Computer Vision
cmput 428/615

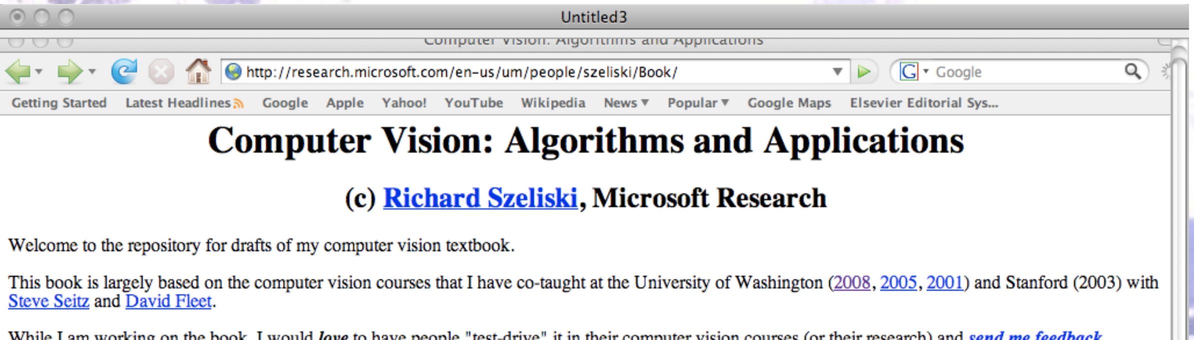
Lecture 2:

Cameras and Images

Martin Jagersand

Readings: HZ Ch 1, 6

Sz 2.3, FP: Ch 1, 3DV: Ch 3



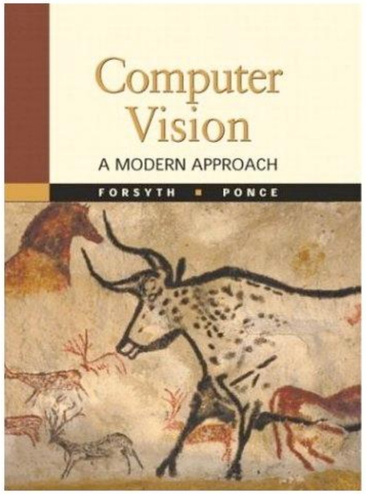
Computer Vision: Algorithms and Applications

(c) Richard Szeliski, Microsoft Research

Welcome to the repository for drafts of my computer vision textbook.

This book is largely based on the computer vision courses that I have co-taught at the University of Washington (2008, 2005, 2001) and Stanford (2003) with Steve Seitz and David Fleet.

While I am working on the book, I would *love* to have people "test-drive" it in their computer vision courses (or their research) and *send me feedback*



Learn the details of each stage

Stages in processing:

1. Physical properties
   - Camera calibration, reflectance models etc.
2. Low level processing
   - Extraction of local features: lines/edges, color, texture
3. Midlevel
   - 3D reconstruction. Regional and interpretation of features
4. High level
   - Task dependent global integration, e.g.
   - Robotics: How to move toward object
   - AI: make inference in scene,
   - Graphics: Render 3D scene model

Now: Learn about cameras and how they form images

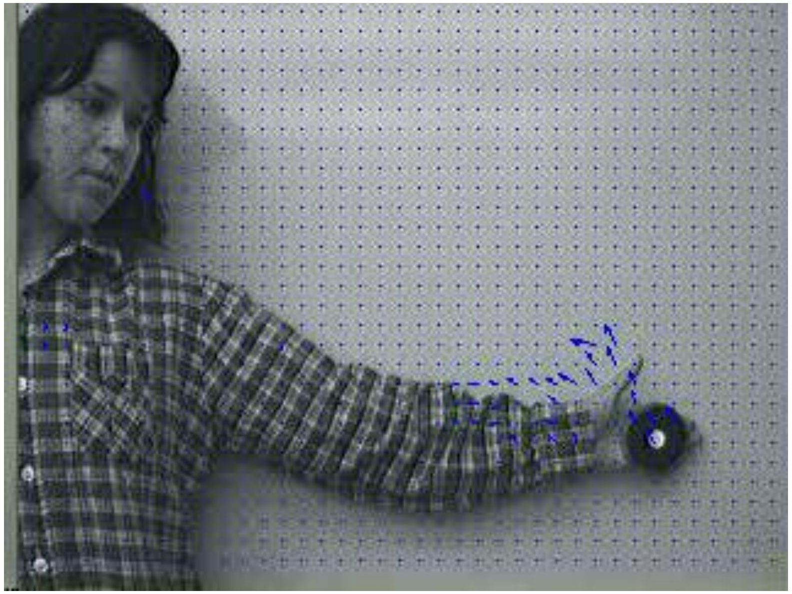
Readings: HC Ch 1, 6, Sz 2.3

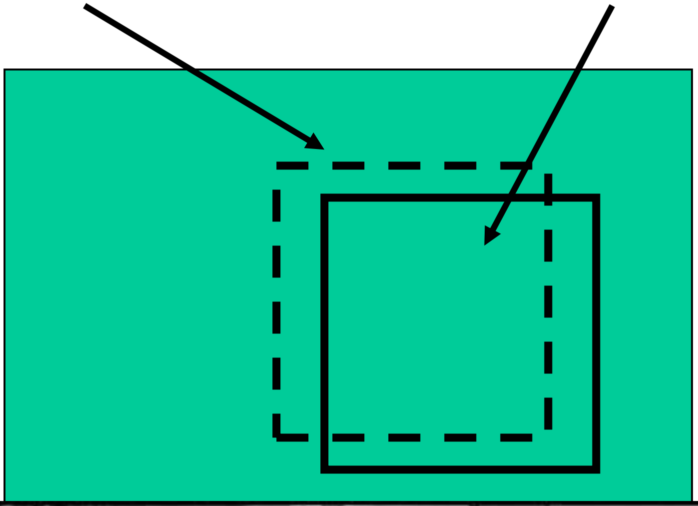
FP: Ch 1, 3DV: Ch 3

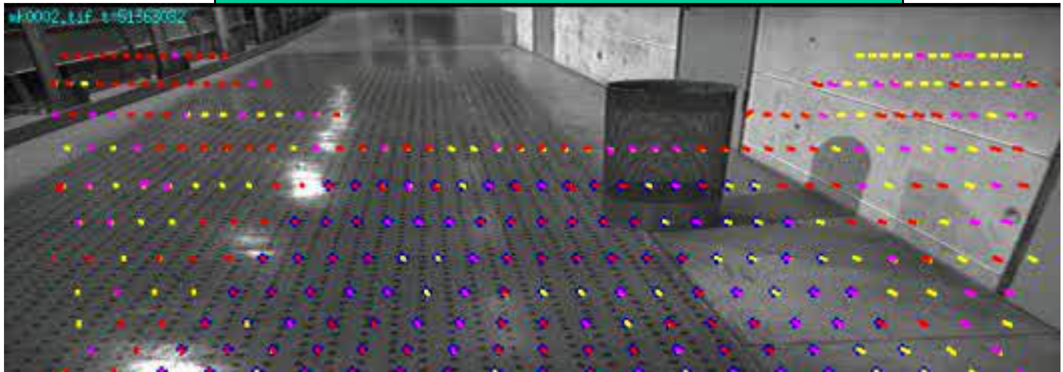
Low level video processing “Visual motion detecton”

Relating two adjacent frames: (small differences):

$$\mathrm{Im}(x+\delta x,\ \ y+\delta y,\ \ t+\delta t)\ \ =\ \ \mathrm{Im}(x,\ \ y,\ \ t)$$

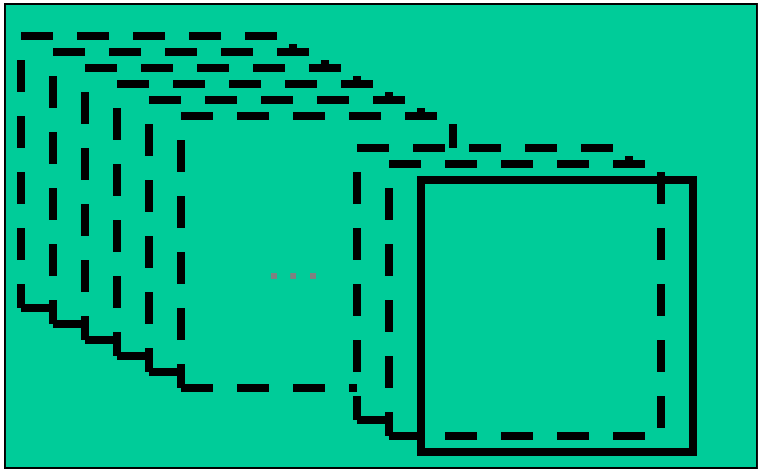


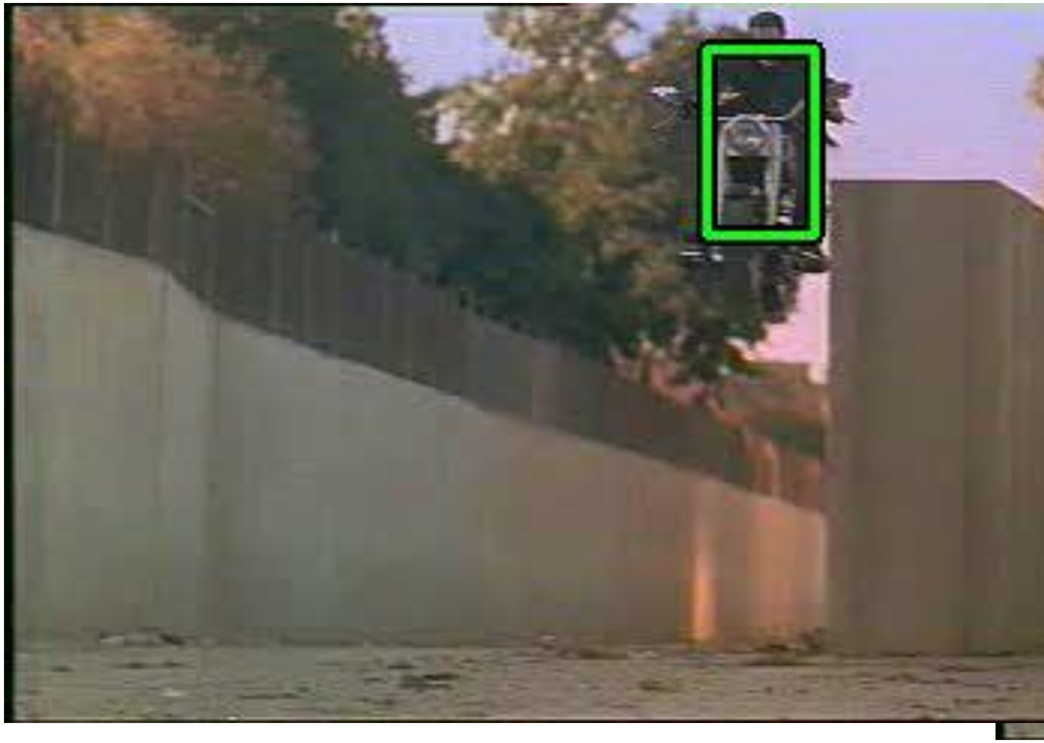




Mid-level video processing
"Video Object Tracking" / Registration

Globally relating all frames: (large differences):





MTF – Modular Tracking Framework

Modular Tracking Framework

A Unified Approach to Registration based Tracking

Abhineet Singh and Martin Jagersand

UNIVERSITY OF ALBERTA

- Open source
- C++ implementation
- ROS interface
- Matlab/Pyhton
- Cross platform

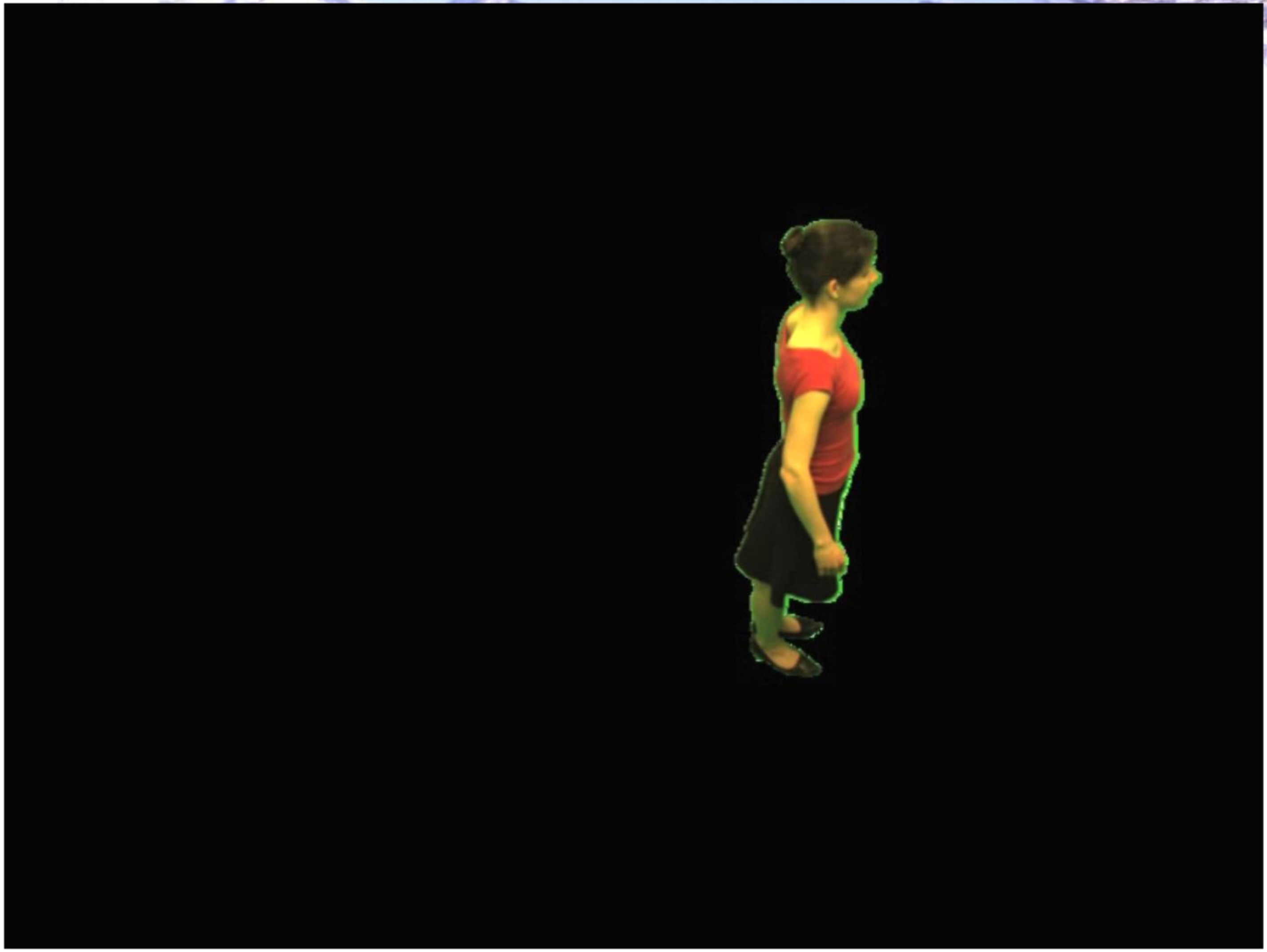
Application:
Augmenter Reality

Vision-guided robot motion using Multi-view geometry

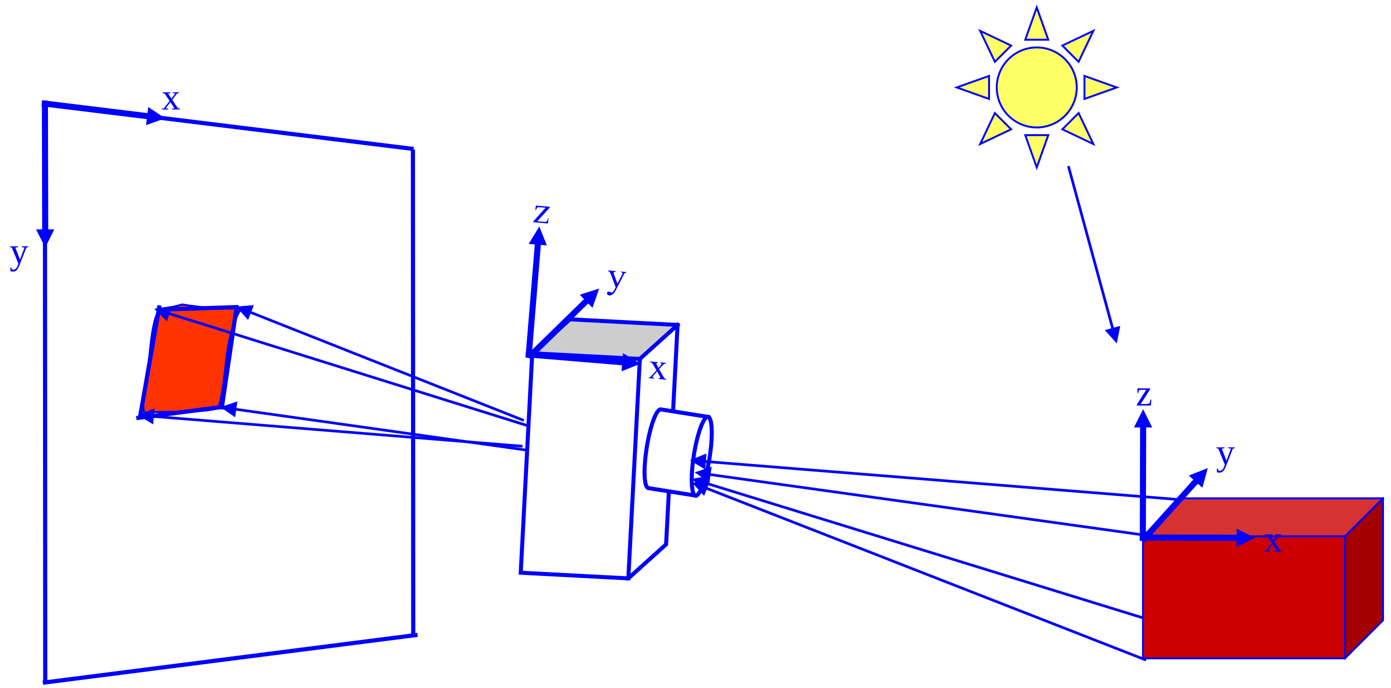
Camera-based 3D capture system

http://webdocs.cs.ualberta.ca/~vis/ibmr/

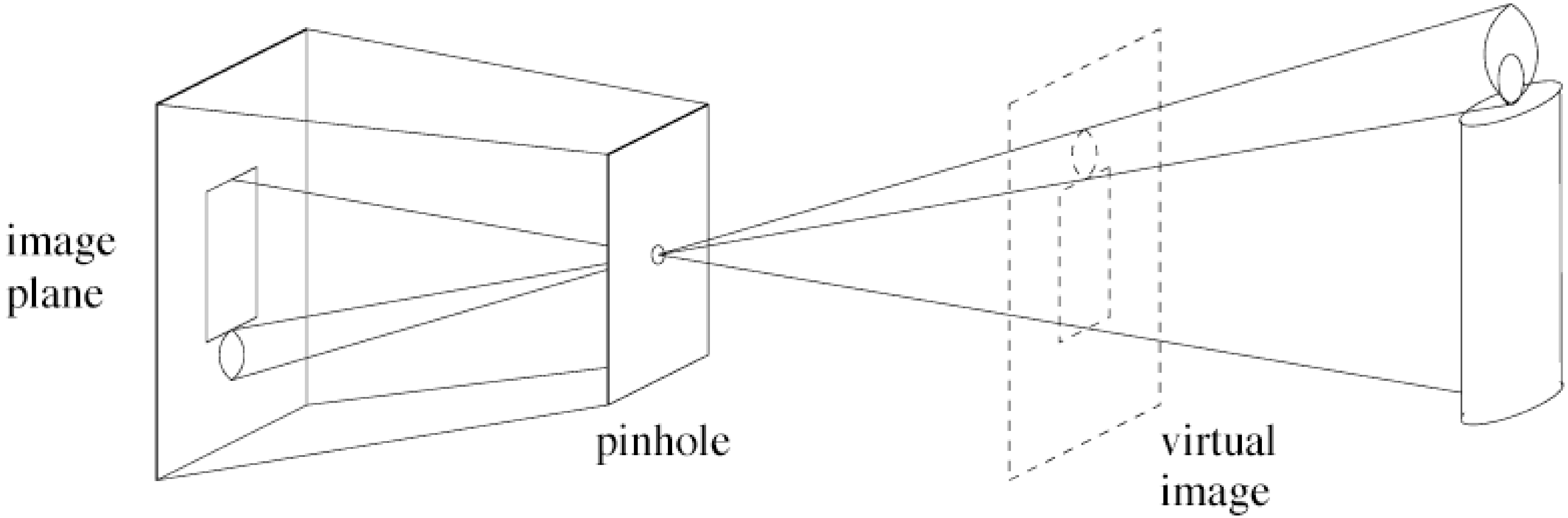
Beyond 3D: Non-rigid and articulated motion



How the 3D physical world is captured on a 2D image plane



Pinhole cameras

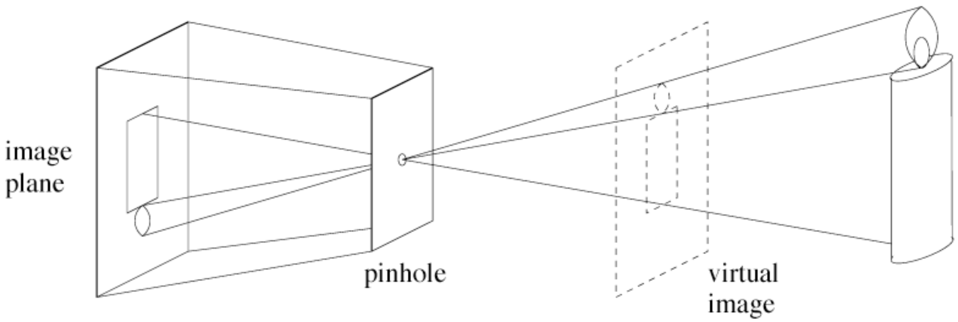


- Abstract camera model - box with a small hole in it
- Image formation described by geometric optics
- Note: equivalent image formation on virtual and real image plane

The equation of projection

How do we develop a consistent mathematical framework for projection calculations?

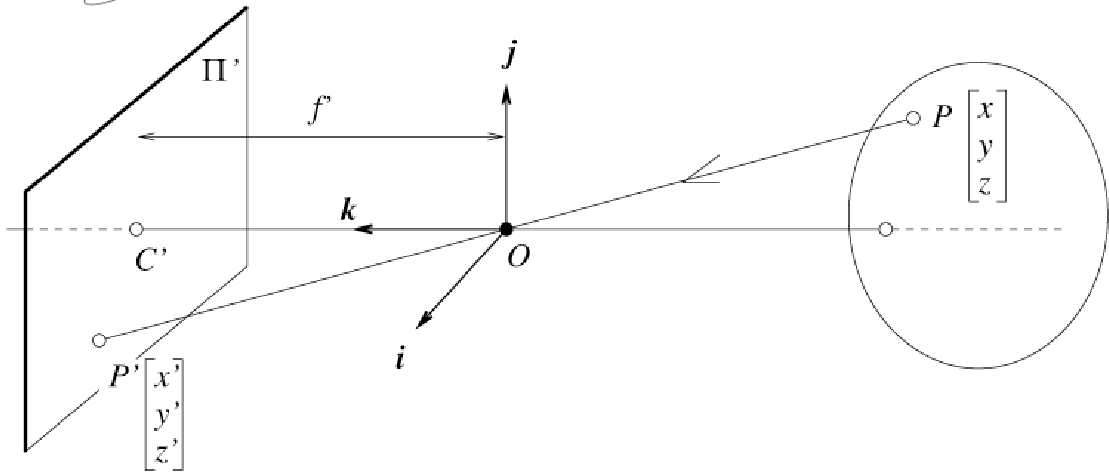
Intuitively:



Mathematically:

- Cartesian coordinates:
- Projectively: x = PX

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



Pinhole cameras: Historic and real

- First photograph due to Niepce,
- First on record shown - 1822
- Basic abstraction is the pinhole camera
  - lenses required to ensure image is not too dark
  - various other abstractions can be applied

Animal Eyes

Land & Nilsson.
Oxford Univ. Press

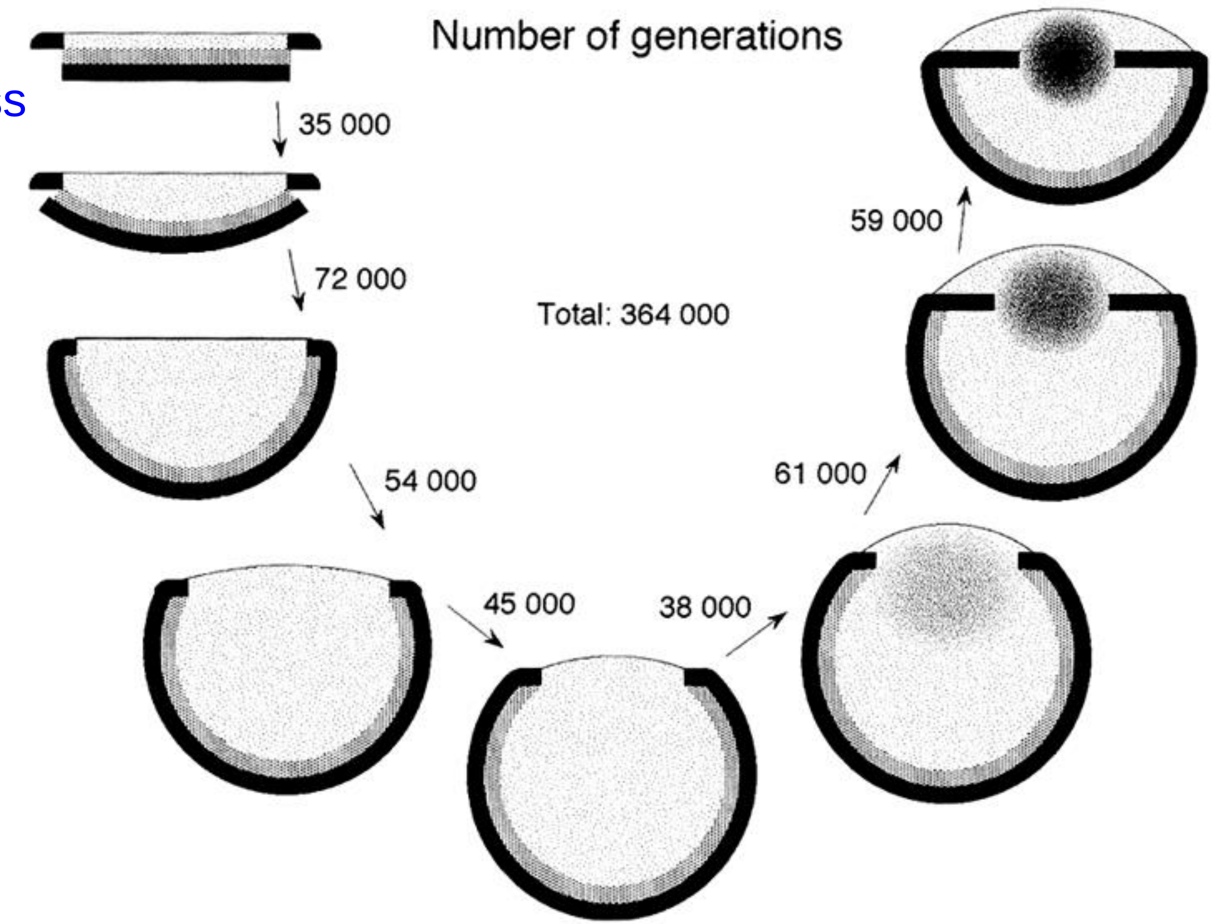


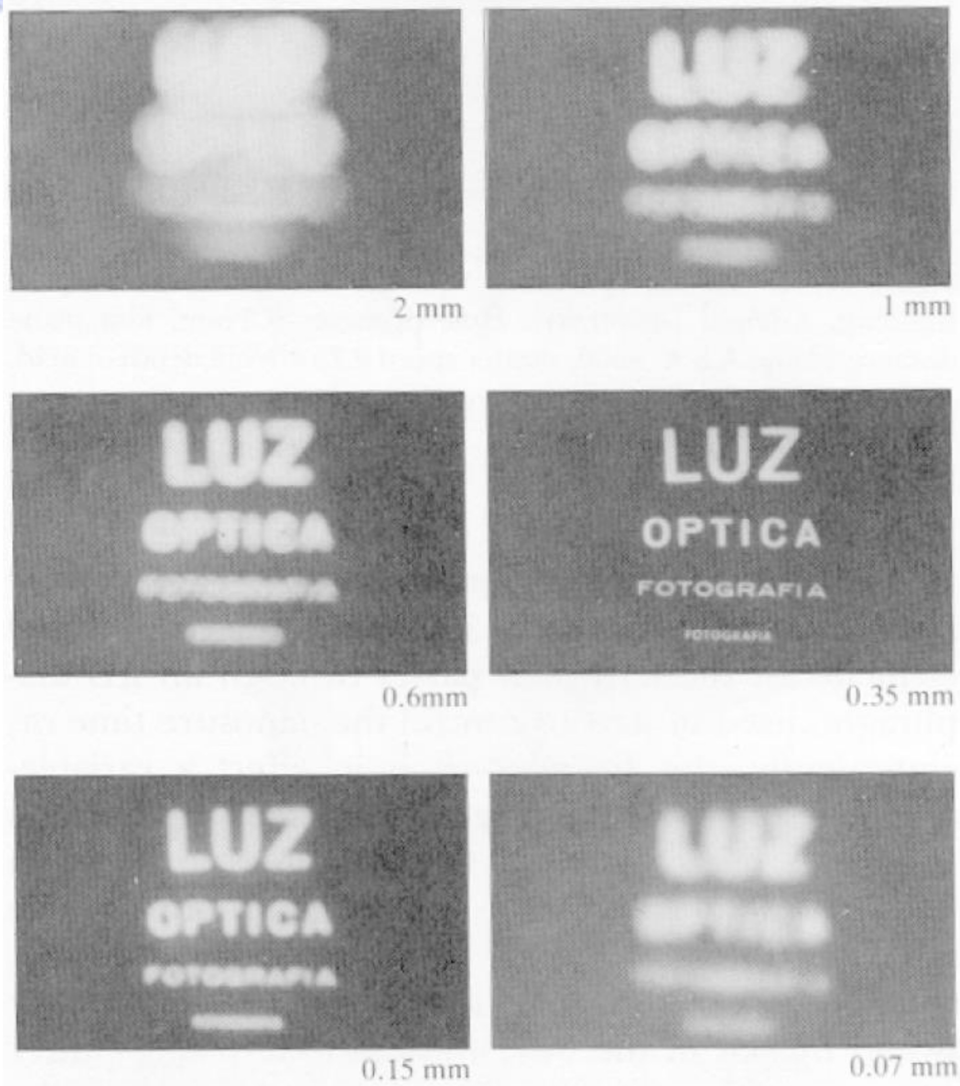
Fig. 1.6 A patch of light sensitive epithelium can be gradually turned into a perfectly focussed camera-type eye if there is a continuous selection for improved spatial vision. A theoretical model based on

Real Pinhole Cameras

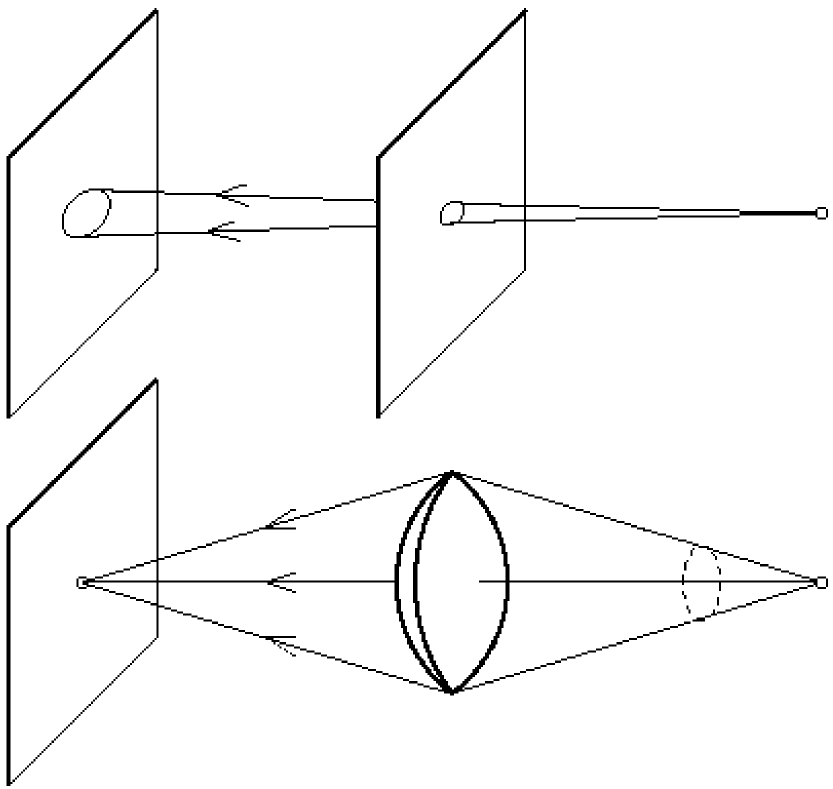
Pinhole too big -
many directions are averaged, blurring the image

Pinhole too small-
diffraction effects blur the image

Generally, pinhole cameras are *dark*, because a very small set of rays from a particular point hits the screen.



Lenses: bring together more rays

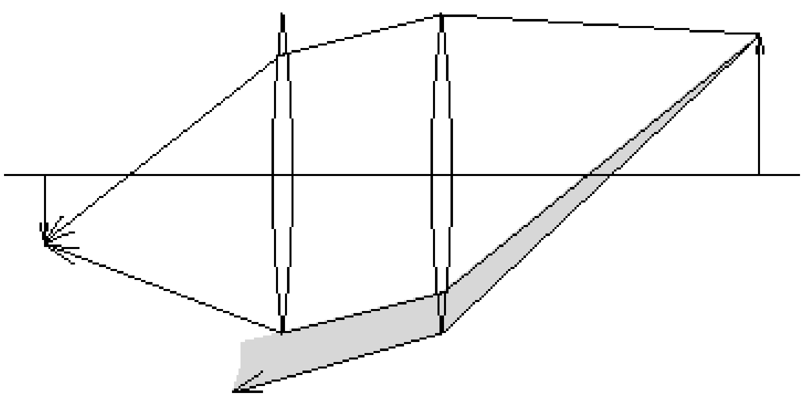


Note: Each world point projects to many image points.

With a 1mm pinhole and f=10mm how many points at 1m distance?

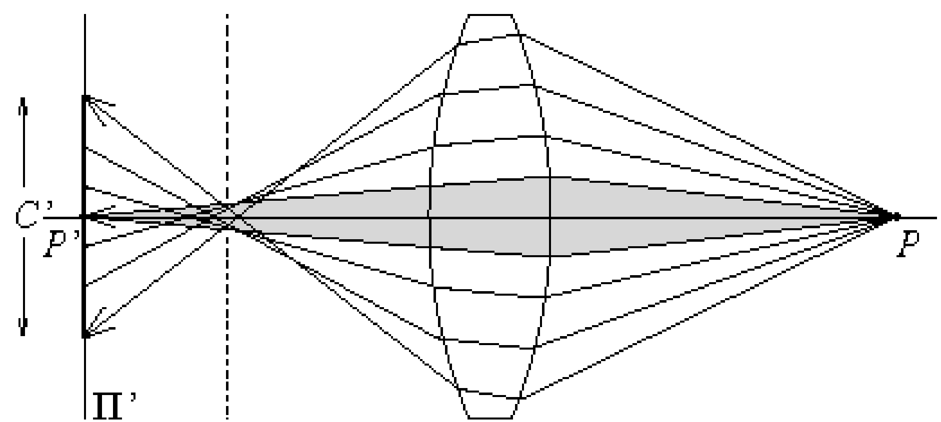
Lens Realities

Real lenses have a finite depth of field, and usually suffer from a variety of defects



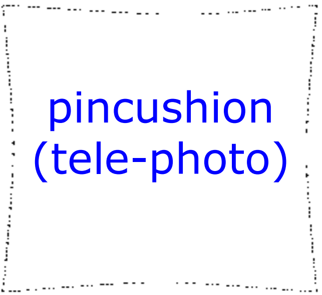
vignetting

Spherical Aberration

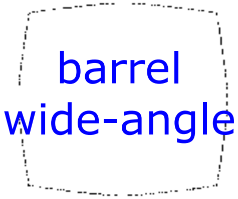


Lens Distortion

magnification/focal length different for different angles of inclination



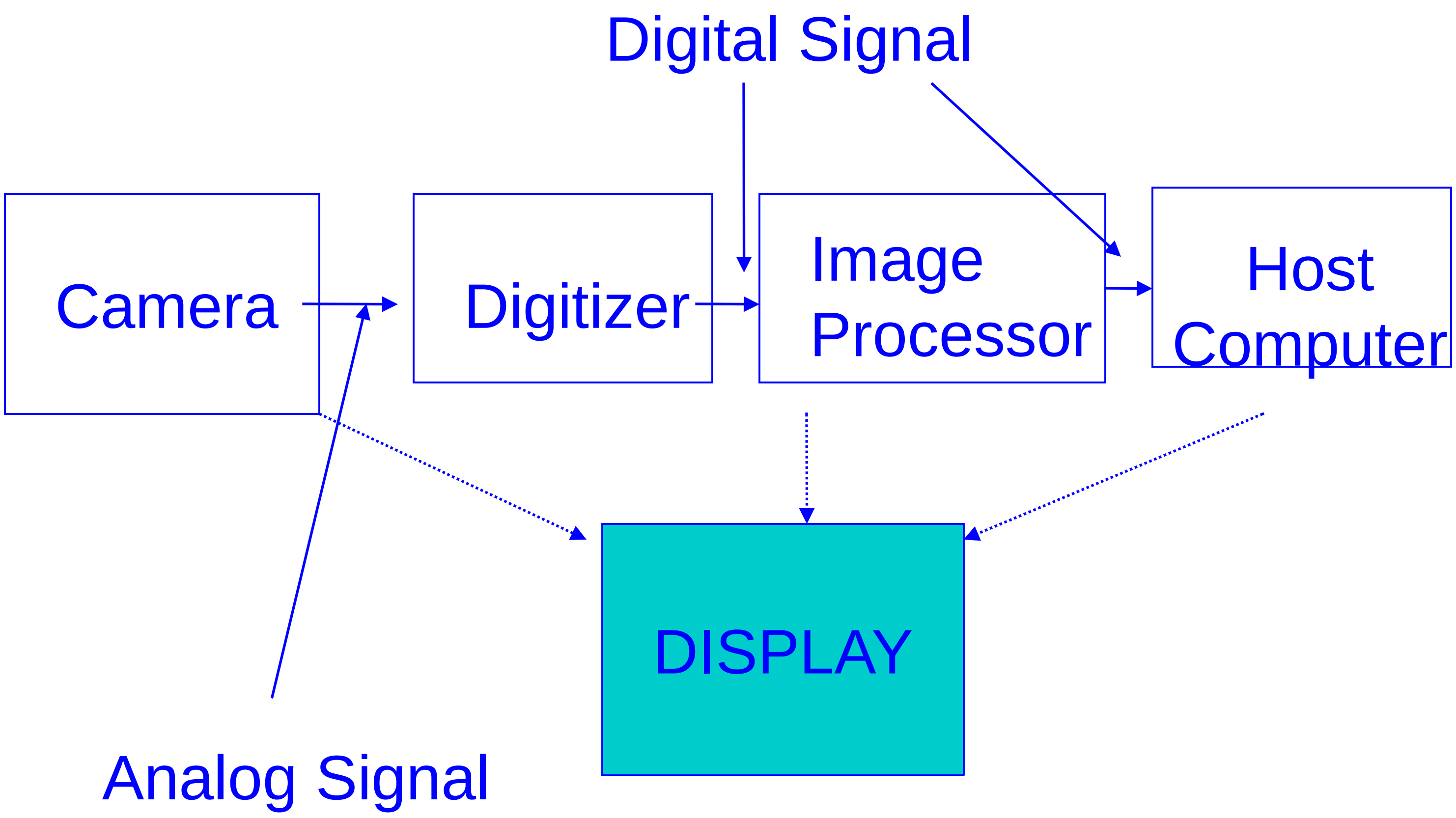
pincushion
(tele-photo)



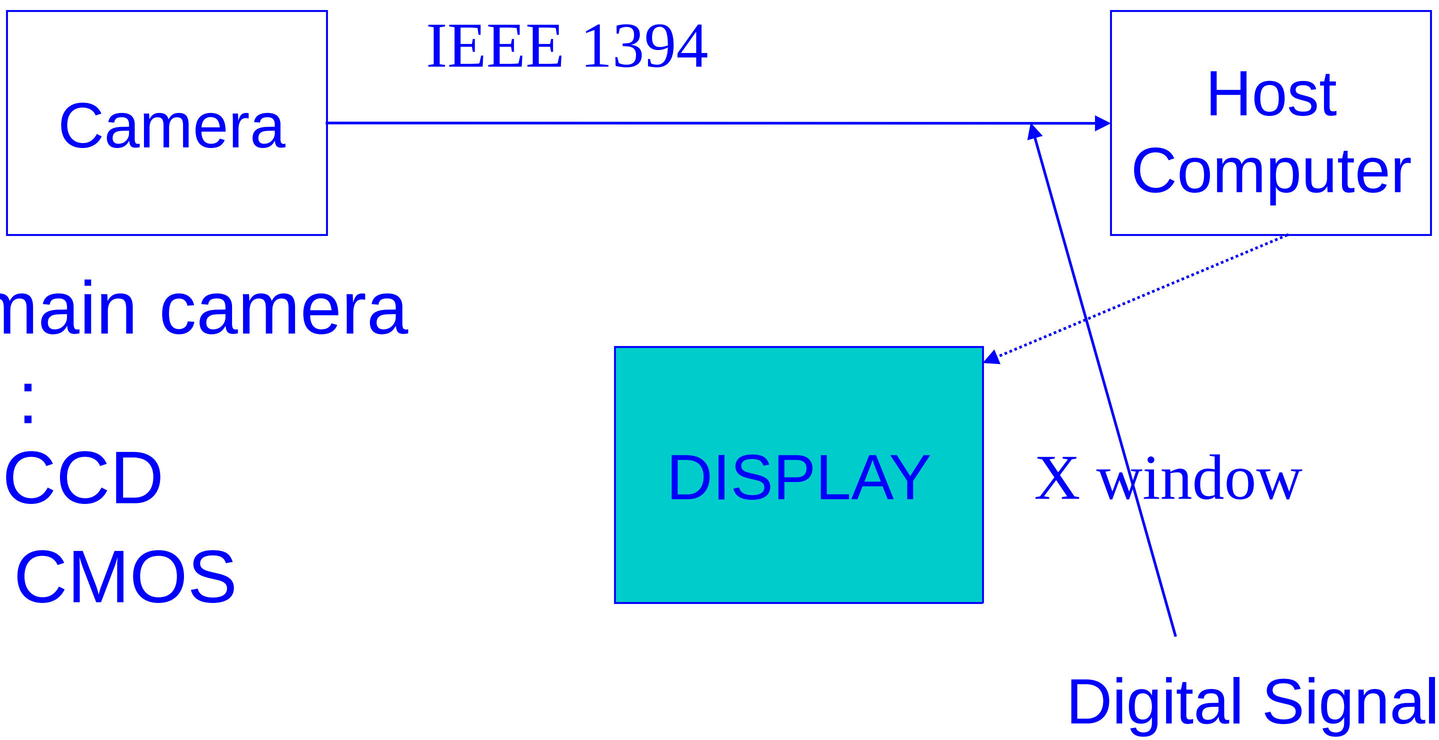
barrel
(wide-angle)

Can be corrected! (if parameters are know)

Image streams -> Computer



A Modern Digital Camera (Firewire)



Two main camera types :

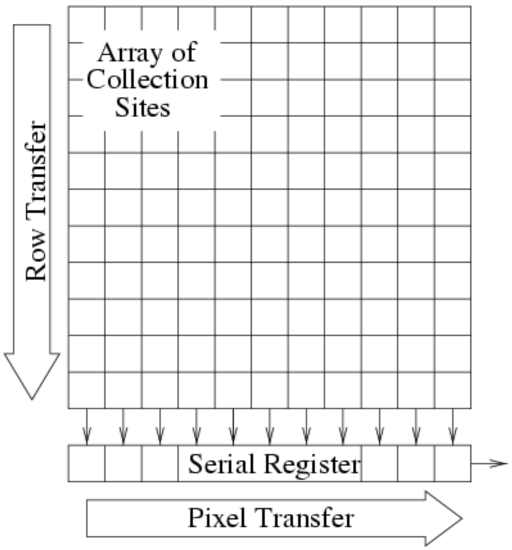
1. CCD
2. CMOS

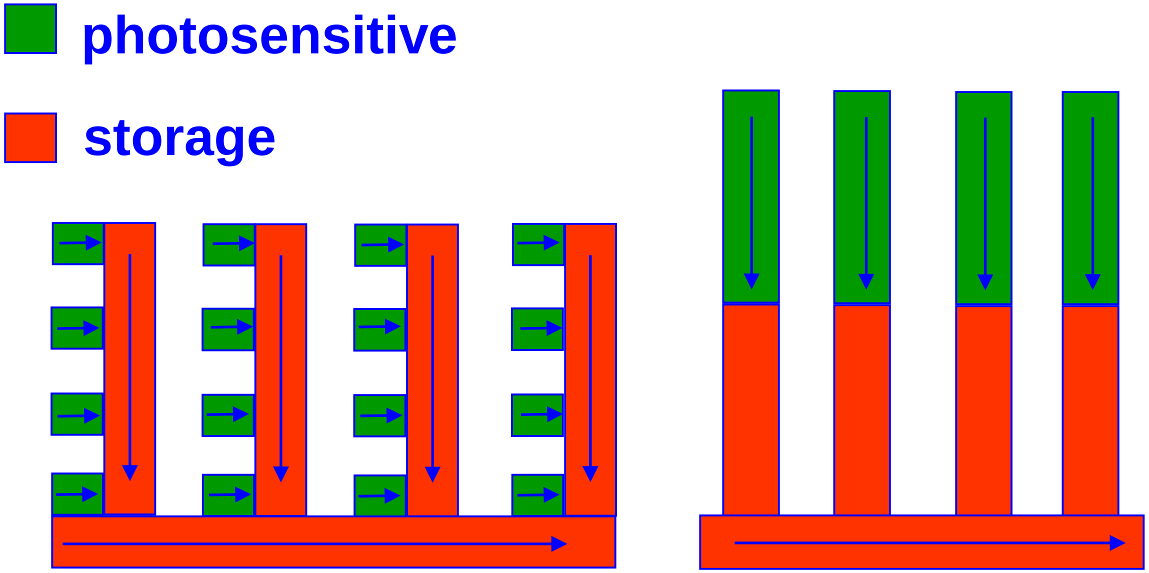
CCD camera

separate photo sensor at regular positions
no scanning

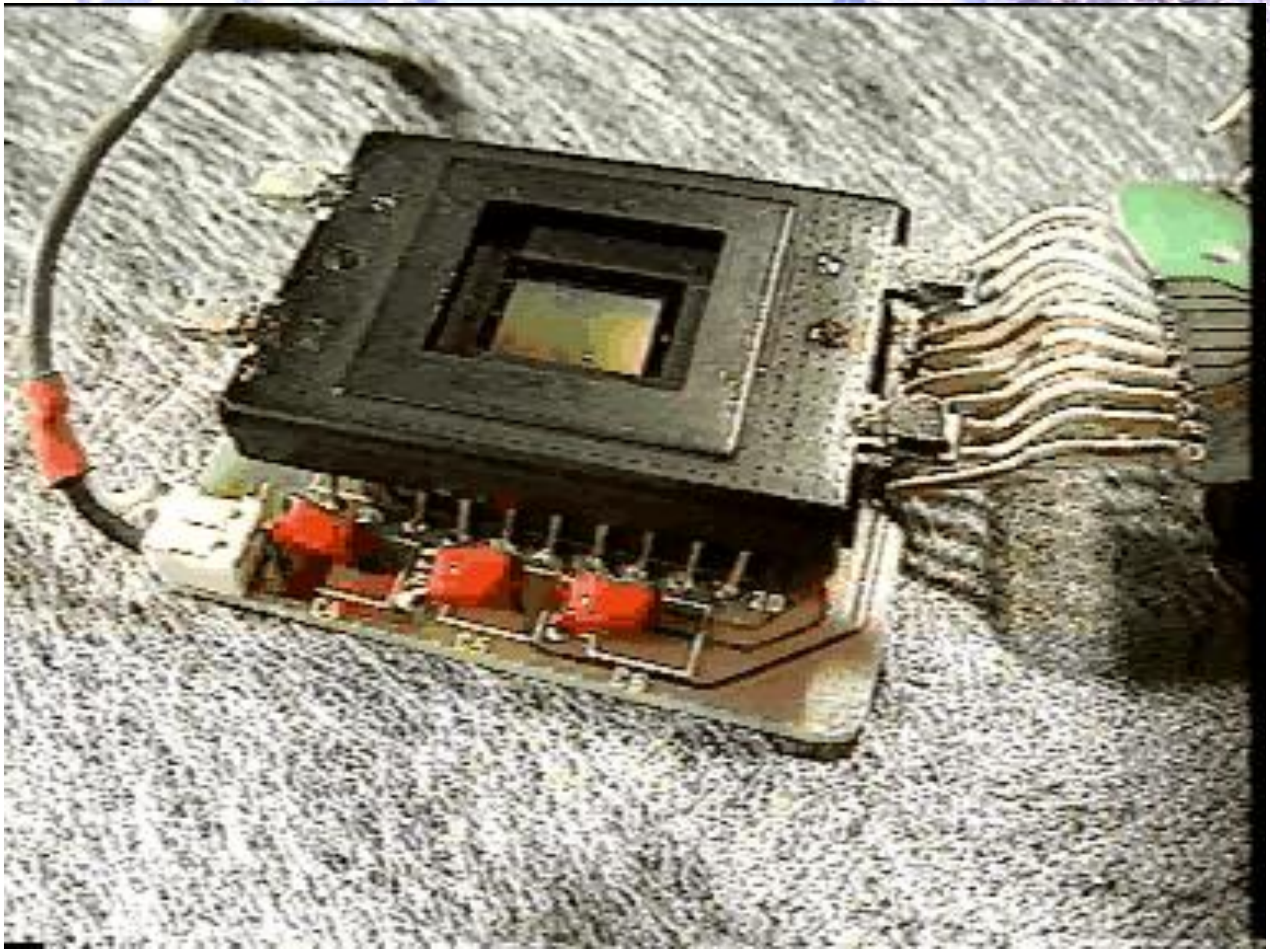
charge-coupled devices (CCDs)

area CCDs and linear CCDs
2 area architectures :
interline transfer and *frame transfer*



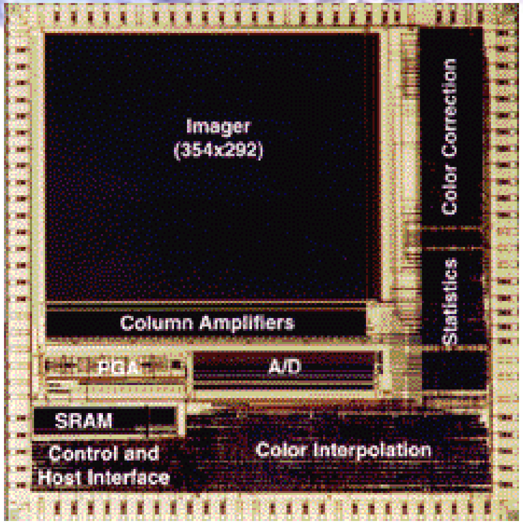


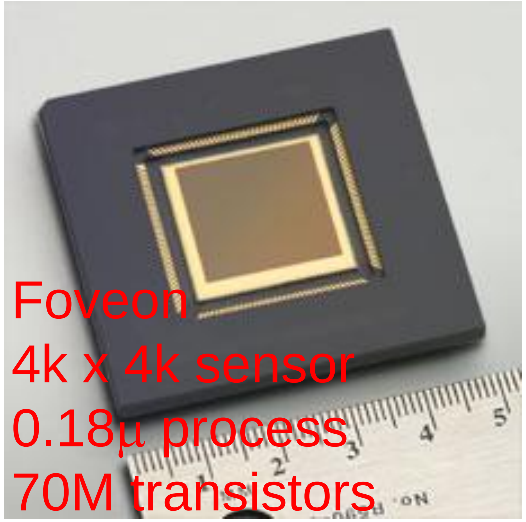
The CCD camera



CMOS

- Same sensor elements as CCD
- Each photo sensor has its own amplifier
  - More noise (reduced by subtracting ‘black’ image)
  - Lower sensitivity (lower fill rate)
- Uses standard CMOS technology
  - Allows to put other components on chip
  - ‘Smart’ pixels



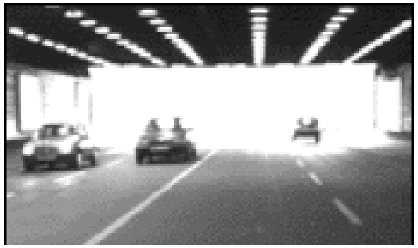


Foveon
4k x 4k sensor
0.18μ process
70M transistors

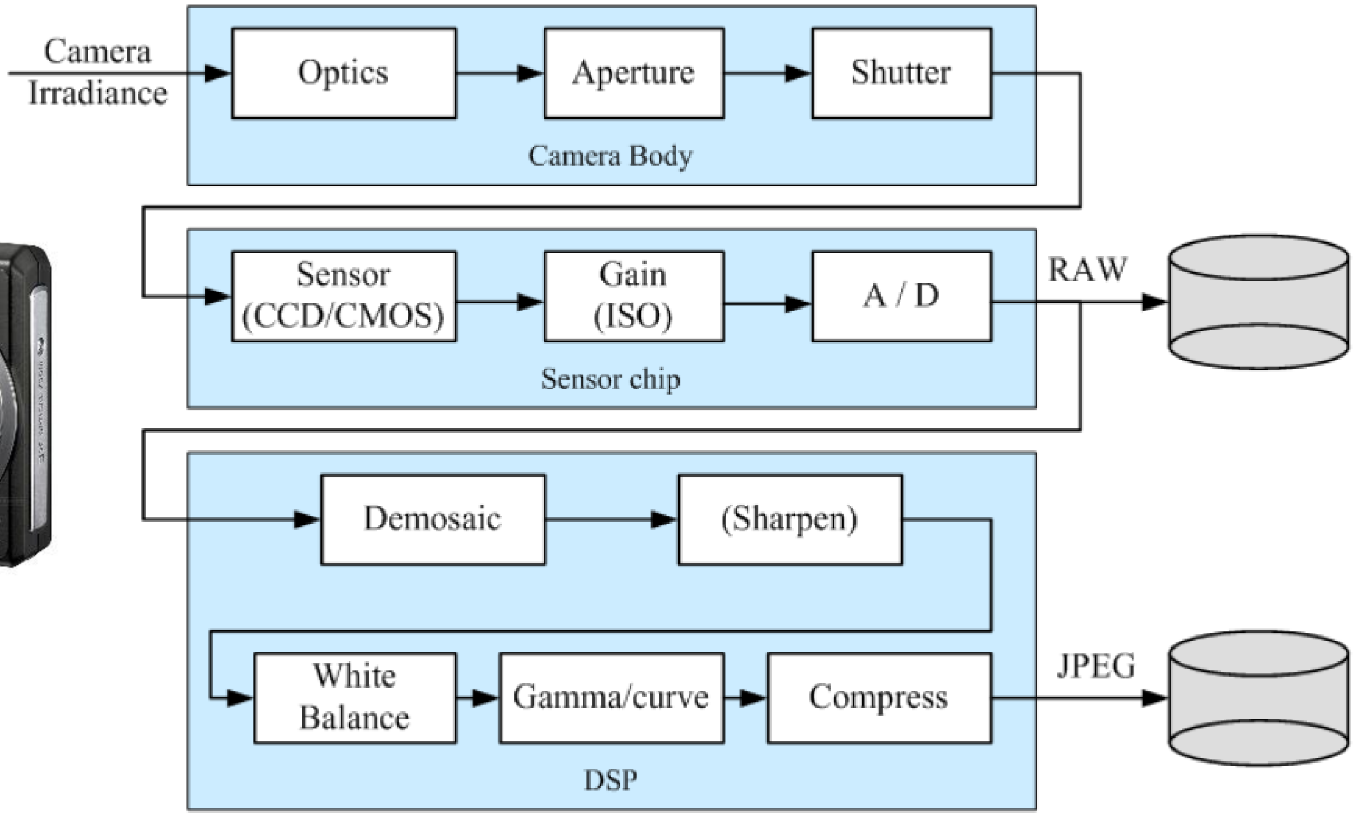
CCD vs. CMOS

- Mature technology
- Specific technology
- High production cost
- High power consumption
- Higher fill rate
- Blooming
- Sequential readout
- Low noise

- Recent technology
- Standard IC technology
- Cheap
- Low power
- Less sensitive
- Per pixel amplification
- Random pixel access
- Smart pixels
- On chip integration with other mponents



A consumer camera



Note: Gamma curve $I_{jpeg} = I^{gamma}$

Warning: Non-linear response!!

Interlacing

- Some video cameras read even lines then odd…

Rolling shutter

- Some cameras read out one line at a time:

Colour cameras

We consider 3 concepts:

1. Prism (with 3 sensors)
2. Filter mosaic
3. Filter wheel

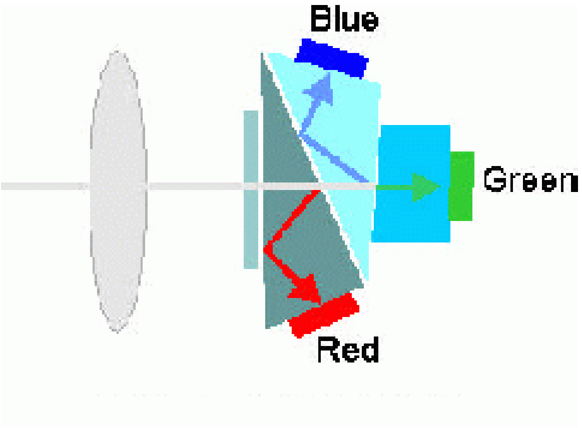
… and X3

Prism colour camera

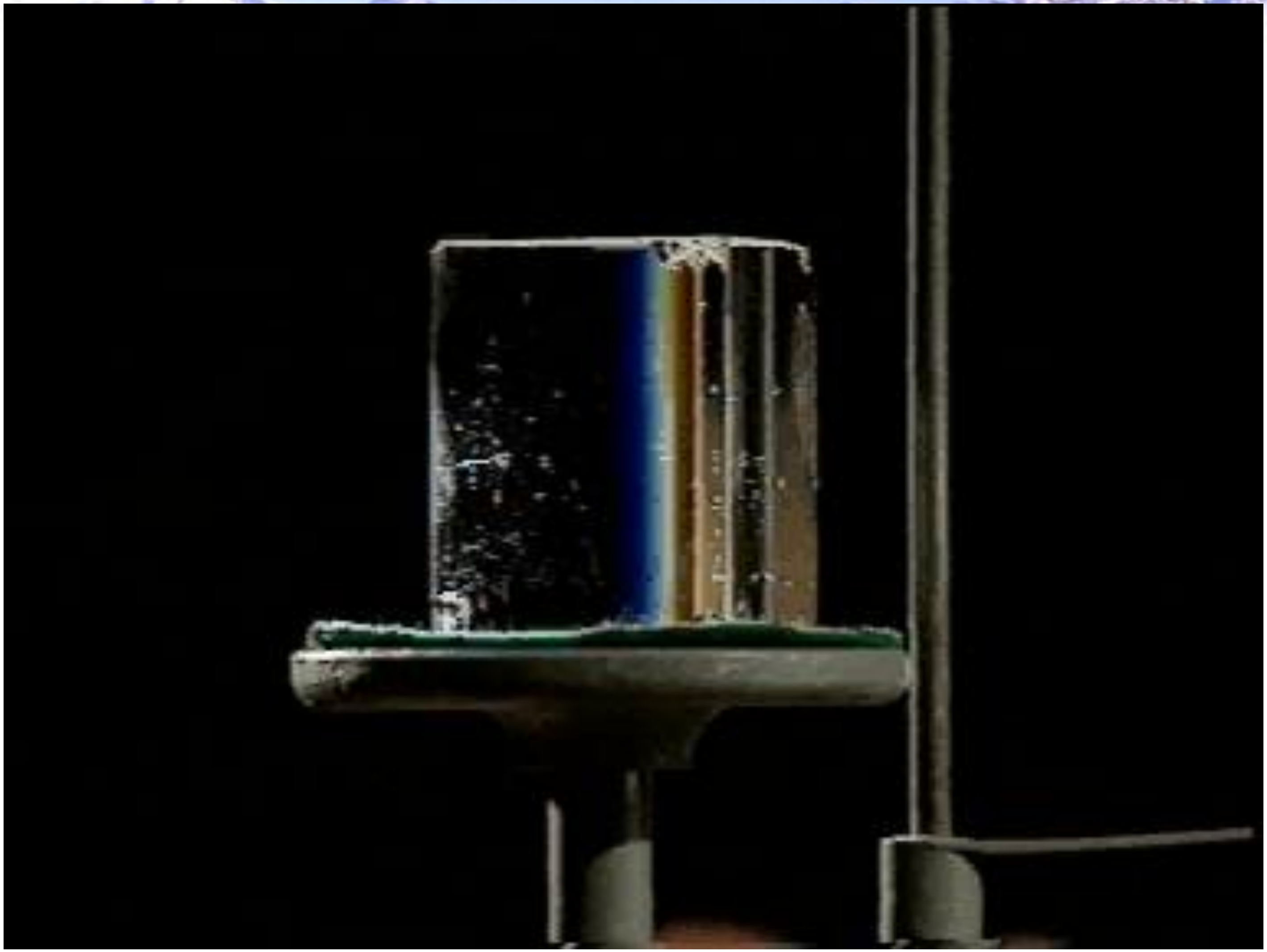
Separate light in 3 beams using dichroic prism

Requires 3 sensors & precise alignment

Good color separation

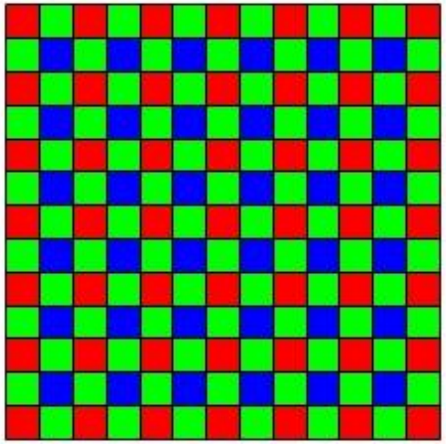


Prism colour camera

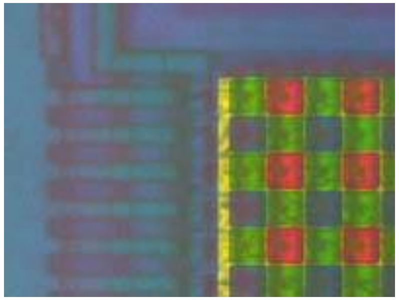


Filter mosaic

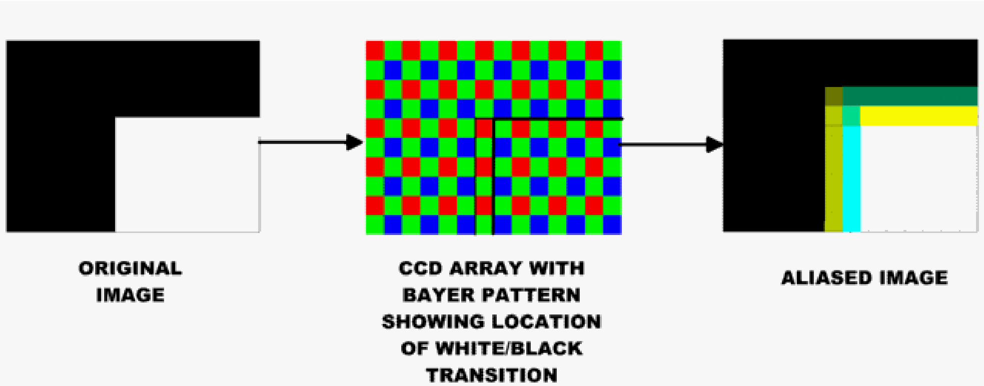
Coat filter directly on sensor



Bayer filter



Demosaicing (obtain full colour & full resolution image)



Filter wheel

Rotate multiple filters in front of lens

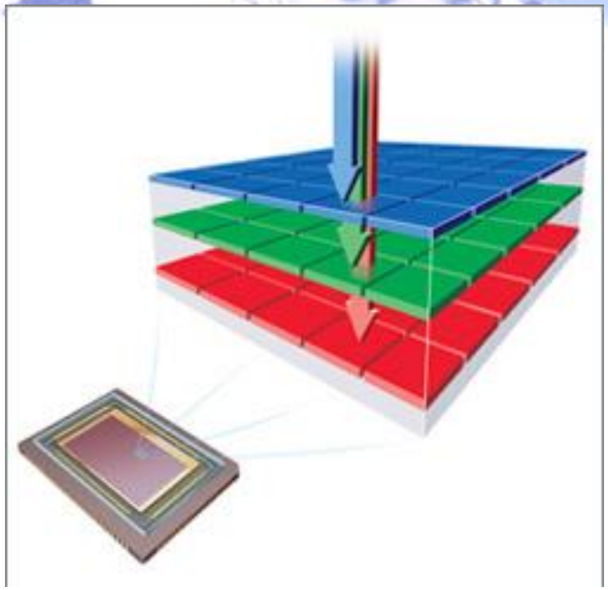
Allows more than 3 colour bands

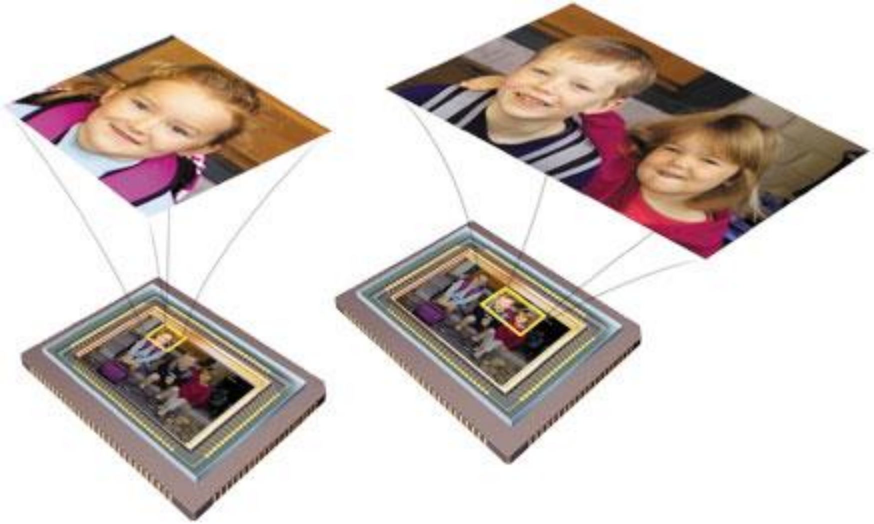
Only suitable for static scenes

Prism vs. mosaic vs. wheel

approach	Prism	Mosaic	Wheel
# sensors	3	1	1
Separation	High	Average	Good
Cost	High	Low	Average
Framerate	High	High	Low
Artefacts	Low	Aliasing	Motion
Bands	3	3	3 or more
Use:	High-end cameras	Low-end cameras	Scientific applications

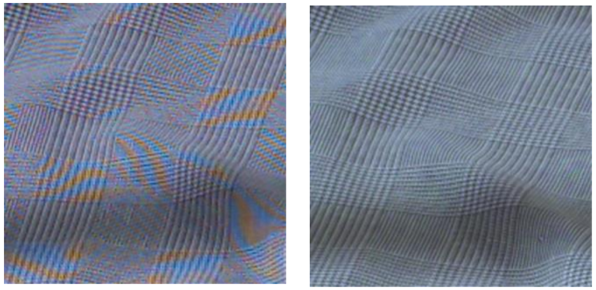
new color CMOS sensor
Foveon's X3

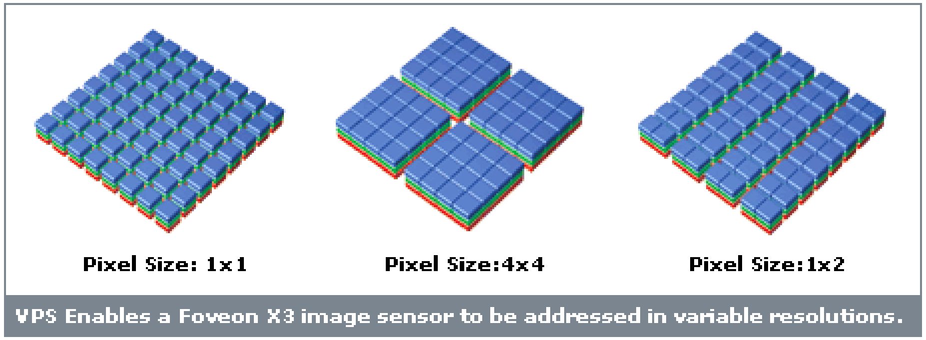




better image quality

smarter pixels

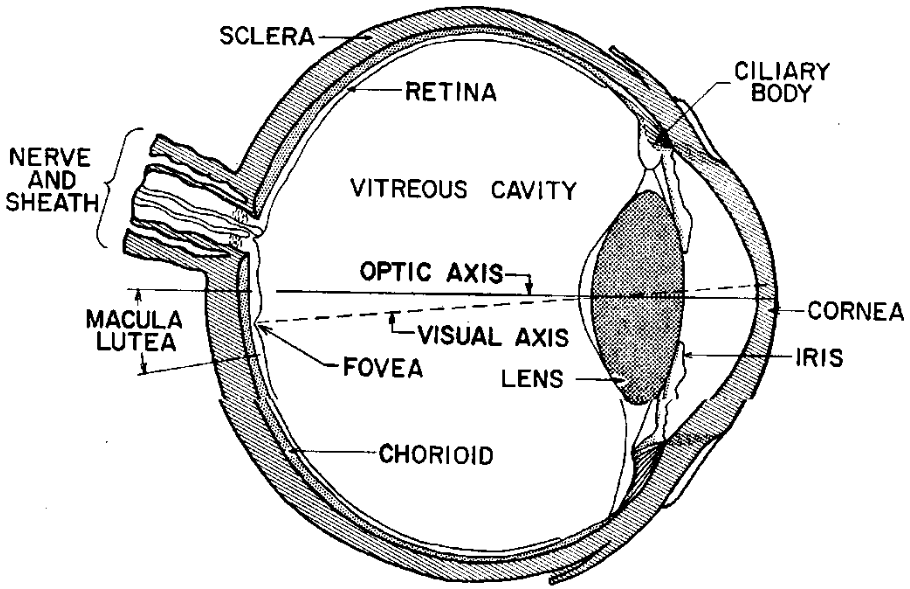




Biological implementation of camera: the eye

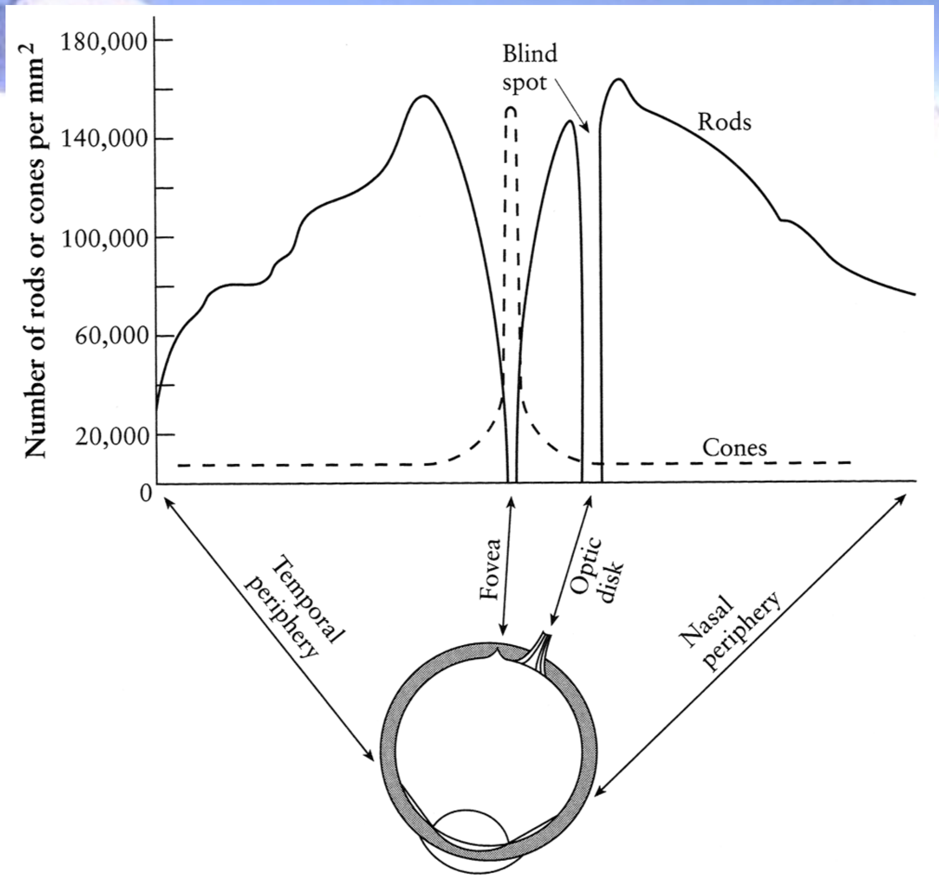
The Human Eye *is a camera…*

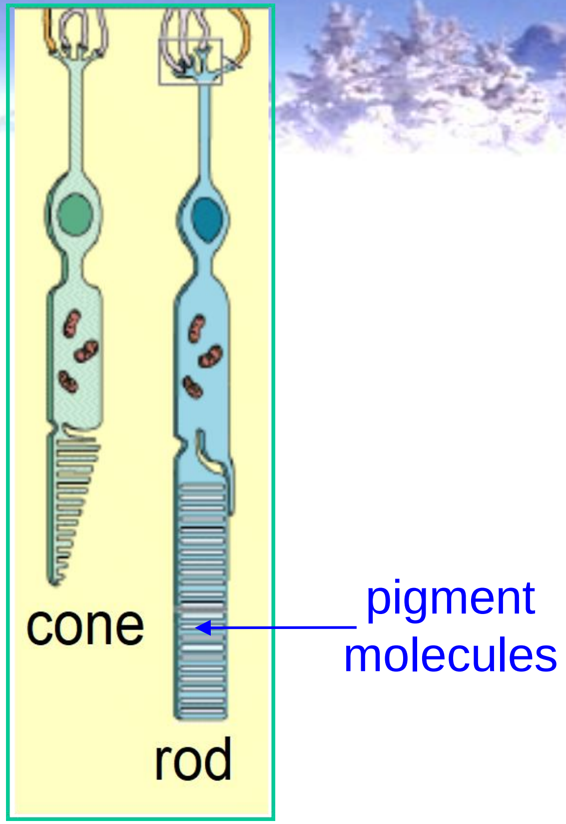
- **Iris -** colored annulus with radial muscles
- **Pupil -** the hole (aperture) whose size is controlled by the iris
- **Lens -** changes shape by using ciliary muscles (to focus on objects at different distances)
- What's the "film"?



– photoreceptor cells (rods and cones) in the **retina**

Density of rods and cones





- Rods and cones are *non-uniformly* distributed on the retina
  - Rods responsible for intensity, cones responsible for color
  - **Fovea -** Small region (1 or 2°) at the center of the visual field containing the highest density of cones (and no rods).
  - Less visual acuity in the periphery—many rods wired to the same neuron

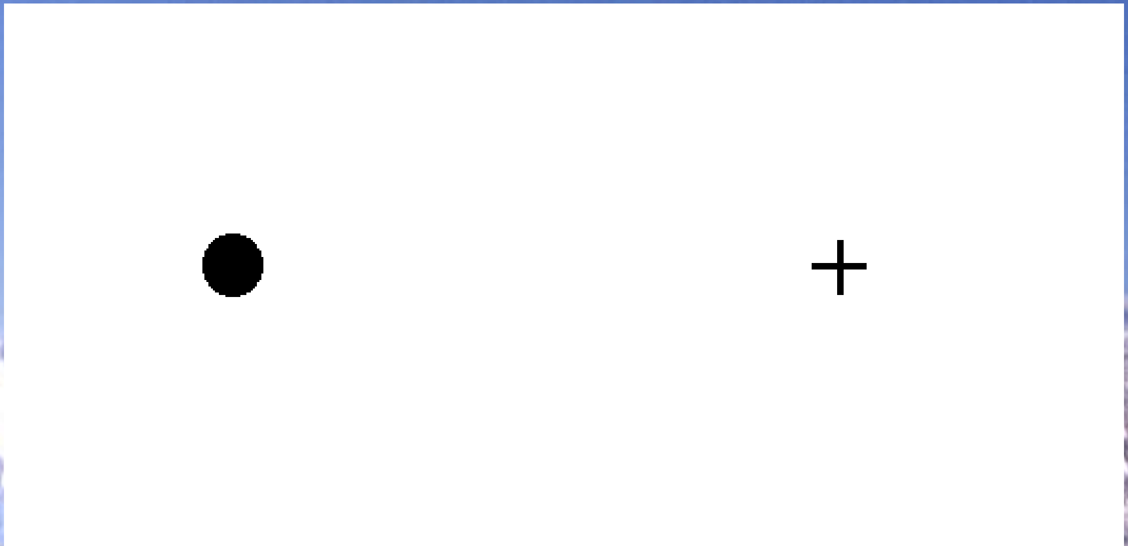
Slide by Steve Seitz

Blindspot

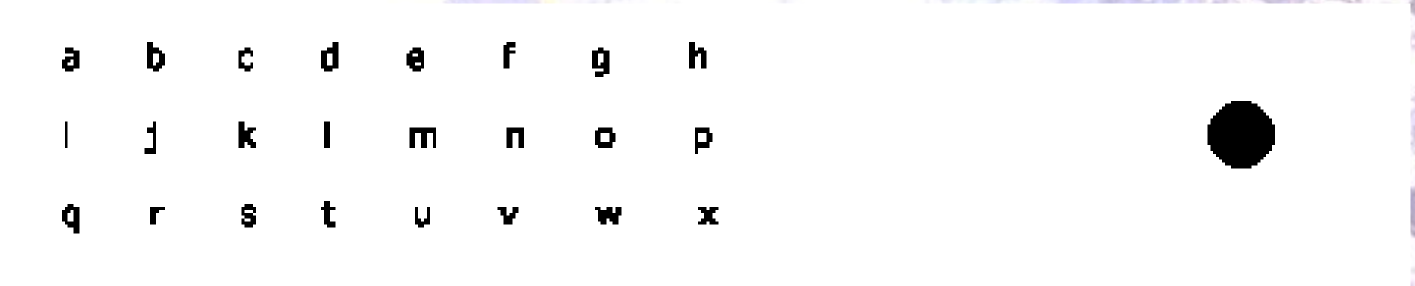
color? structure? motion?

http://ourworld.compuserve.com/homepages/cuius/idle/percept/blindspot.htm

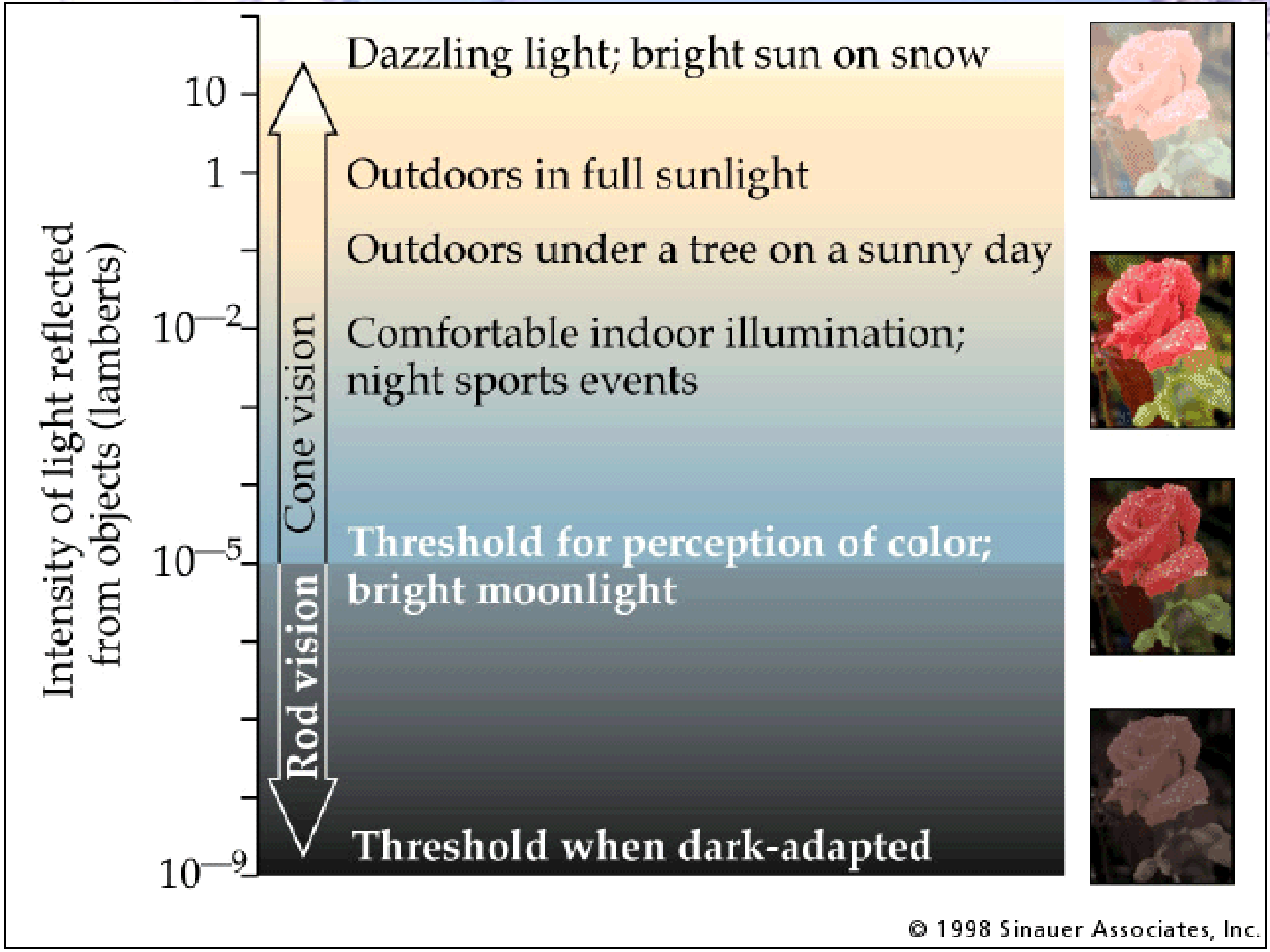
Left eye



Right eye



Rod / Cone sensitivity

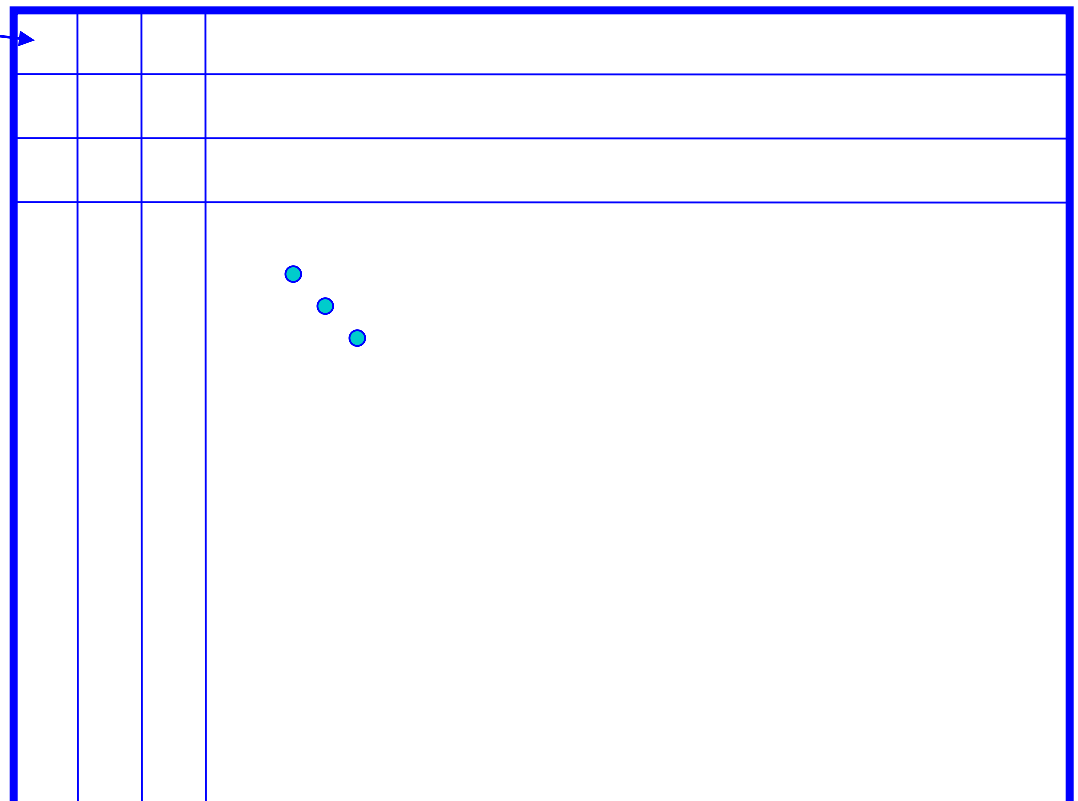


Why can't we read in the dark?

Slide by A. Efros

THE ORGANIZATION OF A 2D IMAGE

Pixel



<u>Binary</u>
1 bit

<u>Grey</u>
1 byte

<u>Color</u>
3 bytes

Mathematical / Computational image models

- Continuous mathematical:

$$I = f(x,y)$$

- Discrete (in computer) adressable 2D array:

$$I = matrix(i,j)$$

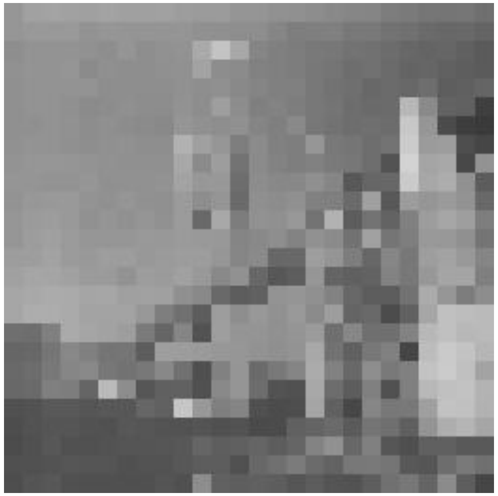
- Discrete (in file) e.g. ascii or binary sequence:

023 233 132 232

125 134 134 212

Sampling

- Standard video: 640x480
- Subsample ½, ¼…
- Quantization: typ 8 bit, sometimes lower

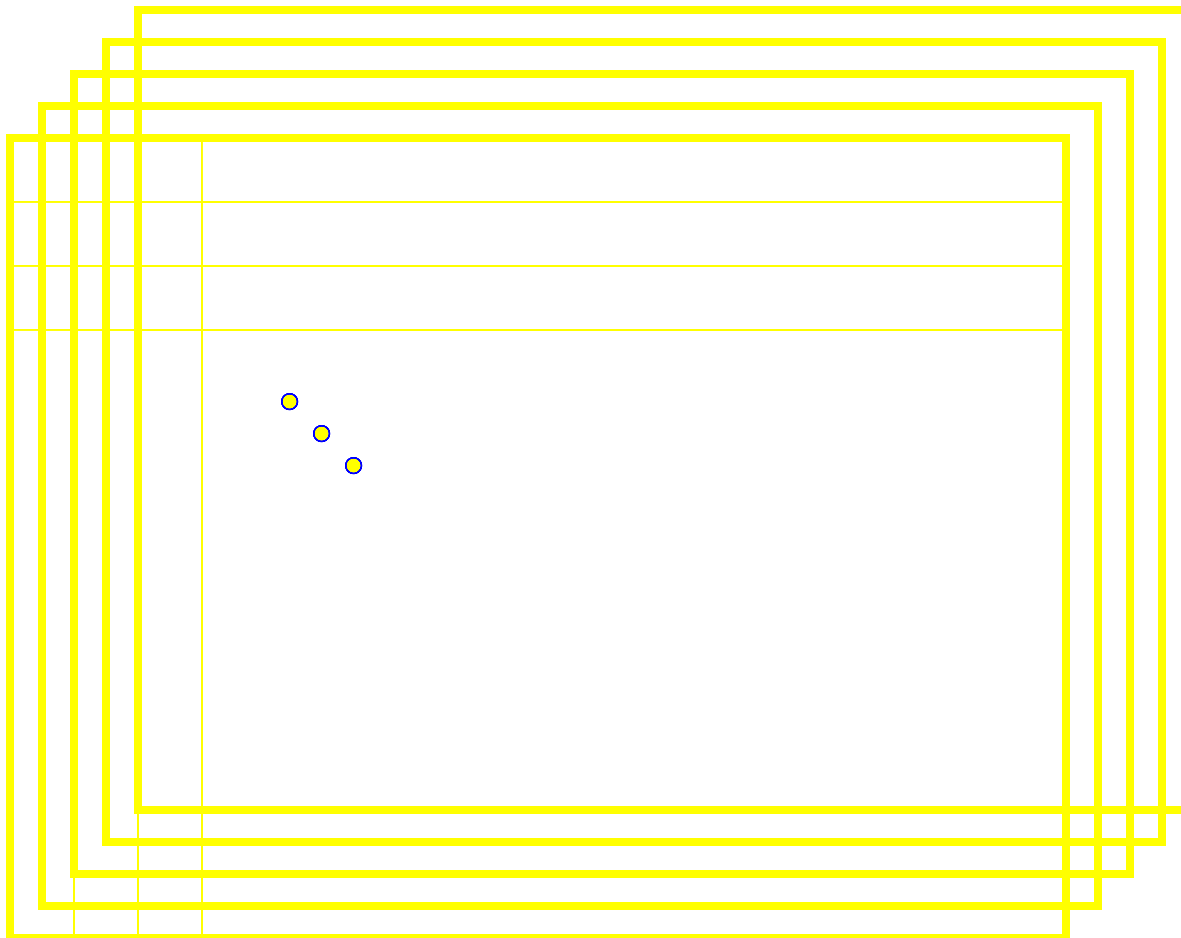



THE ORGANIZATION OF AN IMAGE SEQUENCE

Frames

Frames are acquired at 30Hz (NTSC)

Frames are composed of two *fields* consisting of the even and odd rows of a frame



BANDWIDTH REQUIREMENTS

Standard VGA (web-cam) video:

<u>Binary</u>
1 bit * 640x480 * 30 = 9.2 Mbits/second

<u>Grey</u>
1 byte * 640x480 * 30 = 9.2 Mbytes/second

<u>Color</u>
3 bytes * 640x480 * 30 = 27.6 Mbytes/second (actually about 37 mbytes/sec)

Typical operation: 3x3 convolution
9 multiplies + 9 adds ➔ 180 Mflops

Digitization Effects

- The “diameter” d of a pixel determines the highest frequency representable in an image

$$l = 1/2d$$

- Real scenes may contain higher frequencies resulting in aliasing of the signal.

- In practice, this effect is often dominated by other digitization artifacts.

Other image sources:

- Optic Scanners (linear image sensors)
- Laser scanners (2 and 3D images)
- Radar
- X-ray
- NMRI

Image display

- VDU
- LCD
- Printer
- Photo process
- Plotter (x-y table type)

Image representation for display

- True color, RGB, ….

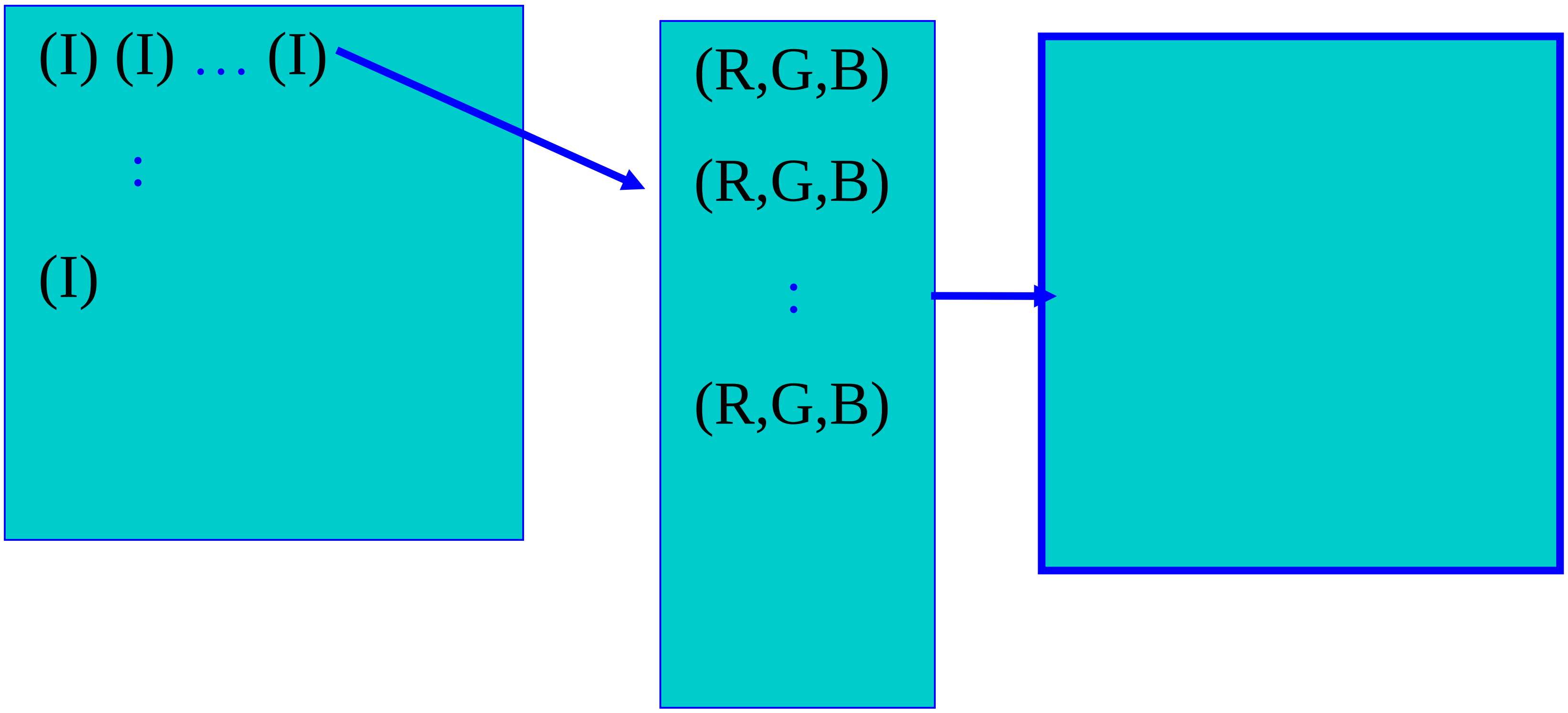
(R,G,B) (R,G,B) … (R,G,B)

:

(R,G,B)

Image representation for display

- Indexed image



Matlab Programming

Raw Material: Images = Matrices

Themes: Build systems, experiment, visualize!

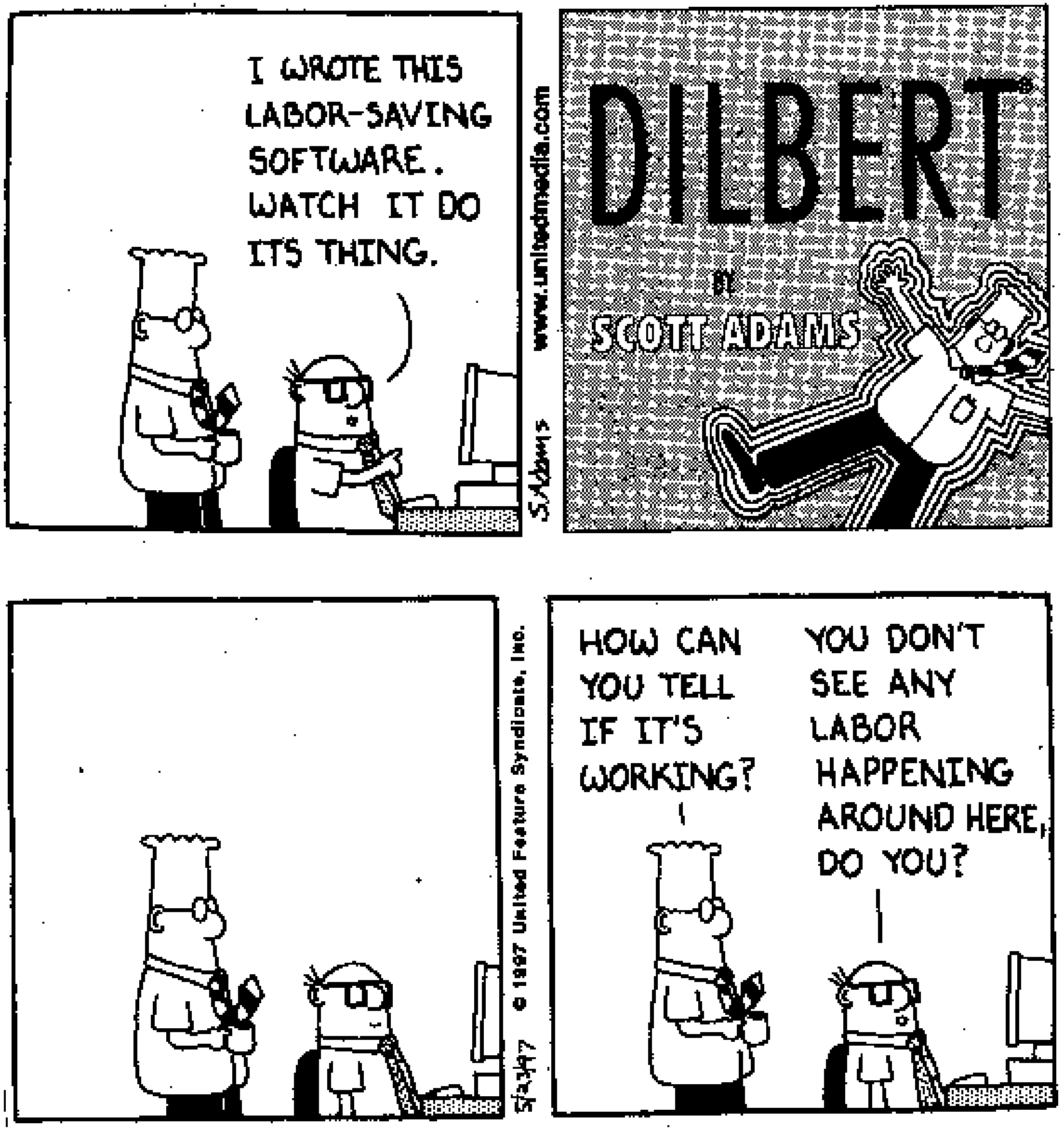
Platform: Matlab (“ matrix laboratory ”)

- Widely-used mathematical scripting language
- Easy prototyping of systems
- Lots of built-in functions, data structures
- GUI-building support
- All in all, hopefully a labor-saving tool

Matlab availability

- In lab, machines ub01 to ub08
- For remote logins: ssh to a server, then ubXX
- For your own use: Download Matlab from UofA OnTheHub

Homework: Go though exercises in matlab compendium posted on lab www-page.


I WROTE THIS LABOR-SAVING SOFTWARE. WATCH IT DO ITS THING.
DILBERT
BY SCOTT ADAMS
www.unitedmedia.com
S.Adams
© 1997 United Feature Syndicate, Inc.
5/23/97
HOW CAN YOU TELL IF IT'S WORKING?
YOU DON'T SEE ANY LABOR HAPPENING AROUND HERE, DO YOU?

Matlab basics

- Starting, stopping, help, demos, math, & variables
- Matrix definition and indexing

>> A = [1 2 3 ; 4 5 6 ; 7 8 9]   or $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

>> A(3,2)

>> A(3,:)

>> A(3,1:2) = [0 0]

>> A'

How would you set the middle row to be the first column?

>> A(:,:,2) = A

>> size(A)

See Assignment 1, part 1 for a more thorough introduction.

Image $\longleftrightarrow$ Matrix

size(A)

$\boxed{\text{Matlab matrix A}}$

A(1:10,1:10,:)

A(200, 50:300, 3)

The large "M"?

The spam's location?

Matlab Built-Ins

- for, if, while, switch -- execution control
- who, whos, clear -- variable listing and removing
- save, load <file> -- saving or restoring a workspace
- diary <file> -- start recording to a file
  diary off ; diary on
- path, addpath -- display or add to search path
- close, close all, clc -- close windows, clear console
- double vs. uint8 -- data casting functions
- zeros(x,y,…) -- creates an all-zero x by y … matrix
  used for basic memory allocation

Images in Matlab (& Functions)

Built-in functions:

A =imread(<filename>, <type>) -- pull from file

imwrite(A, <filename>, <type>) -- write to file

image(A) -- display image

imshow(A) -- display image

Types
'tif'
'jpg'
'bmp'
'png'
'hdf'
'pcx'
'xwd'

single-quoted strings

functions: Add:

show(A) -- display and tools for