Lecture 7

Linear Approach to Camera Calibration

ECEN5283 Computer Vision

Dr. Guoliang Fan

Oklahoma State University

Goals

To use a linear approach to calibrate a camera.

To be ready for Project

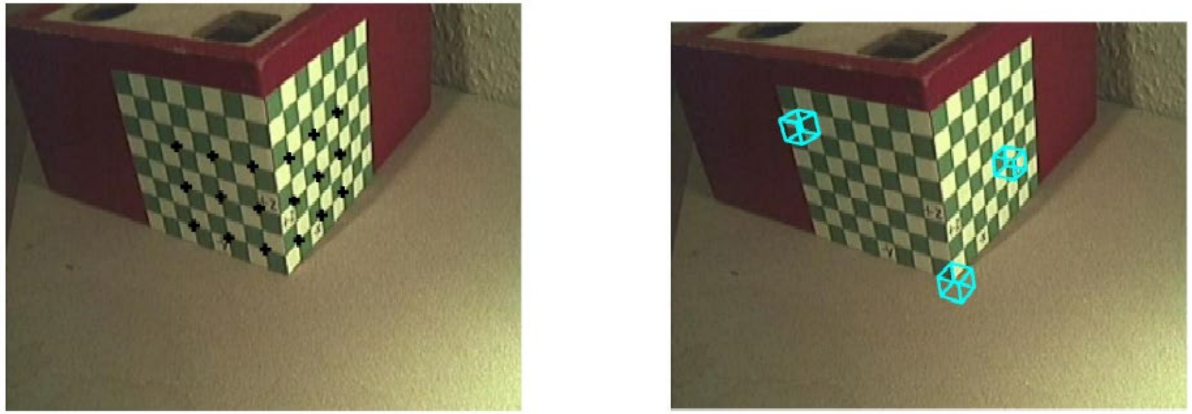


Figure 1: An image of a calibration rig, left with calibration points, right with cubes to check the calibration.

Camera Projection Matrix

A projection matrix is written explicitly as a function of both intrinsic and extrinsic parameters as follows

- Five intrinsic parameters $\alpha,\ \beta, u_0, v_0, \theta$
- Six extrinsic ones (three angles and three coordinates of t).

$$\mathbf{p} = \frac{1}{z} M\mathbf{P} \text{ where } M = K\begin{pmatrix} R & t \end{pmatrix} \quad K = \begin{pmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad R = \begin{pmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{pmatrix} \text{ and } t = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

$$M = K\begin{pmatrix} R & t \end{pmatrix} = \begin{pmatrix} KR & Kt \end{pmatrix} = \begin{pmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\,\mathbf{r}_2^T + u_0\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta\, t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta}\mathbf{r}_2^T + v_0\mathbf{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \mathbf{r}_3^T & t_z \end{pmatrix}_{3\times 4}$$

Camera Projection Matrix

It is important to understand the depth z is not independent of M and P.

$$\mathbf{p} = \frac{1}{z} M\mathbf{P} \quad \text{where } \mathbf{p} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \text{ and } M = \begin{pmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z} \begin{pmatrix} \mathbf{m}_1^T \mathbf{P} \\ \mathbf{m}_2^T \mathbf{P} \\ \mathbf{m}_3^T \mathbf{P} \end{pmatrix} = \frac{1}{z} \begin{pmatrix} \mathbf{m}_1 \cdot \mathbf{P} \\ \mathbf{m}_2 \cdot \mathbf{P} \\ \mathbf{m}_3 \cdot \mathbf{P} \end{pmatrix} \Rightarrow \begin{cases} z = \mathbf{m}_3 \cdot \mathbf{P} \\ u = \dfrac{\mathbf{m}_1 \cdot \mathbf{P}}{\mathbf{m}_3 \cdot \mathbf{P}} \\ v = \dfrac{\mathbf{m}_2 \cdot \mathbf{P}}{\mathbf{m}_3 \cdot \mathbf{P}} \end{cases}$$

Projection Matrix Estimation (1)

$$\begin{cases} u = \dfrac{\mathbf{m}_1 \cdot \mathbf{P}}{\mathbf{m}_3 \cdot \mathbf{P}} \\ v = \dfrac{\mathbf{m}_2 \cdot \mathbf{P}}{\mathbf{m}_3 \cdot \mathbf{P}} \end{cases}$$

Given a 3D - 2D point pair, $\mathbf{P}_i \rightarrow \begin{pmatrix} u_i \\ v_i \end{pmatrix}$

Then, we have $\begin{cases} (\mathbf{m}_1 - u_i \mathbf{m}_3) \cdot \mathbf{P}_i = 0 \\ (\mathbf{m}_2 - v_i \mathbf{m}_3) \cdot \mathbf{P}_i = 0 \end{cases}$

$$\begin{pmatrix} \mathbf{P}_i^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & -u_i \mathbf{P}_i^{\mathrm{T}} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{P}_i^{\mathrm{T}} & -v_i \mathbf{P}_i^{\mathrm{T}} \end{pmatrix}_{2 \times 12} \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix}_{12 \times 1} = \mathbf{0}$$

Projection Matrix Estimation (2)

Given *n* 3D-2D point pairs for calibration, then we have

$$\mathbf{Q} = \begin{pmatrix} \mathbf{P}_1^T & \mathbf{0}^T & -u_1\mathbf{P}_1^T \\ \mathbf{0}^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ \cdots & \cdots & \cdots \\ \mathbf{P}_n^T & \mathbf{0}^T & -u_n\mathbf{P}_n^T \\ \mathbf{0}^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{pmatrix}_{2n \times 12} \quad \text{and} \quad \mathbf{m} = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{pmatrix}_{12 \times 1}$$

Q is composed the 3D (homogeneous) and 2D coordinates of the given points.

$$\boxed{\mathbf{Qm} = \mathbf{0}} \Rightarrow \boxed{\hat{\mathbf{m}} = \arg\min_{\mathbf{m}} |\mathbf{Qm}|^2}$$

The solution can be achieved by solving the eigenvalue problem of $\mathbf{Q}^T\mathbf{Q}$.

Estimation of the Intrinsic and Extrinsic Parameters

Once the project matrix M, its expression in terms of the camera intrinsic and extrinsic parameters can be used to recover these parameters as follows.

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{b} \end{pmatrix} \text{ with } \mathbf{a}_1^T, \mathbf{a}_2^T, \text{ and } \mathbf{a}_3^T \text{ denoting the rows of } \mathbf{A}.$$

What is ρ?

$$\rho \begin{pmatrix} \mathbf{A} & \mathbf{b} \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} = \begin{pmatrix} \mathbf{K}\mathbf{R} & \mathbf{K}\mathbf{t} \end{pmatrix}$$

$$\mathbf{K} = \begin{pmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_0 \\ & 0 & 1 \end{pmatrix} \qquad R = \begin{pmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{pmatrix} \text{ and } t = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

Estimation of Intrinsic Parameters

$$\rho \mathbf{A} = \mathbf{K} \cdot \mathbf{R} \quad \Leftrightarrow \rho \begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \end{pmatrix} = \begin{pmatrix} \alpha \mathbf{r}_1^T - \cot\theta \mathbf{r}_2^T + u_0 \mathbf{r}_3^T \\ \dfrac{\beta}{\sin\theta} \mathbf{r}_2^T + v_0 \mathbf{r}_3^T \\ \mathbf{r}_3^T \end{pmatrix}$$

$$\begin{cases} \rho = \varepsilon / |\mathbf{a}_3| \\ u_0 = \rho^2 (\mathbf{a}_1 \cdot \mathbf{a}_3) \\ v_0 = \rho^2 (\mathbf{a}_2 \cdot \mathbf{a}_3) \end{cases}$$

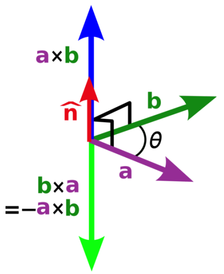
$(\varepsilon = 1 \text{ or } -1 : \text{image plan and scene are on the same or different sides})$

$$\begin{cases} \cos\theta = -\dfrac{(\mathbf{a}_1 \times \mathbf{a}_3) \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}{|\mathbf{a}_1 \times \mathbf{a}_3| |\mathbf{a}_2 \times \mathbf{a}_3|} \\ \alpha = \rho^2 |\mathbf{a}_1 \times \mathbf{a}_3| \sin\theta \\ \beta = \rho^2 |\mathbf{a}_2 \times \mathbf{a}_3| \sin\theta \end{cases}$$

$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos\theta$ (inner product)

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}$$

(outer product/cross product)



Estimation of the Extrinsic Parameters

$$\rho \begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \end{pmatrix} = \begin{pmatrix} \alpha \mathbf{r}_1^T - \cot\theta\, \mathbf{r}_2^T + u_0 \mathbf{r}_3^T \\ \dfrac{\beta}{\sin\theta} \mathbf{r}_2^T + v_0 \mathbf{r}_3^T \\ \mathbf{r}_3^T \end{pmatrix} \Leftrightarrow \begin{cases} \mathbf{r}_1 = \dfrac{1}{|\mathbf{a}_2 \times \mathbf{a}_3|}(\mathbf{a}_2 \times \mathbf{a}_3) \\ \mathbf{r}_3 = \rho \mathbf{a}_3 \qquad \text{(rotation vector)} \\ \mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1 \end{cases}$$

$$\rho \begin{pmatrix} \mathbf{A} & \mathbf{b} \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} \to \mathbf{K}\mathbf{t} = \rho \mathbf{b} \Rightarrow \mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \rho \mathbf{K}^{-1} \mathbf{b} \quad \text{(shift vector).}$$

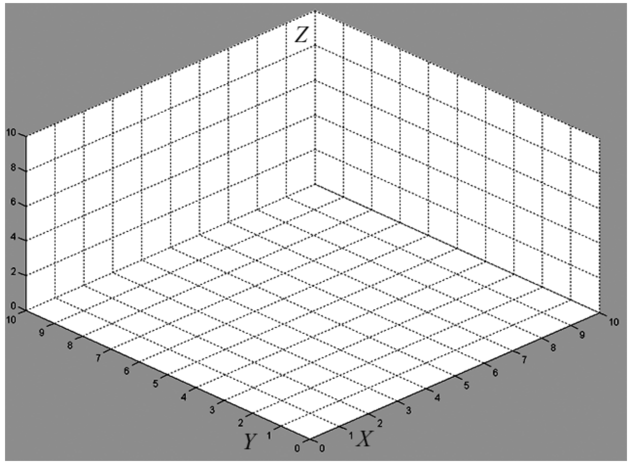
$$\mathbf{K} = \begin{pmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_0 \\ & 0 & 1 \end{pmatrix}$$

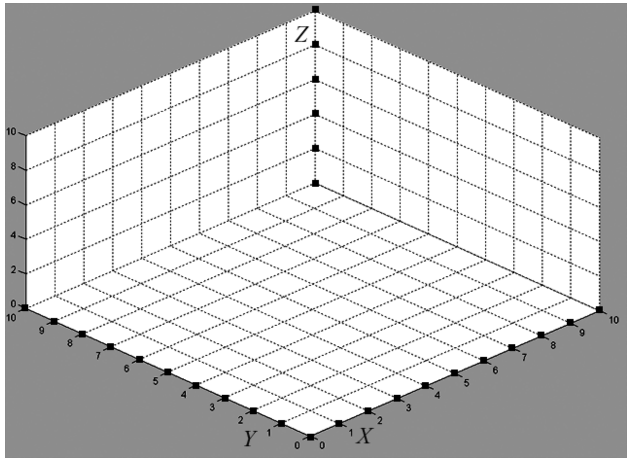
Project 1 (Due: Feb. 9)

Observe.dat contains a set of 2D pixel coordinates in image "*test_image.bmp*"

Model.dat includes a set of 3D coordinates in a world coordinate system

It is suggested to first show these points on the image to ensure the correctness of the coordinate system in your program.





How to write a PPT report?

Project Objective (1 page)

- What is the objective of this project?
- What kind of test data and tools have been given?

Technical Background and Implementation (~5 pages)

- What is the basic theory involved in this project?
- Any major equations or any useful illustrations?

Experimental Results (~5 pages)

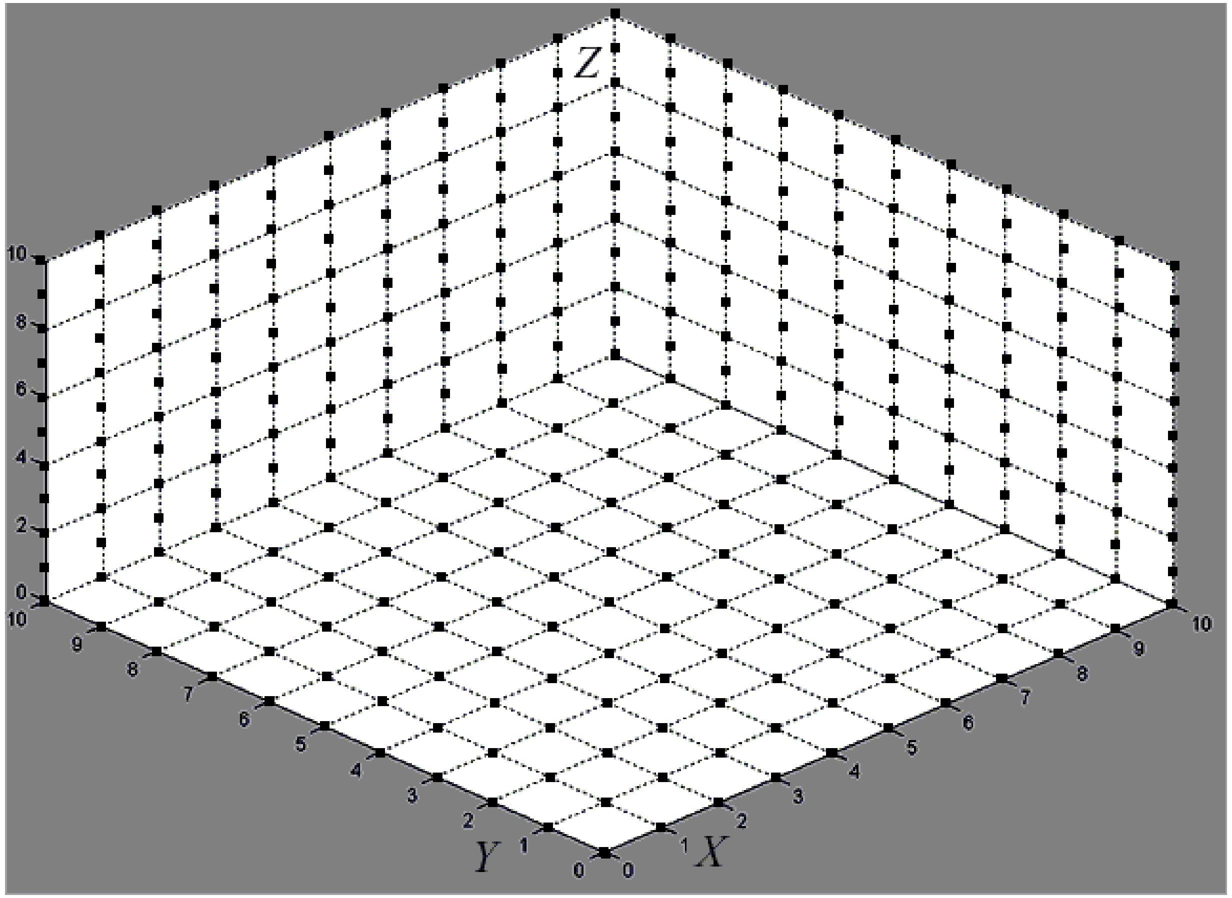
- What do see in your experiment and what do they mean?
- Try different parameter settings and what is the optimal one, why?

Discussion and Conclusion (1 page)

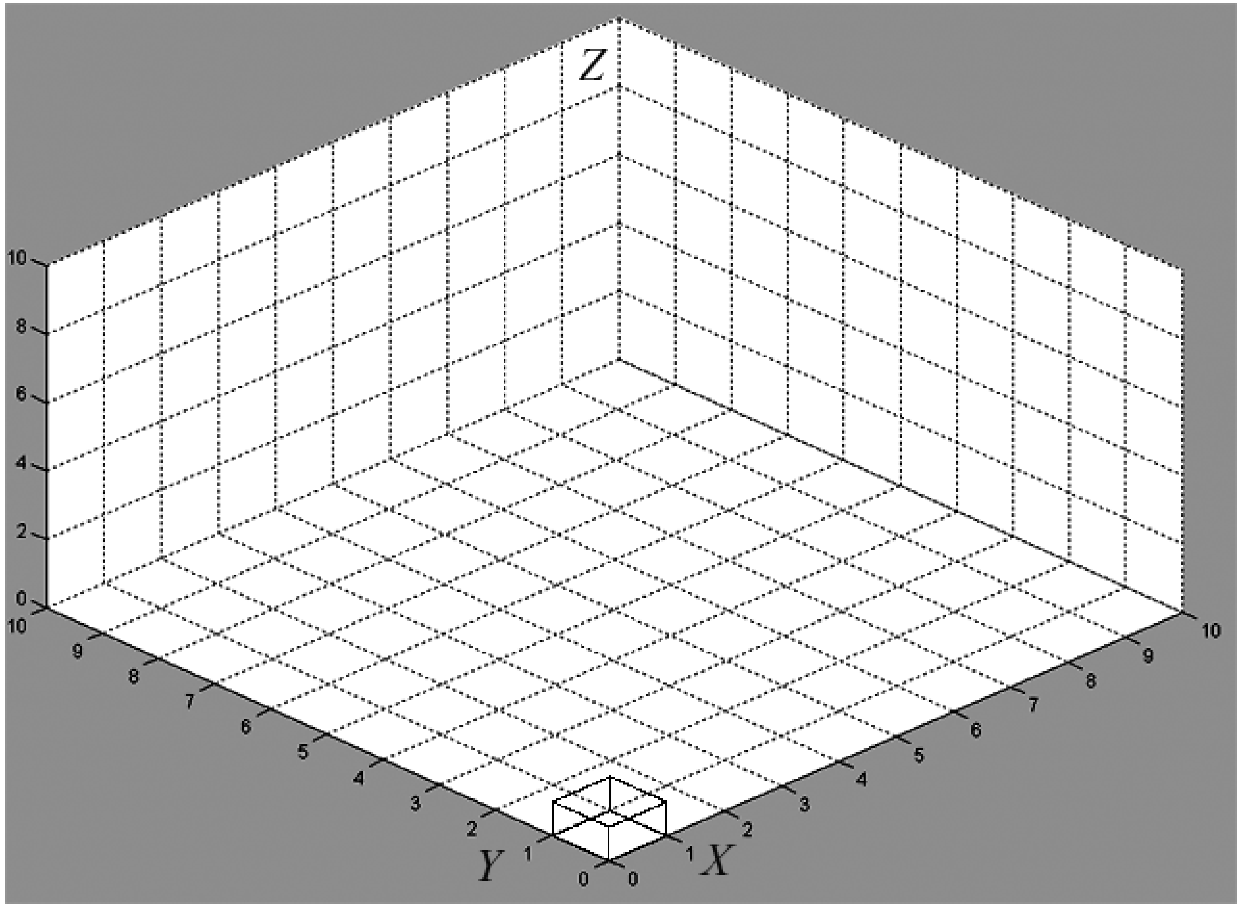
- What did you have learned? Any question you may have?

Appendix: Matlab source code with comments.

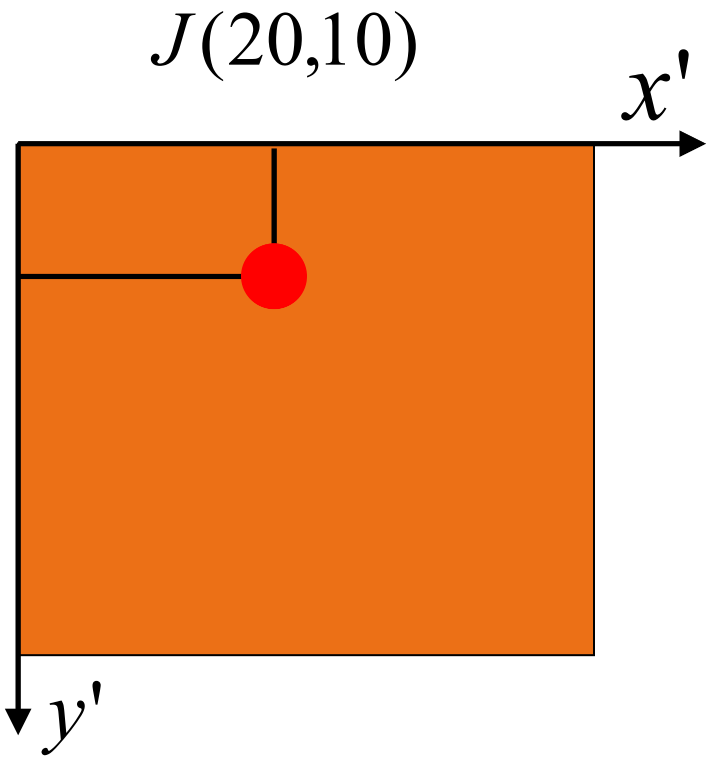
Expected Result (1)



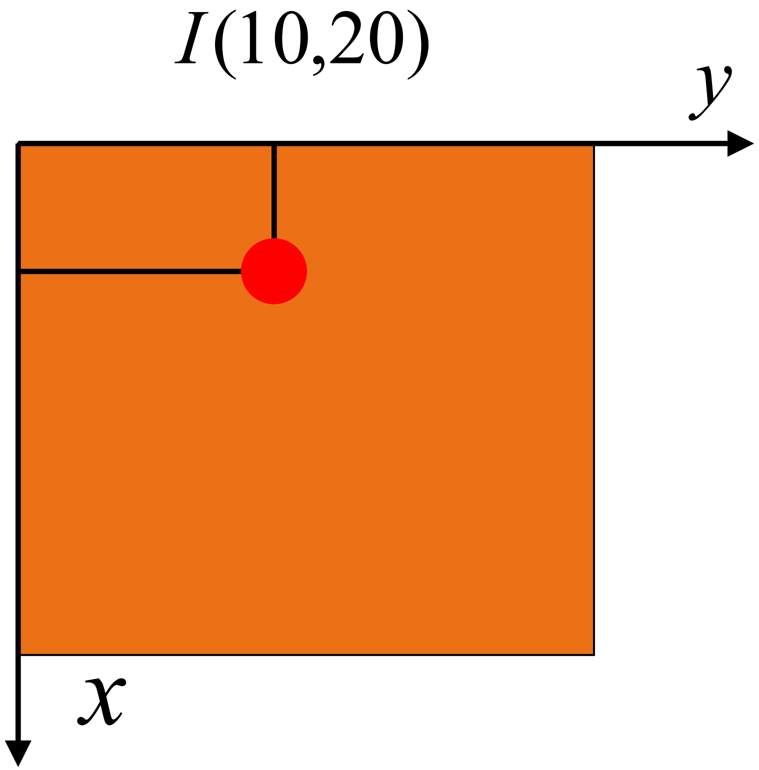
Expected Result (2)



M-file Programming: Matlab 2-D coordinate systems



The 2D coordinate system where the 2D points ("observe.dat") are collected.



The 2D coordinate system used in Matlab for images.

M-file Programming: Show an image

```
I=imread('test_image.bmp'); % read an image into I
[Ix Iy]=size(I);            % the dimension of image I (#row, #column)
figure(1), imshow(I);       % show image I in figure 1
load observe.dat            % read the 2D observation data
[On Ot]=size(observe)       % the dimension of observe data
for i=1:On
    mx=observe(i,1);        % read the y coordinates of each point
    my=observe(i,2);        % read the y coordinates of each point
    for j=mx-2:mx+2         % Mark each point in the image
            for k=my-2:my+2
                I(k,j)=0;
            end
    end
end
figure(2), imshow(I);       % show the marked image in figure 2
```

M-file Programming: Video Creation

T=imread('test_image.bmp');

for i=1:frame_num

- L=T;
- % some processing in L
- Frame(:,:,1)=L; % Red channel
- Frame(:,:,2)=L; % Blue channel
- Frame(:,:,3)=L; % Green channel
- Mo(i)=im2frame(Frame)

end

Movie2avi(Mo,’filename.avi’);