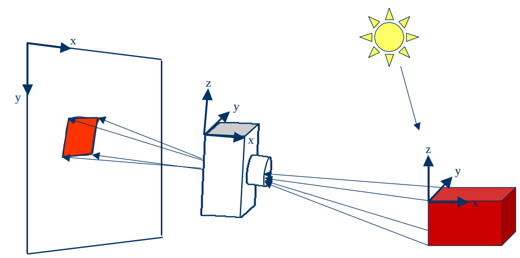
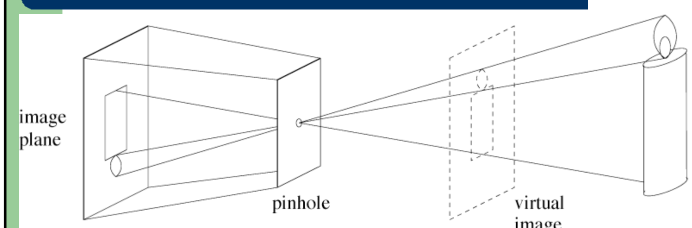
Image Capture and Representation

C306
Fall 2001
Martin Jagersand

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

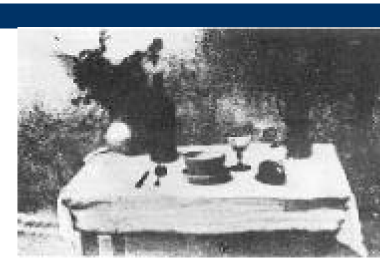


Pinhole cameras



- 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

Pinhole cameras: Historic and real



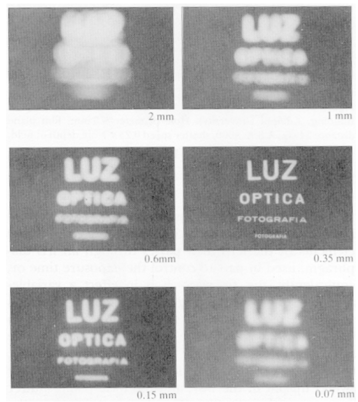
- First photograph due to Niepce,
- First on record shown - 1822
- Basic abstraction is the pinhole camera
  - lenses required to ensure image is not too dark
  - various other abstractions can be applied

Real Pinhole Cameras

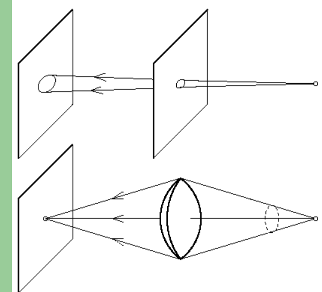
Pinhole too big - many directions are averaged, blurring the image

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

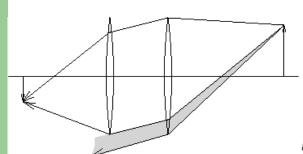


Note: Each world point projects to many image points.

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

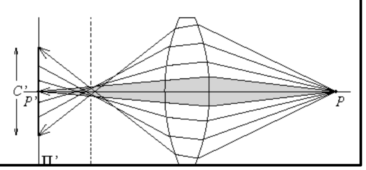
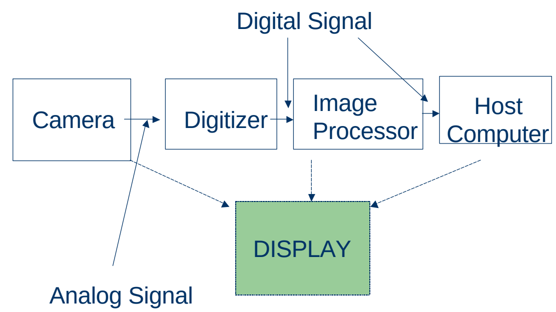
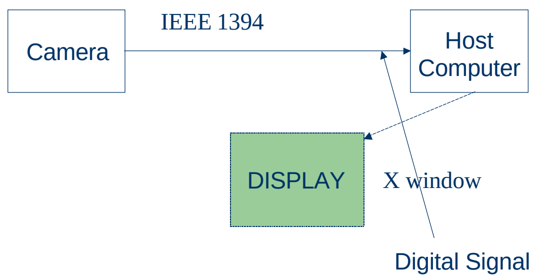


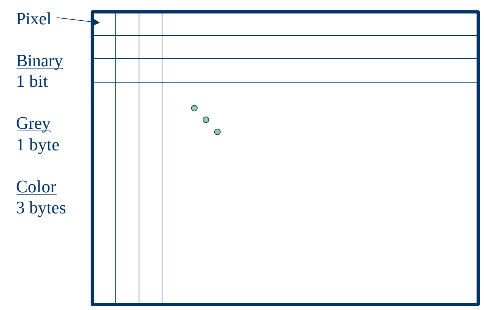
Image streams -> Computer



A Modern Digital Camera (Firewire)



THE ORGANIZATION OF A 2D IMAGE



Mathematical / Computational image models

- Continuous mathematical:

  I = f(x,y)
- Discrete (in computer) adressable 2D array:

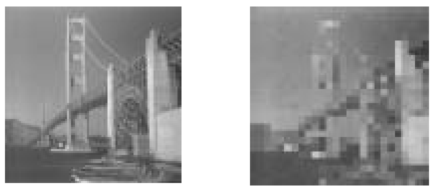
  I = matrix(i,j)
- Discrete (in file) e.g. ascii or binary sequence:

  023 233 132 232

  125 134 134 212

Sampling

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

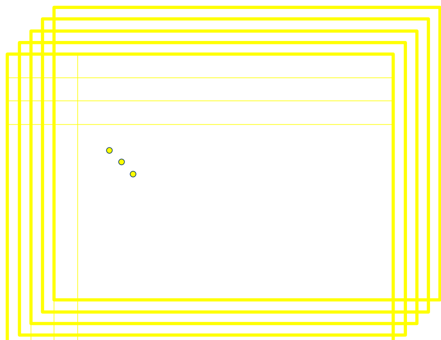


THE ORGANIZATION OF AN IMAGE SEQUENCE

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

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

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

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

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

Today's PC's are just getting to the point they can process images at frame rate

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, ….

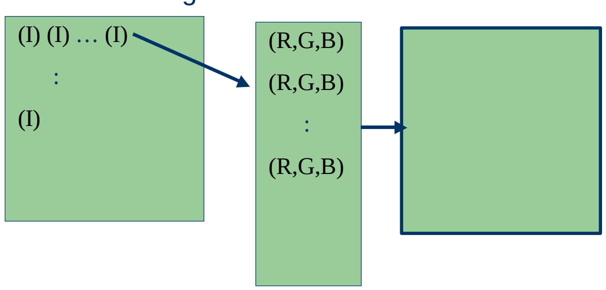
(R,G,B) (R,G,B) … (R,G,B)

:

(R,G,B)

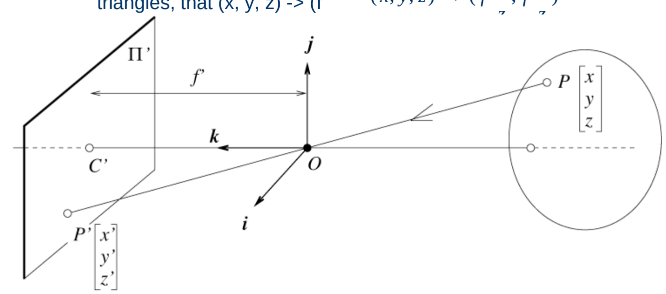
Image representation for display

- Indexed image

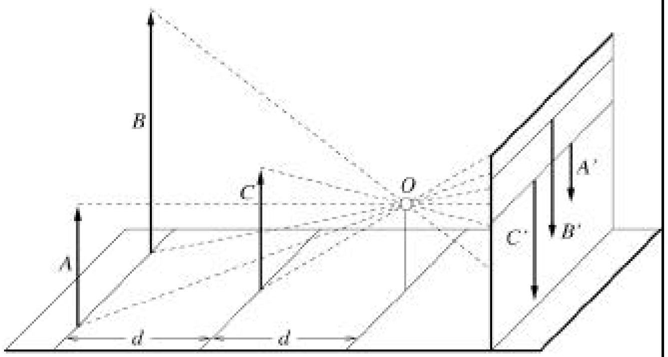


The equation of projection

- Cartesian coordinates:
  - We have, by similar triangles, that (x, y, z) -> (f $(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$

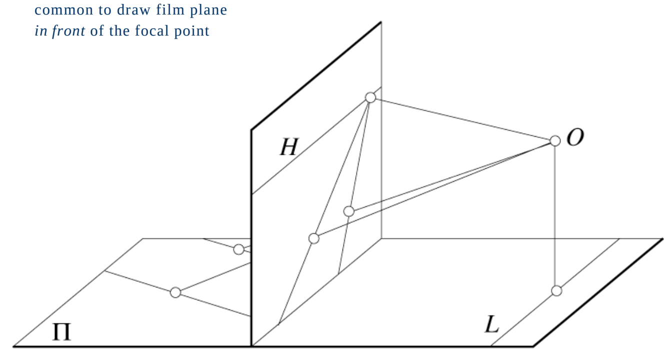


Distant objects are smaller



Parallel lines meet

common to draw film plane
in front of the focal point



Vanishing points

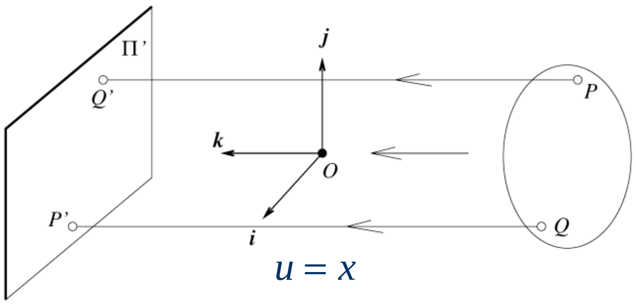
- each set of parallel lines (=direction) meets at a different point
  - The *vanishing point* for this direction
  - How would you show this?
- Sets of parallel lines on the same plane lead to *collinear* vanishing points.
  - The line is called the *horizon* for that plane

The camera matrix

- Homogenous coordinates for 3D
  - four coordinates for 3D point
  - equivalence relation (X,Y,Z,T) is the same as (k X, k Y, k Z,k T)
- Turn previous expression into HC's
  - HC's for 3D point are (X,Y,Z,T)
  - HC's for point in image are (U,V,W)

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \qquad (U,V,W) \rightarrow \left(\frac{U}{W}, \frac{V}{W}\right) = (u,v)$$

Orthographic projection



$$u = x$$

$$v = y$$

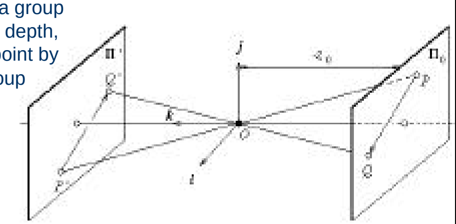
Weak perspective

$$u = Tx$$

$$v = Ty$$

$$T = f / Z$$

- 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: wrong



The fundamental model for orthographic projection

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Camera parameters

- Issue
  - camera may not be at the origin, looking down the z-axis
    - extrinsic parameters
  - one unit in camera coordinates may not be the same as one unit in world coordinates
    - intrinsic parameters - focal length, principal point, aspect ratio, angle between axes, etc.

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{projection model} \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Geometric Transforms

In general, a point in n-D space transforms by

P' = rotate(point) + translate(point)

In 2-D space, this can be written as a matrix equation:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} Cos(q) & -Sin(q) \\ Sin(q) & Cos(q) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} tx \\ ty \end{pmatrix}$$

In 3-D space (or n-D), this can generalized as a matrix equation:

p' = R p + T or p = R^t (p' – T)

Geometric Transforms

Now, using the idea of homogeneous transforms,
we can write:

$$p' = \begin{pmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{pmatrix} p$$

R and T both require 3 parameters. These correspond
to the 6 extrinsic parameters needed for camera calibration

Intrinsic Parameters

Intrinsic Parameters describe the conversion from
metric to pixel coordinates (and the reverse)

$x_{mm} = - (x_{pix} - o_x)\, s_x$
$y_{mm} = - (y_{pix} - o_y)\, s_y$

or

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix}_{pix} = \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}_{mm} = M_{int}\, p$$