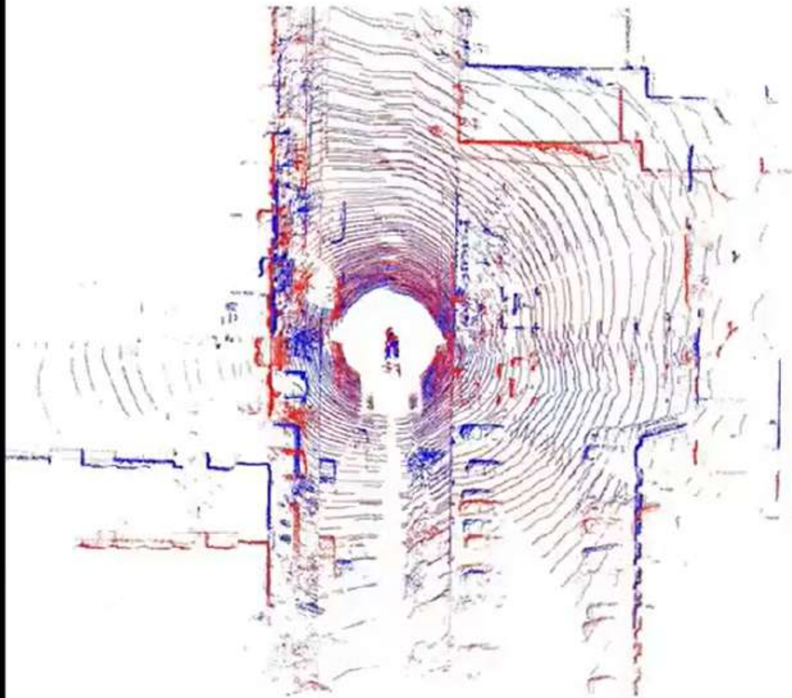
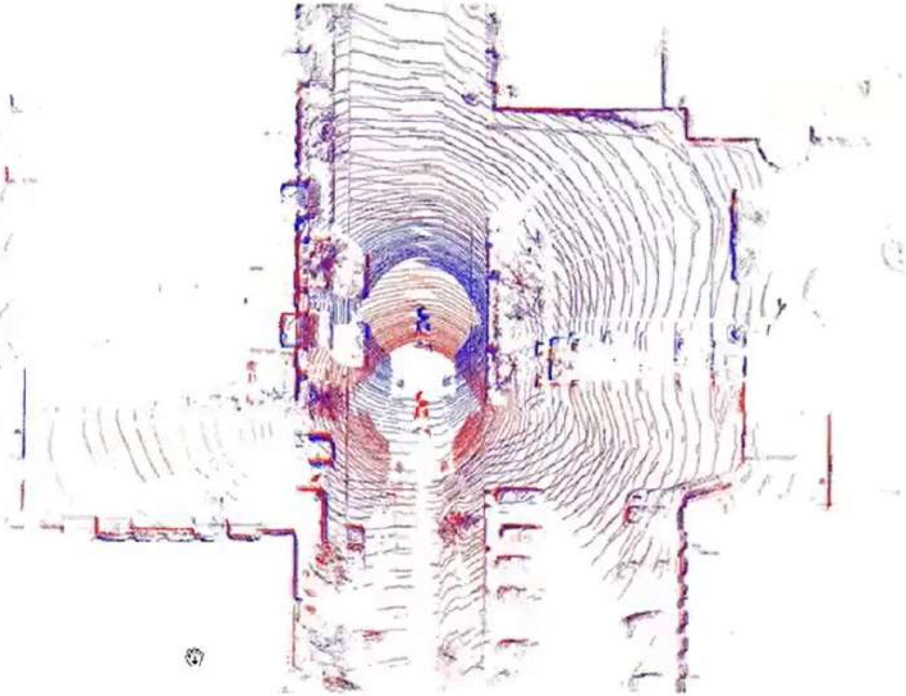
Scan Alignment



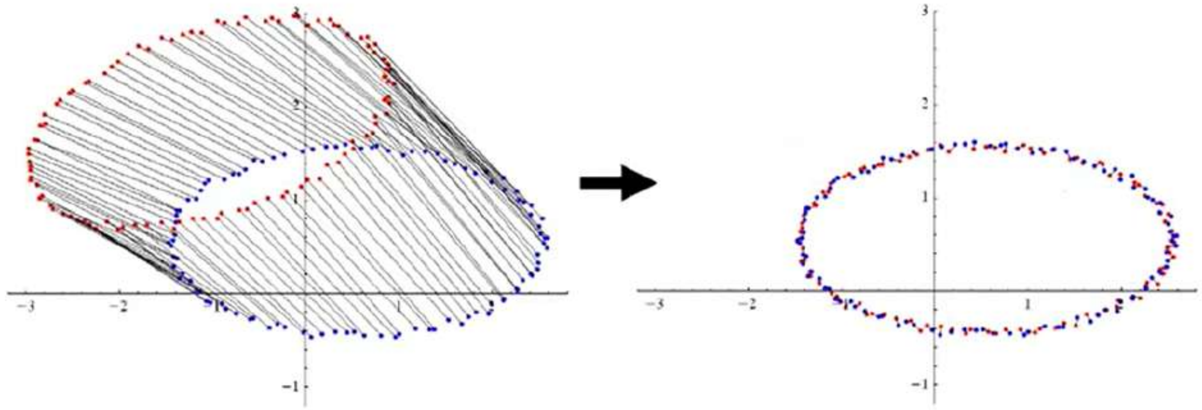
Not-Aligned



Aligned

Scan Alignment: SVD

- Optimal alignment between corresponding points
  - Assuming that for each source point, we know where the corresponding target point is



Slide Credit: http://www.cse.wustl.edu/~taoju/cse554/lectures/lect05_Alignment

Scan Alignment: SVD

- SVD algorithm:
  - Let P be a matrix whose *i*-th column is vector $p_i - c_S$
  - Let Q be a matrix whose *i*-th column is vector $q_i - c_T$
  - Consider the cross-covariance matrix:

$$M = PQ^\top$$

  - Find SVD of M:

$$M = U\Sigma V^\top$$

  - Find Rotation R:

$$R = UV^\top$$

  - Find Translation:

$$\vec{t} = c_T - R * c_S$$

Scan Alignment: SVD

- SVD algorithm:
  - Let P be a matrix whose *i*-th column is vector $p_i - c_S$
  - Let Q be a matrix whose *i*-th column is vector $q_i - c_T$
  - Consider the cross-covariance matrix:

$$M = PQ^\top$$

  - Find SVD of M:

$$M = U\Sigma V^\top$$

  - Find Rotation R:

$$R = UV^\top$$

  - Find Translation:

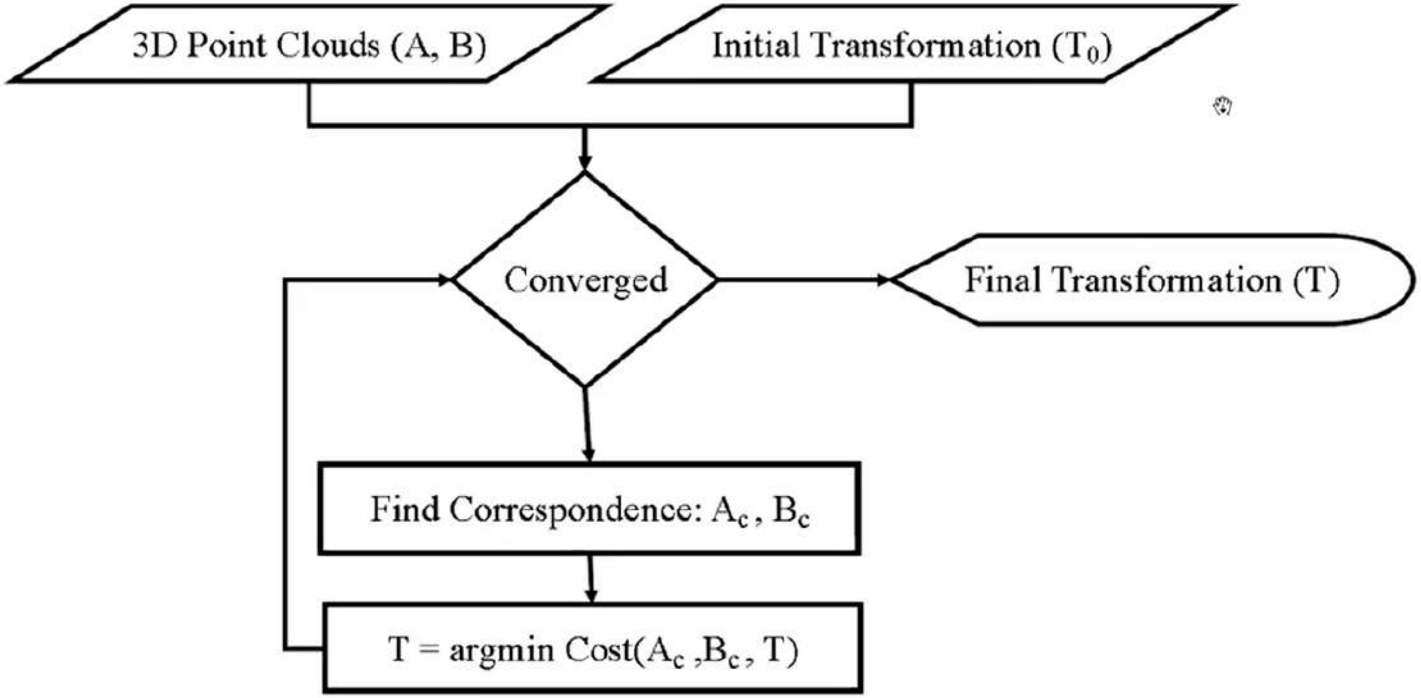
$$\vec{t} = c_T - R * c_S$$

Limitations of SVD

- Requires correct point correspondences
  - Generally correspondences are not known !!

Slide Credit: http://www.cse.wustl.edu/~taoju/cse554/lectures/lect05_Alignment

Iterative Closest Point (ICP)



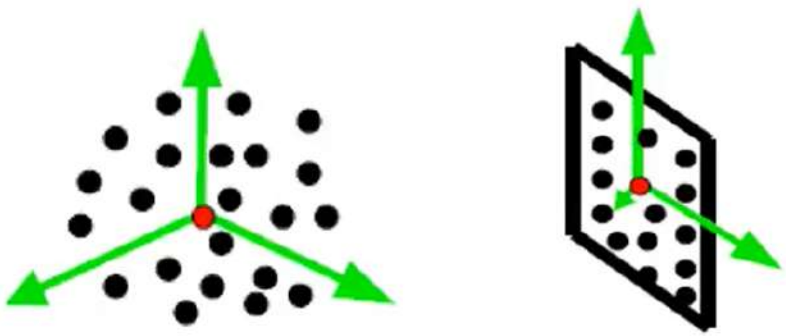
Besl, P. J. and McKay, N. D. (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239-255.

Generalized ICP

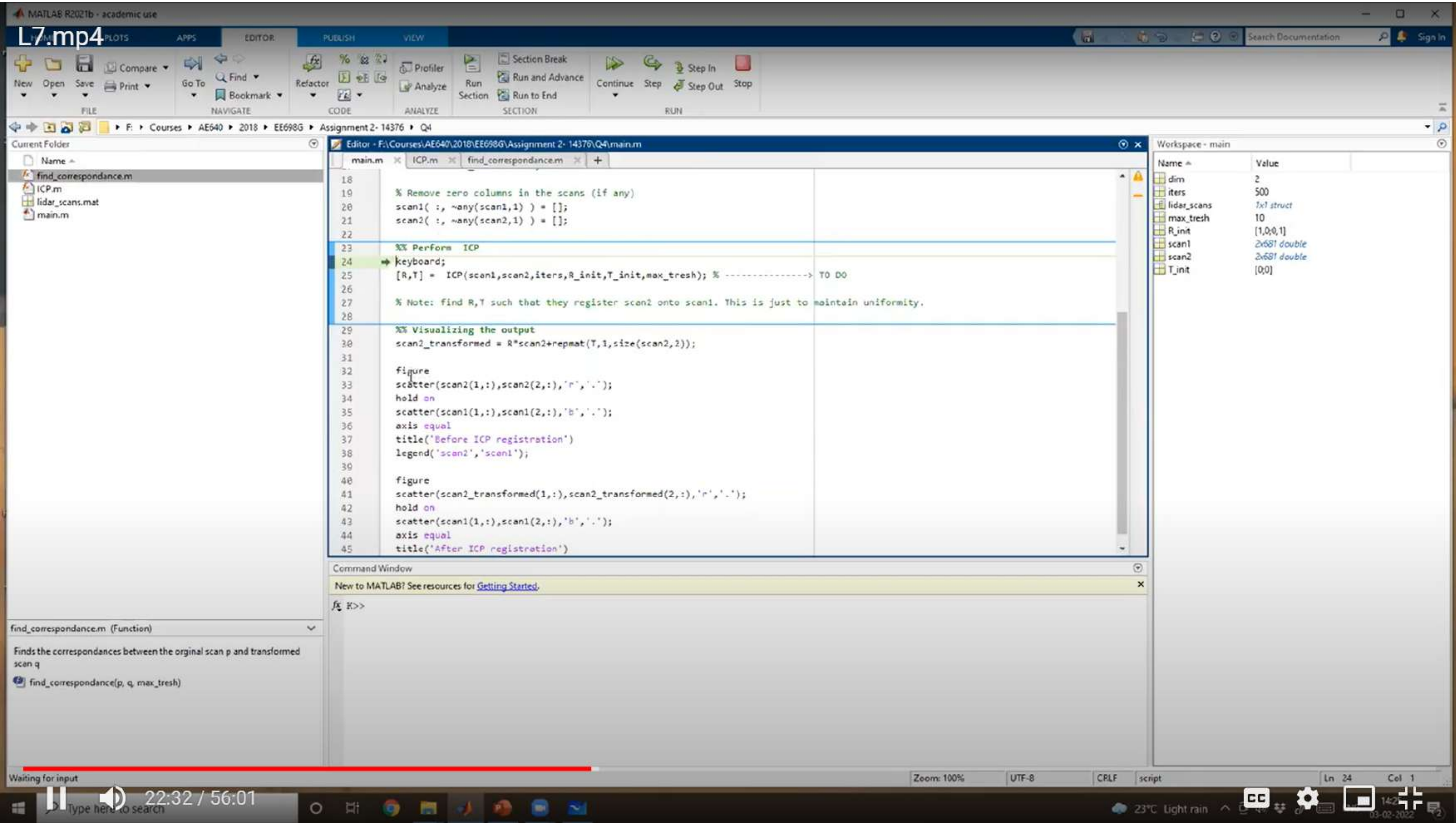
- Generalized-ICP is based on attaching a probabilistic model to the minimization step of the standard ICP algorithm.

$$A = \{a_i\}_{i=1,2...N} \quad B = \{b_i\}_{i=1,2...N} \quad (\text{st } a_i == b_i)$$

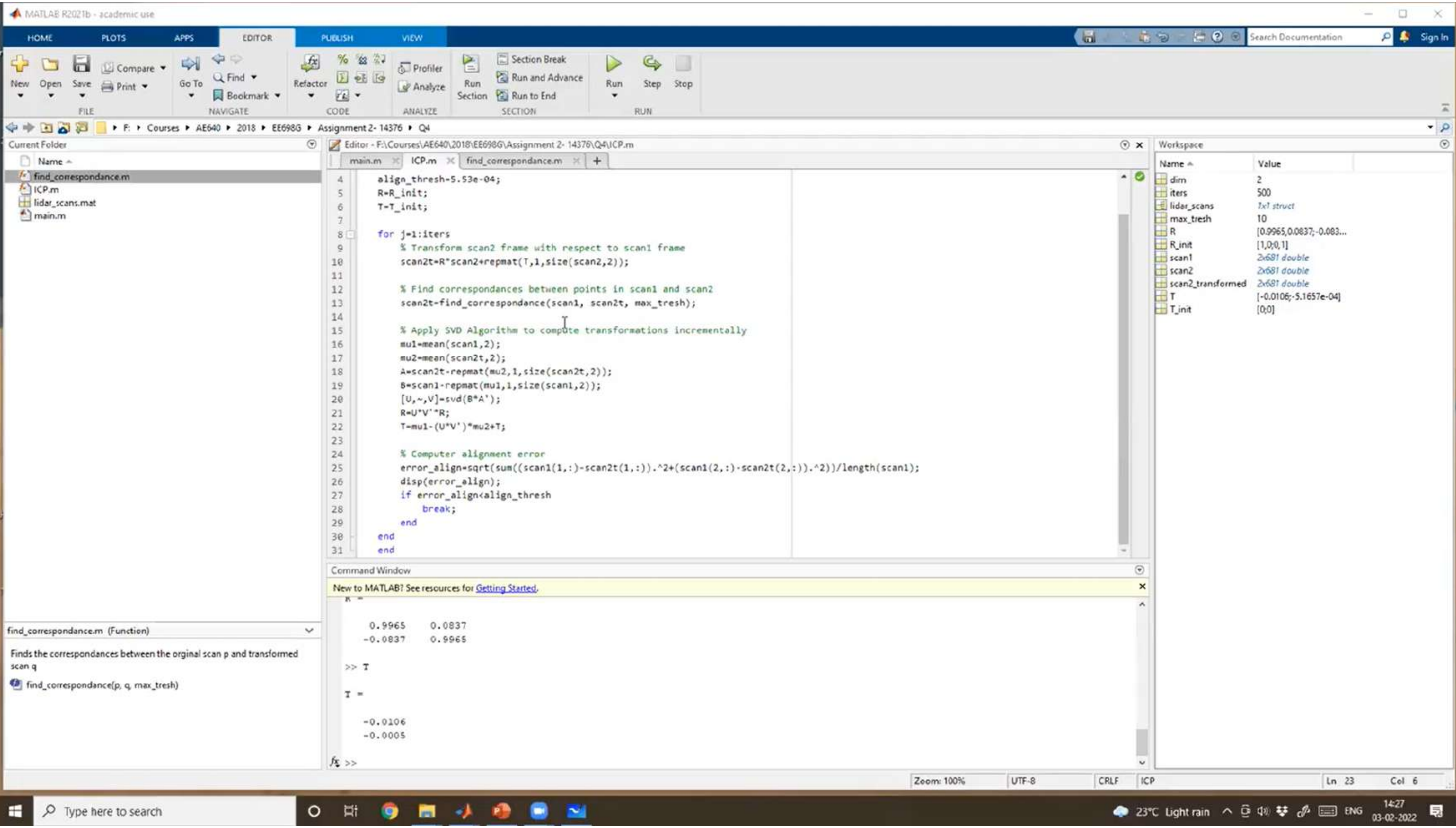
$$a_i \sim N(\mu_{ai}, \Sigma_{Ai}) \; ; \; b_i \sim N(\mu_{bi}, \Sigma_{Bi})$$



Reference: "Generalized ICP" by A. Segal, D. Haehnel, S. Thrun. RSS 2009



MATLAB R2021b - academic use
L7.mp4
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Continue
Step
Step In
Step Out
Stop
RUN
F: ▸ Courses ▸ AE640 ▸ 2018 ▸ EE698G ▸ Assignment 2- 14376 ▸ Q4
Current Folder
Name
find_correspondance.m
ICP.m
lidar_scans.mat
main.m
Editor - F:\Courses\AE640\2018\EE698G\Assignment 2- 14376\Q4\main.m
main.m
ICP.m
find_correspondance.m
% Remove zero columns in the scans (if any)
scan1(:, ~any(scan1,1)) = [];
scan2(:, ~any(scan2,1)) = [];
%% Perform ICP
keyboard;
[R,T] = ICP(scan1,scan2,iters,R_init,T_init,max_tresh); % --------------> TO DO
% Note: find R,T such that they register scan2 onto scan1. This is just to maintain uniformity.
%% Visualizing the output
scan2_transformed = R*scan2+repmat(T,1,size(scan2,2));
figure
scatter(scan2(1,:),scan2(2,:),'r','.');
hold on
scatter(scan1(1,:),scan1(2,:),'b','.');
axis equal
title('Before ICP registration')
legend('scan2','scan1');
figure
scatter(scan2_transformed(1,:),scan2_transformed(2,:),'r','.');
hold on
scatter(scan1(1,:),scan1(2,:),'b','.');
axis equal
title('After ICP registration')
Workspace - main
Name
Value
dim
2
iters
500
lidar_scans
1x1 struct
max_tresh
10
R_init
[1,0;0,1]
scan1
2x681 double
scan2
2x681 double
T_init
[0;0]
Command Window
New to MATLAB? See resources for Getting Started.
K>>
find_correspondance.m (Function)
Finds the correspondances between the orginal scan p and transformed scan q
find_correspondance(p, q, max_tresh)
Waiting for input
Zoom: 100%
UTF-8
CRLF
script
Ln 24
Col 1
22:32 / 56:01
Type here to search
23°C Light rain


MATLAB R2021b - academic use
HOME PLOTS APPS EDITOR PUBLISH VIEW
Editor - F:\Courses\AE640\2018\EE698G\Assignment 2- 14376\Q4\ICP.m
main.m ICP.m find_correspondance.m
align_thresh=5.53e-04;
R=R_init;
T=T_init;
for j=1:iters
% Transform scan2 frame with respect to scan1 frame
scan2t=R*scan2+repmat(T,1,size(scan2,2));
% Find correspondances between points in scan1 and scan2
scan2t=find_correspondance(scan1, scan2t, max_tresh);
% Apply SVD Algorithm to compute transformations incrementally
mu1=mean(scan1,2);
mu2=mean(scan2t,2);
A=scan2t-repmat(mu2,1,size(scan2t,2));
B=scan1-repmat(mu1,1,size(scan1,2));
[U,~,V]=svd(B*A');
R=U*V'*R;
T=mu1-(U*V')*mu2+T;
% Computer alignment error
error_align=sqrt(sum((scan1(1,:)-scan2t(1,:)).^2+(scan1(2,:)-scan2t(2,:)).^2)/length(scan1));
disp(error_align);
if error_align<align_thresh
break;
end
end
end
Current Folder
find_correspondance.m
ICP.m
lidar_scans.mat
main.m
find_correspondance.m (Function)
Finds the correspondances between the orginal scan p and transformed scan q
find_correspondance(p, q, max_tresh)
Command Window
New to MATLAB? See resources for Getting Started.
0.9965 0.0837
-0.0837 0.9965
>> T
T =
-0.0106
-0.0005
Workspace
Name Value
dim 2
iters 500
lidar_scans 1x1 struct
max_tresh 10
R [0.9965,0.0837;-0.083...
R_init [1,0;0,1]
scan1 2x681 double
scan2 2x681 double
scan2_transformed 2x681 double
T [-0.0106;-5.1657e-04]
T_init [0;0]
Zoom: 100% UTF-8 CRLF ICP Ln 23 Col 6
Type here to search
23°C Light rain ENG 14:27 03-02-2022

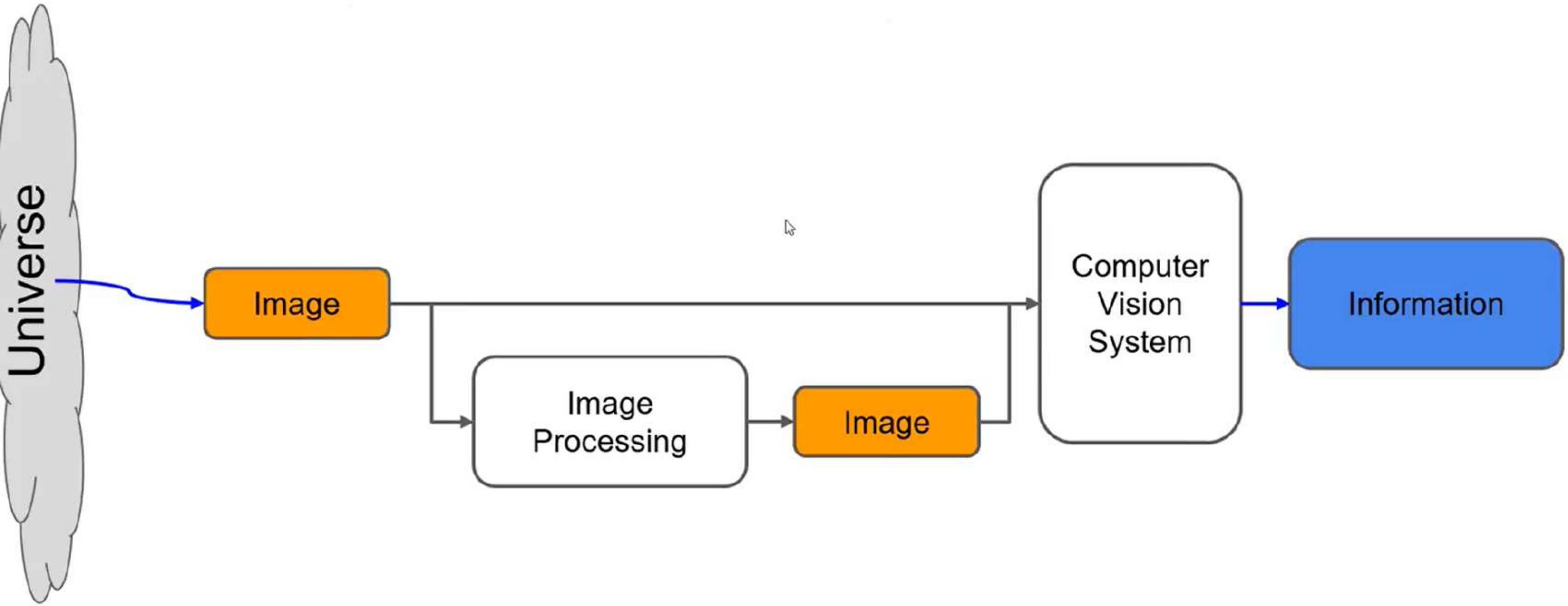
Introduction to Computer Vision for Robotics

AE640A Autonomous Navigation

Mangal Kothari

Credit: Harsh Sinha

What is Computer Vision?



What is Computer Vision?

- *Vision* is about discovering from images what is present in the scene and where it is.
- In *Computer Vision* a camera (or several cameras) is linked to a computer. The computer interprets images of a real scene to obtain information useful for tasks such as navigation, manipulation and recognition.

Overview of the field

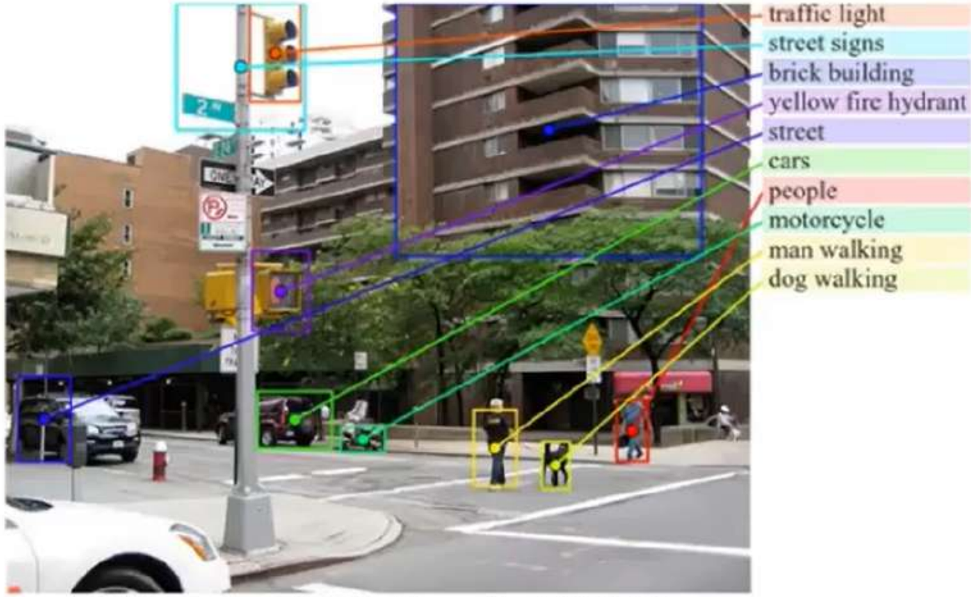


Image Credits: Karpathy, CVPR'15

What kind of Information?

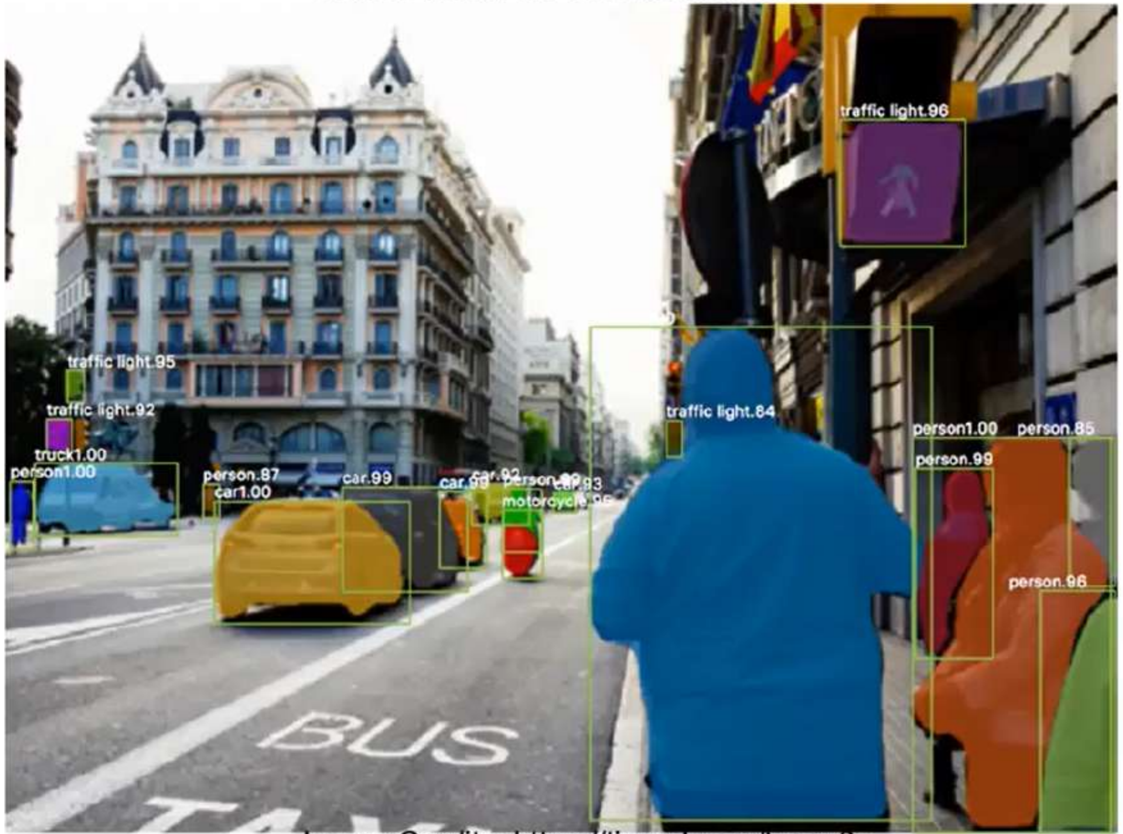


Image Credits: https://tinyurl.com/lxuex6o

Overview of the field

Primary themes in Computer Vision are:

1. Object Detection
2. Segmentation

Segmentation: Which pixels belong to which object?
Credits: Own Work

Overview of the field

Primary themes in Computer Vision are:

1. Object Detection
2. Segmentation
3. Image Modifications/Enhancements

Image Colorization: From Grayscale to Colored Images
Credits: Richard Zhang, CVPR 2016

Overview of the field

Primary themes in Computer Vision are:

1. Object Detection
2. Segmentation
3. Image Modifications/Enhancements
4. Image to Text
5. Image Generation
6. Motion Estimation
7. 3D reconstruction from Images

3D Reconstruction: REMODE,
Real Time Reconstruction
Credits: Matia Pizzoli, ICRA 2014

A look at history

- **Robert Nathan** started writing computer programs for enhancing images from NASA's spacecraft's at Jet Propulsion Lab, NASA.
- The Summer Vision Project: Project at MIT to **solve a significant part of visual system.** Primary Objective was to divide the image into object, background and chaos regions, **over the course of a summer.**

Credits: EE604, nasa.gov

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group — July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

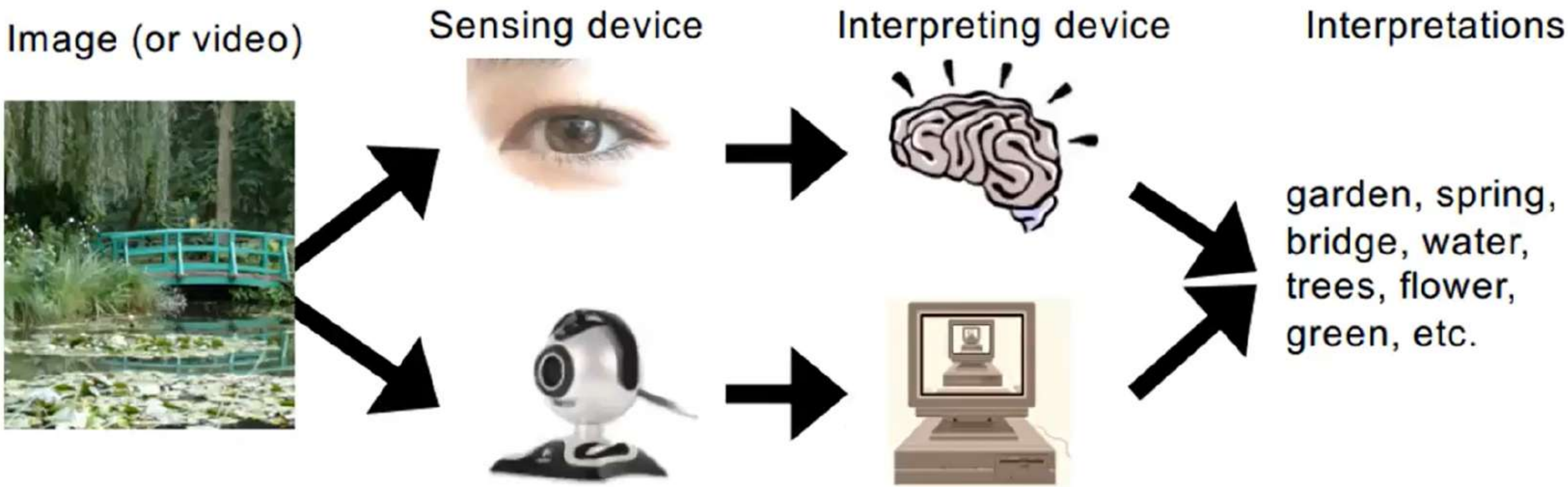
Credits: https://tinyurl.com/y6bpo4nk

Hard Problem?

- Why are we still working on roughly the same problem as the "summer vision project"?
- Why is it that creating 3D models of chairs is easier than identifying them?

➔ There is a large between some ~1920x1080x3 numbers and the high-level abstract meaning we associate with them.
➔ Images are **2D** representation of information from **3D** world.

The ~~(human) (computer)~~ vision system



Credits: CS131, Stanford

Camera Model~~s~~

Not this one but models as
in modelling a phenomena

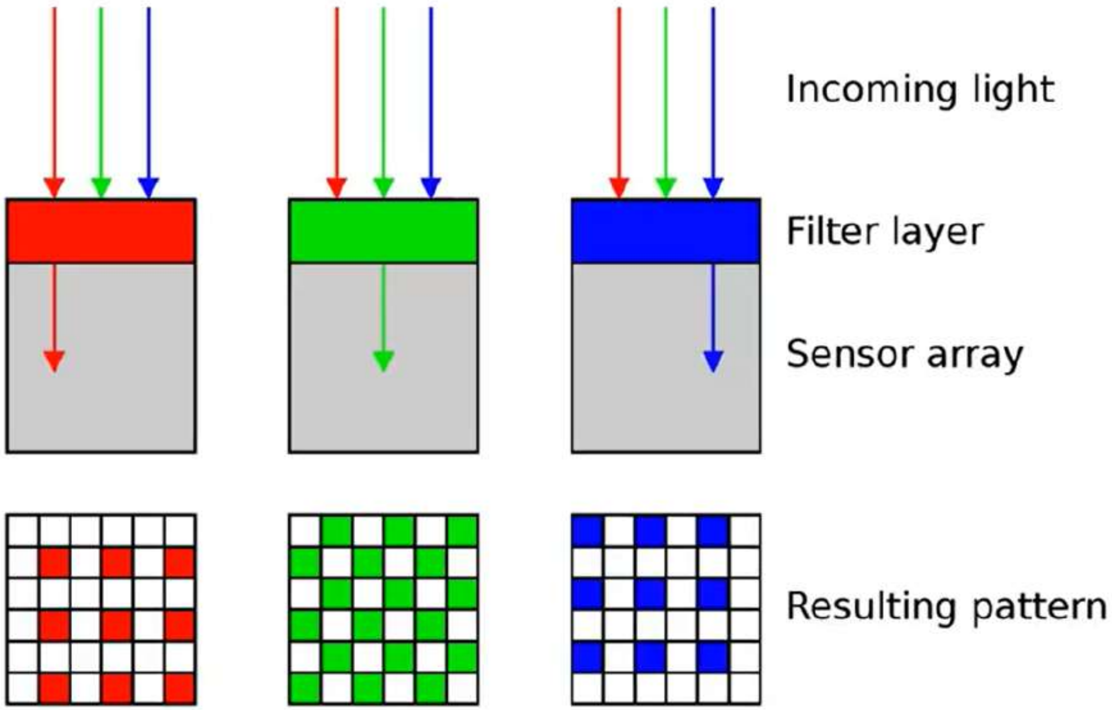
Credits: https://tinyurl.com/y6qen2vb

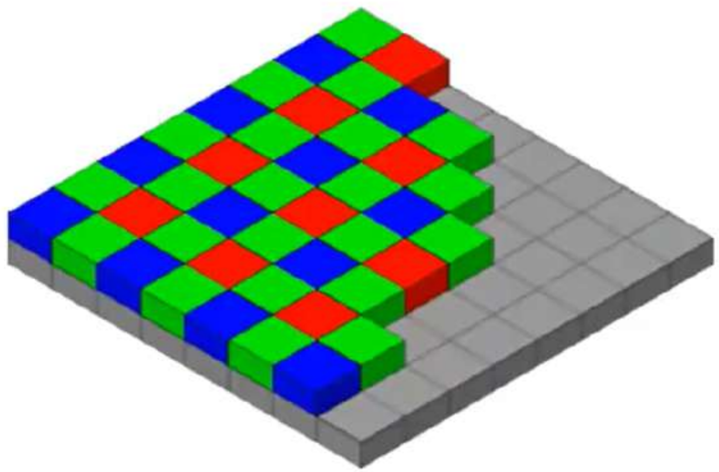
Camera Models

- Like so many things in engineering, we create a simple “model” of a camera to which is easy to understand and can approximate the actual functioning of a camera to a good degree.
- There are different models:
    - Pinhole camera model
    - Lens model
    - ...

Cameras

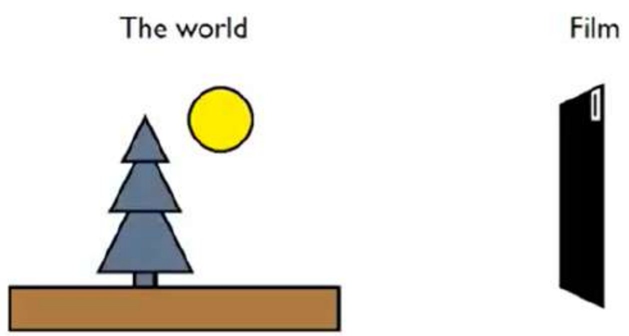
- Demosaicing interpolates RGB subsamples to get colour image





This filter is called Bayer filter

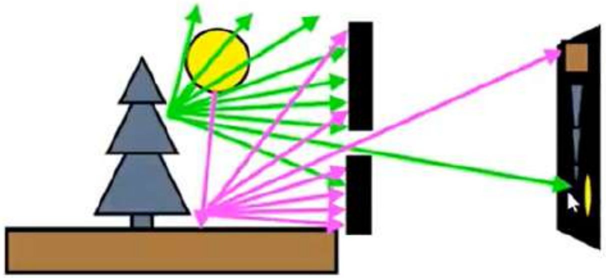
World Simplest Camera?



- Just hold up a piece of film
- Do we get an image on the film?
  - For each piece of the film, where do the photons come from?

Slide Credit: Edwin Olson, University of Michigan

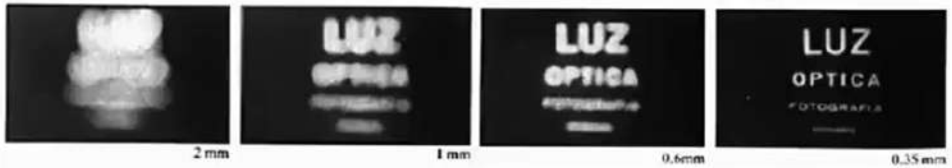
Let's add an aperture



- An aperture blocks all but a small subset of the rays
  - Causes the image to appear in focus!

Slide Credit: Edwin Olson, University of Michigan

Aperture Size

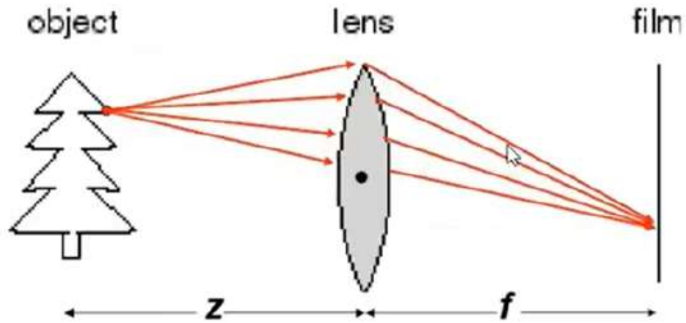


- Why not make the aperture super small?
  - A "pin-hole" lens.
  - Not enough light to "register" on our film
- What happens when the aperture is bigger?
  - More rays can fit through--- blurrier image

- Is there any way of getting a sharp image, but allow more light through?
  - Yes! A *lens*.

Slide Credit: Edwin Olson, University of Michigan

Lenses

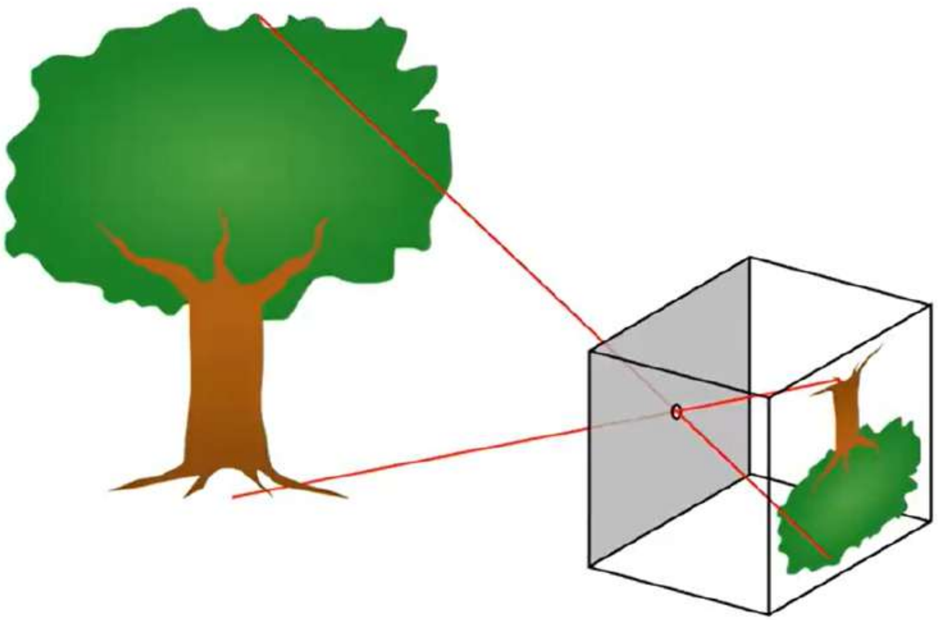


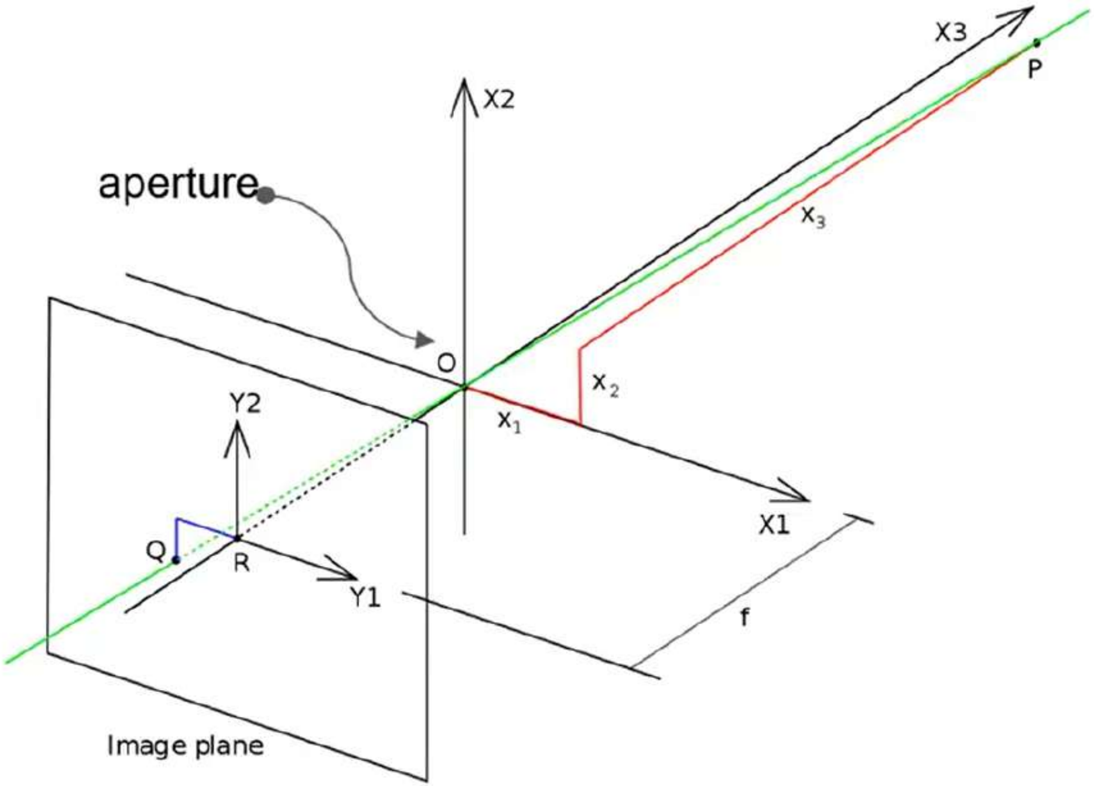
- A lens collects rays with a particular divergence and refocuses them to a point.
  - But points at the "wrong" distance won't be refocused exactly.
- *Depth of field:* how much of the scene is in focus

- We're going to ignore this today, however--- we're going to assume a "pin-hole" model.

Slide Credit: Edwin Olson, University of Michigan

Pinhole camera model





Credits: Wikipedia, Pinhole Camera Model

The Perspective Matrix

- Suppose we write a point in the world (like the position of the candle flame) as a vector:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- Can we write a matrix so that p' = Mp?

$$p' = \begin{bmatrix} fx/z + c_x \\ fy/z + c_y \end{bmatrix}$$

Slide Credit: Edwin Olson, University of Michigan

Do homogeneous coordinates help?

- Eureka!

$$p' = \underbrace{\begin{bmatrix} fx/z + c_x \\ fy/z + c_y \\ 1 \end{bmatrix}}_{\text{2d homogeneous coordinates}} = \underbrace{\begin{bmatrix} fx + c_x z \\ fy + c_y z \\ z \end{bmatrix}}_{\text{2d homogeneous coordinates}} = \underbrace{\begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{perspective transform}} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\text{3d homogeneous coordinates}}$$

Slide Credit: Edwin Olson, University of Michigan

Rigid-Body Transformations

- The product of two rigid-body transformations is ***always*** another rigid-body transformation!
- So no matter how the object has been translated or rotated, we can describe its position with a single 4x4 matrix, which has the structure:

T_x
T_y
T_z

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & T_x \\ R_{10} & R_{11} & R_{12} & T_y \\ R_{20} & R_{21} & R_{22} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Slide Credit: Edwin Olson, University of Michigan

L7.mp4

Putting it all together

$$\begin{bmatrix} x' \\ y' \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{00} & R_{01} & R_{02} & T_x \\ R_{10} & R_{11} & R_{12} & T_y \\ R_{20} & R_{21} & R_{22} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2d homogeneous pixel (camera) coordinates

Focal length and focal center of camera

Rigidly move every object in the world to simulate the camera's true position

3d homogenous (world) coordinates

"Intrinsics" "Extrinsics"

Slide Credit: Edwin Olson, University of Michigan

54:08 / 56:01