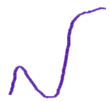
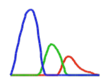
# Image Formation (Spectral)

$$(\mathbf{R},\mathbf{G},\mathbf{B}) = \int_{380}^{780}$$  $*$  $d\lambda$

On the next page, (R,G,B) is the row vector, $\boldsymbol{\rho}$, with elements $\rho^{(k)}$. So, R=$\rho^{(1)}$, G=$\rho^{(2)}$, B=$\rho^{(3)}$.

More formally,

The response of an image capture system to a light signal $L(\lambda)$ associated with a given pixels is modeled by

$$\upsilon^{(k)} = F^{(k)}(\rho^{(k)}) = F^{(k)}\left(\int L(\lambda)R^{(k)}(\lambda)d\lambda\right)$$

where $R^{(k)}(\lambda)$ is the sensor response function for the k$^{th}$ channel and $\upsilon^{(k)}$ is the k$^{th}$ channel response.

In this formulation, $R^{(k)}(\lambda)$ includes the contributions due to the aperture, focal length, sensor position in the focal plane.

$F^{(k)}$ absorbs typical non-linearities such as gamma.

Important

# Discrete Version

Often we represent functions by vectors. For example, a spectra might be represented by 101 samples in the range of 380 to 780 nm in steps of 4nm.

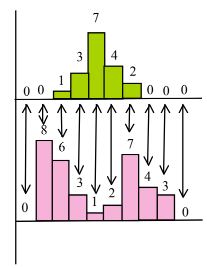
Then $L(\lambda)$ becomes the vector **L**, $R^{(k)}(\lambda)$ becomes the vector $\mathbf{R^k}$, and the response (ignoring non-linearity issues) is given by a dot product:

$$\rho^{(k)} = \mathbf{L} \bullet \mathbf{R^{(k)}}$$

Important

# Sensor/light interaction example

**R**=(0,0,1,3,7,4,2,0,0,0)



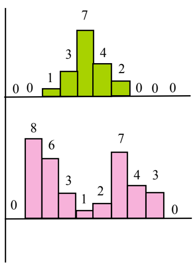

**L**=(0,8,6,3,1,2,7,4,3,0)

Multiply lined up pairs of numbers and then sum up

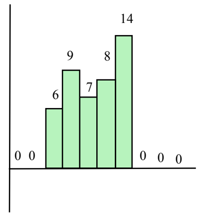
Important

## Sensor/light interaction example

**R**=(0,0,1,3,7,4,2,0,0,0)




**L**=(0,8,6,3,1,2,7,4,3,0)




**L .* R**=
(0*0, 0*8, 1*6, 3*3, 7*1, 4*2, 2*7, 0*4, 0*3, 0*0)
=(0,0,6,9,7,8,14,0,0,0)

**L • P** = 0 + 0 + 6 + 9 + 7 + 8 + 14
= 44

## Image Formation (Spectral)

- Note that (ignoring $F^{(k)}$) image formation is linear.
- Formally this means **if**:

$$L_1(\lambda) -> \rho_1^{(k)} \quad \text{and} \quad L_2(\lambda) -> \rho_2^{(k)}$$

- **Then**:

?

## Image Formation (Spectral)

- Note that (ignoring $F^{(k)}$) image formation is linear.
- Formally this means **if**:

$$L_1(\lambda) \;->\; \rho_1^{(k)} \quad \text{and} \quad L_2(\lambda) \;->\; \rho_2^{(k)}$$

- **Then**:

$$aL_1(\lambda) + bL_2(\lambda) \;->\; a\rho_1^{(k)} + b\rho_2^{(k)}$$

## Image Formation (Spectral)

- Note that image formation loses spectral information
- Technically, the process is a projection
- This means that two **very** different spectra can map into the same color
- This is the key to color reproduction

Supplemental material

## Image Formation (Spectral)

$F^{(k)}$ is often ignored (assumed to be the identity), but this is not a safe assumption, especially when color or radiometric measurements matter. To compensate for the non-linearity of CRT display monitors, camera manufactures will "gamma" correct the signal, typically by raising the signal (normalized to [0,1]) to the power 1/(2.2).

Note that in such an image, a number twice as large does not mean that the light had twice the power!

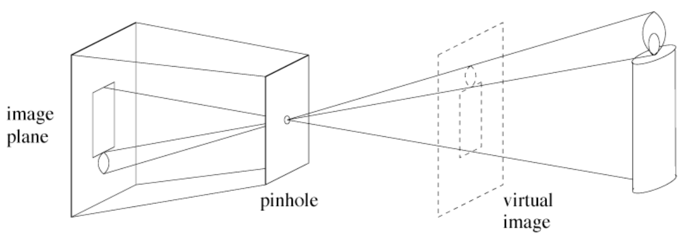
To linearize RGB's from such a signal we compute:

$p=F^{-1}(v)=255*(v/255)^{\wedge}2.2$

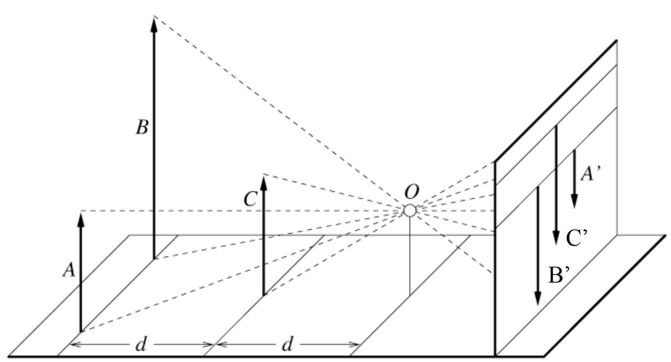
## Image Formation (Geometric)

## Pinhole cameras

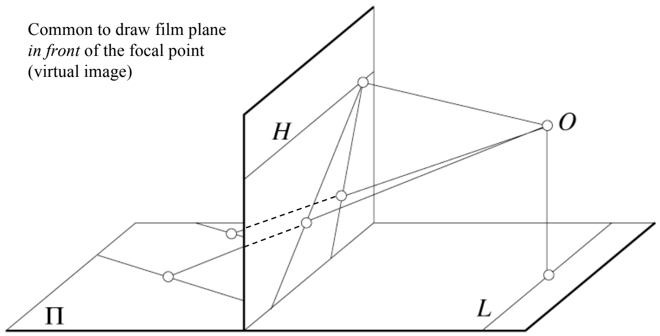
- Abstract camera model--box with a small hole in it
- Pinhole cameras work for deriving algorithms--a real camera needs a lens




## Distant objects are smaller



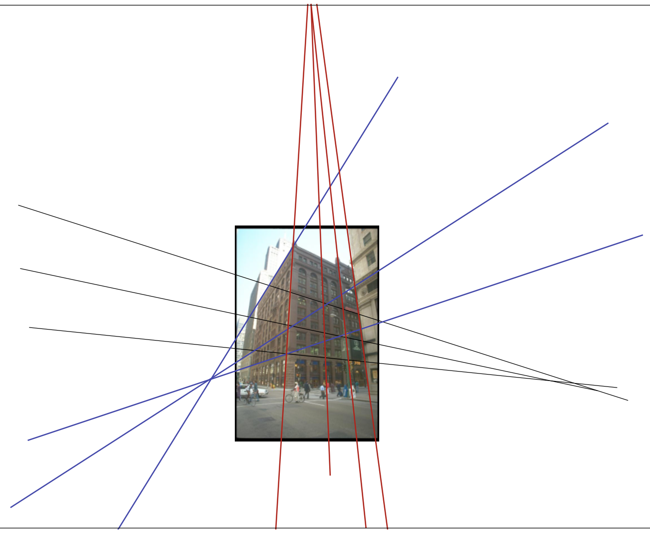
## Parallel lines meet*

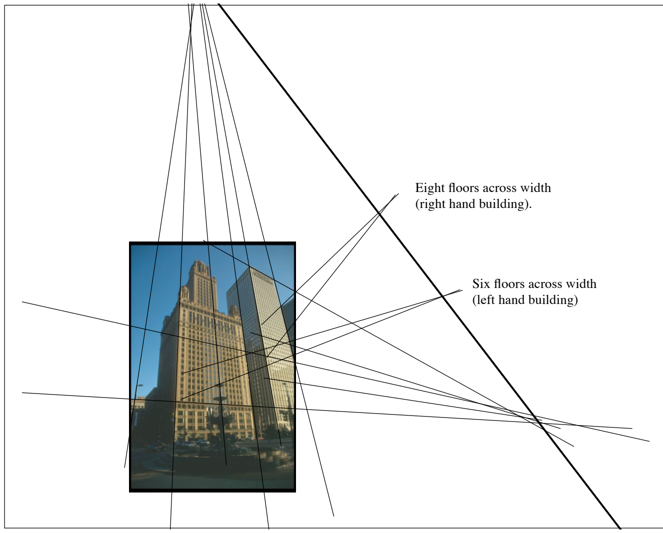



*Exceptions?

## 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
  - Standard horizon is the horizon of the ground plane.
- One way to spot fake images
  - scale and perspective don't work
  - vanishing points behave badly





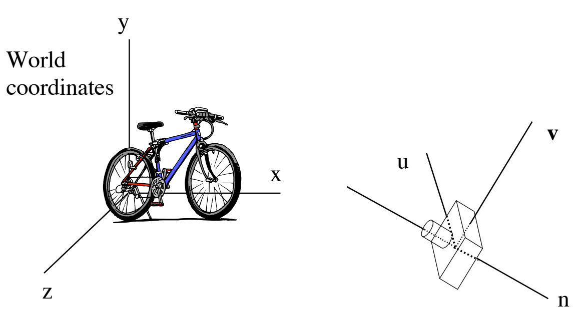
# Geometric properties of projection

- Points go to points
- Lines go to lines
- Polygons go to polygons
- Degenerate cases
  - line through focal point projects to a point
  - plane through focal point projects to a line

# Geometric Camera Model

- Transform world coordinates to standard camera coordinates
  - (Extrinsic parameters)
- Project onto standard camera plane
  - (3D becomes 2D)
- Transform into pixel locations
  - (Intrinsic camera parameters)

# World and camera coordinates




Math aside, #2

# Representing Transformations

- Need mathematical representation for mapping points from the world to an image (and later, from an image taken by one camera to another).
- Represent linear transformations by matrices
- To transform a point, represented by a vector, multiply the vector by the appropriate matrix.
- To transform line segments, transform endpoints
- To transform polygons, transform vertices

## 2D Transformations

- Represent **linear** transformations by matrices
- To transform a point, represented by a vector, multiply the vector by the appropriate matrix.
- Recall the definition of matrix times vector:

$$\begin{pmatrix} a_{11}x + a_{12}y \\ a_{21}x + a_{22}y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

## Matrix multiplication is linear

- In particular, if we define f(**x**)=M • **x**, where M is a matrix and **x** is a vector, then

$$\begin{aligned} f(a\mathbf{x} + b\mathbf{y}) &= M(a\mathbf{x} + b\mathbf{y}) \\ &= aM\mathbf{x} + bM\mathbf{y} \\ &= af(\mathbf{x}) + bf(\mathbf{y}) \end{aligned}$$

- Where the middle step can be verified using algebra (next slide)

Supplemental material

## Proof that matrix multiplication is linear

$$\begin{aligned} M(a\mathbf{x} + b\mathbf{y}) &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} ax_1 + by_1 \\ ax_2 + by_2 \end{pmatrix} \\ &= \begin{pmatrix} a_{11}ax_1 + a_{11}by_1 + a_{12}ax_2 + a_{12}by_2 \\ a_{21}ax_1 + a_{21}by_1 + a_{22}ax_2 + a_{22}by_2 \end{pmatrix} \\ &= \begin{pmatrix} a_{11}ax_1 + a_{12}ax_2 + a_{11}by_1 + a_{12}by_2 \\ a_{21}ax_1 + a_{22}ax_2 + a_{21}by_1 + a_{22}by_2 \end{pmatrix} \\ &= a\begin{pmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{pmatrix} + b\begin{pmatrix} a_{11}y_1 + a_{12}y_2 \\ a_{21}y_1 + a_{22}y_2 \end{pmatrix} \\ &= aM\mathbf{x} + bM\mathbf{y} \end{aligned}$$

## Composition of Transformations

- If we use one matrix, $M_1$ for one transform and another matrix, $M_2$ for a second transform, then the matrix for the first transform followed by the second transform is simply $M_2 M_1$
- This follows from the associativity of matrix multiplication
  - $M_2 (M_1\mathbf{x}) = (M_2 M_1)\mathbf{x}$
- This generalizes to any number of transforms