MAEG 5720: Computer Vision in Practice

Lecture 14b:

Course Summary

Dr. Terry Chang

2021-2022

Semester 1

香港中文大學
The Chinese University of Hong Kong

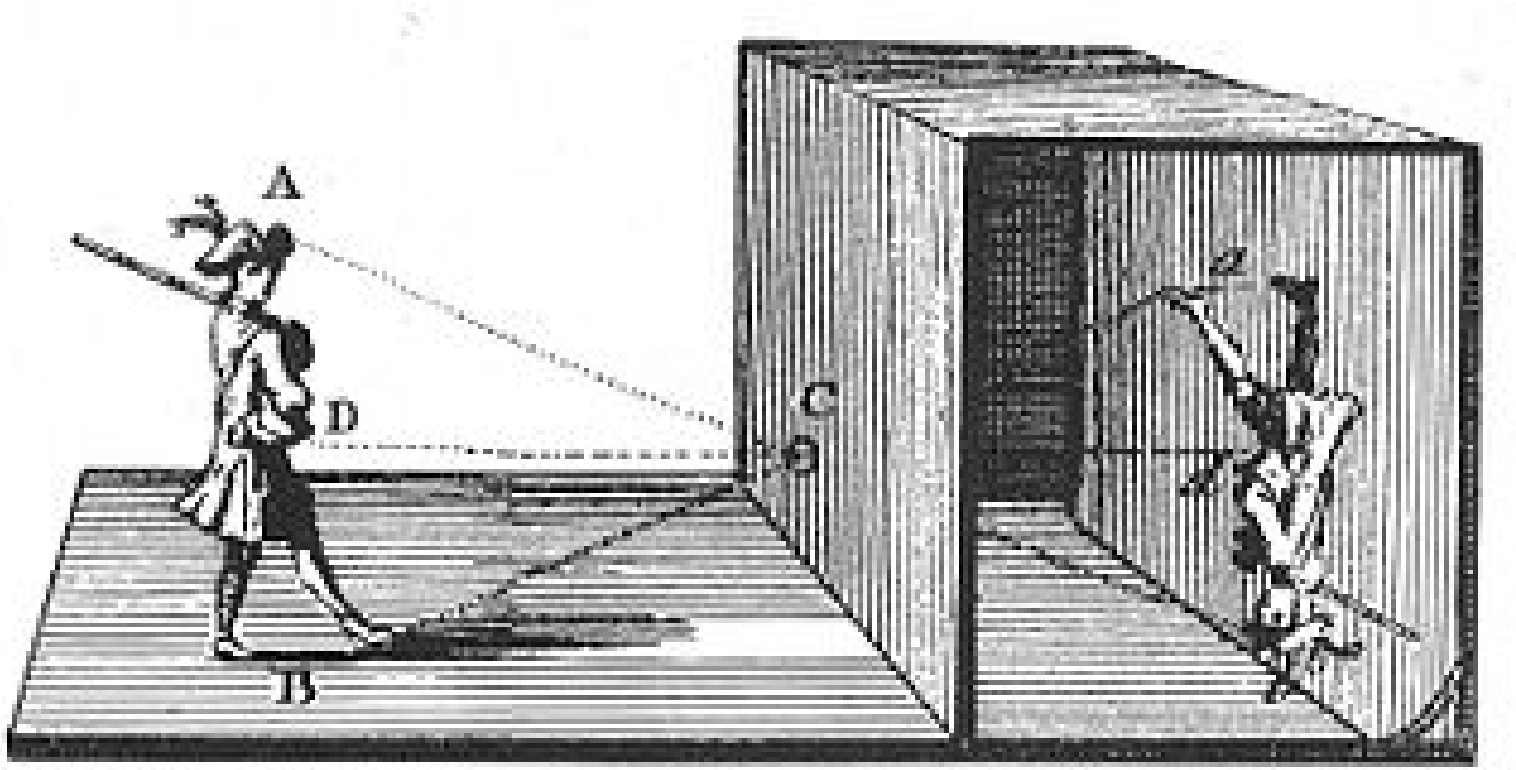
Department of Mechanical and Automation Engineering
機械與自動化工程學系

Class Schedule

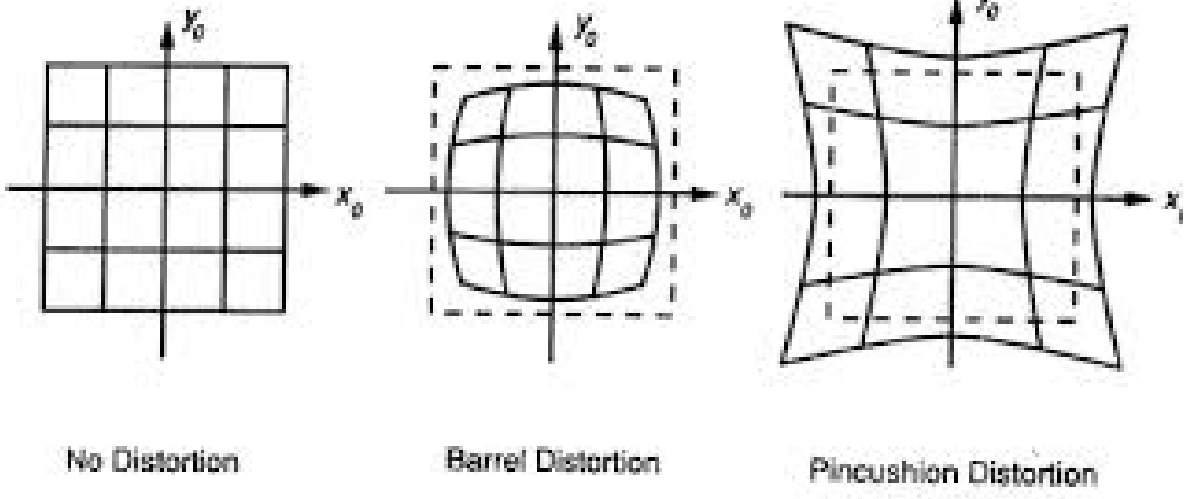
Class	Date	Topic
1	Friday, 10 September 2021	Course Introduction
2	Friday, 17 September 2021	Camera Model and image formation
3	Friday, 24 September 2021	Image filtering and template matching
	Friday, 1 October 2021	National Day Holiday
4	Friday, 8 October 2021	Edge and Interest point detection
5	Friday, 15 October 2021	Feature Descriptor(SIFT/SURF,etc)
6	Friday, 22 October 2021	Image Segementation
7	Friday, 29 October 2021	Optical Flow and Tracking
8	Friday, 5 November 2021	Projective Geometry in 2D
9	Friday, 12 November 2021	Transformation & Projective Geometry in 3D
10	Friday, 19 November 2021	Homography and Image Stitching
11	Friday, 26 November 2021	Camera Parameters and Calibration
12	Friday, 3 December 2021	More on Single view geometry
13	Friday, 10 December 2021	Epipolar Geometry and Stereo Vision
14	Friday, 17 December 2021	More on 3D Reconstruction

Image Formation

- **Camera obscura** means dark room *camera* chamber also refer to a Pinhole Camera

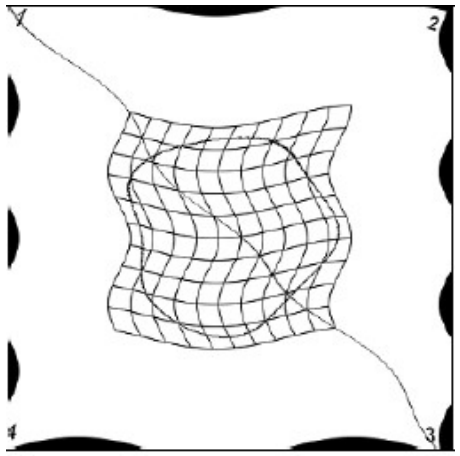


Effect of camera

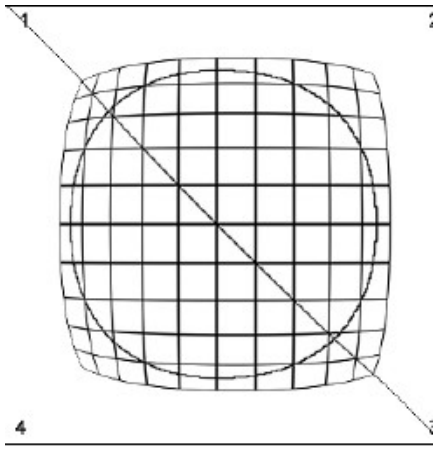


Basic Image Transformation

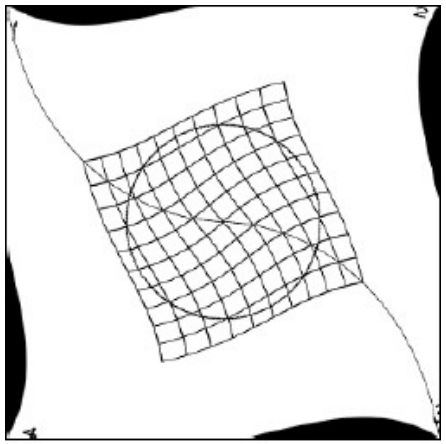
Other transformation



(b)



(c)



(a)

(e)

(f)

(d)

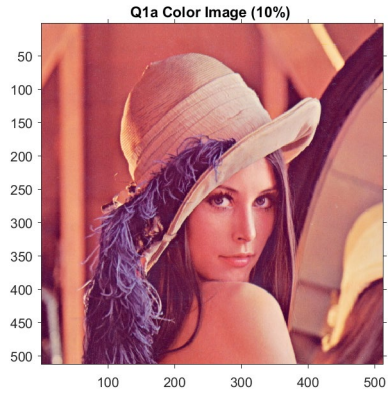
Image Credit: E. Agu. WPI

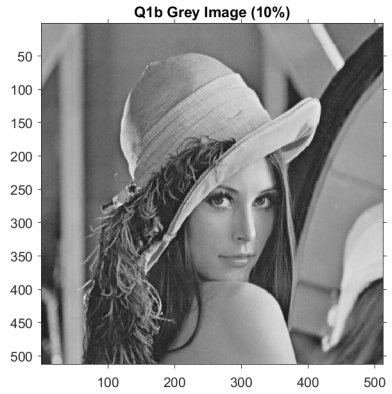
Image Warping

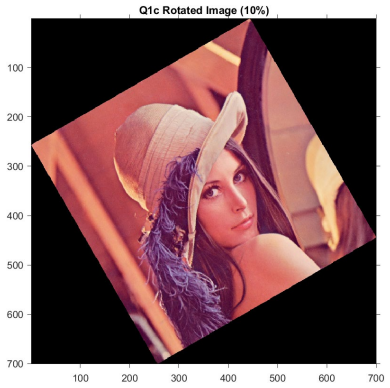
Source: http://medium .com

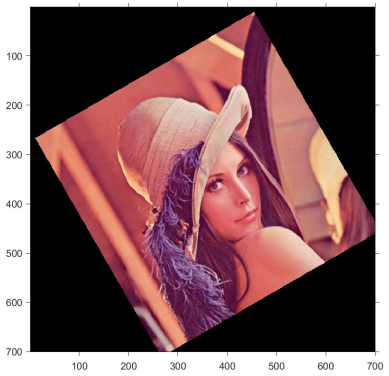
Assignment 1

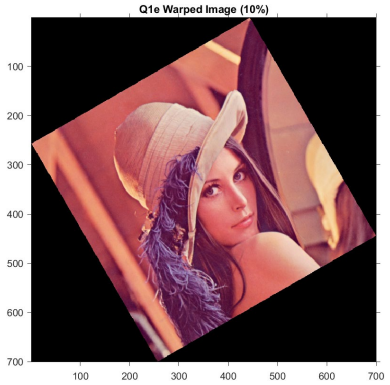
- Aim: The aim of this assignment is to warm you up for MATLAB and get prepared for the coming assignment. Hence, the assignment will be simple and easy.

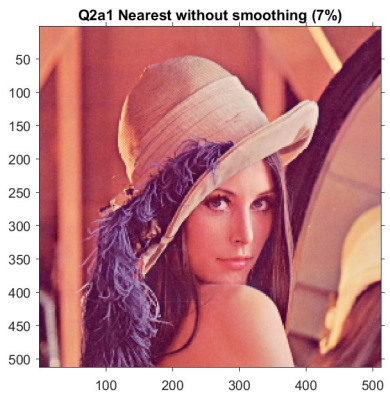


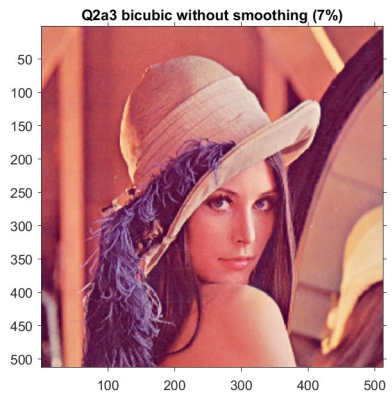


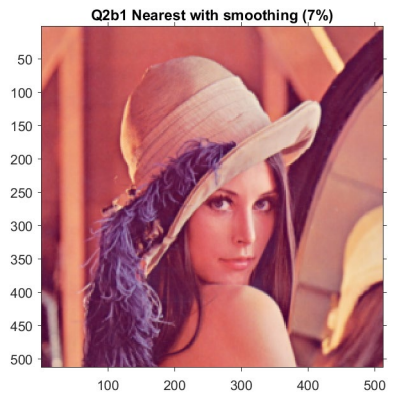


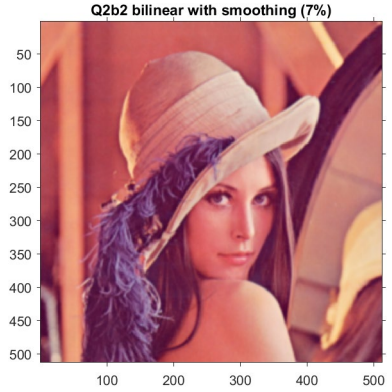












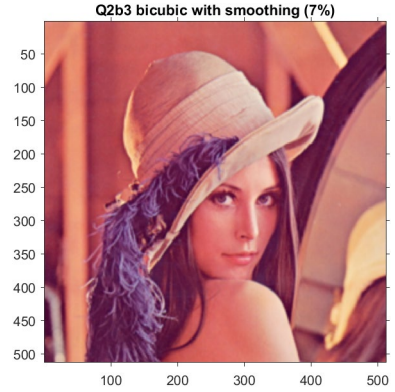
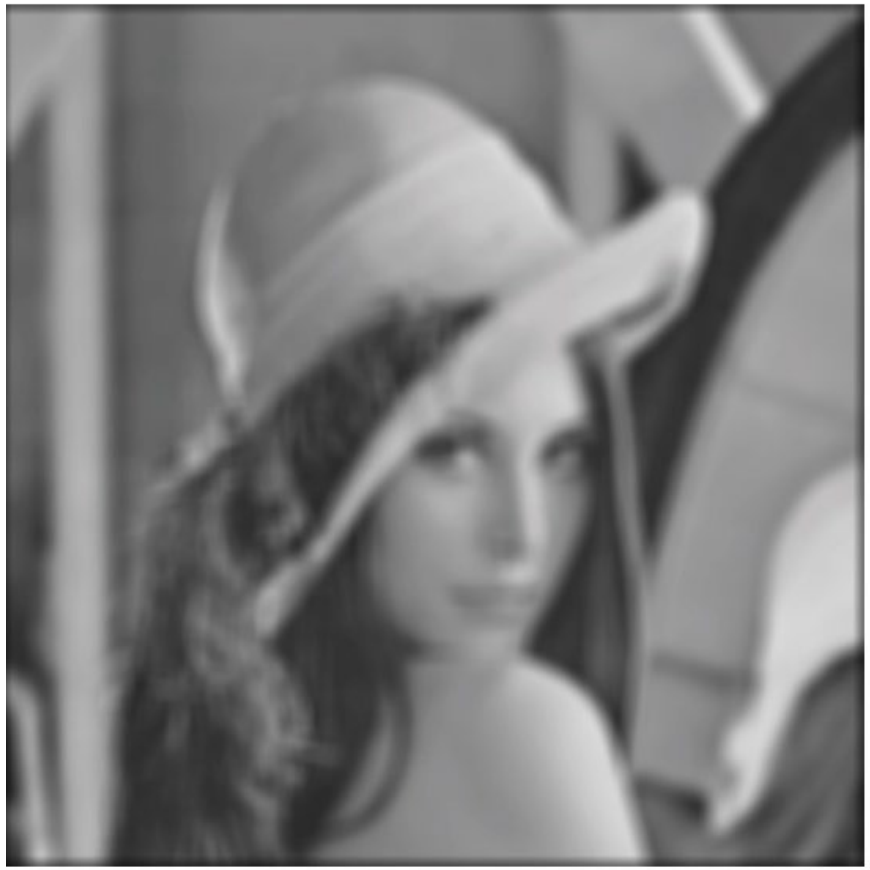


Image Filtering (Smoothing)

Original



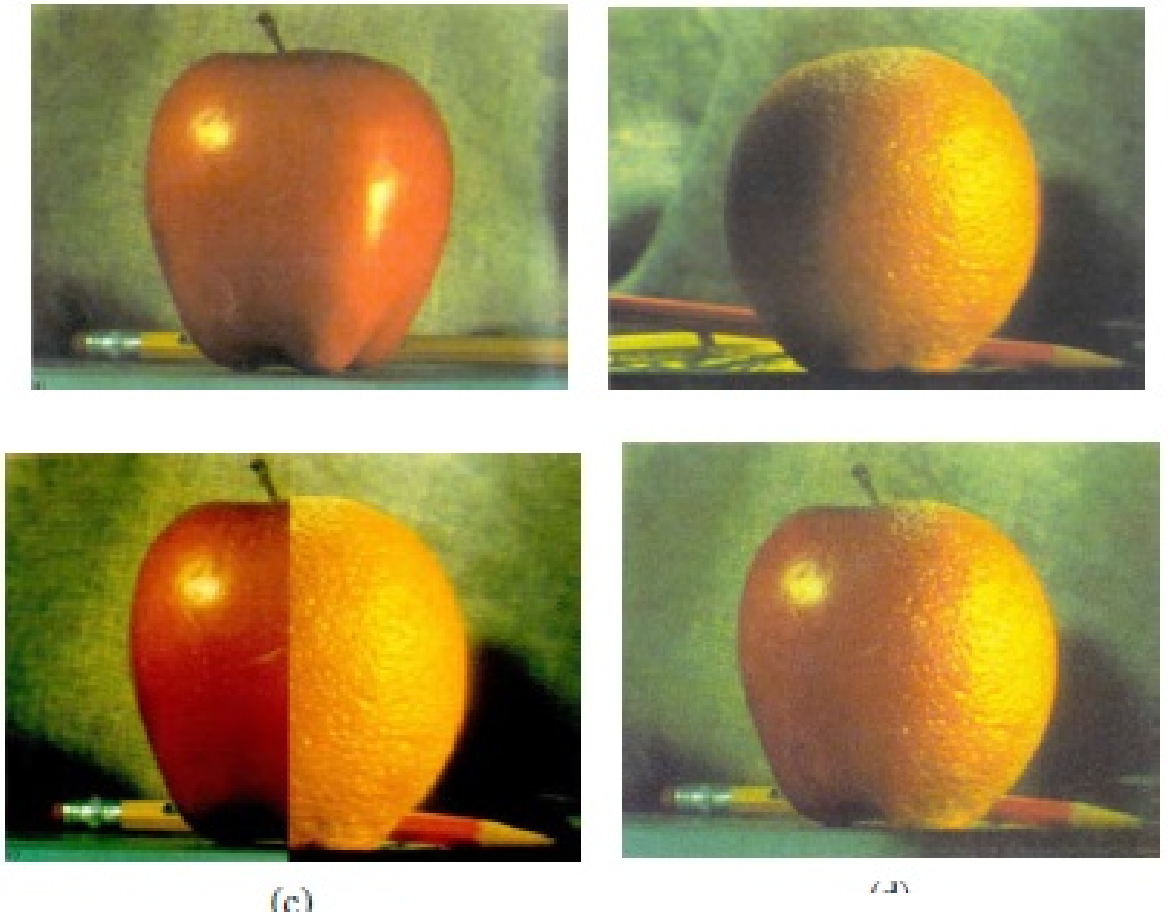
Sharpened

Image Filtering (Sharpening)

Original

Sharpened

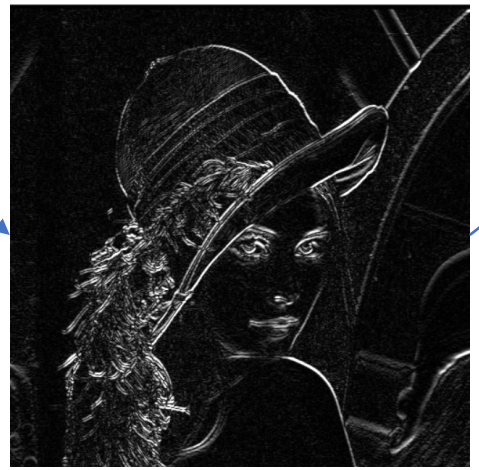
Image Bending



Edge Detection

Image I

Derivative in X

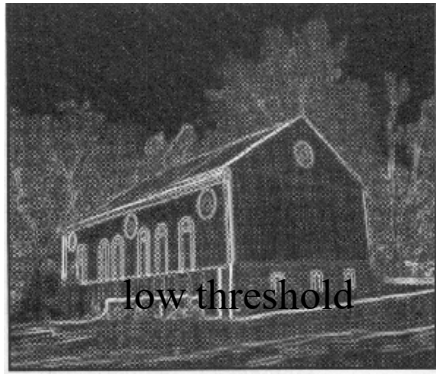


Sobel Edge Detector

Mag>threshold = 0

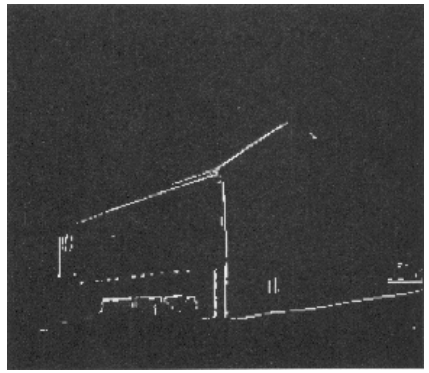
Effect the threshold?

Image



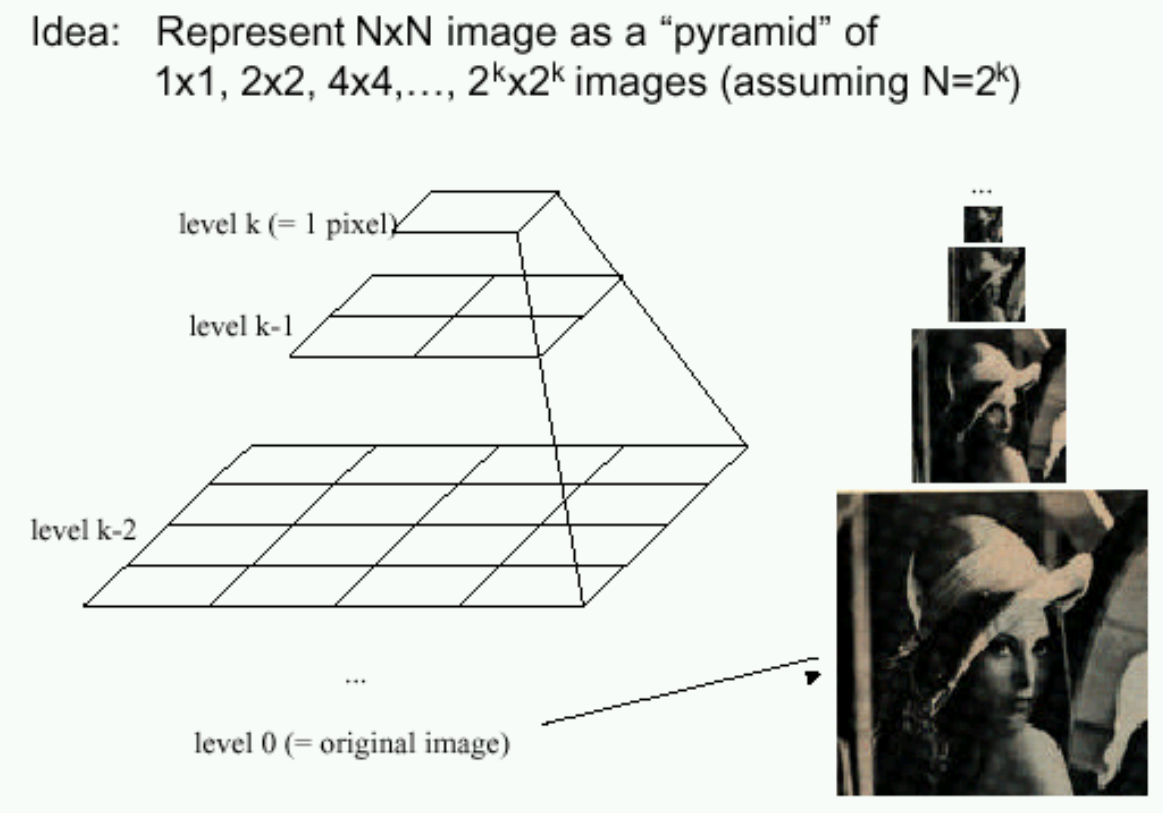
Gradient Mag

Low Threshold



High Threshold

Image Pyramids



- In computer graphics, a *mip map* [Williams, 1983]

Source: S. Seitz

Assignment 2

- Aim: Practice on Image filtering by convolution method

$$average\_filter = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

1/16

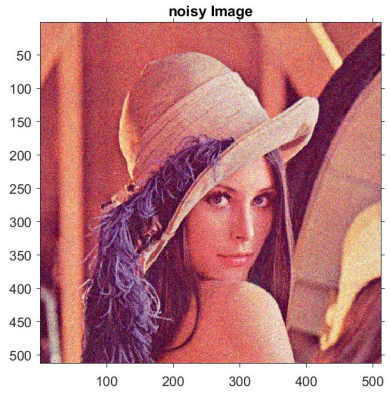
1	2	1
2	4	2
1	2	1

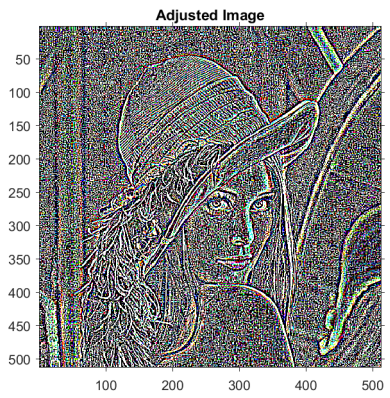
1/273

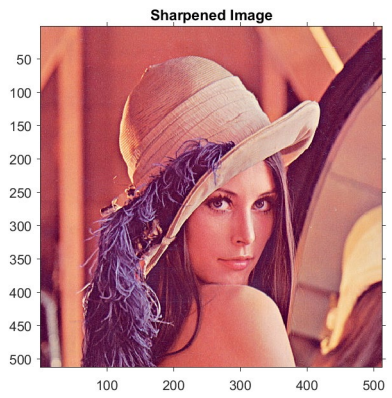
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

1/1003

0	0	1	2	1	0	0
0	3	13	22	13	3	0
1	13	59	97	59	13	1
2	22	97	159	97	22	2
1	13	59	97	59	13	1
0	3	13	22	13	3	0
0	0	1	2	1	0	0



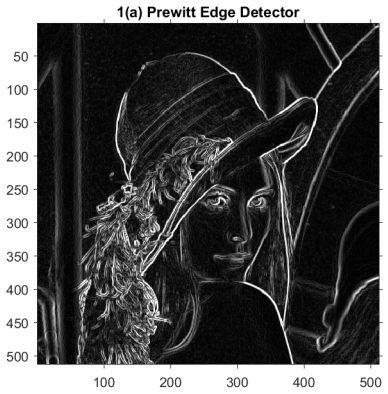


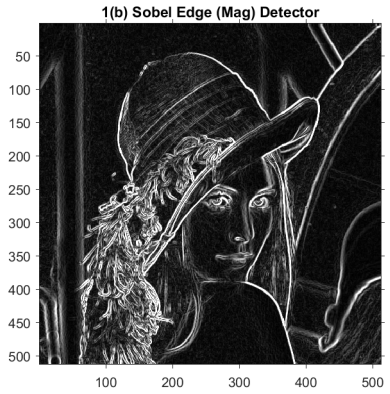


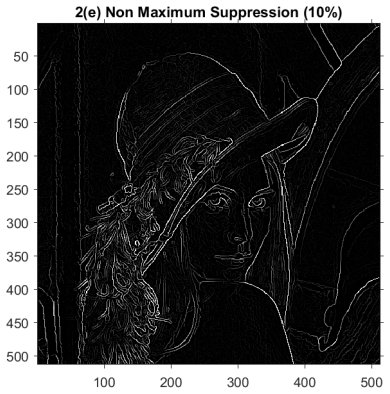
Assignment 2

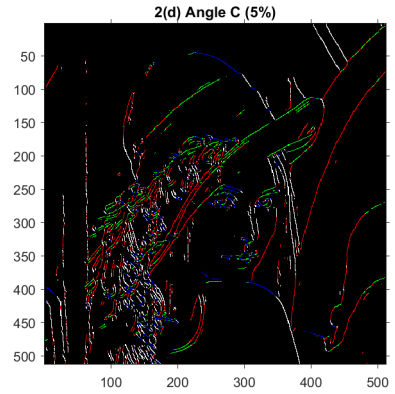
Prewitt Filter $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ $M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

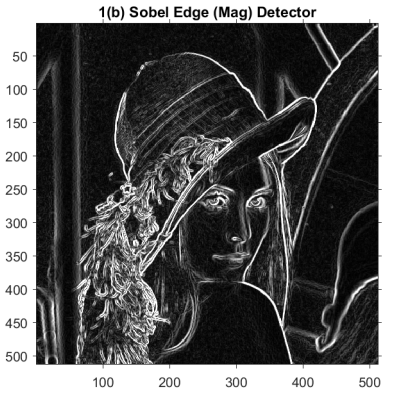
Sobel Filter $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ $M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$









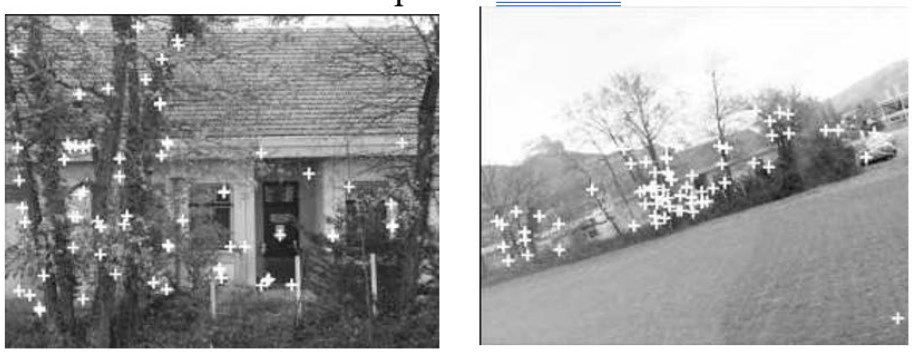


Interest point detection

- What is interest point?
- The use of interest point
  - Autonomous vehicle navigation
  - Image matching and retrieval

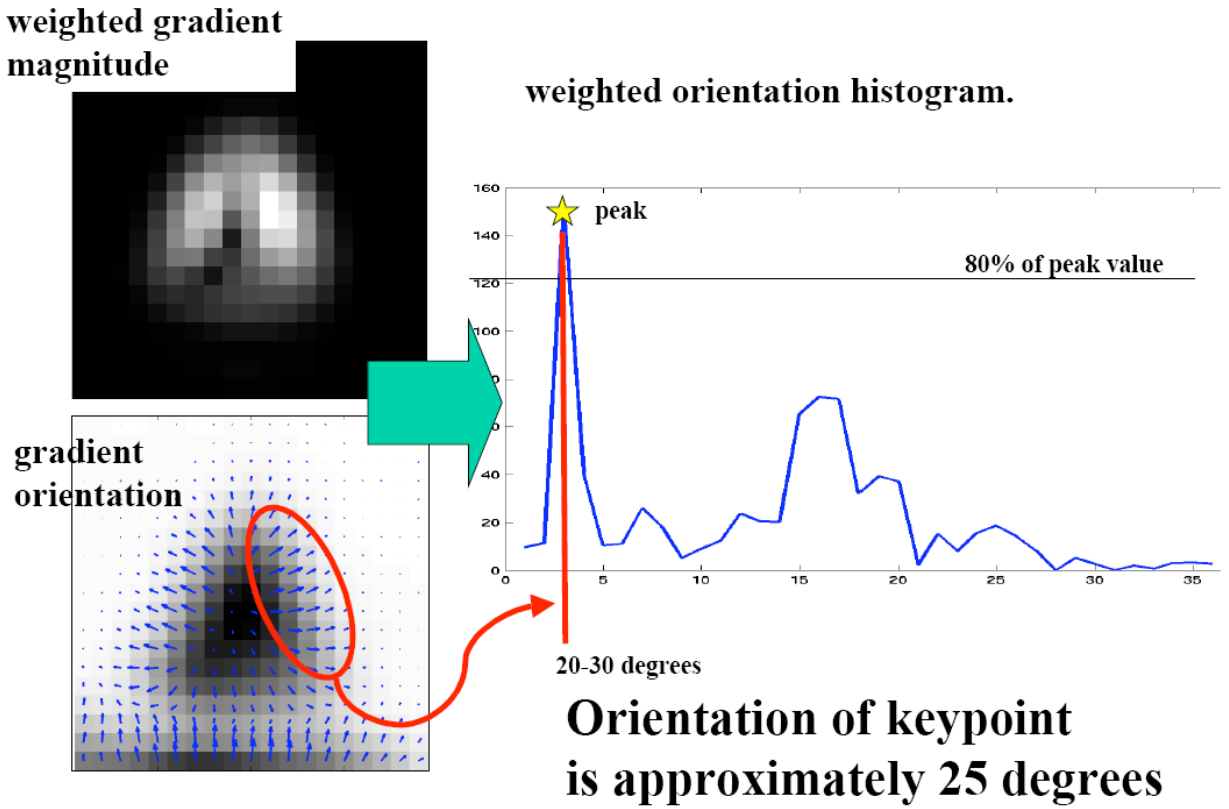
Extracted interest points

Initial points matches

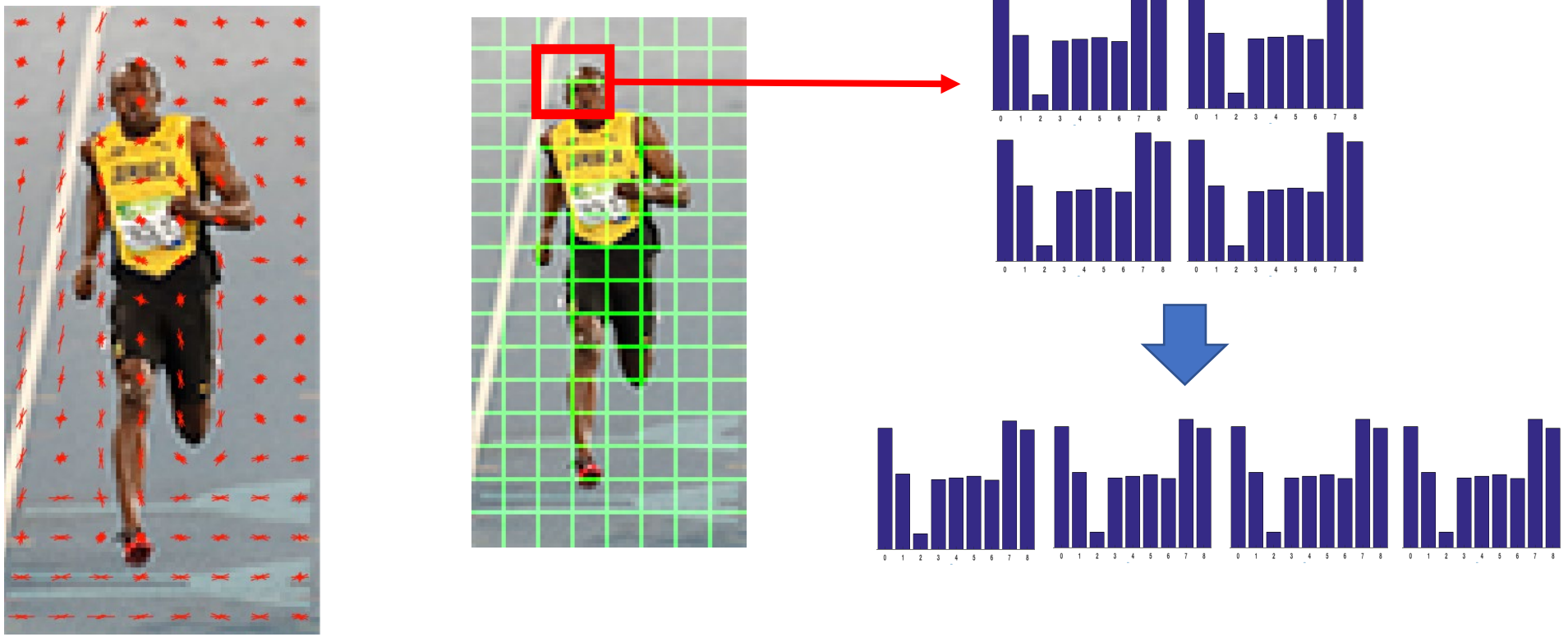


Inliers to the estimated homography

Feature Descriptors (SIFT/SURF/ORB/HOG)



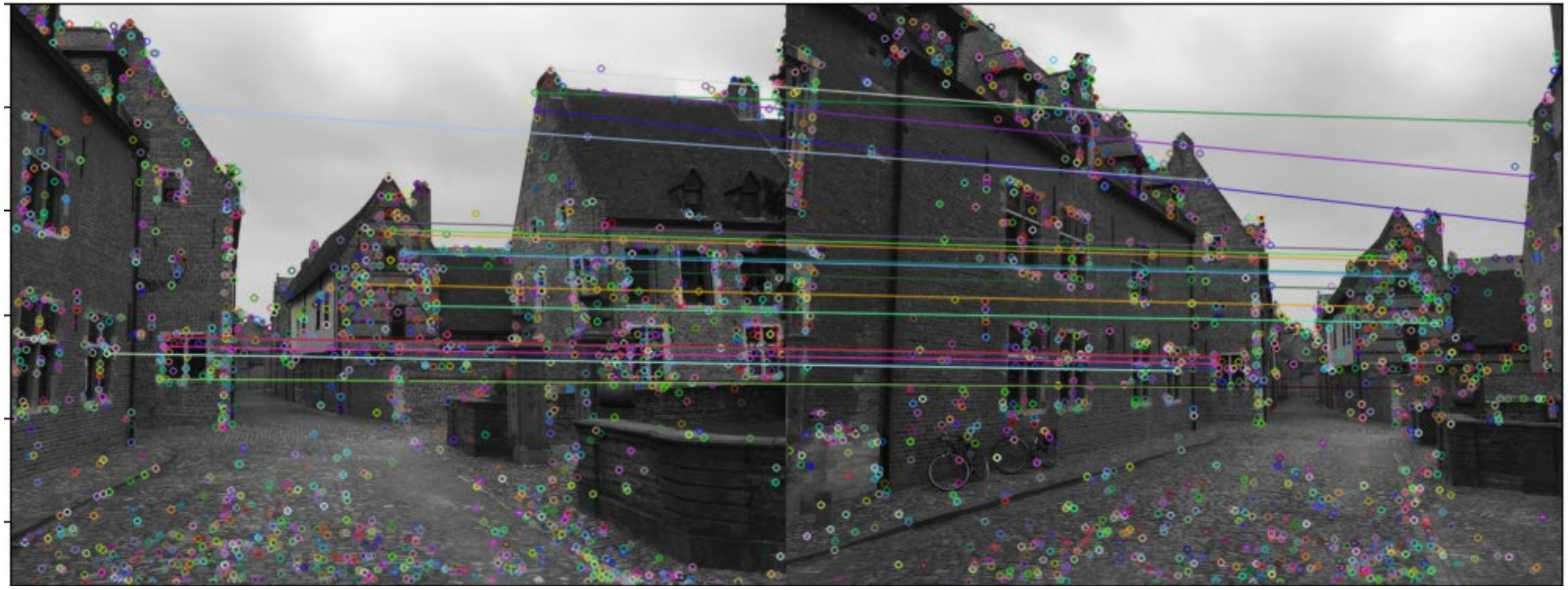
Histogram of Oriented Gradient



Use of Shape Descriptor

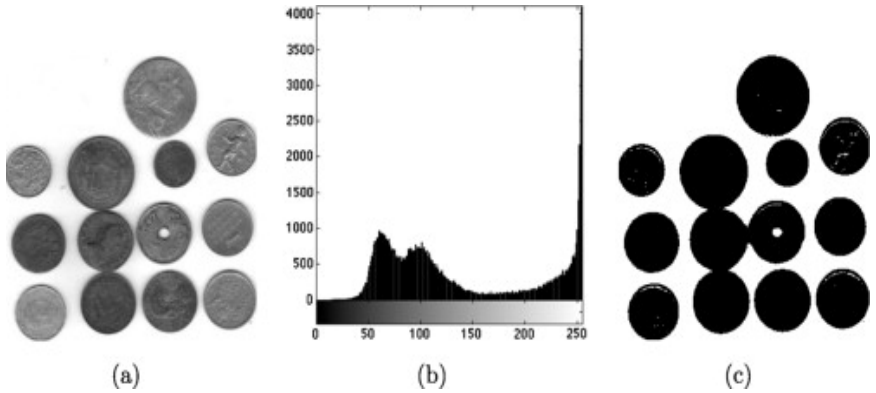
e.g. scale, translation, rotation

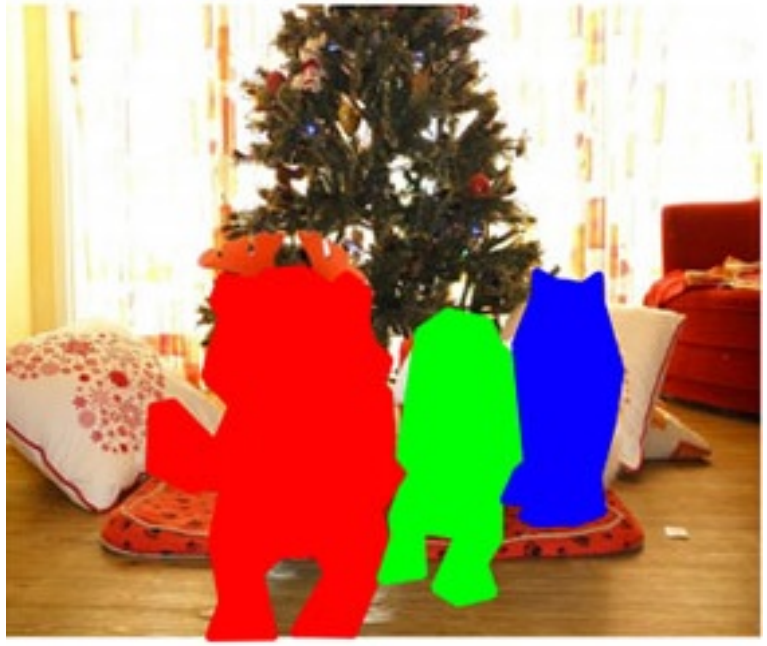
Image matching



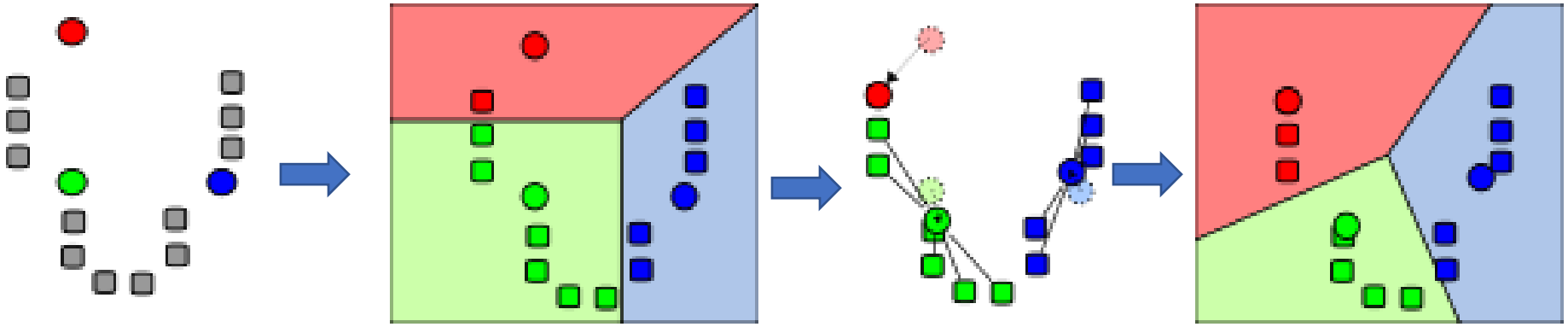
Pedestrian Detection

Image Segmentation





K-means Clustering



1. Initialize Cluster Centers

2. Assign Points to Clusters

3. Re-compute Means

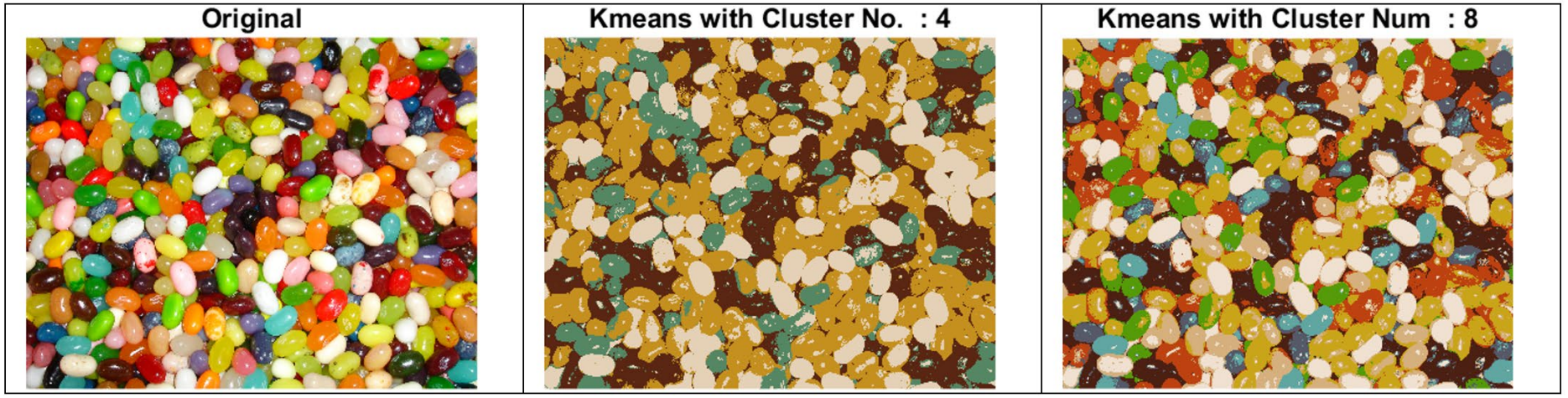
Repeat (2) and (3)

[(118) K-Means Clustering Example - YouTube]

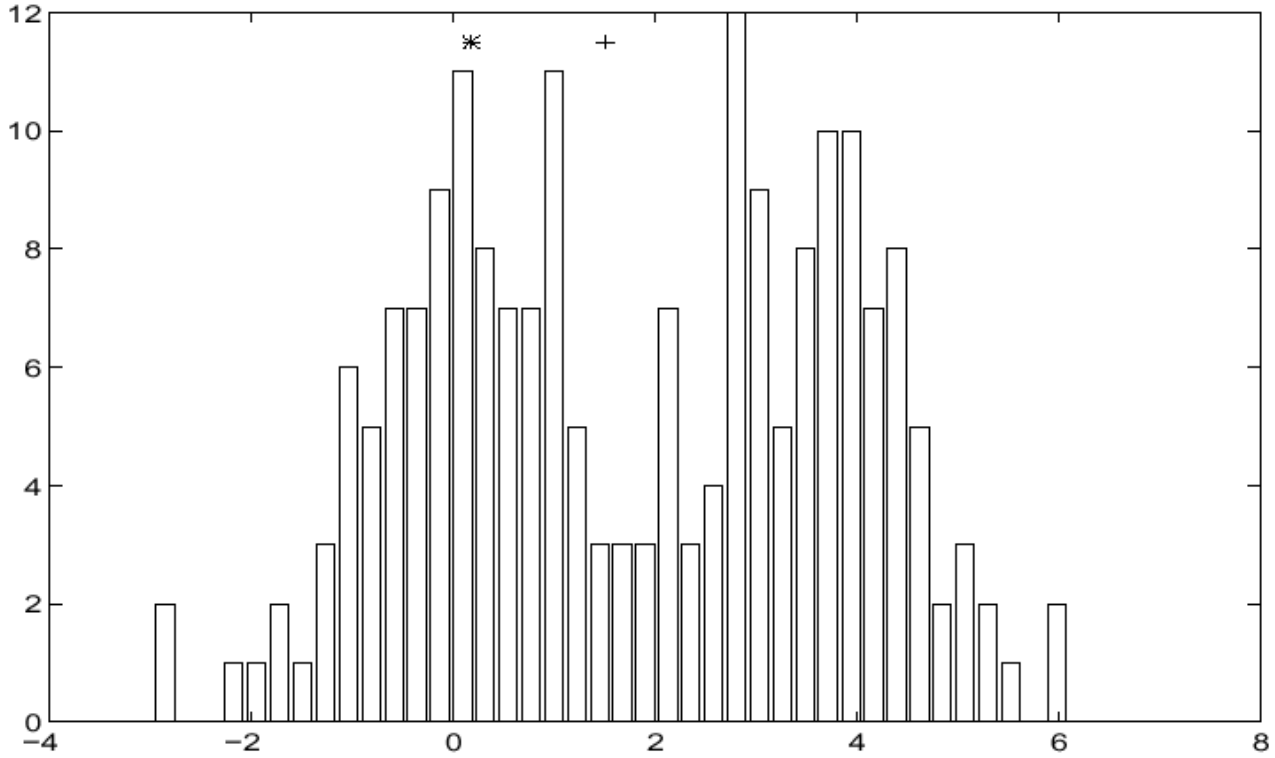
Picture Credit: J. Carlos

Assignment 3

- Aim: Images usually consist of many different objects. To better understand the image, we need to separate the objects for further processing. The process group pixels into different clusters with similar features, such as colours or spatial distance. This process is known as segmentation.



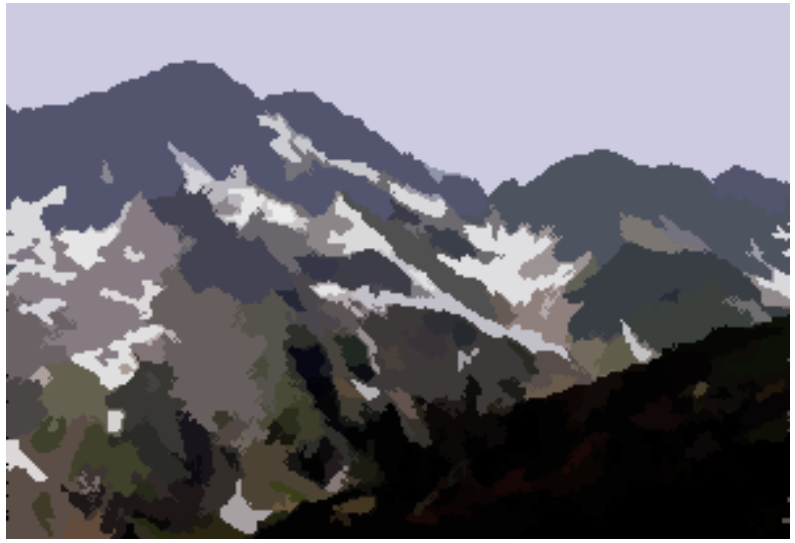
Mean-Shift Algorithm



- Iterative Mode Search
  1. Initialize random seed, and window W
  2. Calculate center of gravity (the "mean") of W: $\sum_{x \in W} xH(x)$
  3. Shift the search window to the mean
  4. Repeat Step 2 until convergence

Slide credit: Steve Seitz

Mean-Shift Segmentation Results



Optical Flow (Horn & Schunck)

- Compute Derivatives

```
fx = conv2(im1,[-1 1; -1 1], 'valid'); % partial on x
fy = conv2(im1, [-1 -1; 1 1], 'valid'); % partial on y
ft = conv2(im1, ones(2), 'valid') + conv2(im2, -ones(2), 'valid'); % partial on t
```

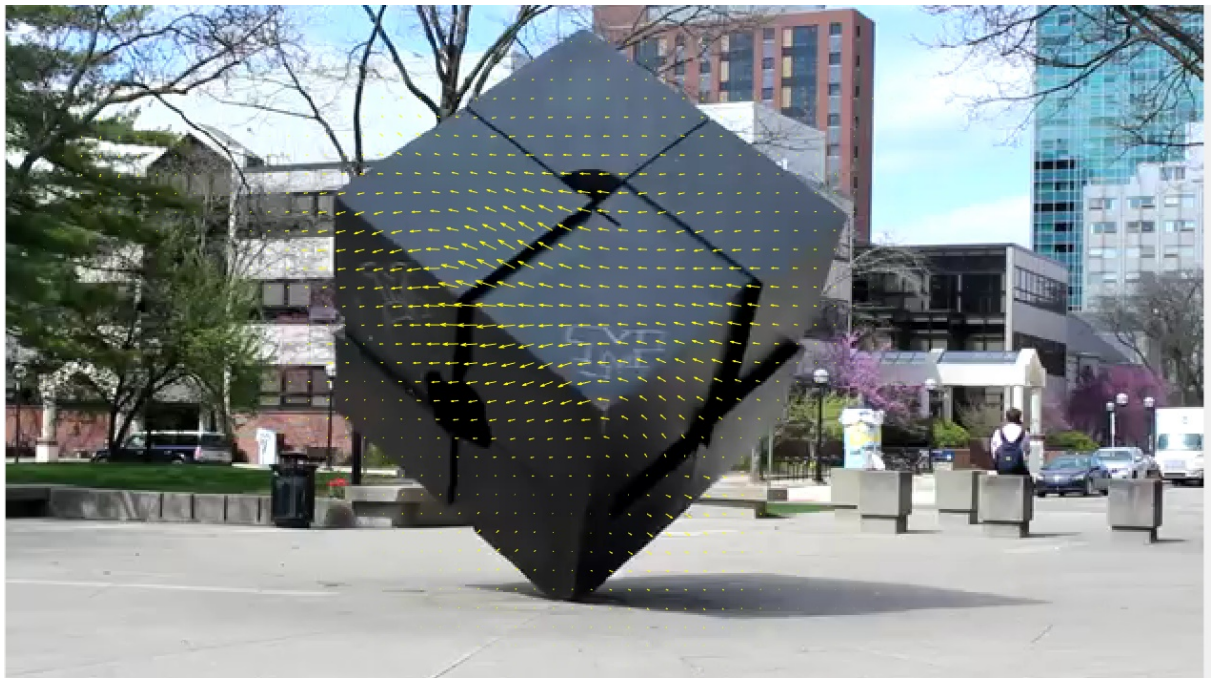
- Laplacian Mask and averaging

```
kernel_1=[0 1/4 0;1/4 0 1/4;0 1/4 0];

% or

kernel_1=[1/12 1/6 1/12;1/6 0 1/6;1/12 1/6 1/12];
```

```
uAvg=conv2(u,kernel_1,'same');
vAvg=conv2(v,kernel_1,'same');
```

- Iteration

```
alpha = (fx.*uAvg+fy.*vAvg+ft)./(1+lamda*(fx.^2+fy.^2));
u = uAvg-alpha.*fx;
v = vAvg-alpha.*fy;
```



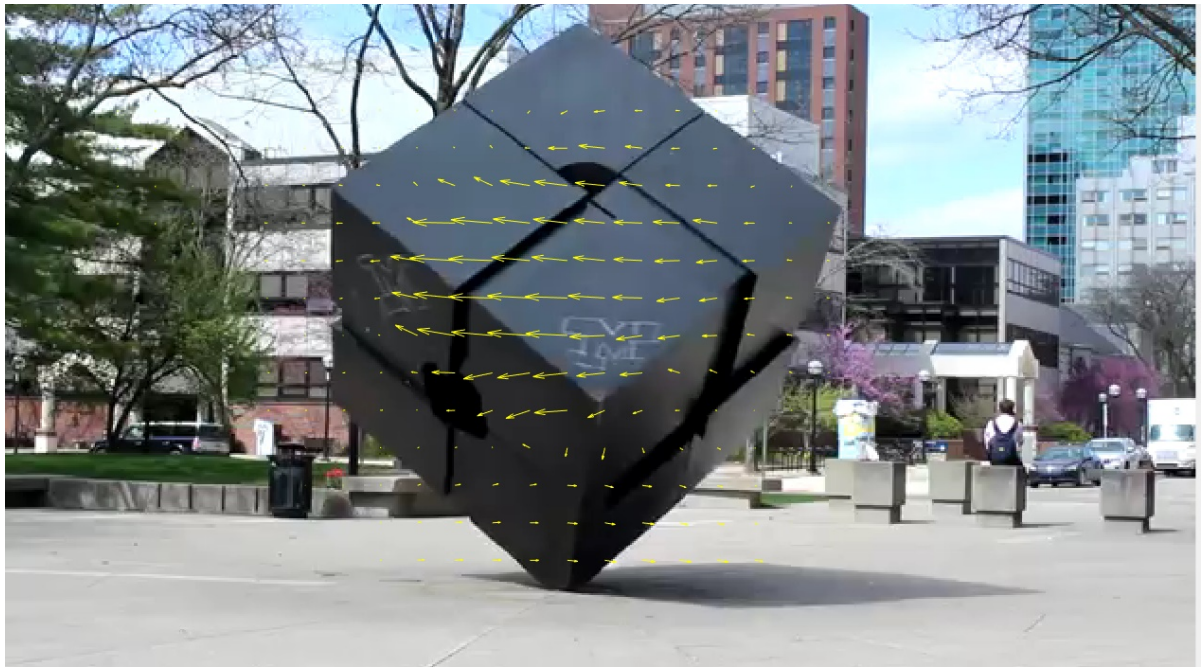
Lucas Kanade Matlab Example

- Compute Derivatives

```
fx = conv2(im1,[-1 1; -1 1], 'valid'); % partial on x
fy = conv2(im1, [-1 -1; 1 1], 'valid'); % partial on y
ft = conv2(im1, ones(2), 'valid') + conv2(im2, -ones(2), 'valid'); % partial on t
```

- Define the local windows

```
for i = w+1:size(Ix_m,1)-w
   for j = w+1:size(Ix_m,2)-w
      % Copy Ix, Iy, It to the windows
      Ix = Ix_m(i-w:i+w, j-w:j+w);
      Iy = Iy_m(i-w:i+w, j-w:j+w);
      It = It_m(i-w:i+w, j-w:j+w);

      Ix = Ix(:);
      Iy = Iy(:);
      b = -It(:); % This defines b

      A = [Ix Iy]; % This defines a
      vel = pinv(A)*b; % vel = pesudo inverse(a)*b

      u(i,j)=Vel(1);
      v(i,j)=Vel(2);
   end;
end;
```



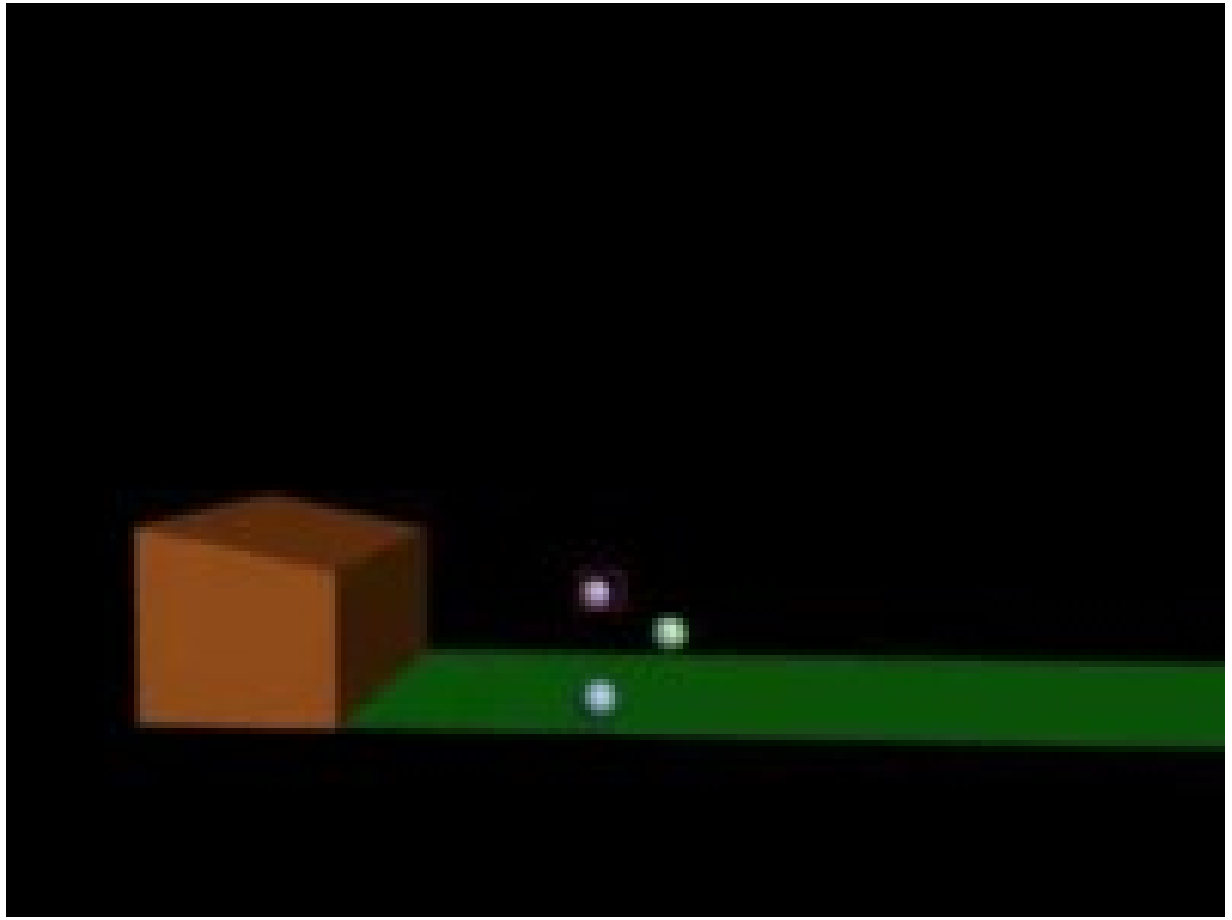
Tracking Objects (KLT, Mean Shift)

- Key assumptions:
  - **Brightness constancy:** projection of the same point looks the same in every frame (uses SSD as metric)
  - **Small motion:** points do not move very far (from guessed location)
  - **Spatial coherence:** points move in some coherent way (according to some parametric motion model)
- For this example, assume whole object just translates in (u,v)

Template T

Image Frame I

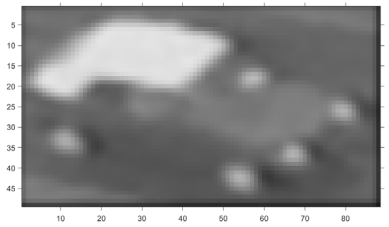
Tracking (Mean-Shift)



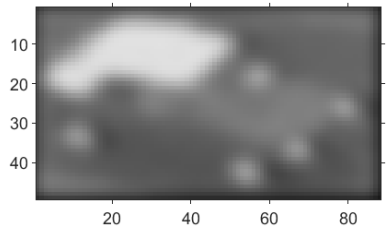
Mini-Project 1

Aim: To familiarize with template tracking and detecting moving object in a video stream

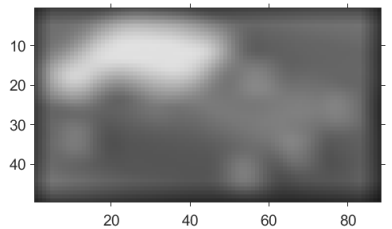
Sigma=1



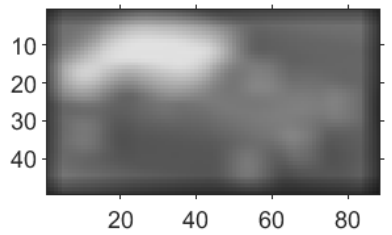
Sigma=2



Sigma=5



Sigma=10

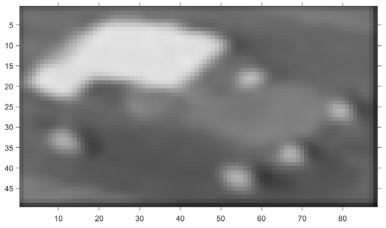


Sigma Value	Number of iteration	Track till the end?
1	797	Y
2	567	Y
3	482	Y
4	457	Y
5	452	Y
6	449	Y
7	443	Y
8	446	Y
9	441	Y
10	443	Y
Max iteration = 100		
Threshold = 0.01		
Gaussian Filter Size = 10x10		

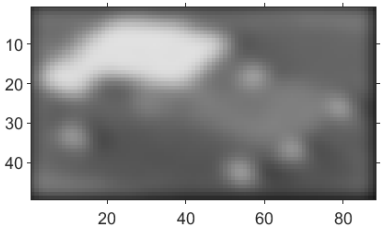
Mini-Project 1 (Effect of sigma, max_it = 6)

Filter size =10x10

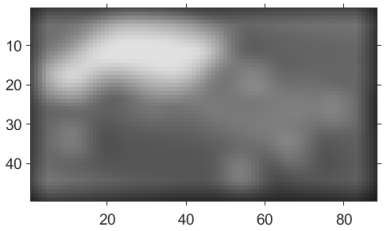
Sigma=1



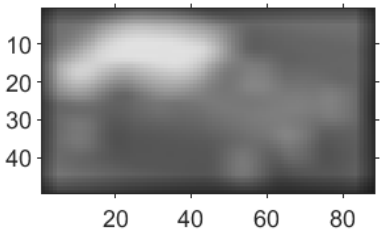
Sigma=2



Sigma=5



Sigma=10



Sigma Value	Number of iteration	Track till the end?
1	797	Y
2	567	Y
3	482	Y
4	457	Y
5	452	Y
6	449	Y
7	443	Y
8	446	Y
9	441	Y
10	443	Y
Max iteration = 100		
Threshold = 0.01		
Gaussian Filter Size = 10x10		

Sigma Value	Number of iteration	Track till the end?
1	567	N
2	481	Y
3	434	Y
4	420	Y
5	418	Y
6	416	Y
7	411	Y
8	413	Y
9	409	Y
10	412	Y
Max iteration = 6		
Threshold = 0.01		
Gaussian Filter Size = 10x10		

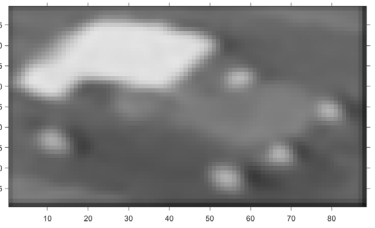
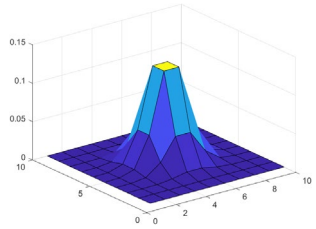
For limited number of iteration, only sigma =1 will lose track.
The difference between 3<=Sigma<=10 not obvious!

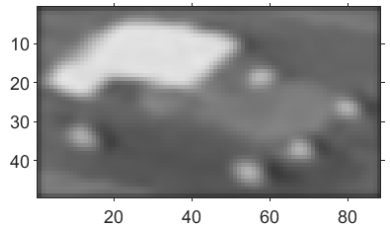
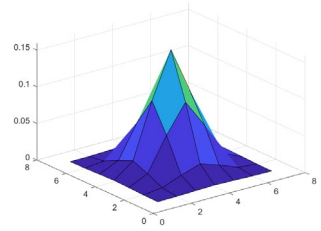
Mini-Project 1 (Effect of Sigma-Opt. filter size)

Filter size =10x10

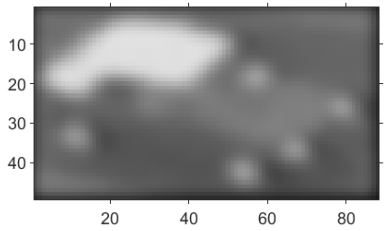
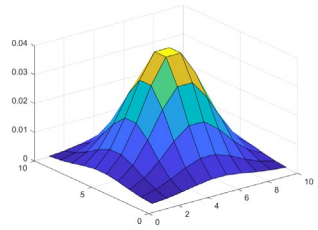
Filter size =6sigma

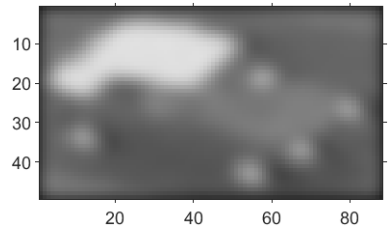
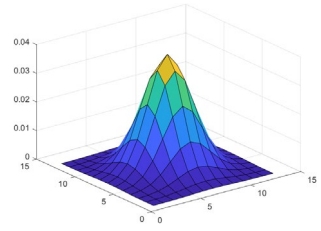
Sigma=1

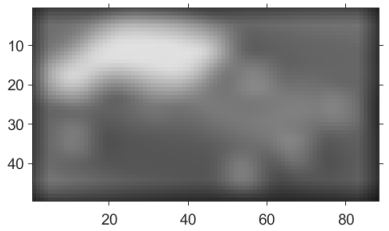
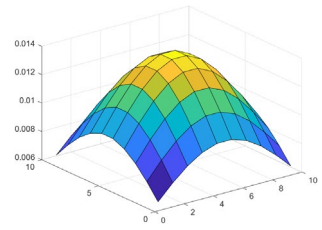
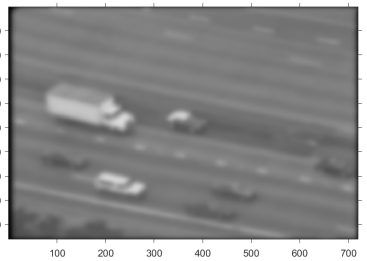
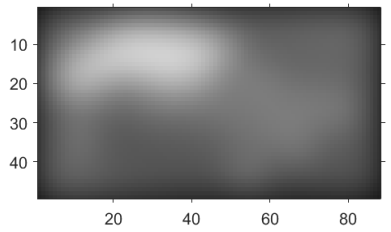
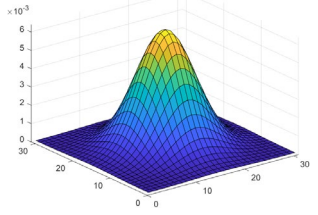



Sigma=2

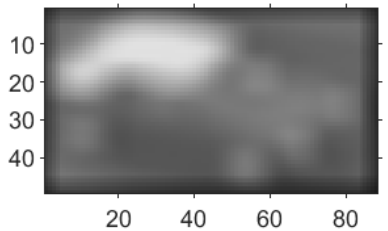
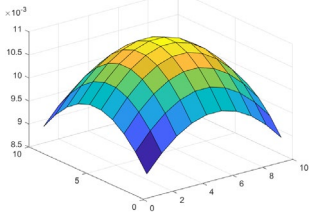
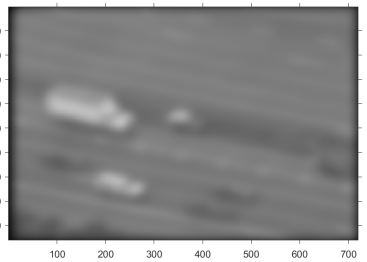
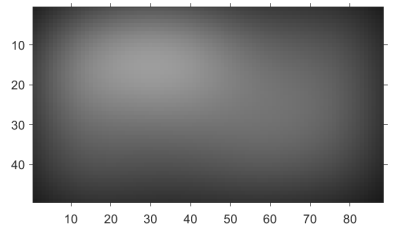
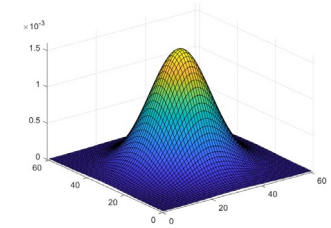




Sigma=5

Sigma=10

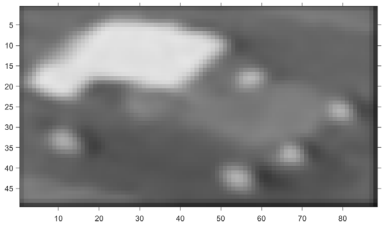







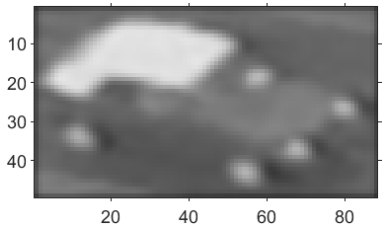
Mini-Project 1 (Effect of Sigma-Opt. filter size)

Filter size =10x10

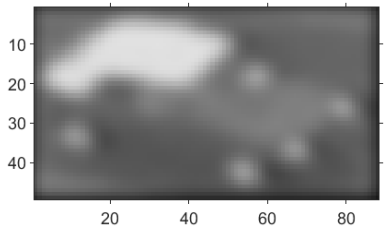
Filter size =6sigma

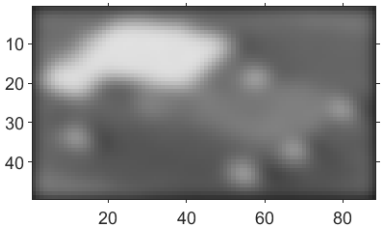
Sigma=1

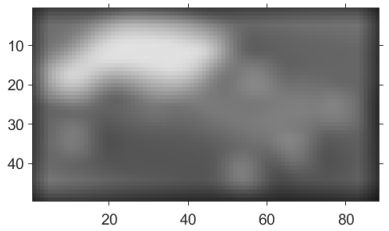
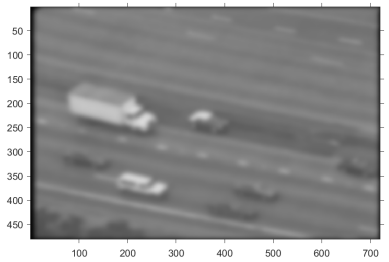
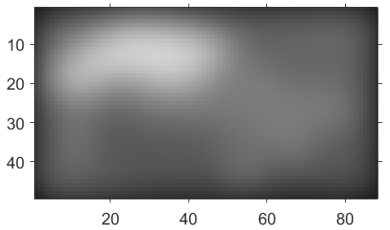


Sigma=2

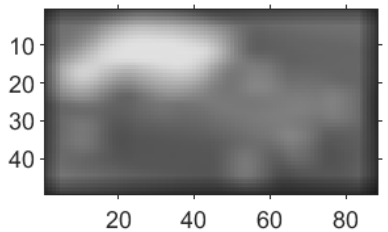
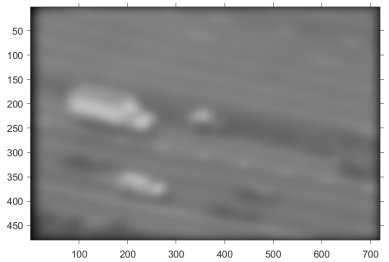
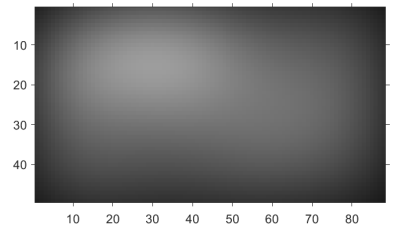





Sigma=5

Sigma=10

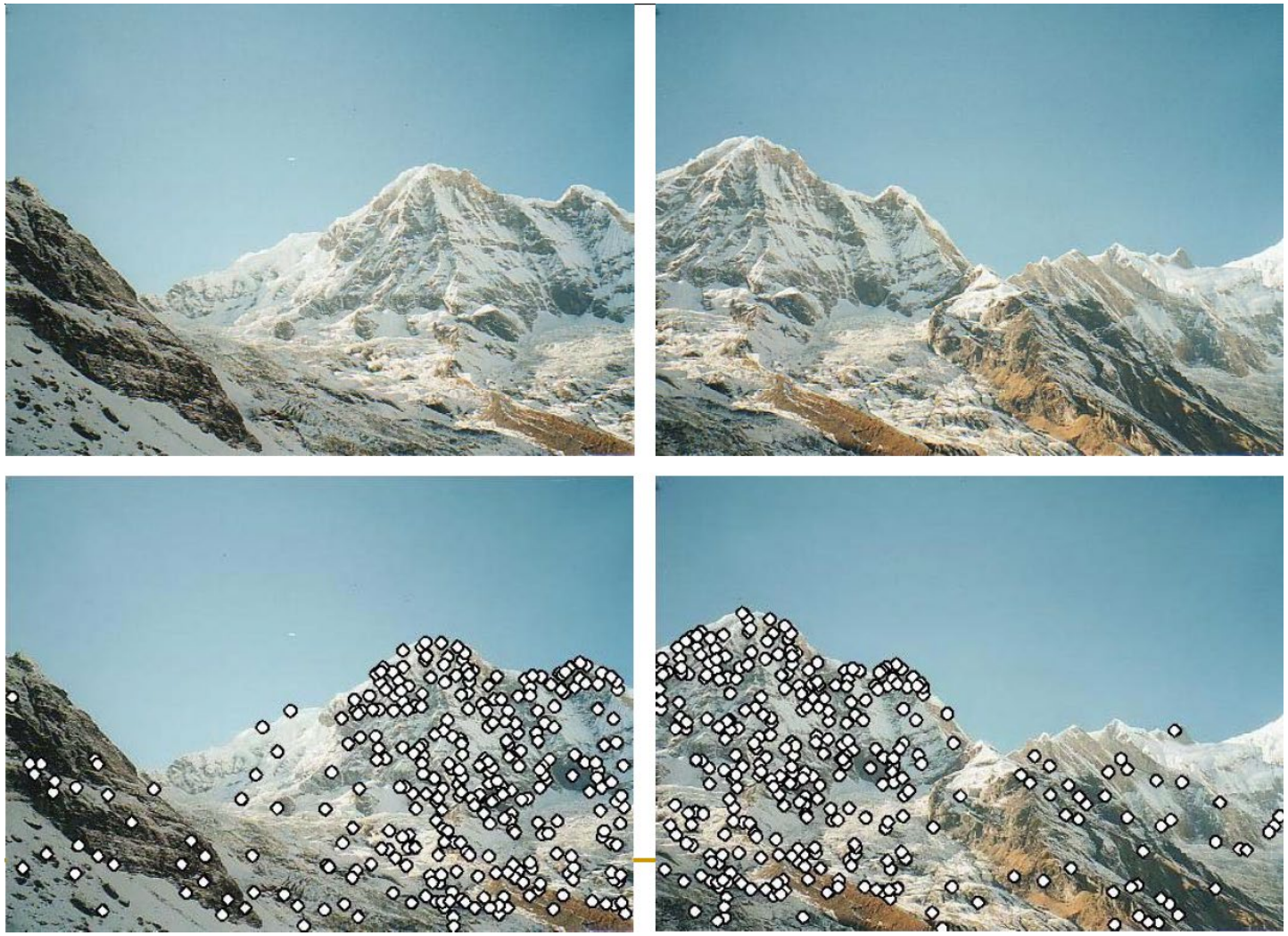





Sigma Value	Number of iteration	Track till the end?
1	783	Y
2	546	Y
3	444	Y
4	437	Y
5	528	Y
6	1460	N
7	1579	N
8	1793	N
9	2636	N
10	7613	N
Max iteration = 100		
Threshold = 0.01		
Gaussian Filter Size = 6 sigma+1		

For sigma value <=2, it has Num_iteration is large

For sigma value >6, lose track!

Image Stitching with SIFT and RANSAC



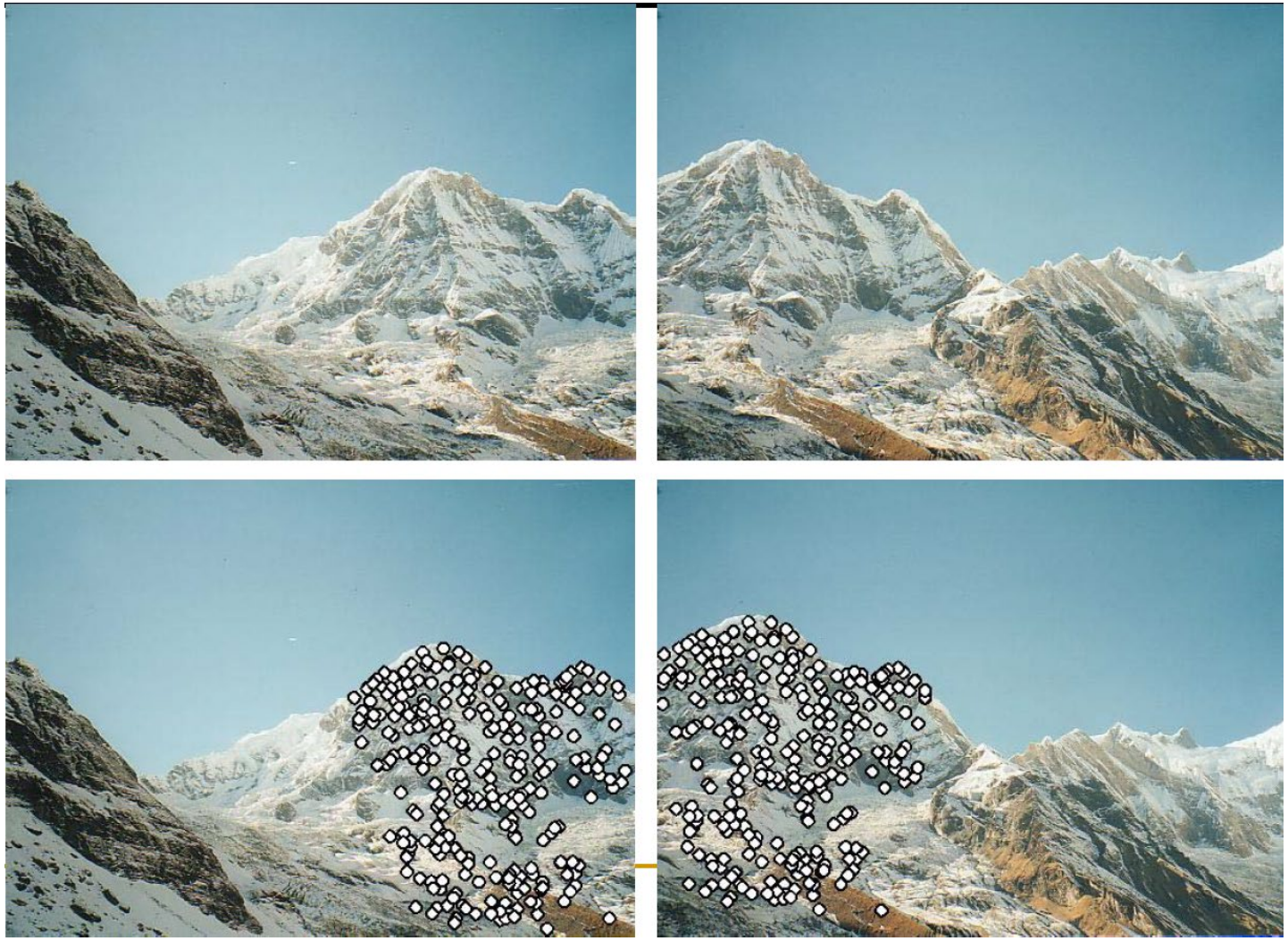
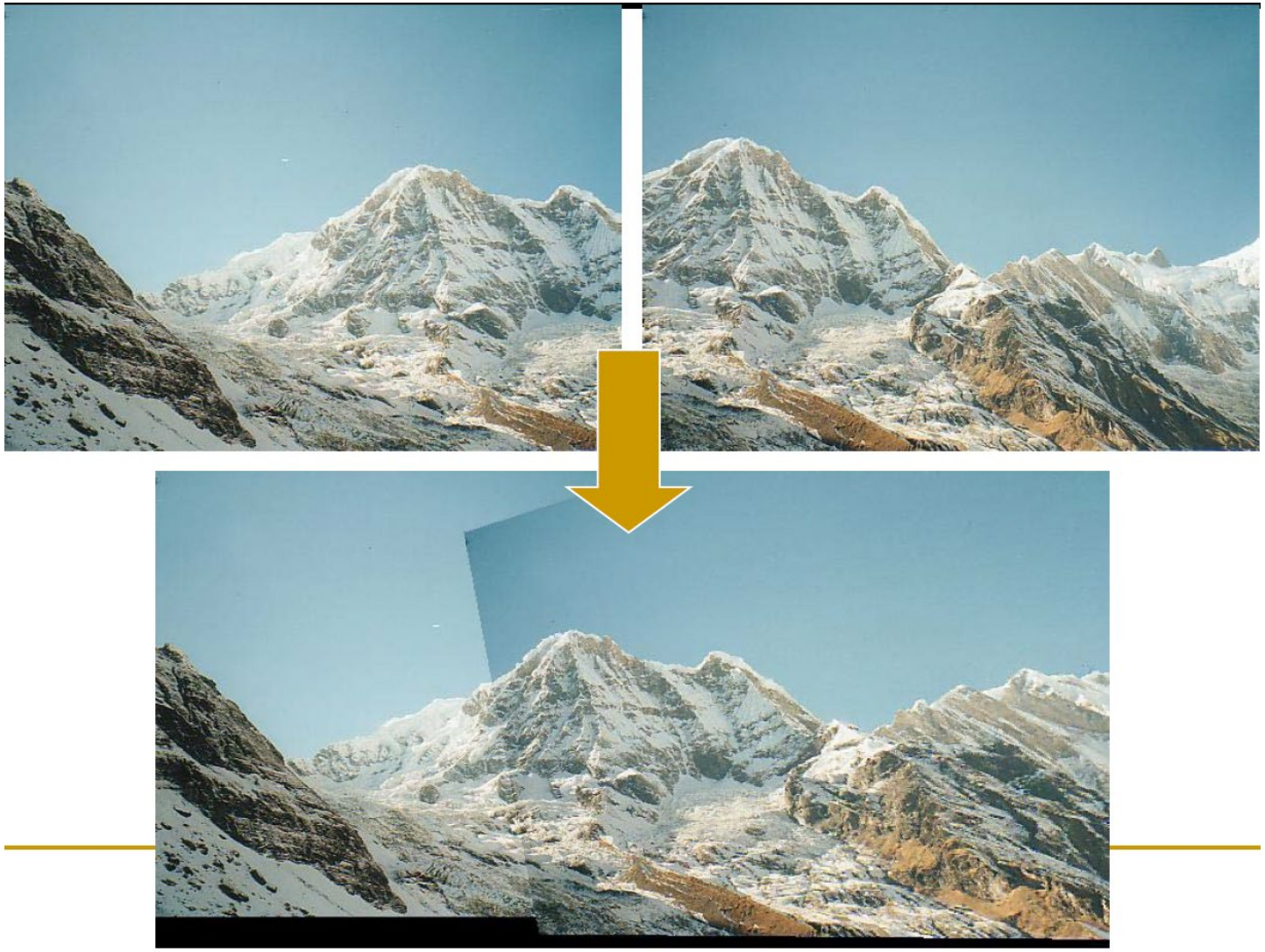
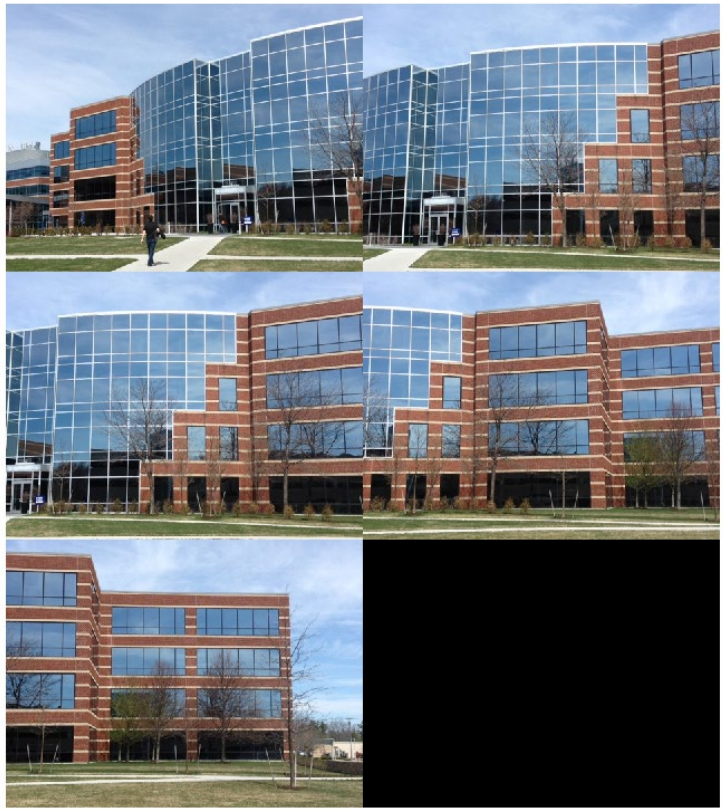
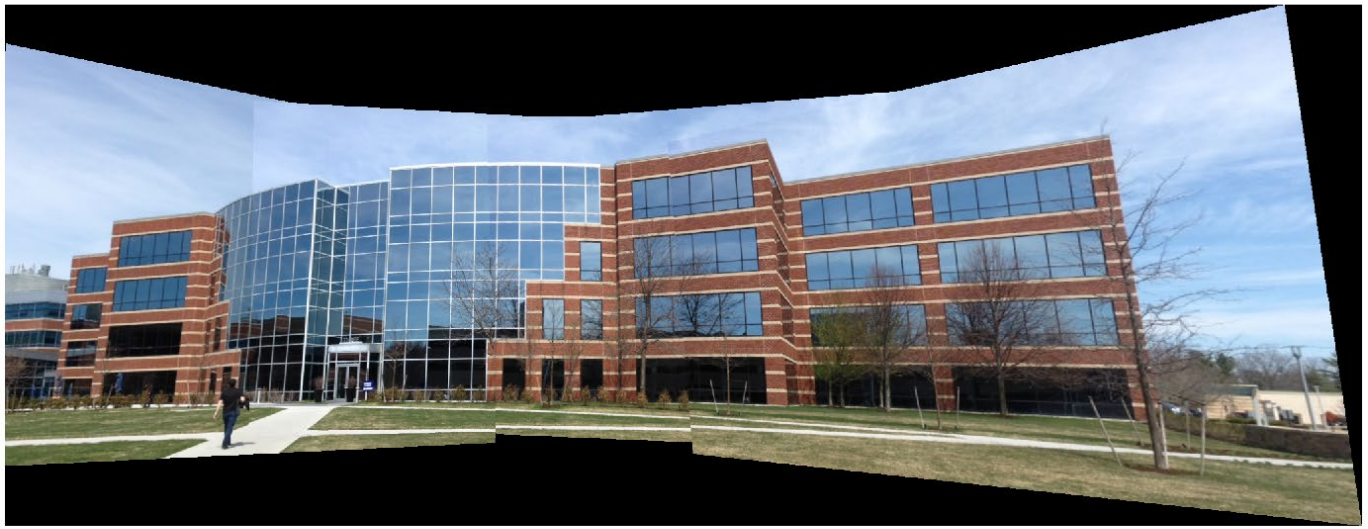


Image Stitching

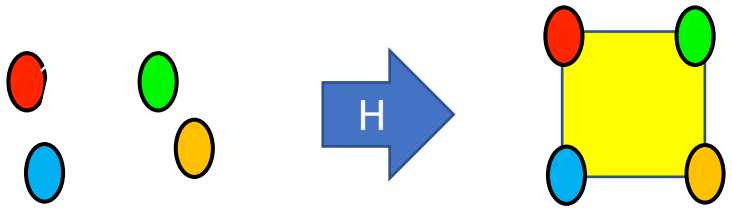


Homography and Image Stitching





Homography and Image Stitching



Original

rectangle

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + h_9}$$

$$y' = \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + h_9}$$

$$\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix}$$

$$\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix}$$

$$\vdots$$

$$\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix}$$

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Solving $\boldsymbol{Ah} = 0$

How?

SVD

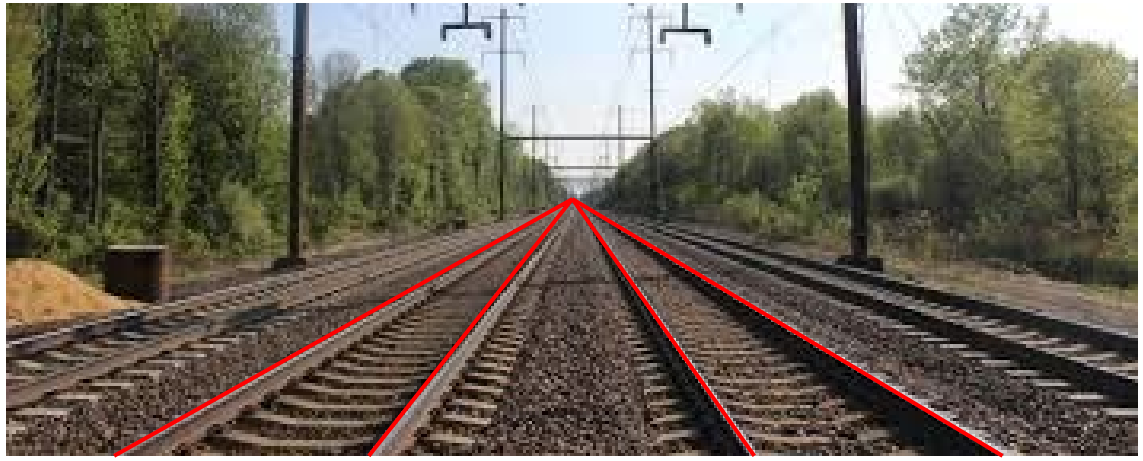
Mini-Project 2

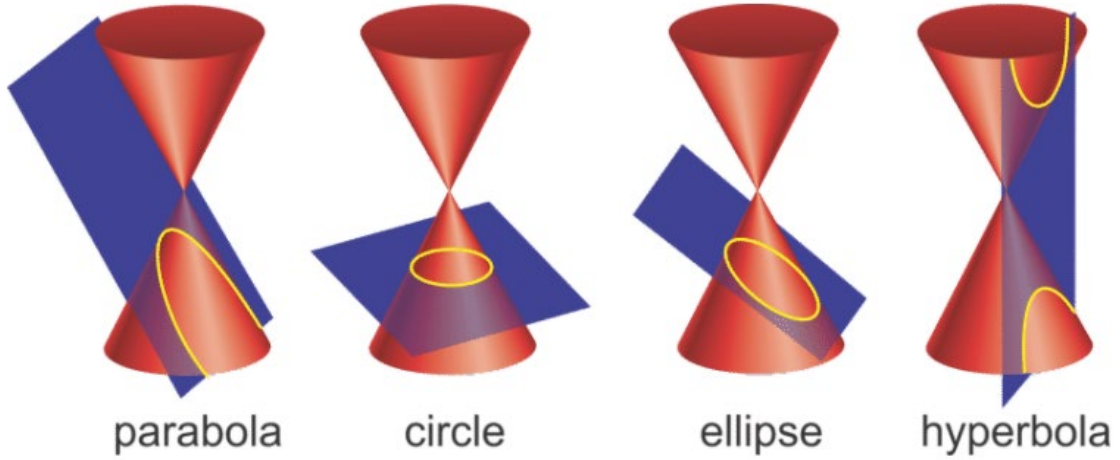
- **Aim:** The field of view of the image is limited by your lens. However, we could solve this issue by combining multiple images together to form a panorama. The aim of this assignment is to automatically stitch the images acquired by a panning camera.

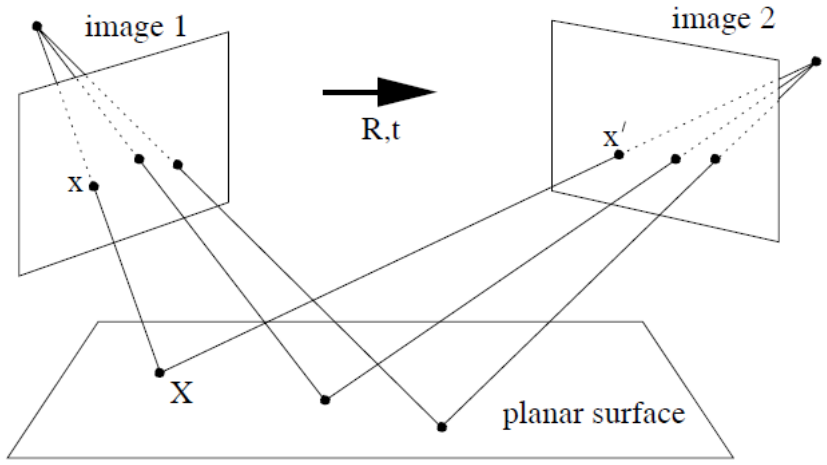


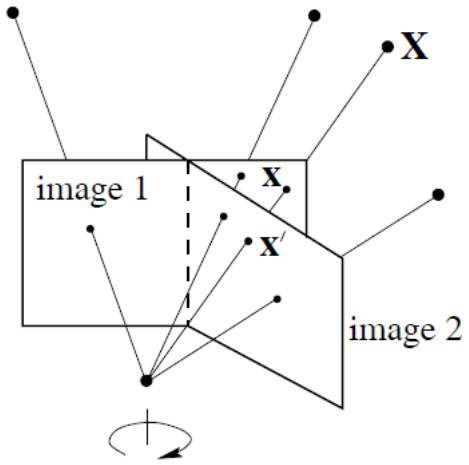
- Technique involved: SIFT Description, nearest neighbour matching, robust estimation(RANSAC), inverse warping

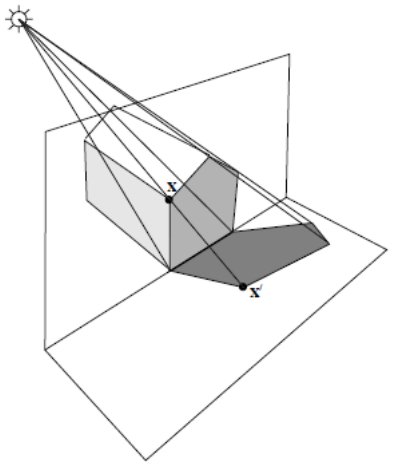
Projective Geometry











A Hierarchy of Transformation

- Distortions arising under different transformations

Similarity

Affine

Projective

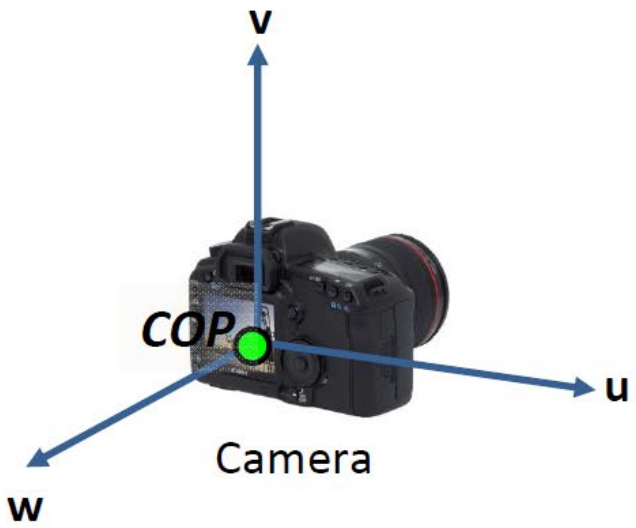
Example

[GitHub - cdbunker/Image-Rectification: Remove projective and affine distortion from images]

Camera Parameters and Calibration

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

Camera and World Co-ordinate Systems



Camera

"The World"

Two important coordinate systems:

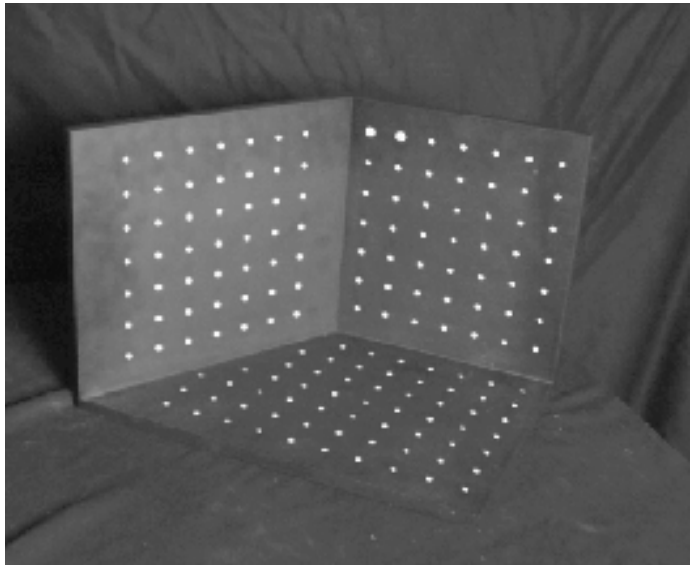
1. *World* coordinate system
2. *Camera* coordinate system

Camera Calibration

- Mapping from World Coord to image

$$\mathbf{x}=\mathrm{K}R[I_3|-\boldsymbol{X}_o]\mathbf{X}$$

- Goal: To estimate the intrinsic and extrinsic parameters of the camera
- Given: Known 3D points
- Observation: corresponding 2d points



Direct Linear Transform (DLT)

$$\mathbf{x}=\mathrm{K}R[I_3|-\boldsymbol{X}_o]\mathbf{X}$$

$$\mathbf{x}=\begin{bmatrix}u\\ \boldsymbol{v}\\ \boldsymbol{w}\end{bmatrix}=\begin{bmatrix}c & s & x_H\\ 0 & c(1+m) & y_H\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13}\\ r_{21} & r_{22} & r_{23}\\ r_{31} & r_{32} & r_{33}\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & t_X\\ 0 & 1 & 0 & t_Y\\ 0 & 0 & 1 & t_Z\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix}$$

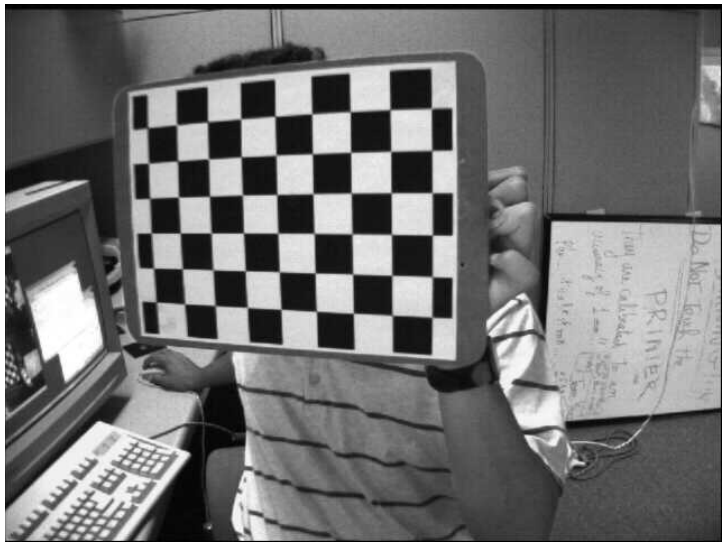
3x3 3x3 3x4

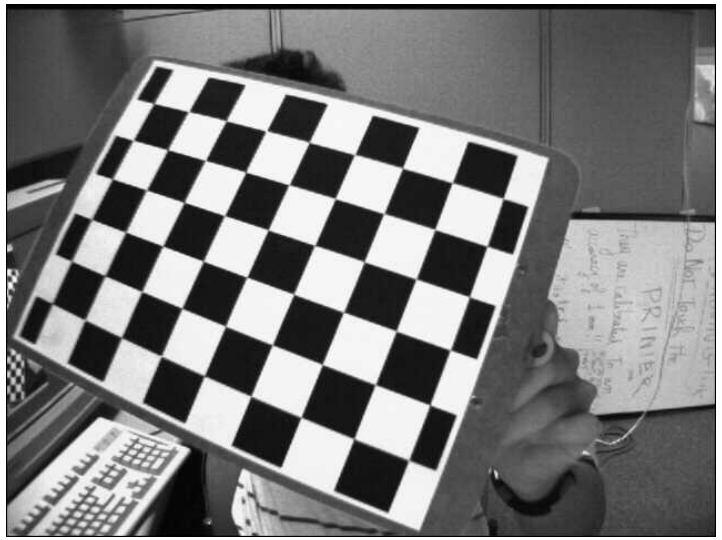
- can be combined as

$$\mathbf{x}=\begin{bmatrix}p_{11} & p_{12} & p_{13} & p_{14}\\ p_{21} & p_{22} & p_{23} & p_{24}\\ p_{31} & p_{32} & p_{33} & p_{34}\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix}$$

Multi-Plane Camera Calibration(Zhang 2000)

- Use a 2D known pattern (checkerboard) for calibration



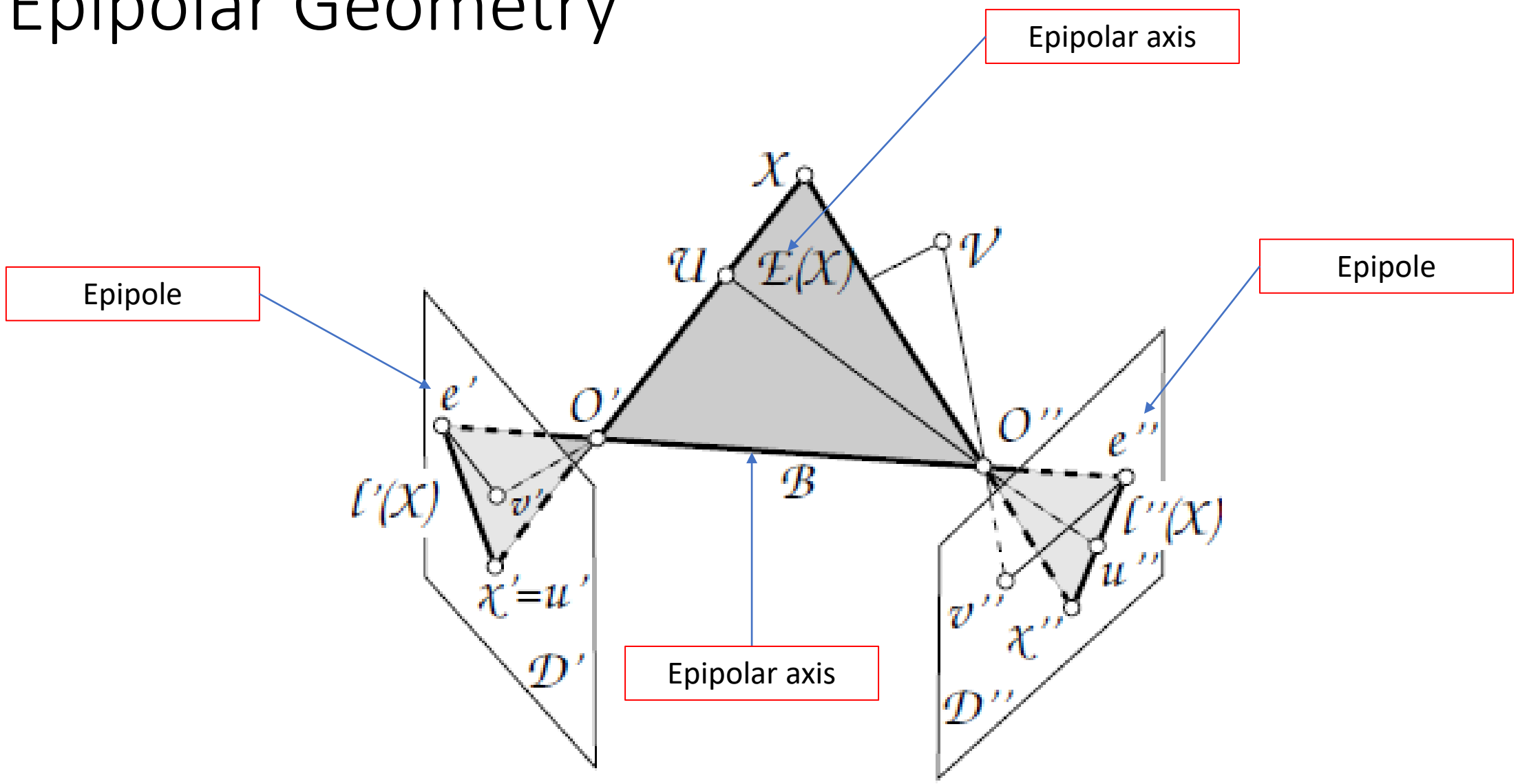


Epipolar Geometry

- Given a single image, we are not able to infer its three-dimensional structure as the depth information is lost.

- Therefore image pairs are considered!

Epipolar Geometry



Example – Epipolar Line

- Given: Two stereo image with known correspondence.
- Aim: to find the epipolar lines and epipoles.

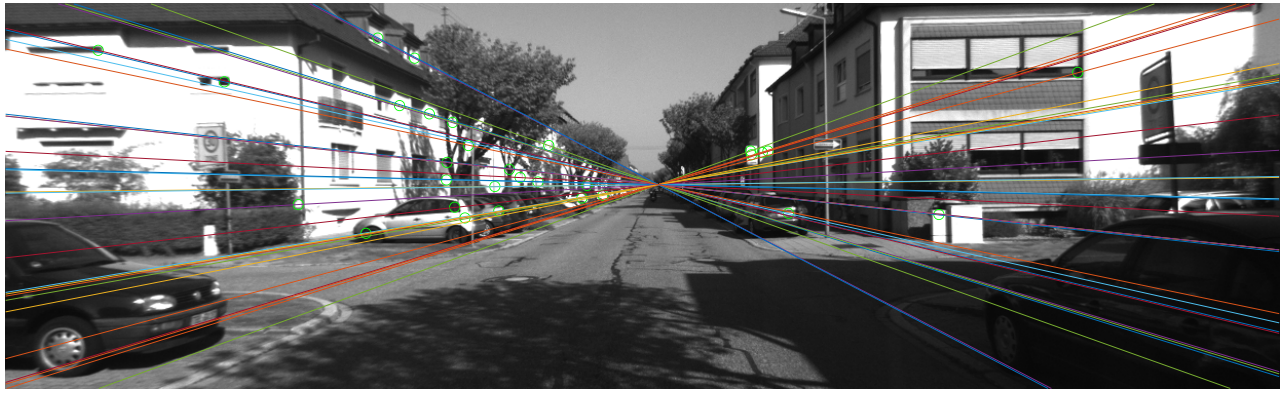
I1 (left); I2 (right)

Correspondence Points

- Given: Correspondence Points in Image 1 and Image 2
- $\mathrm{l}'' = \mathrm{F}^{T}\mathbf{x}'$

Correspondence Points in Image 1

Image 2 Epipolar Lines correspond to $\mathbf{x}'$



MATLAB Codes

```
figure; ax=axes;
imshow(I2);

%showMatchedFeatures(I1, I2, inlierPoints1a, inlierPoints2a);
%legend('Inlier points in I1', 'Inlier points in I2');
hold on;
%plot(inlierPoints2a(:,1),inlierPoints2a(:,2),'go')

%drawing the epipole
epipoleHC1=null(fMatrix2);
plot(epipoleHC1(1)/epipoleHC1(3),epipoleHC1(2)/epipoleHC1(3),'go');
```

```
%Calculate by formula l''=F'X'
inlierPoints1aHC=cart2hom(inlierPoints1a);
lss = zeros(30,3);
for i=1:30
    point = inlierPoints1aHC(i,:);
    lss(i,:)=fMatrix2'*point';  %the epipolar lines in image2
end
```

```
points = lineToBorderPoints(lss, size(I2));
line(points(:, [1,3])', points(:, [2,4])');
truesize;
hold off;
```

Epipolar Lines correspond to $\mathbf{x}'$

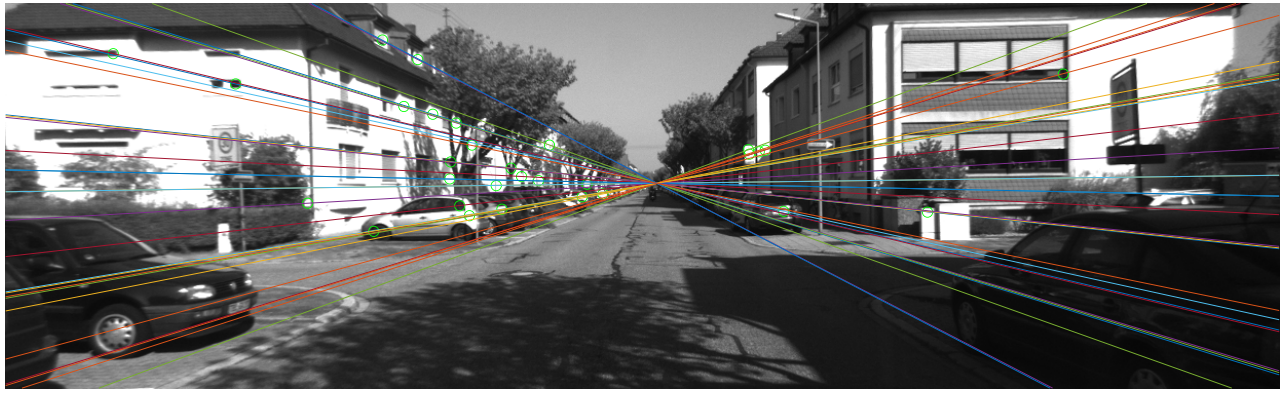
Similarly for Image 2

- Similarly for Image 1
- Epipolar lines in Image 1 are

$$l' = \mathbf{F}x''$$

Correspondence Points in Image 2

Image 1: Epipolar Lines correspond to $\mathbf{x}''$



MATLAB Codes

```
figure; ax=axes;
imshow(I1);
hold on;
%drawing the corresponding points
%plot(inlierPoints1a(:,1),inlierPoints1a(:,2),'go');

%drawing the epipole
epipoleHC1=null(fMatrix2');
plot(epipoleHC1(1)/epipoleHC1(3),epipoleHC1(2)/epipoleHC1(3),'go');


% Compute the Epipolar Lines from corresondence point on second image
inlierPoints2aHC=cart2hom(inlierPoints2a);
ls = zeros(30,3);
for i=1:30
    point = inlierPoints2aHC(i,:);
    ls(i,:)=fMatrix2*point'; % The epipolar line in image 1
end

%drawing the lines
points = lineToBorderPoints(ls, size(I2));
line(points(:, [1,3])', points(:, [2,4])');
truesize;
hold off;

figure; ax=axes;
imshow(I2);
```

Epipolar Lines correspond to $\mathbf{x}''$

Fundamental Matrix

F is the *fundamental matrix* of the *relative orientation* of a pair of images of *uncalibrated cameras*

$$\mathrm{F}=(\mathrm{K}')^{-T} R'\, S_b\, (R'')^{T} (\mathrm{K}'')^{-1}$$

Which fulfills the equation

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

- Remarks:
  - Fundamental matrix F 3x3 matrix with 7 DoF.
  - F is homogenous (AF=0)
  - F is singular det(F)=0

Essential Matrix

- The essential matrix for calibrated camera

$$\mathrm{E} = R' \, S_b \, (R'')^T$$

- E express the R.O. and can be parametrized through

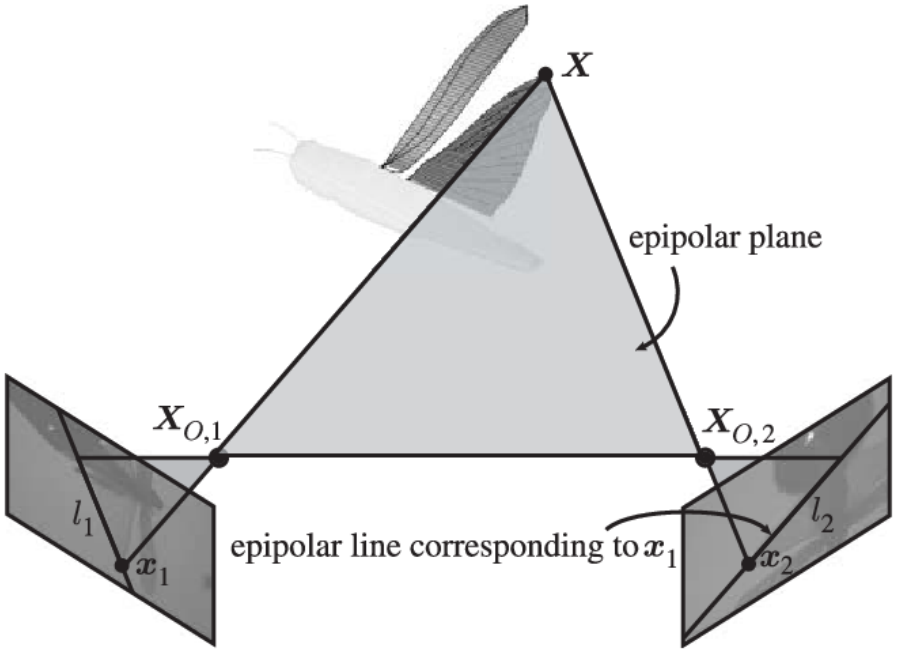
$$\mathrm{E} = \; S_b \, R^T$$

The coplanarity constraint for calibrated cameras

$${}^{c}\mathrm{x}'^{\,T} \mathrm{E} \; {}^{c}\mathrm{x}'' \; = 0$$

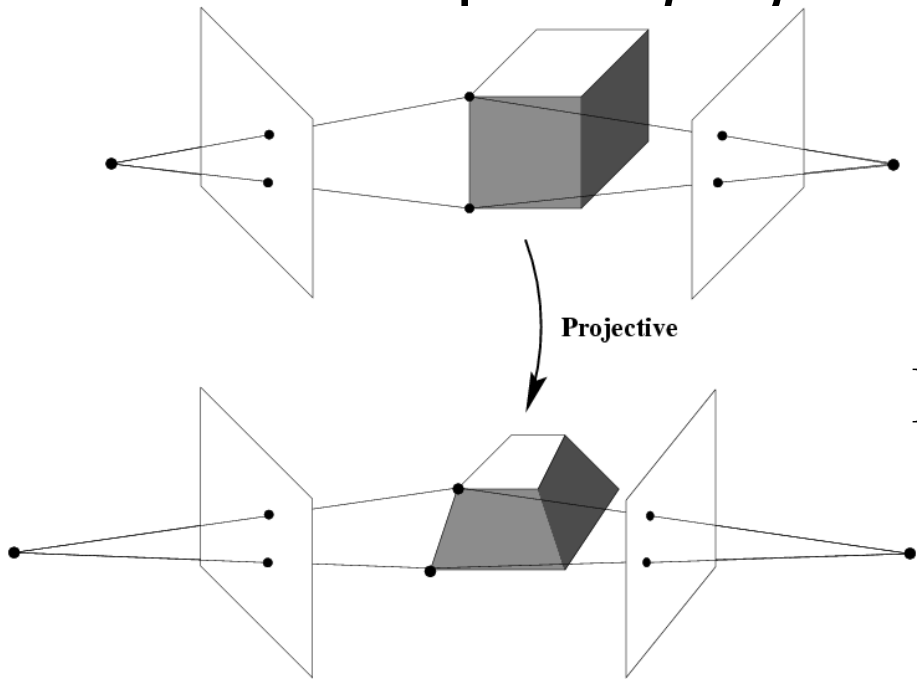
Triangulation

- **Given**: A pair of cameras with known relative orientation
- **Aim**: To compute the points in 3D



Projective ambiguity

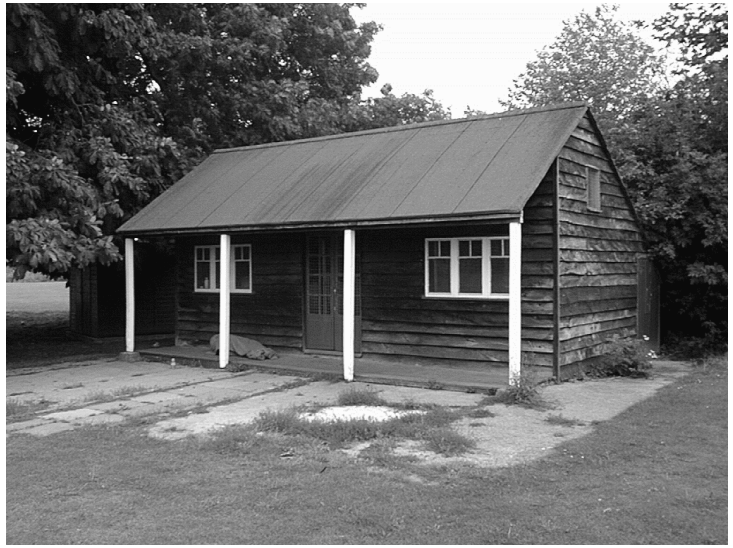
- With no constraints on the camera calibration matrix or on the scene, we can reconstruct up to a *projective* ambiguity

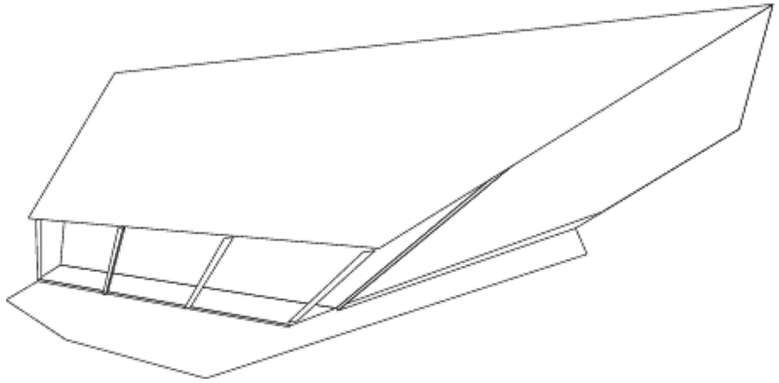


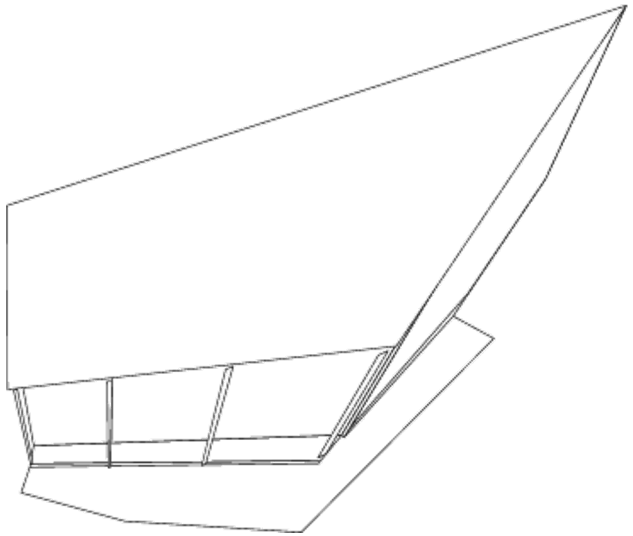
$$\mathbf{H}_p = \begin{bmatrix} \mathrm{A} & \mathrm{t} \\ \mathrm{v}^\mathsf{T} & v \end{bmatrix}$$

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{H}_\mathbf{P}^{-1}\right)\left(\mathbf{H}_\mathbf{P}\,\mathbf{X}\right)$$

Projective ambiguity

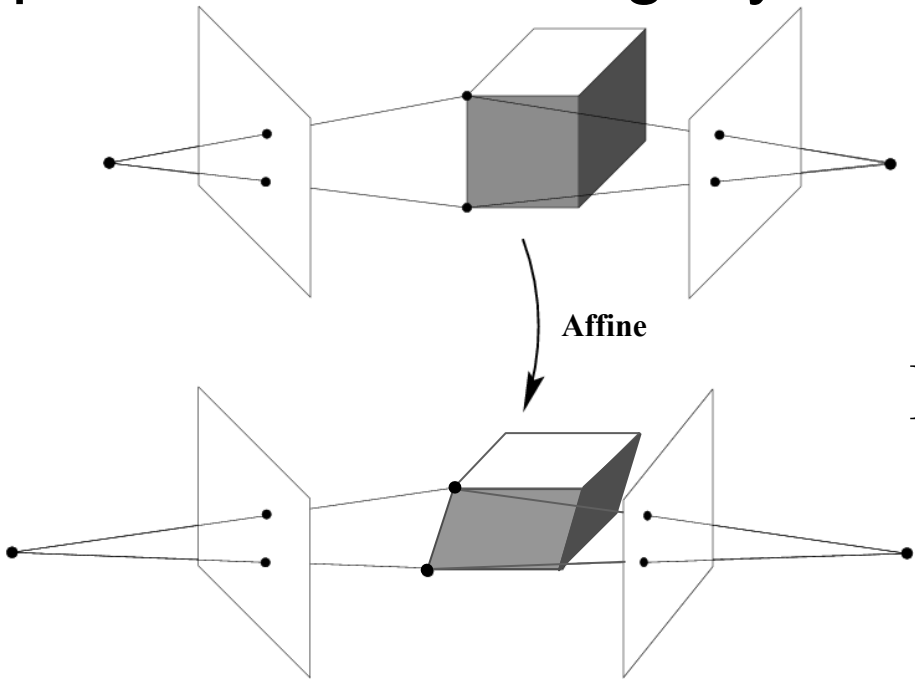






Affine ambiguity

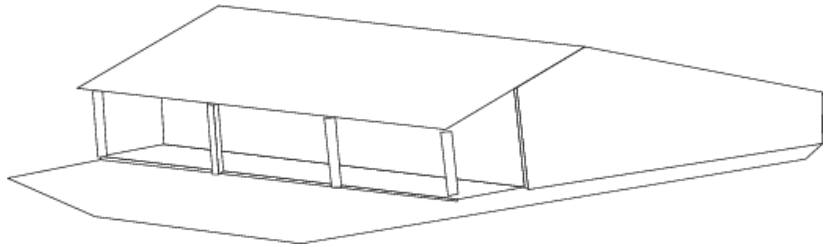
- If we impose parallelism constraints, we can get a reconstruction up to an *affine* ambiguity

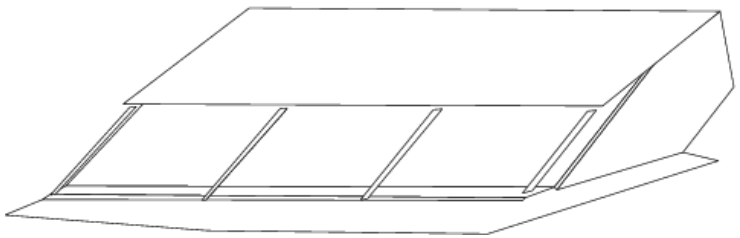


$$\mathbf{H}_A = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{PX} = \left(\mathbf{PH}_A^{-1}\right)\left(\mathbf{H}_A\,\mathbf{X}\right)$$

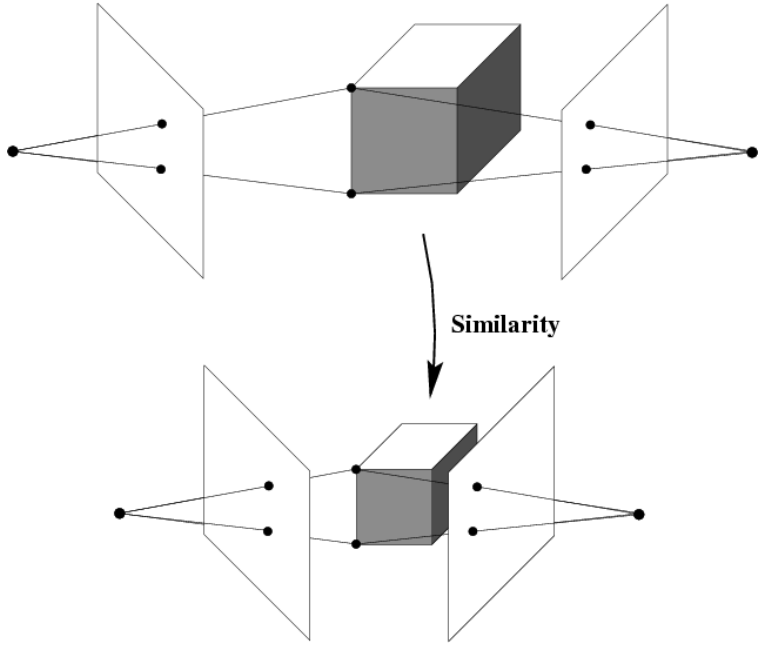
Affine ambiguity





Similarity ambiguity

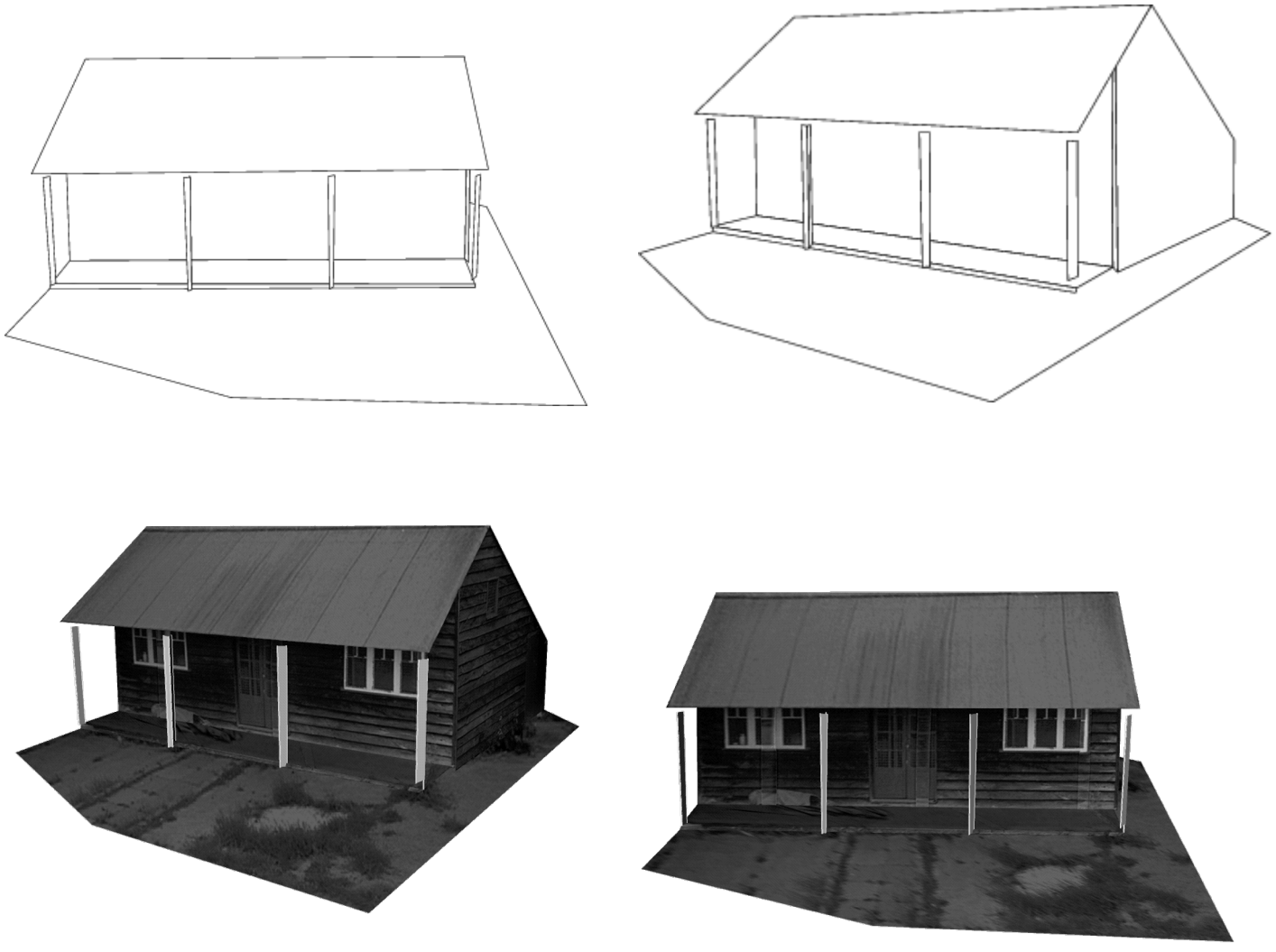
- A reconstruction that obeys orthogonality constraints on camera parameters and/or scene



$$\mathbf{H}_s = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{PX} = \left(\mathbf{PH}_S^{-1}\right)\left(\mathbf{H}_S\mathbf{X}\right)$$

Similarity ambiguity



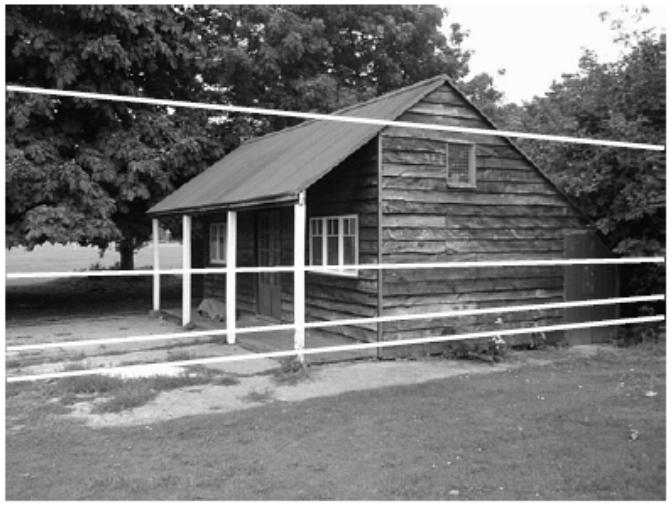
Affine properties of a reconstruction

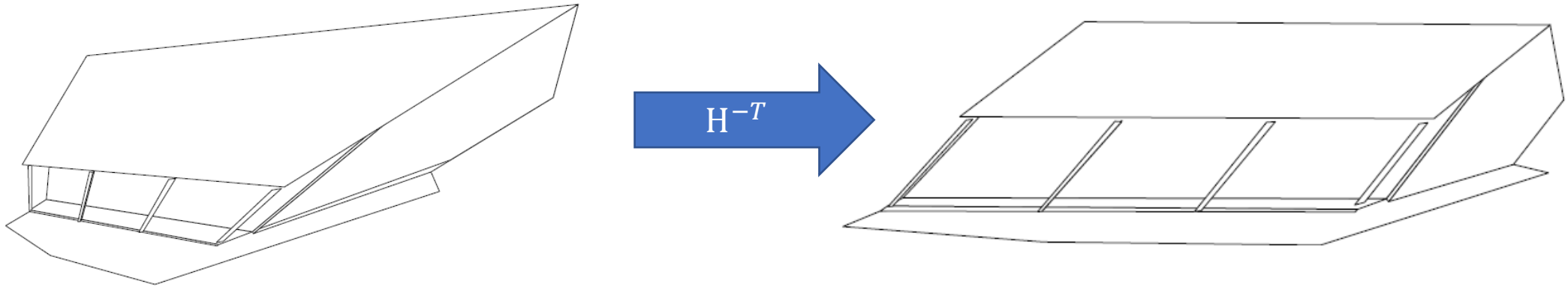
- We have seen how to restore the affine properties in image early in this lecture.
- Once we have π_∞ is identified in projective 3-space, it is then possible to determine affine properties of the reconstruction.
- A more algorithmic approach is to transform $\mathbb{P}^3$ so that the identified π_∞ is moved to its canonical position at

$$\pi_\infty = (0,0,0,1)^T$$

- We will talk more in reconstruction section.

From Projective to Affinity





Metric Reconstruction Result.

- Once the Ω_∞ (and its support plane π_∞) have been identified in projective 3-space, we can remove the affine distortion of the reconstructed scene.

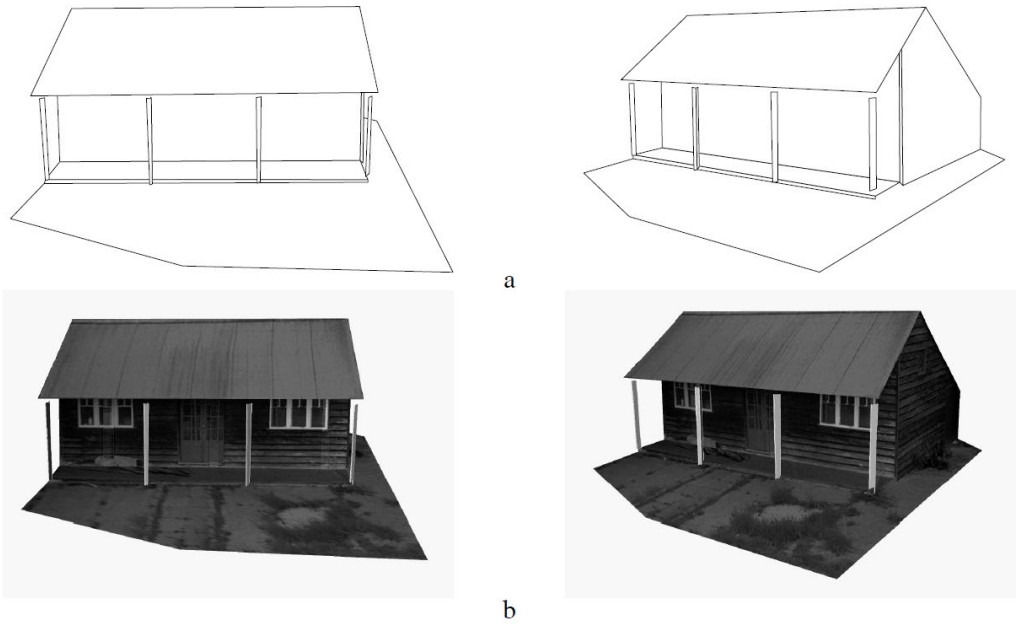
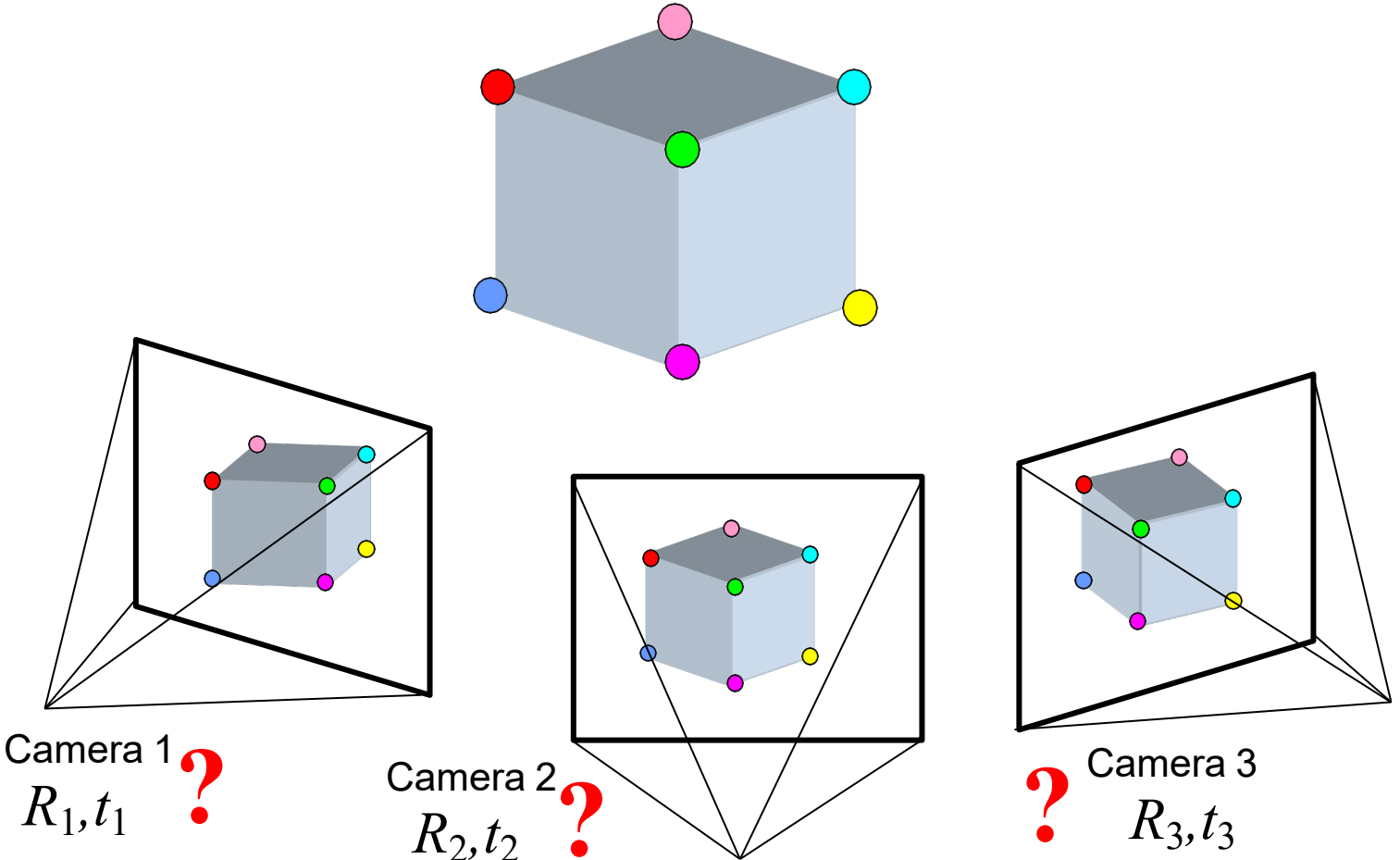


Image Credit: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

Structure from motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



Slide credit:
Noah Snavely

Clap for yourself!


thank you