Precept 6: Quiz Review

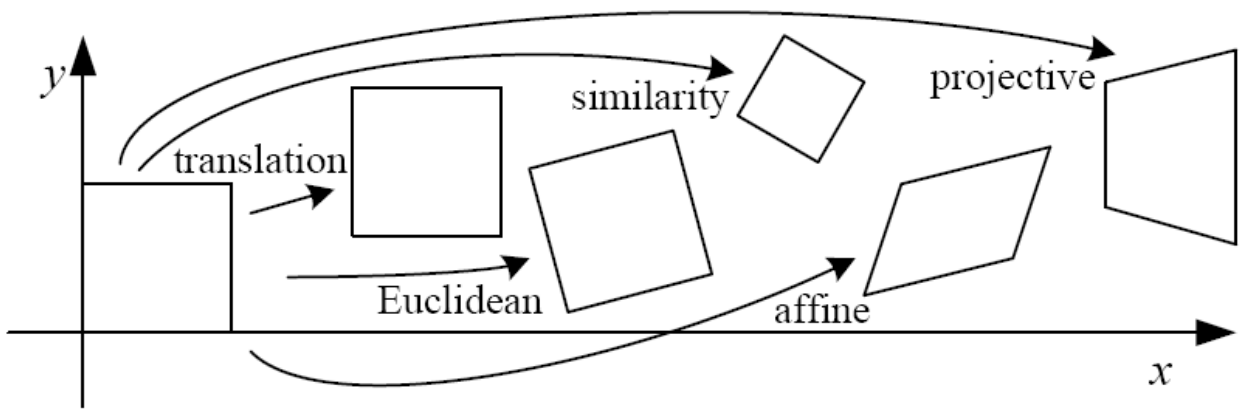
Quiz Date: Thursday Dec 17

Format

- 50 minutes
- 6 problems, 10 points each
- A one-page double-sided letter-size color cheat sheet allowed
- Similar to the review questions in format and in topic
- Mostly conceptual

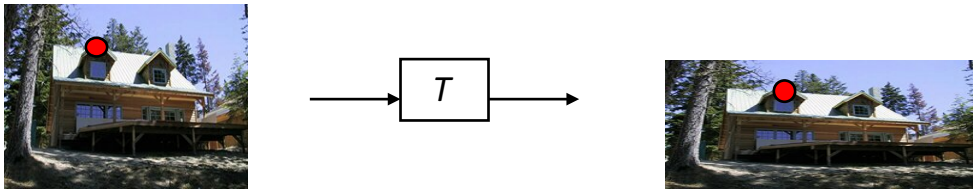
Problem 1: Affine Transformations

2D Geometric Transformations



2D Geometric Transformations

- Transformation T is a coordinate-changing machine:
  - p' = *T*(p)
- Transformation T is global
  - Is the same for any point p
  - Can be described by just a few parameters



p = (x,y)

p' = (x',y')

Linear Transformation

$$\mathbf{p}' = \mathbf{T}\mathbf{p} \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix}$$

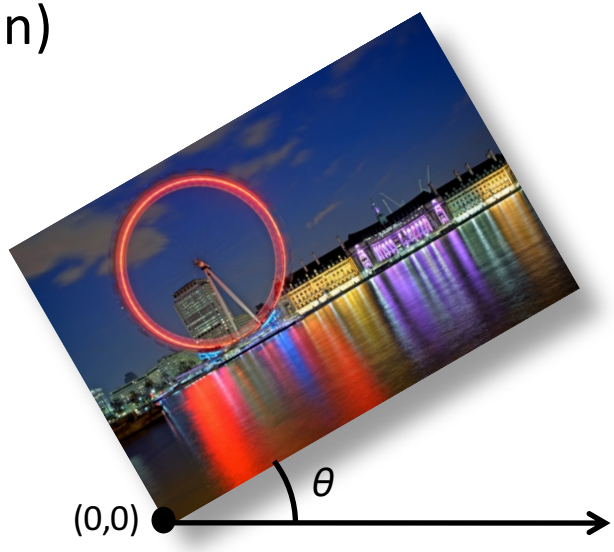
Common linear transformations

- Uniform scaling by *s*:

$$\mathbf{S} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$$

Common linear transformations

- Rotation by angle θ (about the origin)



$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

2x2 Matrices

- What types of transformations can be represented with a 2x2 matrix?

2D mirror about Y axis?

$$\begin{aligned} x' &= -x \\ y' &= y \end{aligned} \qquad \mathbf{T} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

2D mirror across line y = x?

$$\begin{aligned} x' &= y \\ y' &= x \end{aligned} \qquad \mathbf{T} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Translation

- Solution: homogeneous coordinates to the rescue

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

Affine transformations

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

any transformation with last row [0 0 1] we call an *affine* transformation

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

Affine transformations

- Affine transformations are combinations of …
  - Linear transformations, and
  - Translations

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Problem 1: Affine Transformations

1) How many degrees of freedom does an affine transformation have?

Problem 1: Affine Transformations

1) How many degrees of freedom does an affine transformation have?

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \boxed{\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

a. Origin will be mapped to origin

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

a. Origin will be mapped to origin

False: Translation

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

b. Lines will be mapped to lines

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

b. Lines will be mapped to lines

True: y = mx + n

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

b. Lines will be mapped to lines

True: y = mx + n

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

x’ = ax + by + cw = ax + b(mx + n) + cw = (a + bm)x + bn + cw

y’ = dx + ey + fw = dx + e(mx + n) + fw = (d + em)x + en + fw

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

b. Lines will be mapped to lines

True: $y = mx + n$

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$x' = ax + by + cw = ax + b(mx + n) + cw = (a + bm)x + bn + cw$

$y' = dx + ey + fw = dx + e(mx + n) + fw = (d + em)x + en + fw$

$\Rightarrow \quad x = (x' - bn - cw)/(a+bm)$

$\Rightarrow \quad y' = (d + em)/(a + bm)\ (x' - bn - cw) + en + fw$

$y' = (d + em)/(a + bm)\ x' - (bn+cw)(d + em)/(a + bm) + en + fw$

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

c. Parallel lines will not remain parallel

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

c. Parallel lines will not remain parallel

False:

$y = mx + n_1$

$y = mx + n_2$

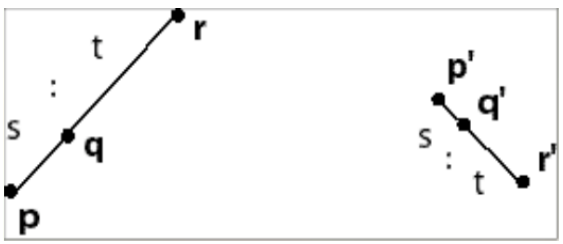
$y' = (d + em)/(a + bm)\ x' - (bn_1+cw)(d + em)/(a + bm) + en_1 + fw$

$y' = (d + em)/(a + bm)\ x' - (bn_2+cw)(d + em)/(a + bm) + en_2 + fw$

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

d. Ratios (along any line) will be preserved



Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

d. Ratios (along any line) will be preserved

True:

$(x_1, mx_1+n, w) \Rightarrow (ax_1 + b(mx_1+n) + cw, dx_1 + e(mx_1+n) + fw, w)$

$(x_2, mx_2+n, w) \Rightarrow (ax_2 + b(mx_2+n) + cw, dx_2 + e(mx_2+n) + fw, w)$

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

d. Ratios (along any line) will be preserved

True:

$(x_1, mx_1+n, w) \Rightarrow (ax_1 + b(mx_1+n) + cw, dx_1 + e(mx_1+n) + fw, w)$

$(x_2, mx_2+n, w) \Rightarrow (ax_2 + b(mx_2+n) + cw, dx_2 + e(mx_2+n) + fw, w)$

$\Rightarrow |X_1X_2|^2 = (1 + m^2)(x_2 - x_1)^2$

$\Rightarrow |X_1'X_2'|^2 = ((a + bm)^2 + (d + em)^2)\,(x_2 - x_1)^2$

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

e. The transformation is not closed under composition

Problem 1: Affine Transformations

2) Determine whether the below properties of affine transformations are true/false.

e. The transformation is not closed under composition

False:

```
A =

[ A1_1, A1_2, A1_3]
[ A2_1, A2_2, A2_3]
[    0,    0,    1]
```

```
B =

[ B1_1, B1_2, B1_3]
[ B2_1, B2_2, B2_3]
[    0,    0,    1]
```

```
>> A*B

ans =

[ A1_1*B1_1 + A1_2*B2_1, A1_1*B1_2 + A1_2*B2_2, A1_3 + A1_1*B1_3 + A1_2*B2_3]
[ A2_1*B1_1 + A2_2*B2_1, A2_1*B1_2 + A2_2*B2_2, A2_3 + A2_1*B1_3 + A2_2*B2_3]
[                     0,                     0,                             1]
```

Problem 2: Filtering

Box Filter

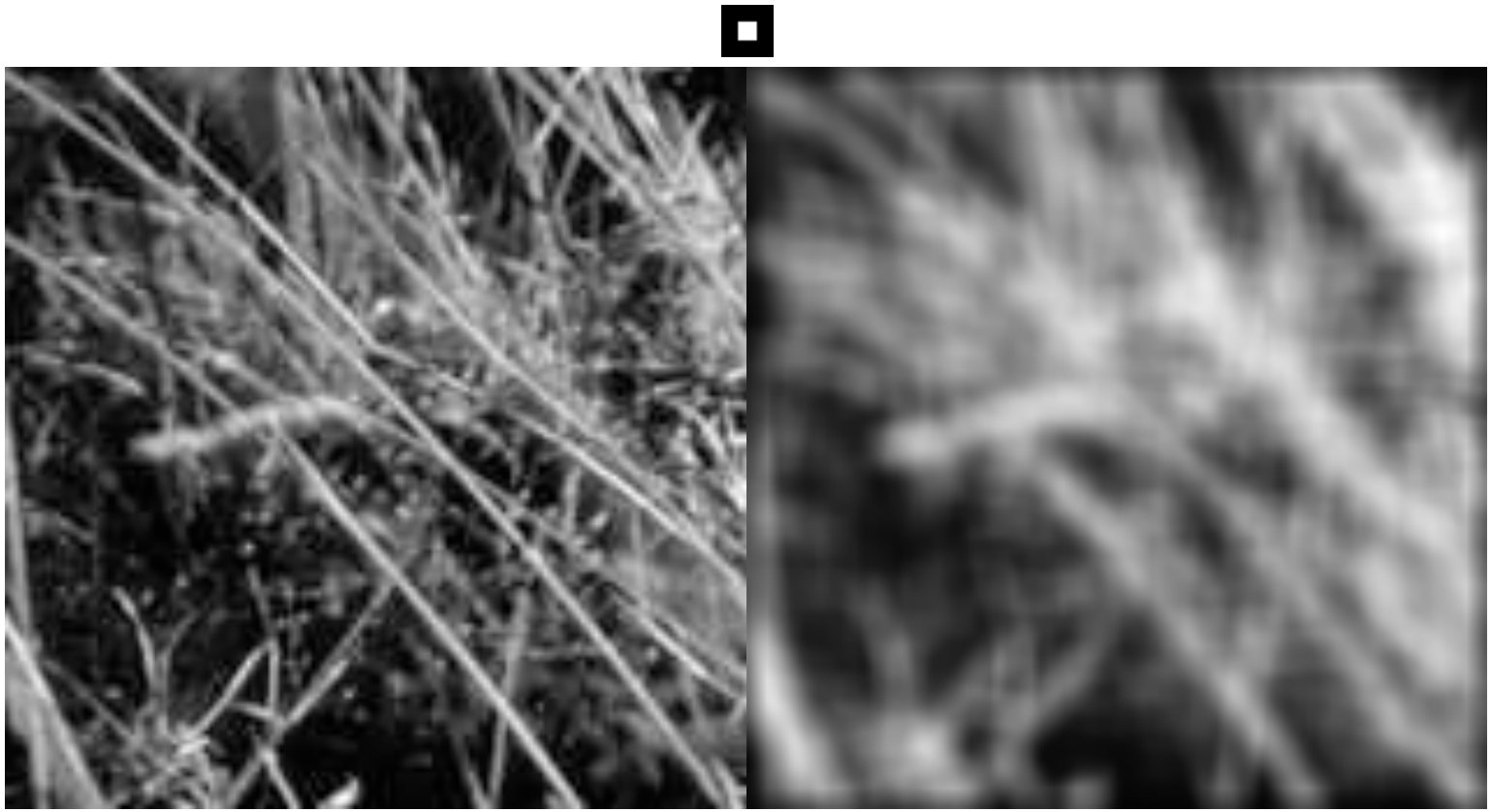
What does it do?

- Replaces each pixel with an average of its neighborhood
- Achieve smoothing effect (remove sharp features)

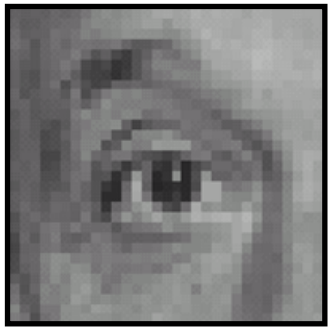
$$g[\cdot,\cdot] \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Slide credit: David Lowe (UBC)

Smoothing with box filter



Practice with linear filters



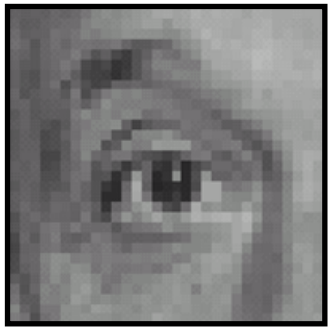
Original

0	0	0
0	1	0
0	0	0

?

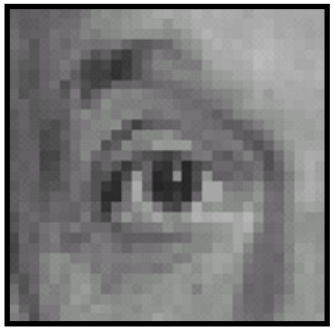
Source: D. Lowe

Practice with linear filters



Original

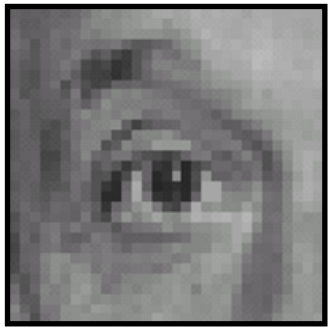
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Practice with linear filters



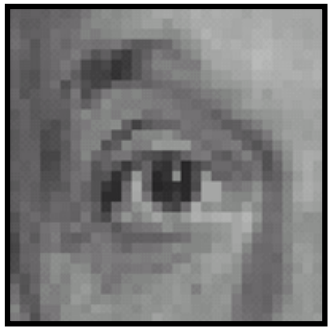
Original

0	0	0
0	0	1
0	0	0

?

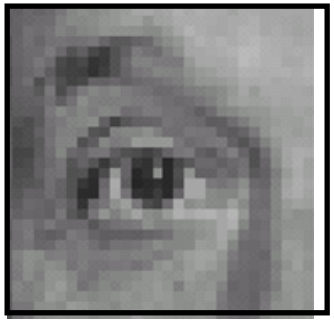
Source: D. Lowe

Practice with linear filters



Original

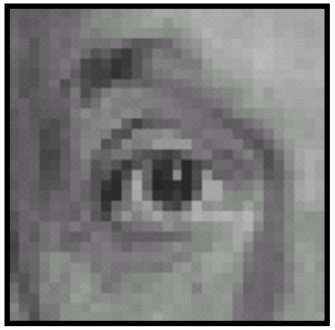
0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Source: D. Lowe

Practice with linear filters



Original

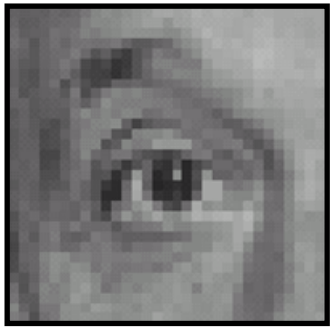
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(Note that filter sums to 1)

?

Source: D. Lowe

Practice with linear filters



Original

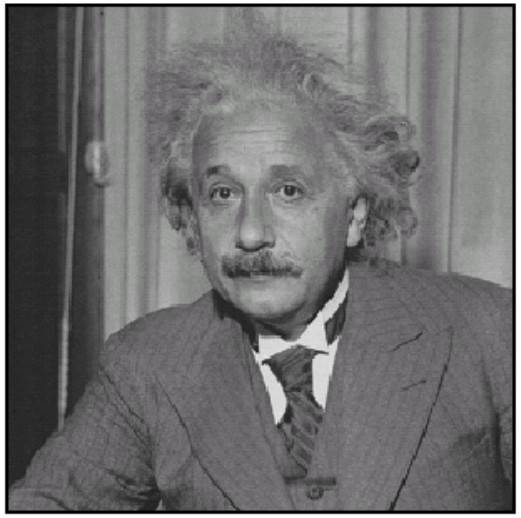
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening filter

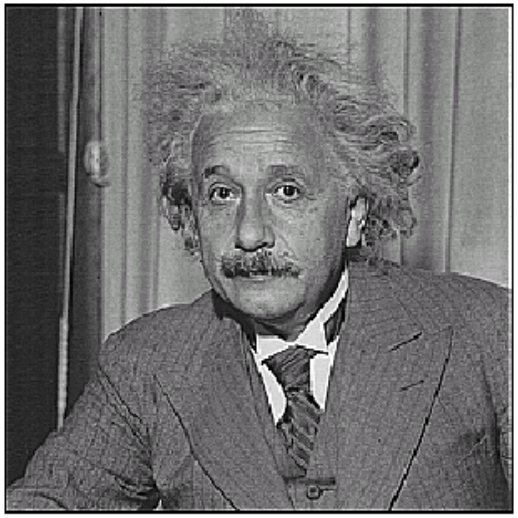
- Accentuates differences with local average

Source: D. Lowe

Sharpening



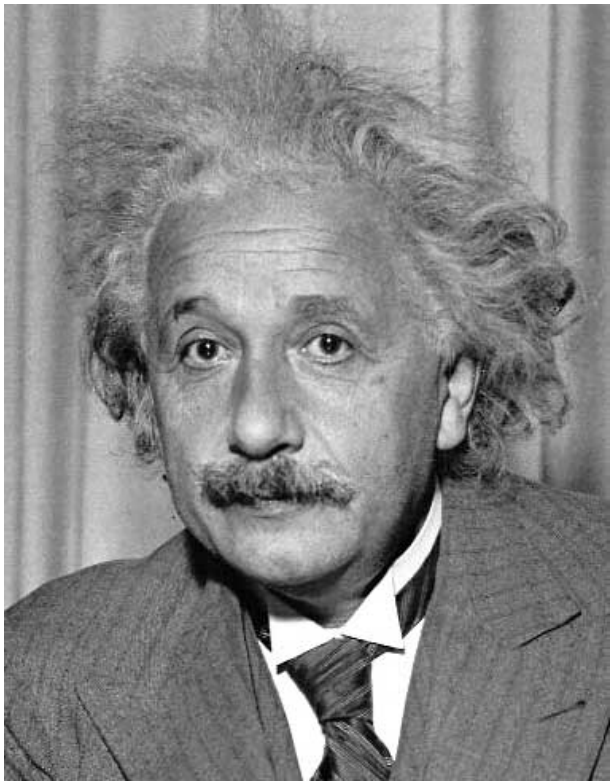
before



after

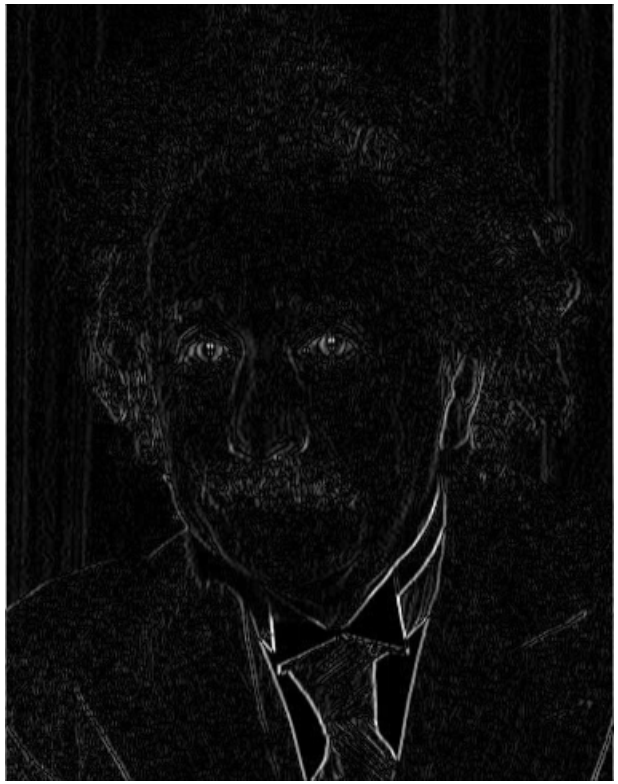
Source: D. Lowe

Other filters



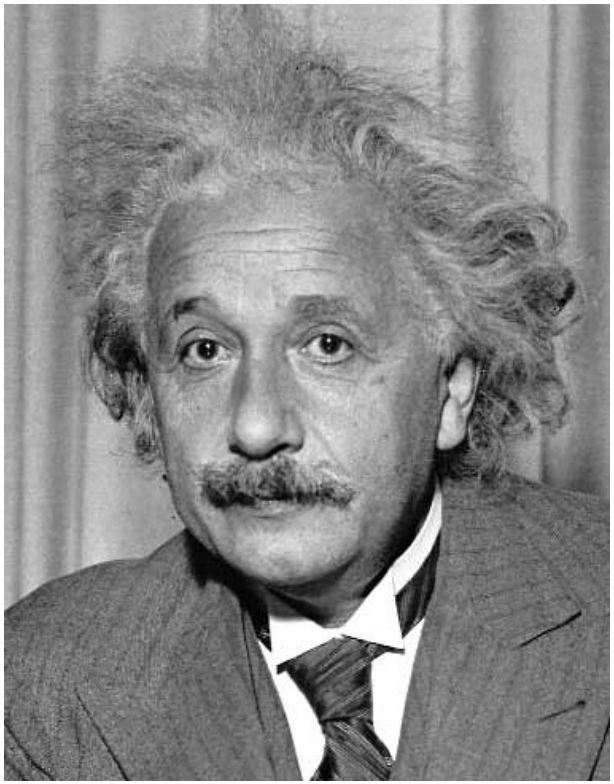
1	0	-1
2	0	-2
1	0	-1

Sobel



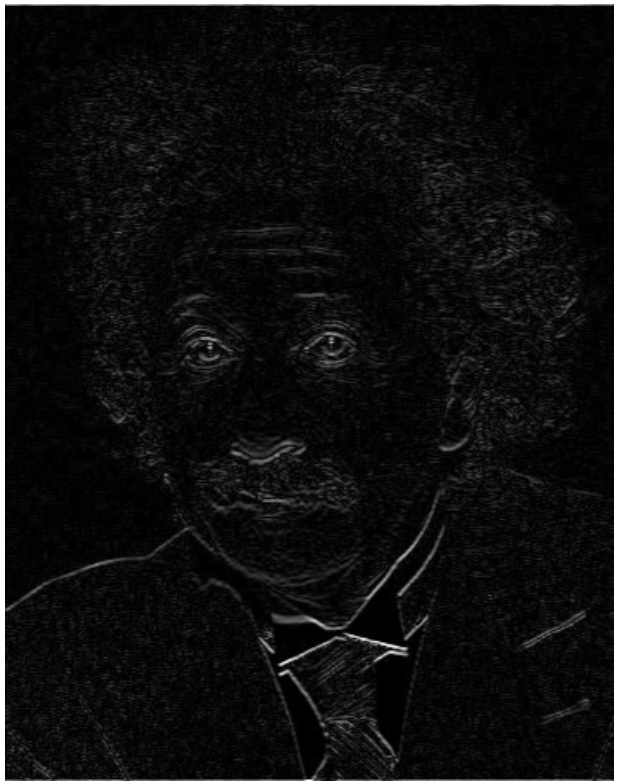
Vertical Edge
(absolute value)

Other filters



1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)

Problem 2: Filtering

1) Briefly explain what the effects would kernel 1 and kernel 2 produce when applied to a 2-D image.

0	0	0
0	0	1
0	0	0

Kernel 1

1	0	-1
2	0	-2
1	0	-1

Kernel 2

Problem 2: Filtering

2) Please design a 1-D width-3 sharpening filter

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Problem 2: Filtering

2) Please design a 1-D width-3 sharpening filter

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 2 & 0 \end{bmatrix} - \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

Problem 2: Filtering

2) and apply it to the above image and show the result.

a	b	c	d	e	f	g	h	i

$-\frac{1}{3}$	$\frac{5}{3}$	$-\frac{1}{3}$

$\frac{1}{3}$

	5b-a-c	5c-b-d	5d-c-e	5e-d-f	5f-e-g	5g-f-h	5h-g-i	

Problem 3: Camera Parameters

The Camera Matrix

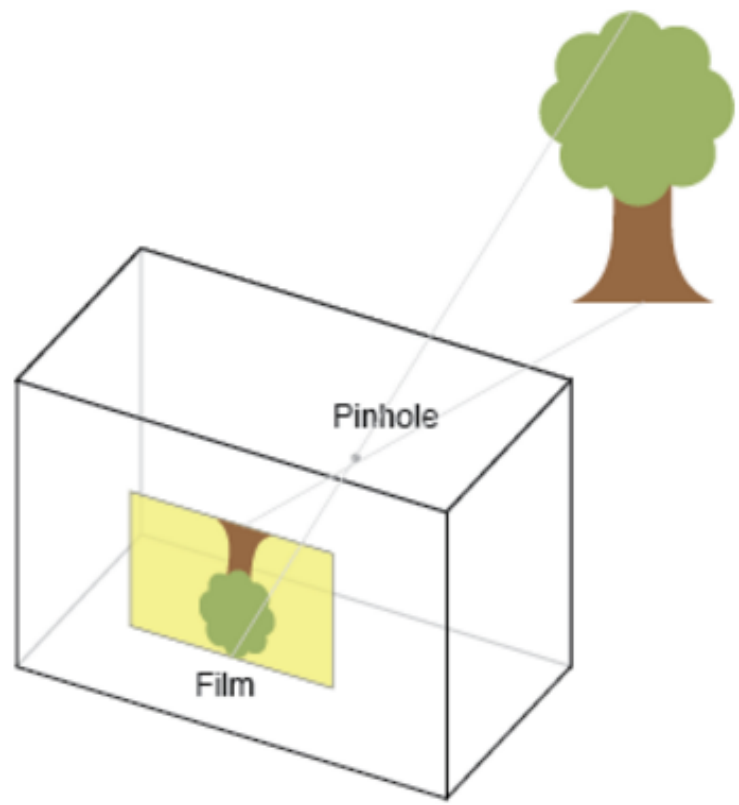
$$\mathbf{x} = \mathbf{K}\left[\mathbf{R} \middle| \mathbf{t}\right]\mathbf{X}$$

The Extrinsic Matrix

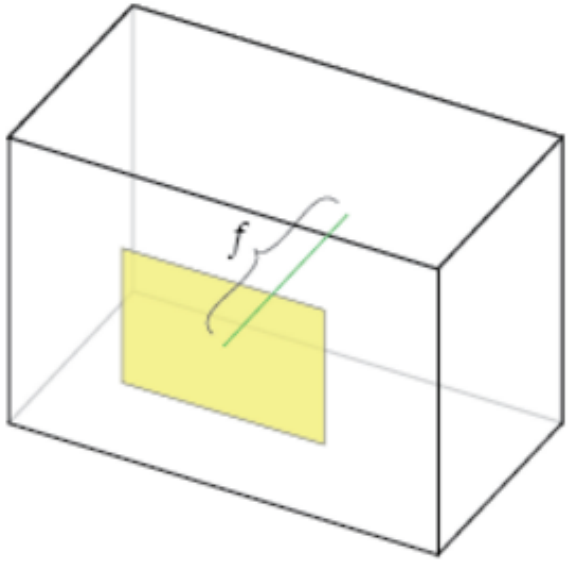
$$[R \mid \boldsymbol{t}] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right]$$

The Intrinsic Matrix

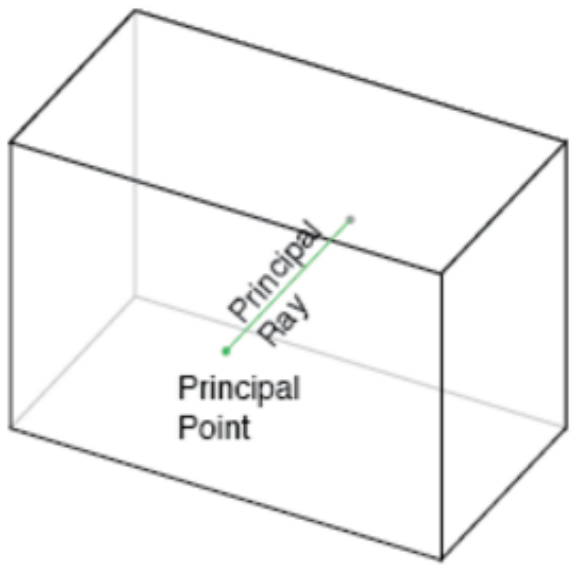
$$K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

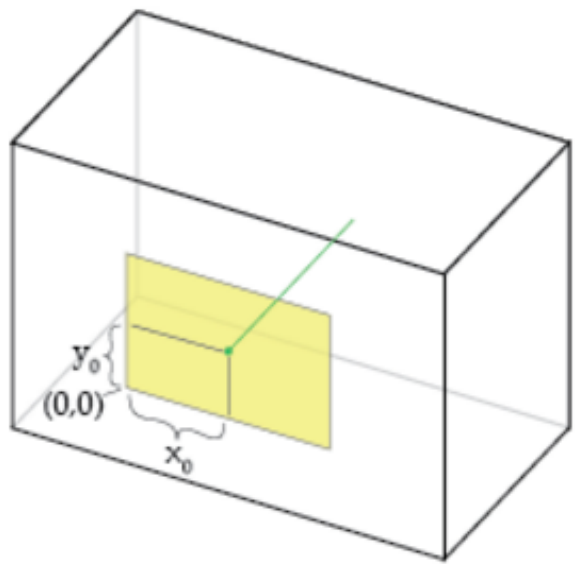


Focal Length, f_x, f_y



Principal Point Offset, x_0, y_0





Skew

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{K} = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Problem 3: Camera Parameters

1) Briefly explain what information is embedded in the camera matrices R, K, and t.

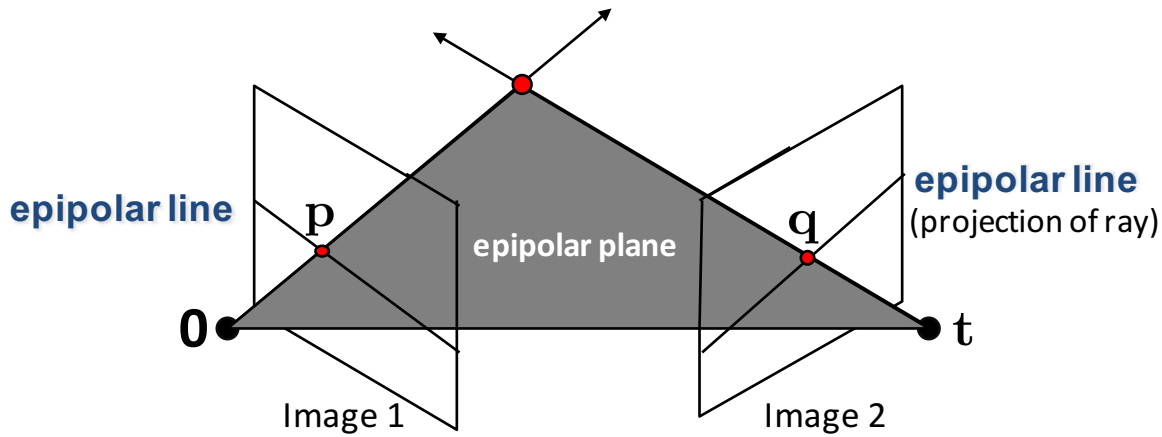
R: The directions of the world-axes in camera coordinates.
K: Focal length, principle point offset, and axis skew.
t: The position of the world origin in camera coordinates.

1) Given a 3-D point X = (X,Y,Z) and the camera matrices R, K, and t, give the expression for the image coordinates of the point.

$$\mathbf{x} = \mathbf{K}\left[\mathbf{R}|\mathbf{t}\right]\mathbf{X}$$

Problem 4: Fundamental Matrix

Fundamental matrix



- This *epipolar geometry* of two views is described by a Very Special 3x3 matrix $\mathbf{F}$, called the *fundamental matrix*
- $\mathbf{F}$ maps (homogeneous) *points* in image 1 to *lines* in image 2!
- The epipolar line (in image 2) of point **p** is: $\mathbf{F}\mathbf{p}$
- *Epipolar constraint* on corresponding points: $\mathbf{q}^T\mathbf{F}\mathbf{p} = 0$

Problem 4: Fundamental Matrix

What is the difference between a fundamental matrix and a homography? Both of them can be estimated by some point to point correspondences from two images. When should we use which, and why?

Problem 4: Fundamental Matrix

What is the difference between a fundamental matrix and a homography? Both of them can be estimated by some point to point correspondences from two images. When should we use which, and why?

Homography is only a transformation between two 2D plannar structure, including warping between two plannar objects or warping between two images taken at the same camera center. Fundamental matrix models the generic relationship between any two images taken towards the same 3D world.

Problem 5: RANSAC

Estimating **F**

- If we don't know $\mathbf{K}_1$, $\mathbf{K}_2$, $\mathbf{R}$, or $\mathbf{t}$, can we estimate $\mathbf{F}$ for two images?

- Yes, given enough correspondences

Estimating F – 8-point algorithm

- The fundamental matrix F is defined by

$$\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x} = 0$$

for any pair of matches x and x' in two images.

- Let x=($u,v,1$)^{T} and x'=($u',v',1$)^{T},

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

each match gives a linear equation

$$uu' f_{11} + vu' f_{12} + u' f_{13} + uv' f_{21} + vv' f_{22} + v' f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

8-point algorithm

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

- In reality, instead of solving $\mathbf{Af} = 0$, we seek $\mathbf{f}$ to minimize $\|\mathbf{Af}\|$, least eigenvector of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$.

8-point algorithm

- Pros: it is linear, easy to implement and fast
- Cons: susceptible to noise

Problem 5: RANSAC

List the major steps of the RANSAC algorithm in the context of estimating a fundamental matrix from point-to-point correspondences between two images.

Problem 5: RANSAC

List the major steps of the RANSAC algorithm in the context of estimating a fundamental matrix from point-to-point correspondences between two images.

- For many times
  - Pick 8 points
  - Compute a solution for $\boldsymbol{F}$ using these 8 points
  - Count number of inliers that with $\boldsymbol{x}_1^T \boldsymbol{F} \boldsymbol{x}_2$ close to 0
- Pick the one with the largest number of inliers

Problem 6: Deep Learning

Gradient Descent

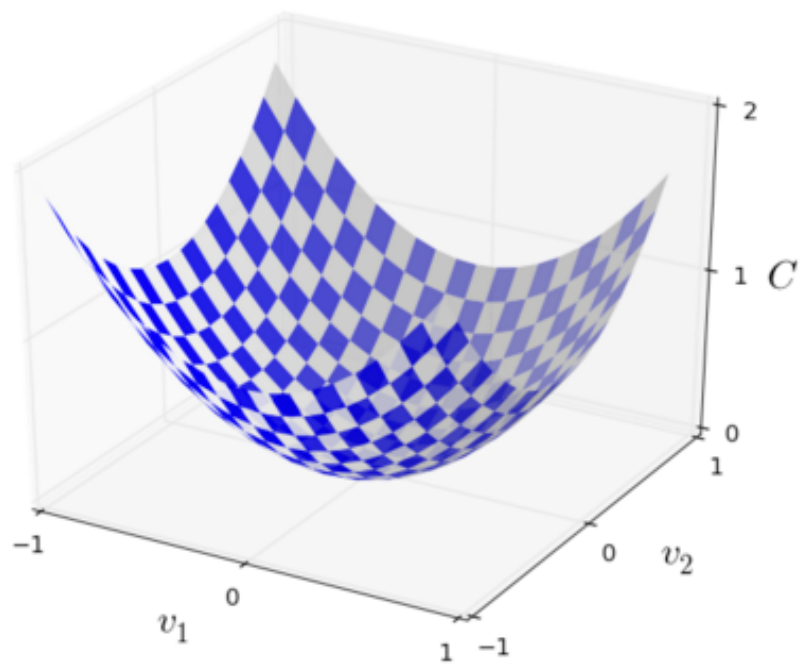
$$C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2.$$

$$\nabla C \equiv \left(\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2} \right)^T.$$

$$\Delta v = -\eta \nabla C,$$

$$w_k \rightarrow w_k' = w_k - \eta \frac{\partial C}{\partial w_k}$$

$$b_l \rightarrow b_l' = b_l - \eta \frac{\partial C}{\partial b_l}.$$



Stochastic Gradient Descent

$$w_k \rightarrow w'_k = w_k - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial w_k}$$

$$b_l \rightarrow b'_l = b_l - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial b_l},$$

Problem 6: Deep Learning

What is the stochastic gradient descent algorithm? Why do we need this algorithm for training deep convolutional neural network?

The stochastic gradient descent algorithm estimates gradient of the loss function by computing it for a small sample of randomly chosen training inputs. By averaging over this small sample we can quickly get a good estimate of the true gradient, thereby speeding up gradient descent, and thus learning. This algorithm is important in training deep convolutional neural network in which case the number of training inputs is very large because it helps speed up the learning.

See you on Thursday!