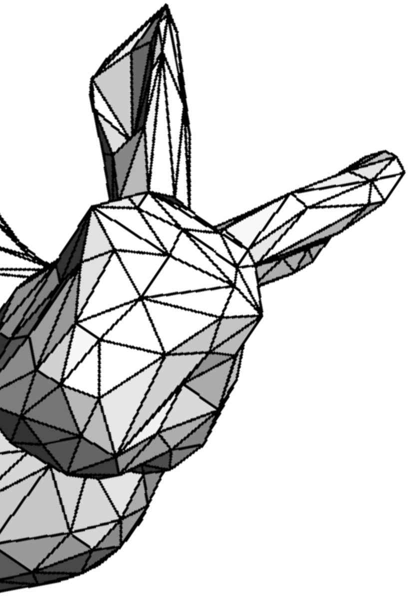
Introduction to Visualization and Computer Graphics

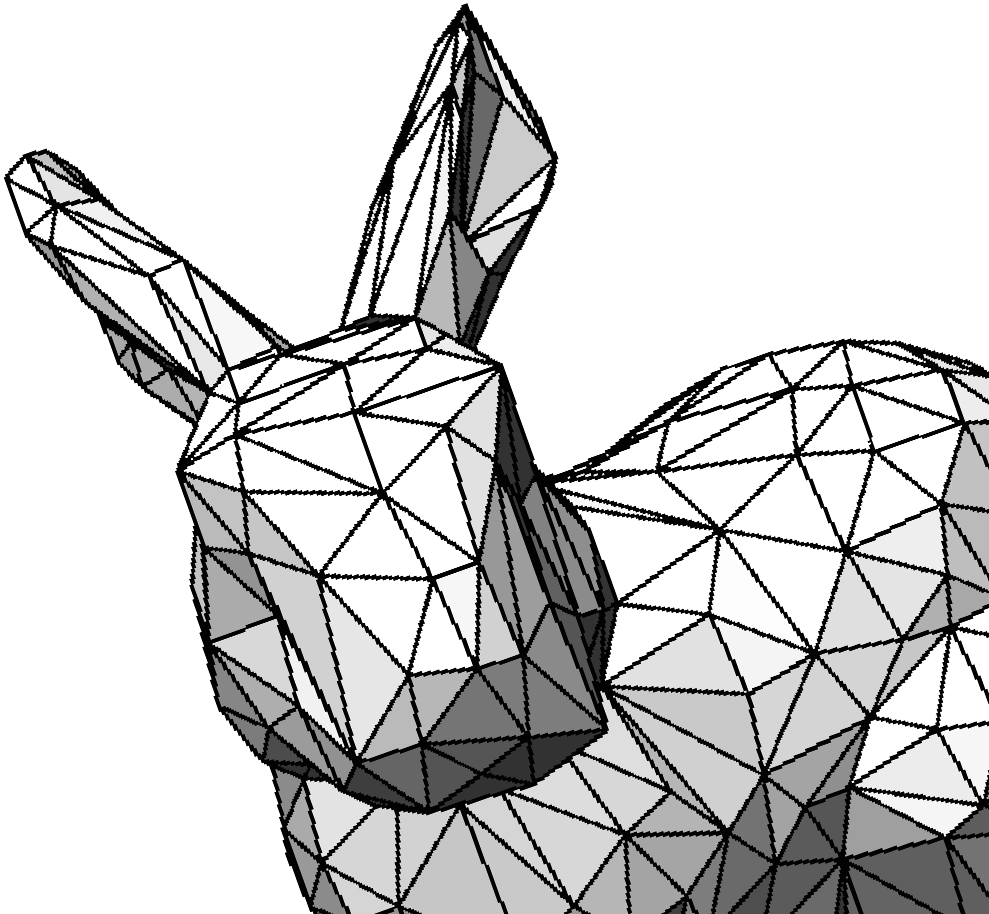
Color
Projection

- Thursday, 14 January 2016, at 08:00 - 10:00
- Location: V2, V3, V32
- 4 hand-written pages allowed

Now for

3D Rendering



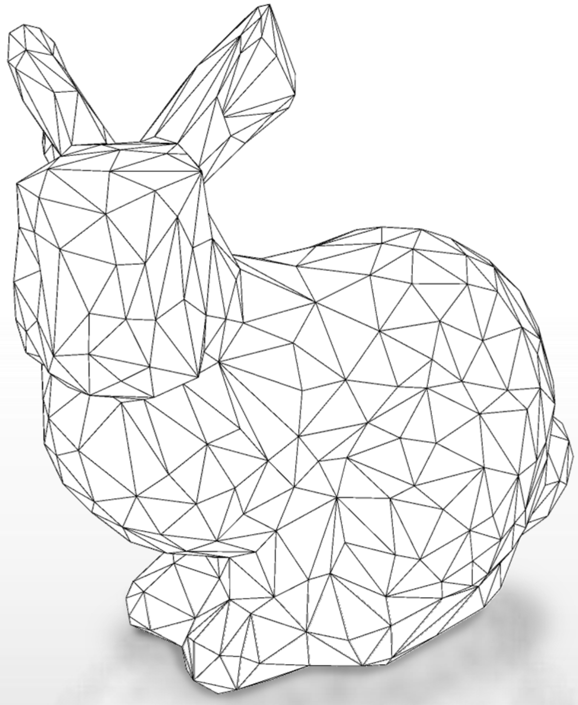


3D Rendering

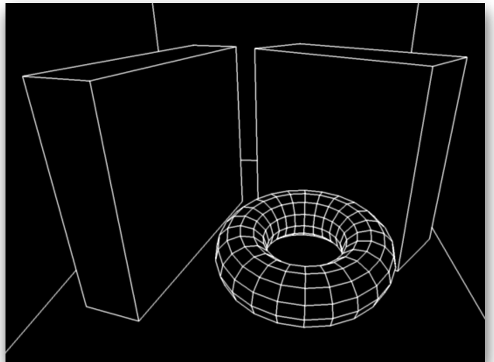
Assumption

- 3D Model is given
- Triangle mesh
  (for simplicity)

How do we get it to the screen?

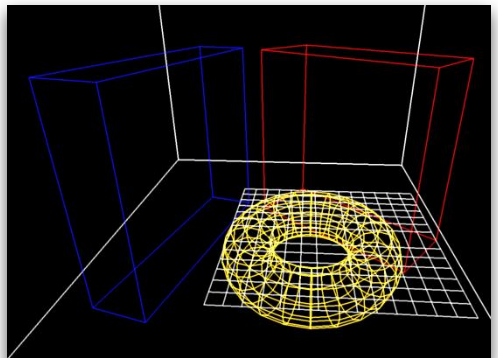


3D Rendering

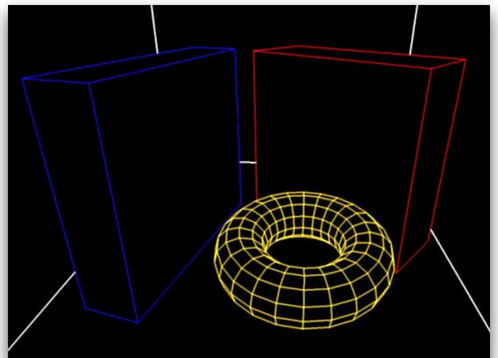


Geometric Model

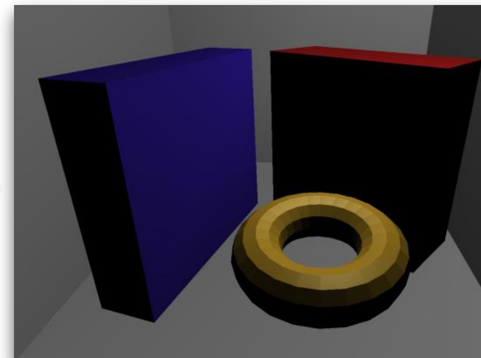
Color



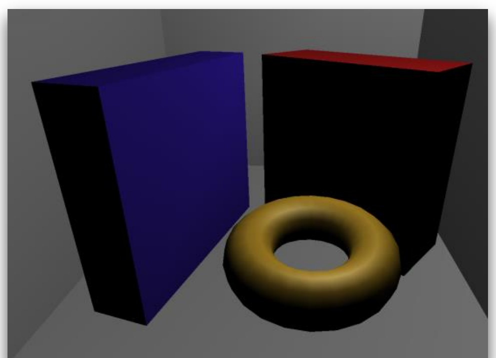
Perspective



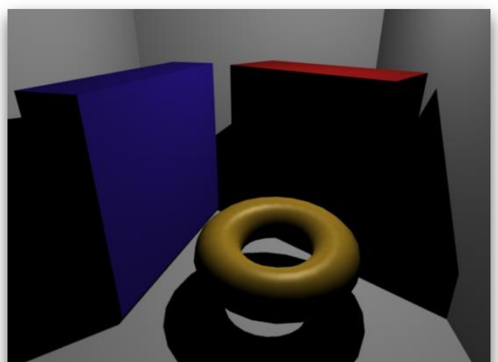
Visibility



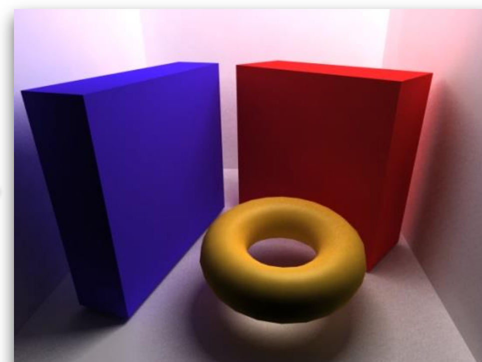
Local Illumination



Smooth Shading

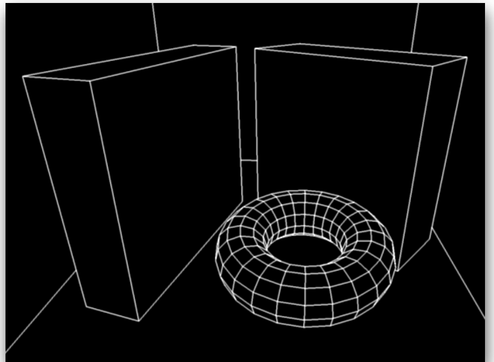


Simple Shadows



Global Illumination

D Rendering



Geometric Model

Color

Perspective

Visibility

Local Illumination

Smooth Shading

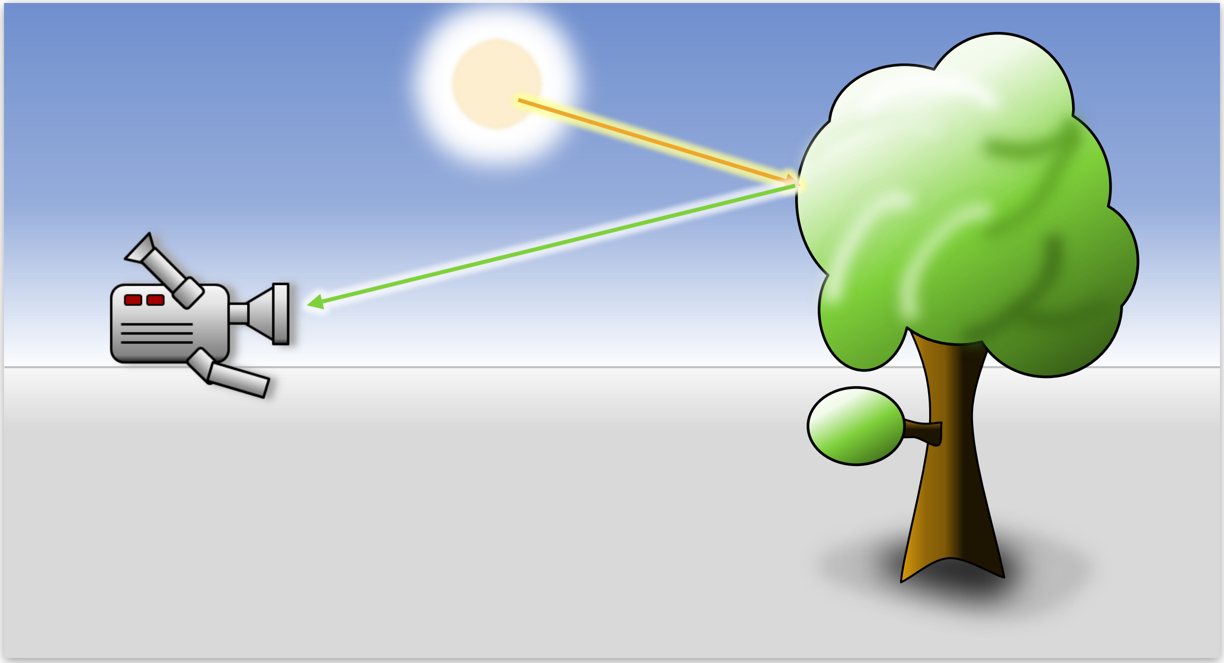
Simple Shadows

Global Illumination

Physics, Biology

Ray Optics & Color

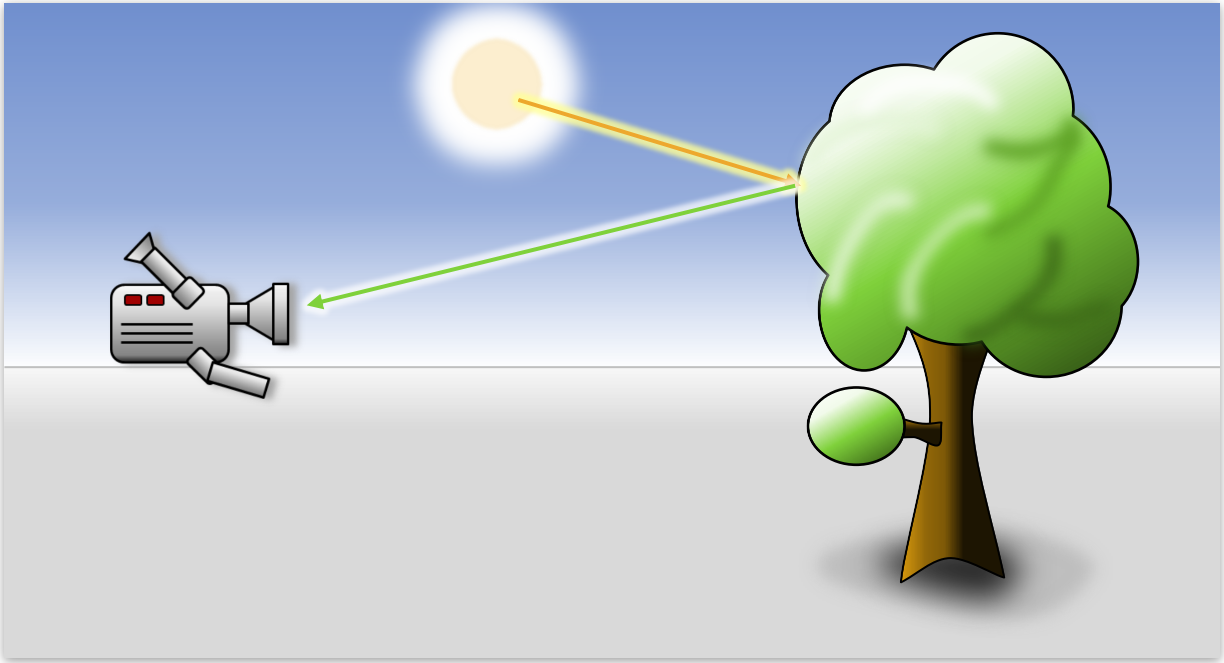
Ray Optics



Geometric ray model

- Light travels along rays

Ray Optics



Geometric ray model

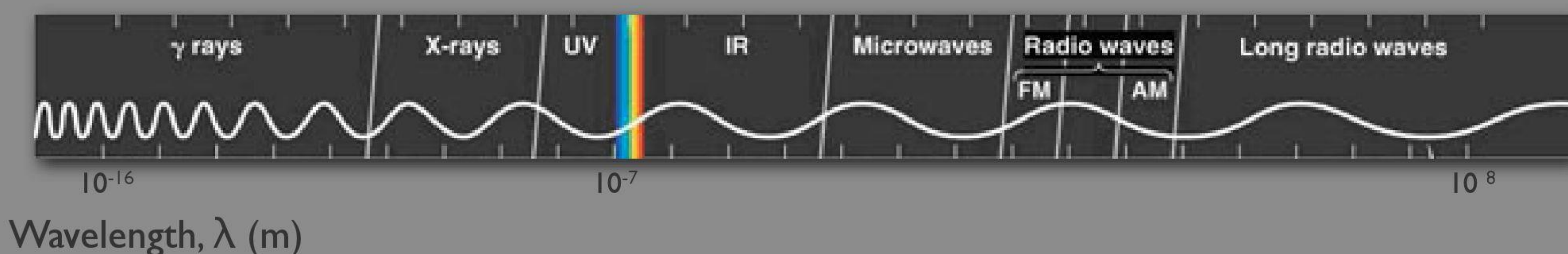
- Rays have “intensity” and “color”

What is COLOR?

- Next slides mostly from Kristi Potter (U Utah)

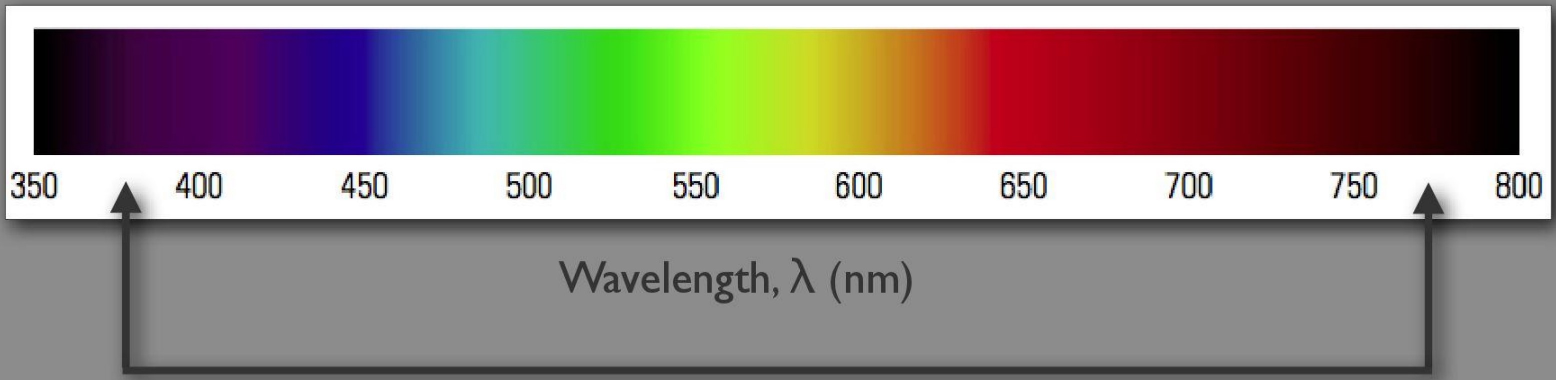
The Electromagnetic Spectrum

Range of all possible frequencies of electromagnetic radiation



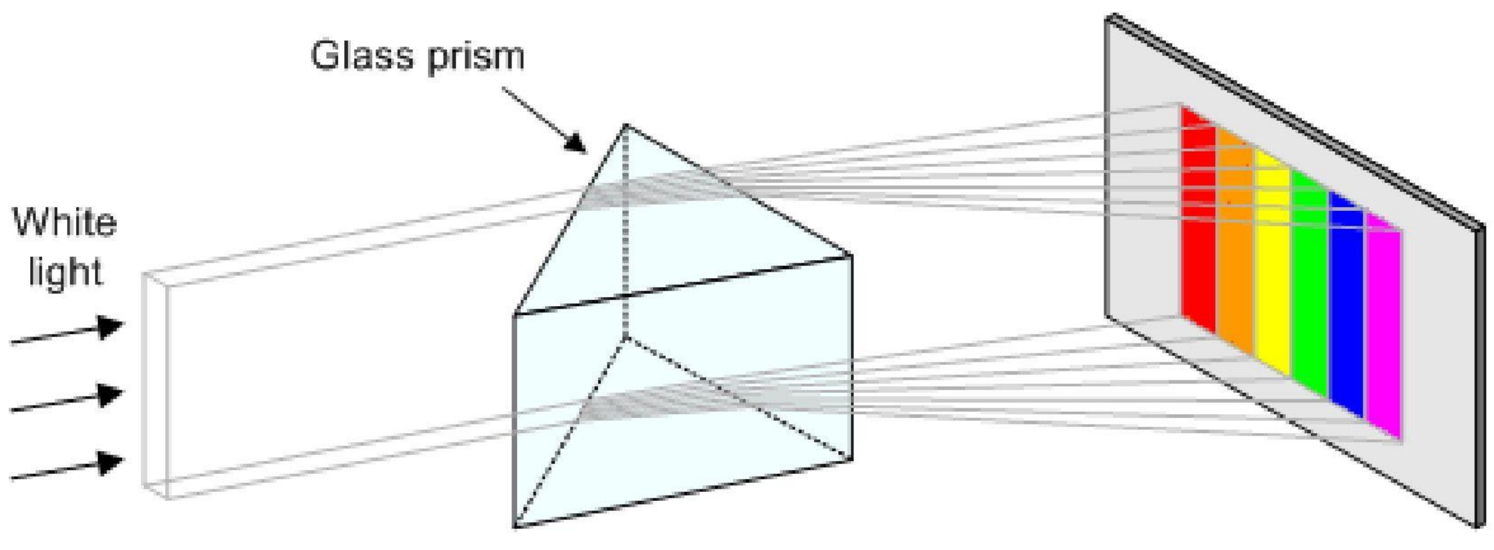
The Visible Spectrum

Human Visual System Sensitive to 380-780 nm



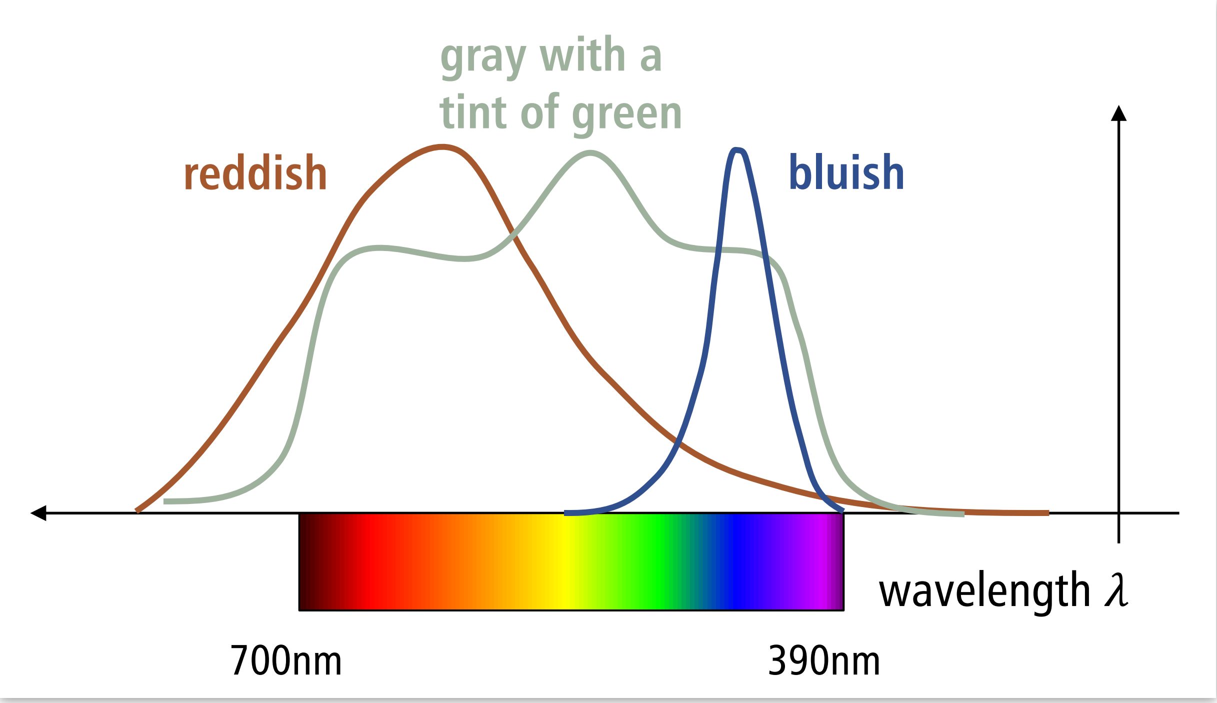
Isaac Newton

Objects appear colored by the character of the light they reflect



Newton's experiment for splitting white light into a spectrum

Ray Optics



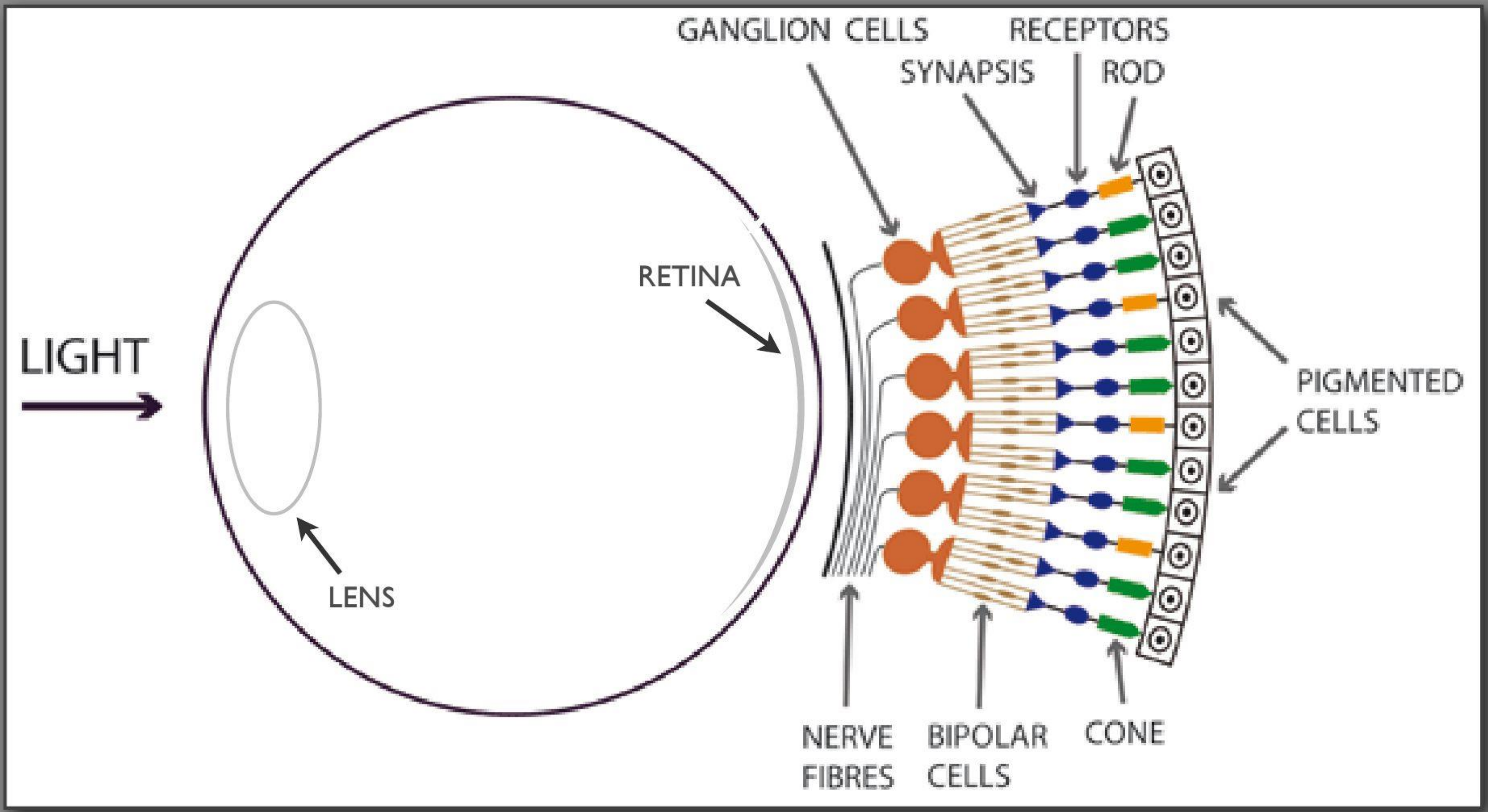
Color spectrum

- Continuous spectrum
- Intensity for each wavelength

Human Color Perception

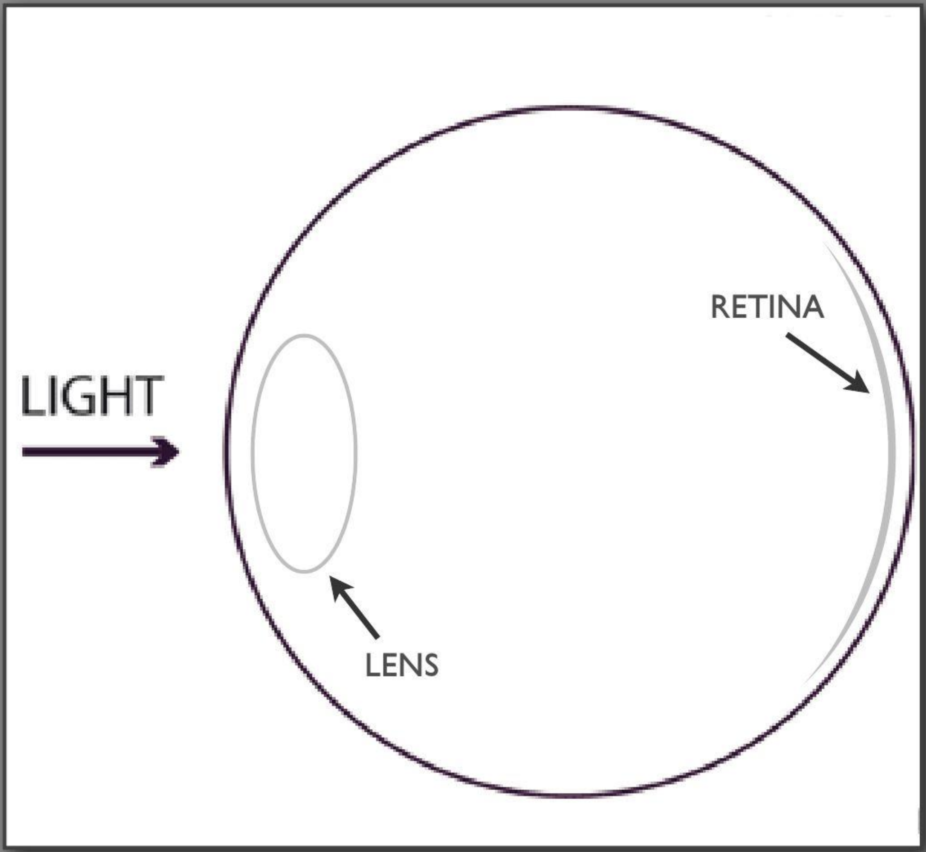
The Eye

Not like a camera



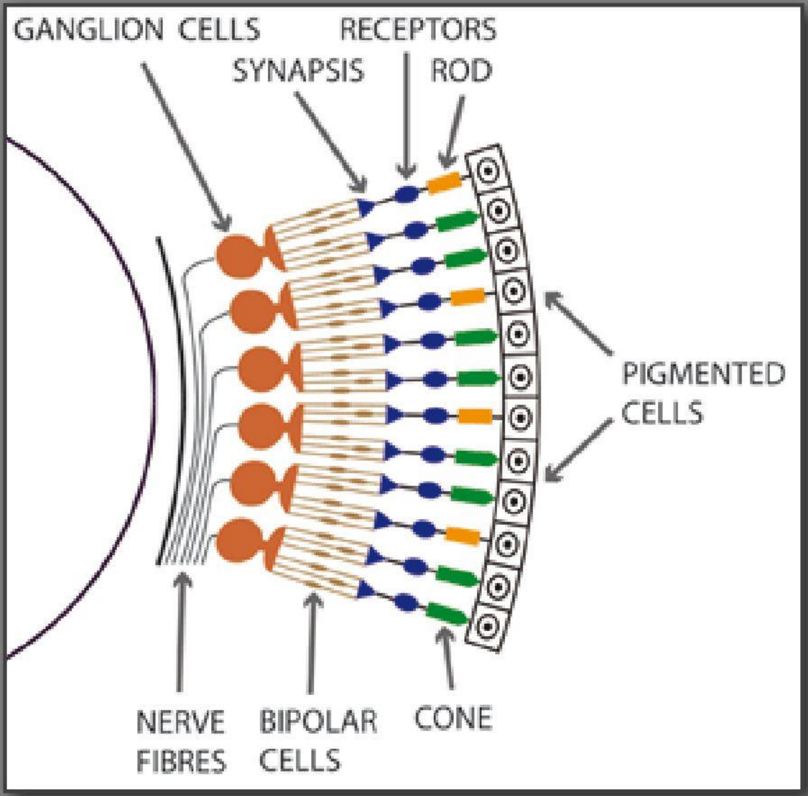
Passage of Light

- Light → Lens
- Lens → Retina



Retina

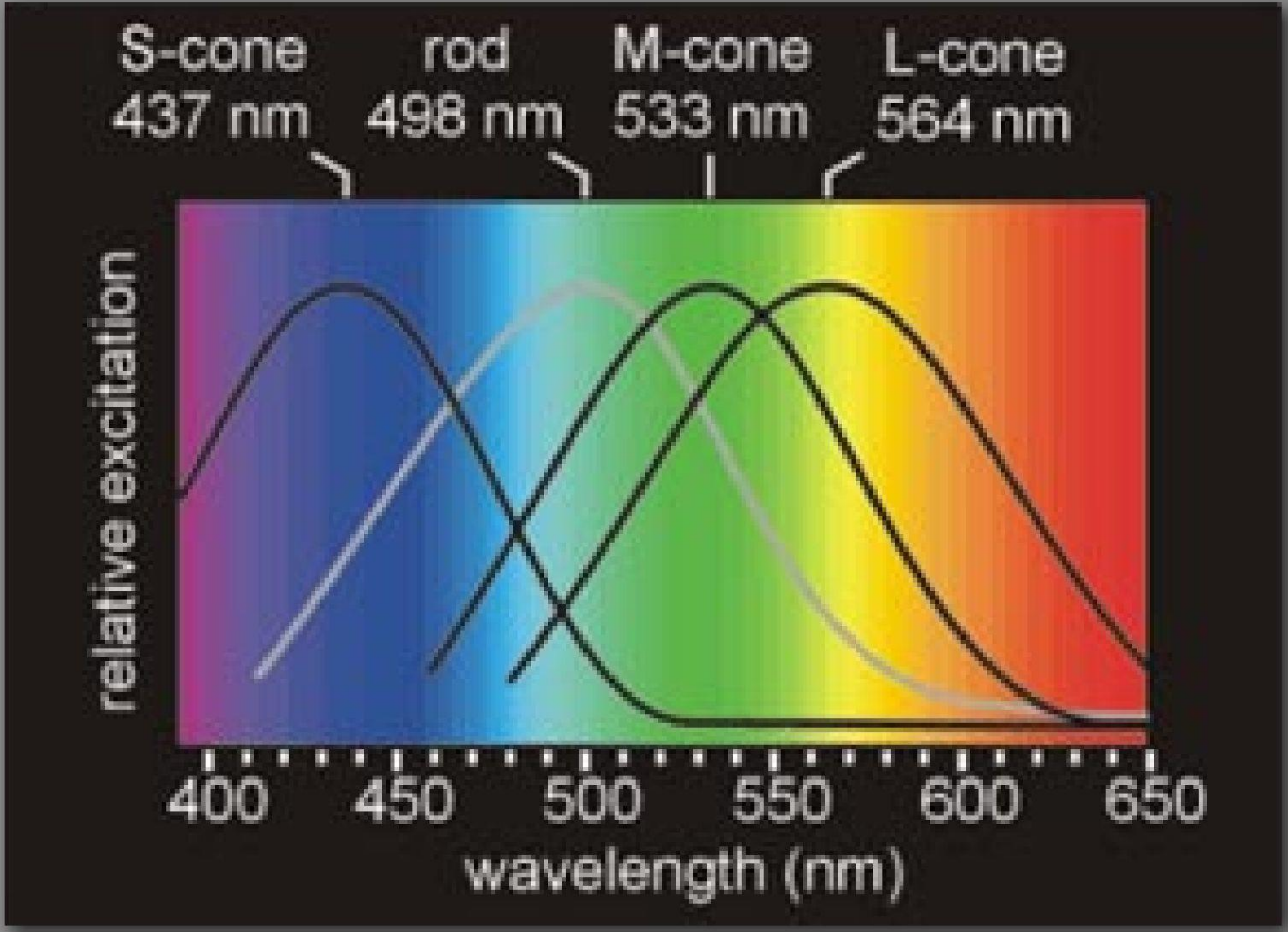
- Optimized for acuity (rather than light sensitivity)
- Initiate information extraction
- Pigmented cells absorb light



Photoreceptors

- Cones for day vision (small, medium, long)
- Rods for night vision
- Binary signal on/off
- Info indicated by which cell & how often

Color Coded through Signal Comparison



Why 3?

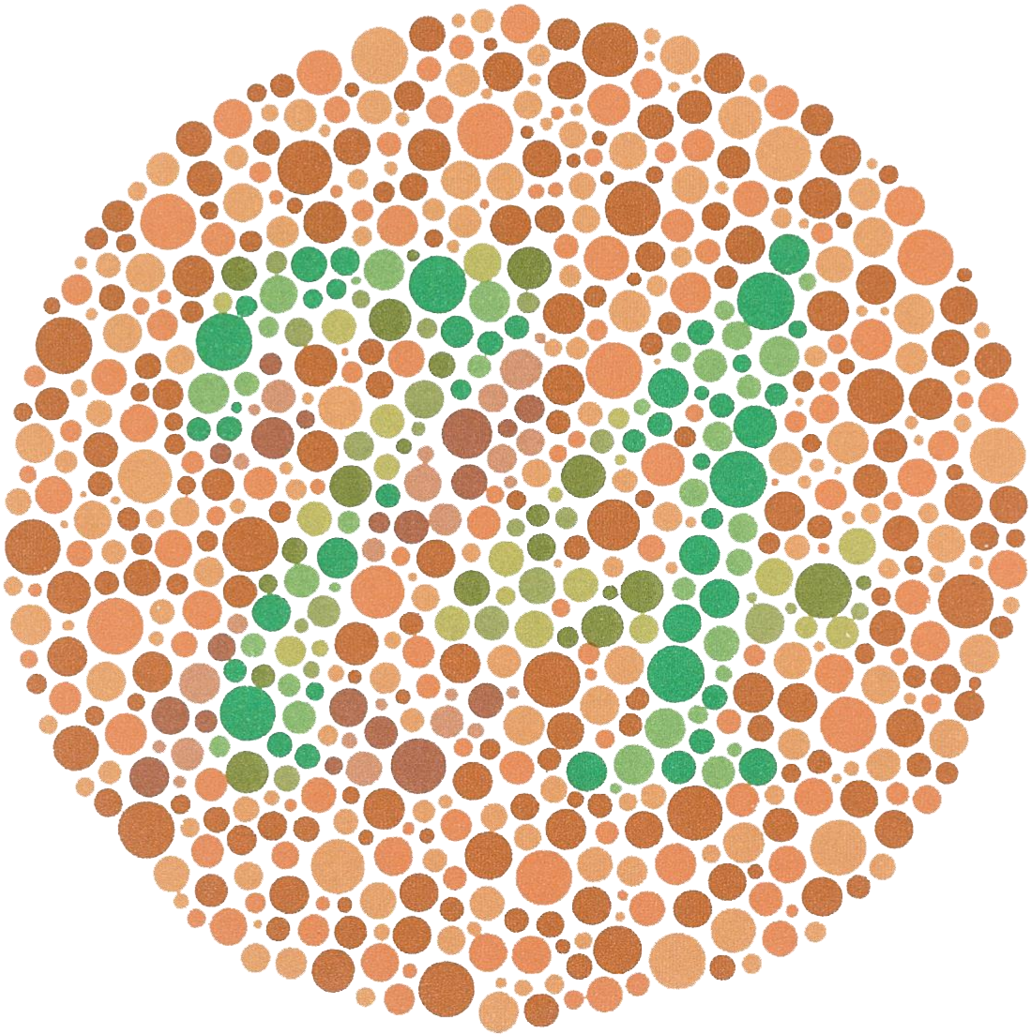
- Our 3 cones cover the visible spectrum
- Theoretically possible with only 2 cones
- Most birds, some fish, reptiles & insects have 4

- Color is:
  - A spectral distribution of light
  - Perceptual response to spectral distribution of light
  - A way of encoding a spectral distribution of light

Computer Graphics → Visualization ← Vision Science / Neuro Science

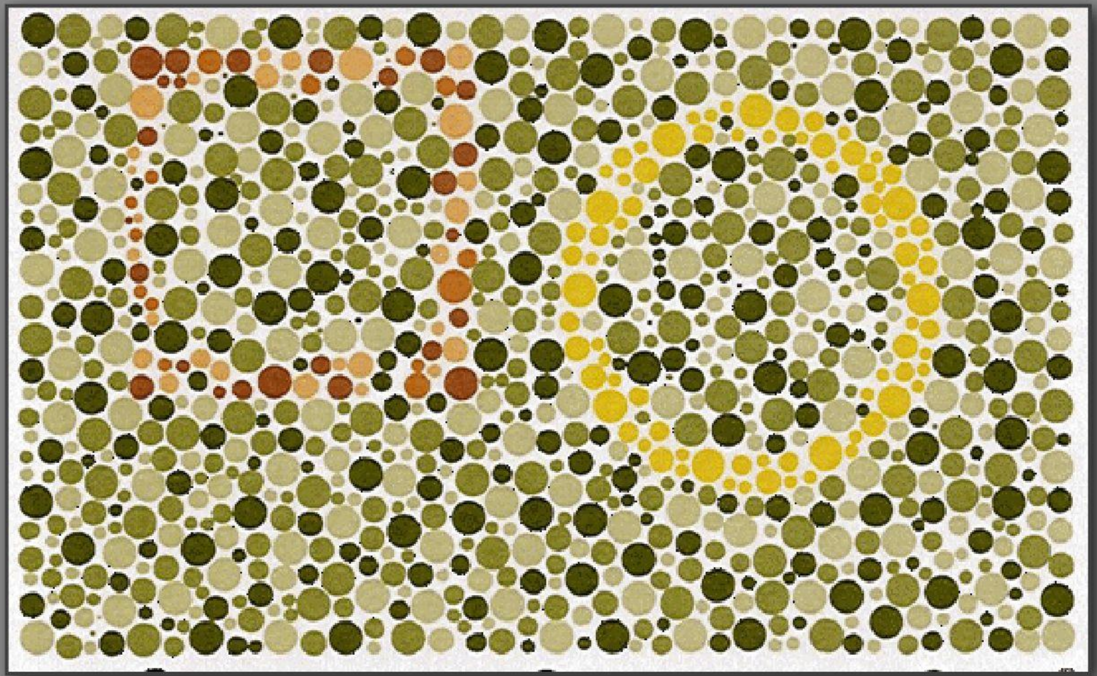
- It would be too simplistic to describe color just as
  - A particular wavelength of light
  - RGB

Color Blindness



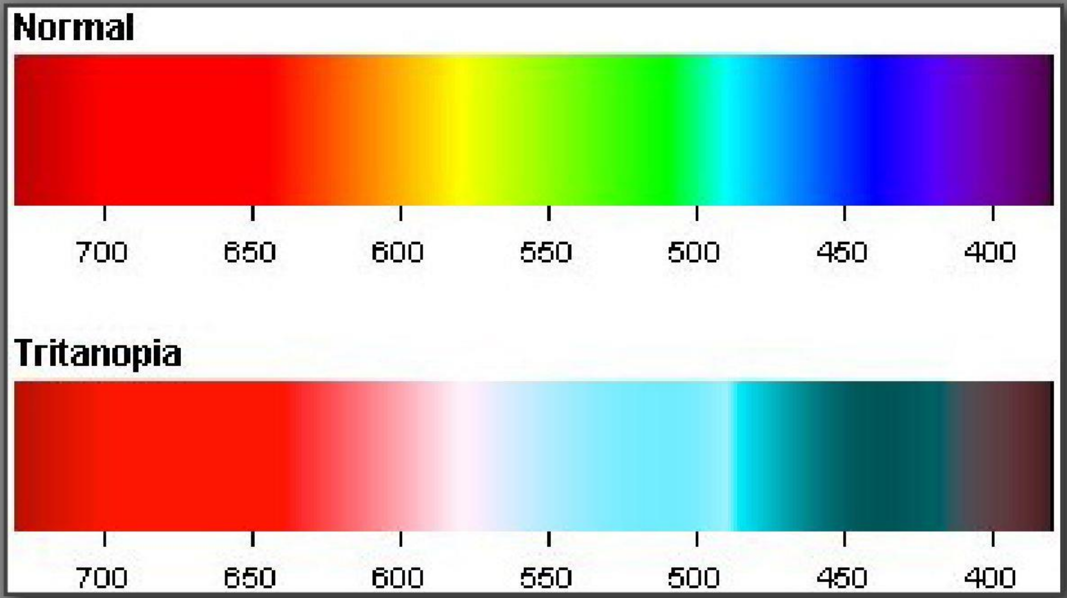
The numeral "74" should be clearly visible to viewers with normal color vision.
Viewers with dichromacy or anomalous trichromacy may read it as "21“.
Viewers with achromatopsia may not see numbers.
From http://en.wikipedia.org/wiki/Color_blindness

Red/Green

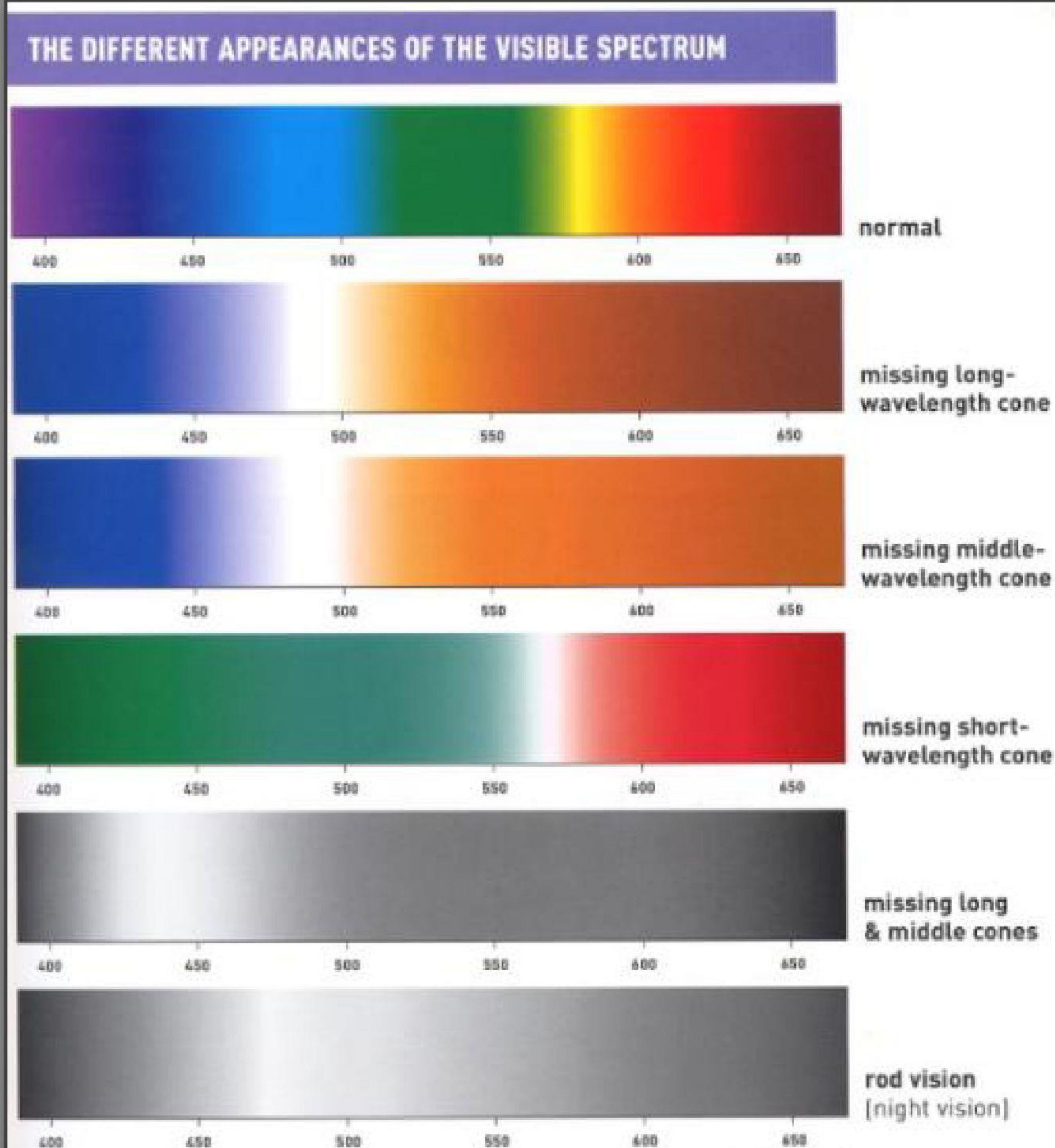


- Lack of or mutations in red or green cones
- Genes located on the X chromosome (women have 2, men have 1)
- 10% of men, less than 1% of women

Blue/Yellow



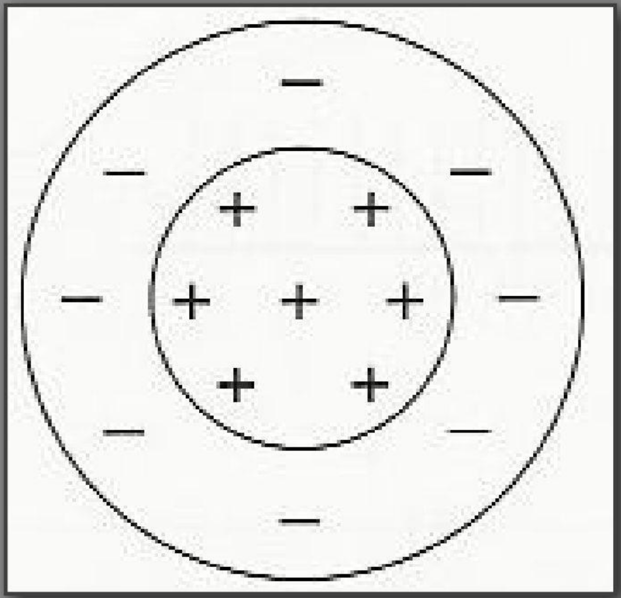
- Equally found in men & women
- mutation in short wave cone


THE DIFFERENT APPEARANCES OF THE VISIBLE SPECTRUM
normal
missing long-wavelength cone
missing middle-wavelength cone
missing short-wavelength cone
missing long & middle cones
rod vision [night vision]
400
450
500
550
600
650

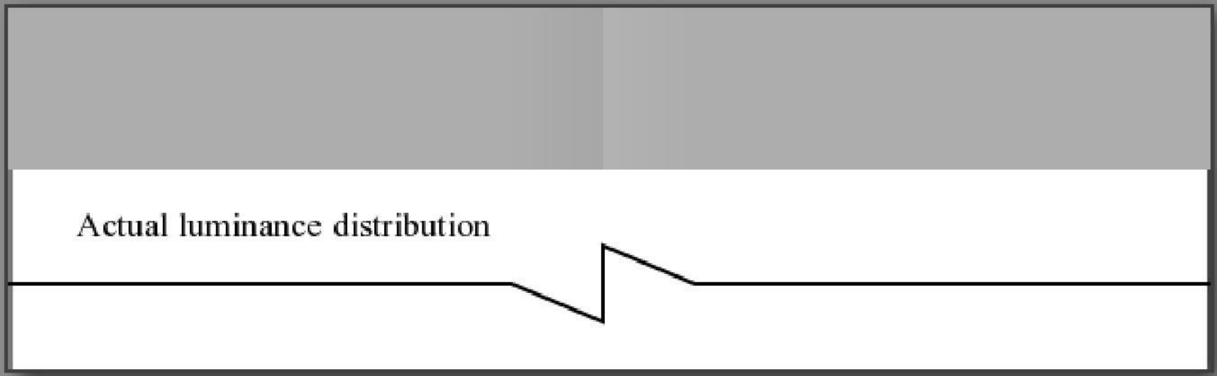
Color Illusions

Center/Surround

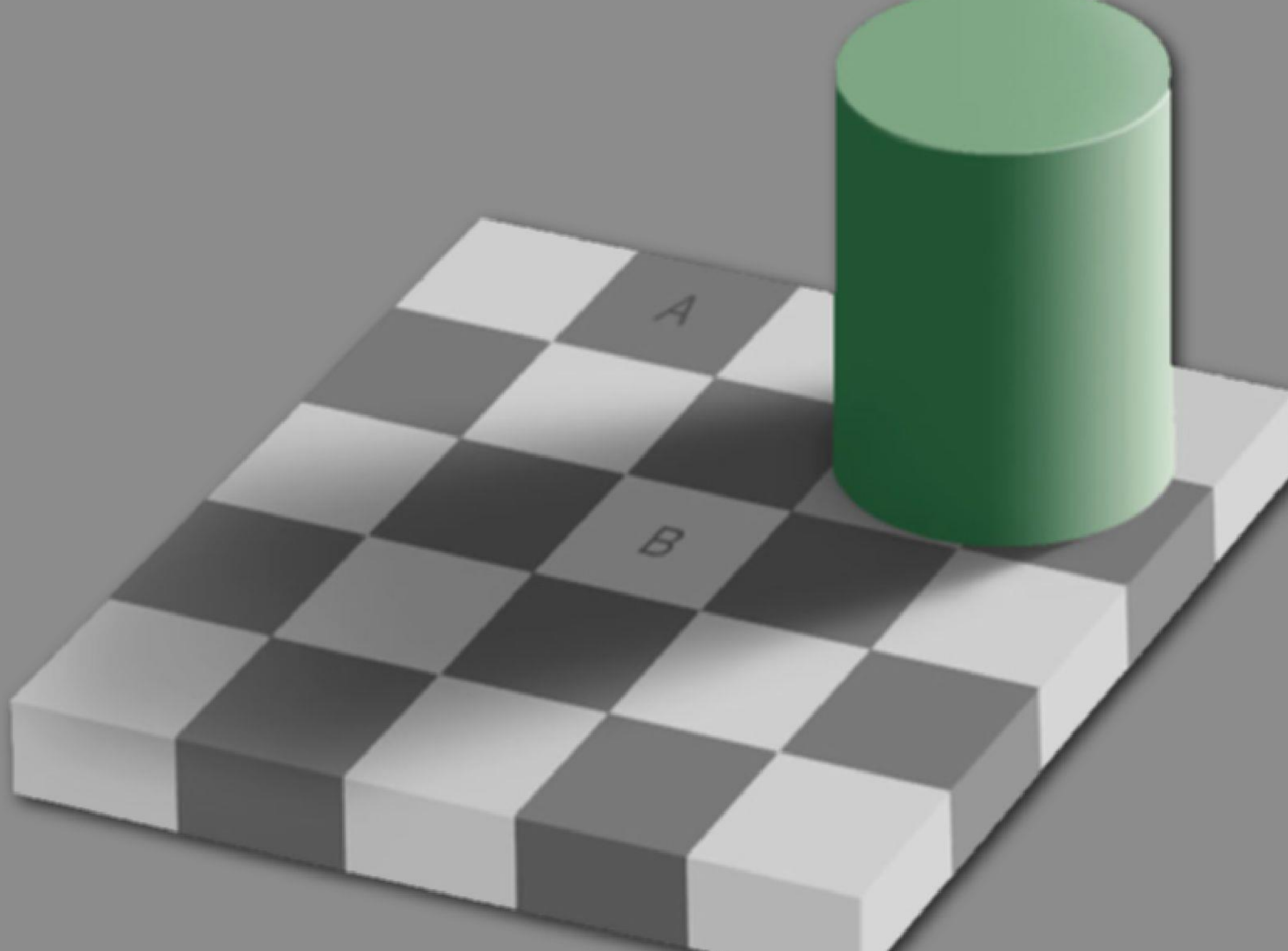
- Retinal ganglion cells
- First stage of visual processing
- Triggered by light in the center suppressed by light in the surround
- Selectively sensitive to discontinuities in light

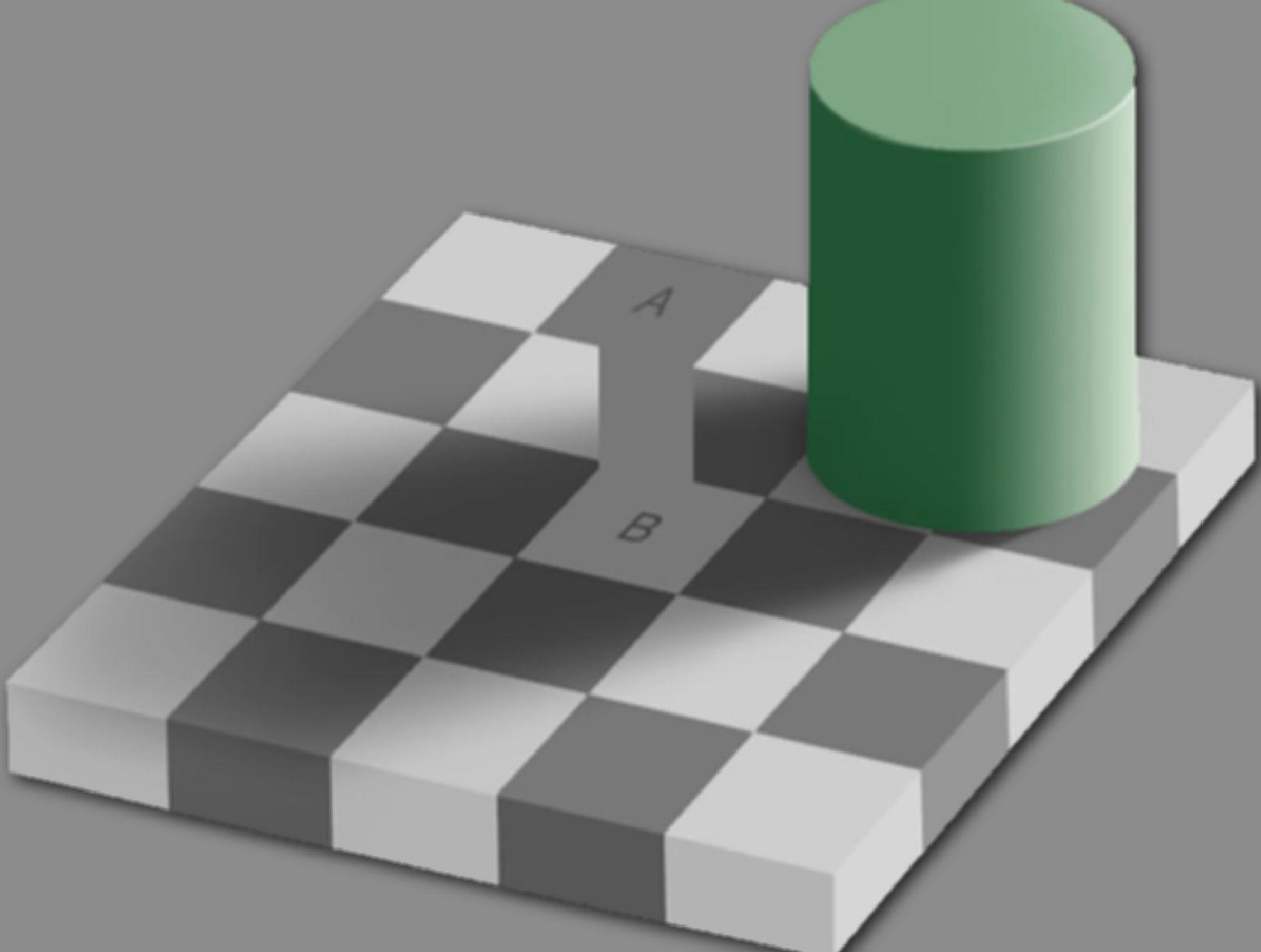


Cornsweet Illusion



- Luminance the same at the ends
- Many perceptions more sensitive to abrupt change (luminance, color, motion, depth)
- Attributed to center/surround organization


A
B

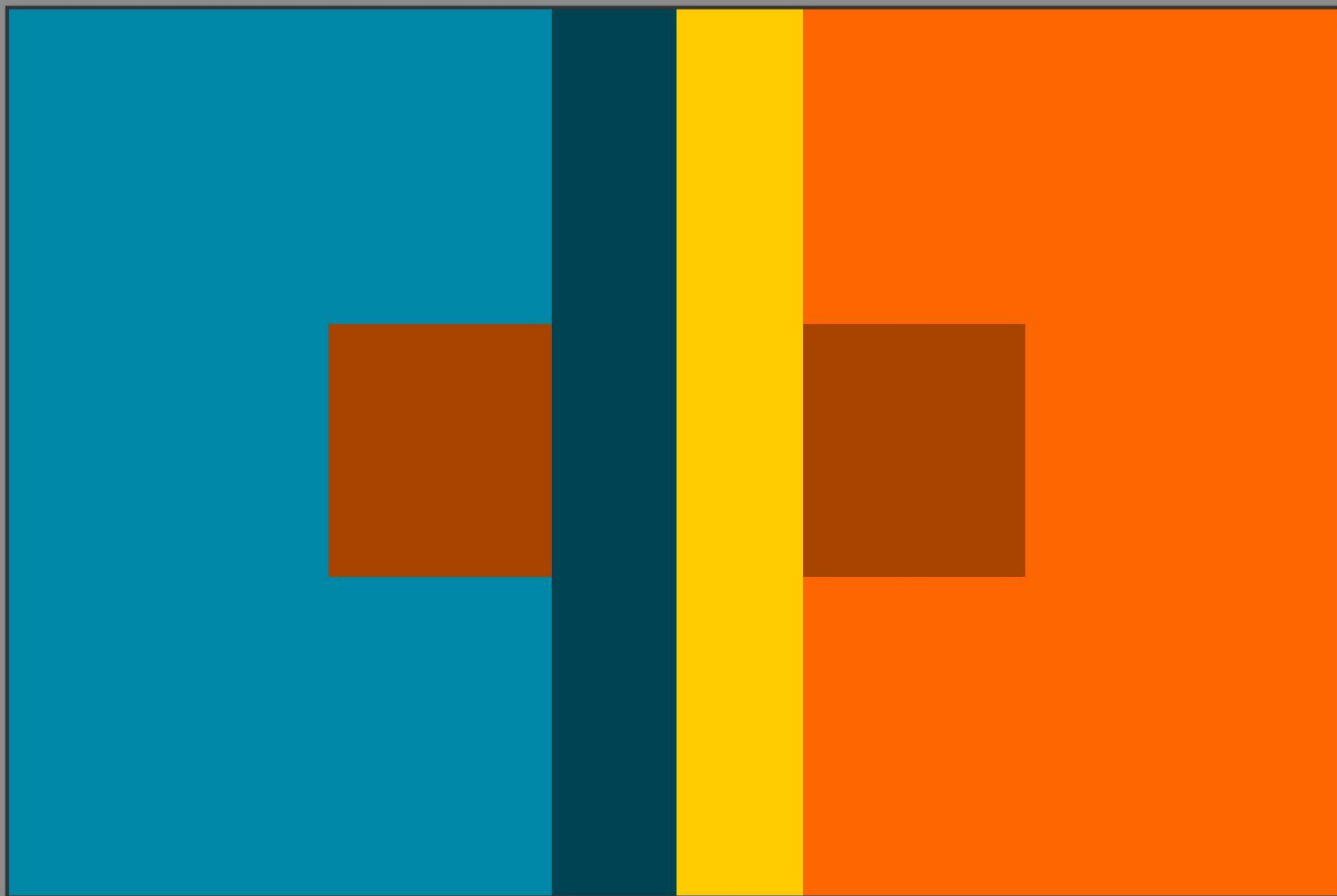

A
B

Contrast Effects

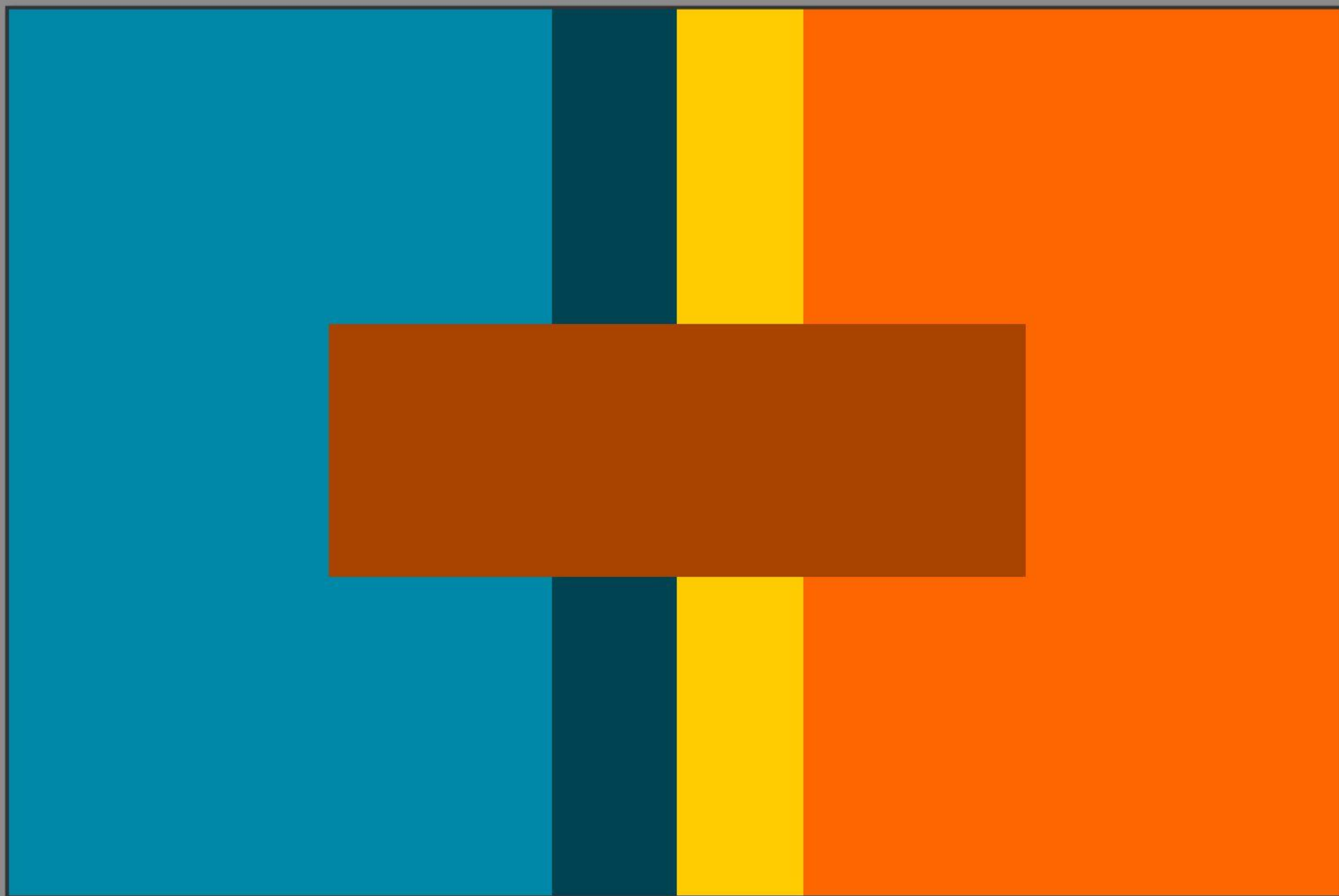
- Result of center/surround cells
- Simultaneous or successive
- Juxtaposition of colors effects our perception of them
- Complimentary colors often most effected

- The terms "simultaneous contrast" and "successive contrast" refer to visual effects in which the appearance of a patch of light (the "test field") is affected by other light patches ("inducing fields") that are nearby in space and time, respectively.
- The names are somewhat misleading since both simultaneous and successive contrast involve inducing fields that are close in both time and space.

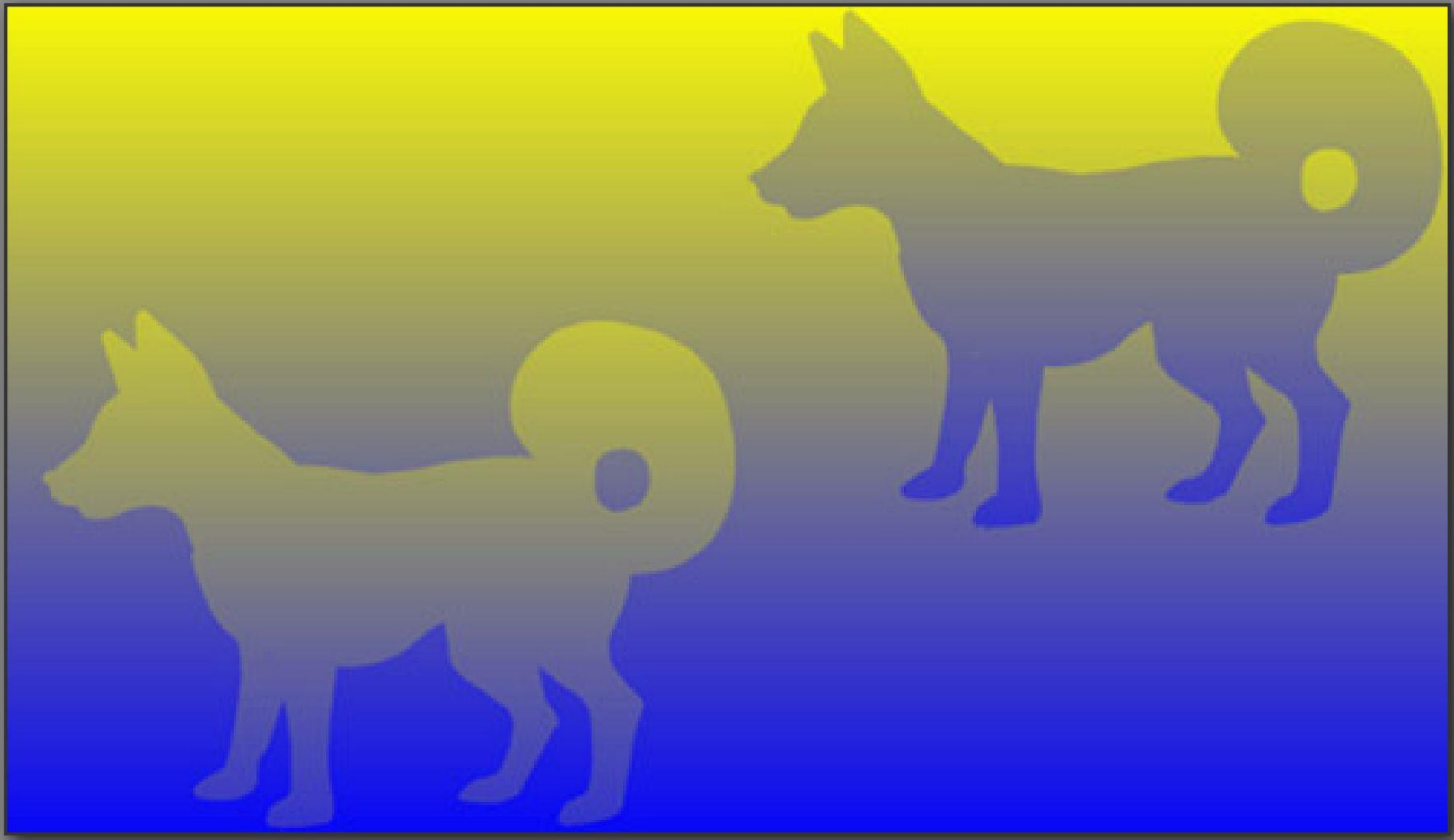
Simultaneous Contrast



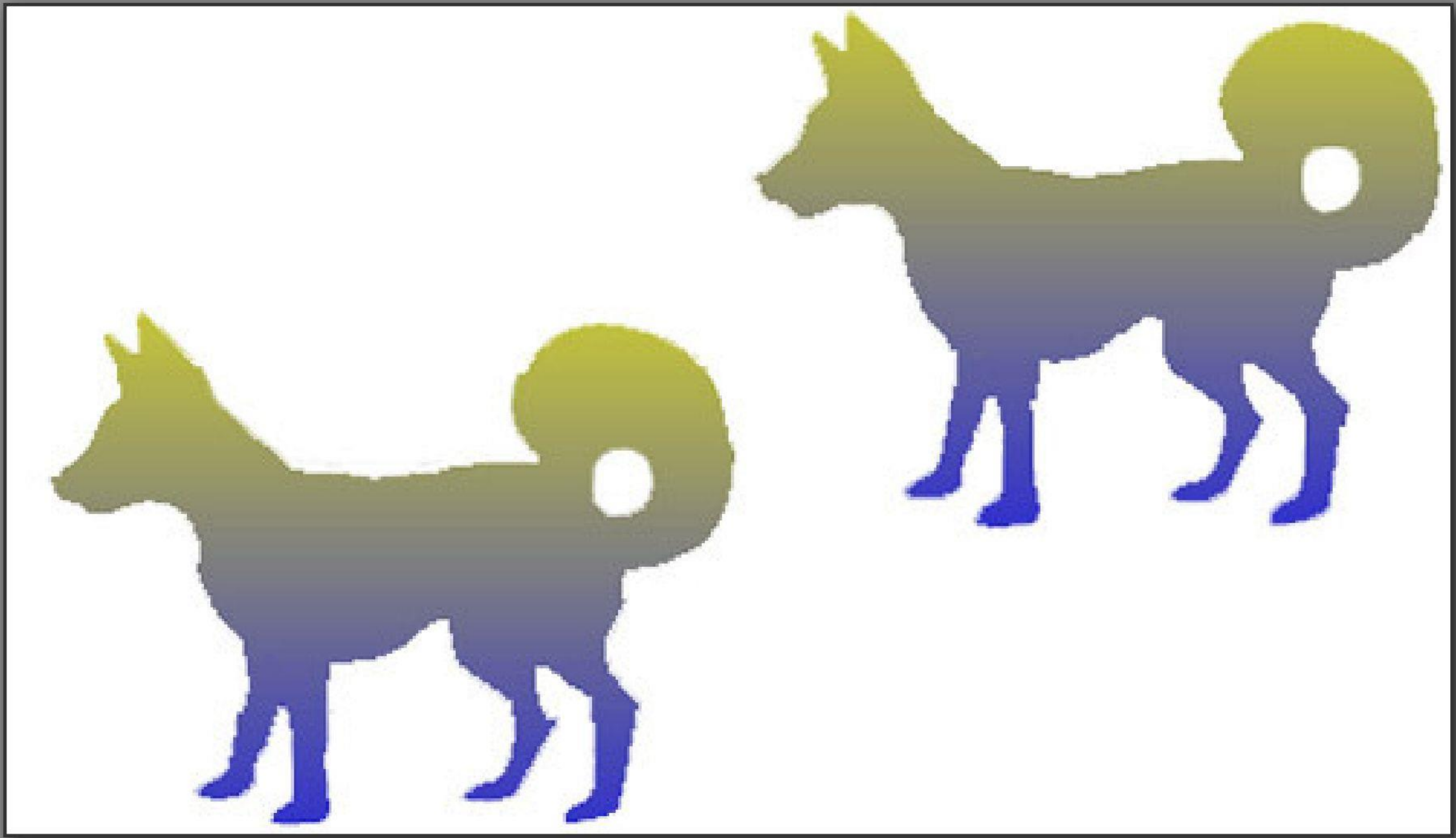
Simultaneous Contrast



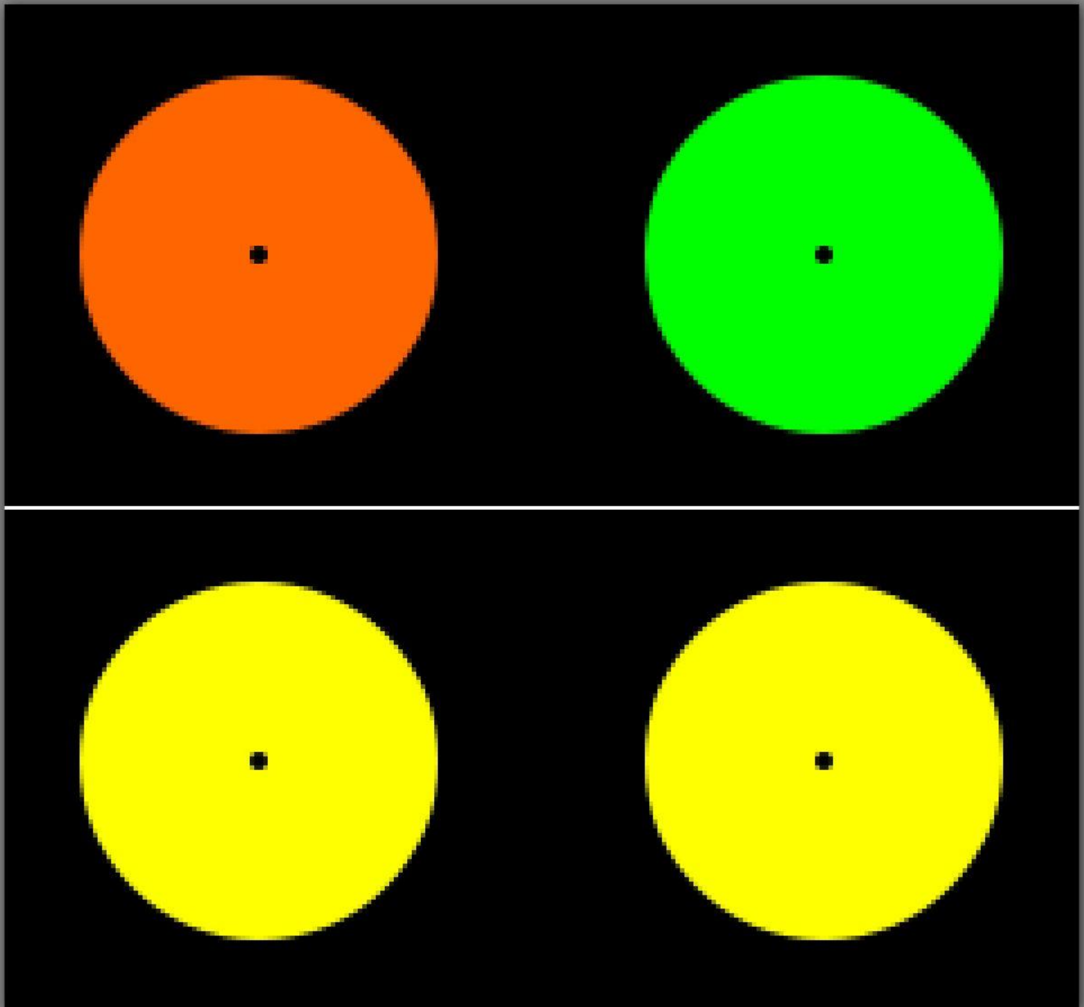
Simultaneous Contrast



Simultaneous Contrast



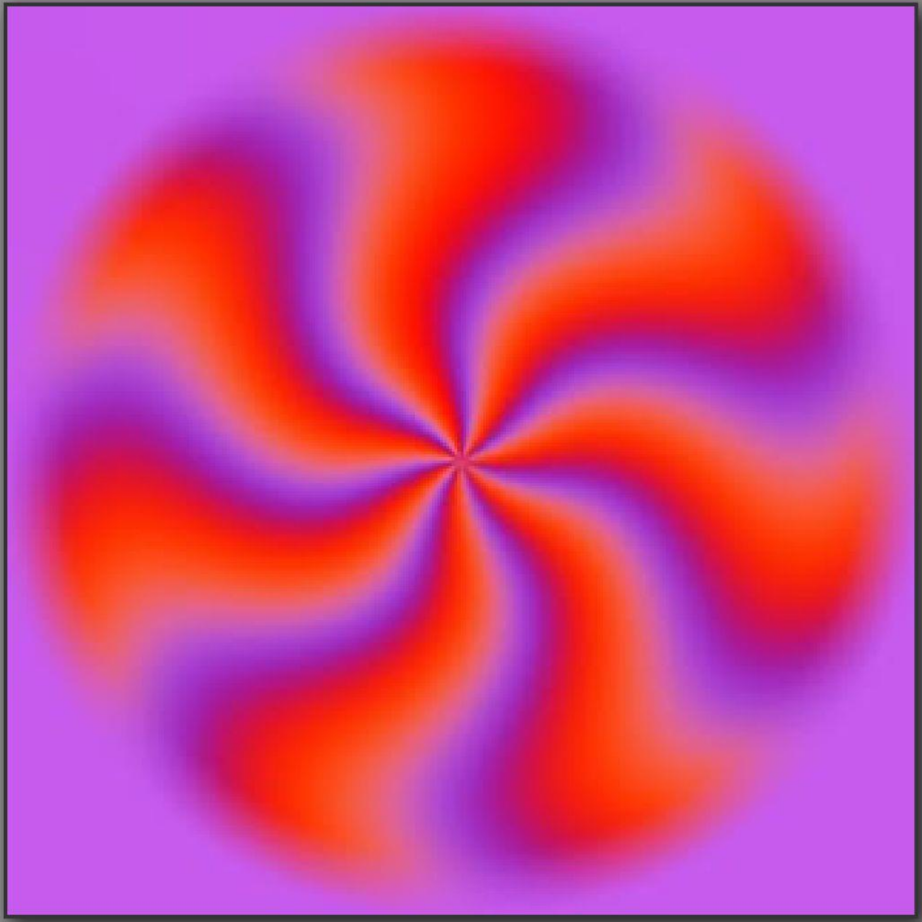
Successive Contrast



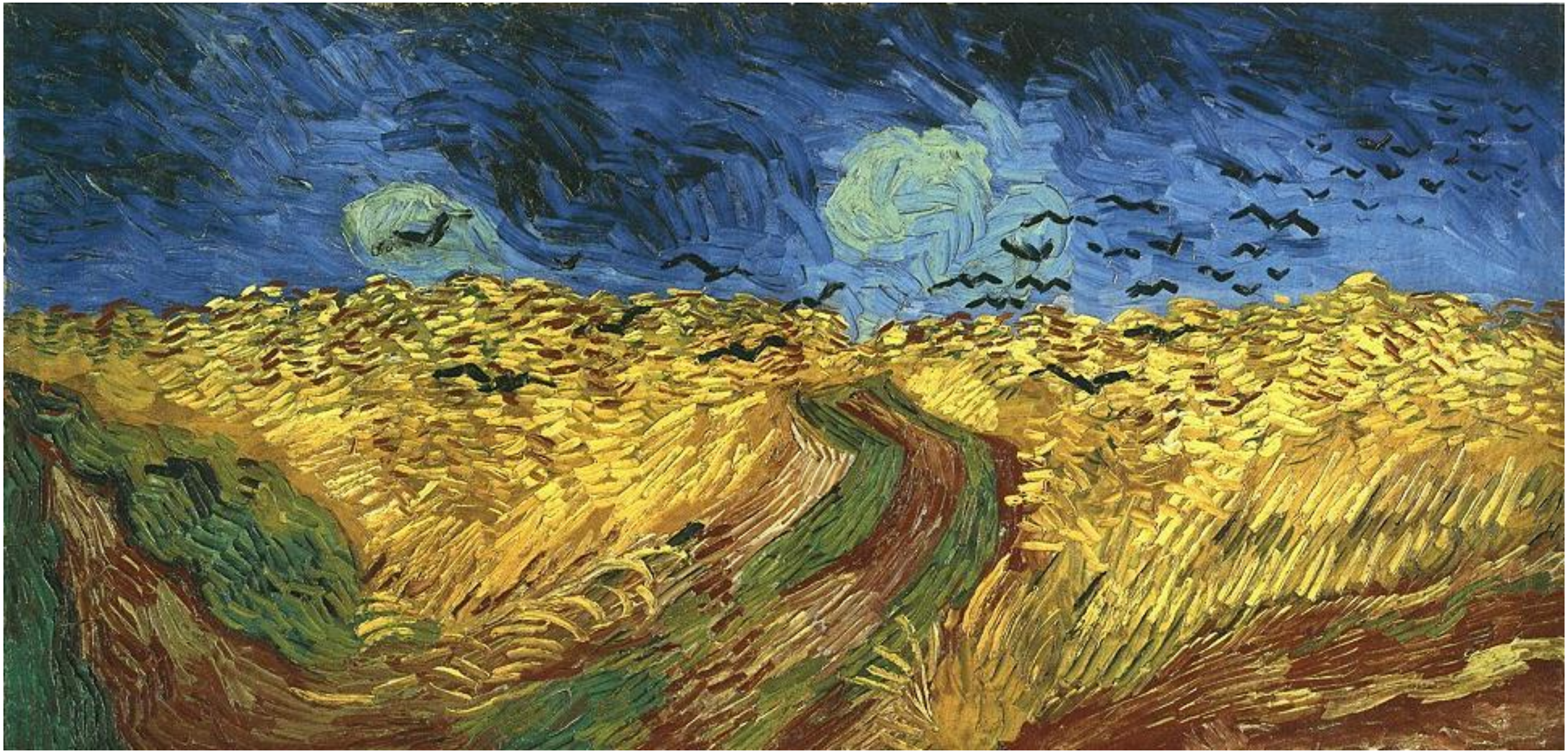
Successive Contrast

Equiluminant Colors

- Strong contrast causes shapes to be seen by color sensitive cell
- Equiluminance hides positions from light sensitive cells
- Flickering/movement caused by this disconnect

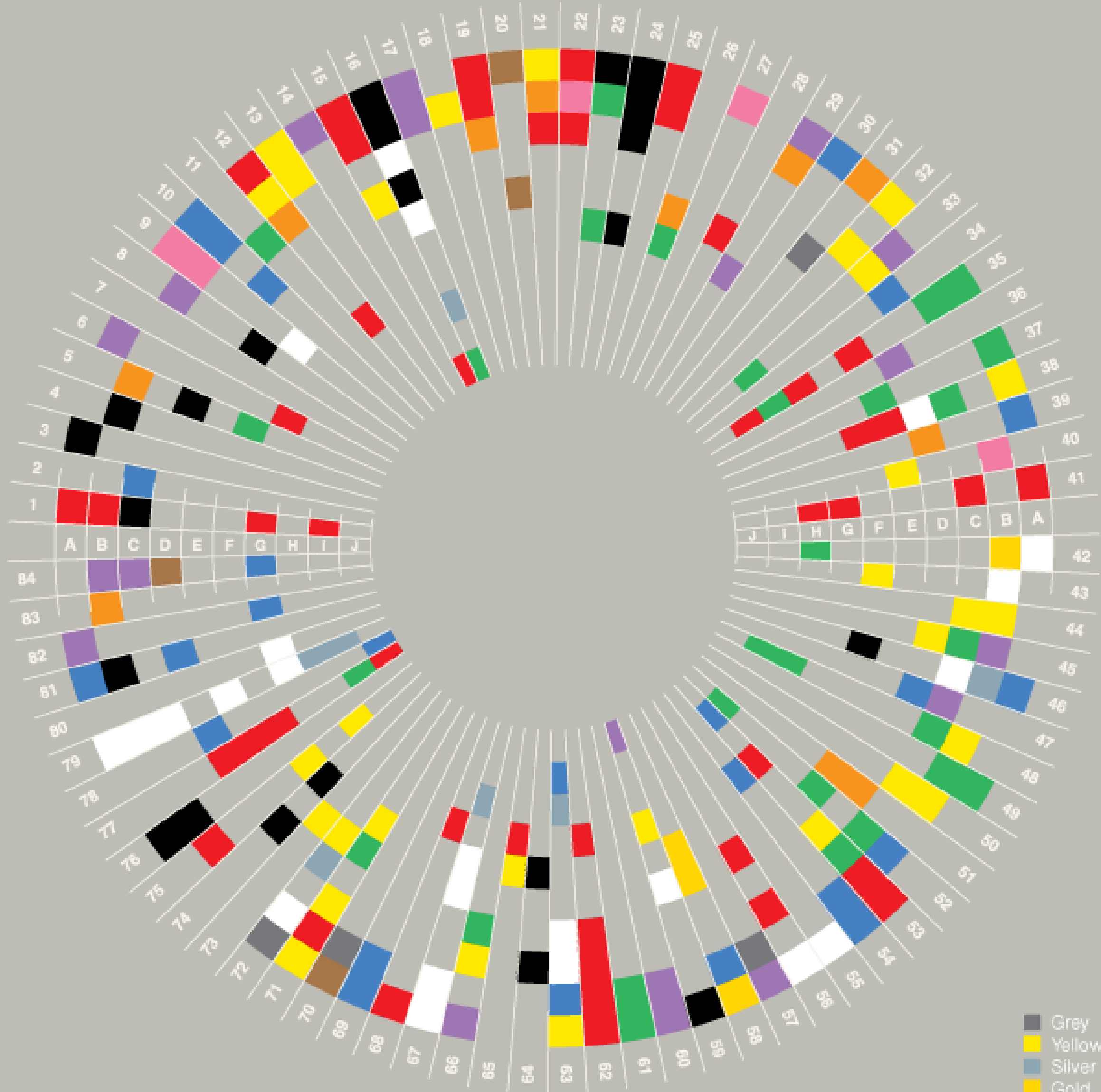


Other perceptual aspects



„This is Van Gogh's last painting before he committed suicide."

Colours In Culture



A	Western / American	F	Asian
B	Japanese	G	Eastern European
C	Hindu	H	Muslim
D	Native American	I	African
E	Chinese	J	South American

1 Anger
2 Art / Creativity
3 Authority
4 Bad Luck
5 Balance
6 Beauty
7 Calm
8 Celebration
9 Children
10 Cold
11 Compassion
12 Courage
13 Cowardice
14 Cruelty
15 Danger
16 Death
17 Decadence
18 Deceit
19 Desire
20 Earthy
21 Energy
22 Erotic
23 Eternity
24 Evil
25 Excitement
26 Family
27 Femininity
28 Fertility
29 Flamboyance
30 Freedom
31 Friendly
32 Fun
33 God
34 Gods
35 Good Luck
36 Gratitude
37 Growth
38 Happiness
39 Healing
40 Healthy
41 Heat
42 Heaven
43 Holiness
44 Illness
45 Insight
46 Intelligence
47 Intuition
48 Religion
49 Jealousy
50 Joy
51 Learning
52 Life
53 Love
54 Loyalty
55 Luxury
56 Marriage
57 Modesty
58 Money
59 Mourning
60 Mystery
61 Nature
62 Passion
63 Peace
64 Penance
65 Power
66 Personal power
67 Purity
68 Radicalism
69 Rational
70 Reliable
71 Repels Evil
72 Respect
73 Royalty
74 Self-cultivation
75 Strength
76 Style
77 Success
78 Trouble
79 Truce
80 Trust
81 Unhappiness
82 Virtue
83 Warmth
84 Wisdom

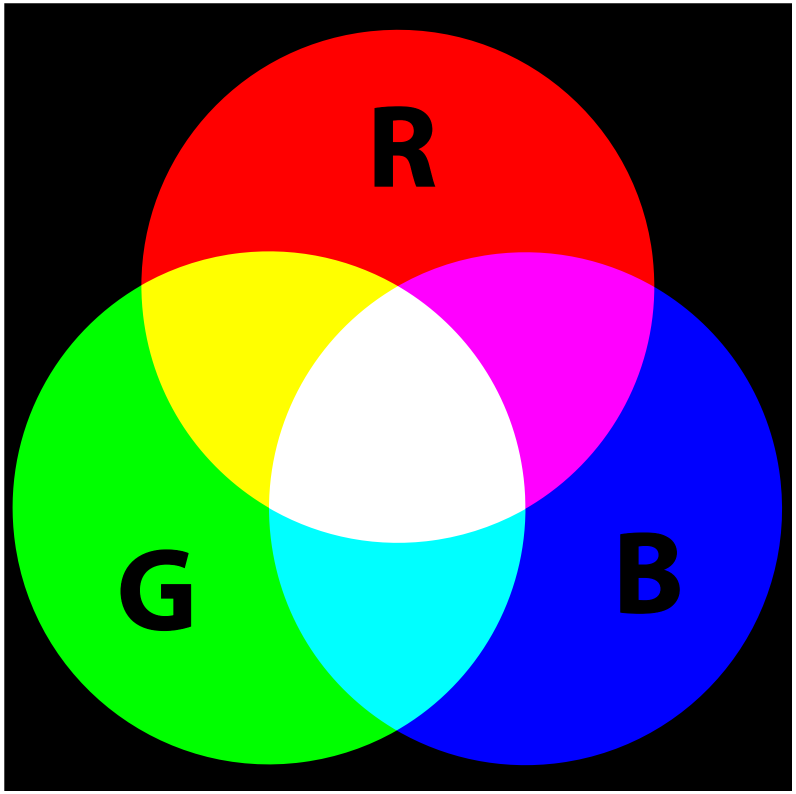
David McCandless & AlwaysWithHonor.com // v1.0 // Apr 09 // InformationIsBeautiful.net

source: Pantone, ColorMatters & web sources

Color Mixing, Color Models, Color Interpolation

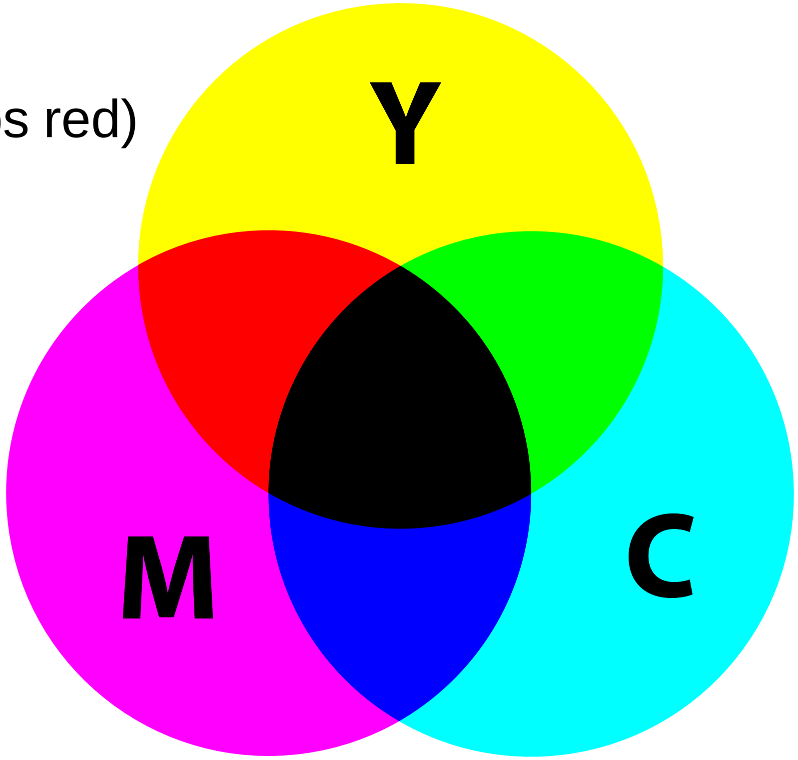
- Additive color mixing:
  - Light rays with different spectra of light come together
  - The spectra add up
  - The result is a different spectrum of light, i.e., color.

- Example RGB:
  - Monitors



- Subtractive color mixing:
  - A light ray with a (white) spectrum of light hits a surface
  - It is being reflected
  - The surface **absorbs** some wavelengths of light
  - The result is a different spectrum of light, i.e., color.

- Example CMY(K):
  - Cyan: complement of red (= absorbs red)
  - Magenta: complement of green
  - Yellow: complement of blue
  - K = black ink to hide color mixing imperfections



- Color Models are a way to encode a spectrum of light
  - HSL
  - HSV
  - RGB
  - CMYK
  - many more…

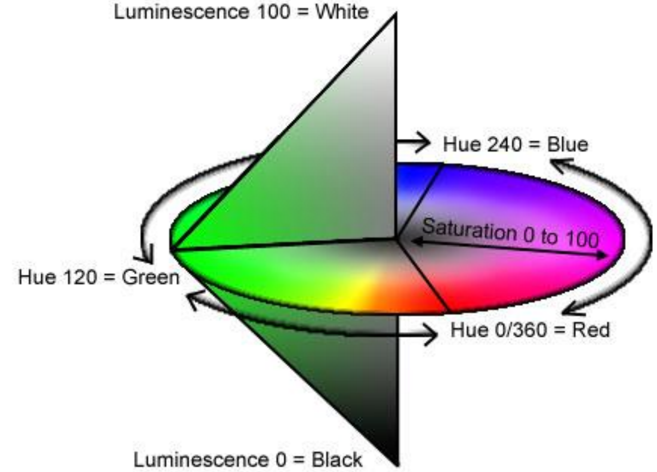
HLS System

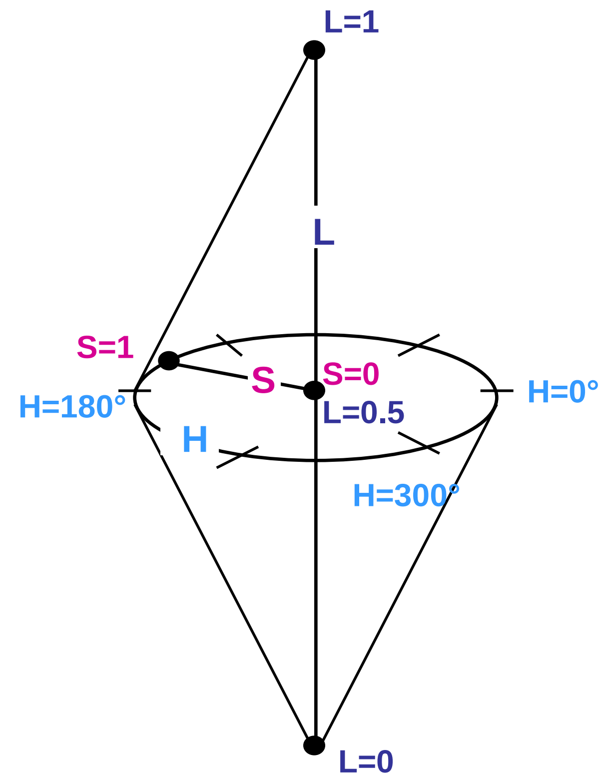
- **H**ue
  classifies a color as red, green, blue, or mixture of these. The hues are given on a circle.

- **L**ightness
  depends on the amount of light

- **S**aturation
  describes the gray portion of the color

- Perception-oriented color system





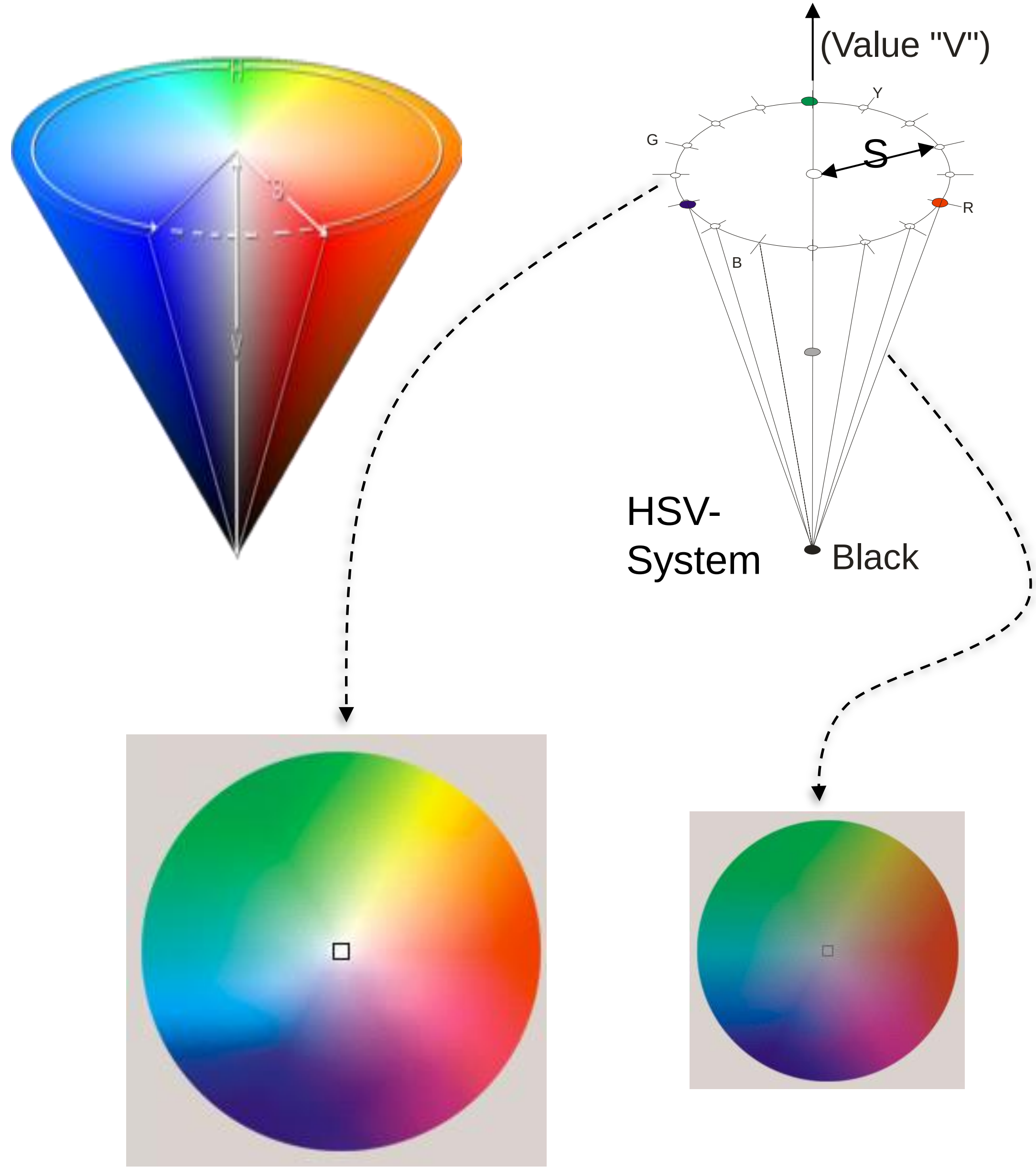
HSV System

- **H**ue
  classifies a color as red, green, blue, or mixture of these. The hues are given on a circle.

- **S**aturation
  describes the gray portion of the color

- **V**alue
  depends on the amount of light

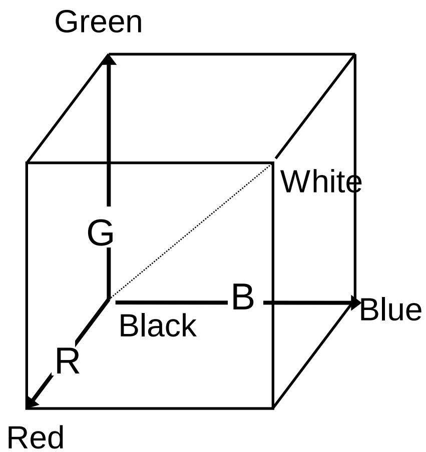
- Perception-oriented color system



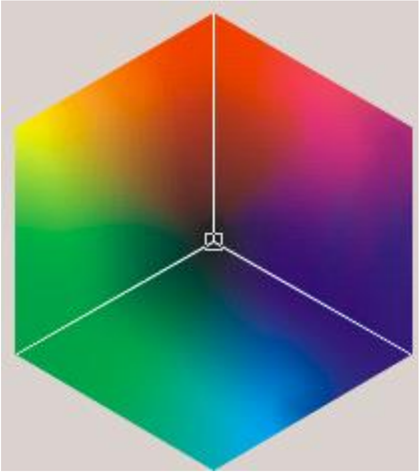
cuts through the HSV cone at v=1 and v=0.5

RGB System

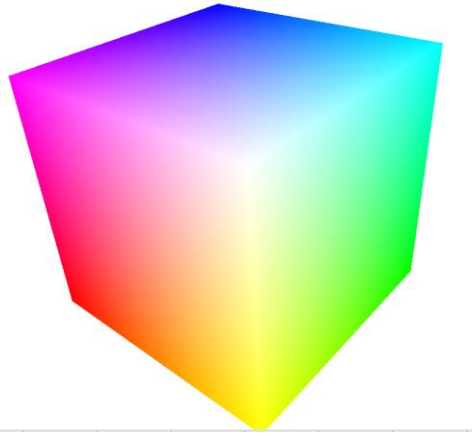
- **Red**
- **Green**
- **Blue**

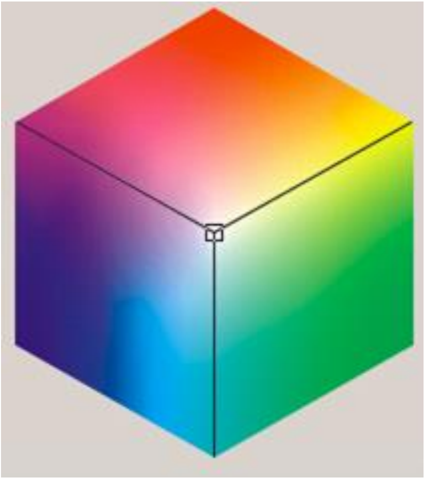


RGB color cube



- Technology-oriented color system
- Describes a color by mixing three primary colors

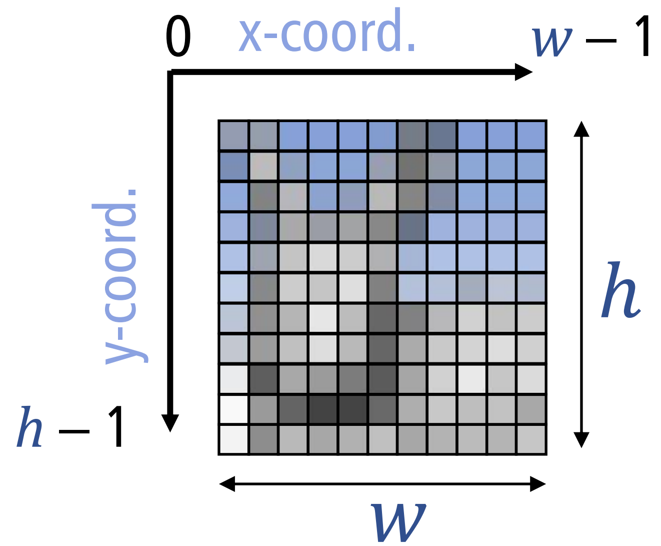




RGB Model

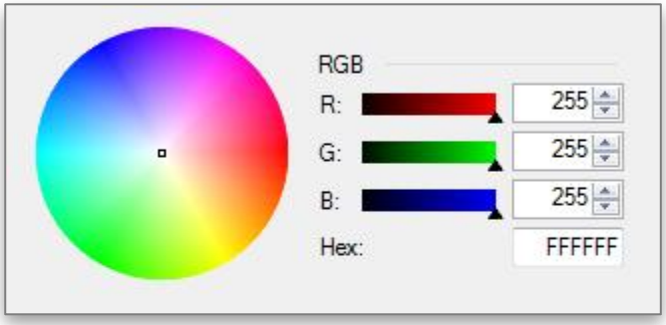
Bitmap (Pixel Display)

- Screen: $w \cdot h$ discrete pixels
  - Origin: usually upper left
- Varying color per pixel

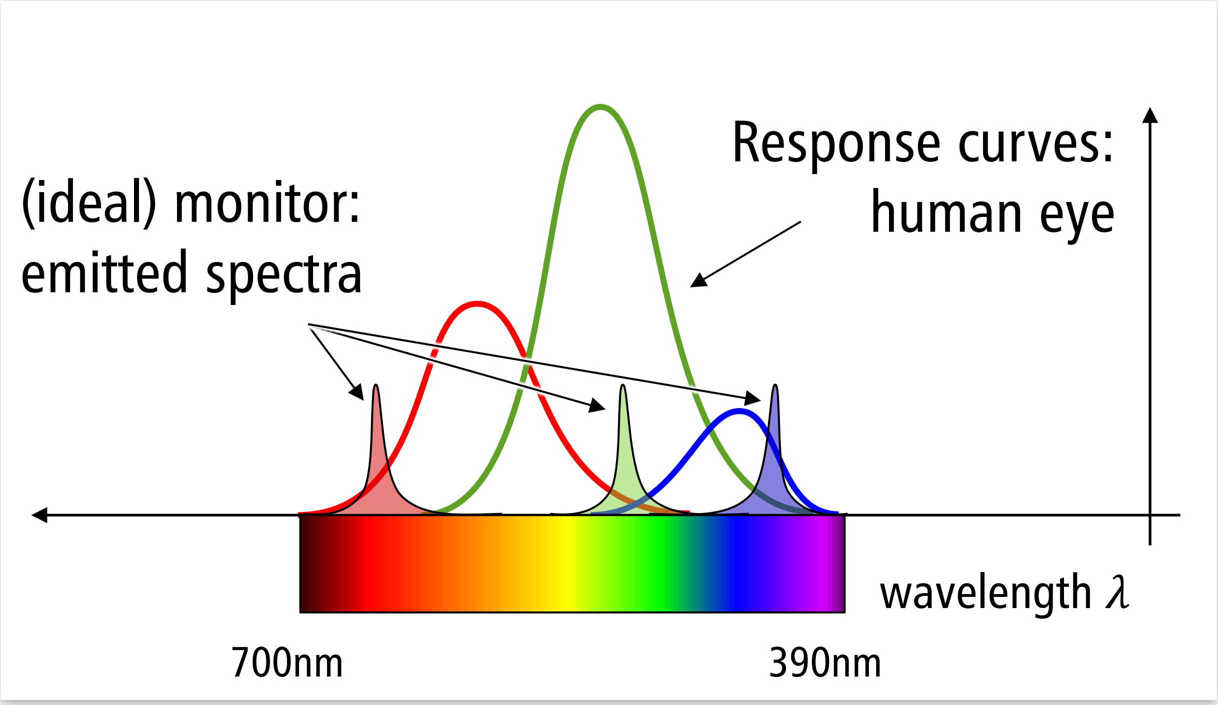


RGB Model

- Every pixel can emit *red*, *green*, *blue* light
- Intensity range:
  - Usually: bytes 0…255
  - $0 = \text{dark}$
  - $255 = \text{maximum brightness}$



Human Vision



(**curves:** schematic, not accurate)

Create color impressions

- Basis for three-dimensional color space
- Wide spacing, narrow bands: purer colors
  - Otherwise: washed out colors

HSL and HSV - Wikipedia

https://en.wikipedia.org/wiki/HSL_and_HSV#Converting_to_RGB

amounts of *R*, *G*, and *B* to reach the proper lightness or value.[19]

From HSV [edit]

Given a color with hue $H \in [0°, 360°]$, saturation $S_{HSV} \in [0, 1]$, and value $V \in [0, 1]$, we first find chroma:

$$C = V \times S_{HSV}$$

Then we can find a point (R_1, G_1, B_1) along the bottom three faces of the RGB cube, with the same hue and chroma as our color (using the intermediate value *X* for the second largest component of this color):

$$H' = \frac{H}{60°}$$

$$X = C(1 - |H' \bmod 2 - 1|)$$

$$(R_1, G_1, B_1) = \begin{cases} (0,0,0) & \text{if } H \text{ is undefined} \\ (C, X, 0) & \text{if } 0 \leq H' < 1 \\ (X, C, 0) & \text{if } 1 \leq H' < 2 \\ (0, C, X) & \text{if } 2 \leq H' < 3 \\ (0, X, C) & \text{if } 3 \leq H' < 4 \\ (X, 0, C) & \text{if } 4 \leq H' < 5 \\ (C, 0, X) & \text{if } 5 \leq H' < 6 \end{cases}$$

Finally, we can find *R*, *G*, and *B* by adding the same amount to each component, to match value:

$$m = V - C$$

$$(R, G, B) = (R_1 + m, G_1 + m, B_1 + m)$$

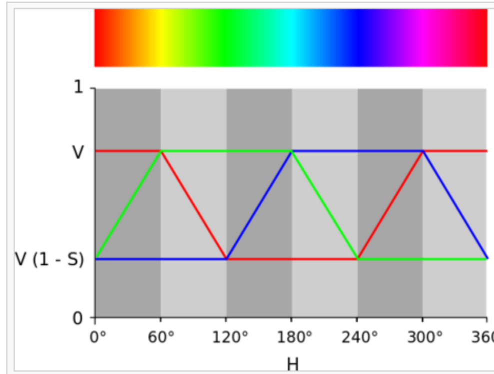


Fig. 24. A graphical representation of RGB coordinates given values for HSV.

From HSL [edit]

- When using a specific color model, we can interpolate between two colors by treating them like vectors and using linear interpolation.
  Example:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = (1-t)\begin{pmatrix} R_1 \\ G_1 \\ B_1 \end{pmatrix} + t\begin{pmatrix} R_2 \\ G_2 \\ B_2 \end{pmatrix}$$

- It is often perceptually better, to interpolate in the HSV or other perception-based models!
  Example of *changing the saturation:*

$$\begin{pmatrix} H \\ S \\ V \end{pmatrix} = \begin{pmatrix} H \\ (1-t)S_1 + tS_2 \\ V \end{pmatrix}$$

Transparency

Transparency

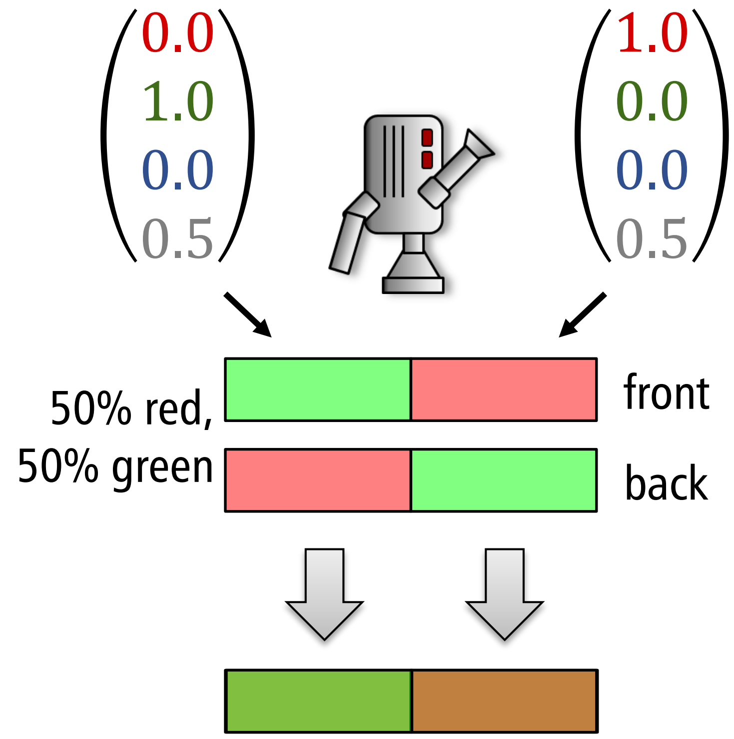
- "Alpha-blending"
- α = "opacity"
- Color + opacity: RGBα

Blending

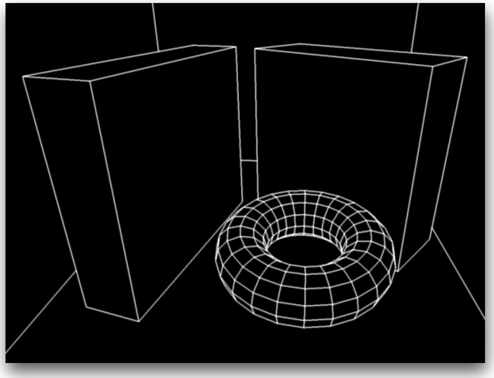
- Mix in α of front color, keep $1-\alpha$ of back color

$$\mathbf{c} = \alpha \cdot \mathbf{c}_{front} + (1-\alpha) \cdot \mathbf{c}_{back}$$

- Not commutative! (order matters)
  - unless monochrome

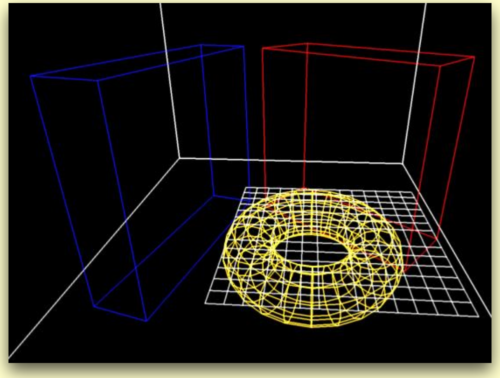


3D Rendering



Geometric Model

Color



Perspective

Visibility

Local Illumination

Smooth Shading

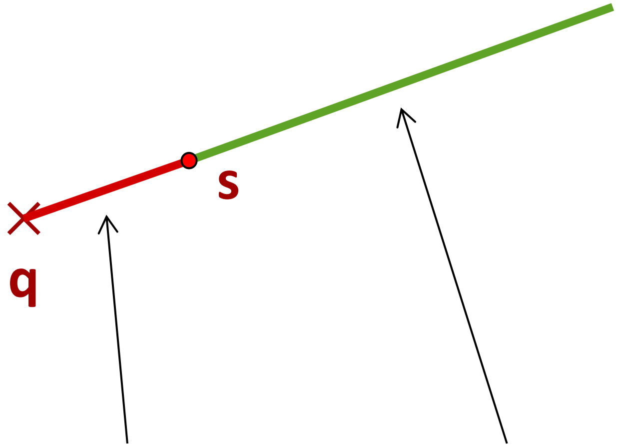
Simple Shadows

Global Illumination

More about homogenous coordinates

Projective Geometry

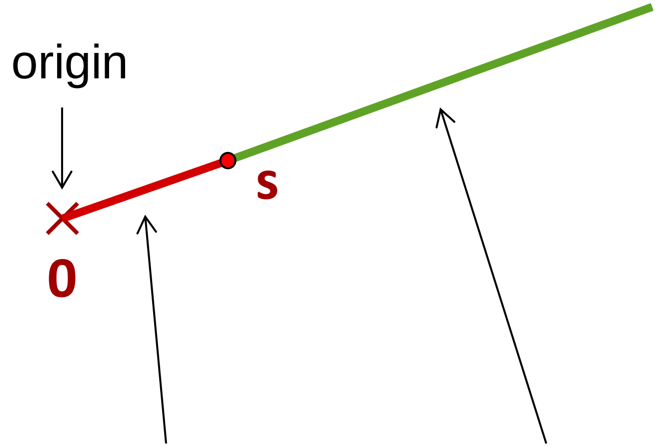
Constructing Projective Spaces



$(1-t)\mathbf{q} + t\mathbf{s}$
for $t \in [0,1]$

$(1-t)\mathbf{q} + t\mathbf{s}$
for $t > 1$

Constructing Projective Spaces



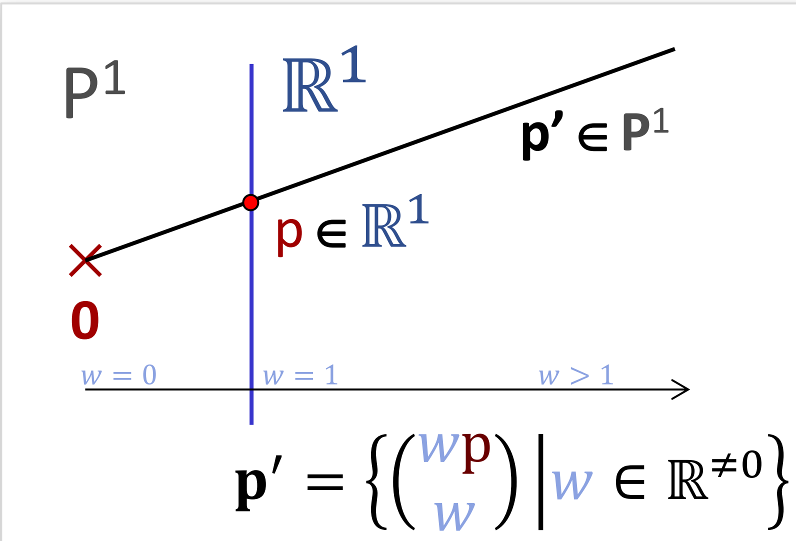
$(1-t)\mathbf{0} + t\mathbf{s}$
for $t \in [0,1]$

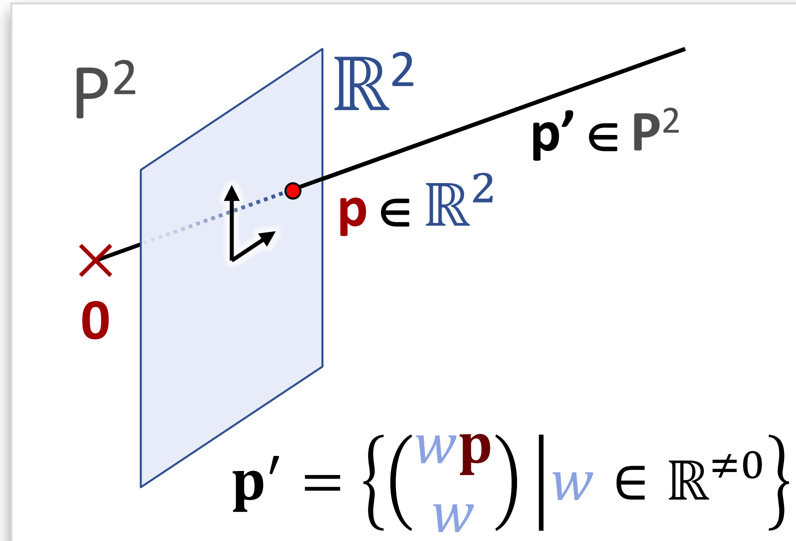
$(1-t)\mathbf{0} + t\mathbf{s}$
for $t > 1$

Since the first point is the origin,
we just have for all points along the ray:

$$\mathbf{s}' = t\mathbf{s} = \begin{pmatrix} ts_x \\ ts_y \end{pmatrix}$$

Constructing Projective Spaces

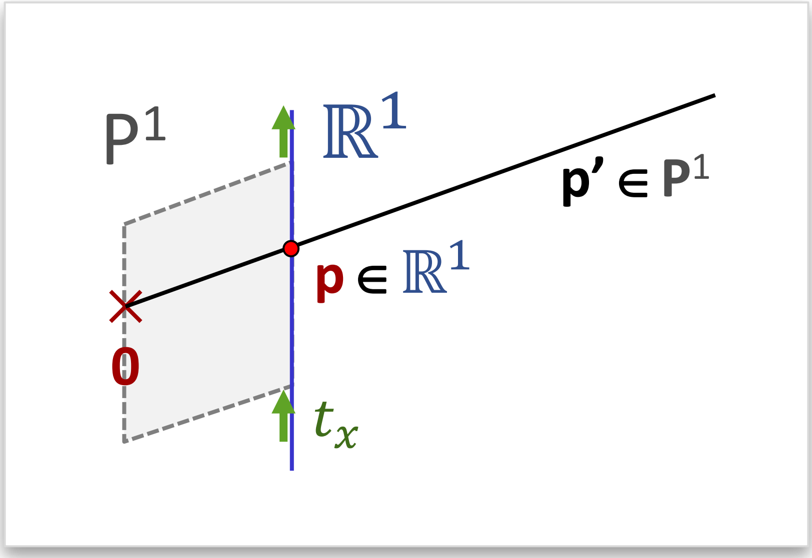




Projective Space P^d:

- Euclidean ("affine") space $\mathbb{R}^d$ embedded in $\mathbb{R}^{d+1}$
- At $w = 1$
- Identify all points on lines through the origin
  - *Representing* the same Euclidean point

Constructing Projective Spaces



Translations:

- Sheering of the projective space

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

= Translation of the embedded affine space

Normalization

Conversion between

- Cartesian coordinates (Euclidian space)
- Homogeneous coordinates (projective space)

$$\mathbf{x} \rightarrow \begin{pmatrix} w\mathbf{x} \\ w \end{pmatrix}$$

Cartesian coordinates
(Euclidian space)

homogenous coordinates
(projective space)

$$\frac{1}{w}\mathbf{x} \leftarrow \begin{pmatrix} \mathbf{x} \\ w \end{pmatrix}$$

normalization*)

*) overloaded name
do not confuse with $\mathbf{x}/||\mathbf{x}||$

Vectors & Points

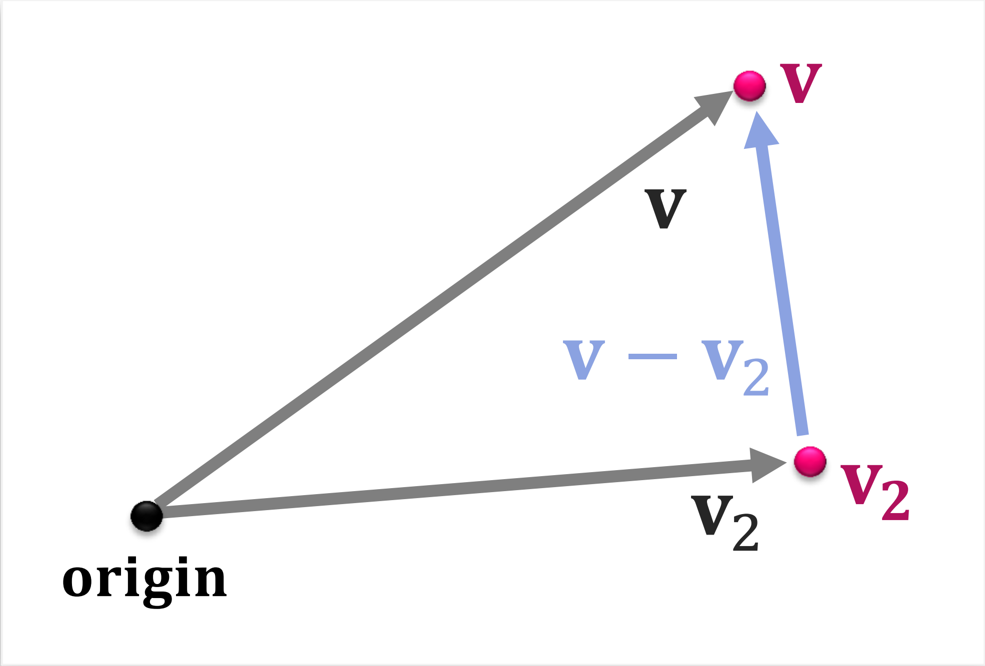
Interpretation

- Points: $\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}, w \neq 0$
- Vectors: $\begin{pmatrix} x \\ y \\ z \\ 0 \end{pmatrix}$ – "pure directions"

Vectors & Points

Rules

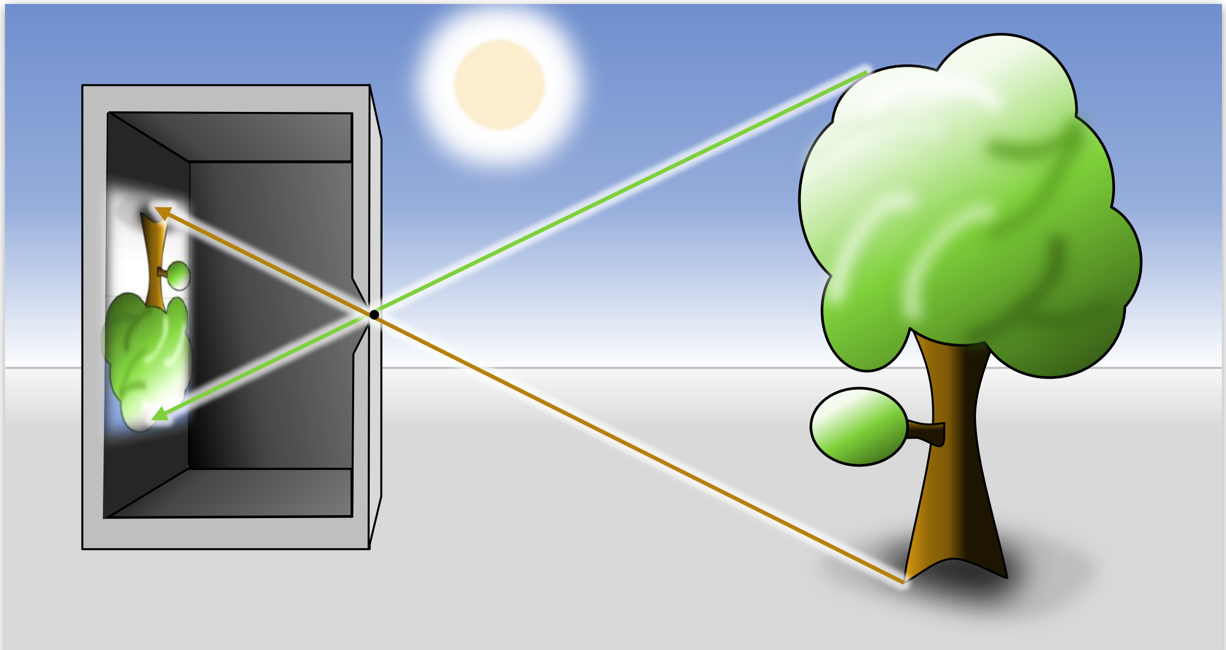
- Substracting points yields vectors
  - Normalize first!
- Vectors can be added to
  - Other vectors
  - Points (normalize first!)



Physics

Perspective Projection

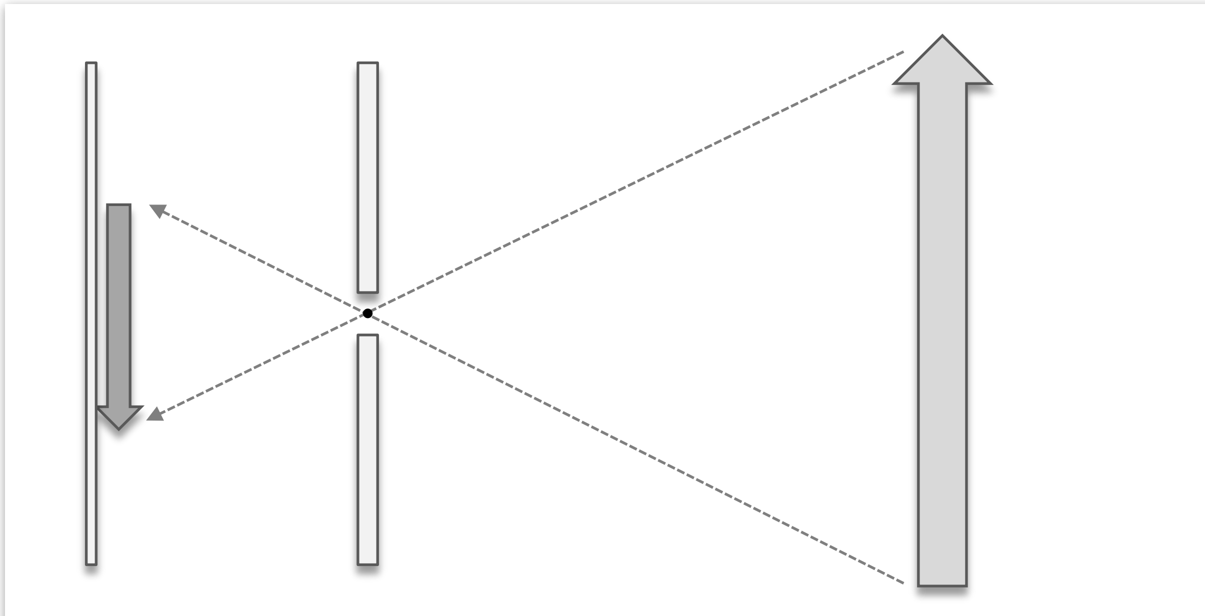
Pinhole Camera



Pinhole camera

- Create image by selecting rays of specific angles
- Low efficiency (small holes for sharp images)

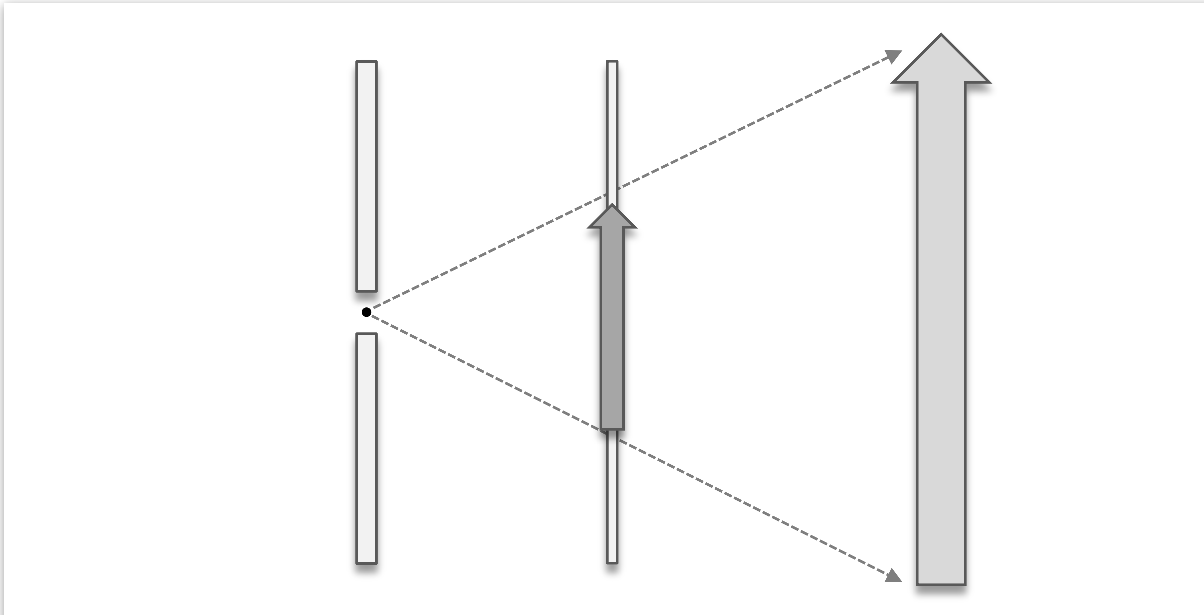
Pinhole Camera



Pinhole camera

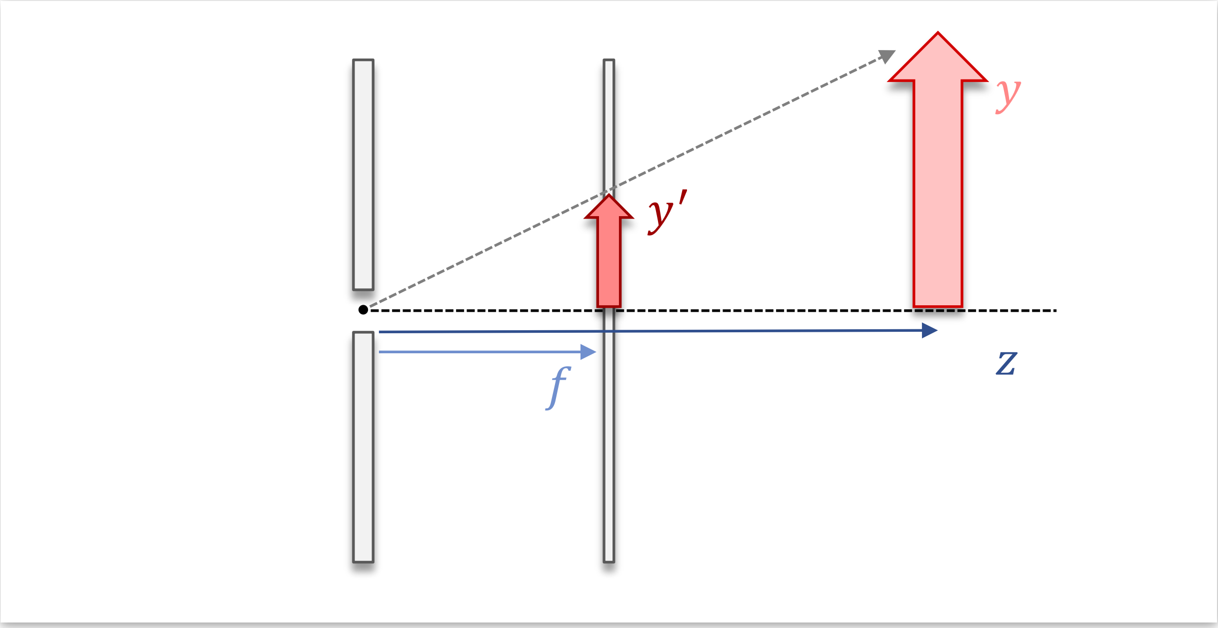
- Create image by selecting rays of specific angles
- Low efficiency (small holes for sharp images)

Pinhole Camera



Central Projection

Pinhole Camera

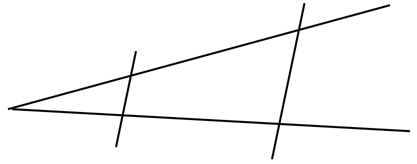


Central projection

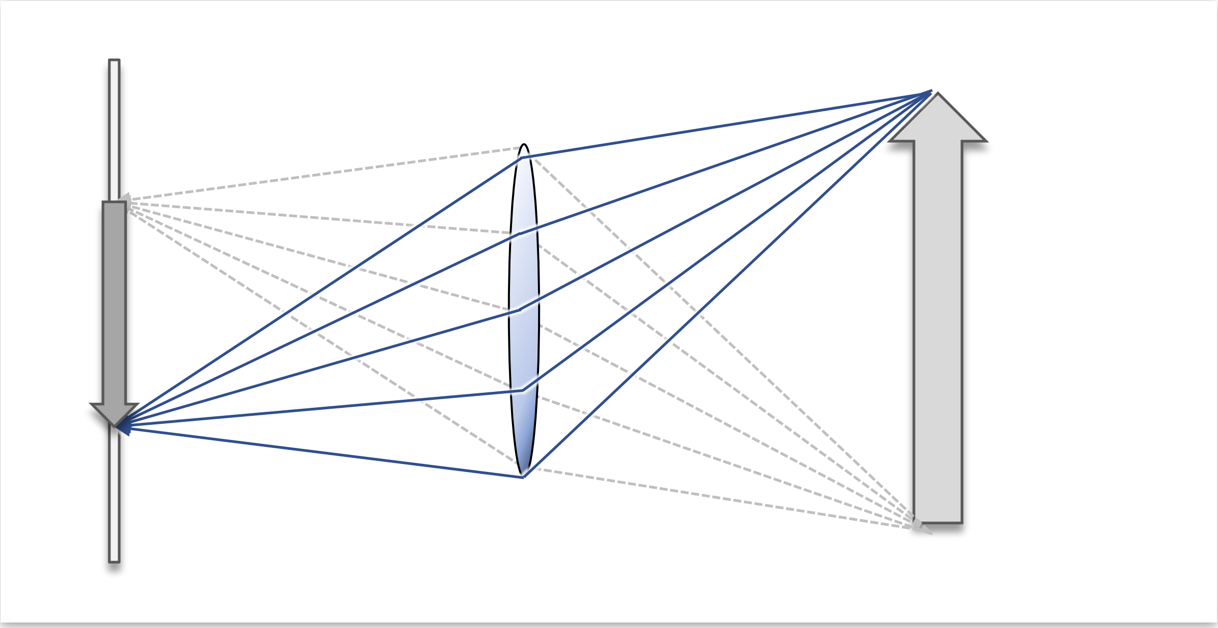
$$x' = f\frac{x}{z}$$

$$y' = f\frac{y}{z}$$

Proof:
Intercept theorem!



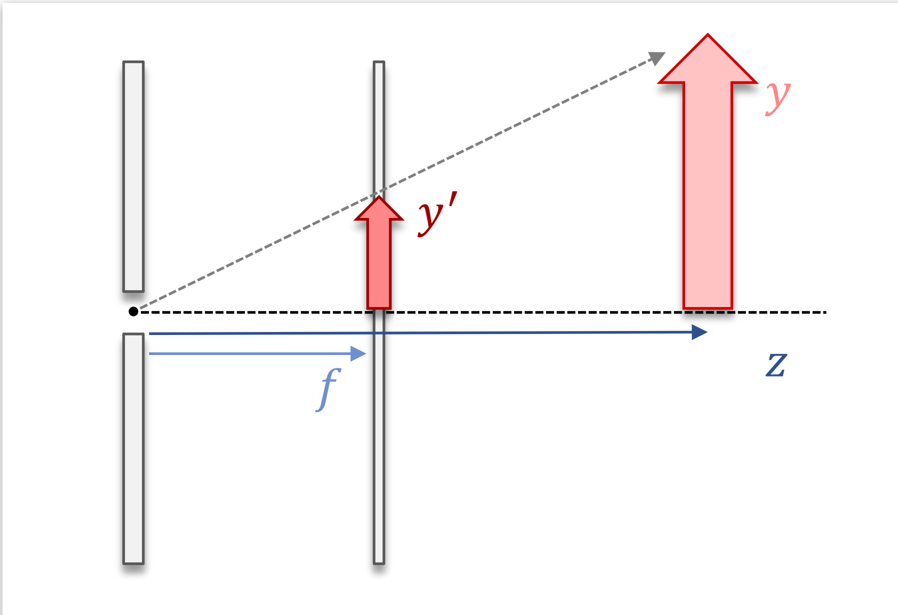
(Actual Camera)



Camera with Lens

- Higher efficiency (bundles many rays)
- Finite Depth of field
- We will consider pinhole cameras only.

Pinhole Camera



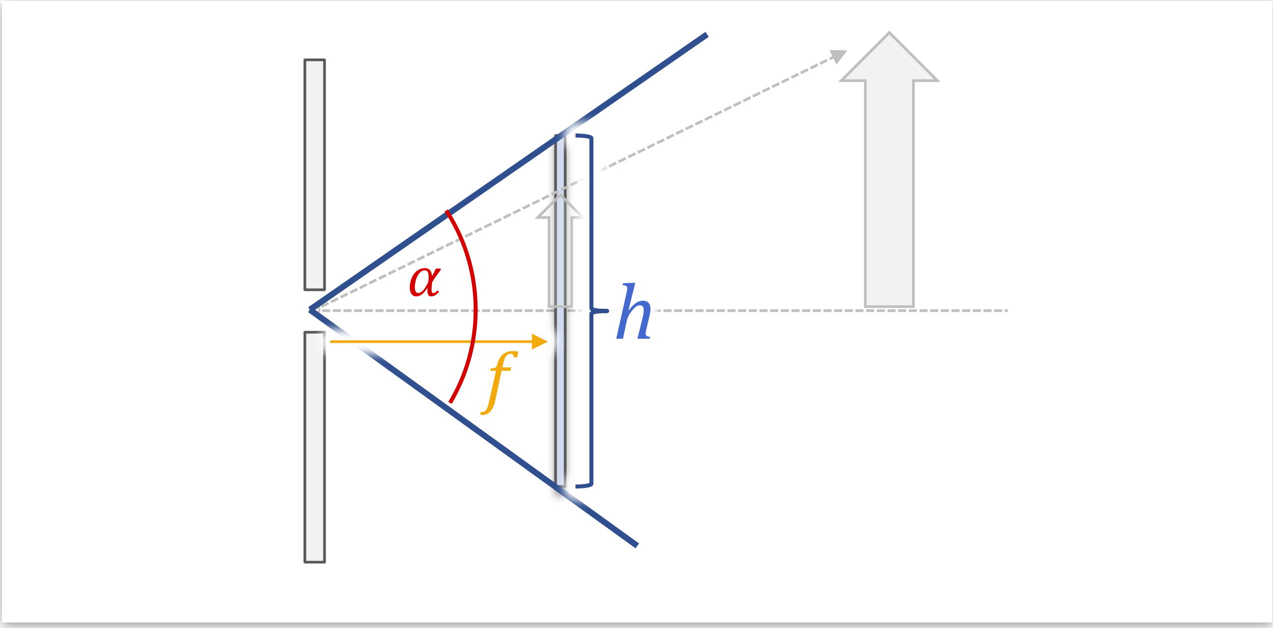
$$x' = f\frac{x}{z}$$

$$y' = f\frac{y}{z}$$

Undetermined degree of freedom

- Focal length vs. image size
- Source of a lot of confusion!

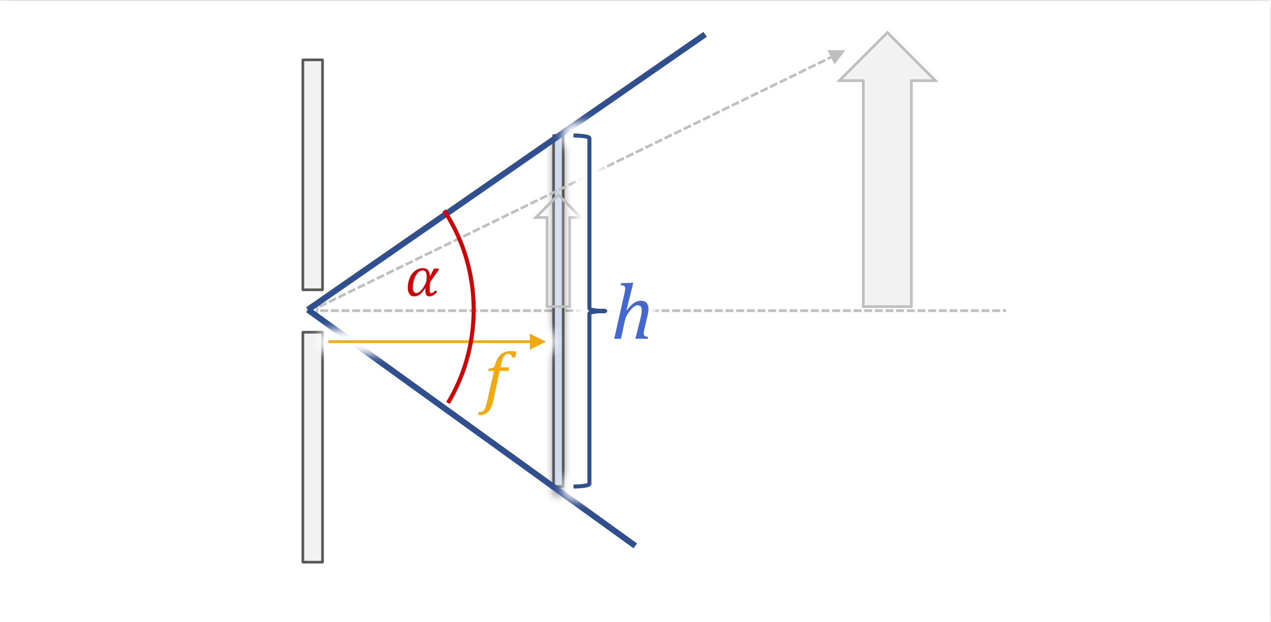
Pinhole Camera



Parameters

- h - size of the screen (pixels, cm, ± 1.0,...)
- f – focal length (classical photography)
- Meaningful parameter: α – viewing angle

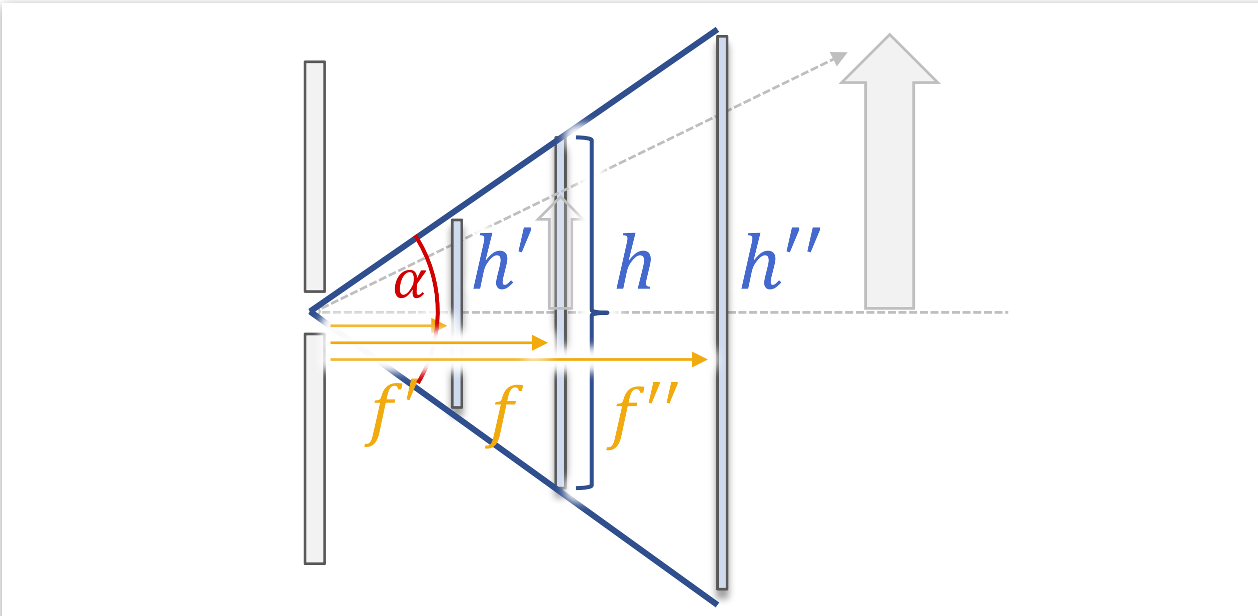
Pinhole Camera



Relation:

$$\tan\frac{\alpha}{2} = \frac{h}{2f}$$

Pinhole Camera

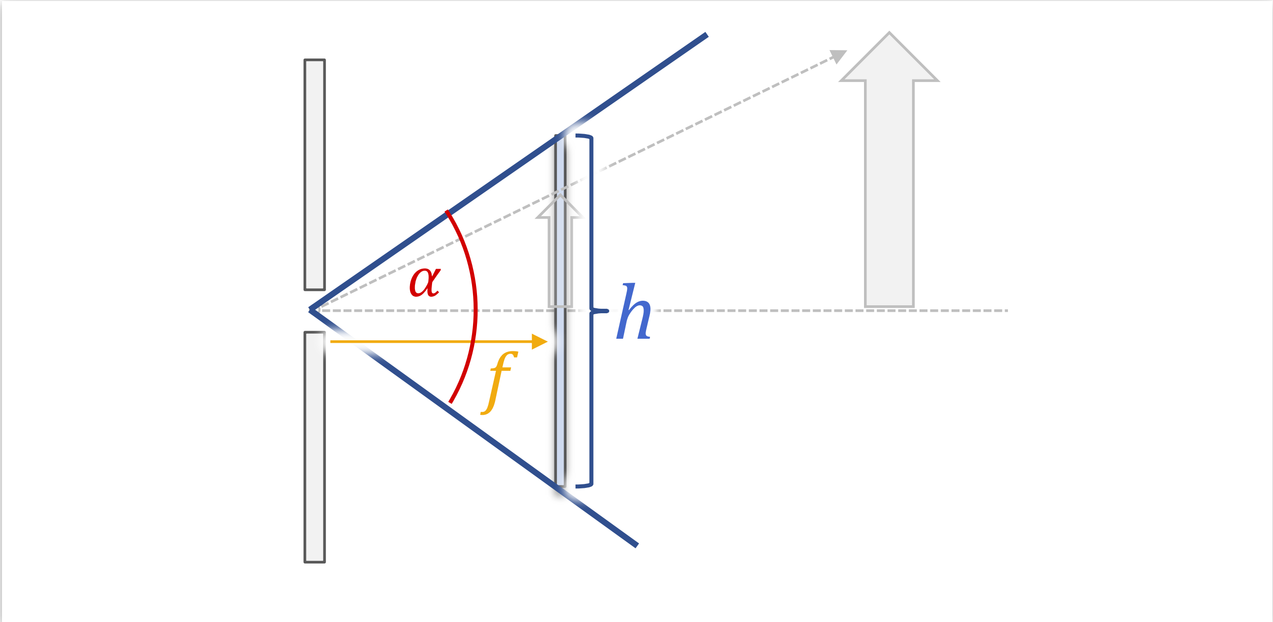


Invariance

$$\tan\frac{\alpha}{2} = \frac{h}{2f} = \frac{h'}{2f'} = \frac{h''}{2f''}$$

- Scaling *h* and *f* by a common factor: *no change*

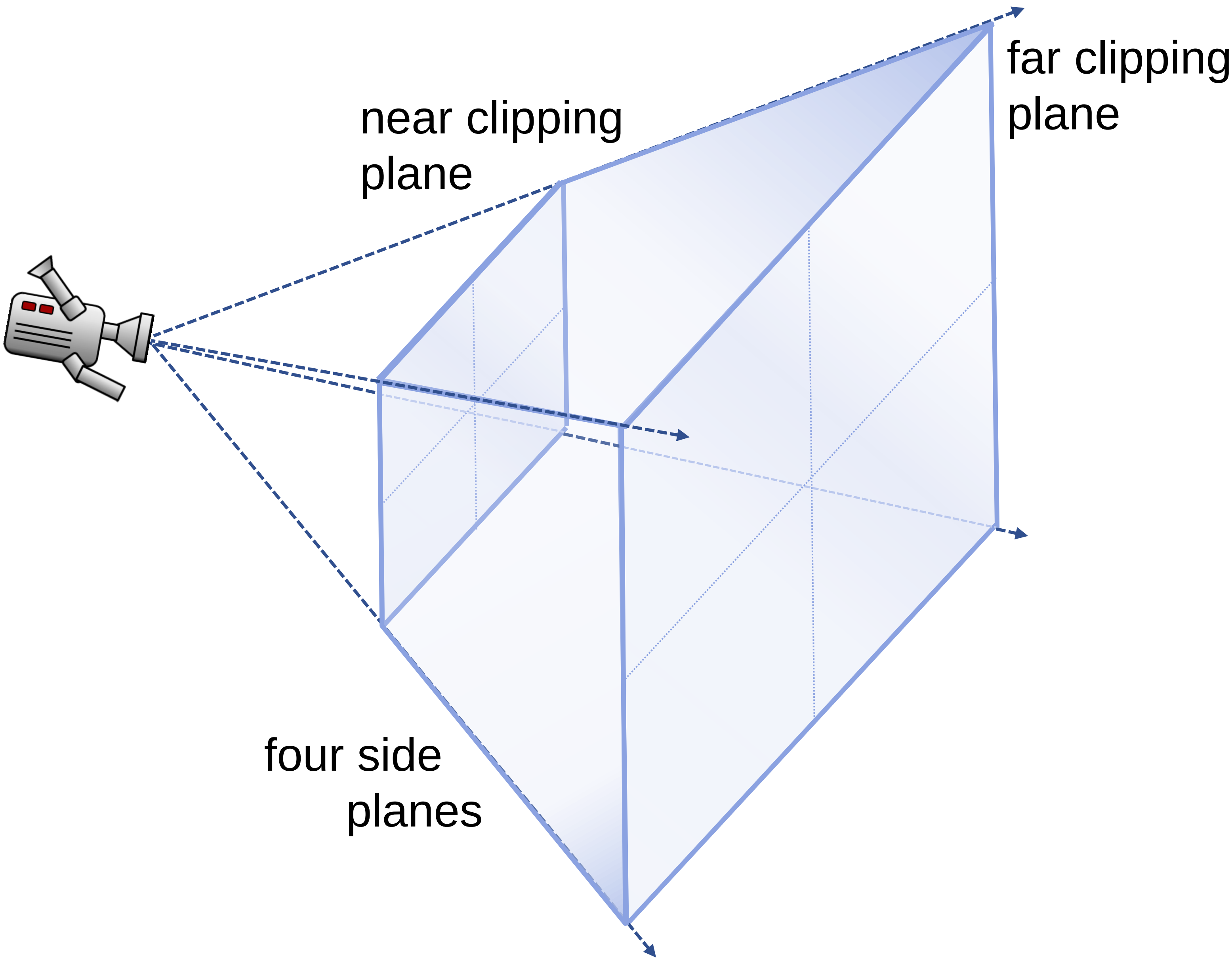
Pinhole Camera



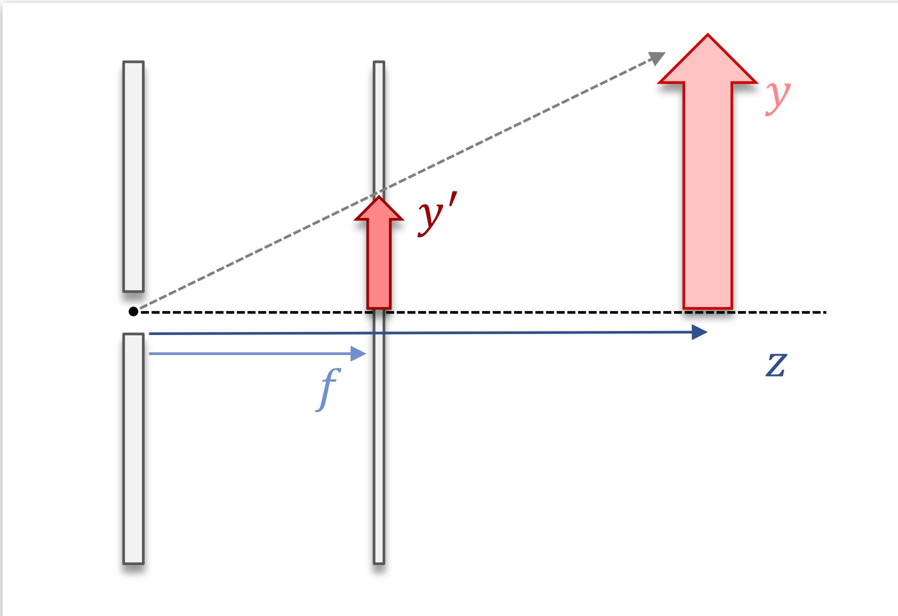
Typical choices (vertical angles)

- "Normal" perspective: $\alpha \approx 30°$ ("50mm" lens: 27°)
- Tele photography: $\alpha \approx 5° - 20°$ (275–70mm)
- Wide angle lens: $\alpha \approx 45° - 90°$ (28–12mm)

View Volume



Pinhole Camera



$$x' = f\,\frac{x}{z}$$

$$y' = f\,\frac{y}{z}$$

Our camera:

- Focus point: origin
- View direction: z-axis

Homogeneous Coordinates

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Projection Matrix } \mathbf{P}} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$x' = fx \qquad y' = fy \qquad z' = z - 1 \qquad w' = z$$

before normalization

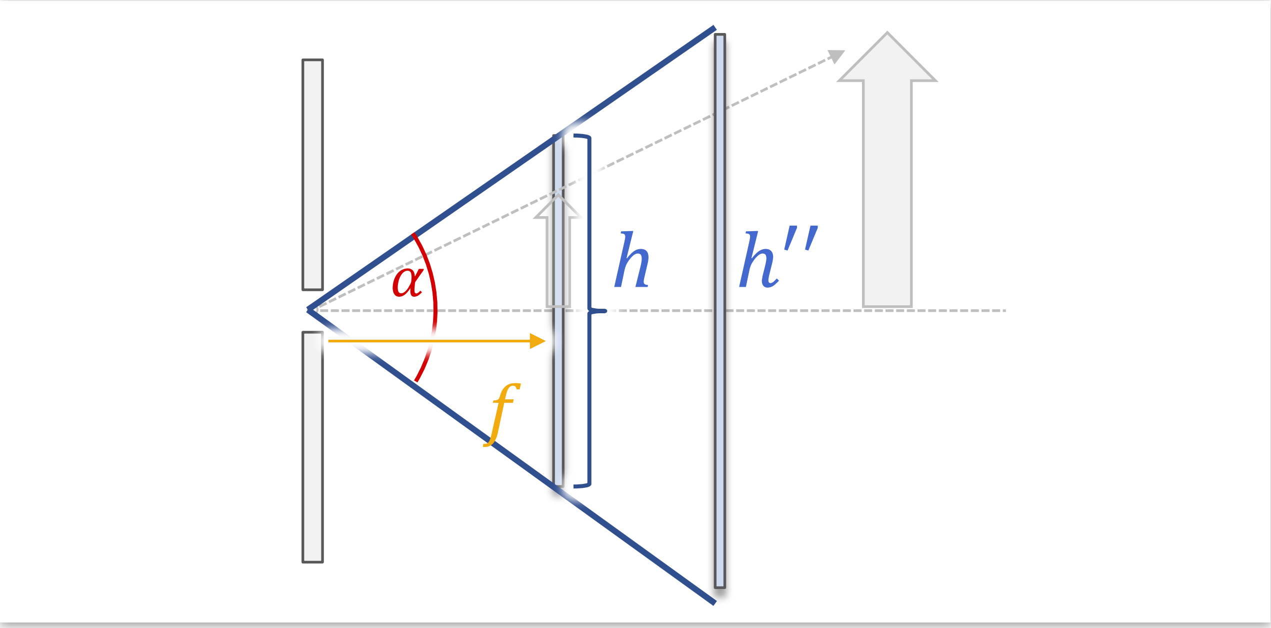
$$x' = f\frac{x}{z} \qquad y' = f\frac{y}{z} \qquad z' = \frac{z-1}{z} \qquad w' = 1$$

after normalization

Write in homogeneous coordinates

- Third row is arbitrary (for now), not used.

View transform

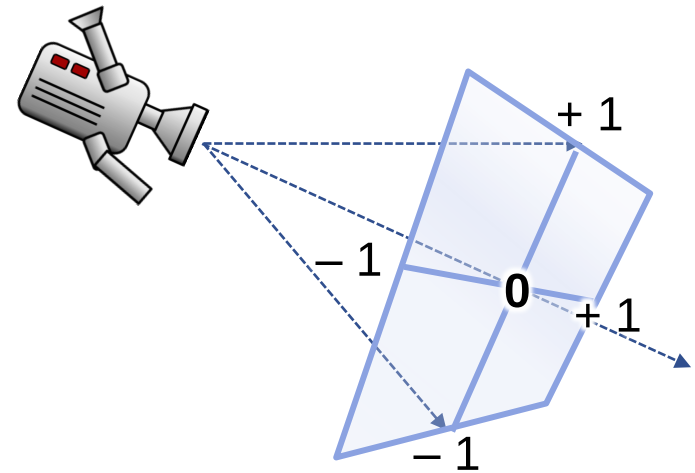


Reminder:

$$\tan\frac{\alpha}{2} = \frac{h}{2f}$$

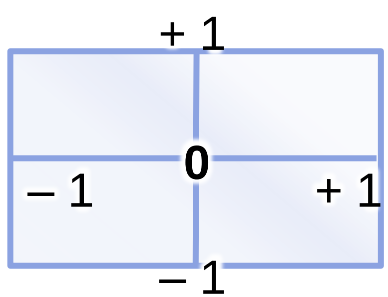
To Screen Coordinates

$$\begin{pmatrix} \frac{1}{\tan\left(\frac{\alpha}{2}\right)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\left(\frac{\alpha}{2}\right)} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Scale to unit screen coordinates

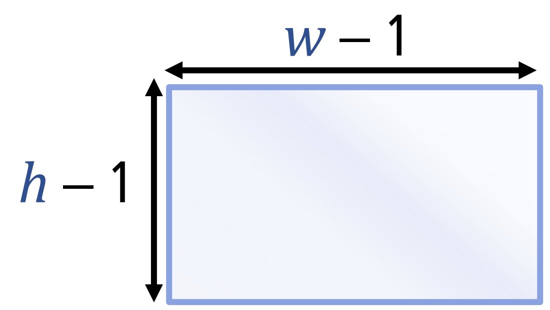
- We set f to 1 in previous matrix
- Third row is arbitrary (for now), not used.



normalized screen coordinates

Aspect Ratio

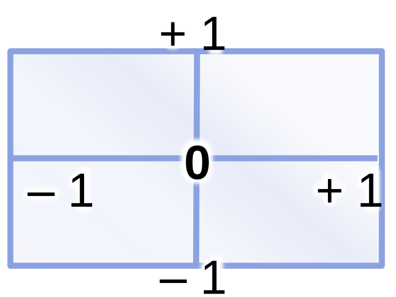
$$\begin{pmatrix} \frac{1}{\frac{w}{h} \cdot \tan\left(\frac{\alpha}{2}\right)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\left(\frac{\alpha}{2}\right)} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



non-square screen

Non-square screens?

- Screen: $w \times h$ pixels
- Aspect ratio $\frac{w}{h}$
- Different horizontal angle!



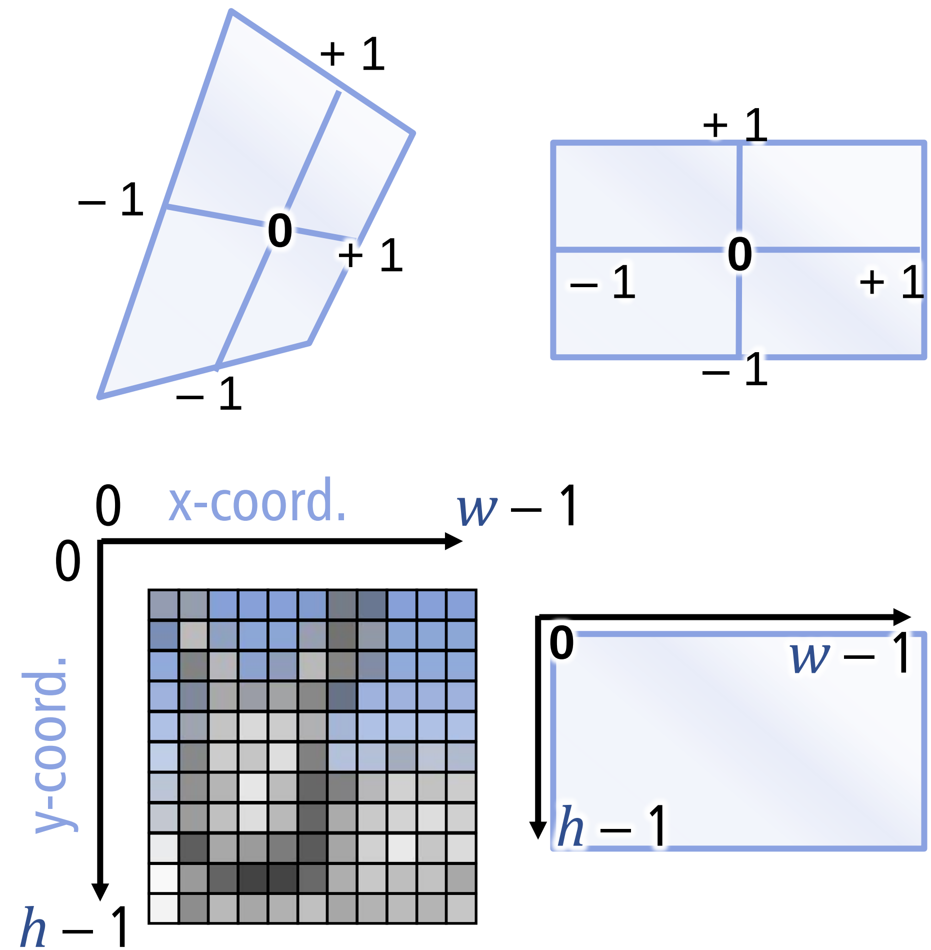
normalized screen coordinates

To Screen Coordinates

$$\begin{pmatrix} w/2 & 0 & 0 & w/2 \\ 0 & -h/2 & 0 & h/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Scale to pixels

- Third row is arbitrary (for now), not used.



To Screen Coordinates

$$\begin{pmatrix} \frac{h/2}{\tan\left(\frac{\alpha}{2}\right)} & 0 & 0 & \frac{w/2}{\tan\left(\frac{\alpha}{2}\right)} \\ 0 & -\frac{h/2}{\tan\left(\frac{\alpha}{2}\right)} & 0 & \frac{h/2}{\tan\left(\frac{\alpha}{2}\right)} \\ 0 & 0 & a & b \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Overall

- Multiply both

$$a = \frac{z_{far} + z_{near}}{z_{near} - z_{far}}$$

$$b = \frac{2 \cdot z_{near} \cdot z_{far}}{z_{near} - z_{far}}$$

Additionally:

Also scale + shift such that

$$z' = \frac{z - 1}{z}$$

are in value [0..1] for inputs

$$z \in [z_{near}, z_{far}]$$

Summary

Projection matrix

$$\mathbf{P} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Projection & conversion to screen coords

$$\mathbf{P}_s = \begin{pmatrix} w/2 & 0 & 0 & w/2 \\ 0 & -h/2 & 0 & h/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\frac{w}{h}\tan\left(\frac{\alpha}{2}\right)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\left(\frac{\alpha}{2}\right)} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

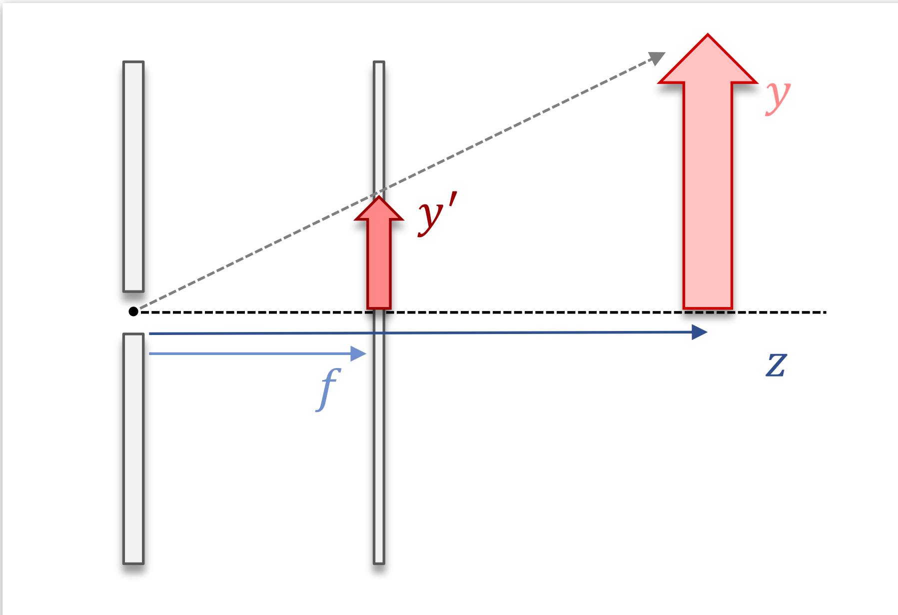
$(f = 1)$

scaling to pixels, upper left origin

normalized screen coord's

projection matrix

General Camera



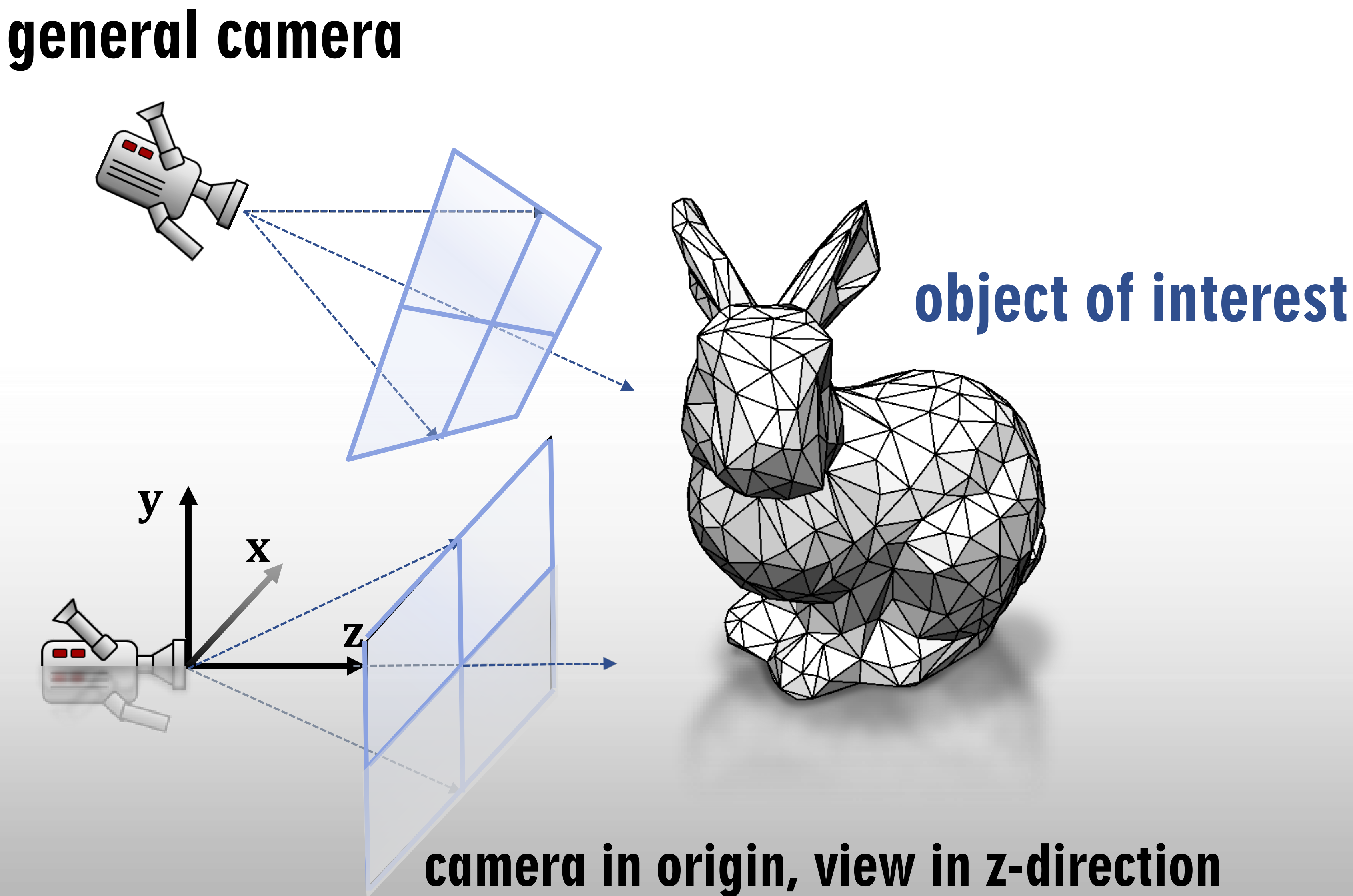
$$x' = f\frac{x}{z}$$

$$y' = f\frac{y}{z}$$

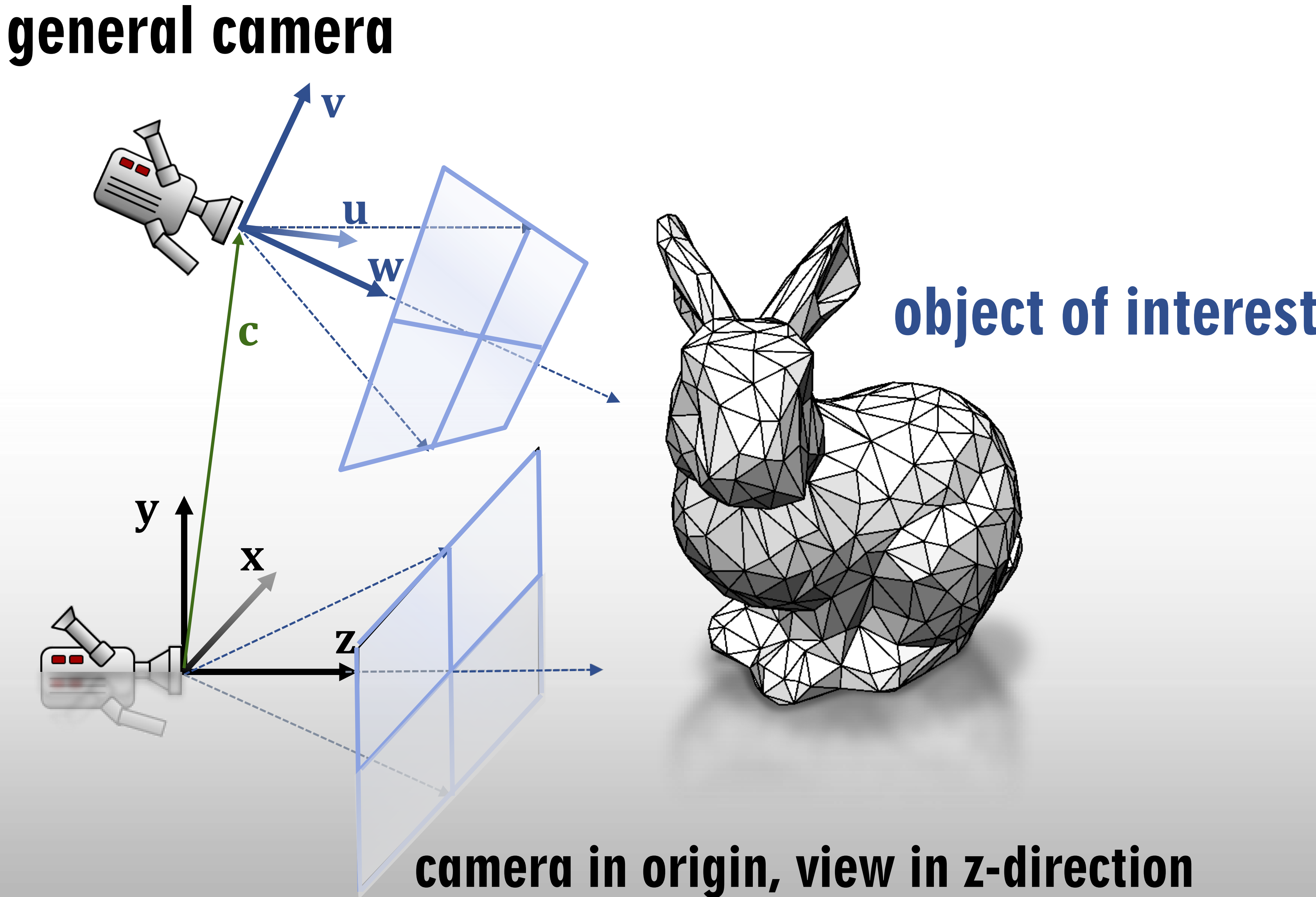
Our camera so far:

- Focus point: origin
- View direction: z-axis
- General position/orientation?

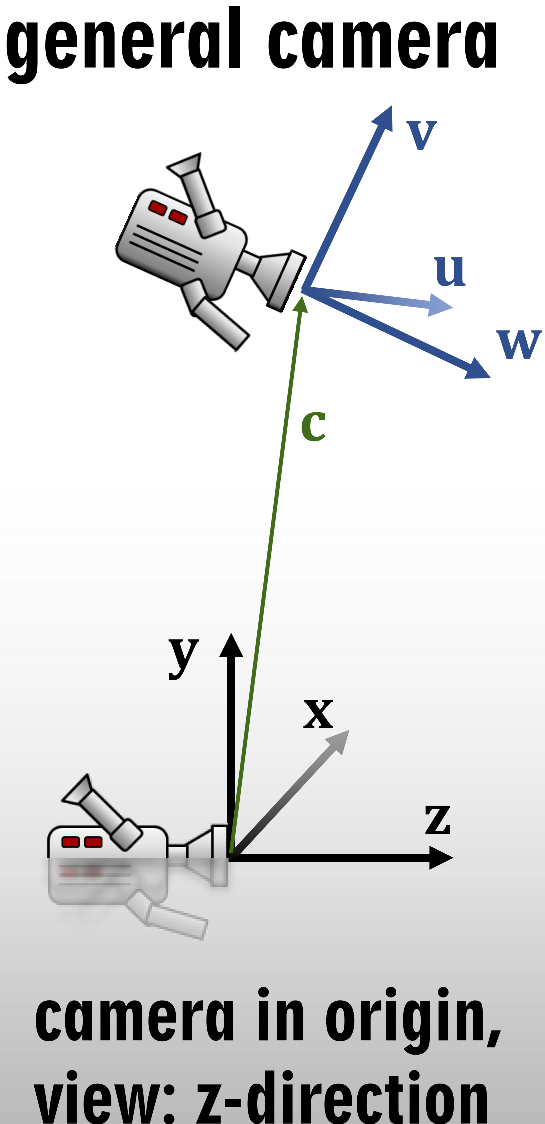
General Camera



General Camera



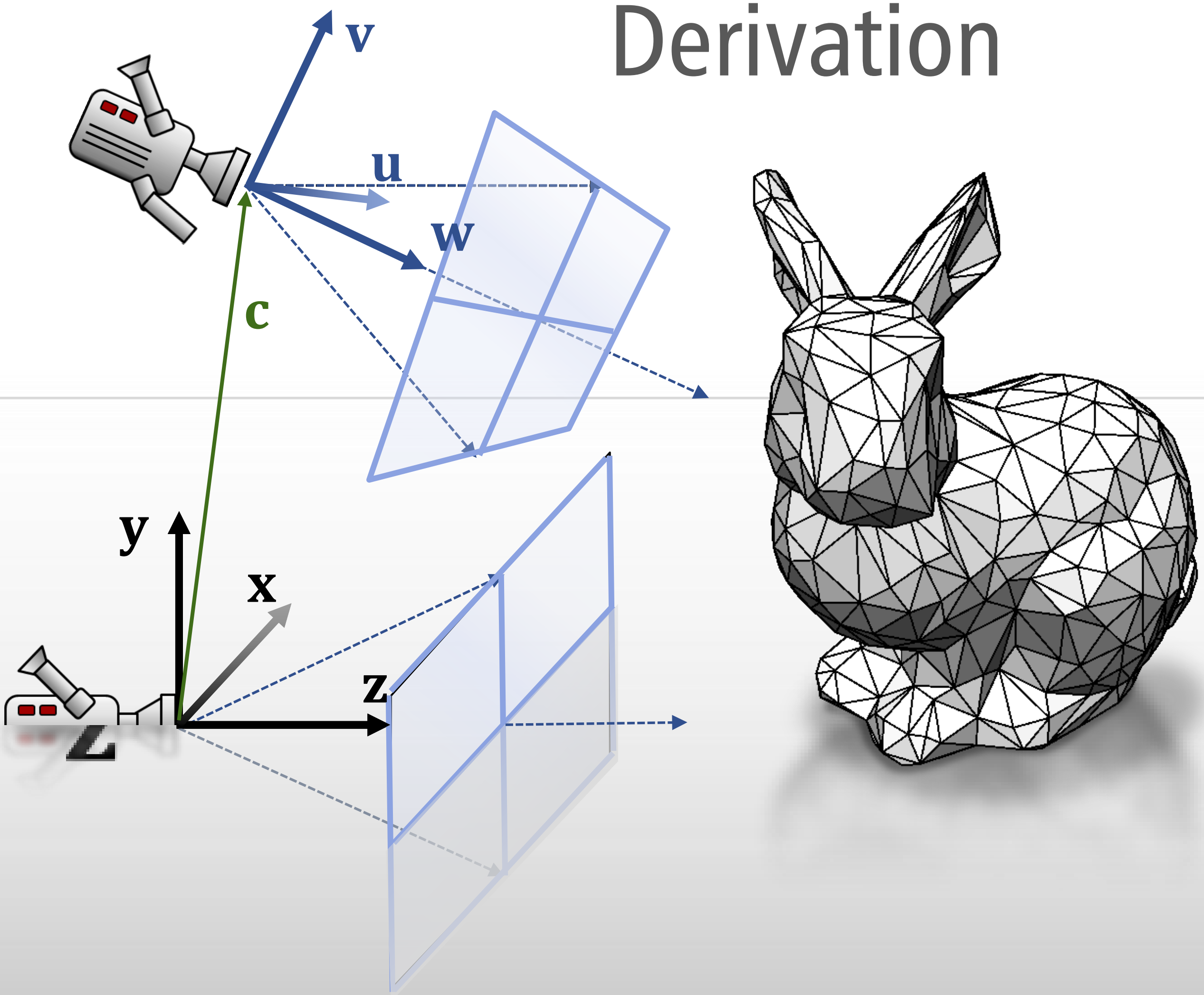
General Camera



Camera coordinate system $(\mathbf{u}, \mathbf{v}, \mathbf{w})$
Origin: $\mathbf{c}$

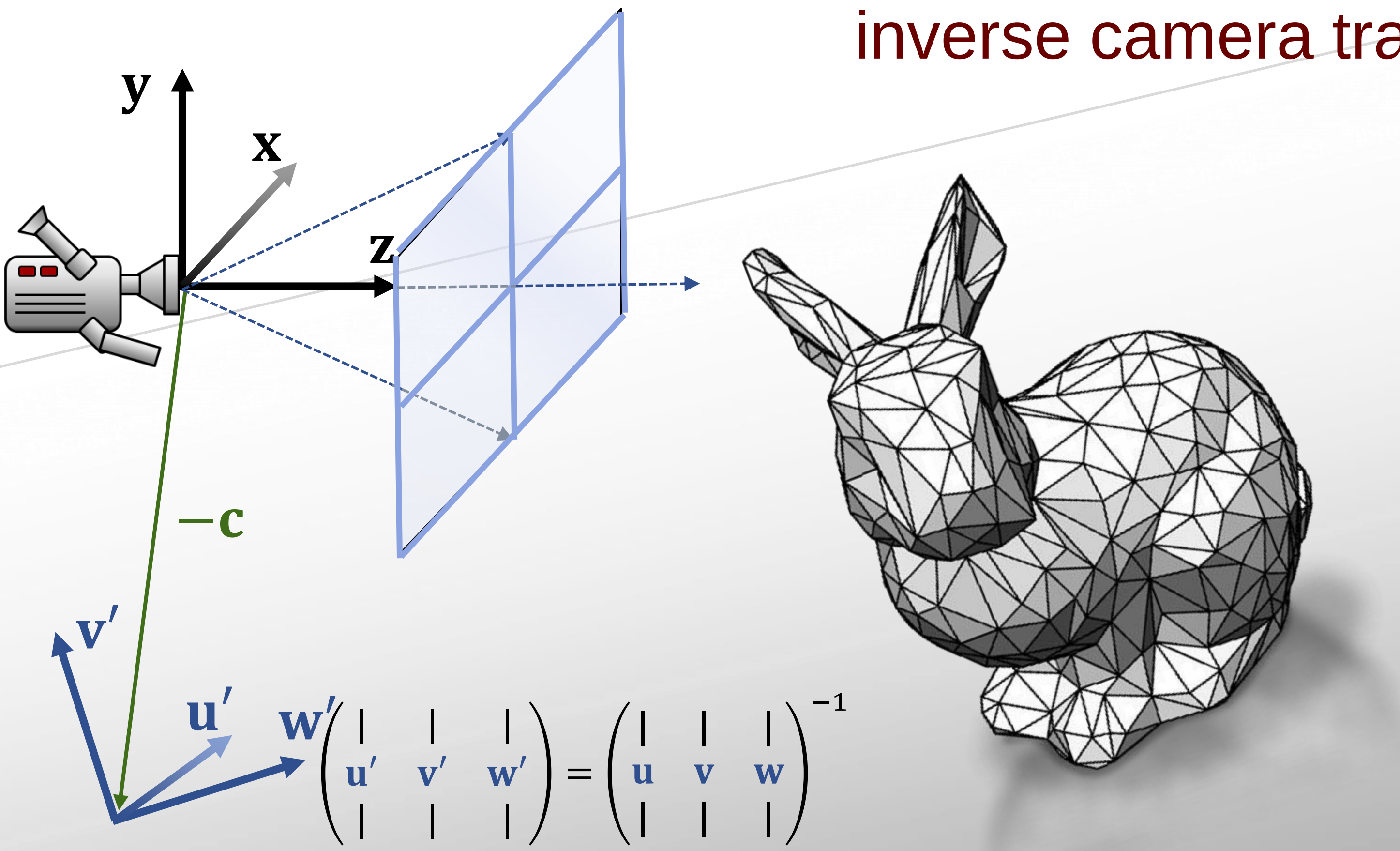
Standard coordinates $(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$

Derivation



Derivation

Same effect:
Transform the world with inverse camera transform

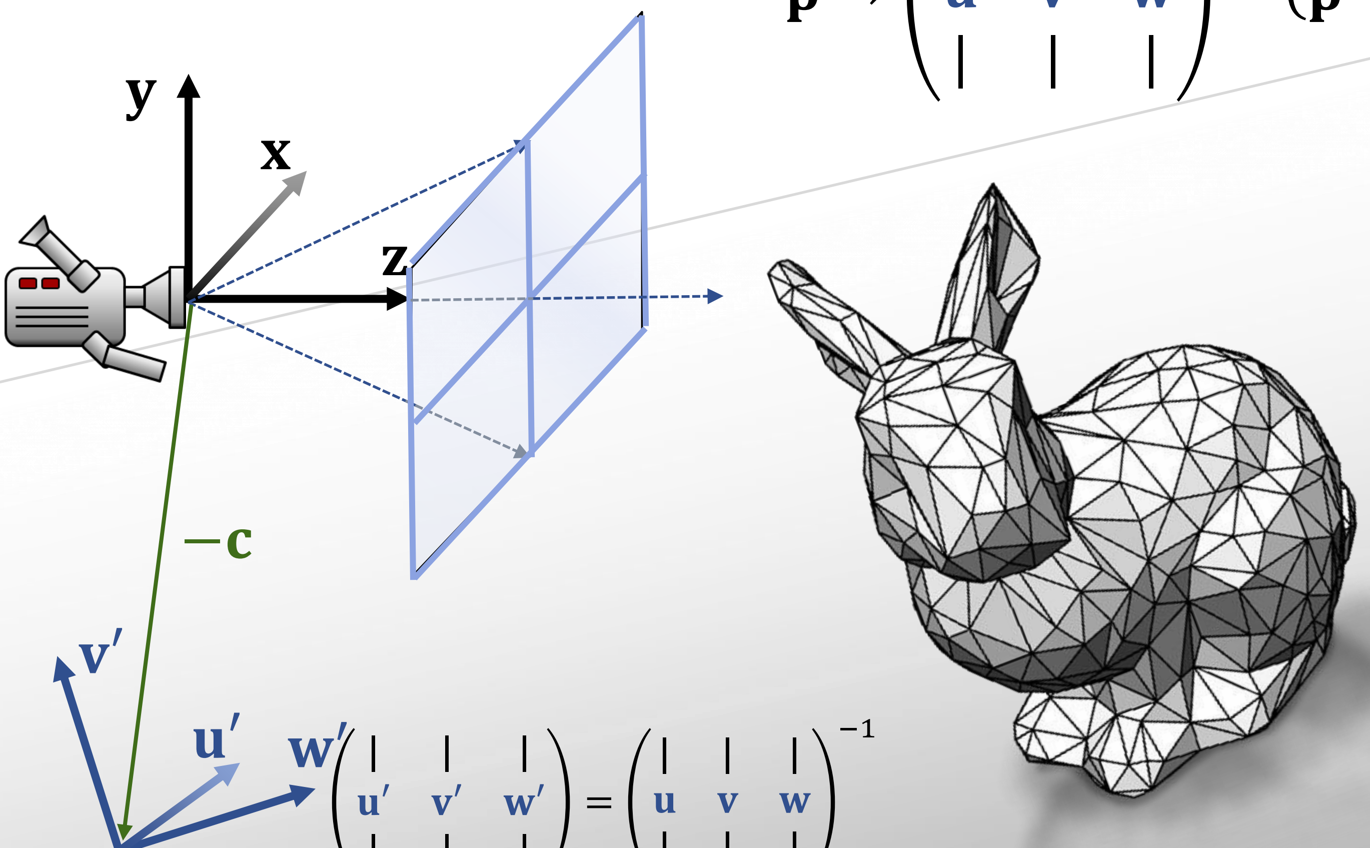


$$\begin{pmatrix} | & | & | \\ \mathbf{u}' & \mathbf{v}' & \mathbf{w}' \\ | & | & | \end{pmatrix} = \begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{v} & \mathbf{w} \\ | & | & | \end{pmatrix}^{-1}$$

Derivation

Transform:

$$\mathbf{p} \rightarrow \begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{v} & \mathbf{w} \\ | & | & | \end{pmatrix}^{-1} (\mathbf{p} - \mathbf{c})$$

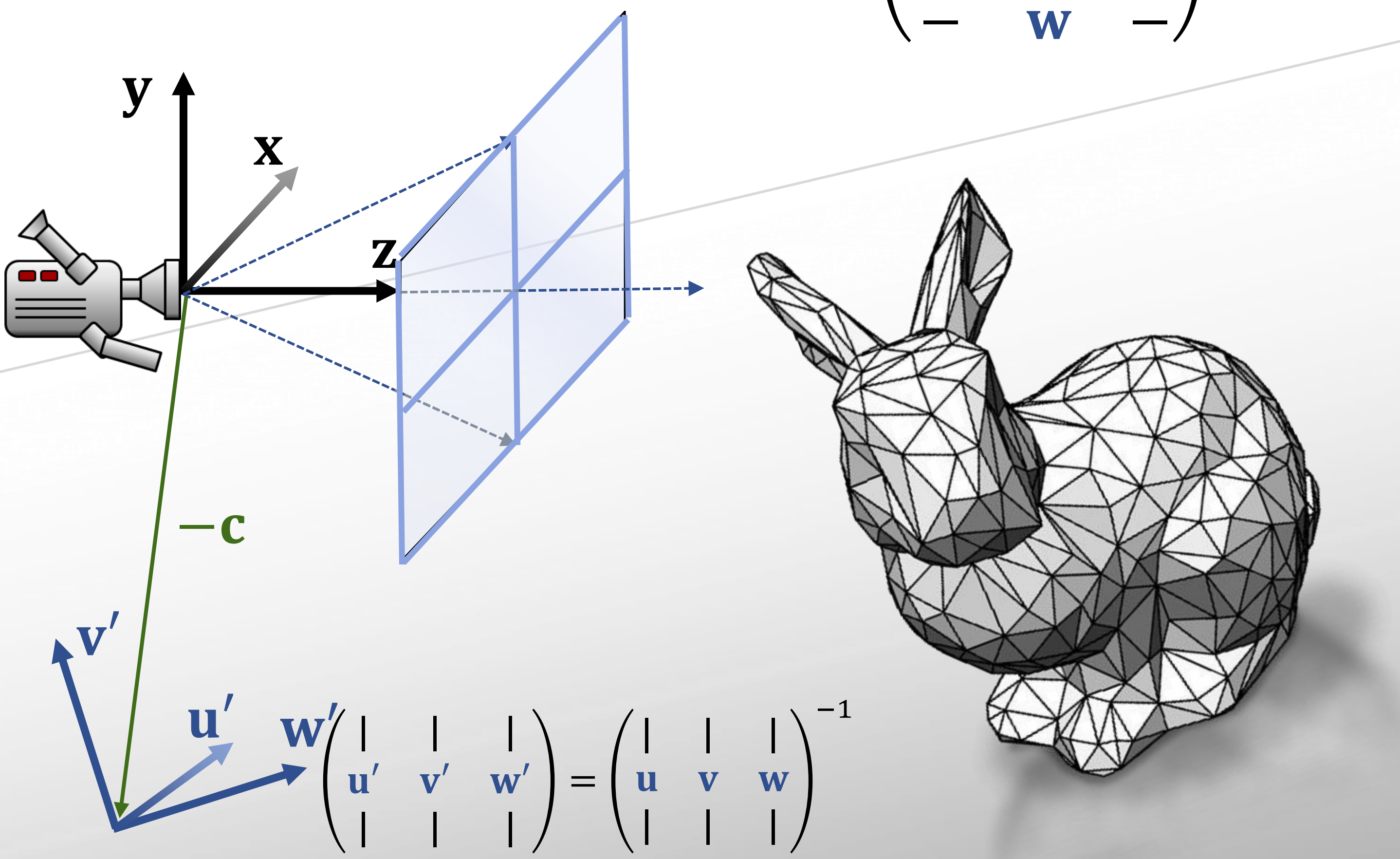


$$\begin{pmatrix} | & | & | \\ \mathbf{u}' & \mathbf{v}' & \mathbf{w}' \\ | & | & | \end{pmatrix} = \begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{v} & \mathbf{w} \\ | & | & | \end{pmatrix}^{-1}$$

Derivation

Transform: $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ orthogonal!

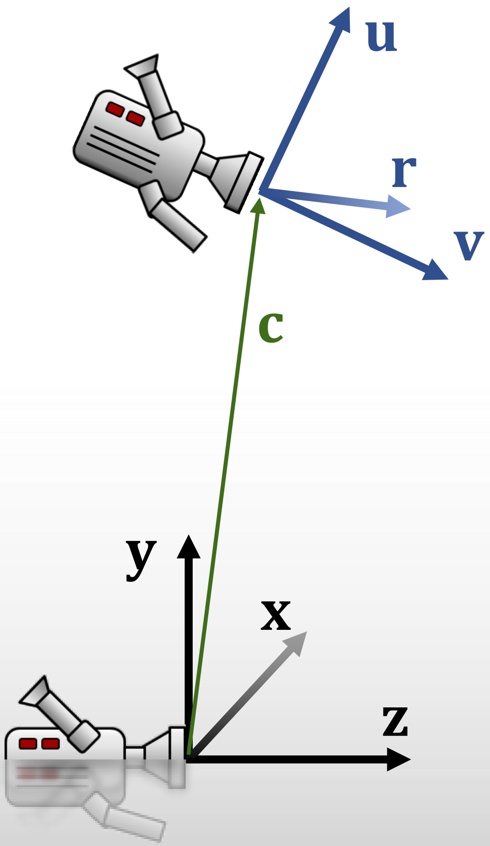
$$\mathbf{p} \to \begin{pmatrix} - & \mathbf{u} & - \\ - & \mathbf{v} & - \\ - & \mathbf{w} & - \end{pmatrix} (\mathbf{p} - \mathbf{c})$$



$$\begin{pmatrix} | & | & | \\ \mathbf{u}' & \mathbf{v}' & \mathbf{w}' \\ | & | & | \end{pmatrix} = \begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{v} & \mathbf{w} \\ | & | & | \end{pmatrix}^{-1}$$

General Camera

general camera



camera in origin,
view: z-direction

Camera coordinate system $(\mathbf{u}, \mathbf{r}, \mathbf{v})$
Origin: $\mathbf{c}$

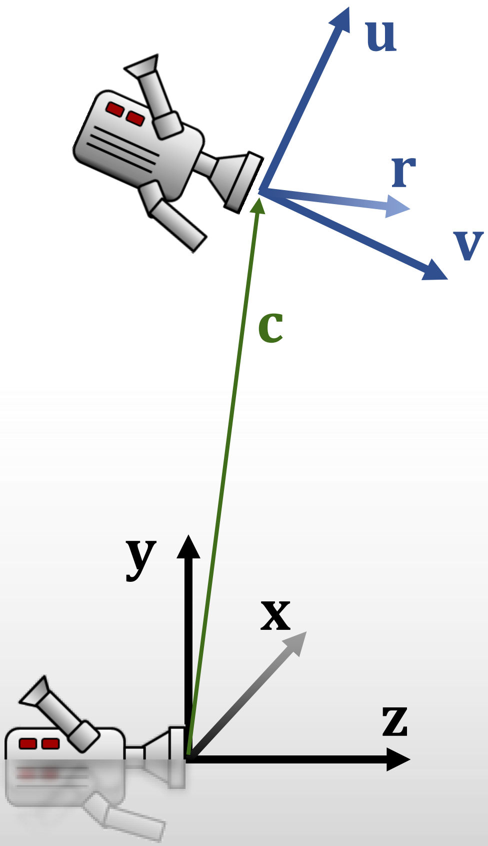
Transform:

$$\mathbf{p} \to \begin{pmatrix} - & \mathbf{u} & - \\ - & \mathbf{v} & - \\ - & \mathbf{w} & - \end{pmatrix} (\mathbf{p} - \mathbf{c})$$

Standard coordinates $(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$

General Camera

general camera



Camera coordinate system $(\mathbf{u}, \mathbf{r}, \mathbf{v})$
Origin: $\mathbf{c}$

Homogeneous:

$$\mathbf{p} \rightarrow \begin{pmatrix} - & \mathbf{u} & - & | \\ - & \mathbf{v} & - & -\mathbf{c}' \\ - & \mathbf{w} & - & | \\ 0 & 0 & 0 & 1 \end{pmatrix} (\mathbf{p})$$

$$\mathbf{c}' = \begin{pmatrix} - & \mathbf{u} & - \\ - & \mathbf{v} & - \\ - & \mathbf{w} & - \end{pmatrix} \mathbf{c}$$

Summary

Projection (screen coord's)

$$(f = 1)$$

$$\mathbf{P}_s = \begin{pmatrix} h/2 & 0 & 0 & w/2 \\ 0 & -h/2 & 0 & h/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\tan\left(\frac{\alpha}{2}\right)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\left(\frac{\alpha}{2}\right)} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Add View Matrix

Benefit:

Still only one overall 4×4 matrix to multiply with!

$$\mathbf{P}_s \cdot \begin{pmatrix} - & \mathbf{u} & - & | \\ - & \mathbf{v} & - & -\mathbf{c}' \\ - & \mathbf{w} & - & | \\ 0 & 0 & 0 & 1 \end{pmatrix}$$