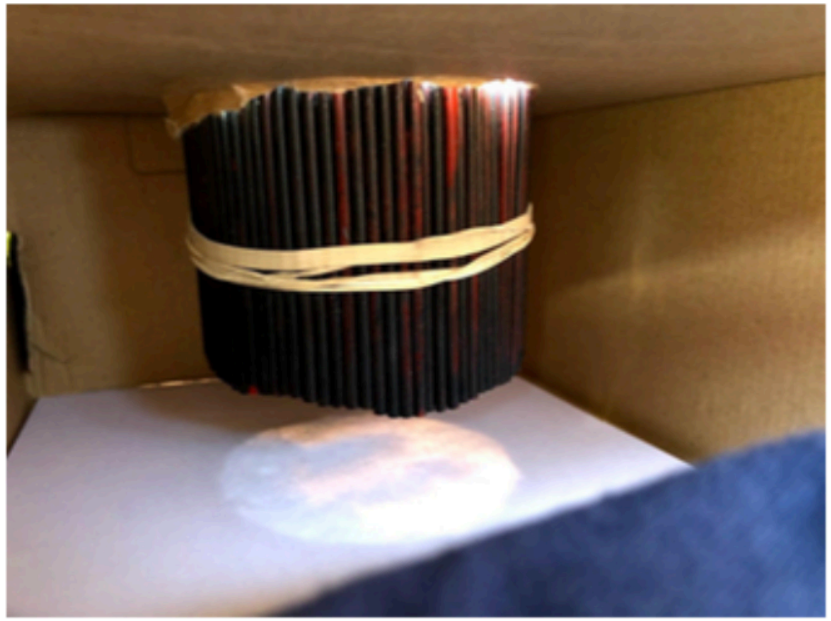
$$y'(t) = \frac{f \cdot y_0}{z_0}$$

All bricks have same z_0. Those in same row have same y_0

Thus, a brick wall, photographed head-on, gets rendered as set of parallel lines in the image plane.

Straw camera

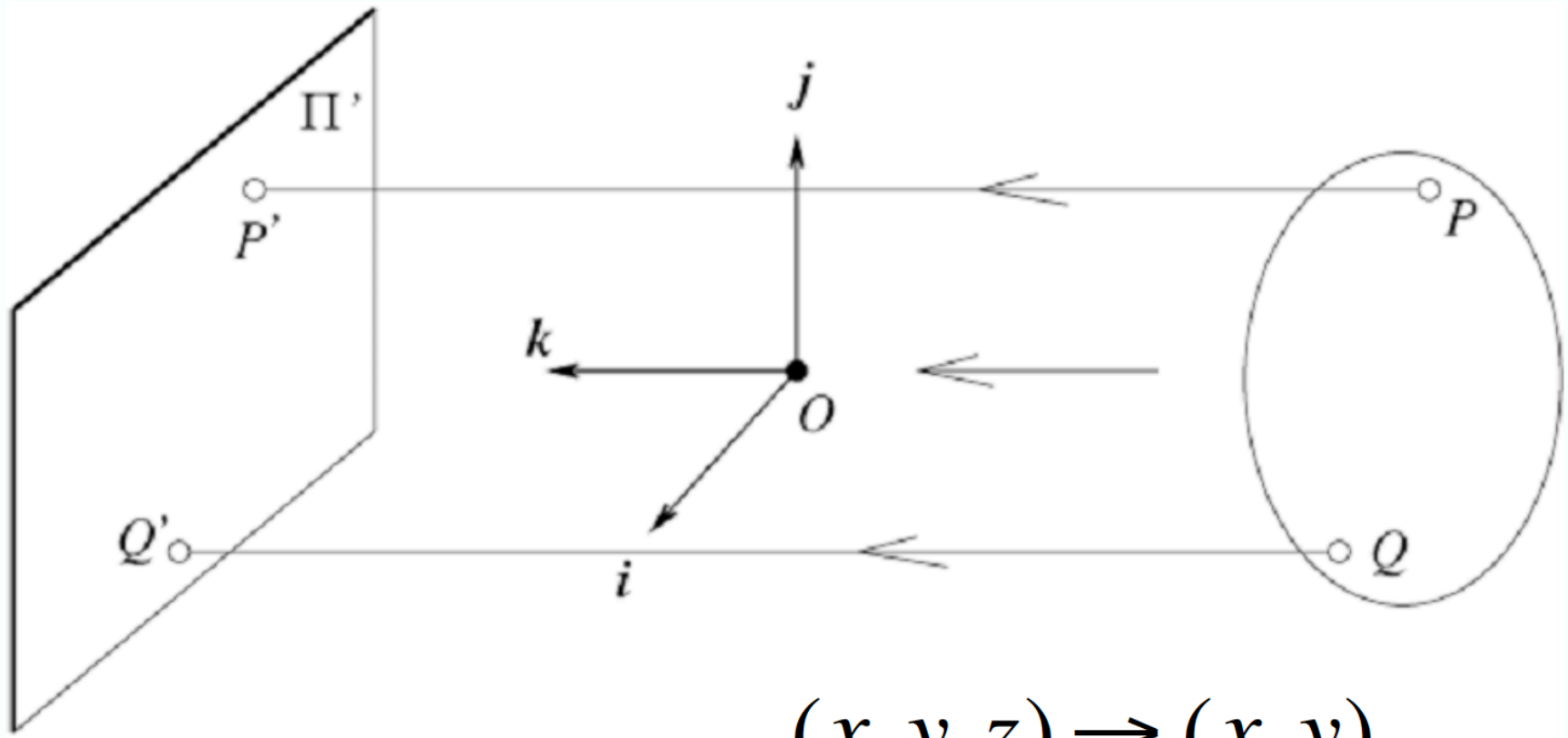
(a)



(b)

Straw camera

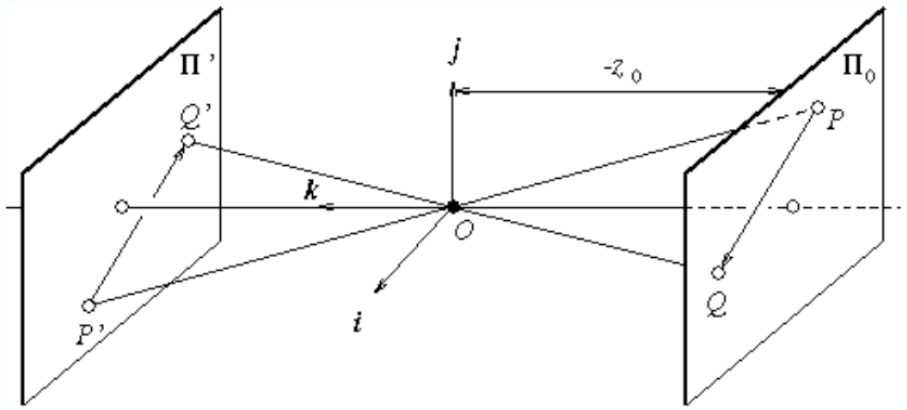
Other projection models: Orthographic projection



$$(x, y, z) \rightarrow (x, y)$$

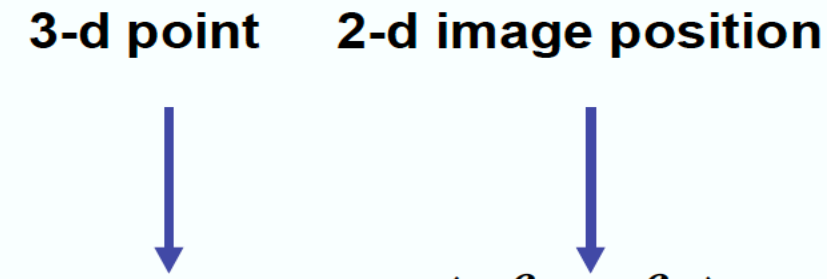
Other projection models: Weak perspective

- Issue
  - perspective effects, but not over the scale of individual objects
  - collect points into a group at about the same depth, then divide each point by the depth of its group
  - Adv: easy
  - Disadv: only approximate



$$(x, y, z) \rightarrow \left(\frac{fx}{z_0}, \frac{fy}{z_0} \right)$$

Three camera projections



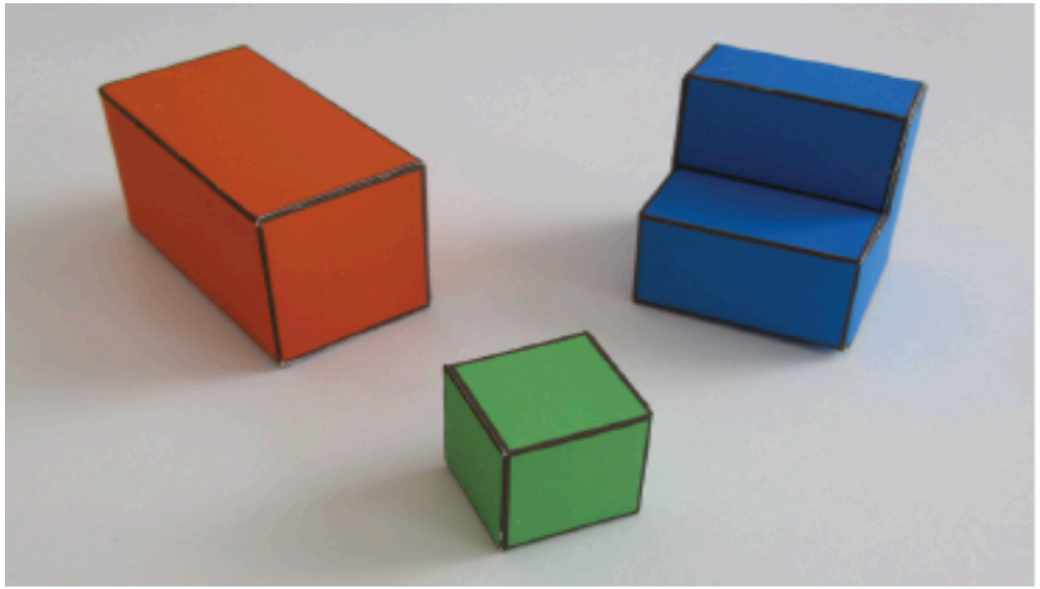
(1) Perspective: $(x,y,z) \rightarrow \left(\frac{fx}{z}, \frac{fy}{z}\right)$

(2) Weak perspective: $(x,y,z) \rightarrow \left(\frac{fx}{z_0}, \frac{fy}{z_0}\right)$

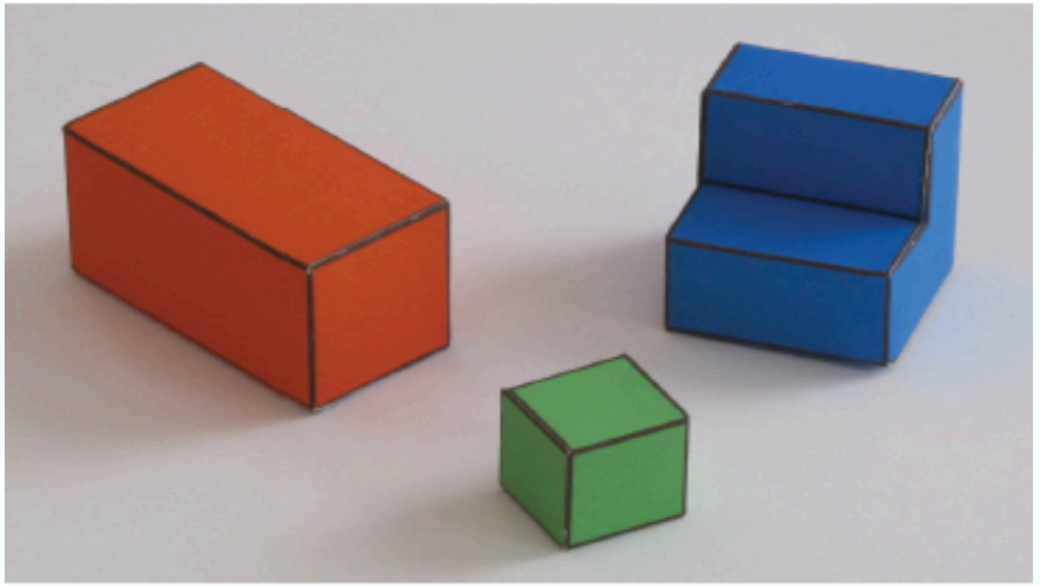
(3) Orthographic: $(x,y,z) \rightarrow (x,y)$

which is perspective, which orthographic?

Perspective projection

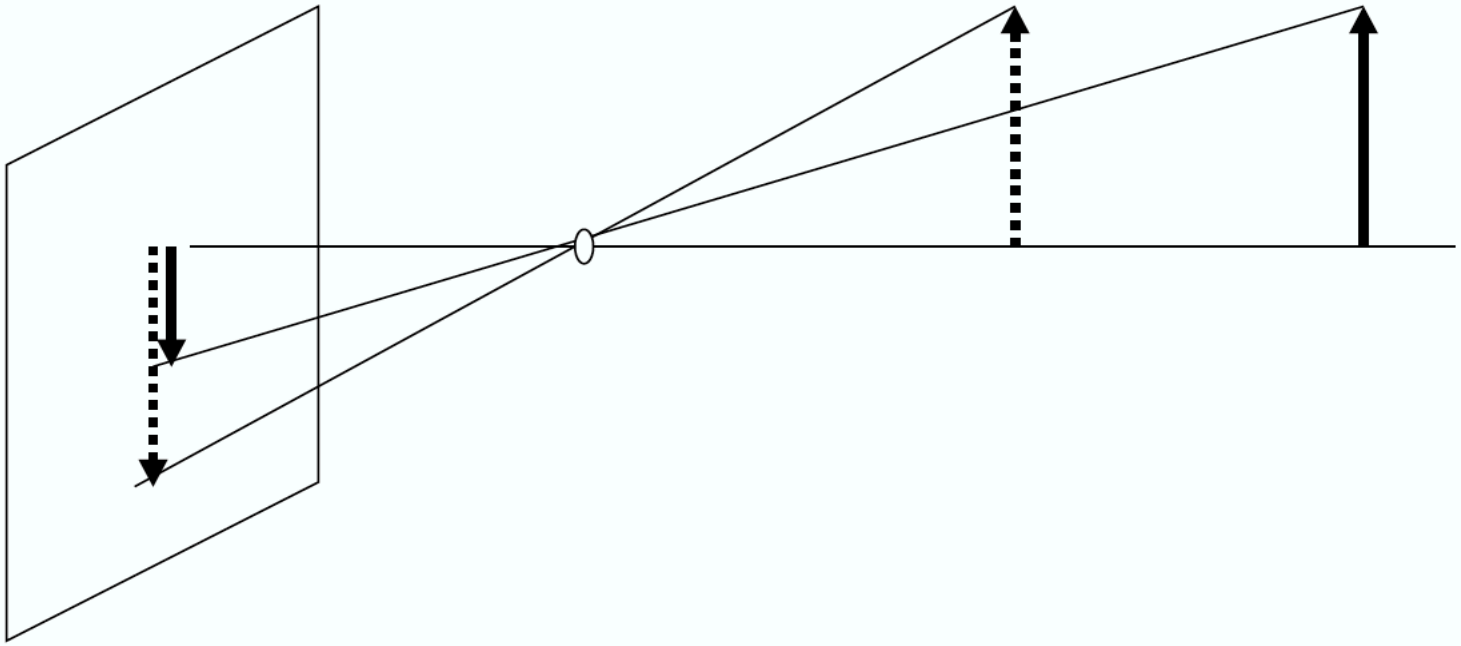


Parallel (orthographic) projection



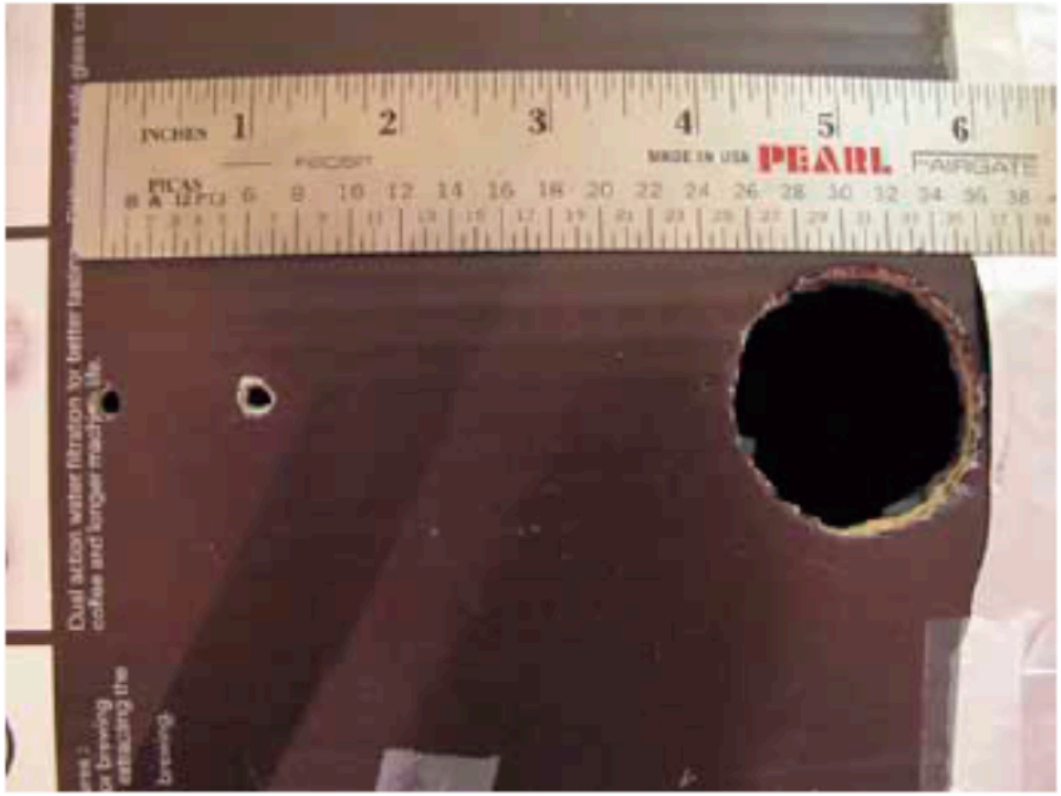
Example images from pinhole camera

Measuring distance

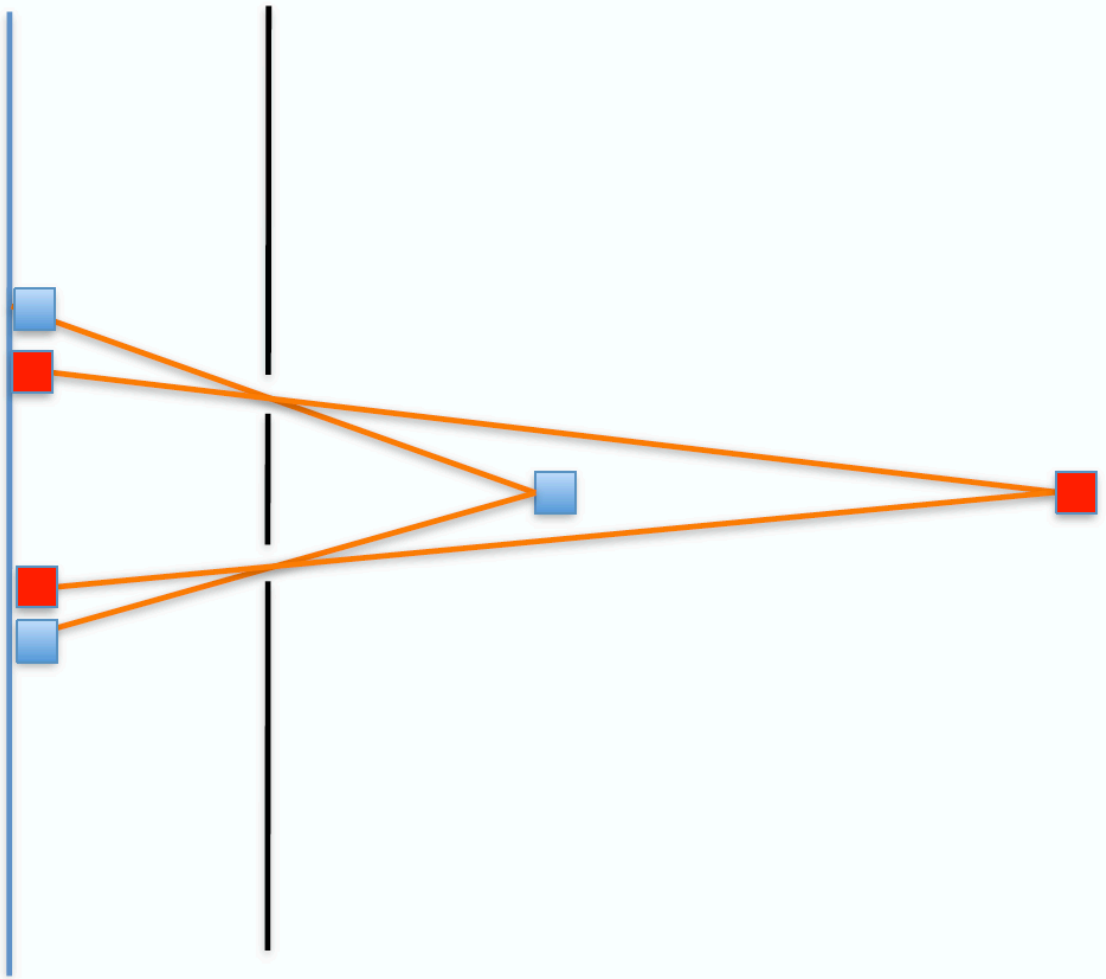


- Object size decreases with distance to the pinhole
- There, given a single projection, if we know the size of the object we can know how far it is.
- But for objects of unknown size, the 3D information seems to be lost.

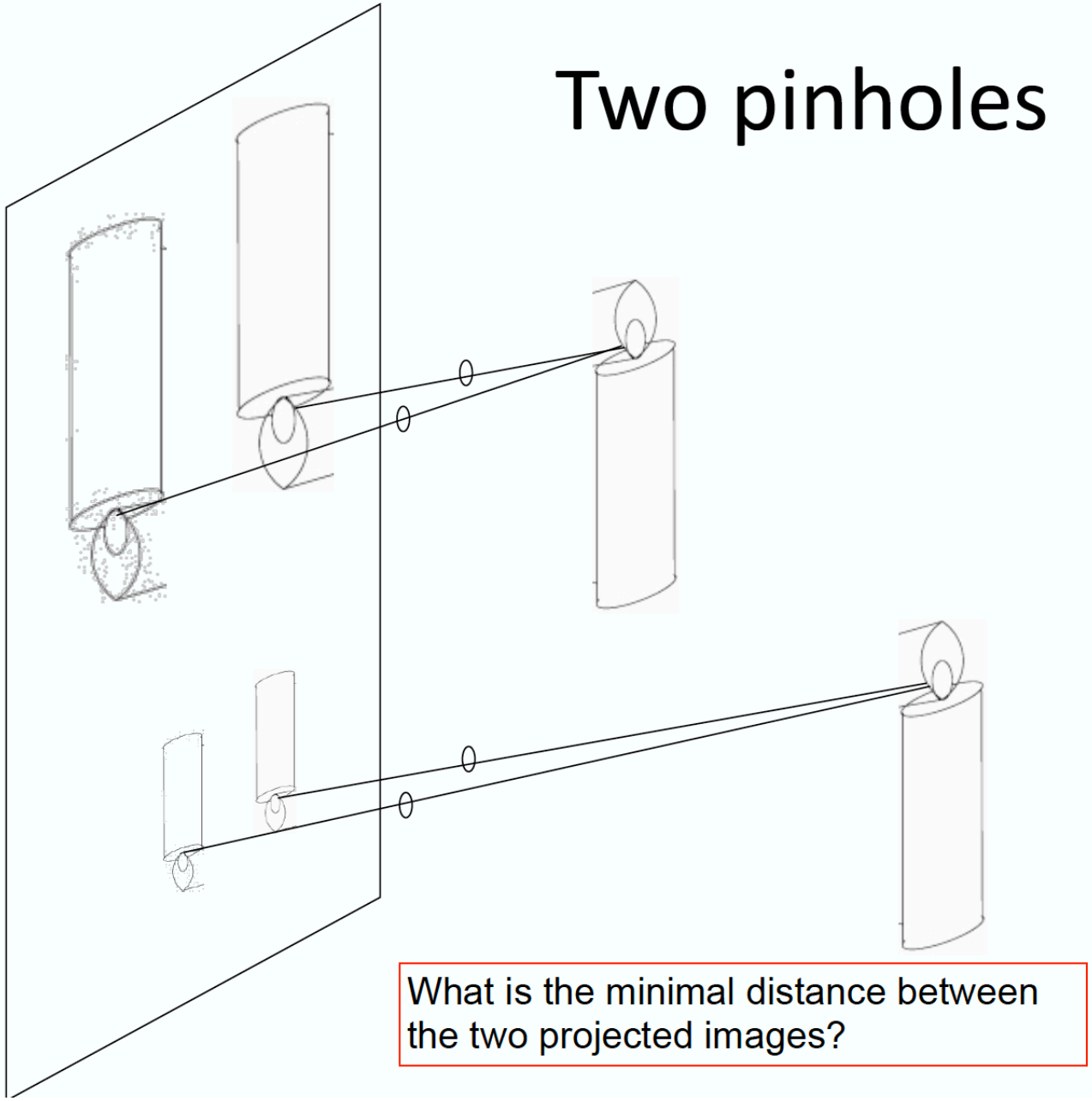
Playing with pinholes



Two pinholes



Two pinholes

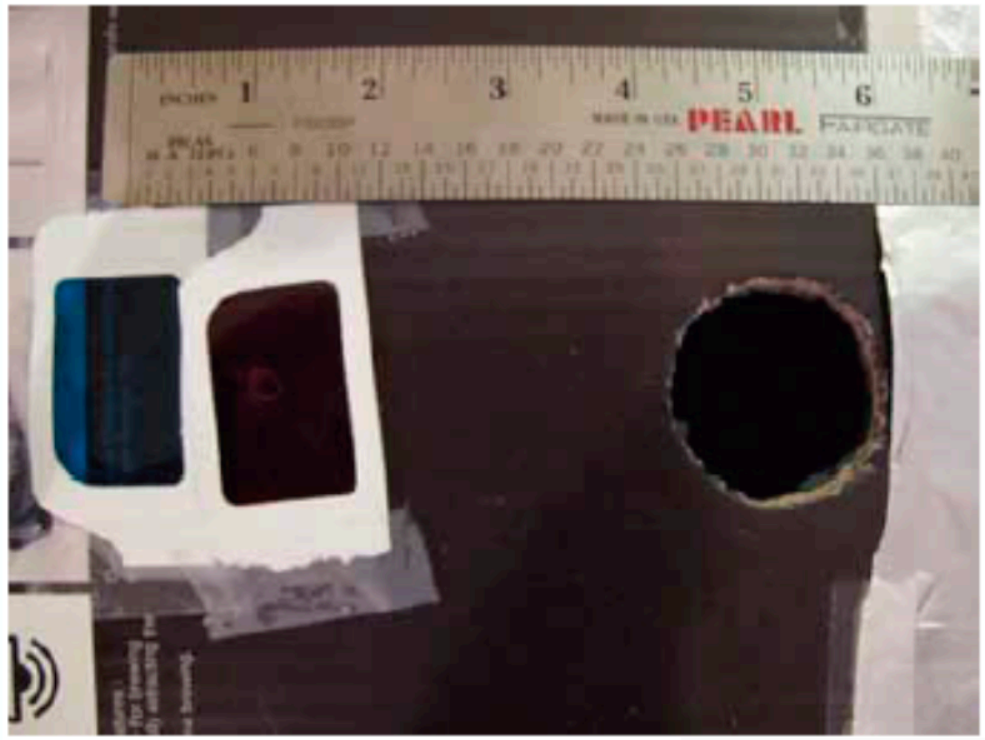


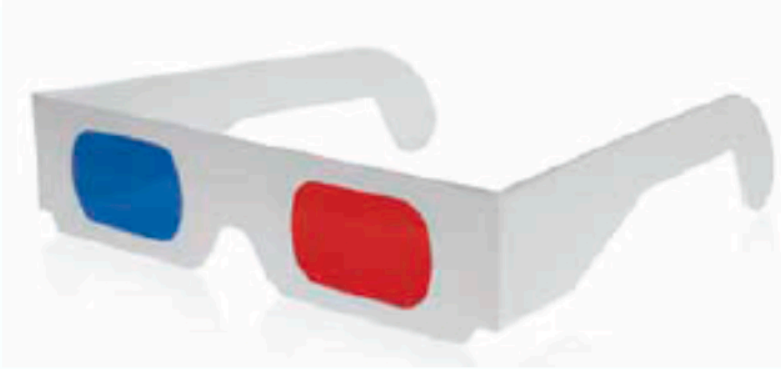
What is the minimal distance between the two projected images?

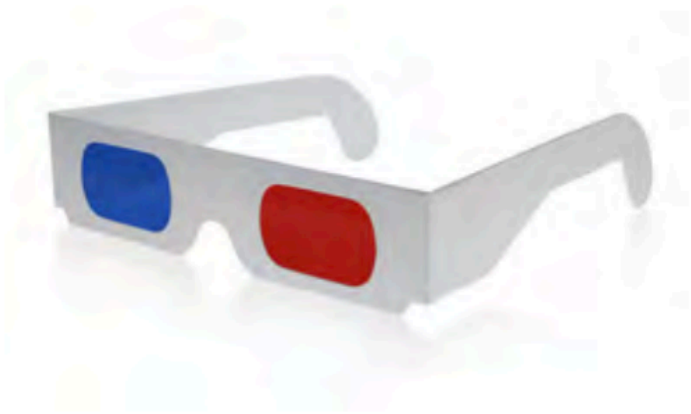
Anaglyph image

Anaglyph images are a particular method to produce 3D images. Anaglyph images are formed by superimposing a pair of stereo images into two different color channels. They provide a 3D effect when viewed with color glasses.

Anaglyph pinhole camera

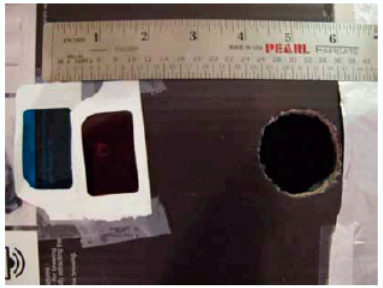






We used one of the glasses to get the filters and placed them in front of the two holes. Figure 3 shows two pictures of the two holes (top left) and the filters positioned in front of each hole (top right). On the bottom of that figure is a resulting 3D image, with the two views superimposed. That image should give you a 3D percept when viewed with anaglyph glasses.

Anaglyph pinhole camera



front of camera

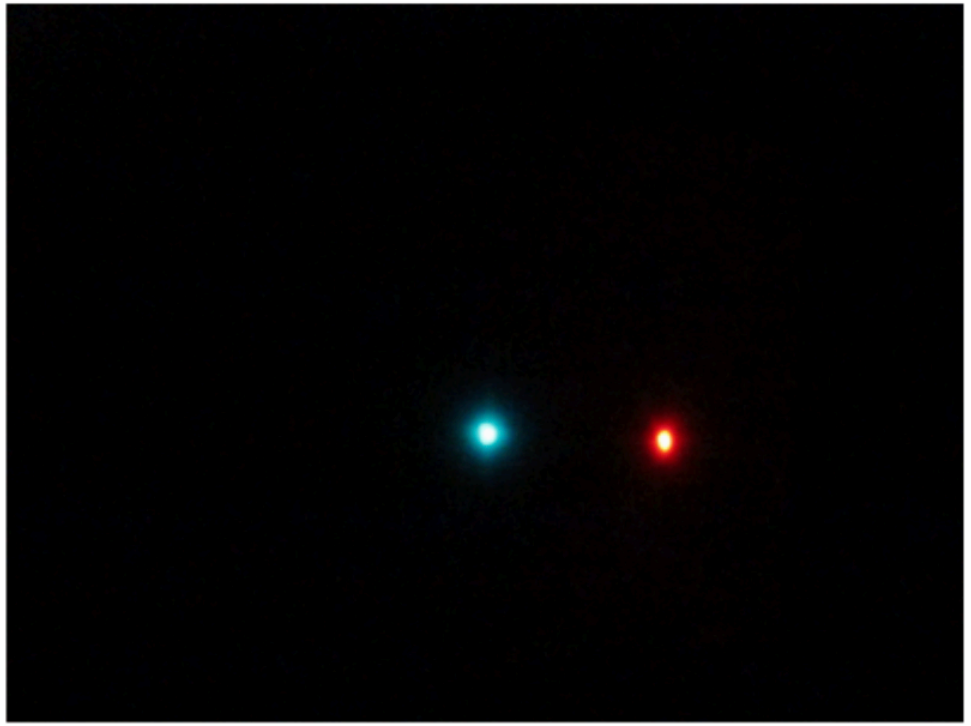


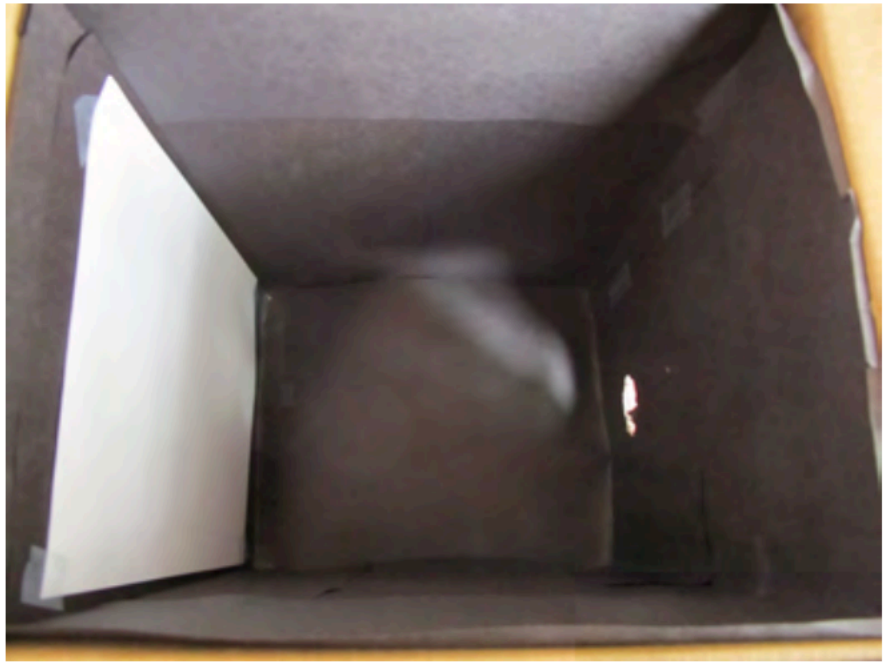
image of a point of light

Anaglyph pinhole camera

Problem set 2: Make your own pin-hole camera

- Build the camera: e.g. shoe box, make sure no light is leaking through.
- Decide where to put the hole: Make sure that the distance between the pinhole face and the screen face -- the focal length of your pinhole camera -- is not less than the minimum focus distance of your camera.
- Make pinkhole(s): I recommend cutting a larger hole in the cardboard box and attaching a thinner sheet of card stock, and having holes in the card stock.
- Cover the inside wall facing the pinhole with white copy paper.
- Create a hole for the digital camera

Problem set 2: Make your own pin-hole camera



Take-home reading and next class

- Lecture notes: cameras.pdf (uploaded on blackboards)
- Next class: linear filter and convolution