ECE 470: Introduction to Robotics
Lecture 26

Liangjing Yang

Assistant Professor, ZJU-UIUC Institute

liangjingyang@intl.zju.edu.cn

Wechat ID: Liangjing_Yang

Our Learning Roadmap

- **Schedule Check on our Learning Roadmap**

O. Overview
- Science & Engineering in Robotics

I. Spatial Representation & Transformation
- Coordinate Systems; Pose Representations; Homogeneous Transformations

II. Kinematics
- Multi-body frame assignment; D-H Convention; Joint-space; Work-space; Forward/Inverse Kinematics

III. Velocity Kinematics and Static Forces
- Translational/Rotational Velocity; Joint torque; Generalized Force Coordinates; Jacobian; Singularity

Fundamentals
Week 1-4
Revision/ Quiz on Week 5

IV. Dynamics
- Acceleration of Body; Newton-Euler Equations of Motion; Lagrangian Formulation

V. Control
- Closed-Loop Control and Feedback, Control of 2nd order system, Independent Joint Control, Force Control

VI. Planning
- Joint-Based Scheme; Cartesian-Based Scheme; Collision Free Path Planning

Essentials
Week 6-9
Revision/ Quiz on Week 10

VII. Robot Vision (Perception)
- Image Formation; Image Processing; Visual Tracking & Pose Estimation; Vision-based Control & Image-guided robotics

Applied
Week 11-14
Reading Wk/ Exam on Week 15-16

Last week: Image Processing

Completed Image Processing

✓Thresholding & Histogram Processing

✓Filtering

✓Edge & Corner Detection

✓Interest Points/ Feature Description

✓Lines & Shapes

Image Processing Quick Recap

Thresholding & Histogram Processing

6

Filtering

7

Edge & Corner Detection

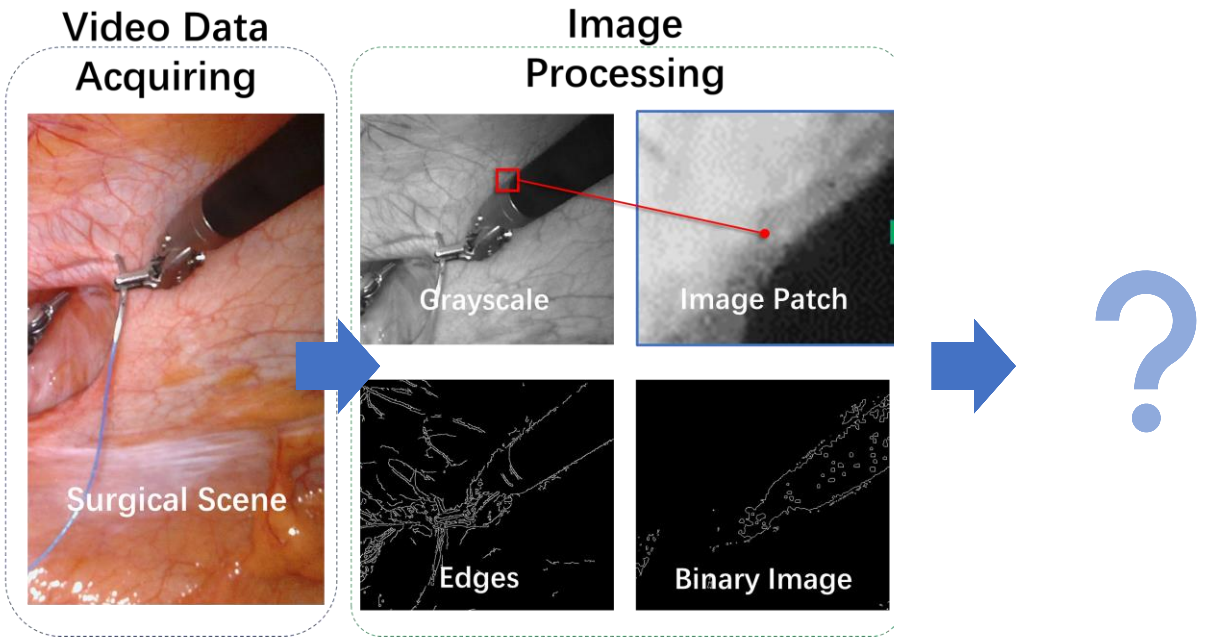
8

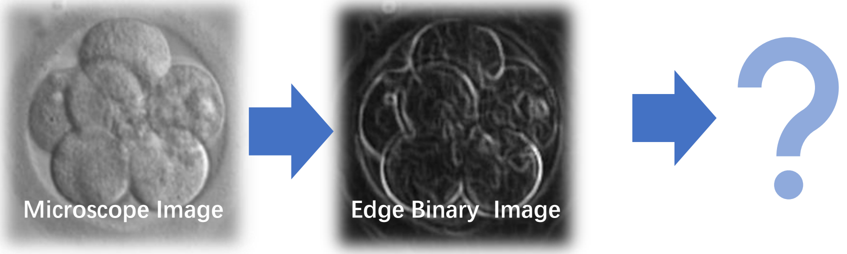
Interest Points/ Feature Description

9

Detection of Line and Shape

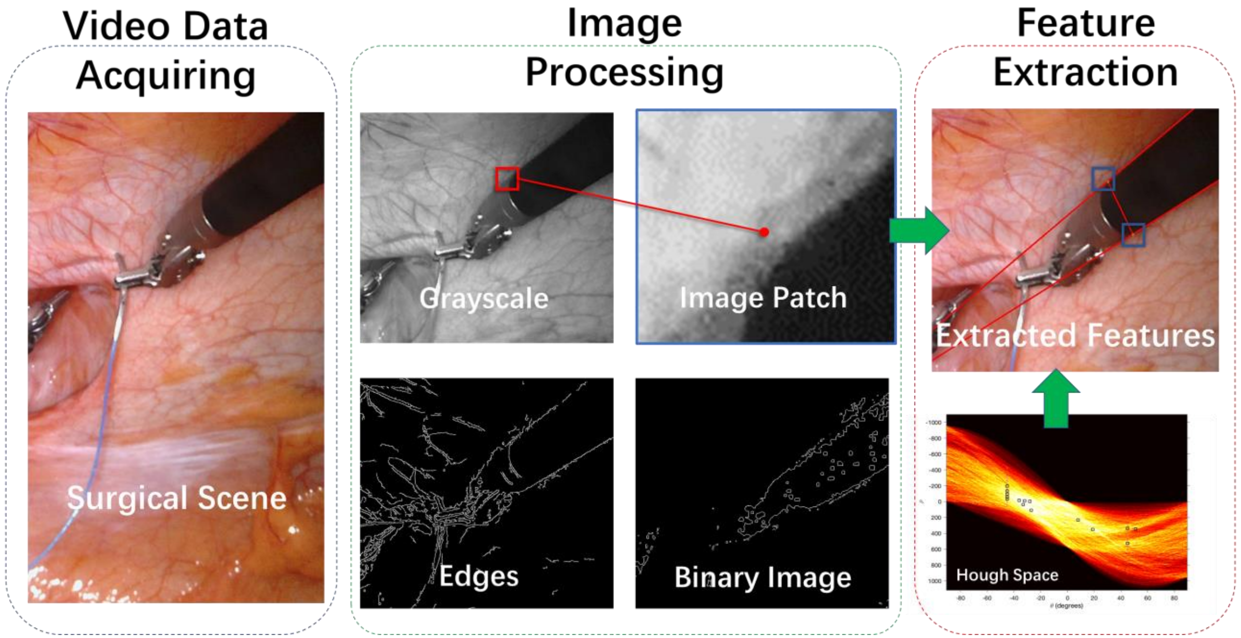
- After detecting the edges and local interest points, how do we detect lines and other geometries?
  - ➢A problem of pattern recognition





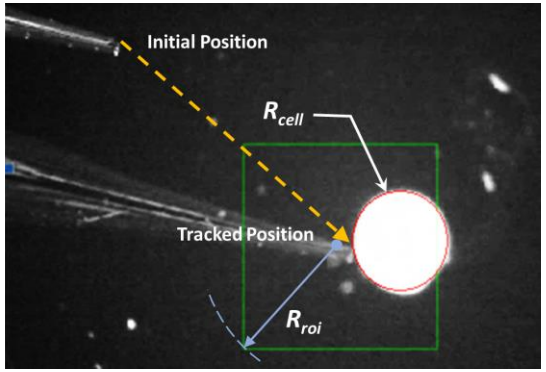
Detection of Line and Shape

- After detecting the edges and local interest points, how do we detect lines and other geometries?
  - ➢A problem of pattern recognition

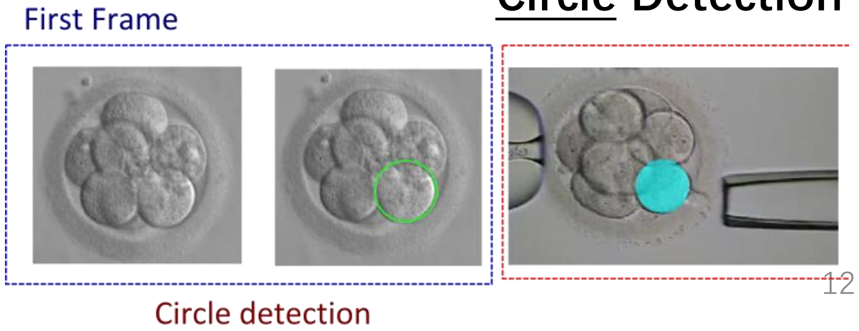


Line Detection +Template Matching

Circle Detection + Template Matching



Circle Detection



Huang, J., Li, X., Kesavadas, T. and Yang, L., 2019, July. Feature Extraction of Video Data for Automatic Visual Tool Tracking in Robot Assisted Surgery. In *Proceedings of the 2019 4th International Conference on Robotics, Control and Automation* (pp. 121-127).

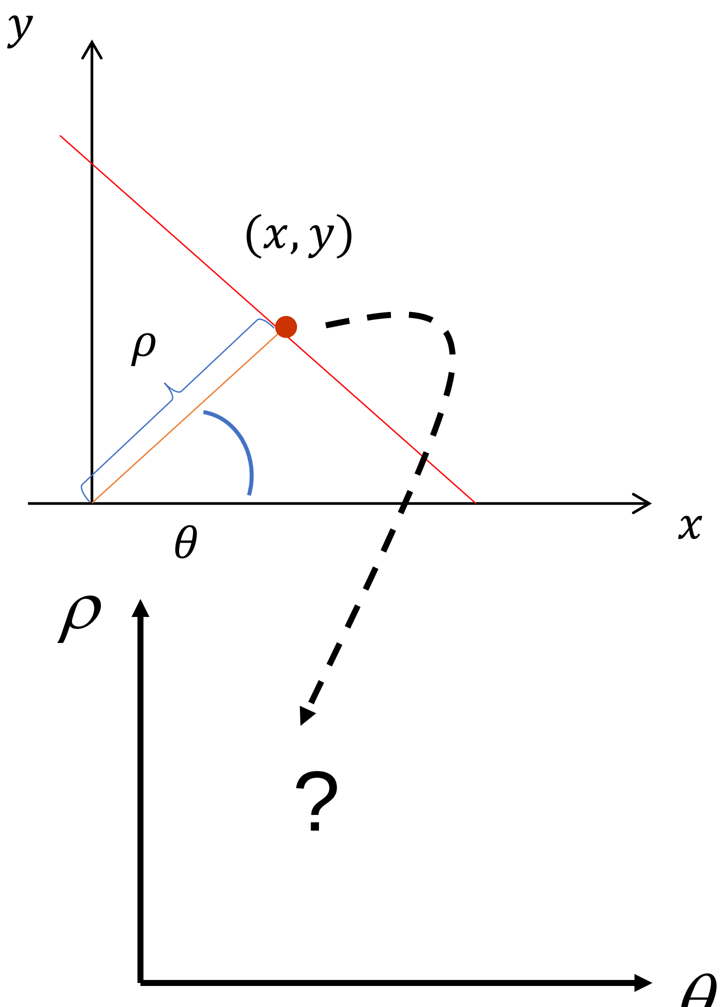
Hough Transform

- Elegant method for direct object recognition
- Edges need not be connected
- Key concept: Hough space
- Key idea: “vote” for the possible model

ZJU-UIUC INSTITUTE
Zhejiang University/University of Illinois at Urbana-Champaign Institute
浙江大学伊利诺伊大学厄巴纳香槟校区联合学院

Hough Space – $(\boldsymbol{\rho}, \boldsymbol{\theta})$ space

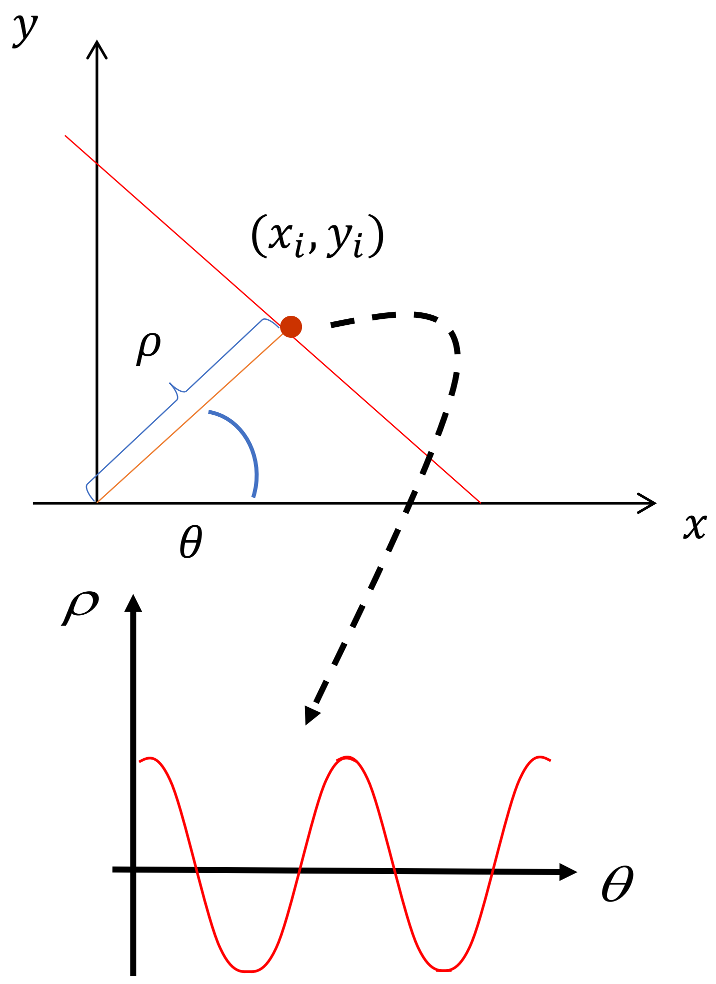
- For a line with equation *l*: *y*=m*x*+c
- How to map onto the Hough Space (A.K.A parameter space)?
  - $\rho = x\cos\theta + y\sin\theta$
- Line equation: $\rho = x\cos\theta + y\sin\theta$
  - Parameters: ρ and θ
  - Where $0 \le \theta \le 2\pi$



Hough Space – $(\boldsymbol{\rho}, \boldsymbol{\theta})$ space

For a point i (x_i, y_i), in Hough space there could be infinite set of (ρ, θ) defined by S_i :

$$\rho = x_i \cos\theta + y_i \sin\theta$$
$$y_i = \frac{\rho}{\sin\theta} - \frac{x_i}{\tan\theta}$$



Hough Space

Hough Space – $(\boldsymbol{\rho}, \boldsymbol{\theta})$ space

For N points (x, y), there will be $S_1 \dots S_i \dots S_N$ sinusoidal curves in the Hough space associated with the points:

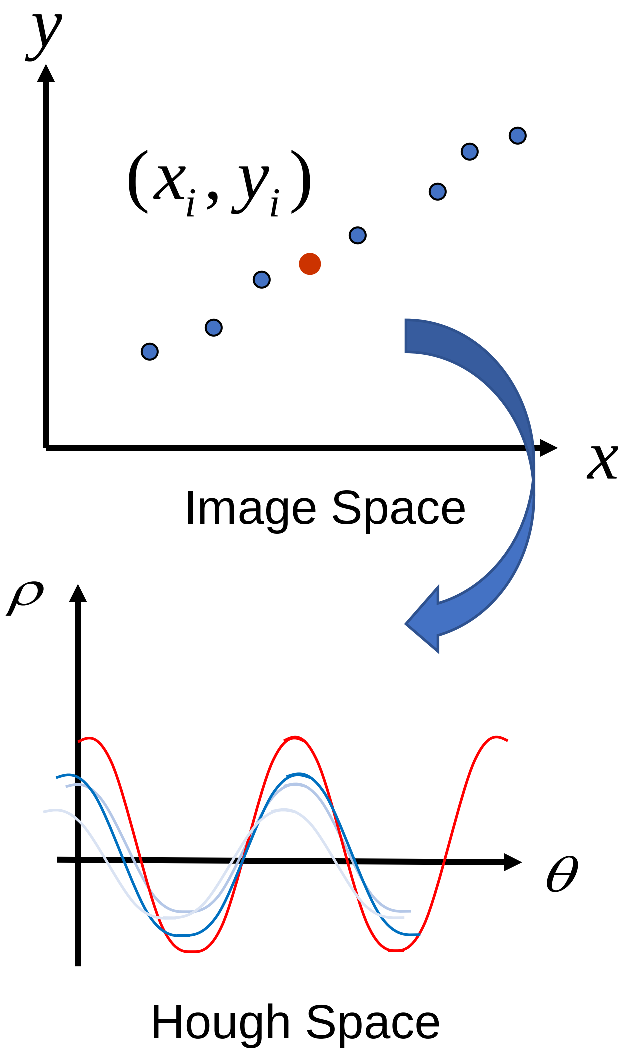
S_1 $\qquad \rho = x_1 \cos\theta + y_1 \sin\theta$

$\vdots$ $\qquad\qquad \vdots$

S_N $\qquad \rho = x_N \cos\theta + y_N \sin\theta$

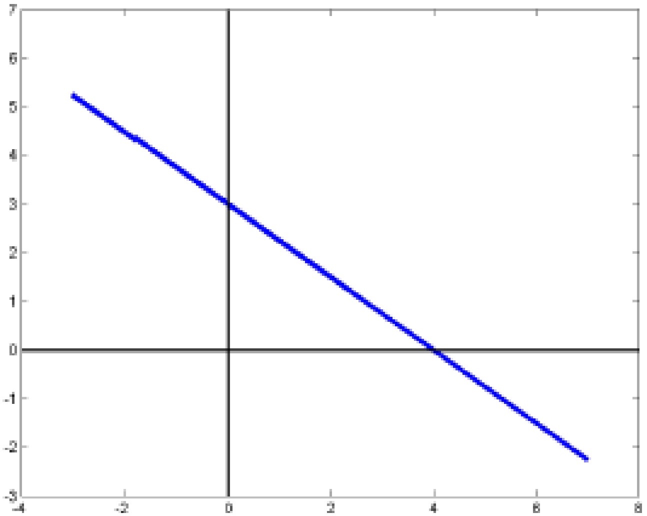
However, if the points lies on a line, there exist a common set of (ρ, θ) representing this line.

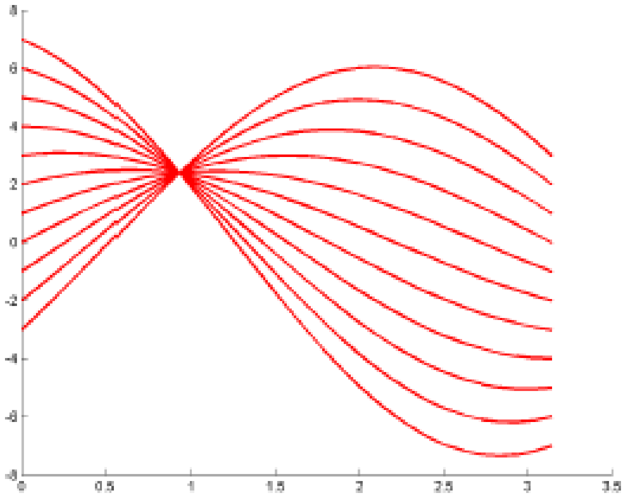
i.e.
If there exist a line that joins the points, $S_1 \dots S_i \dots S_N$ intersect at a point (ρ, θ) in the Hough space



Hough Space – $(\boldsymbol{\rho}, \boldsymbol{\theta})$ space

- $\rho = x \cos\theta + y \sin\theta$
- Points in picture → sinusoids in parameter space
- Points in parameter space → lines in picture
- There will be a unique intersection point if the points in the picture form a straight line





But how to choose the solution if there are noise?
In practice, is it likely for all points to lie perfectly on a line i.e. all sinuosoidal curves intersect perfectly on a line.

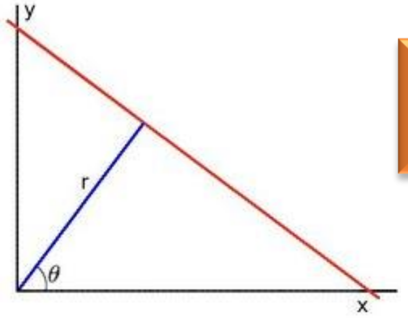
Quantizing parameter space and Voting

- Discretize the (ρ, θ) space
  - For each point (x_i, y_i), compute only for a finite set of angles $\theta = \theta_1, \theta_2, \ldots, \theta_N$
  - For each θ_j, obtain $\rho_{ij} = x_i \cos\theta_j + y_i \sin\theta_j$
- Create a matrix, called the accumulator matrix
  - Each column corresponds to angles $\theta = \theta_1, \theta_2, \ldots, \theta_N$
  - Each row corresponds to the “bins” (intervals) of the resulting distance ρ
- Voting:
  - For each point in the image and for each θ_j, compute the ρ_{ij}, and increment the corresponding element of the accumulator matrix
  - Highest value means highest “vote”

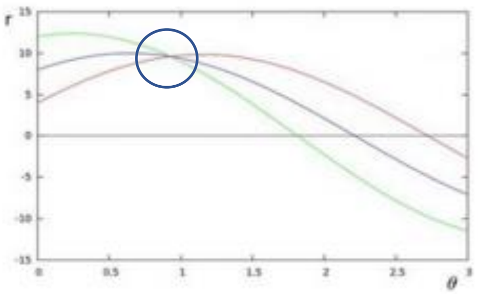
Quantizing parameter space and Voting

- Voting (continue)
  - If more than one line, you can set a threshold value (number of vote) to obtain more lines
  - For example, if number of votes (ie value in that element) is more than the threshold value, then consider that (ρ, θ) to be a line

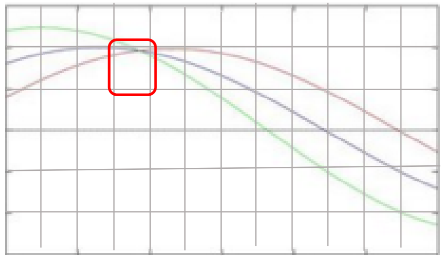
Cartesian Coord. → Polar Coord. → Discrete Polar Coord



Feature Extraction



Discretize Coord. Space



Detection of other shapes

- Hough Transform can be generalized to find other shapes like circles and ellipses.
- However, the computational complexity increases
  - More computational time is required

Hough circle transform

- Equation of circle:

$$(x - a)^2 + (y - b)^2 = R^2$$

  - (a, b) is the center of the circle
  - R is the radius of the circle
- Parameters: a, b, R
- Rewriting the equation:

$$(a - x)^2 + (b - y)^2 = R^2$$

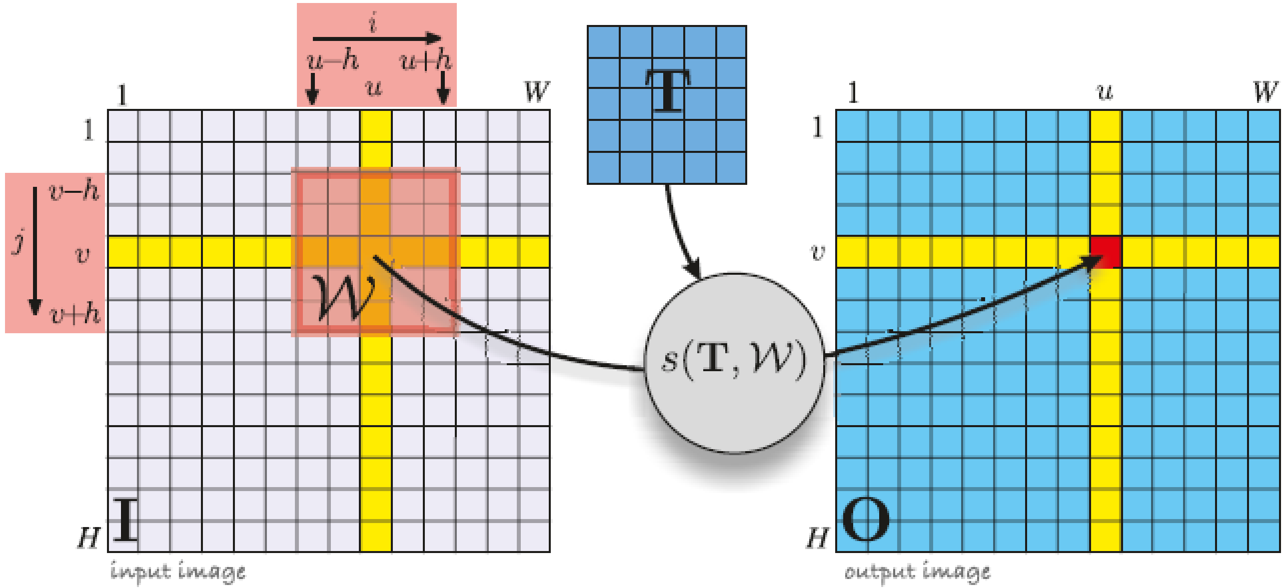
Object Tracking

Object Tracking

- In our context: localizing a target in the image over time (throughout frames)
- Possible approaches:
  - Locate objects in individual frames independently (E.g. template match)
  - Estimate motion of objects/ change in image over time

Object Tracking

- Template Matching
  - Compare patches (on image) against template (of target)



Object Tracking

- Template Matching
  - Compare patches (on image) against template (of target)
- How?
  - Recall what was learned in feature descriptor with neighbourhood block

Object Tracking

- Template Matching
  - Compare patches (on image) against template (of target)
- Target tracked based on similarity
- How do we measure similarity?
  - Compute differences?
  - Compute correlation?

Discussed previously in *feature description* using neighborhood block

Object Matching

- Sum of Square Differences

$$\boldsymbol{SSD}(\boldsymbol{u},\boldsymbol{v}) = \sum_{\boldsymbol{p}}^{\boldsymbol{P}}\sum_{\boldsymbol{q}}^{\boldsymbol{Q}}[\boldsymbol{g}(\boldsymbol{p},\boldsymbol{q}) - \boldsymbol{f}(\boldsymbol{p}+\boldsymbol{u},\boldsymbol{q}+\boldsymbol{v})]^{2}$$

sum of square differences between a p x q template $\boldsymbol{g}$ and the neighbourhood of patch $\boldsymbol{f}$ centred at (u,v)

Object Matching

- Cross-Correlation

$$w_{cc}(u,v)=\sum_{p=0}^{P}\sum_{q=0}^{Q} g(p,q)f(p+u,q+v)$$

cross correlation between a *p* x *q* template *g* and the neighbourhood of patch *f* centred at (u,v)

Object Matching

- Normalized Cross-Correlation

$$w_{ncc}(u,v)=\frac{\sum_{p=0}^{P}\sum_{q=0}^{Q}\left(g(p,q)-\bar{g}\right)\left(f(p+u,q+v)-\bar{f}(u,v)\right)}{\left[\left(\sum_{p=0}^{P}\sum_{q=0}^{Q}\left(g(p,q)-\bar{g}\right)^2\right)\left(\sum_{p=0}^{P}\sum_{q=0}^{Q}\left(f(p+u,q+v)-\bar{f}(u,v)\right)^2\right)\right]^{0.5}}$$

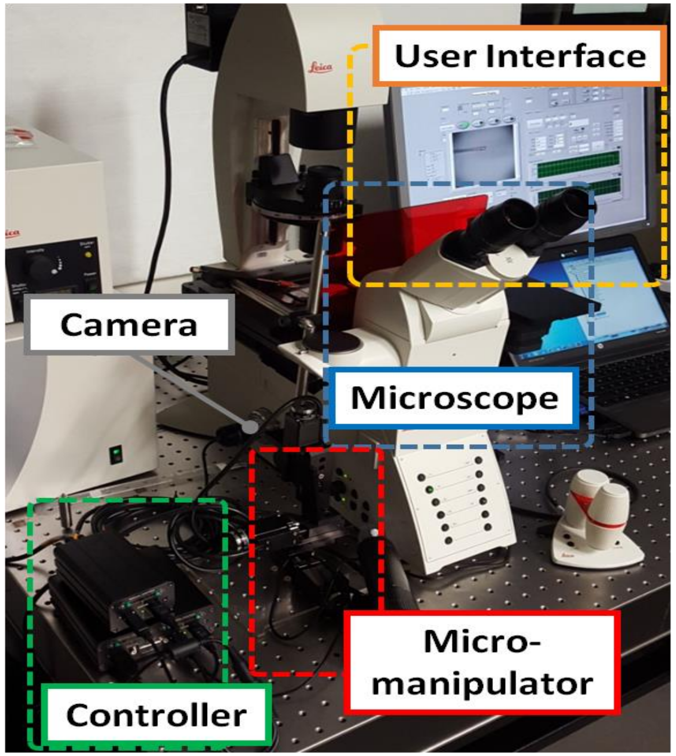
normalize using local mean and variance

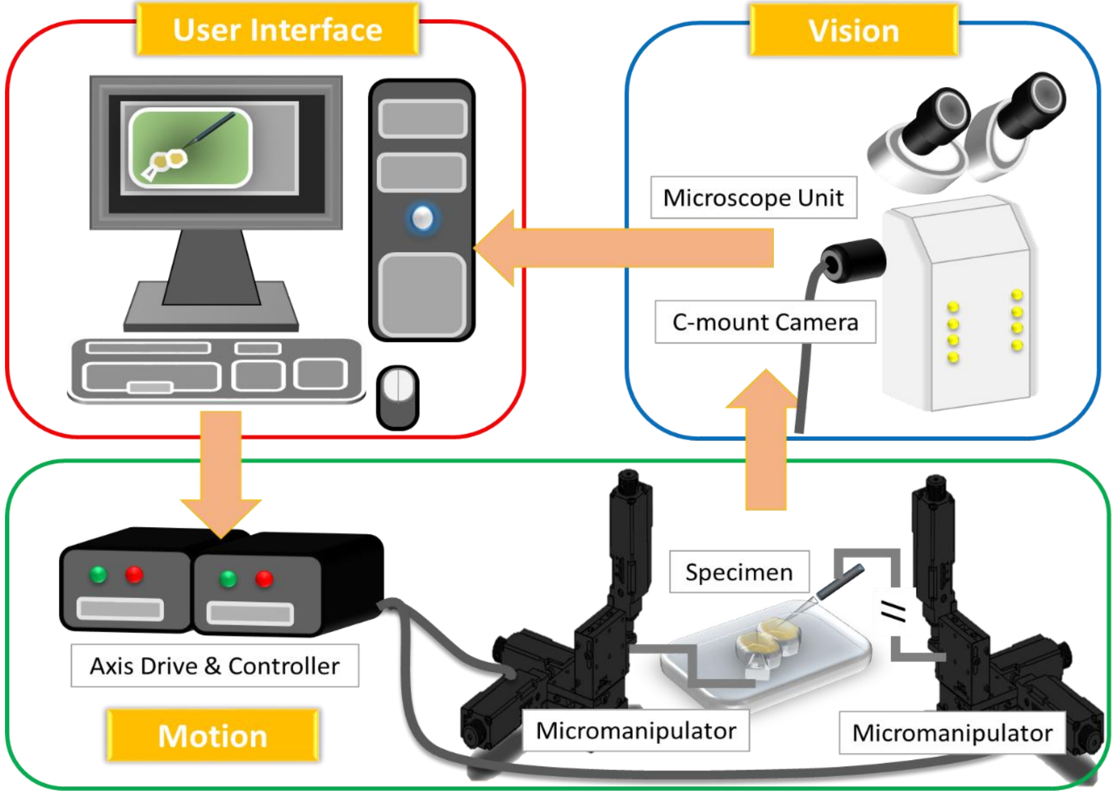
Object Matching

- Depending on image size can be done in frequency domain
- Conventional similarity measurement alone is usually insufficient
  - Need to be scale- and rotation-invariant in many robot applications
  - Commonly work with image pyramid or coarse-to-fine strategies

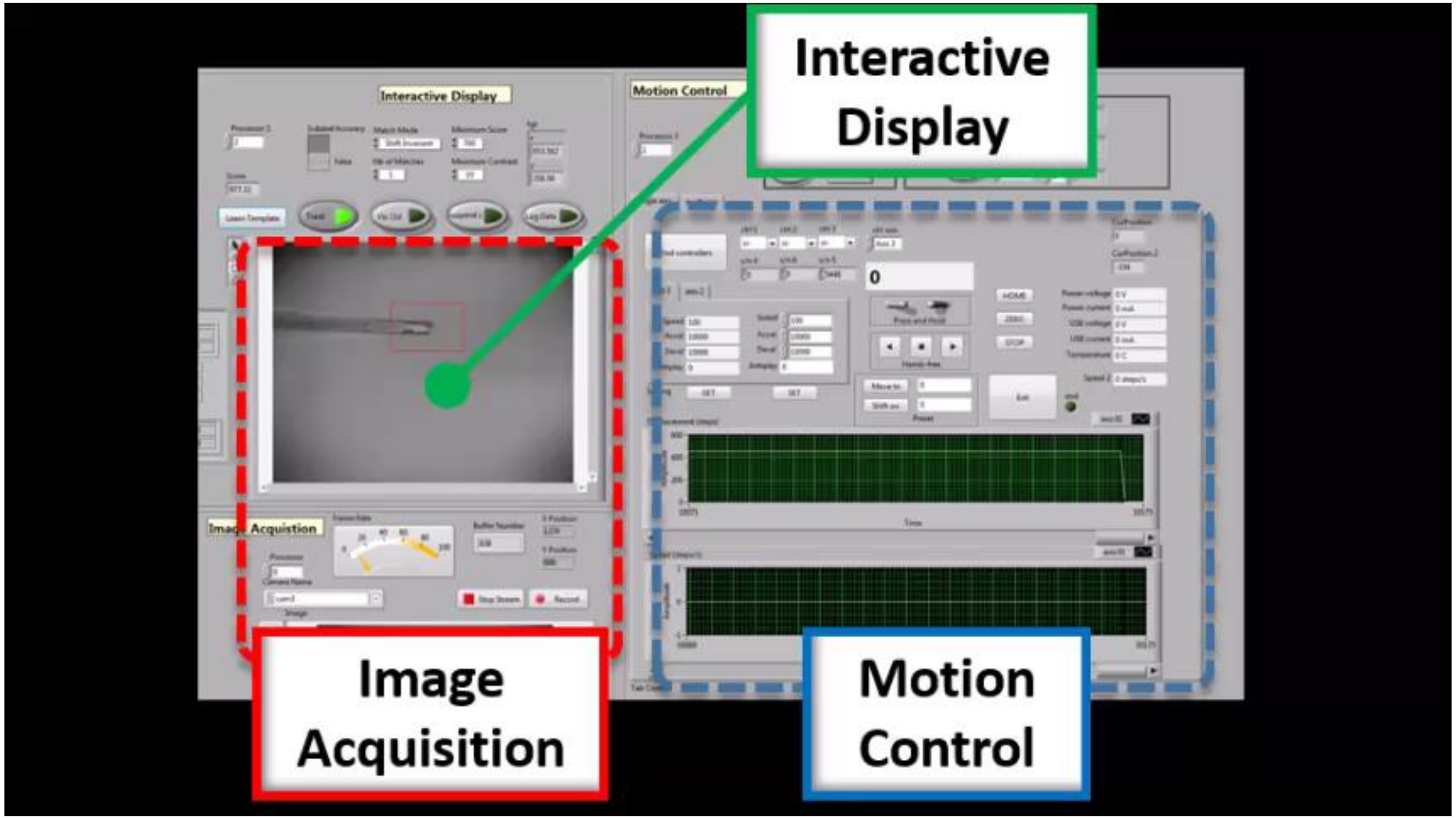
Application Example of Template Matching

- Vision-based Control of Micromanipulator under Microscope



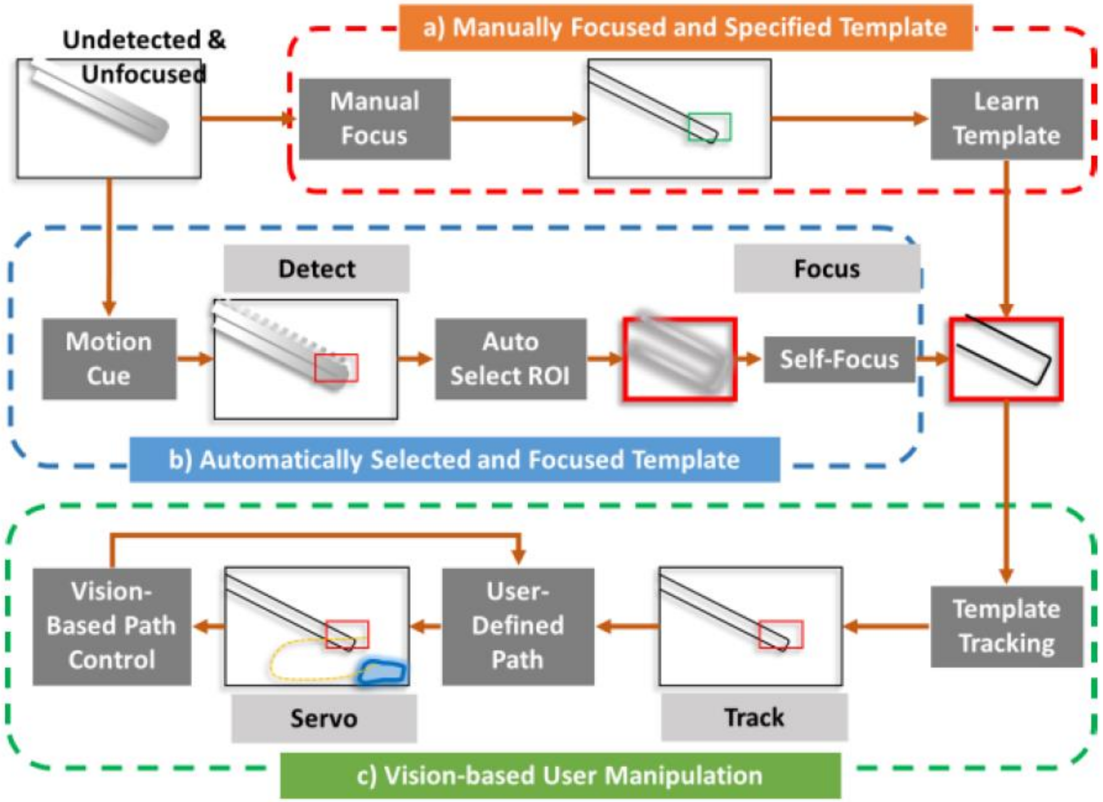


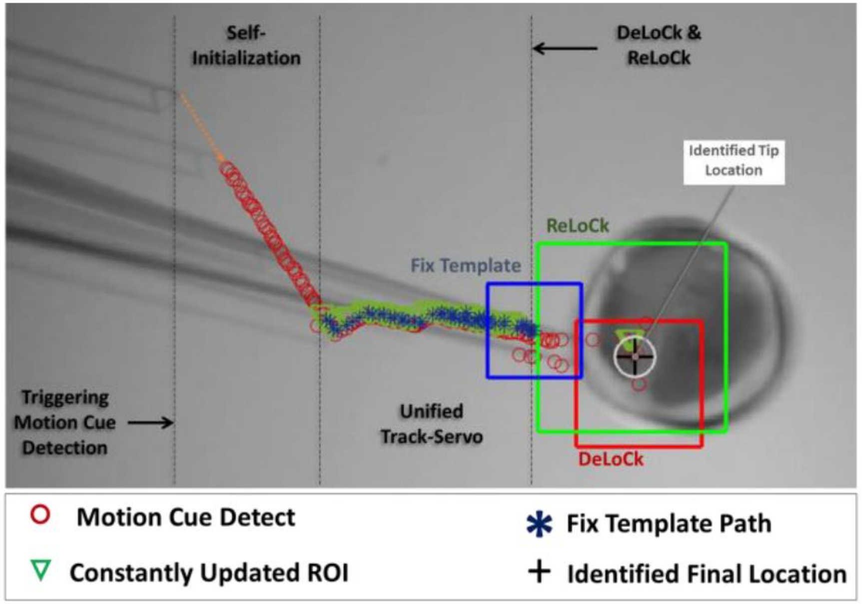
Application Example of Template Matching



1. L. Yang, K. Youcef-Toumi, U. Tan, "Towards Automatic Robot-Assisted Microscopy: An Uncalibrated Approach for Robotic Vision-Guided Micromanipulation," in Intelligent Robots and System, IROS 2016, Daejeon, Korea, 2016.

Application Example of Template Matching

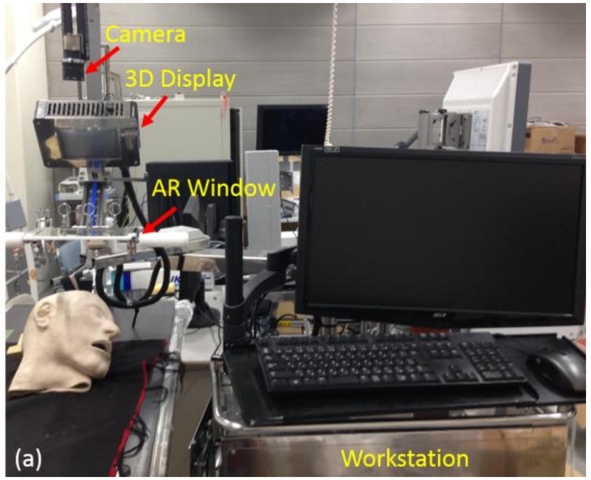


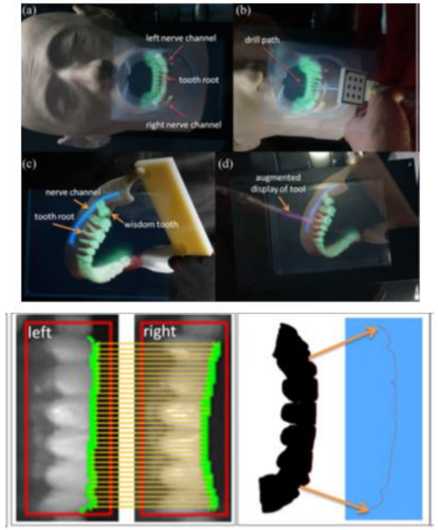


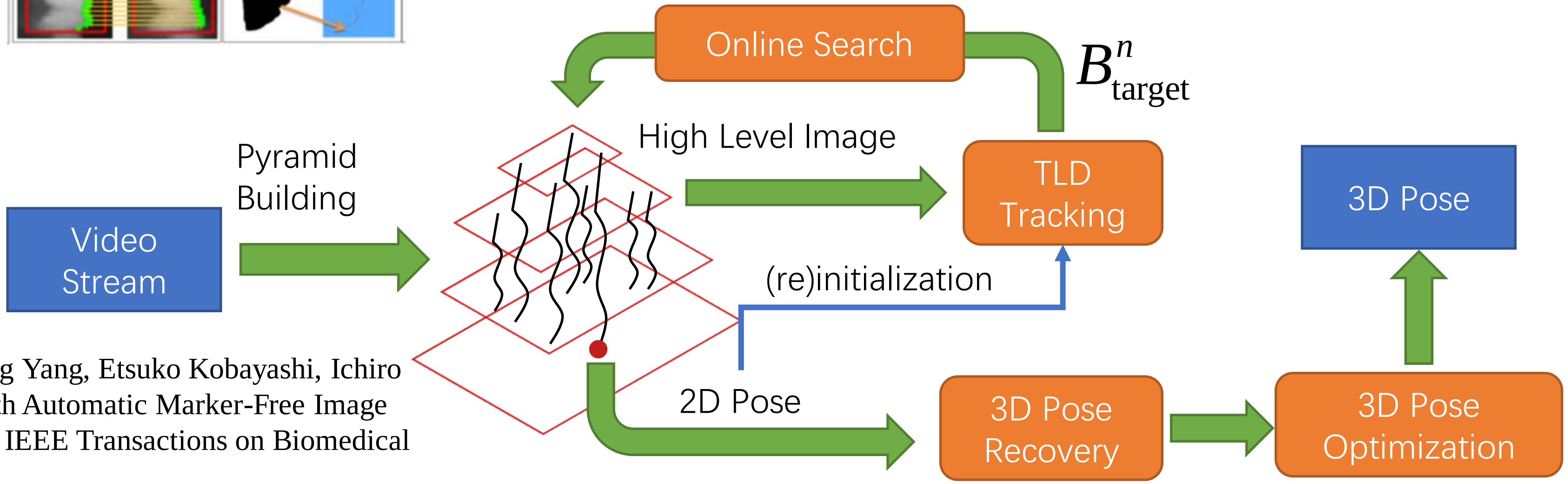
L. Yang, I. Paranawithana, K. Youcef-Toumi, U. Tan, "Self-initialization and recovery for uninterrupted tracking in vision-guided micromanipulation," in Intelligent Robots and System, IROS 2017, Vancouver, Canada, 2017

Application Example of Template Matching

- Example: Image registration in AR application







Junchen Wang, Hideyuki Suenaga, Kazuto Hoshi, Liangjing Yang, Etsuko Kobayashi, Ichiro Sakuma, Hongen Liao. Augmented Reality Navigation With Automatic Marker-Free Image Registration Using 3-D Image Overlay for Dental Surgery. IEEE Transactions on Biomedical Engineering, 2014, 61(4): 1295-1304

Image Mapping

Image Mapping

- Relating different viewpoints of on a common scene
- Image registration and geometric transformation
- How?
  - Recall the techniques learn so far: detect, describe, match···
  - Then, solve for transformation (homography) based on a specific model: translation, rigid, similarity, affine, projective ······

Image Mapping

- Homography
  - Relationship (i.e. transformation; mapping) between two (planar) images
- Geometric image transformation

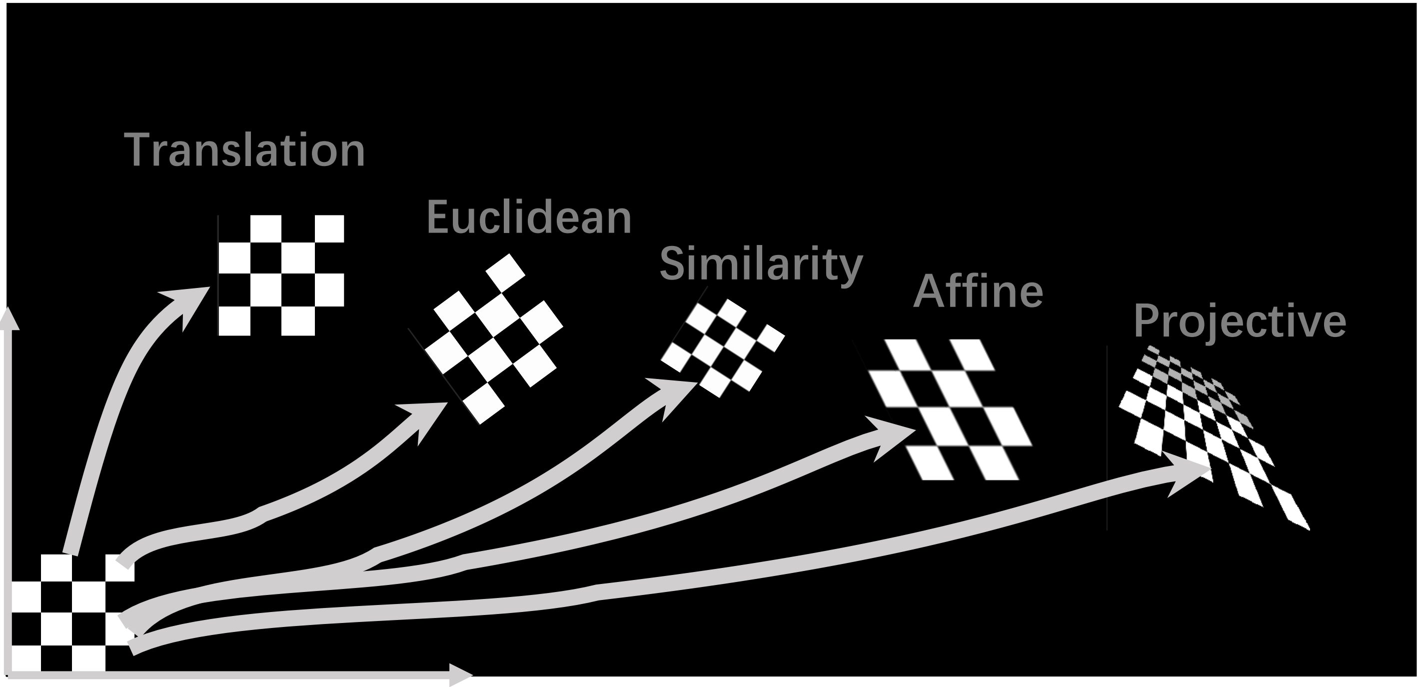


Image Mapping

- Geometric image transformation: Types, Representations, DOF, Attributes

Name	Matrix	# D.O.F.	Preserves:
translation	$\begin{bmatrix} I \mid t \end{bmatrix}_{2\times 3}$	2	orientation $+\cdots$
rigid (Euclidean)	$\begin{bmatrix} R \mid t \end{bmatrix}_{2\times 3}$	3	lengths $+\cdots$
similarity	$\begin{bmatrix} sR \mid t \end{bmatrix}_{2\times 3}$	4	angles $+\cdots$
affine	$\begin{bmatrix} A \end{bmatrix}_{2\times 3}$	6	parallelism $+\cdots$
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times 3}$	8	straight lines

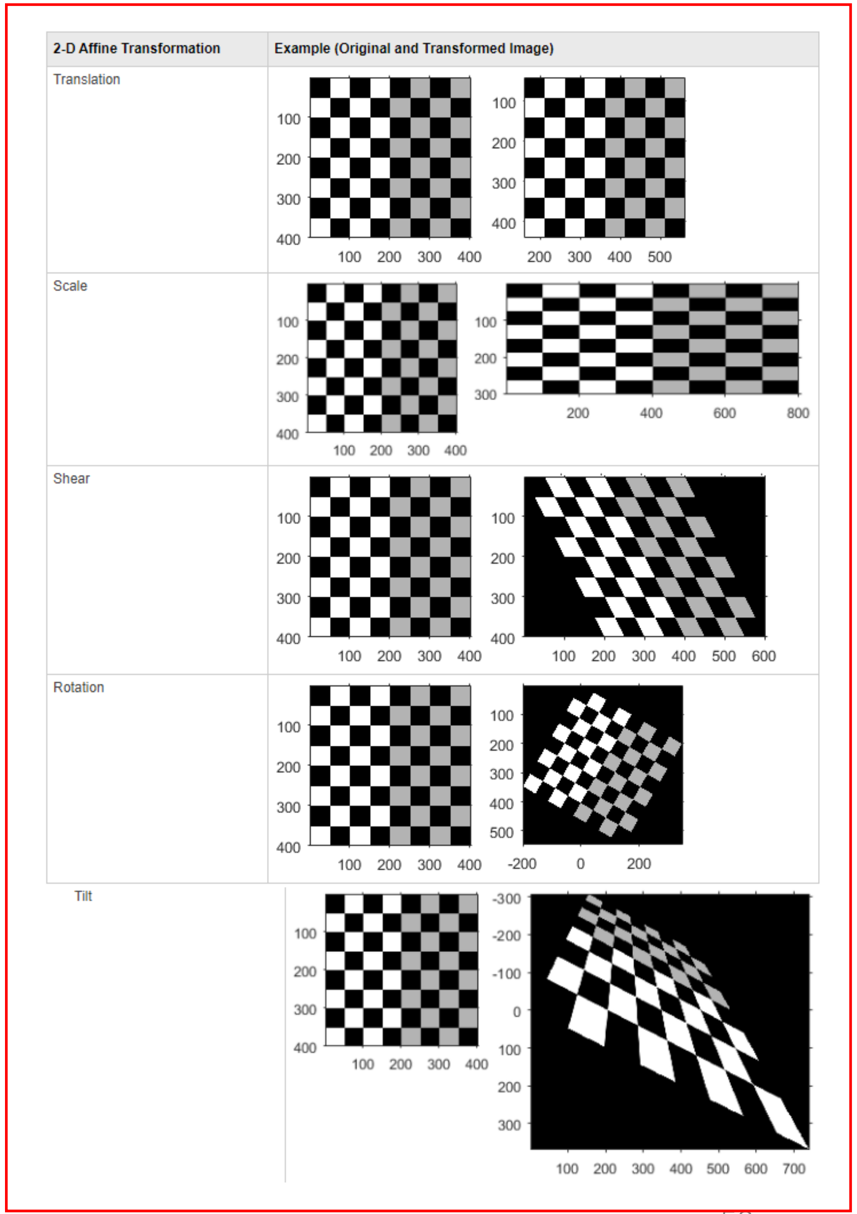
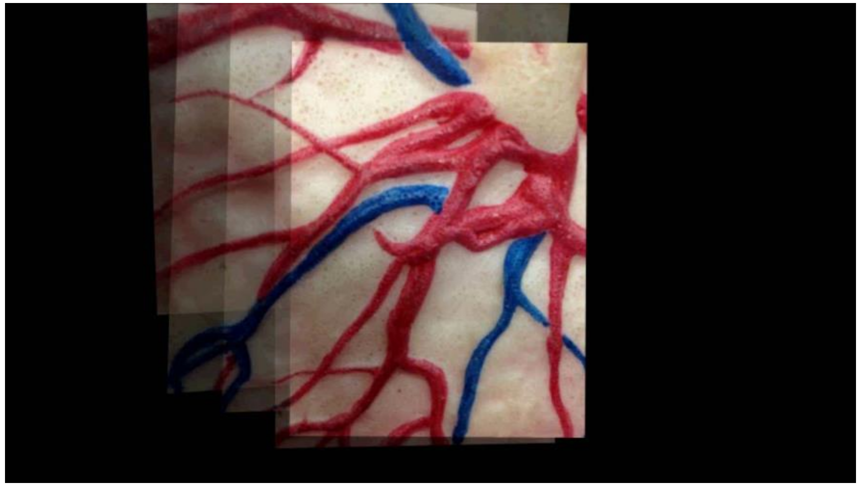
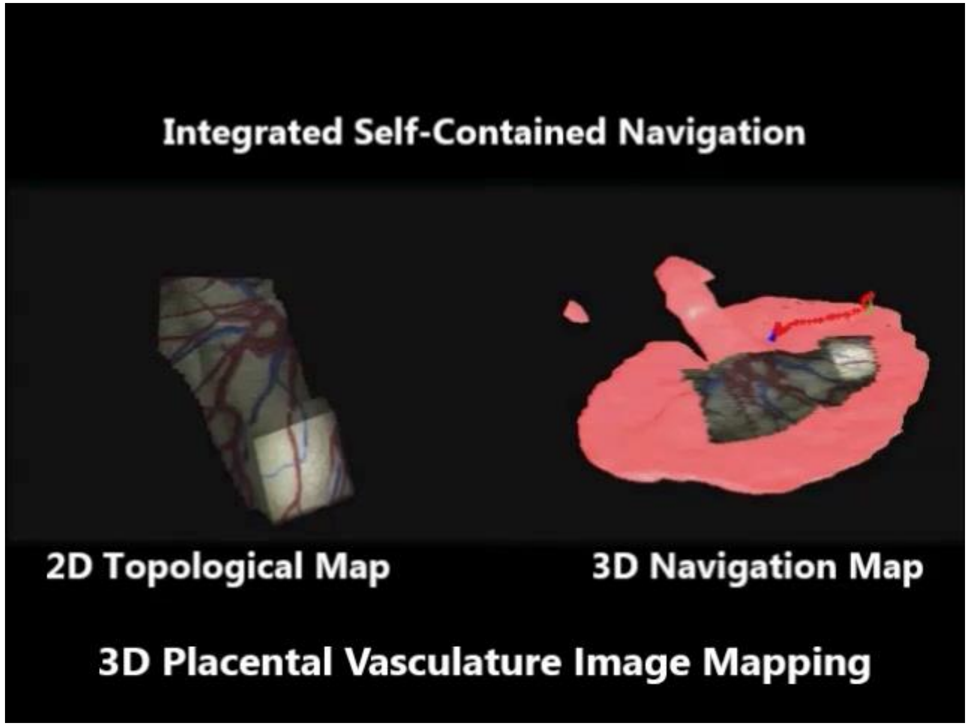


Image Mapping: Application Examples

Constructing a map for the environment



Visualization and Navigation

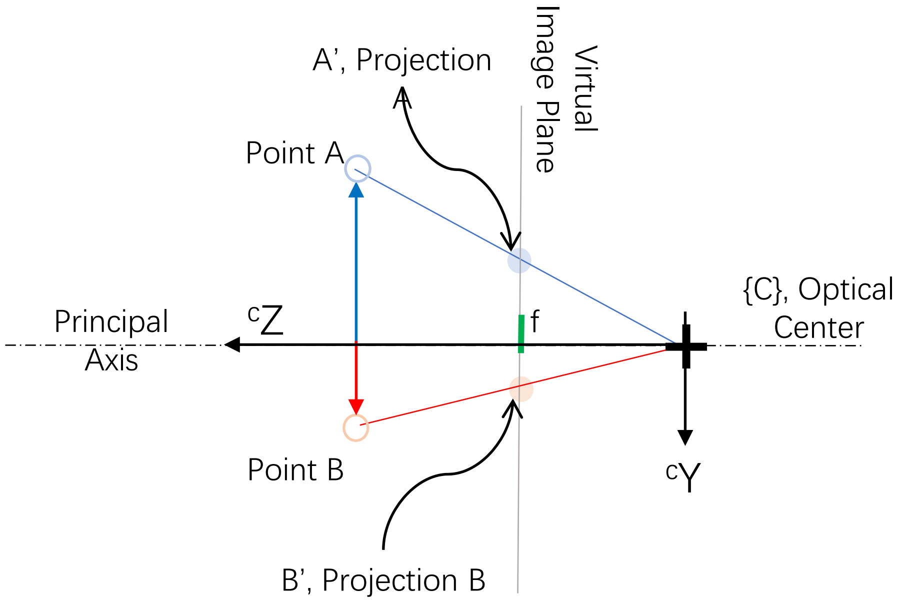


Camera Model

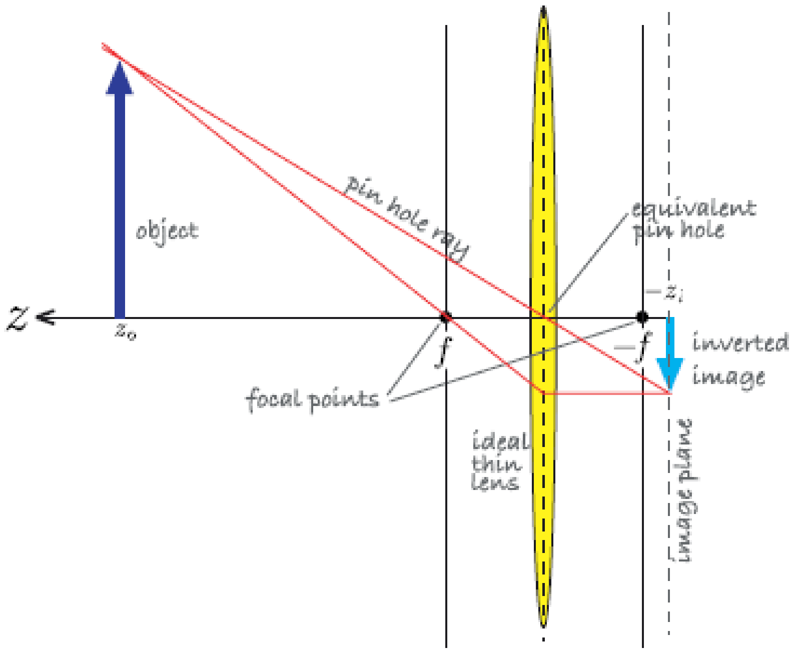
ECE 470 Introduction to Robotics

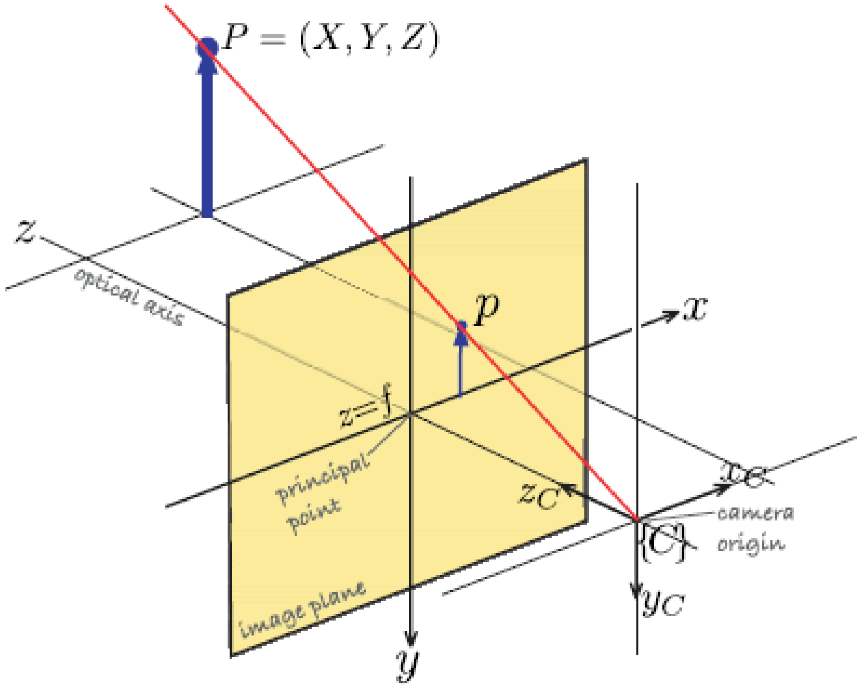
Recall: Camera Model

- Central Projection Camera Model
  - A simplified model for camera geometry



Camera Model





$$x = f_x \frac{X}{Z}$$

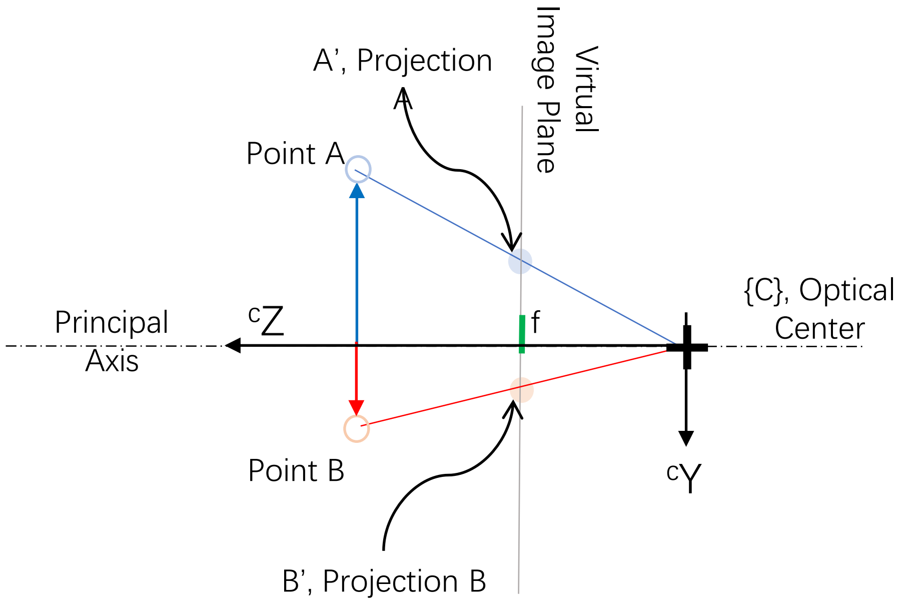
$$y = f_y \frac{Y}{Z}$$

Corke, Peter. *Robotics, vision and control: fundamental algorithms in MATLAB*.

Matrix Representation: Intrinsic Matrix

- Intrinsic Matrix, $[K]$
  - focal length: $(f_x, f_y)^T$
  - principal point: $(i_c, j_c)^T$
  - skew coefficient: a

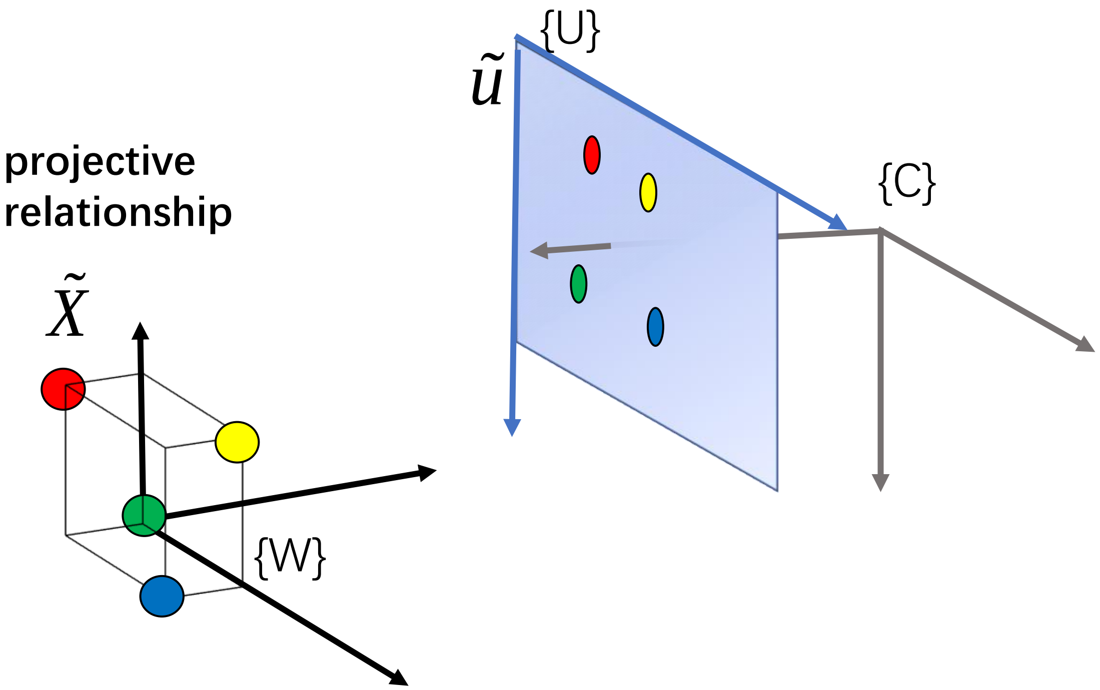
$$K = \begin{bmatrix} f_x & a & i_c \\ 0 & f_y & j_c \\ 0 & 0 & 1 \end{bmatrix}$$



What about a moving camera?

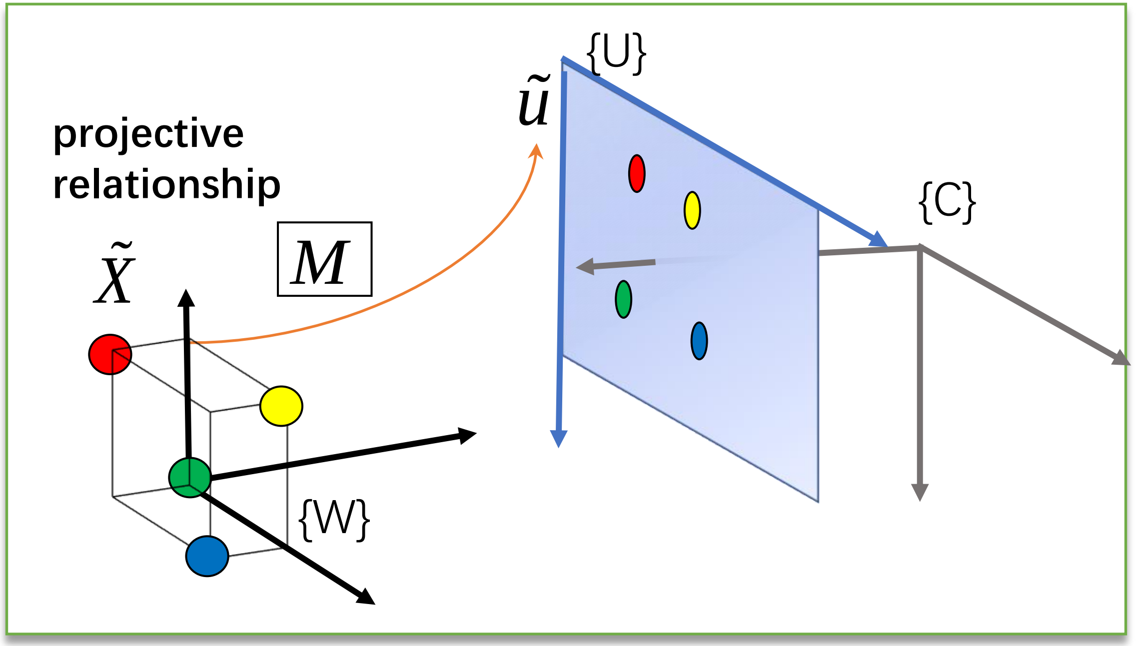
Matrix Representation: Extrinsic Matrix

- Extrinsic Matrix, ${}^{c}[\,R \mid t\,]$
  - R: Orientation of world reference frame w.r.t. camera coord.
  - t: Position offset of world reference frame w.r.t. camera coord.



Camera Matrix

- Camera Matrix, *M*
  - Relates world with image coord. System
  - 2 Components:
    - Extrinsic Matrix
    - Intrinsic Matrix



<u>Camera Matrix, *M*</u>

For a given set of points

${}^{W}\tilde{X}$ in 3D,

the projected set of points can be expressed as

$s\tilde{u} = M\,{}^{W}\tilde{X}.$ where $M = K\,{}^{C}[R\,|\,t]_{W}$,

K = intrinsic matrix

$[R\,|\,t]$ = extrinsic matrix