Image Processing:
1. Camera Models

Aleix M. Martinez
aleix@ece.osu.edu

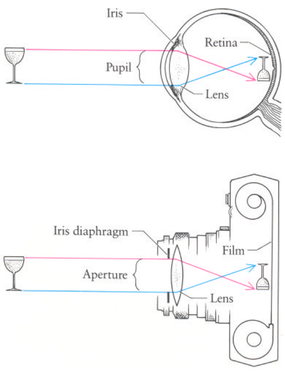
Why Camera Modeling?

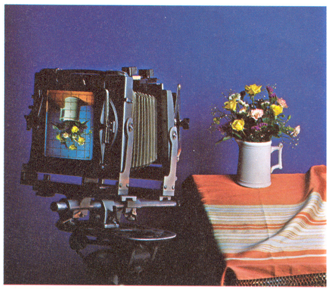
- Think it this way: You look at the world trough the lens of a camera and take a picture.

WORLD → CAMERA → PICTURE

- Now you show the photo to a friend. S/he needs to interpret the image; i.e.,

WORLD ← CAMERA MODEL ← PICTURE





Definition: A **camera** is an imaging device that captures light and imprints it into a translucent plate (which is usually located at the back of the device).

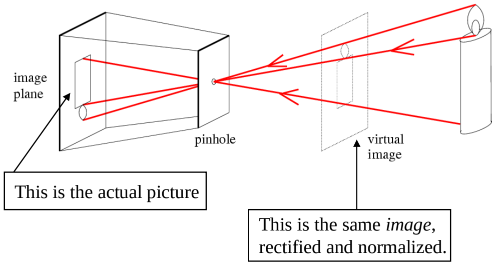
Another applications of camera modeling: Rendering

- Imagine you want to superimpose a graphic animation on top of a football field. To be able to draw the projection of a 3D object on a 2D image of a 3D surface, you first need to recover the 3D parameters of the "world."
- This is known as rendering.

What do we need to do, then?

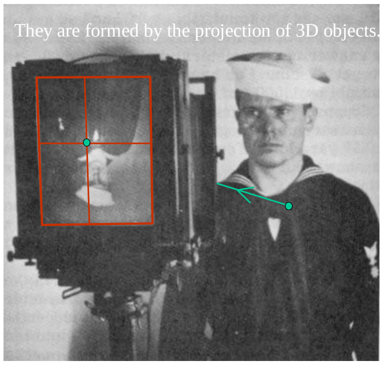
- In this part of the course, we will formulate the most useful model: **the pinhole camera**.
- Pinhole cameras can be modeled using two main types of projections:
  - Prespective projection.
  - Affine projections.
- These do not consider lenses. These are more difficult to model and **not** as useful.

Pinhole Cameras



This is the actual picture

This is the same *image*, rectified and normalized.

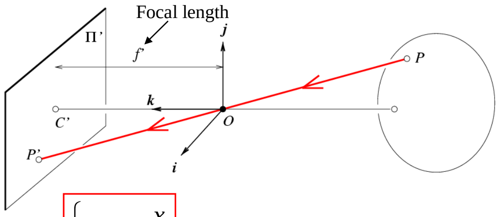


They are formed by the projection of 3D objects.

Perspective Projection

- We now know what a pinhole camera is.
- Let’s see how we can model the projection of a 3D *world* point to a 2D *image* point.
- We will start defining the most realistic projection.
- Later, we will define simplifications of this.
- Simplifications are useful for computational reasons only.

Pinhole Perspective Equation



$$\begin{cases} x' = f'\dfrac{x}{z} \\ y' = f'\dfrac{y}{z} \end{cases}$$

NOTE: *z* is always negative.

You can normalize the image plane (in front of the pinhole) to solve this problem.

The perspective equation

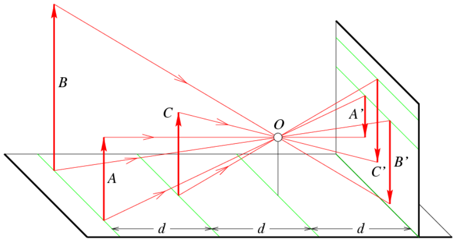
- We have a 3D world point **P**=$(x,y,x)^T$.
- The image point (as described by the 3D world coordinate system) is **P**’=$(x',y',z')^T$.
- Note that **P**, **P**’ and the origin **O** are collinear. This means: $\overrightarrow{\mathbf{OP}}'=\lambda\overrightarrow{\mathbf{OP}}$.
- I.e., $\begin{cases} x' = \lambda x \\ y' = \lambda y \\ z' = \lambda z \Leftrightarrow f' = \lambda z \end{cases}$

- From: $\begin{cases} x' = \lambda x \\ y' = \lambda y \\ z' = \lambda z \Leftrightarrow f' = \lambda z \end{cases}$

we have λ = x'/x = y'/y = f'/z.

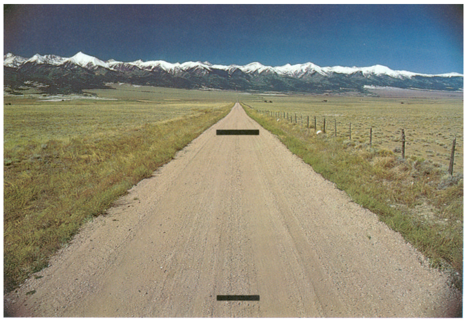
- Hence, $\begin{cases} x' = f'\dfrac{x}{z} \\ y' = f'\dfrac{y}{z} \end{cases} \Leftrightarrow \underbrace{\mathbf{p}' = \begin{pmatrix} x' \\ y' \end{pmatrix} = (x', y')^T}_{\text{Image Point}}.$

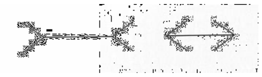
Some properties of the perspective projection



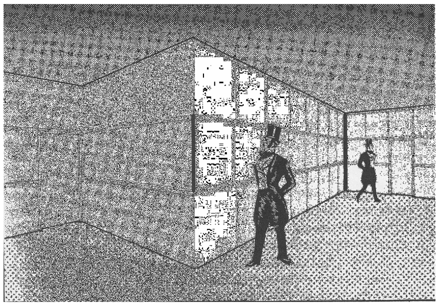
1. The image obtained using perspective projection is inverted.
2. The apparent size of objects depends on their distance from the camera. E.g., some vectors have the same length on the image, but not in the 3D word (see previous slide).

- This second property is the physics behind two well-known visual illusions (see the two slides that follow).



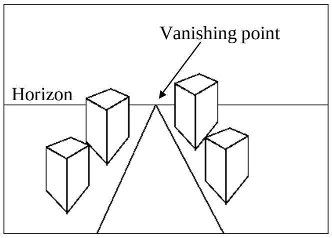


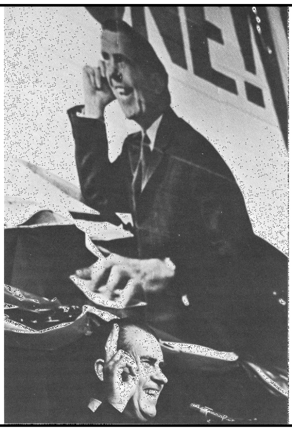
3D recovery: It's been suggested that this may be used to discern between convex and concave plane intersections.



1. The (perspective) projection of two parallel lines from 3D to 2D converges into a point. This point is known as: *the vanishing point* (see next slide).
2. The line where all parallel lines converge is known as: *the horizon*.

- Note that two parallel lines that are also parallel to the image plane will converge at infinity.





Other distortions:

The picture of a picture is a distorted image.

Note the difference in shape between the “candidate” as seen above and below.

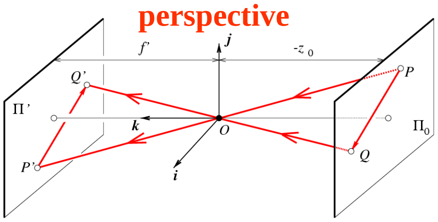
Affine Projections

- We have seen that in perspective projection, we need to know the focal length of the camera *f* and the depth values for each of the image points *z*, because

$$\begin{cases} x' = f'\dfrac{x}{z} \\ y' = f'\dfrac{y}{z}. \end{cases}$$

- We can change the values of *f* and *z* for a single constant (to be specified by us, *m* = *f*/*z*). That is, $\begin{cases} x' = mx \\ y' = my. \end{cases}$
- This projection is known as *weak-perspective*.
- In this case, *m* can be the ratio between a known *f* and the average distance from the camera to the object, or any other option.
- When *m*=1, this projection is called *orthographic*.

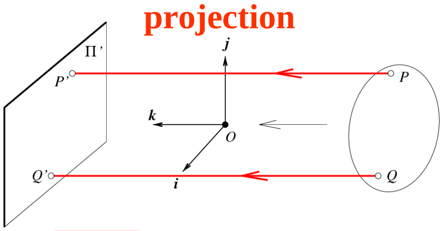
Affine Projection: Weak-perspective



$$\boxed{\begin{cases} x' = -mx \\ y' = -my \end{cases}}$$ where $m = -\dfrac{f'}{z_0}$ is the magnification.

When the scene relief is small compared to its distance from the camera, *m* can be assumed to be constant.

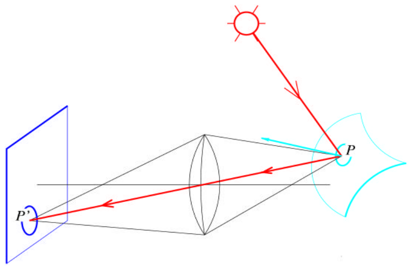
Affine: Orthographic projection



$$\boxed{\begin{cases} x' = x \\ y' = y \end{cases}}$$

When the camera is at a (roughly constant) distance from the scene, take *m*=1.

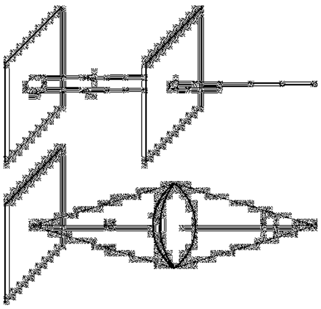
Modeling Lenses



Are we interested in lenses?

- Most of the time, we can (and will) ignore lenses. When one wants to improve precision (e.g., in rendering), lens modeling is needed.
- The problem with pinhole cameras is that:
  - To be precise, the pinhole has to be infinitely small. Otherwise the image is blurred.
  - To allow light to reach all image points, the pinhole needs be large.

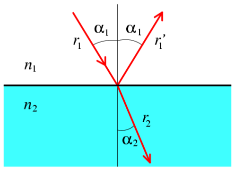
The reason for lenses



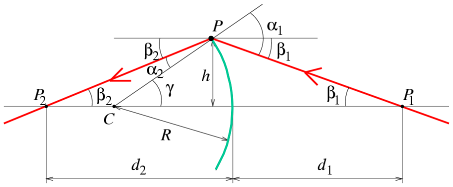
Snell's law

$n_1 \sin \alpha_1 = n_2 \sin \alpha_2$

Descartes' law

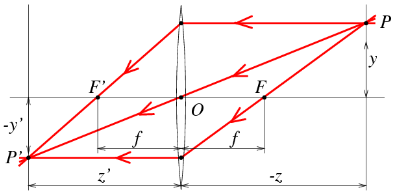


Paraxial (1st order) optics



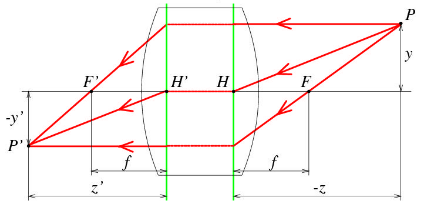
Snell's law:
$n_1 \sin a_1 = n_2 \sin a_2$ ➡ Small angles:
$n_1 a_1$ ¼ $n_2 a_2$ ➡ $\frac{n_1}{d_1} + \frac{n_2}{d_2} = \frac{n_2 - n_1}{R}$

Thin Lenses

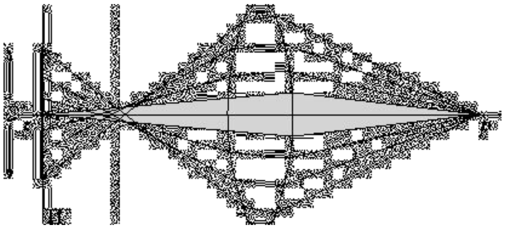


$$\begin{cases} x' = z' \dfrac{x}{z} \\ y' = z' \dfrac{y}{z} \end{cases}, \quad \text{where} \quad \frac{1}{z'} - \frac{1}{z} = \frac{1}{f} \quad \text{and} \quad f = \frac{R}{2(n-1)}.$$

Thick Lenses



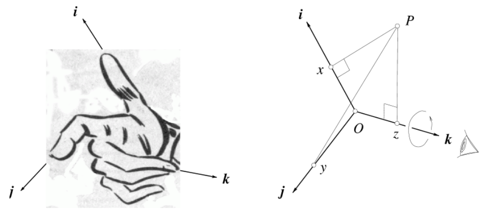
Spherical aberration



Geometric Camera Models

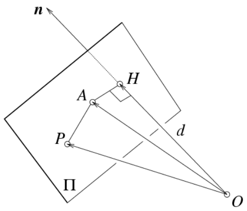
- The pinhole camera model and the three projections introduced so far are very useful in practice.
- However, these need to be define in an appropriate manner to facilitate the use of simple, basic linear algebra operations.
- Otherwise, these would required non-linear computations.

Euclidean Coordinate Systems



$$\begin{cases} x = \overrightarrow{OP}.\mathbf{i} \\ y = \overrightarrow{OP}.\mathbf{j} \\ z = \overrightarrow{OP}.\mathbf{k} \end{cases} \Leftrightarrow \quad \overrightarrow{OP} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \quad \Leftrightarrow \quad \mathbf{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Planes



$$\overrightarrow{AP}.\mathbf{n} = 0 \Leftrightarrow ax + by + cz - d = 0 \Leftrightarrow \mathbf{\Pi}.\mathbf{P} = 0$$

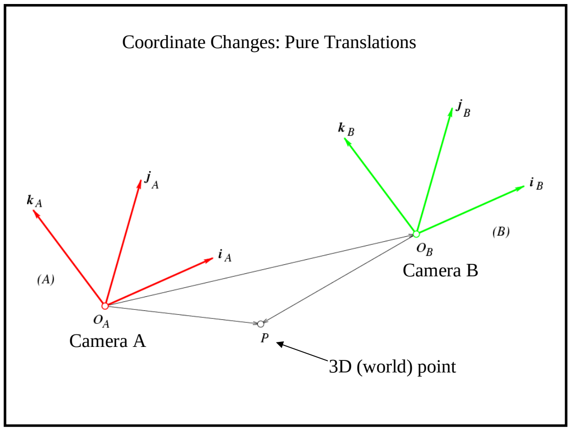
where $\mathbf{\Pi} = \begin{bmatrix} a \\ b \\ c \\ -d \end{bmatrix}$ and $\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ homogeneous coordinates

HOMOGENEOUS COORDINATES

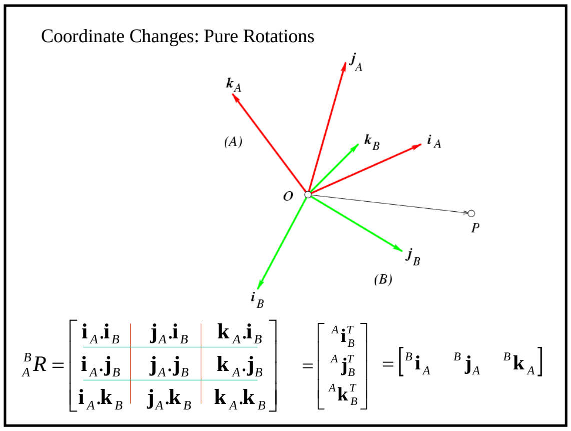
- This way of defining vectors is *key* to much of what we will do in the first part of the course.
- This is called homogenous coordinates.
- A vector **P**=$(x,y,z)^T$ can be written in homogenoeous coordinates by simply adding a "1" at the end, i.e., **P**=$(x,y,z,1)^T$.
- To go back to non-homogeneous, simply remove the last component of the vector as follows **P**=$(x,y,z,d)^T$ ➡ **P**=$(x/d,y/d,z/d)^T$.

Coordinate System Change

- Moving the camera from one location to another can be interpreted as a simple translation and rotation of the 3D coordinate system **O**.
- This can also be used when we have more than one camera.
- If we have two cameras, A and B, we can write $\mathbf{O}_\mathrm{A}$ and $\mathbf{O}_\mathrm{B}$.

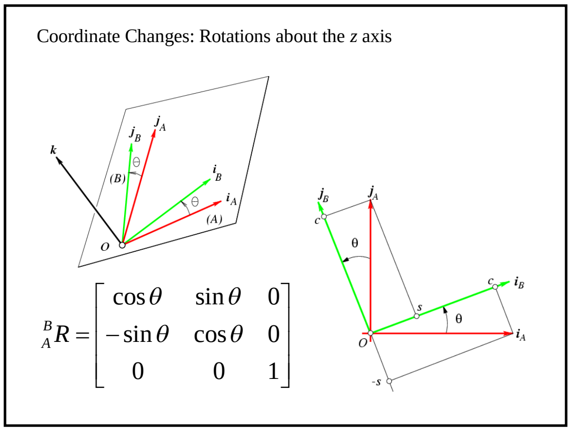


Coordinate Changes: Pure Translations



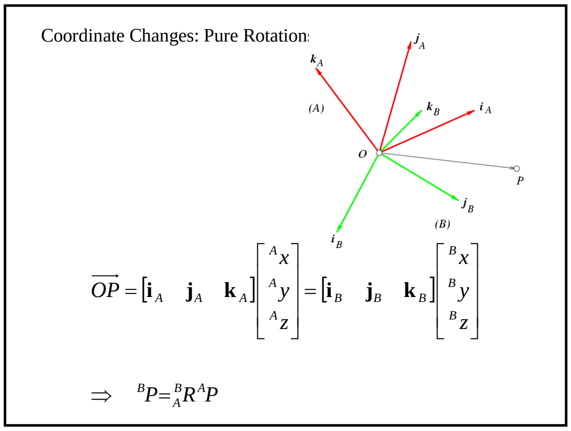
Coordinate Changes: Pure Rotations

$$ {}^B_AR = \begin{bmatrix} \mathbf{i}_A.\mathbf{i}_B & \mathbf{j}_A.\mathbf{i}_B & \mathbf{k}_A.\mathbf{i}_B \\ \mathbf{i}_A.\mathbf{j}_B & \mathbf{j}_A.\mathbf{j}_B & \mathbf{k}_A.\mathbf{j}_B \\ \mathbf{i}_A.\mathbf{k}_B & \mathbf{j}_A.\mathbf{k}_B & \mathbf{k}_A.\mathbf{k}_B \end{bmatrix} = \begin{bmatrix} {}^A\mathbf{i}_B^T \\ {}^A\mathbf{j}_B^T \\ {}^A\mathbf{k}_B^T \end{bmatrix} = \begin{bmatrix} {}^B\mathbf{i}_A & {}^B\mathbf{j}_A & {}^B\mathbf{k}_A \end{bmatrix} $$



Coordinate Changes: Rotations about the *z* axis

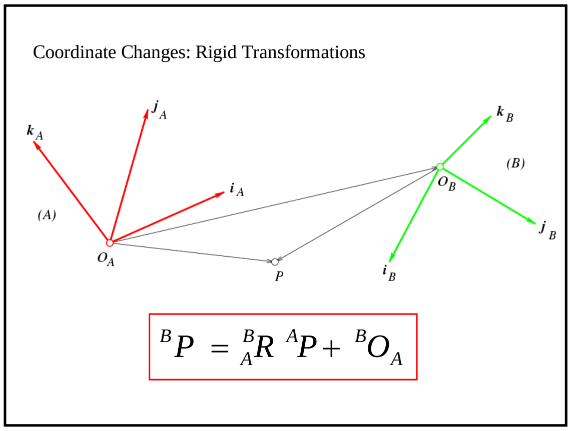
$$ {}^B_AR = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} $$



Coordinate Changes: Pure Rotation

$$ \overrightarrow{OP} = \begin{bmatrix} \mathbf{i}_A & \mathbf{j}_A & \mathbf{k}_A \end{bmatrix} \begin{bmatrix} {}^Ax \\ {}^Ay \\ {}^Az \end{bmatrix} = \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix} \begin{bmatrix} {}^Bx \\ {}^By \\ {}^Bz \end{bmatrix} $$

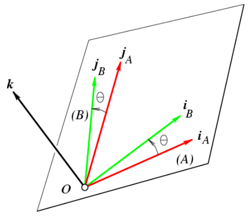
$$ \Rightarrow \quad {}^BP = {}^B_AR\,{}^AP $$



Coordinate Changes: Rigid Transformations

$$ \boxed{{}^BP = {}^B_AR\;{}^AP + {}^BO_A} $$

Rigid Transformations as Mappings: Rotation about the **k** Axis



$$^{F}P' = \mathcal{R}\,^{F}P, \quad \text{where} \quad \mathcal{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Camera Model

- We can also use the formulation we have seen thus far to define the parameters of a camera.
- There are two types of parameters:
  - *Intrinsic*: relate to the camera’s coordinate system to an “idealized” coordinate system.
  - *Extrinsic*: relates the camera’s coordinate system to a fixed world coordinate system.

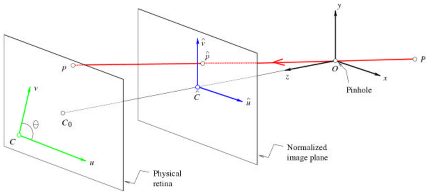
The Intrinsic Parameters of a Camera

Units:

k,*l* : pixel/m

f : m

α,*β* : pixel



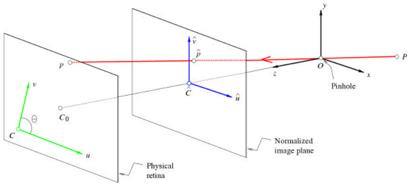
$$\begin{cases} \hat{u} = \dfrac{x}{z} \\ \hat{v} = \dfrac{y}{z} \end{cases} \Longleftrightarrow \hat{\boldsymbol{p}} = \frac{1}{z}\,(\,\mathrm{Id} \quad \boldsymbol{0}\,)\begin{pmatrix} \boldsymbol{P} \\ 1 \end{pmatrix}$$

Physical Image Coordinates

Normalized Image Coordinates

$$\begin{cases} u = kf\dfrac{x}{z} \\ v = lf\dfrac{y}{z} \end{cases} \rightarrow \begin{cases} u = \alpha\dfrac{x}{z} + u_0 \\ v = \beta\dfrac{y}{z} + v_0 \end{cases} \rightarrow \begin{cases} u = \alpha\dfrac{x}{z} - \alpha\cot\theta\dfrac{y}{z} + u_0 \\ v = \dfrac{\beta}{\sin\theta}\dfrac{y}{z} + v_0 \end{cases}$$

The Intrinsic Parameters of a Camera



Calibration Matrix

$$\boldsymbol{p} = \mathcal{K}\hat{\boldsymbol{p}}, \quad \text{where} \quad \boldsymbol{p} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad \text{and} \quad \mathcal{K} \overset{\text{def}}{=} \begin{pmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

The Perspective Projection Equation

$$\boldsymbol{p} = \frac{1}{z}\mathcal{M}\boldsymbol{P}, \quad \text{where} \quad \mathcal{M} \overset{\text{def}}{=} (\,\mathcal{K} \quad \boldsymbol{0}\,)$$

Extrinsic Parameters

- When the camera frame (C) is different from the world frame (W),

$$\begin{pmatrix} ^{C}P \\ 1 \end{pmatrix} = \begin{pmatrix} ^{C}_{W}\mathcal{R} & ^{C}O_W \\ \boldsymbol{0}^T & 1 \end{pmatrix} \begin{pmatrix} ^{W}P \\ 1 \end{pmatrix}.$$

- Thus,

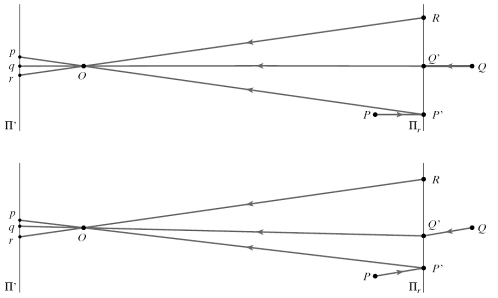
$$\boxed{\boldsymbol{p} = \frac{1}{z}\mathcal{M}\boldsymbol{P},} \quad \text{where} \quad \begin{cases} \mathcal{M} = \mathcal{K}\,(\,\mathcal{R} \quad \boldsymbol{t}\,)\,, \\ \mathcal{R} = ^{C}_{W}\mathcal{R}, \\ \boldsymbol{t} = ^{C}O_W, \\ \boldsymbol{P} = \begin{pmatrix} ^{W}P \\ 1 \end{pmatrix}. \end{cases}$$

- Note: z is *not* independent of $\mathcal{M}$ and $\boldsymbol{P}$:

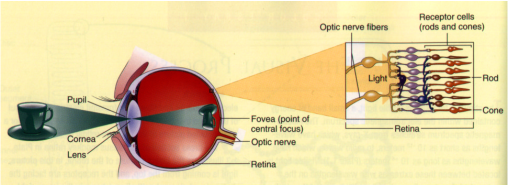
$$\mathcal{M} = \begin{pmatrix} \boldsymbol{m}_1^T \\ \boldsymbol{m}_2^T \\ \boldsymbol{m}_3^T \end{pmatrix} \Longrightarrow z = \boldsymbol{m}_3 \cdot \boldsymbol{P}, \quad \text{or} \quad \begin{cases} u = \dfrac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}, \\ v = \dfrac{\boldsymbol{m}_2 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}. \end{cases}$$

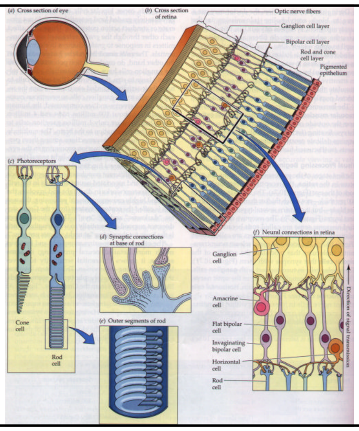
Models to remember:

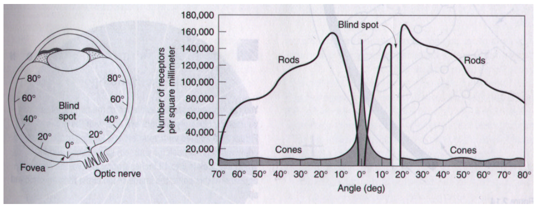
1. Perspective projection.
2. Weak-perspective.
3. Parallel projection.
4. Orthographic (orthogonal).

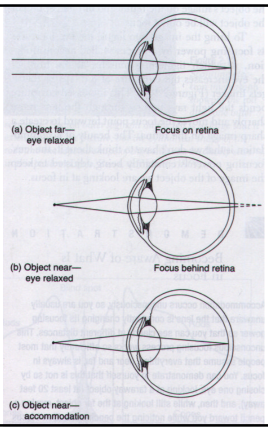


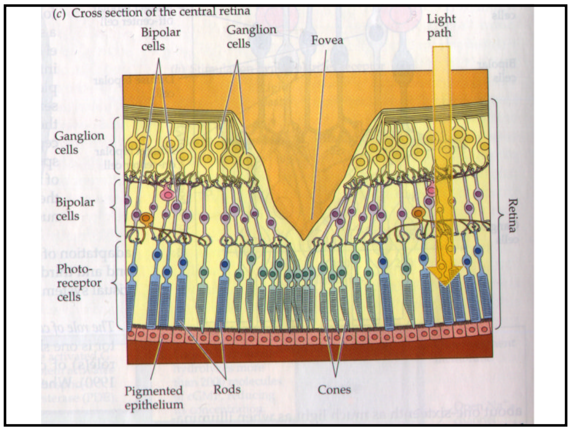
The Human Eye








(a) Object far—eye relaxed
Focus on retina
(b) Object near—eye relaxed
Focus behind retina
(c) Object near—accommodation



Diopters

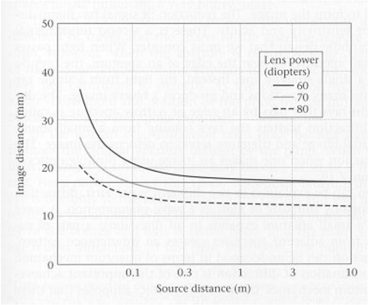


Image distance (mm)
50
40
30
20
10
0
0.1
0.3
1
3
10
Source distance (m)
Lens power
(diopters)
60
70
80